Herausgeber

H. SCHULTE
F. HOFFMANN
R. MIKUT



Berlin | 19. – 20. November 2026

PROCEEDINGS **36. WORKSHOP** COMPUTATIONAL INTELLIGENCE

Karlsruher Institut für Technologie

H. Schulte, F. Hoffmann, R. Mikut (Hrsg.)

Proceedings. 36. Workshop Computational Intelligence

Berlin, 19. – 20. November 2026

PROCEEDINGS **36. WORKSHOP** COMPUTATIONAL INTELLIGENCE

Berlin, 19. – 20. November 2026

Herausgegeben von
H. Schulte
F. Hoffmann
R. Mikut

Karlsruher Institut für Technologie

Impressum



Karlsruher Institut für Technologie (KIT)
Kaiserstraße 12
76131 Karlsruhe

Institut für Automation und angewandte Informatik
www.iai.kit.edu



This document – excluding parts marked otherwise, the cover, pictures and graphs – is licensed under a Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0): https://creativecommons.org/licenses/by-sa/4.0/deed.en



The cover page is licensed under a Creative Commons Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0): https://creativecommons.org/licenses/by-nd/4.0/deed.en

2026

DOI 10.5445/IR/1000197311

Inhaltsverzeichnis

T. Bartz-Beielstein, A. Hinterleitner, N. Pütz, J. Brandt, M. Müller, N. Nadirov **1**
(TH Köln, Everllence SE)
Many-to-Many Recurrent Neural Networks for Predicting Compressor Performance Maps

M. Beichter, M. Turowski, A. Döring, A. Kreusel, N. Friederich, S. Waczowicz, J. A. González Ordiano, A. Iser, J. Hitzel, R. Weschenfelder, M. Rink, V. Hagenmeyer, R. Mikut **17**
(Karlsruhe Institute of Technology, Stadtwerke Karlsruhe GmbH, Universidad Iberoamericana Ciudad de México)
Real-World Short-Term District-Heating Load Forecasting: A Machine-Learning Benchmark Against an Operational Utility Model

L. Schilpp, A. J. Yamachui Sitcheu, T. Lautenschlager, L. Saur, N. Friederich, R. Mikut, S. Allgeier **51**
(Karlsruhe Institute of Technology)
Biomedical Self-Supervised Learning using Sequential and Action-Conditioned Prediction: A Proof of Concept in Histopathology

B. Schwabe, Z. Li, X. Höchst **79**
(Infineon Technologies AG, Technische Universität München)
Safe Reinforcement Learning for Power Converters with Hardware in the Loop

M. Zehner, A. Cavaterra, S. Lambeck **93**
(Hochschule Fulda)
Comparing Extensions of DeePC: A Simulation Study

K. Lange, R. Mikut, A. Kroll **117**
(University of Kassel, Karlsruhe Institute of Technology)
On Interpretability and Scheduling Space Choice for System Identification with Dynamic Takagi–Sugeno Models

T. Pütz, O. Nelles **141**
(University of Siegen)
Joint Structure and Dynamics Optimization in Local Model Networks with Orthonormal Basis Functions

D. Sytnychenko, T. Bertram **157**
(TU Dortmund)
An Interpretable LLM-Assisted Framework for Passenger Safety Incident Analysis and Mixed Reality Scenario Development in Automated Shuttle

B. Jaster, M. Kohlhase **217**
(Hochschule Bielefeld)
Cold-Start Active Learning on Tabular Data: A Systematic Benchmark

T. Bartz-Beielstein, M. S. Klimm, A. Hinterleitner, A. Prediger . . **241**
(TH Köln)
Foundation Models Versus Machine-Learning Algorithms: Results from a Load-Forecasting Challenge Under EU AI Act Requirements

A. Singh, A. P. Vedurmudi **263**
(Physikalisch-Technische Bundesanstalt)
Synthetic Sensor Data Generation for Sensor Network Metrology Using Large Language Models

Y. Kwak, R. Würth, A. Yamachui Sitcheu, D. Hübschmann, R. Mikut **283**
(Karlsruhe Institute of Technology (KIT), German Cancer Research Center (DKFZ), HIDSS4Health, Heidelberg Institute for Stem Cell Technology and Experimental Medicine (HI-STEM gGmbH), National Center for Tumor Diseases (NCT) Heidelberg, German Cancer Consortium (DKTK), Heidelberg University)
Clustering-Based Rare-Event Detection Beyond Singlets in Flow Cytometry

M. Hieb, V. Lohweg **307**
(Phoenix Contact GmbH & Co. KG. , Technische Hochschule Ostwestfalen-Lippe)
Closing the Loop: Bridging Deep Learning-Based Multiview Defect Detection and Information Fusion for Injection Molding Optimization

T. Dommaschk, M. Nobach, C. Lauhoff, A. Bolender, H. Chapman, Y. Chen, F. Sket, P. D. Lee, T. Niendorf **333**
(Universität Kassel, University College London, Research Complex at Harwell, Royal Melbourne Institute of Technology, IMDEA Materials Institute)
Partikelverfolgung im Schmelzbad von Directed-Energy-Deposition-Bauteilen mittels In-situ-Röntgenbildgebung

M.R. Gevari, H. Schulte **343**
(HTW Berlin, Control Systems Group)
Data-driven Identification of VSC Control Modes in Active Distribution Grids

F. Glatzel, S. Schütte, T. Osterburg, T. Bertram **351**
(Institute for Control Theory and Systems Engineering, TU Dortmund University)
Evaluation of Training Curricula for Collaborative 3D Object Detection Under Communication Latency

Many-to-Many Recurrent Neural Networks for Predicting Compressor Performance Maps

Thomas Bartz-Beielstein[1], Alexander Hinterleitner[1], Noah Pütz[1], Jens Brandt[1], Matthias Müller[2], Nadir Nadirov[2]

[1]Institute for Data Science, Engineering, and Analytics (IDE+A), TH Köln
Steinmüllerallee 1, 51643 Gummersbach, Germany
E-Mail: thomas.bartz-beielstein@th-koeln.de

[2]Everllence SE, Turbolader Entwicklung
Augsburg, Germany

Abstract

Compressor performance maps (CPMs) describe a compressor by a family of speed lines that relate volume flow to pressure ratio. They are used for the design and simulation of turbomachinery. In an earlier study, we treated compressor data for the first time as sequences: every speed line is an input sequence for a bidirectional many-to-many recurrent neural network (RNN). Three prediction tasks were discussed: (i) T-inter, interpolation within a map, (ii) T-extra, extrapolation at its boundaries, and (iii) T-new, extrapolation to an entirely new map. The first two were experimentally studied, while the third was only briefly shown as a proof of concept on a single held-out configuration. This paper presents the first systematic evaluation of T-new on industrial data: a leave-one-map-out study over 15 CPMs from two compressor series, A and B, in which a fresh RNN is trained on the pooled speed lines of all remaining maps of the same series and predicts the complete held-out map. The two series are never mixed during training. At the original protocol of 10 seeds and 5000 epochs, the models reach a mean absolute percentage error (MAPE) of 2.3 % on the six-map A series and 0.9 % on the nine-map B series, practically unchanged from a reduced budget of 3 seeds and 1000 epochs. As expected,

errors concentrate on the top speed line of each map, whose measurements the data owner considers least reliable. Two additional models of the structure between the speed lines were not able to improve on the simple, plain line model. A complementary cross-series task T-series, training on one series and predicting the other, raises the error to roughly 7 % MAPE in both directions. Interestingly, more training compute makes it worse: the models transfer the shape of a map but not its series-specific calibration (overfitting). All experiments run in the open-source package `spotoptim`, validated bit-identically against the original study code, and every number and figure in this paper is computed from the persisted artifacts.

Keywords: compressor performance maps, recurrent neural networks, sequence modeling, leave-one-map-out, cross-series transfer, reproducibility

1 Introduction

Compressor performance maps (CPMs) describe the behavior of a compressor through families of speed lines that relate volume flow to pressure ratio. They are a standard tool for the design, control, and simulation of turbomachinery [1, 2]. The generation of a CPM is expensive: a typical CPM comprises 20 to 150 steady-state operating points from engine tests or computational fluid dynamics (CFD) simulations [3], and physics-based calibration involves many interdependent parameters [4]. Machine-learning models, feed-forward networks, and support vector machines in particular, were studied as data-driven alternatives [5–8].

In Schulz et al. [9] we proposed a new method that treats compressor data as *sequences*: each speed line becomes one input sequence for a bidirectional many-to-many recurrent neural network (RNN), tuned with surrogate-based hyperparameter optimization. That study distinguished three prediction tasks of increasing difficulty, T-inter interpolation within a map, T-extra extrapolation at the boundaries of a map, and T-new extrapolation to an *entirely new map*. It demonstrated the first two extensively, while T-new was only shown as a proof of concept for a single held-out configuration.

This paper addresses T-new systematically on industrial data. We evaluate *global training*: for each of 15 industrial CPMs from two compressor series, a fresh RNN is trained on the pooled speed-line sequences of all other maps of the same series and then predicts the entire held-out map. Repeating this leave-one-map-out (LOMO) protocol over several random seeds and two computational budgets yields a complete picture of how well a sequence model generalizes to a new compressor configuration. A fourth task, T-series, trains on one series and predicts the other. We ask four research questions (RQ):

- *RQ1 (T-new):* How accurately does a globally trained many-to-many RNN predict an entirely unseen map of the same compressor series? Is the result stable over held-out maps, seeds, and training budget?
- *RQ2 (T-new):* Does modeling the structure *between* the speed lines of a map improve the plain line model on T-new?
- *RQ3 (T-series):* How much of what the model learns on one series transfers to the other, and does more compute help?
- *RQ4 (T-new):* Where in a predicted map do the errors concentrate, in particular on the top speed line, whose measurements the data owner considers least reliable?

2 Background and Data

2.1 Compressor Performance Maps as Sequences

A CPM is measured line by line: at a fixed reduced rotational speed, the flow is throttled through a range of stable operating points, producing one *speed line*. The map is a family of such lines. Because compressor behavior depends on the rotor-stator configuration and the casing geometry, maps of different compressors in the same series are related but not identical, which makes predicting an unseen configuration from its siblings plausible, yet non-trivial [9].

Points along a speed line are not independent: the pressure ratio falls with increasing flow in a smoothly varying pattern. RNNs process such ordered data by carrying a hidden state along the sequence [10, 11]. *Bidirectional*

variants [12] add a backward pass so that every prediction can use the whole line as context, and a *many-to-many* configuration [13] emits one output per input point, mapping a measured speed line to the predicted curve in a single pass. Schulz et al. [9] showed that this sequence view outperforms feed-forward networks and support vector machines for T-inter and T-extra, with the plain Elman RNN as the most reliable member of the tuned family of RNN, long short-term memory (LSTM), and gated recurrent unit (GRU) models.

All experiments are implemented with the package `spotoptim` [14]. It is the successor of `spotPython` [15], which was used in the original study [9]. Its `torch` extra is an exact port of the original sequence-training code (see the section "Method").

2.2 Data

The dataset was provided by Everllence SE (formerly MAN Energy Solutions) within the collaboration described in Schulz et al. [9] and remains confidential. Only summary statistics and normalized figures are published. It contains 571 stable operating points on 105 speed lines from 15 CPMs of two compressor series: six maps (49 lines) of the compressor series A and nine maps (56 lines) of the compressor series B. Only speed lines with at least five points are retained.

Each operating point carries two input features, the speed-band identifier of its line (a numeric input so that the model can distinguish lines) and the reduced total volume flow, and the target, the total pressure ratio. This orientation is the reverse of the public single-map dataset of the original study [9], so the two settings are not numerically comparable. In the published figures, maps are labeled A_1 to A_6 and B_1 to B_9 in ascending order of their mean prediction error. Both axes are normalized by the per-map maxima of the measured points.

3 Method

3.1 Global Training with LOMO Validation

Training and prediction always stay *within one series*: at no point is a model trained on one series and used to predict a held-out map of the other. *Cross-series* prediction is the separate task T-series (see the section "Cross-Series Prediction"). *Global training* means pooling the speed-line sequences of several maps of the same series, as opposed to training on a single map. Task T-new is evaluated by a LOMO loop per series. For each map M of series S:

M is withheld entirely. The speed lines of *all other* maps of S are pooled into one training set on which a fresh model is trained, five maps for a compressor A holdout and eight for a compressor B holdout. The trained model then predicts every speed line of M, and only that map. Each map is the holdout once, giving 15 folds, and every fold is repeated with several random seeds, each training a freshly initialized model, so that per-map results come with a seed-to-seed spread.

3.2 Model and Training Budgets

The model is the `ManyToManyRNN` of spotoptim at its defaults, identical to the architecture in Schulz et al. [9]: a single bidirectional Elman RNN layer with 256 hidden units processes the packed variable-length sequences [16], followed by a fully connected layer with 256 units, a rectified linear unit (ReLU) activation, and a linear output head. Training uses full-batch gradient descent, the Adam optimizer with learning rate 10^{-4}, and a mean squared error loss on the padded tensors, faithful to the original study code.

Two budgets are used throughout. The *reduced* budget, 3 seeds $\times$ 1000 epochs, re-runs the complete study in minutes on a notebook. The *full* budget is the protocol of the original study, 10 seeds $\times$ 5000 epochs, run on CPU on two Apple M2 Ultra machines for bit reproducibility.

3.3 Metrics and Reproducibility

For a speed line with n points, predictions $\hat{y}_i$, and measured values y_i, the error measure is the mean absolute percentage error (MAPE),

$$\mathrm{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| . \tag{1}$$

It is computed per speed line and averaged over the lines of the held-out map. All tables report the mean and the standard deviation over all (map, seed) runs.

All randomness is seeded per (fold, seed) combination, which makes the CPU pipeline exactly reproducible. The spotoptim sequence stack was validated against the original study helpers by an equivalence check that runs one complete fold with both implementations and asserts *exact* equality of loss history, predictions, and metrics (`BIT-IDENTICAL`)[1].

4 Within-Series Results on T-new

4.1 Accuracy of Unseen-Map Prediction (T-new, RQ1)

With the plain line model, unseen compressor B maps are predicted with a mean MAPE of 0.89 % and unseen compressor A maps with 2.30 % at the full budget. The reduced budget gives 0.89 % and 2.29 %: LOMO prediction is *budget-stable*, and the reduced-budget numbers of the accepted abstract are confirmed.

The gap between the two series is consistent across held-out maps (Figure 1): the compressor A maps range from 1.52 % to 3.25 % mean MAPE, the compressor B maps from 0.58 % to 1.66 %. A compressor B holdout leaves eight maps to pool, a compressor A holdout only five, and the seed-to-seed spread is larger on compressor A (standard deviation 0.58 versus 0.35 percentage points): with fewer pooled sequences the fit is less stable. The errors sit in the line ends

[1] Every number, table, and figure of this paper is computed from the persisted artifacts by a render-time analysis script that aborts if a recomputed aggregate disagrees with a persisted summary.

Table 1: Task T-new, within-series LOMO prediction: mean ± standard deviation of the MAPE (%) over all (map, seed) runs, per series and budget (reduced: 3 seeds × 1000 epochs; full: 10 seeds × 5000 epochs). The line model is the plain many-to-many RNN of the section "Model and Training Budgets"; the context RNN and the two-orientation ensemble are introduced in the section "Structure Between the Speed Lines".

	series A		series B	
Model	reduced	full	reduced	full
line model	2.29 ± 0.50	2.30 ± 0.58	0.89 ± 0.31	0.89 ± 0.35
context RNN	2.13 ± 0.52	2.30 ± 0.56	0.98 ± 0.43	0.98 ± 0.44
ensemble	3.23 ± 1.25	2.68 ± 0.85	1.25 ± 0.70	1.39 ± 1.07

and in the spacing between adjacent lines. The worst compressor B case, B_9, shows the main failure mode: its lines lie partly outside the region covered by the pooled maps, so the model extrapolates in feature space as well as across configurations.

4.2 Error Anatomy: the Top Speed Line (T-new, RQ4)

The top speed line of a map, measured at the highest reduced speed, carries larger measurement uncertainty than the other lines. Figure 2 resolves the prediction error by the position of the line within the map, counted from the top: in both series the top line is the worst-predicted line on average, and the error decreases nearly monotonically towards the bottom of the map. On compressor B the top line reaches 1.65 % MAPE against 0.75 % for all other lines, a factor of 2.2. On compressor A the values are 3.38 % against 2.20 % (factor 1.5), and the bottom line is the most accurate in both series (0.65 % and 1.38 %). The per-map dots show that the effect is systematic on compressor B (the top line is the worst line in 7 of nine maps) and weaker on compressor A (1 of six maps), where the spread between maps dominates. The line was retained in training and evaluation. The section "Discussion" weighs the alternatives.

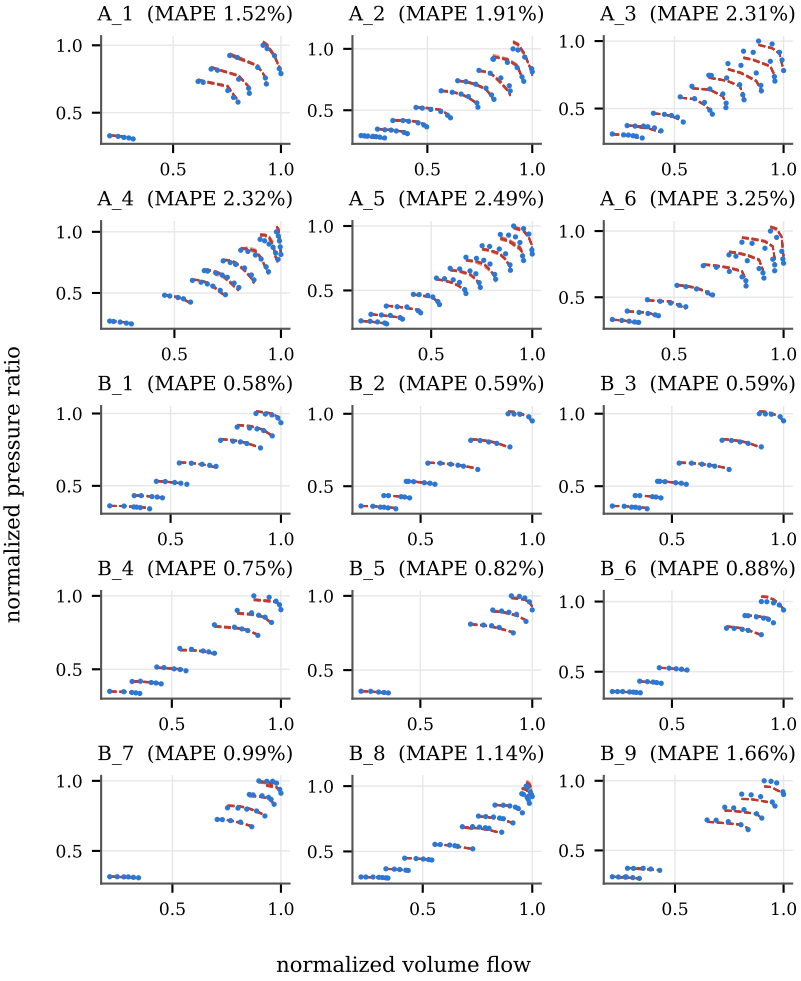


Figure 1: Task T-new, full budget: all 15 held-out maps of the LOMO study, with the measured points (blue) and the predicted curves of every seed (red dashed), ordered by series and mean MAPE (rows 1 and 2: series A, rows 3 to 5: series B). Both axes are normalized by the per-map maxima of the measured points.

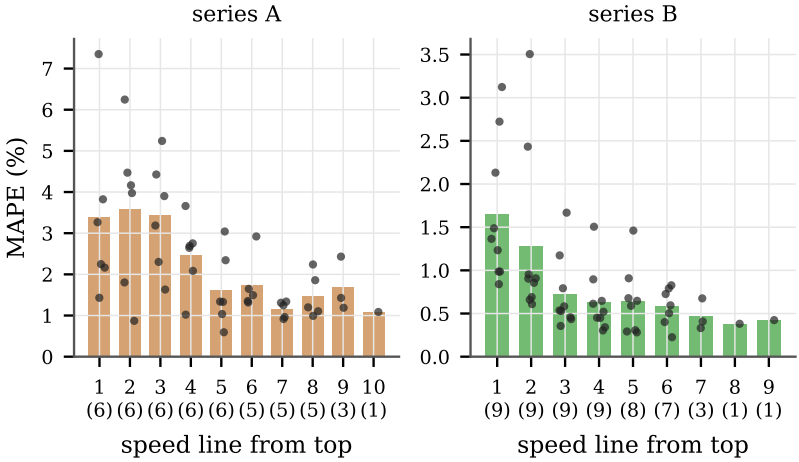


Figure 2: Task T-new: mean MAPE (%) per speed line, counted from the top of each held-out map (full budget). Bars: mean over all maps and seeds with a line at that position; dots: per-map means; in parentheses: number of maps contributing.

5 Structure Between the Speed Lines (T-new, RQ2)

The line model processes every speed line in isolation, so information between the lines of a map flows only through the speed-band input feature. Two approaches to the structure *between* the lines were compared with it on T-new under the identical LOMO protocol and both budgets. The *context RNN* (`MapContextRNN` in spotoptim) adds an outer bidirectional RNN over the line embeddings in speed-band order and concatenates its context vector to every point, so each line prediction sees the rest of the map. The *two-orientation ensemble* trains a second `ManyToManyRNN` on the sequences formed *across* the speed lines at seven interpolated flow stations, interpolates its predictions back to the measured points, and averages them with the line model.

Table 1 shows that neither approach improves on the line model. At the full budget the context RNN ties the line model on compressor A (2.30 % versus 2.30 %) and trails it slightly on compressor B (0.98 % versus 0.89 %). Its small compressor A advantage at the reduced budget (2.13 %) disappears with more

Table 2: Cross-series prediction T-series: mean ± standard deviation of the MAPE (%) over all (map, seed) runs per direction and budget; last row: within-series prediction of the same target series with the line model (Table 1).

	A → B		B → A	
Model	reduced	full	reduced	full
line model	7.40 ± 0.58	9.73 ± 1.17	7.05 ± 0.58	7.43 ± 0.98
context RNN	7.76 ± 1.09	8.78 ± 0.96	6.94 ± 1.10	8.45 ± 1.50
within series	0.89 ± 0.31	0.89 ± 0.35	2.29 ± 0.50	2.30 ± 0.58

seeds and epochs. The ensemble is clearly worse (2.68 % and 1.39 %), because the transposed reading alone is much weaker (3.91 % and 2.59 %): the direction across the lines carries little signal that the speed-band feature does not already provide. The hyperparameters of the original study, tuned on the public single-map dataset [17], do not transfer to global training (8.62 % and 4.13 % at the full budget).

6 Cross-Series Prediction (T-series, RQ3)

In task T-series a model is trained on *all* maps of one series, without holdout, and predicts *every* map of the other series, in both directions, with one model per direction and seed and the configuration and budgets of the section “Method”.

Table 2 shows the base finding: across series, the line model degrades to 7.40 % (compressor A → compressor B) and 7.05 % (compressor B → compressor A) MAPE at the reduced budget, factors of 8.3 and 3.1 above within-series prediction of the same target series. The degradation is uniform over the maps. Its main driver is feature space: the reduced volume flows of the two series occupy largely disjoint ranges while speed bands and pressure ratios overlap, so every model extrapolates far beyond its training range in the flow direction. Shape and ordering of the speed lines are still captured, but level and spread, the *calibration*, no longer match.

The full budget *worsens* cross-series prediction with the line model to 9.73 % and 7.43 %. Longer training imprints the calibration of the training series more strongly instead of learning transferable structure. The context RNN shows the same effect (8.78 % and 8.45 % at the full budget) and leads the line model in the compressor A $\rightarrow$ compressor B direction only, without closing the calibration gap.

7 Discussion

RQ1 T-new: unseen maps within a series. Global training lets a bidirectional many-to-many RNN predict entirely unseen maps with 0.89 % (compressor B) and 2.30 % (compressor A) mean MAPE. The LOMO design shows that the result is not an artifact of a favorable holdout: every map of both series is predicted at a comparable level, and the full budget confirms the reduced-budget means.

RQ2 T-new: structure between lines. Neither the context RNN nor the two-orientation ensemble improves on the plain line model within a series: the speed-band input already gives the line model the between-line information it needs, and the direction across the lines carries little additional signal.

RQ3 T-series: transfer across series. Roughly 7 % MAPE in both directions at the reduced budget, and more at the full budget, show that what transfers is the qualitative map structure, not the calibration. More compute does not help. Calibration information from the target series is needed, for example by fine-tuning on a few of its maps. A related question raised by the data owner is whether *pooled* training, on the remaining maps of the own series plus all maps of the other series, would help or hurt the prediction of a held-out map. This experiment was not run for this paper. The cross-series results make a gain unlikely, since the other series contributes sequences from a largely disjoint flow range with a different calibration. It remains future work.

RQ4 T-new: the top speed line. The top line of each map is the worst-predicted line in both series, by a factor of about two on compressor B, and the error decreases towards the bottom of the map. Because the top line is both the

boundary of the map in the speed direction and the line with the least reliable measurements, its error mixes extrapolation and measurement noise, and the two cannot be separated with the present data. We kept the line in training and evaluation so that the numbers remain comparable with the accepted abstract. Excluding it from training, or from training and evaluation, is a cheap follow-up experiment for the data owner to choose.

Threats to validity. The study covers 15 maps of two series of one manufacturer. Series sizes of six and nine maps are small, and the per-map spread on compressor A shows that individual holdouts deviate noticeably from the series mean. The hyperparameters are architecture defaults, and dedicated tuning of the global models was not attempted.

8 Conclusion

This paper presented the first systematic evaluation of the new-map extrapolation task T-new of Schulz et al. [9] on industrial data. Global training over the sibling maps of a compressor series lets a bidirectional many-to-many RNN predict entirely unseen compressor maps at 0.89 % to 2.30 % MAPE, stable over held-out maps, seeds, and budget. Errors concentrate on the top speed line. Modeling the structure between the speed lines yields no benefit on T-new, and cross-series prediction T-series fails on calibration rather than on shape, increasingly so with more compute.

Future work covers (i) the pooled two-series training and the top-speed-line exclusion experiments requested by the data owner, (ii) fine-tuning on a few maps of the target series, (iii) dedicated hyperparameter tuning of the global models with spotoptim, and (iv) the LSTM and GRU variants of the original study.

Data, Code, and Acknowledgements

The compressor data are the property of Everllence SE and are confidential, and the map labels are anonymized. The package `spotoptim` is open source

(AGPL-3.0)[2]. The experiment artifacts, the analysis script, and the source of this paper are available from the authors. We thank Everllence SE for providing the data and reviewing the abstract.

Competing Interests

The first author develops the open-source package `spotoptim` used in this paper. Two co-authors are employees of Everllence SE, the owner of the dataset. No financial competing interests are declared.

Use of AI Tools

Transparency notice: parts of this paper were researched, drafted, and translated with the support of artificial intelligence. This information is provided voluntarily for complete transparency, as no legal requirement mandates its disclosure.

References

[1] M. Sandberg, "Compressor performance and thermodynamics," *Turbomachinery Magazine*, May 2022. Publisher: MJH Life Sciences.

[2] A. Cortinovis, M. Zovadelli, M. Mercangoz, D. Pareschi, A. De Marco, and S. Bittanti, "Online adaptation of performance maps for centrifugal gas compressors," in *2014 European Control Conference (ECC)*, (Strasbourg, France), pp. 1036–1041, IEEE, June 2014.

[3] K. Joachim, "An enhanced compressor map extension method suited for spool speeds down to 1%," *J. Glob. Power Propuls. Soc.*, vol. 8, pp. 215–226, 2024.

[2] `https://gitlab.git.nrw/thk-f10/spotsevenlab/spotoptim`

[4] X. Llamas and L. Eriksson, "Control-oriented compressor model with adiabatic efficiency extrapolation," *SAE International Journal of Engines*, vol. 10, no. 4, pp. 1903–1916, 2017.

[5] J. Fei, N. Zhao, Y. Shi, Y. Feng, and Z. Wang, "Compressor performance prediction using a novel feed-forward neural network based on gaussian kernel function," *Advances in Mechanical Engineering*, vol. 8, pp. 1–14, 01 2016.

[6] K. Ghorbanian and M. Gholamrezaei, "An artificial neural network approach to compressor performance prediction," *Applied Energy*, vol. 86, no. 7, pp. 1210–1221, 2009.

[7] T. Palmé, P. Waniczek, H. Hönen, M. Assadi, and P. Jeschke, "Compressor map prediction by neural networks," *Journal of energy and power engineering*, vol. 6, no. 10, p. 1651, 2012.

[8] Y. Ying, S. Xu, J. Li, and B. Zhang, "Compressor performance modelling method based on support vector machine nonlinear regression algorithm.," *R Soc Open Sci*, vol. 7, p. 191596, Jan 2020.

[9] R. Schulz, A. Hinterleitner, N. C. Pütz, J. U. Brandt, T. Bartz-Beielstein, and M. Müller, "Automated prediction of compressor performance maps: Surrogate-based optimization with RNNs for enhanced extrapolation and interpolation," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain*, (New York, NY, USA), ACM, 2025.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[13] A. Graves, A. Mohamed, and G. E. Hinton, “Speech recognition with deep recurrent neural networks,” *arXiv: Neural and Evolutionary Computing*, 2013.

[14] T. Bartz-Beielstein, “spotoptim: Sequential parameter optimization with a scipy-style interface,” 2026. Version 3.2.0, available on PyPI.

[15] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, eds., *Hyperparameter Tuning for Machine and Deep Learning with R —A Practical Guide*. Singapore: Springer, 2022.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.

[17] H. Shen, C. Zhang, J. Zhang, B. Yang, and B. Jia, “Applicable and comparative research of compressor mass flow rate and isentropic efficiency empirical models to marine large-scale compressor,” *Energies*, vol. 13, no. 1, 2020.

Real-World Short-Term District-Heating Load Forecasting: A Machine-Learning Benchmark Against an Operational Utility Model

Maximilian Beichter[1], Marian Turowski[1], Andreas Döring[1], Alexander Kreusel[1], Nils Friederich[1], Simon Waczowicz[1], Jorge Ángel González Ordiano[1,3], Andreas Iser[2], Jürgen Hitzel[2], René Weschenfelder[2], Manuel Rink[2], Veit Hagenmeyer[1], Ralf Mikut[1]

[1]Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT)
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
E-Mail: {firstname.lastname}@kit.edu

[2]Stadtwerke Karlsruhe GmbH
Daxlander Straße 72, 76185 Karlsruhe, Germany
E-Mail: {firstname.lastname}@stadtwerke-karlsruhe.de

[3]Instituto de Investigación Aplicada y Tecnología, Universidad Iberoamericana Ciudad de México
Prolongación Paseo de Reforma 880, Mexico City 01219, Mexico
E-Mail: jorge.gonzalez@ibero.mx

Abstract

In urban district-heating networks, operators plan unit dispatch, storage operation, and fuel procurement on the basis of short-term heat-demand forecasts. At the Stadtwerke Karlsruhe GmbH (Germany), this forecast has been generated since 2017 by a compact gray-box model that adjusts the load profile of a reference-day using a 72 h temperature forecast and a weekday-dependent calendar correction. We benchmark this operational reference model against machine-learning models (namely ordinary least-squares, Ridge, and ElasticNet

regression, a multi-layer perceptron, XGBoost, LightGBM, and the tabular foundation models TabPFN and TabFM). On the 2025/26 test season, the best-performing model, TabPFN, reduces the mean absolute error from 14.46 MW of the operational reference model to 10.69 MW, a reduction of 26 %. TabPFN obtains these results without a task-specific weight update, using the training seasons only as in-context examples. In comparison, the regularized linear model ElasticNet reaches 10.78 MW and thus achieves 97.6 % of the improvement achieved by TabPFN, while remaining interpretable: its coefficients reveal a plausible learned calendar correction and temperature forecast dependency. For similar applications, the reported results can serve as a guide to trade implementation effort against accuracy and explainability.

Keywords: district-heating load forecast, short-term heat load forecasting, data-driven forecasting, time-series forecasting, machine learning, tabular foundation models, day-ahead forecasting

1 Introduction

District-heating utilities need short-term forecasts of the aggregated network heat load, that is, the total hourly heat supplied to the district-heating network. These forecasts serve to schedule the following day's heat production [4] and to plan cogeneration and boiler operations [2,23] up to several days in advance [23]. Beyond the operation of the production units, they support the planning of thermal energy storage units [7, 19], ordering fuel such as natural gas [19, 23], and maintenance [19]. Short-term heat load forecasts thus underlie a wide range of operational decisions, from the next day's heat production to storage operation, fuel purchases, and maintenance.

The Stadtwerke Karlsruhe GmbH (Germany), the municipal utility operating the district-heating network of Karlsruhe, uses a forecast of the total hourly heat supplied to the district-heating network for operational planning, so this forecast must be accurate and interpretable at once. To support this planning task, the Stadtwerke Karlsruhe GmbH (Germany) and the Karlsruhe Institute of Technology (KIT) cooperate on developing forecasting models. During this cooperation, they have jointly developed several forecasting models while the

requirements placed on these models have grown. The first of these models originated in 2015 as a parameter-linear model that forecasts the total daily heat supplied to the district-heating network. An evaluation of nonlinear and machine-learning methods confirmed the use of this parameter-linear model, since the evaluated methods did not yield a substantial improvement [24]. An hourly model was deployed in 2017 as an extension of the 2015 model and has remained in operation since then; we refer to it as the *operational reference model*.

By intention, the structure of the operational reference model was kept compact and interpretable: it follows the two main drivers of the heat demand, the outdoor temperature and the social behavior of the consumers, which differs between working days and weekends [6], so that each component has a clear physical or operational interpretation. Moreover, the model requires no dedicated training infrastructure and can be implemented and executed in a spreadsheet with negligible computational effort. During operation, the responsible operators use the forecast resulting from the operational reference model as decision support and review it before applying it in operational planning. Based on operational experience, the operators may adjust the forecast, in particular on public holidays and during school vacations. These adjustments are made case by case, are not part of the model, and are not logged.

In a broad sense, the design of the operational reference model follows the gray-box principle described by Nielsen and Madsen [20, 21]. In their terminology, gray-box modeling combines physical knowledge with data-based statistical modeling. Physical knowledge determines the main model structure, while statistical modeling estimates the model coefficients. In contrast, they characterize black-box modeling as an almost purely data-driven approach. However, the earlier evaluation [24] concerned the daily model. Since then, eight heating seasons of hourly data have accumulated, allowing a broader range of models to be evaluated on a common, longer data basis, and tabular foundation models such as TabPFN [10] and TabFM [15] have emerged that have not yet been evaluated for this task. It therefore remains unclear to what extent retaining the compact gray-box structure impacts the forecast accuracy.

This work therefore benchmarks the gray-box operational reference model against general-purpose machine-learning black-box models (from linear regression to tabular foundation models) using the same information available to the operational reference model at inference. Holding the input information constant allows differences in forecasting performance to be attributed to the model class rather than to differences in data availability at inference or feature sets.

Therefore, this work makes four contributions. First, it benchmarks the operational reference model as a gray-box model that has been in continuous operational use since 2017. Second, it establishes a controlled comparison by providing all benchmark models, commonly viewed as black-box modeling approaches, with exactly the same information as the operational reference model. Third, it evaluates tabular foundation models under these operational input constraints and compares them with conventional machine-learning models that are trained specifically for the task. Fourth, it interprets the best-performing linear model through its coefficients and input-block contributions, including its learned weekday-dependent calendar correction.

The remainder of this paper is structured as follows. Section 2 reviews related work on heat load forecasting. Section 3 describes the operational data, and Section 4 defines the forecast task, the evaluated models, and the evaluation. Section 4 presents the benchmark results, and Section 6 interprets the learned linear model through its coefficients and input-block contributions. Section 7 discusses the operational implications, the trade-off between accuracy and complexity, and the limitations of this work, and Section 5 concludes. Appendix 8 details the benchmark models.

2 Related Work

This section positions the paper within three areas of related work: short-term forecasting of aggregated loads in city-scale district-heating networks and substantial urban subnetworks, gray-box modeling approaches for district-heating load forecasting, and the application of tabular foundation models to district-heating forecasting.

The studies considered in this section focus on forecasting aggregated heat loads at the network or substantial-subnetwork level. They employ data-driven forecasting approaches based on historical operational data, weather information, calendar variables, or combinations thereof. Most studies forecast hourly heat loads over horizons ranging from one hour to several days, with day-ahead forecasting being particularly common.

Forecasting aggregated loads in city-scale district-heating networks. The studies collected in Table 1 investigate aggregated heat load forecasting in municipal district-heating networks and substantial urban subnetworks, including Aarhus [4], Cottbus [25], Flensburg and Munich [26], Helsinki [12], Ljubljana [23], Stockholm [6], Ulm [7, 19], Warsaw [18], and an unnamed European city [2]. The studies in the first part of Table 1 employ black-box modeling approaches in which the forecasting relationship is inferred from observed input and output data without the model structure itself being derived from explicit physical or operational knowledge. The applied methods range from statistical and regression-based approaches, such as linear regression, Generalized Additive Models (GAM), Generalized Additive Mixed Models (GAMM), Autoregression with Exogenous Inputs (ARX), and Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX), to machine-learning approaches, including neural networks, support vector regression (SVR), random forests, boosting, fuzzy models, ensembles, and foundation models (e.g., TabPFN-TS, Chronos-2). These black-box studies are relevant here because they forecast the same quantity with the same model classes that serve as benchmark models in this work. Unlike the present work, however, most of them use a longer load history or measured future weather as inputs and compare their models with generic baselines rather than with a forecasting model in long-term operational use. A study is assigned to the gray-box group only if physical reasoning determines the model structure itself, in the sense of Nielsen and Madsen [20, 21], while measured data are used to identify structural details or estimate model parameters.

Gray-box models in district-heating. Based on the literature reviewed, gray-box models appear to be relatively uncommon in short-term forecasting of the

aggregated load in district-heating networks. They are more commonly used to represent the thermal dynamics of individual buildings or to support the operational optimization of district-heating systems, including the operation of heat generators, thermal storage systems, and network components. One of the few identified forecasting approaches is that of Nielsen and Madsen, who developed a gray-box model for hourly forecasts of aggregated district-heating load with forecast horizons of up to 72 hours [20, 21]. Their approach combines physical knowledge with data-based statistical modeling: physical considerations determine the main model structure, while structural details and model coefficients are identified from measured data. Relations derived from heat transfer through walls and windows, solar gains, and ventilation or infiltration determine how the climate variables enter the model. Unlike detailed physical models, their specific approach uses physically motivated input–output relationships and does not require a detailed representation of individual buildings, physical states, or actual values of physical constants [20, 21].

Following a broader definition of gray-box modeling, the approach proposed by Dotzauer [6] can also be regarded as a gray-box model. The forecasting model combines a knowledge-based decomposition of heat demand into a temperature-dependent and a social component with data-based parameter estimation, using historical measurements to estimate the piecewise-linear temperature function and recurring calendar-based hourly load patterns [6].

The operational reference model by the Stadtwerke Karlsruhe GmbH (Germany) can be regarded as a gray-box model in a broad sense because domain knowledge determines its forecasting structure, whereas historical operational data are used to parameterize the individual model components.

Table 1: Studies forecasting the aggregated load of a district-heating network or of a substantial urban subnetwork, ordered alphabetically. All of them are data-driven; the gray-box studies are listed separately at the bottom. Reported error levels are deliberately not tabulated – network size, climate and load level differ too much across these studies for the numbers to be comparable. All models use an hourly resolution; Spoek et al. [26] additionally consider a 15-minute resolution. The horizon column reports the longest horizon considered in each study.

Reference	Model(s)	Forecast object	Features	Horizon
Black-box forecasting of the aggregated network load				
Bujalski et al. [2][a]	GAM, GAMM (cubic splines, AR error)	Entire municipal network / central CHP plant output	Calendar variables (hour, day of week, day of year), lagged residuals (AR error); no weather inputs.	24 h
Dahl et al. [4]	OLS, MLP, SVR	Entire city-scale network (Aarhus)	Historical heat load, calendar and holiday variables, weather forecasts (air temperature, wind speed, solar irradiance), and measured weather.	38 h
Finkenrath et al. [7]	AdaBoost, RF, FNN, LSTM, Deep DHC ensemble	Substantial urban subnetwork (Ulm; 40 km, 1,100 households, 75 GWh/a)	Historical heat load (lags, rolling means), calendar variables, and weather forecasts (air temperature, dew point, wind, pressure, sunshine, visibility).	72 h
Hua et al. [12]	MLR, ANN	Multiple levels: entire city-scale network (Helsinki) and 5 individual buildings	Calendar variables (season, day, hour), past measured weather (temperature, solar radiation, wind speed), and measured future temperature.	1 h
Kurek et al. [18]	Linear/Ridge regression, ARX, fuzzy model, FFNN, AR-ANN	Multiple levels: entire city-scale network (Warsaw), regional areas, and substations	Historical heat load, calendar variables, total-network load forecast (hierarchical), measured and forecast weather (temperature, wind, humidity, solar radiation, cloudiness).	72 h

continued on next page

Table 1 – continued from previous page

Reference	Model(s)	Forecast object	Features	Horizon
Leiprecht et al. [19]	SARIMAX, AdaBoost, LSTM	Substantial urban subnetwork (Ulm; 40 km, 1,100 households, 75 GWh/a)	Historical heat load (lags, rolling means), calendar variables, and weather forecasts (air temperature, dew point, wind, pressure, sunshine, visibility).	72 h
Potočnik et al. [23]	Linear/robust regression, NN, ELM, RVFL, SVM, GPR	Entire city-scale network (Ljubljana; transmitted heat at plant border)	Historical heat load, calendar variables, past measured weather, weather forecasts (temperature, solar irradiance), and measured future weather.	48 h
Sakkas & Abang [25]	RNN, FFNN, 1D polynomial look-up table	Entire municipal network (Cottbus; 386 km piping, 20 substations)	Historical heat load, calendar variables, past measured weather (humidity, wind, sunshine, rain), and measured future temperature.	24 h
Spoek et al. [26]	TabPFN-TS, Chronos-2, AutoGluon suite, MRRC	Substantial urban subnetwork / district quarter (Munich) and city-scale network (Flensburg)	Historical heat load (rolling context), calendar variables, measured future temperature (primary), and weather forecasts.	24 h
Gray-box forecasting of the aggregated network load				
Dotzauer [6]	Piecewise-linear regression + residual profile	Two municipal/regional networks (Stockholm; System 1: ∼700 MW, System 2: ∼300 MW)	Historical heat load, calendar variables (hour, day of week), and measured future temperature.	168 h

continued on next page

Table 1 – continued from previous page

Reference	Model(s)	Forecast object	Features	Horizon
Nielsen & Madsen [20, 21]	Physical–statistical transfer functions, adaptive recursive least squares	Entire regional transmission network (VEKS Greater Copenhagen, $>$ 8,000 TJ/a)	Calendar variables (day types), ambient temperature and solar radiation (low-pass transfer functions), wind speed, AR error lags	72 h

[a] Bujalski et al. [2] consider only the summer off-season from June to August; the models use calendar variables and lagged model residuals without weather inputs.

Abbreviations: AdaBoost, adaptive boosting; ANN, artificial neural network; AR-ANN, autoregressive ANN; ARX, autoregression with exogenous inputs; ELM, extreme learning machine; FFNN, feed-forward neural network (in Kurek et al. an ANN without autoregressive input); FNN, feed-forward neural network; GAM(M), generalized additive (mixed) model; GPR, Gaussian process regression; LSTM, long short-term memory network; MLP, multi-layer perceptron; MLR, multiple linear regression; MRRC, multi-resolution residual-correction forecaster; NN, neural network; OLS, ordinary least squares; RF, random forest; RNN, recurrent neural network; RVFL, random vector functional link network; SARIMAX, seasonal autoregressive integrated moving average with exogenous regressors; SVM, support vector machine; SVR, support vector regression.

Foundation models for district-scale and city-scale district-heating forecasting. Spoek et al. [26] test TabPFN-TS zero-shot against Chronos-2 and a suite of trained baselines for probabilistic heat load forecasting on two German networks: an expanding mixed-use district in Munich as the main proprietary case, and the city-scale Flensburg network with open data as a transfer check. Both zero-shot models outperform every trained baseline in the day-ahead setting, with Chronos-2 achieving the lowest aggregate error and TabPFN-TS providing better-calibrated prediction intervals than Chronos-2. Their baselines are, however, generic models fitted per network rather than a forecasting model in operational use, and the reported accuracies assume perfect future weather, with a separate sensitivity analysis quantifying the degradation under archived numerical weather forecasts.

Contributions.

- **A benchmark against a long-running operational gray-box model.** Previous studies have compared newly developed forecasting models with commercial forecasts used in district-heating operation [4,25]. In contrast, the present work uses a documented gray-box model, the operational

reference model, whose forecasting structure and inputs are physically and operationally interpretable. The model has remained in operation without reparameterization since 2017. Its performance is assessed over eight heating seasons, and its unchanged operational configuration serves as the reference in the held-out 2025/26 benchmark.

- **A controlled comparison under a common information set.** All benchmark models receive the same operational input information: the measured heat load and plant-level outdoor temperature of a reference-day, the operational 72 h outdoor-temperature forecast, and calendar information. Unlike time-series forecasting approaches that use a longer historical load context, this setup holds the available information constant across model classes.

- **Tabular foundation models under operational input constraints.** Complementing recent work on time-series foundation models for district-heating forecasting [26], this work evaluates TabPFN and TabFM as tabular regression models using only the compact input vector of the operational forecasting system. Their pretrained weights remain frozen, and no task-specific weight updates are performed.

- **Interpretation of the best-performing linear model.** The best-performing linear model is examined through its coefficients and input-block contributions. This analysis shows how the model combines the reference-day profile, temperature forecast, and calendar information over the forecast horizon.

3 Data

All results in this work are derived from the operational data of the Karlsruhe district-heating network. This section first introduces the network and its load–temperature relationship, then describes the sources of the data, and finally explains how the raw record is turned into forecast samples and split chronologically into training, validation, and test seasons.

The network. The Karlsruhe network, operated by the Stadtwerke Karlsruhe GmbH (Germany), is among the larger district-heating systems in Germany, with roughly 240 km of piping in 2025 [27] supplying more than 55,000 residential and commercial units [28]. For 2025, the Stadtwerke Karlsruhe GmbH (Germany) reports 885.8 GWh fed into the network and 778.8 GWh delivered to customers, corresponding to distribution losses of 12.1 % [27, 28]. In 2025, the MiRO refinery supplied the largest share of heat to the Karlsruhe district-heating network at 57.9 %, followed by the EnBW Rheinhafen steam power plant at 26.6 % and the Schwarz Produktion paper mill in Maxau at 7.0 %. The Stadtwerke Karlsruhe GmbH (Germany) generated the remaining 8.5 % in its own facilities [27].

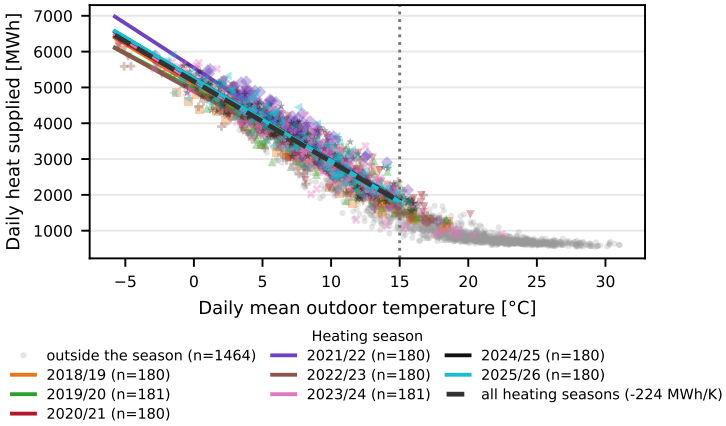


Figure 1: Daily heat supplied against daily mean outdoor temperature, with one point per complete day over the entire record. Both quantities are calculated from exactly 24 hourly values; thus, the daylight-saving days with 23 or 25 values are left out, as is the 2017/18 heating season. The dotted line marks 15 °C, above which the load approaches an approximately temperature-independent base level. Colored lines are per-season least-squares fits below that threshold, the dashed line the fit pooled over all seasons.

Sources. The first source is the measurement record. It is the hourly operational log of the network from November 2017 to April 2026, recorded by the Stadtwerke Karlsruhe GmbH (Germany): the total heat supplied to the network and the outdoor temperature measured at the plant site. The heat supplied to the

network is governed by the outdoor temperature. Figure 1 shows the daily load against the daily mean temperature for the full year, including both the summer months and the heating season. Below approximately 15 °C, the relation is almost linear; above it, the daily heat supplied levels off at a temperature-independent base level associated with domestic hot-water supply and network losses. The per-season fits are similar, so the load–temperature relationship underlying the operational reference model remains stable across the record. The second data source is the 72 h air-temperature forecast, a product provided to the Stadtwerke Karlsruhe GmbH (Germany) by an external provider.

From raw record to samples. Forecast samples are derived from these data. A calendar day is considered complete if it contains exactly 24 hourly values. Days containing 23 or 25 hourly values occur at the daylight-saving-time (DST) transitions and are excluded. A five-day window (reference-day, forecast-issue day, and the three-day horizon) becomes a sample if its days are consecutive and complete, if its forecast day falls in the heating season (i.e., October–March), if the temperature forecast exists, and if all 73 target values are present. Since the first season 2017/18 is incomplete, it is not considered. This procedure yields 1369 samples across the eight heating seasons from 2018/19 to 2025/26. Figure 2 resolves this count by day and gives the reason for every exclusion: the recurring gaps at the end of March and at the turn of October and November are forecast days whose five-day window contains a daylight-saving changeover. The remaining days are excluded for missing temperature forecasts.

Split. The split is chronological and cuts along season boundaries. The last season 2025/26 is considered the test season (164 forecast days), the second-to-last season (2024/25) the validation season (168 forecast days), and the six seasons from 2018/19 to 2023/24 the training seasons (1037 forecast days).

4 Experimental Setting

This section defines the forecast task, the operational reference and benchmark models, and the metrics used for their evaluation.

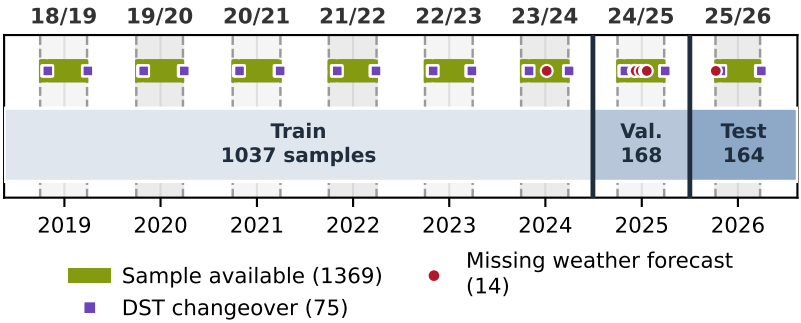


Figure 2: Every day of the eight heating seasons the benchmark models use, classified by whether it yields a sample and otherwise by the reason for its exclusion, with the chronological split on the same axis. It shows the days that yield no sample and those excluded from the training data for other reasons.

4.1 The forecast task

Each forecast day yields a five-day window. The 129-dimensional feature vector contains 25 hourly values of the reference-day heat load, 25 corresponding values of the outdoor temperature measured at the plant site, 72 hourly values of the air-temperature forecast, and a seven-dimensional one-hot encoding of the weekday of the first forecast day. Figure 3 illustrates this composition. The target is the 73 hourly values of the horizon.[1] We consider the task described above as a direct multi-output regression from a fixed feature vector onto the 73 hourly values of the horizon, rather than as an auto-regressive continuation of the load series. The reason is comparability: the operational reference model uses the reference-day two days before the forecast day, an air-temperature forecast, and calendar information, but no additional load history. For this reason, every benchmark model also receives the same inputs.

[1] For compatibility with other systems, the horizon spans 73 h, from 00:00 of the forecast day to 00:00 of forecast day +3, sampled at the 73 grid points that include both midnight endpoints (and thus also the last hour of the issue day); the temperature forecast therefore covers hours 1 . . . 72. Every value carries the timestamp at which its interval *ends*, so grid point 0 is the hour from 23:00 to 00:00 and thus the closing hour of the forecast-issue day. Grid point 0 has no temperature forecast of its own; the operational reference model reuses the value of hour 1 for that one input, while the benchmark models emit it as one of their 73 outputs.

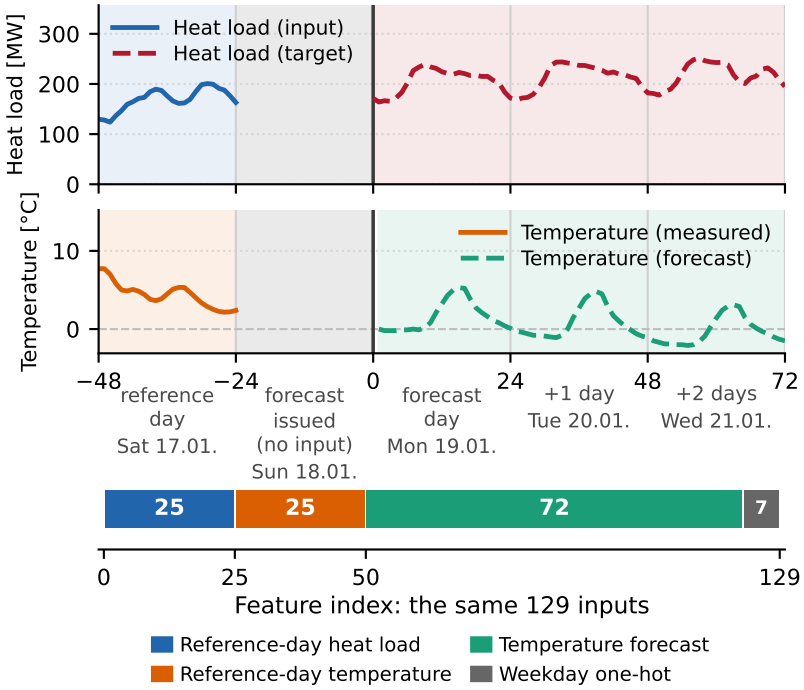


Figure 3: The forecast task. *Top and center:* one window from the test season, forecast day Monday 19 January 2026, with hour 0 at 00:00 of that day, so the reference-day spans hours $-48 \ldots -24$; the gray band marks the issue day, which is not an input. *Bottom:* the same quantities as the 129-dimensional input vector. The feature index is the axis along which the coefficient matrix of Figure 5 (a) is read. In this case, the binary feature for the weekday is Monday.

This is also the reason why time-series foundation models are absent from the set of benchmark models. Chronos-2 [1], TimesFM 3.0 [5], and TabPFN-TS [11] forecast from a long history of the target series. Supplying that history would provide them with an input that the operational reference model does not have. It would further require the issue day to be masked and a decision on whether the history carries measured, forecast, or corrected temperature.

4.2 Operational reference and benchmark models

Table 3 in the Appendix lists all benchmark models, the operational reference model, and the naive baselines with their implementation and preprocessing.

The *operational reference model* always denotes the gray-box model in operation, evaluated in its deployed configuration; for this work, we re-implemented it from the operational tool with its current parameters. The operational reference model starts from the measured load profile of the reference-day and adjusts it hour by hour in two steps. First, a temperature correction shifts each hourly value in proportion to the difference between the forecast air temperature and the outdoor temperature measured at the same hour on the reference-day. Above 15 °C, the load is treated as independent of temperature. Second, a calendar correction maps the weekday of the reference-day onto the weekday of the forecast day using fixed hourly weekday corrections. The colder it is, the more weight this correction receives. Three scalar parameters are involved: a temperature factor, a constant correction of the temperature forecast, and a weight for the calendar term. These parameters and the calendar correction are estimated from operational data before deployment in 2017. Importantly, the operational reference model is not refitted using the data considered in this work.

The *naive baselines* repeat the load of the reference-day or the heat load of the same day one week earlier and thus deviate from the equal input data approach for broader comparability. In contrast to the operational reference model, all benchmark models are black-box models in the sense of Nielsen and Madsen [20, 21]: they learn the forecasting relationship from data, without a structure derived from physical or operational knowledge. The benchmark models are three linear

regressions (ordinary least squares, *OLS*; *Ridge*; and *ElasticNet*), a simple neural network (a multi-layer perceptron, *MLP*), and two tree-based models (extreme gradient boosting, *XGBoost*; and light gradient-boosting machine, *LightGBM*). The three linear models differ in their regularization: OLS uses none, Ridge uses an L_2 penalty on the coefficients, and ElasticNet uses a combination of L_1 and L_2 that drives some coefficients to exactly zero. In addition, we evaluate two tabular foundation models: the tabular prior-data fitted network *TabPFN* and the tabular foundation model *TabFM*. Although their weights stay frozen, they hold all 1037 training days in their context at inference time. They do not receive more information about the district-heating network, since all fitted models draw on the same training days. However, these models use the information differently: instead of condensing it into fitted parameters, they combine the full training set with a prior learned during pretraining. Supplying the training days as context can therefore be regarded as their form of training. Where Table 2 marks a benchmark model *(tuned)*, its hyperparameters are selected based on its MAE on the validation season.

4.3 Evaluation

One forecast day is one sample and carries the $H = 73$ hourly values of its horizon. The metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are first computed separately for each forecast day over its 73 target values. The values reported in Table 2 are then obtained by averaging the daily metric values across all forecast days of the evaluated season. For a day with forecast $\hat{y}$ and measurement y, they are defined as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{H}\sum_{t=0}^{H-1}|\hat{y}_t - y_t|, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{H}\sum_{t=0}^{H-1}(\hat{y}_t - y_t)^2}, \\ \mathrm{MAPE} &= \frac{100}{H}\sum_{t=0}^{H-1}\left|\frac{\hat{y}_t - y_t}{y_t}\right|. \end{aligned} \tag{1}$$

Hyperparameters are selected by MAE on the validation season (Table 4 in the Appendix), and the scaler is fitted on the training seasons alone. No benchmark model is fitted or selected on the test season. All results on the test season reported in Table 2 are based on its 164 forecast days.

5 Results

We first examine the operational reference model over the eight available seasons and then compare the benchmark models on the test season.

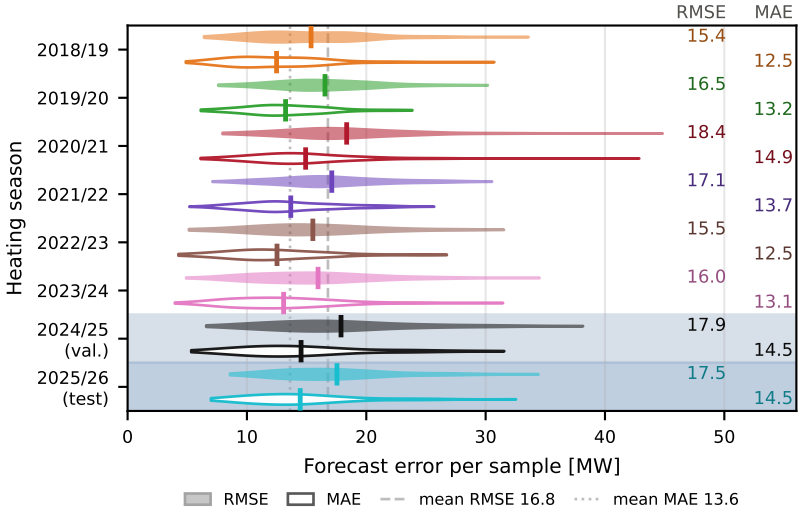


Figure 4: Per-day error of the operational reference model across the eight available seasons. Color identifies the season; the filled violin shows the RMSE and the outlined one the MAE; the vertical line within each violin indicates the mean. For the 2025/26 test season, these means correspond to the values of the operational reference model reported in Table 2. The vertical lines are the unweighted season means, against which a drift would show.

The operational reference model has held its level for eight seasons. The operational reference model was established in 2017 and has not been refitted since. Because its parameters were determined before the evaluation

period, every season can be seen as a valid test season for this model. Figure 4 shows the errors across the different seasons. Its seasonal mean MAE ranges from 12.5 MW to 14.9 MW, without an apparent temporal trend over the years. Differences between seasons are reflected in both the mean and the median of the daily errors.

Table 2: Forecast error on the held-out 2025/26 test season comprising 164 forecast days. MAE and RMSE are reported in MW, and MAPE in percent. Each value is the mean of the corresponding daily metric across the test season. The best value in each column is shown in bold. The final two blocks report the operational reference model and the naive baselines, respectively.

Model	MAE ↓	RMSE ↓	MAPE ↓
ElasticNet (tuned)	10.78	13.27	7.73
Ridge (tuned)	10.87	13.36	7.80
LightGBM (tuned)	11.16	13.89	7.75
XGBoost (tuned)	11.30	13.94	7.88
MLP (tuned)	11.74	14.37	8.17
Linear regression (OLS)	11.90	14.60	8.59
TabPFN (v3)	**10.69**	**13.26**	**7.43**
TabFM (v1.0)	10.97	13.55	7.61
Operational reference model	14.46	17.54	10.67
Naive baseline: reference-day	24.05	28.49	16.52
Naive baseline: previous week	29.90	34.47	20.67

Benchmark against the other models. Across the 164 days of the test season, every benchmark model outperforms the operational reference model (see Table 2), reducing the MAE from 14.46 MW to as low as 10.69 MW, corresponding to an improvement of up to 26 %. TabPFN achieves the lowest MAE, followed closely by the ElasticNet model (10.78 MW). Ridge and TabFM also achieve similar error levels, with MAEs of 10.87 MW and 10.97 MW, respectively. LightGBM (11.16 MW) and XGBoost (11.30 MW) follow; in MAE and RMSE, both gradient-boosting models are worse than the regularized linear models. The MLP, at 11.74 MW, ranks second to last among the benchmark models, ahead only of ordinary least squares (11.90 MW). The operational reference model, in turn, performs clearly better than both naive baselines.

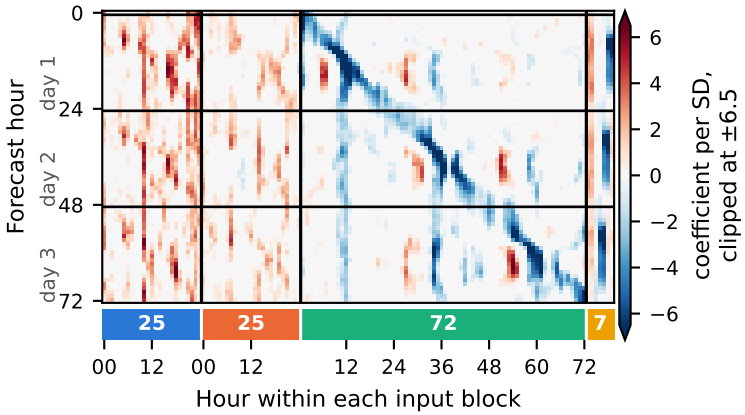
A linear model reaches almost the whole gain. ElasticNet exceeds the MAE of TabPFN by only 0.09 MW. Relative to the operational reference model, ElasticNet retains 97.6 % of the MAE reduction achieved by TabPFN, corresponding to 3.68 MW of the 3.77 MW reduction.

6 Interpretation of the ElasticNet Model

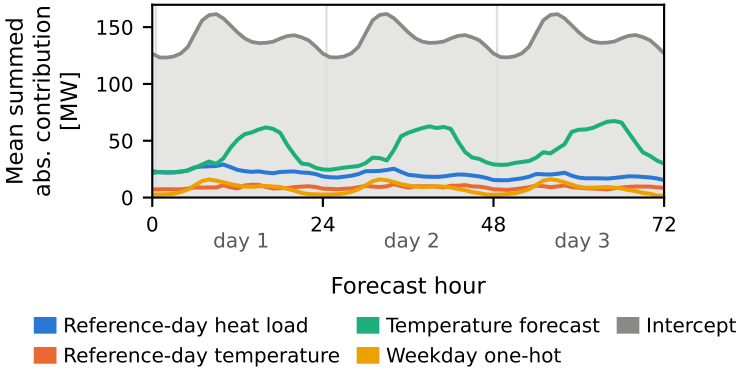
The three linear models differ mainly in their type of regularization. We examine ElasticNet in more detail because it achieves the lowest test-season MAE among the linear models and provides a sparse coefficient matrix. Its L_1 penalty sets coefficients exactly to zero, which makes it the sparsest of the three linear models with 58 % of the 9417 coefficients set to zero. We look at ElasticNet from two angles (Figure 5): the coefficient matrix and the mean absolute contribution of each input block over the horizon.

The coefficient matrix (see Figure 5a) reflects the structure of the inputs. Within the 72 temperature-forecast input columns, the largest coefficients form a near-diagonal band: each forecast output hour is associated primarily with the temperature forecast for the same hour and the adjacent hours $t-1$ and $t+1$. The division of this band into three visible segments results from the day boundaries in the forecast horizon. Additional off-diagonal coefficients link the midday temperature forecast of the first day to outputs on the second and third days, and the corresponding inputs of the second day to outputs on the third day. The final seven input columns represent the weekday block and encode a learned calendar correction. By contrast, the coefficients associated with the reference-day load and temperature are more dispersed and do not exhibit an equally clear temporal pattern.

Viewed over the forecast horizon (see Figure 5b), which shows the mean contribution of each block over the test season, the temperature forecast contributes most from late morning into the afternoon of each day and least at night, while the calendar correction counts less at night than by day. The reference-day load and the reference-day temperature contribute less.



(a) Coefficient matrix, with one row per forecast output hour and one column per input feature. Color indicates the coefficient associated with a one-standard-deviation change. Values are clipped symmetrically at the 99th percentile of the absolute coefficient values.



(b) Mean absolute contribution of each input block in MW at every output hour, averaged across the test season. Contributions are obtained by summing the absolute contributions of the inputs within each block. The gray curve shows the intercept, which corresponds to the forecast horizon profile when all standardized inputs are zero. It therefore represents a baseline level rather than an input contribution.

Figure 5: Interpretation of the selected ElasticNet model.

The learned weekday pattern. The seven columns of the weekday block encode a calendar correction. Figure 6 illustrates the relationship among weekdays based on the mean daily load profile for each weekday over the six training seasons; they serve as a reference for the correction shown in Figure 7. The working days show only small differences of a few MW, whereas Saturday and Sunday profiles remain below the mean profile across all days. This difference is most pronounced during the morning peak, which is substantially flatter on weekends.

Figure 7 shows the contribution of the first forecast day feature. It is obtained by applying the fitted standardization to the one-hot vector for the first forecast day d and multiplying the resulting vector by the coefficient columns of the weekday block. This contribution is added to the intercept. The resulting pattern in Figure 7 therefore resembles the final seven columns of the coefficient matrix in Figure 5a but is not identical to them as they are individual coefficients. Because setting one weekday in the one-hot encoding simultaneously determines the other six entries, an individual column cannot be interpreted in isolation. Only the contribution of the block as a whole is therefore meaningful. The visualized contributions in Figure 7 show how the model adjusts the reference-day-based forecast to the weekday structure of the forecast horizon. When the reference-day is a Saturday or Sunday, the subsequent working days are corrected upward toward the working-day profile. Conversely, forecast hours falling on a Saturday or Sunday are corrected downward. For transitions between working days, the correction remains close to zero.

7 Discussion

We discuss the implications of the results for a utility like the Stadtwerke Karlsruhe GmbH (Germany), how much model complexity is required to achieve the observed accuracy gains, and which aspects are not covered in the present work.

Operational implications of the results. All benchmark models beat the operational reference model; the best, TabPFN, lowers the MAE from 14.46 MW

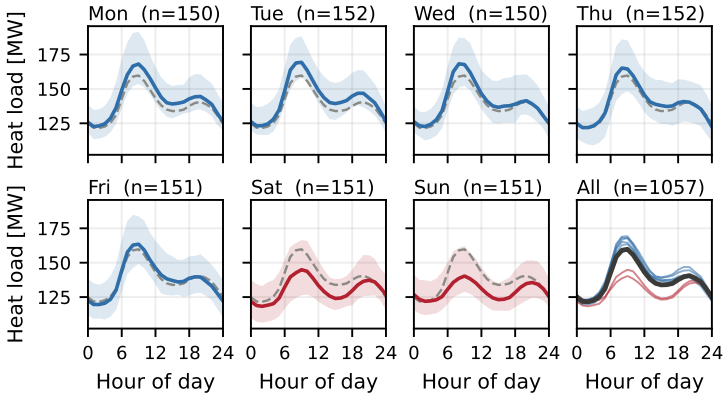
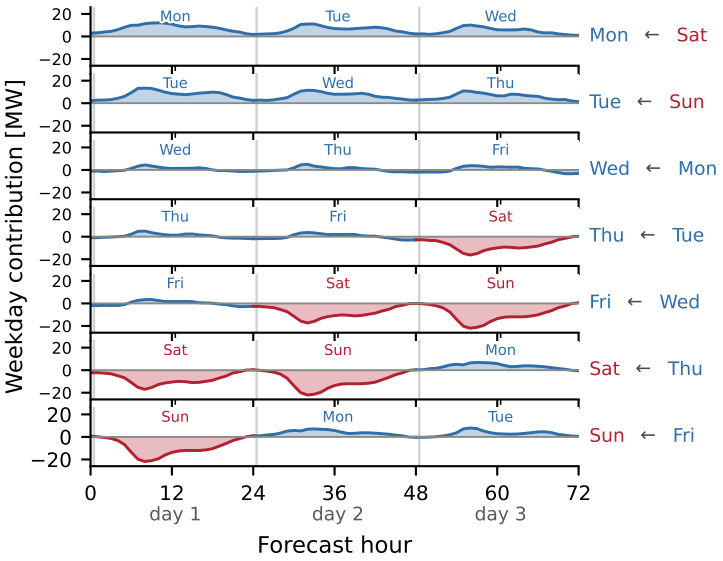


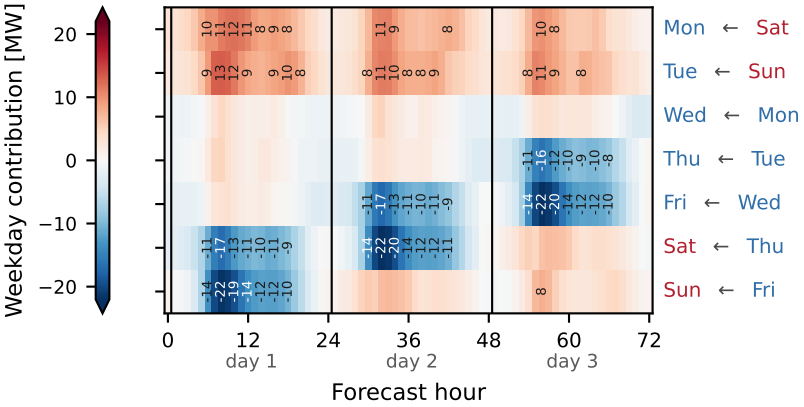
Figure 6: Mean daily load profile per weekday over the six heating seasons (2018/19–2023/24), public holidays excluded; n is the number of days per panel. Color marks the kind of day forecast, working day in blue and weekend in red, and is used the same way in Figure 7. The dashed gray curve, identical in all seven panels, is the mean over all days, so the gap between the curves is the weekday effect. The band spans the minimum and maximum of the per-season means of that weekday for all six seasons: for the working days this year-to-year spread exceeds the distance between the weekdays themselves. The last panel overlays the seven means and the all-day mean.

to 10.69 MW. Unlike the operational reference model, the benchmark models may draw on any hour of the input rather than on forecast hour t alone; this may allow them to capture temporal dependencies across the forecast horizon. The operational reference model is nonetheless not weak: it holds 12.5 MW to 14.9 MW across eight unseen seasons without ever being refitted and clearly beats the naive baselines. For the operator, this means that, on the test season, the forecast error could be reduced by about a quarter without additional data, since all benchmark models use only the inputs already available to the operational reference model. In particular, ElasticNet achieves almost the entire gain while its forecast remains a weighted sum of these inputs; it could therefore be evaluated with an effort comparable to that of the operational reference model and remain open to inspection by the operators who review the forecast.

The amount of complexity required to improve on the operational reference model is considerable. The leading models achieve similar error levels (Section 5). The remaining 2.4 % of the improvement between ElasticNet and



(a) One panel for each forecast day, corresponding to the rows in (b). Vertical lines mark the day boundaries, and each section is labeled with the weekday it covers.



(b) The same contribution shown as a heatmap in (a), with one row per forecast day ← reference-day.

Figure 7: The weekday block of the ElasticNet in MW; blue working day, red weekend. The negative blocks follow Saturday and Sunday wherever they fall in the horizon. Both visualizations show how much is added to the intercept, that is, the mean profile (gray curve in Figure 5b), when the one-hot encoded weekday of the forecast day is set, with all other inputs at their training mean. Zero indicates no contribution from the weekday block relative to the intercept.

TabPFN costs the ability to read the forecast off the model, and the more complex gradient-boosting models and the MLP do not improve on the regularized linear models.

One possible explanation for the strong performance of the regularized linear models is that the input representation already provides the main determinants of the forecast: the reference-day load anchors the load level and profile, the temperature forecast describes the expected non-linear weather conditions, and the weekday block represents recurring calendar effects. Given a weather forecast as input, much of the remaining mapping may be approximately linear over the observed heating seasons. This interpretation remains a hypothesis and would require targeted ablation or nonlinear-dependence analyses for confirmation.

Beyond forecast accuracy, additionally, interpretability is important in the present operational context because forecasts inform dispatch, storage operation, and fuel procurement. ElasticNet supports model inspection through its sparse coefficient matrix, identifiable input blocks, and learned weekday correction. This provides a higher degree of model-level transparency than the nonlinear benchmark models, although it does not by itself guarantee that every individual forecast can be fully explained.

Limitations. Our findings are subject to several limitations. First, this work covers a single network run by a single operator. It therefore varies neither network size nor climate nor operating practice. Whether the results transfer to other district-heating networks also remains untested.

Second, the evaluation relies on a single test season of 164 days. How strongly a single winter can shift the results is illustrated by the operational reference model itself: evaluated across the eight seasons, its error ranges from 12.5 MW to 14.9 MW. Since the best-performing models differ by far less than this, their exact ranking should be interpreted with caution; evaluating all benchmark models on several seasons would make it more robust. In addition, the preconditions of the comparison between the operational reference model and the benchmark models are not identical, because only the latter are fitted on the training seasons. In the operational past, the parameters of the operational reference model were

also re-estimated, which yielded only marginal improvements. This suggests that the remaining gap stems from the structure of the gray-box model itself rather than from its calibration.

Third, the set of benchmark models is limited. We evaluate only two tabular foundation models, each in a single version. As described in Section 4.2, the two tabular foundation models are not fitted in the same way as the other benchmark models. Moreover, time-series foundation models are a natural candidate for this task but are absent from the comparison, because they do not fit our data setup, which holds every benchmark model to the same information base. Including them would require masking the issue day, providing a long historical context, and deciding whether to use measured or forecast temperature, or how to combine them. Further adding covariates for calendar features would be important; covariate-informed models such as Chronos-2 [1] offer a natural way to do so: Kreusel et al. [17] apply them to probabilistic building energy forecasting, and Kreusel et al. [16] evaluate time-series foundation models for cooling demand. Furthermore, the second requirement, explainability, could be addressed: Hertel et al. [8] show that covariate-informed time-series foundation models can provide explainable load forecasts.

Finally, the forecast setting itself is restricted. Because the information base is fixed by design, influences it does not contain, such as holidays, are only captured if they are added explicitly. Furthermore, all forecasts are point forecasts, whereas a probabilistic setting could be more natural for operational planning and may improve decisions. Moreover, it could be interesting to examine how errors in the temperature forecast propagate into errors of the heat load forecast.

8 Conclusion

District-heating forecasts support unit dispatch, storage operation, and fuel procurement, so they must be both accurate and interpretable. At the Stadtwerke Karlsruhe GmbH (Germany), a transparent gray-box model has provided these forecasts since 2017. In this paper, we benchmark the operational reference model against linear models, gradient-boosting models, a neural network, and

two tabular foundation models, all given the same inputs, to determine how much accuracy can be gained and at what cost in interpretability.

On a held-out test season of 164 days, every benchmark model outperforms the operational reference model, and the best, TabPFN, reduces the MAE from 14.46 MW to 10.69 MW. A regularized linear model, ElasticNet, reaches 10.78 MW and thus 97.6 % of this improvement while remaining interpretable: its coefficients reflect the input structure, and it learns a calendar correction.

Future work could pursue several directions. First, it could incorporate time-series foundation models, which would require adapting the data setup. Second, all models could be extended with additional inputs (e.g., holidays, more detailed weather variables) and reformulated in a probabilistic setting, for example via quantile regression or distributional outputs, to support uncertainty-aware decisions. Third, dedicated experiments could quantify how temperature-forecast errors propagate into heat load forecast errors. Finally, evaluating the benchmark models across multiple test seasons and replicating the analysis across networks, operators, and climates would provide stronger evidence regarding the robustness and external validity of the findings.

Acknowledgments

The authors thank the Helmholtz Association for its support under the program “Energy System Design” and its predecessor, the Joint Initiative “Energy System 2050 – A Contribution of the Research Field Energy”. In addition to the Helmholtz support, this work is also part of the research partnership “Forschungspartnerschaft Karlsruhe Energie” (FPS KA Energie), within which the four partners Karlsruhe Institute of Technology (KIT), the Stadtwerke Karlsruhe GmbH, the Stadtwerke Karlsruhe Netzservice GmbH, and the DVGW Research Center at the Engler-Bunte-Institut of KIT intensify and coordinate their collaboration on sustainable energy supply.

Declaration of Generative AI and AI-Assisted Technologies

During the preparation of this work, the authors used Claude Code (Anthropic), running the Claude Opus model, to assist with the development of the analysis and evaluation code and with structuring and language editing of the manuscript. No other generative AI tools were used. All scientific content, including the study design, the experimental protocol, the evaluation and the interpretation of the results, was conceived and verified by the authors. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Abdul Fatir Ansari et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. doi:10.48550/arXiv.2510.15821.

[2] Maciej Bujalski, Paweł Madejski, and Krzysztof Fuzowski. Day-ahead heat load forecasting during the off-season in the district heating system using generalized additive model. *Energy and Buildings*, 278:112630, 2023. doi:10.1016/j.enbuild.2022.112630.

[3] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016. doi:10.1145/2939672.2939785.

[4] Magnus Dahl, Adam Brun, Oliver S. Kirsebom, and Gorm B. Andresen. Improving short-term heat load forecasts with calendar and holiday data. *Energies*, 11(7):1678, 2018. doi:10.3390/en11071678.

[5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[6] Erik Dotzauer. Simple model for prediction of loads in district-heating systems. *Applied Energy*, 73(3-4):277–284, 2002. doi:10.1016/S0306-2619(02)00078-8.

[7] Matthias Finkenrath, Till Faber, Fabian Behrens, and Stefan Leiprecht. Holistic modelling and optimisation of thermal load forecasting, heat generation and plant dispatch for a district heating network. *Energy*, 250:123666, 2022. doi:10.1016/j.energy.2022.123666.

[8] Matthias Hertel, Alexandra Nikoltchovska, Sebastian Pütz, Benjamin Schäfer, Ralf Mikut, and Veit Hagenmeyer. Explainable load forecasting with covariate-informed time series foundation models. In *Proceedings of the 17th ACM International Conference on Future and Sustainable Energy Systems*, E-Energy '26, pages 612–626, New York, NY, USA, June 2026. Association for Computing Machinery. doi:10.1145/3744255.3811724.

[9] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi:10.1080/00401706.1970.10488634.

[10] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi:10.1038/s41586-024-08328-6.

[11] Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. The tabular foundation model TabPFN outperforms specialized time series forecasting models based on simple features. *arXiv preprint arXiv:2501.02945*, 2025. doi:10.48550/arXiv.2501.02945.

[12] Pengmin Hua, Haichao Wang, Zichan Xie, and Risto Lahdelma. District heating load patterns and short-term forecasting for buildings and city level. *Energy*, 289:129866, 2024. doi:10.1016/j.energy.2023.129866.

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 3146–3154, 2017.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6980.

[15] Weihao Kong, Erez Louidor Ilan, Shuxin Nie, Taman Narayan, Rajat Sen, Yichen Zhou, Deqing Fu, Samet Oymak, and Abhimanyu Das. TabFM: A zero-shot foundation model for tabular data. *arXiv preprint arXiv:2609.37959*, 2026. doi:10.48550/arXiv.2609.37959.

[16] Alexander Kreusel, Matthias Hertel, Moritz Noskiewicz, Heiko Maaß, Ralf Mikut, and Veit Hagenmeyer. Evaluating time-series foundation models for cooling demand forecasting with little data. In *Proceedings 35. Workshop Computational Intelligence: Berlin, 20.–21. November 2025*, page 301, 2025.

[17] Alexander Kreusel, Matthias Hertel, Moritz Noskiewicz, Frederik Zahn, Heiko Maaß, Ralf Mikut, and Veit Hagenmeyer. Covariates are the key to accurate probabilistic building energy forecasting with time series foundation models. In *Proceedings of the 2026 ACM Sustainability Week*, ACM Sustainability Week '26, pages 172–182, New York, NY, USA, June 2026. Association for Computing Machinery. doi:10.1145/3765611.3815507.

[18] Teresa Kurek, Artur Bielecki, Konrad Świrski, Konrad Wojdan, Michał Guzek, Jakub Białek, Rafał Brzozowski, and Rafał Serafin. Heat demand forecasting algorithm for a Warsaw district heating network. *Energy*, 217:119347, 2021. doi:10.1016/j.energy.2020.119347.

[19] Stefan Leiprecht, Fabian Behrens, Till Faber, and Matthias Finkenrath. A comprehensive thermal load forecasting analysis based on machine learning algorithms. *Energy Reports*, 7:319–326, 2021. doi:10.1016/j.egyr.2021.08.140.

[20] Henrik Aalborg Nielsen and Henrik Madsen. Predicting the heat consumption in district heating systems using meteorological forecasts. Technical Report IMM-REP-2000-16, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2000.

[21] Henrik Aalborg Nielsen and Henrik Madsen. Modelling the heat consumption in district heating systems using a grey-box approach. *Energy and Buildings*, 38(1):63–71, 2006. doi:10.1016/j.enbuild.2005.05.002.

[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Primož Potočnik, Primož Škerl, and Edvard Govekar. Machine-learning-based multi-step heat demand forecasting in a district heating system. *Energy and Buildings*, 233:110673, 2021. doi:10.1016/j.enbuild.2020.110673.

[24] Nicolas Renkamp. Short-term load forecasting in district heating networks. Master's thesis, Karlsruher Institut für Technologie, 2016.

[25] Nikolaos P. Sakkas and Roger Abang. Thermal load prediction of communal district heating systems by applying data-driven machine learning methods. *Energy Reports*, 8:1883–1895, 2022. doi:10.1016/j.egyr.2021.12.082.

[26] Ben Spoek, Karim K. Ben Hicham, Kai Derzsi, Philipp Althaus, Alexander Mitsos, and Dirk Müller. Systematic evaluation of TabPFN-TS for zero-shot probabilistic heat load forecasting in district heating networks. arXiv:2608.20024, 2026. doi:10.48550/arXiv.2608.20024.

[27] Stadtwerke Karlsruhe GmbH. Umwelterklärung 2026 mit Klimareport und Energiebericht, May 2026.

[28] Stadtwerke Karlsruhe GmbH. Zahlen, Daten und Fakten zur Fernwärme in Karlsruhe, 2026. Kennzahlen Stand 01.04.2026, Energieträgermix 2025.

[29] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005. doi:10.1111/j.1467-9868.2005.00503.x.

The Benchmark Models in Detail

Table 3 specifies every benchmark model with its implementation and preprocessing, and Table 4 the search spaces and selected configurations.

Standardization. Each benchmark model is built as a scikit-learn pipeline consisting of a `StandardScaler` and the estimator [22], the scaler fitted on the training seasons alone and applied unchanged to the validation and test seasons. For the tree ensembles, a monotonic rescaling leaves the splits untouched; it is kept only so that the pipeline is identical across learners. The MLP alone also standardizes the 73 targets and inverts that transformation before any reported error is computed; all errors are reported in MW.

The two tabular foundation models. Both are applied without task-specific training: their weights stay frozen and the 1037 training days are supplied as in-context examples.[2] TabFM's limit on context rows per ensemble member is disabled so that all training days are available; TabPFN has no such limit at this data set size.

[2] We avoid the term *zero-shot* here. The weights are never updated, but the models do see the 1037 labeled training days as context, which makes this in-context learning rather than forecasting without examples.

Table 3: Every benchmark model, together with the operational reference model and the naive baselines: implementation, preprocessing and source. *Feat.* and *Targ.* mark standardization of the 129 inputs and of the 73 targets, fitted on the training seasons only; the operational reference model and the naive baselines are not scaled.

Model	Implementation	Feat.	Targ.	Reference
Naive baselines				
Naive baseline: reference-day	take the reference-day as forecast	n/a	n/a	n/a
Naive baseline: previous week	repeat the same 73 h one week earlier	n/a	n/a	n/a
Operational reference				
Operational reference model	own	n/a	n/a	gray-box archetype
Learned models				
ElasticNet (tuned)	`ElasticNet`	✓	n/a	[22, 29]
Ridge (tuned)	`Ridge`	✓	n/a	[9, 22]
LightGBM (tuned)	`LGBMRegressor` × 73	✓	n/a	[13]
XGBoost (tuned)	`XGBRegressor`, multi-output tree	✓	n/a	[3]
MLP (tuned)	`MLPRegressor`, Adam	✓	✓	[14, 22]
Linear regression (OLS)	`LinearRegression`	✓	n/a	[22]
Tabular foundation models (frozen weights, training days as in-context examples)				
TabPFN (v3)	`tabpfn`, 73 sequential regressions	n/a	n/a	[10]
TabFM (v1.0) Ensemble n=32	`tabfm`, 73 sequential regressions	n/a	n/a	[15]

Table 4: Search spaces and the configuration selected by MAE on the 2024/25 validation season (Val. MAE). For the MLP, the grid search ranks the configurations by MAE on the standardized targets; the Val. MAE is given in MW. The selected models keep their fit on the training seasons and are not refitted on the validation season. Benchmark models not listed here use fixed settings.

Model	Search space (grid)	Selected	Val. MAE↓
ElasticNet (tuned)	`alpha` ∈ {0.001, 0.01, 0.05, 0.1, 0.3, 1.0}; `l1_ratio` ∈ {0.1, 0.3, 0.5, 0.7, 0.9}	`alpha` = 0.1; `l1_ratio` = 0.9	10.85
LightGBM (tuned)	`n_estimators` ∈ {100, 200, 400, 800}; `num_leaves` ∈ {15, 31, 63}; `learning_rate` ∈ {0.03, 0.05, 0.1}	`learning_rate` = 0.03; `n_estimators` = 200; `num_leaves` = 15	10.83
MLP (tuned)	`hidden_layer_sizes` ∈ {(100,), (128, 64), (256, 128), (256, 128, 64), (512, 256, 128)}; `alpha` ∈ {0.0001, 0.001, 0.01}	`alpha` = 0.0001; `hidden_layer_sizes` = (100,)	10.64
Ridge (tuned)	`alpha` ∈ {0.1, 1.0, 3.0, 10.0, 30.0, 100.0, 300.0}	`alpha` = 30.0	10.87
XGBoost (tuned)	`n_estimators` ∈ {100, 200, 400, 800}; `max_depth` ∈ {3, 5, 6}; `learning_rate` ∈ {0.03, 0.05, 0.1, 0.3}	`learning_rate` = 0.05; `max_depth` = 3; `n_estimators` = 800	11.21

Biomedical Self-Supervised Learning using Sequential and Action-Conditioned Prediction: A Proof of Concept in Histopathology

Linus Schilpp[1], Angelo Jovin Yamachui Sitcheu[1], Thomas Lautenschlager[1], Leonard Saur[1], Nils Friederich[1,2*], Ralf Mikut[1*], Stephan Allgeier[1*]

[1]Institute for Automation and Applied Informatics,
Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen

[2]Institute of Biological and Chemical Systems,
Karlsruhe Institute of Technology
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen

* Supervised equally

Abstract

Self-supervised foundation models for biomedical imaging are typically trained with non-sequential objectives that encourage invariance between augmented crops or predict missing content within a limited spatial resolution or range. This paradigm can be limiting for biomedical images, where diagnostically relevant structure often depends on relationships across widely different spatial extents and resolutions. We investigate sequential, action-conditioned prediction as an alternative self-supervised pretraining objective for biomedical imaging. As a proof of concept, we adapt seq-JEPA, originally developed for natural images, to 256×256 px histopathology images by extending its action space with zoom, and introduce seq-JEPA-dense, which predicts the next representation for

every step in the sequence. Under the same per-crop augmentation distribution and with one run per method, seq-JEPA-dense improved tissue classification accuracy in the aggregated linear-probe evaluation from 88.3% to 91.9%, narrowing the gap to SimCLR, which achieved 93.3%. Both sequential variants recover a held-out crop from the preceding crops and the relative action leading to it at roughly twice the chance rate within the same image, indicating that the prediction depends on the action; whether position, zoom, or both drive this remains to be isolated. These results provide proof-of-concept evidence that sequential, action-conditioned self-supervision can be adapted to histopathology by extending its action space with zoom, yielding representations that approach a contrastive baseline, a first step toward its investigation as a self-supervised approach for extended biomedical images such as whole-slide images.

1 Introduction

Self-supervised foundation models have been increasingly developed for biomedical images in recent years, enabling a path toward annotation-free learning from raw images, while generalizing to a variety of downstream tasks with only a small amount of annotated data. Such models can be trained by several paradigms: contrastive methods (e.g., SimCLR [1], with biomedical variants such as MICLe [2]), non-contrastive methods based on self-distillation (e.g., DINO [3], with biomedical variants such as RAD-DINO [4]), reconstruction-based masked image modeling (e.g., MAE [5], with biomedical variants such as CA-MAE [6]), and joint-embedding predictive architectures (JEPAs), which predict target representations in latent space rather than reconstructing pixels (e.g., I-JEPA [7], with biomedical variants such as RadJEPA [8]). Beyond purely image-based signals, self-supervised vision–language models exploit naturally paired image–text data as the supervisory signal (e.g., CLIP [9], with biomedical variants such as BiomedCLIP [10]).

Despite their diversity, image-based objectives share a common recipe. Training rests on single-step objectives that encourage invariance between augmented crops (e.g., SimCLR [1], DINO [3]), or predict missing content within a limited spatial resolution or range (e.g., MAE [5], I-JEPA [7]), in pixel or latent space,

using the remaining visible content. At inference, typically only the pretrained encoder is retained and applied to each input independently, followed by task-specific downstream modules such as classifiers, decoders, or aggregators.

Biomedical images often depict extended specimens whose informative content spans an area or a volume that a single model input typically cannot capture at both wide range and high resolution simultaneously within practical compute and memory budgets [11, 12]: an input at native resolution covers only a small region, while one that covers a larger extent of the specimen does so at reduced resolution [13]. In histopathology, foundation models typically operate at the small-input end of this trade-off, on tiles of 224×224 px that are far smaller than the field of view used in microscopic diagnosis, and have consequently been argued to capture fine-grained texture at the expense of mesoscale tissue organization [14]. We use the term *crop* for a subregion sampled at variable position and size as a model input, whereas a *tile* denotes one of the fixed-size, non-overlapping pieces into which a whole-slide image is partitioned on a regular grid to match the encoder's input size. Single-step objectives can combine inputs from several points along this range–resolution trade-off, e.g., by training on crops of varying sizes [15], but they relate these inputs directly within one step, which limits how far apart in position and scale related inputs can be while remaining similar or directly predictable from one another. Both the training and the inference recipe described above are thus at odds with the structure of the data.

Furthermore, many self-supervised training objectives primarily optimize for invariance across augmented crops (e.g., SimCLR- and DINO-family models [3, 16, 17]) without explicitly modeling the transformations relating them. These relative relations are themselves diagnostically meaningful: whether two abnormal areas are part of one region or of two separate ones cannot be answered from either alone. Such invariance assumes that crops at different positions and scales depict the same object, as made explicit in local–global crop training, an assumption that has been argued to break down for tissue, where a single crop typically contains several heterogeneous structures, and to cause models to encode stain texture rather than biological organization [14].

Modeling long-range dependencies between tiles is then left to downstream aggregation, commonly multiple instance learning (MIL). Attention-based, hierarchical, and graph-based variants do improve task-level performance [14], yet they operate on frozen representations from an encoder that was not optimized to capture these dependencies. Tizhoosh [14] argues that such aggregation can compensate for deficiencies in the upstream representation while masking them. Since any aggregator is bounded by the information present in these representations, we argue that long-range dependencies should instead be learned during initial pretraining.

The value of moving beyond isolated tile representations is also evident from approaches that incorporate spatially structured context during or after pretraining. For example, Quinones Robles et al. [18] add positive pairs formed from spatially adjacent tiles to several invariance-based self-supervised learning (SSL) methods, including Barlow Twins [19], BYOL [20], VICReg [21] and DINOv2 [16], in addition to the usual augmentation-based pairs, improving both tile- and slide-level classification in most settings. Since tiles farther apart are increasingly likely to belong to different tissue classes, the authors restrict contextual pairs to immediate neighbors, and downstream gains are correspondingly highest for encoders pretrained with adjacent tiles and fall off with distance. This confines the spatial range over which such pairs can usefully incorporate context, since they must remain visually and semantically similar to serve as positives. TICON [22] instead keeps the tile encoder frozen and pretrains a transformer to reconstruct masked tile representations from the remaining, unmasked ones inside a larger context of 16×16 tiles, extrapolating to the full slide at inference. This produces a context-enriched representation for every tile and likewise improves tile- and slide-level classification. Together, these approaches expose complementary limitations: the former incorporates spatial context during pretraining, but only through direct pairwise relations between sufficiently similar tiles, whereas the latter introduces broader context only after tile-level encoder pretraining. Neither addresses the more general limitation of single-step objectives, which do not model long-range dependencies through a sequence of intermediate crops during initial pretraining from pixels.

An alternative is to model these dependencies sequentially during pretraining, while explicitly conditioning on the transformations that connect successive

crops. A recent self-supervised method, seq-JEPA [23], processes a sequence of crops, each paired with the action relating it to the next, and aggregates them into a single representation. From this aggregate representation and the upcoming action, it then predicts the representation of the final crop in the sequence. This removes the similarity requirement: two regions need not be similar, only predictable from one another given the action between them. Relative transformations between crops are also better preserved than by invariance-trained encoders [23]. Because the context consists of a chain of intermediate crops and actions rather than a single pair, predictions can depend on information accumulated across the sequence rather than only on a direct relation between two crops. At the same time, the transformations relating successive crops can compose across the sequence. This is reflected in path integration, where the cumulative transformation from the first to the last crop can be decoded from the aggregate representation together with the final action embedding [23]. Beyond composition, a sequence can accumulate context from a limited observation budget rather than from all observations at once, a property relevant to whole-slide images, whose exhaustive encoding at high resolution can become computationally expensive. This approach is more closely aligned with biological vision, which acquires context through a sequence of saccades, each accompanied by an internal copy of the motor command leading to the next observation, rather than from a single observation [24].

We therefore argue that sequential, action-conditioned prediction points toward an alternative direction for biomedical self-supervised models. As a proof of concept, we adapt seq-JEPA [23] to 256×256 px histopathology images and evaluate both the original formulation and our per-timestep variant against a SimCLR baseline on tissue classification and held-out-crop retrieval.

Our contributions are as follows.

- **We introduce a variable-scale extension of seq-JEPA's original saccade setting**, adding a zoom dimension to its 2D shift actions so that successive crops can differ in scale as well as position. While seq-JEPA's separate augmentation setting already encodes crop position, height, and width independently as actions, our formulation retains the saccade setting's 2D shift structure while leaving photometric augmentations

outside the action space. This yields a hybrid setting in which seq-JEPA and SimCLR share the same per-crop augmentation pipeline.

- **We propose seq-JEPA-dense**, extending seq-JEPA's action-conditioned prediction objective from a single prediction of the final representation to next-representation prediction at every timestep under causal masking. This yields dense next-representation supervision, as also explored in recent sequential image SSL methods [25–27], while retaining seq-JEPA's explicit relative-action conditioning: each token carries its incoming action, and the outgoing action is supplied to the predictor at every step.

2 Related Work

Sequential prediction has a long history as a self-supervised learning objective for static images. Contrastive Predictive Coding [28] recasts it as a general-purpose pretraining objective, summarizing the patches seen so far with an autoregressive model and predicting the representations of future patches contrastively. Within this line, AIM [29] and NEPA [25] retain an autoregressive structure but replace the contrastive objective with direct regression, predicting future patches in pixel and latent space respectively. NEPA predicts the next patch representation at each sequence position under causal masking, using a cosine-similarity loss averaged across positions. These approaches, however, traverse a fixed grid of patches in the same order for every image, typically row by row from top to bottom, so that the next target is given by the order rather than specified by a transformation.

In contrast, I-JEPA [7] predicts the representations of held-out target regions directly from the visible image context, with each target predicted independently rather than as part of a sequence. DSeq-JEPA [26] introduces sequential prediction into this framework, using a saliency map derived from the target encoder's attention to both select the target regions and define a sequential prediction order. The target-region representations are then predicted sequentially along this content-dependent order. While this yields a content-dependent rather than fixed prediction order, each target is still specified by an absolute positional

token rather than by a relative transformation from the preceding region. DSeq-JEPA outperforms an I-JEPA baseline across several vision benchmarks, with ablations indicating that the improvement depends on combining sequential prediction with discriminative region selection [26].

Image World Models (IWM) [30] are a complementary extension of I-JEPA. IWM retain I-JEPA-style mask tokens, whose positional embeddings specify where target representations are to be predicted, while conditioning the predictor on known photometric and corruption transformations between source and target views. Depending on the conditioning and predictor capacity, the learned representation can then retain transformation-dependent information rather than becoming invariant to those transformations. The spatial component of the prediction task, however, is still specified by absolute target positions rather than by relative transformations between views, and prediction remains pairwise rather than sequential.

seq-JEPA [23], which forms the basis of our work, brings together transformation conditioning and sequential prediction. In its saccade setting, the observation sequence is derived from DeepGaze IIE [31], a neural model for predicting human visual saliency and fixation distributions, rather than from a fixed raster traversal or the target encoder's attention. Fixations are drawn from this saliency map with inhibition of return, which suppresses already visited regions so that successive crops overlap less and the sequence is less redundant. Each observation is paired with the relative transformation, such as a change in position, that connects it to the succeeding one. A transformer aggregator summarizes this history of observation–action pairs, and a predictor then predicts the representation of the next observation from the aggregate conditioned on the upcoming action. This formulation allows the encoder to retain observation–action-equivariant information while the aggregated representation remains suitable for invariant downstream tasks. Following [23], equivariance means that for each action a relating two views x_1 and x_2 of an image, and encoder f, there exists a transformation u_a in latent space such that $f(x_2) = u_a(f(x_1))$. When u_a is the identity, this reduces to invariance, $f(x_2) = f(x_1)$.

Related self-supervised methods in biomedical imaging have explored the same ingredients in different combinations. CheXWorld [32] predicts target-region

representations from their relative displacement in chest X-ray images among other self-supervised objectives, while X-WIN [33] predicts representations of novel X-ray projections conditioned on changes in viewpoint around a corresponding computed tomography (CT) volume. Both explicitly model transformations between observations, but formulate prediction as a single-step relation rather than over an extended sequence.

GazeWorld [27] instead introduces sequential prediction in a biomedical setting. It refines a pretrained encoder by autoregressively predicting representations along expert gaze trajectories. In its next-fixation branch, however, the location of the next fixation is not supplied to the predictor and must be inferred from the fixation history, while an equally weighted spatial-completion branch specifies unvisited target patches by their absolute positions, as in I-JEPA. It therefore models an extended observation sequence while relying on pretrained encoder representations rather than learning them from scratch.

Wang and Chen [34] combine sequential and action-conditioned prediction for volumetric magnetic resonance imaging (MRI) data. They render a fixed 3D volume as a sequence of 2D slices accompanied by control vectors describing parameters such as position, orientation, and scale, and train a causal dynamics model to predict representations along the resulting trajectory. To our knowledge, this is the closest prior formulation in a biomedical context to our setting. It nevertheless differs in two important respects. First, the visual representations are learned in a separate masked-reconstruction stage and frozen before dynamics training, so the sequential objective shapes only the dynamics model and not the encoder. Second, its actions encode absolute per-frame rendering states rather than relative transformations between successive observations. Relative transformations, in contrast, do not require a global reference frame, as each action is defined only with respect to the preceding observation.

As discussed above, prior work has explored both sequential prediction in latent space and explicit transformation conditioning as self-supervised objectives for static images, separately and in combination. On natural images, IWM condition prediction on the transformation between pairs of views, while seq-JEPA combines explicit inter-view actions with sequential prediction. In biomedical

imaging, transformation-conditioned methods remain predominantly single-step, while sequential approaches either leave the upcoming target implicit or specify it by absolute positions or acquisition states, or build on pretrained or separately learned representations. Our work studies sequential self-supervised representation learning in histopathology conditioned on the relative changes in position and scale between successive crops.

3 Methods

3.1 Dataset

We use the PathoROB Tolkach ESCA dataset [35, 36], comprising 16,300 histopathology images of size 256×256 px across four cohorts and six tissue classes (Figure 1), at an approximate physical resolution of 0.75 µm/px [35]. We pool all four cohorts and construct a train/test split for the present study, which differs from the original PathoROB evaluation protocol. We use a five-fold stratified group split, stratifying by tissue class and grouping by `slide_id`, with shuffling and random seed 42; the first fold is used as the test set. This yields 13,400 training images (82.2%) and 2,900 test images (17.8%). Of the 77 slide/case identifiers, 62 are assigned to training and 15 to testing, with no `slide_id` shared between the two sets.

3.2 Augmentation Strategy

In seq-JEPA's original saccade setting, actions encode only the relative displacement $\delta_i = (\Delta u_i, \Delta v_i) \in \mathbb{R}^2$ between fixed-size crops, where $(u_i, v_i) \in [0,1]^2$ denotes the center of crop i (row, column) normalized by the source-image size, whereas its augmentation setting additionally represents crop height and width separately in the action space. We extend the saccade setting to variable-scale crops by adding the change in zoom as an additional action dimension, where the zoom of crop i with side length l_i in source-image pixels is $\zeta_i = \log(\text{input\_size}/l_i)$, with $\text{input\_size} = 128$ px the fixed encoder input size to which every sampled crop is resized. This results in $\delta_i = (\Delta u_i, \Delta v_i, \Delta\zeta_i) =$

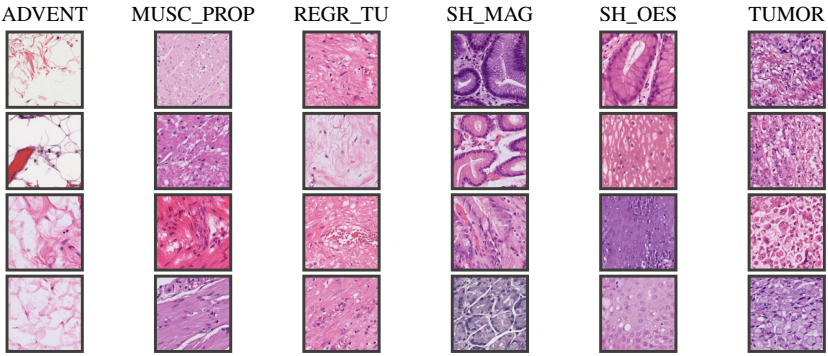


Figure 1: Example images of the six tissue classes in the PathoROB Tolkach ESCA dataset [35, 36]. Each column corresponds to one tissue class and contains four randomly sampled images. Class abbreviations follow [35]: `ADVENT` = adventitial tissue; `MUSC_PROP` = muscularis propria; `REGR_TU` = regression areas; `SH_MAG` = gastric mucosa; `SH_OES` = oesophageal mucosa; `TUMOR` = vital tumor tissue.

$(u_{i+1}-u_i,\ v_{i+1}-v_i,\ \zeta_{i+1}-\zeta_i) \in \mathbb{R}^3$, where $\Delta\zeta_i = \log(l_i/l_{i+1})$ is independent of input_size and positive when zooming in. The logarithmic parameterization makes multiplicative changes in zoom compose additively across successive crops. Thus, the actions for seq-JEPA encode changes in crop location and scale, but not the photometric augmentations applied to the crop. This design allows for SimCLR and seq-JEPA to share an identical per-crop augmentation pipeline. From each 256×256 px image, square crops covering 8%–100% of the source-image area are sampled, resized by bicubic interpolation to 128×128 px (input_size), and individually augmented with color jitter ($p = 0.8$), random grayscale ($p = 0.2$), and Gaussian blur ($p = 0.5$; 25×25 kernel, $\sigma \in [0.1, 2.0]$). SimCLR samples $S = 2$ such crops and uses them as a positive contrastive pair. seq-JEPA samples $S = 5$ crops, of which the first $N = 4$ are observed together with the action embeddings of their $(\Delta u_i, \Delta v_i, \Delta\zeta_i)$ transformations, while the last is held out as the prediction target. Crops are sampled at random and independently of the model's state. Unlike the original saccade setting [23], crop centers are not guided by a saliency map or inhibition of return, crops always lie fully inside the image, and each crop additionally receives photometric augmentation. For the test split and for visualization, photometric augmentation is disabled and crop sampling is seeded per image, so that all models are

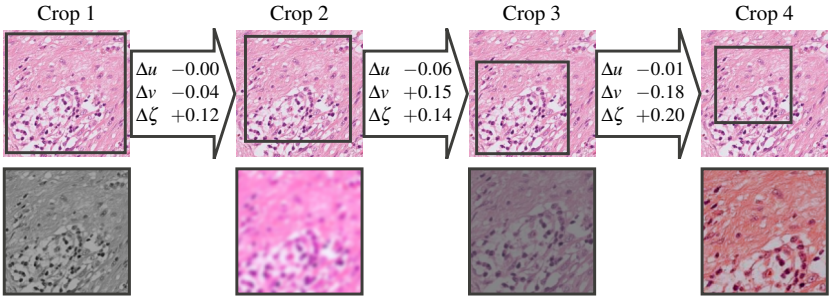


Figure 2: Example sequence of four crops sampled from a single pathology image. **Top:** the source image with the bounding box for each crop outlined. **Bottom:** the corresponding extracted and augmented crop. Arrows show the transform between consecutive crops, given as the change in position and log-zoom, $(\Delta u, \Delta v, \Delta\zeta)$.

evaluated on identical crops. An example sequence of augmented crops is shown in Figure 2.

3.3 Models

We compare three models: A SimCLR [1] baseline, seq-JEPA [23], and seq-JEPA-dense, which replaces the single loss calculation at the end of the sequence with supervision for every consecutive step. All three methods use a ResNet-18 [37] backbone with random initialization, yielding crop representations $z_i \in \mathbb{R}^{512}$.

SimCLR. The encoder is followed by a 3-layer multilayer perceptron (MLP) projection head (2048-2048-2048 hidden units, batch-normalized), trained with the NT-Xent contrastive loss at temperature $\tau = 0.5$ on the two augmented crops described in Section 3.2. This follows the SimCLR baseline from the original seq-JEPA implementation rather than SimCLR's ImageNet defaults ($\tau = 0.1$, two layers, 128-d).

seq-JEPA. Following the original setup, a sequence of $N = S - 1$ encoded crops, each concatenated with an embedding $a_i \in \mathbb{R}^{128}$ of its *outgoing* action, with a zero vector in place of the action for the last crop, is aggregated by a lightweight transformer encoder (4 heads, 3 layers) via a learnable `[AGG]` token $\in \mathbb{R}^{640}$.

The aggregator uses no positional encoding. The aggregate representation at the [AGG] token, $z_{\text{AGG}} \in \mathbb{R}^{640}$, is concatenated with the final action embedding and passed through a two-layer MLP predictor (hidden dimension 1024, output dimension 512) to predict the encoder's representation of the held-out final crop, with a stop-gradient applied to the target and trained with a negative cosine similarity loss.

seq-JEPA-dense. This variant shares the same backbone, aggregator, and predictor architecture as seq-JEPA but predicts at every position in the sequence instead of only the last, using a causal attention mask so that each token only attends to itself and preceding tokens. The prepended [AGG] token is retained from seq-JEPA but is no longer used as an aggregate readout. Instead, it serves as a learned beginning-of-sequence token [BOS] that provides a common initial context for the causally masked sequence. Each token now carries its *incoming* relative action, with a zero vector for the first crop, which has no incoming action, so that the action pointing at a given position's target is withheld from the aggregator and injected only at the predictor. The predictor is applied at each position, conditioning that position's causally-attended transformer output $h_i \in \mathbb{R}^{640}$ on its own outgoing action to forecast the next crop's representation, using teacher forcing, i.e., conditioning on the encoded observed crops rather than on earlier predictions. Targets are the encoder's representations at every step with a stop-gradient applied, and the loss averages the negative cosine similarity over all N predictions, yielding denser supervision per training step.

The prediction objectives of seq-JEPA and seq-JEPA-dense are illustrated in Figure 3. The official seq-JEPA code [23] supports both exponential moving average (EMA) and online targets. Unlike the original paper, which uses an EMA target encoder (decay 0.996), we use the online-target option, computing targets with the online encoder under stop-gradient. SimSiam [38] showed that stop-gradient with a predictor can prevent collapse without a momentum encoder. This choice also keeps all compared methods free of a momentum encoder and, for seq-JEPA-dense, lets targets reuse the observed crop encodings, saving $N - 1$ encoder passes per sequence.

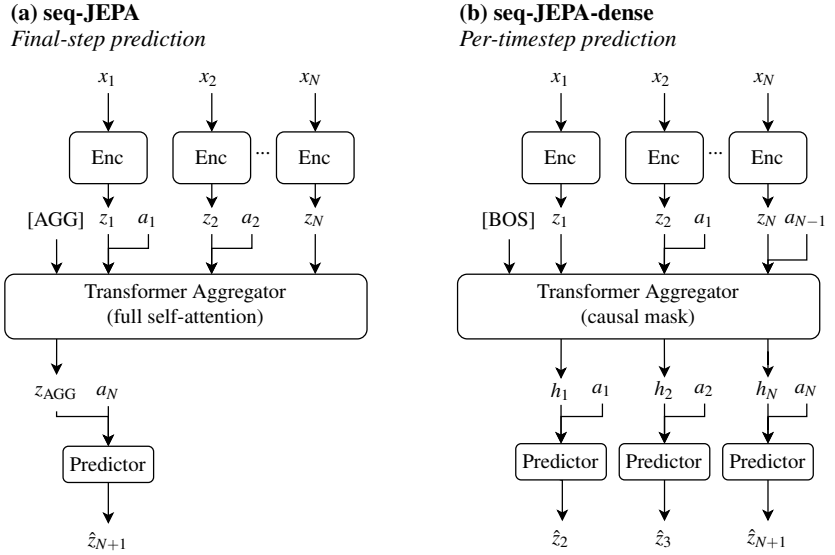


Figure 3: Prediction scheme of (a) seq-JEPA [23] and (b) seq-JEPA-dense. a_i denotes the relative action embedding from crop x_i to x_{i+1}. In (a), the aggregator attends over (z_i, a_i) for $i = 1, \ldots, N-1$ and $(z_N, \mathbf{0})$, and the predictor uses the `[AGG]` output z_{AGG}, conditioned on a_N, to predict z_{N+1}. In (b), a causal mask is applied, and each token instead carries its incoming action (z_i, a_{i-1}), with $a_0 = \mathbf{0}$; the predictor uses the output h_i, computed from `[BOS]` and the tokens $(z_1, \mathbf{0}), (z_2, a_1), \ldots, (z_i, a_{i-1})$, conditioned on a_i, to predict the encoder representation of the next crop z_{i+1}. The aggregate token `[AGG]` is repurposed as a beginning-of-sequence token `[BOS]`, and the representation used for the classification task is read from h_N instead of z_{AGG}. All Enc blocks denote the same encoder, and all Predictor blocks the same predictor.

3.4 Experimental Setup

3.4.1 Training

SimCLR, seq-JEPA, and seq-JEPA-dense are pretrained from scratch on the training split with the same augmentation pipeline and epoch budget. We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate of 4×10^{-4}, weight decay of 10^{-3}, and batch size of 512. The learning rate is linearly warmed up from 10^{-5} over 20 epochs and kept constant thereafter. All models are trained for 1,000 epochs in bf16 mixed precision with a fixed seed on a single NVIDIA RTX 3090 GPU, and the final checkpoint is evaluated.

3.4.2 Classification Evaluation

After pretraining, we freeze each model and train linear classification probes for 100 epochs using Adam with a learning rate of 10^{-3}. The probes are trained on the training split and evaluated by top-1 accuracy (percentage of test samples correctly classified) on the 6-class classification task corresponding to the six tissue classes.

For each method, we train two linear classification probes. The *res* probe operates on the single-crop ResNet representation, while the *agg* probe operates on a multi-crop representation aggregated over $N = 4$ crops. For SimCLR, which has no learned aggregator, we mean-pool the encoder representations across the four crops as a parameter-free aggregation. For seq-JEPA, we use the z_{AGG} token; for seq-JEPA-dense, we use the transformer output at the last crop token, h_N.

3.4.3 Retrieval Evaluation

We evaluate retrieval with Hit@1, the fraction of queries for which the correct target is the top-ranked candidate. This requires recovery of the specific held-out crop rather than any crop with the same tissue label. For each test image, we sample $S = 5$ crops as during training: $N = 4$ context crops and one held-out target. Candidates are ranked by the cosine similarity between the query and their encoder representations. For both seq-JEPA variants, the query is the predictor output conditioned on the action leading to the target, taking the final-step prediction in the dense case. For SimCLR, the query is the mean-pooled encoder representations of the four context crops, matching the classification evaluation *agg* probe.

Our primary setting, within-image retrieval, adapts the same-instance retrieval protocol of Garrido et al. [39], also used by Ghaemi et al. [23] on the 3DIEBench dataset [39]. We retain the same-instance restriction, but tailor the candidate set to our sequential setting, using the four observed context crops as distractors alongside the held-out target ($N + 1 = 5$; chance 20%). Because all candidates

come from the same image, slide-, stain-, and center-specific cues are held constant, so success requires identifying the requested crop. SimCLR is excluded because its query averages the context crops and is not action-conditioned.

The secondary setting enlarges the candidate set: we keep one held-out target crop per test image, resulting in $M = 2{,}900$ crops, and rank each query against all of them (chance $\approx 0.034\%$). A hit thus requires identifying the correct *image*, not the correct crop within a known image. Since the distractors now come from other images, they can differ from the target in stain and slide-level appearance, not only in morphology. Hit@1 therefore cannot separate morphological specificity from those cues in this setting.

3.4.4 Representation Space Analysis

To qualitatively examine the learned representation spaces, we visualize the *res* and *agg* representations of each model using Uniform Manifold Approximation and Projection (UMAP) [40] over all 16,300 images in the dataset. The *res* representation corresponds to the encoder output of a single crop ($N = 1$), while *agg* corresponds to the aggregated representation over $N = 4$ crops, as used for the respective classification probes. We fit UMAP separately to each model and representation using Euclidean distance on the unnormalized feature vectors, with 30 nearest neighbors, a minimum distance of 0.1, and a fixed random seed of 42. Points in the resulting two-dimensional projections are colored by tissue class.

4 Results

4.1 Classification Results

Table 1 shows that SimCLR achieves the highest top-1 accuracy in both settings, followed by seq-JEPA-dense and seq-JEPA. Relative to seq-JEPA, seq-JEPA-dense improves accuracy by 3.3 and 3.6 percentage points in the single-crop and aggregated settings, narrowing the gap to SimCLR from 5.1 to 1.8 and from 5.0 to 1.4 percentage points.

Table 1: Linear-probe accuracy for tissue classification across six classes on the test split used in this study for PathoROB Tolkach ESCA (2,900 images). Results are shown for SimCLR, seq-JEPA, and seq-JEPA-dense, comparing single-crop and aggregated evaluation across $N = 4$ crops, reported as top-1 accuracy.

Method	Crops (N)	Top-1 Acc. (%)
SimCLR (*res*)	1	**91.7**
seq-JEPA (*res*)	1	86.6
seq-JEPA-dense (*res*)	1	89.9
SimCLR (*agg*)	4	**93.3**
seq-JEPA (*agg*)	4	88.3
seq-JEPA-dense (*agg*)	4	91.9

4.2 Retrieval Results

In the within-image retrieval setting (Table 2), both seq-JEPA variants exceed chance (42.1% and 41.8% against 20%), with comparable Hit@1. In the cross-image retrieval setting, seq-JEPA-dense identifies the held-out crop most often (95.3%), ahead of SimCLR (87.0%) and seq-JEPA (80.6%).

4.3 Representation Space Analysis

Figure 4 shows UMAP projections of the *res* ($N = 1$) and *agg* ($N = 4$) representations for SimCLR, seq-JEPA, seq-JEPA-dense, colored by tissue class. SimCLR's clusters appear most visually separated of the three methods, forming distinct, well-isolated groups in both the *res* and *agg* settings. Among seq-JEPA and seq-JEPA-dense, the relative separation depends on the setting: at $N = 1$, seq-JEPA-dense's clusters appear more visually separated than seq-JEPA's, while at $N = 4$ this pattern reverses, with seq-JEPA's clusters appearing more separated.

Table 2: Held-out-crop retrieval on the test split used in this study for PathoROB Tolkach ESCA (2,900 images), reported as Hit@1. Within-image: the target is ranked against the four context crops from the same image (5 candidates). Cross-image: the target is ranked against the held-out targets of all $M = 2{,}900$ test images. SimCLR is not action-conditioned and is therefore evaluated only in the cross-image setting.

Method	Within-image (%)	Cross-image (%)
Chance	*20.0*	*0.034*
SimCLR	–	87.0
seq-JEPA	**42.1**	80.6
seq-JEPA-dense	**41.8**	**95.3**

5 Discussion

In our experiments, both seq-JEPA variants support action-conditioned retrieval within an image, and seq-JEPA-dense produces the most image-specific retrieval queries among the compared methods while approaching the contrastive baseline in tissue classification.

seq-JEPA-dense narrows the classification gap to SimCLR relative to seq-JEPA (Table 1). As it also differs from seq-JEPA in its attention mask, the assignment of actions to tokens, and its readout, this gain cannot be attributed to denser supervision alone. The UMAP projections, however, show less separated aggregate clusters for seq-JEPA-dense than for seq-JEPA, the reverse of their classification ordering; apparent separation in these 2D projections is thus not a reliable proxy for linear separability in the feature space. SimCLR nevertheless remains strongest for tissue classification. A plausible explanation is our single-image setting: with all crops drawn from one 256×256 px image and an input resolution of 128×128 px, a single crop can already capture most of the image's information, so the sequence adds little class-relevant information beyond its largest crop. An advantage of sequential, action-conditioned prediction over single-step objectives such as SimCLR would therefore be expected mainly when the relevant content cannot be captured by any single crop, as in large whole-slide images.

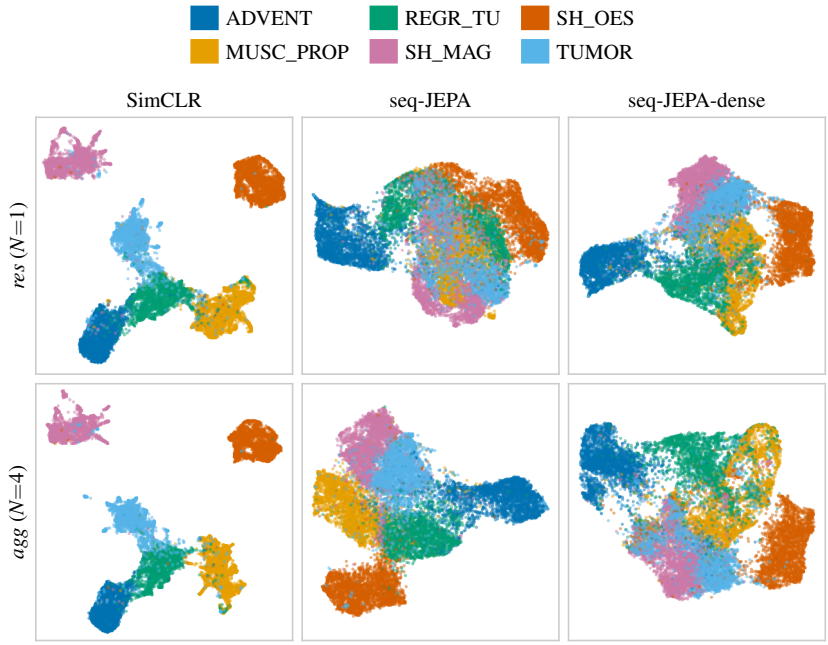


Figure 4: UMAP projections of the learned representations of SimCLR, seq-JEPA, and seq-JEPA-dense for all 16,300 images in the PathoROB Tolkach ESCA dataset [35, 36], colored by tissue class. Top row: *res* representations (encoder output for a single crop, $N = 1$). Bottom row: *agg* representations (aggregated over $N = 4$ crops).

Recovering a crop specified by an action requires a query that encodes the requested relative action. Both seq-JEPA variants exceed chance substantially within an image (42.1% and 41.8% against 20% chance rate), indicating that the predictor uses the queried action rather than returning an image-level average, although this evaluation does not show whether it relies on position, zoom, or both. seq-JEPA-dense provides no advantage in this setting over seq-JEPA despite its classification gain. We hypothesize that this results from the redundancy of the sampled sequence: crops are drawn independently and may cover up to 100% of the image, whereas seq-JEPA's saccade setting uses inhibition of return to avoid overlapping, redundant sequences. Consequently, a representation similar to that of the held-out crop may be constructed largely from the preceding crop representations, reducing pressure on the predictor

to rely on the relative actions. An inhibition-of-return-like constraint on crop overlap is difficult to impose here, as on 256×256 px images it would require restricting crops to small areas and thus depart from the crop distribution shared with SimCLR.

In the cross-image retrieval task, seq-JEPA-dense identifies the held-out crop more often than SimCLR, the reverse of the classification ordering. Because candidate images span medical centers, success requires only that the query is specific to the source image, and slide- or center-specific batch effects may contribute. seq-JEPA-dense matches seq-JEPA within an image but reaches a considerably higher cross-image Hit@1 (95.3% vs. 80.6%), indicating that it retains more image-specific information. This may stem from the denser training signal, but the accompanying architectural changes were not isolated. The matching within-image results do not show that this additional information is non-positional: because redundant crop sequences could allow the target to be approximated largely without the action, within-image retrieval may be only weakly sensitive to positional information. Both SimCLR and the seq-JEPA variants encourage image-specific representations: SimCLR explicitly discriminates between crops of different images, whereas the seq-JEPA variants predict a crop from other crops of the same image, which requires image-specific information. One interpretation of seq-JEPA-dense ranking above SimCLR is that its query matches the target crop more specifically: the SimCLR query averages the context crops and thus represents the image as a whole, whereas seq-JEPA-dense predicts the particular target crop from the requested action, which may help to distinguish it from similar images of the same slide. How much of the preserved signal is morphological, and therefore potentially useful for tasks that depend on local structure, or technical, cannot be determined from this evaluation alone.

6 Limitations

Our evaluation is limited to a single dataset comprising 13,400 training images. Although the dataset splits and training and evaluation runs are seeded for

reproducibility, we train only a single run per method and do not perform cross-validation. Consequently, we cannot report variability across random seeds for either classification or retrieval performance.

Moreover, context and action do not fully determine the content of non-overlapping crops, yet the deterministic predictor outputs a single point estimate, likely approximating an average over plausible targets rather than any specific one.

While we aim to match the training settings of SimCLR and the seq-JEPA variants as closely as possible, the methods differ in the amount and type of self-supervision received per loss update. SimCLR operates on two augmented crops per update, whereas the seq-JEPA variants use five. Moreover, seq-JEPA-dense receives a denser training signal by predicting each consecutive step. Conversely, SimCLR benefits from explicit in-batch negatives as part of its contrastive objective, while the seq-JEPA variants carry a larger parameter count from the transformer aggregator their objective requires. These differences make a strictly controlled comparison difficult and should be considered when interpreting relative performance across the methods. The comparison is also limited to a single baseline family (contrastive, SimCLR); non-contrastive self-distillation (e.g., BYOL [20], DINO [3]) and reconstruction-based objectives (e.g., MAE [5]) are outside the scope of this proof-of-concept study.

7 Conclusion and Future Work

We adapt seq-JEPA to computational pathology by extending its action space with a zoom dimension, and introduce seq-JEPA-dense, a variant that predicts at every position in the crop sequence under a causal mask. In a single pretraining run per method, seq-JEPA-dense improves linear-probe accuracy over seq-JEPA by 3.3 and 3.6 percentage points in the single-crop and aggregated settings, narrowing the gap to a SimCLR baseline to 1.8 and 1.4 points. Both variants recover a held-out crop within its own image at roughly twice the chance rate, in a setting where slide-, stain-, and center-specific cues are constant by construction. In the cross-image retrieval setting, seq-JEPA-dense additionally retrieves the held-out crop more accurately than both seq-JEPA and SimCLR,

suggesting that seq-JEPA-dense preserves image-specific information not fully reflected by linear-probe classification.

Beyond the single-image setting evaluated here as a proof of concept, sequential, action-conditioned prediction could extend to biomedical images whose informative content spans large areas or volumes, such as whole-slide images, microscopy z-stacks, or volumetric scans. For whole-slide images, a trajectory could move across both position and magnification through the same shift and zoom actions used here, allowing the model to relate fine-grained local morphology to tissue organization over much larger spatial scales. In this way, the same predictive objective could capture long-range dependencies and local details jointly, rather than treating tile-level representation learning and slide-level aggregation as separate stages. Scaling the approach to gigapixel slides would, however, require more efficient handling of long crop sequences and more efficient crop encoders. Furthermore, the crop trajectory would need to be sampled such that subsequent crops are sufficiently predictable for the model to learn meaningful representations. A simple sampling policy could be to sample crops that are close in position and zoom level, i.e., with small relative action magnitude. Alternatively, the sampling policy could itself be learned without supervision, e.g., by selecting actions that minimize the uncertainty of the aggregated representation. On small-scale active vision tasks, Sharafeldin et al. [41] show that such uncertainty-driven sampling learns to focus on informative image regions and improves downstream classification accuracy compared to random sampling. Adapting such an approach to seq-JEPA and to whole-slide images is left for future work.

A second direction is task-specific adaptation of the sampling policy after initial pretraining. On whole-slide images, random sampling is unlikely to provide an efficient exploration strategy for downstream tasks, making it important to learn which regions and magnifications should be queried for a given task. One possibility is supervised adaptation, for example by freezing the pretrained encoder and training an action-selection policy from downstream labels or region-level supervision so that the resulting crop sequence becomes task-directed. The causal, action-conditioned formulation also lends itself naturally to reinforcement learning, where the policy could instead be optimized directly for downstream performance.

Segmentation could be an appropriate next downstream evaluation task. The retrieval results suggest that seq-JEPA-dense retains information that classification probes do not reward, and segmentation requires spatially resolved predictions, whereas the tissue-classification probe produces one label per image. Action-conditioned world models have been applied to biomedical image segmentation before [32, 34] (Section 2), though not with relative actions over an observation sequence.

8 Code Availability

The code for all experiments is available at
`https://github.com/lischilpp/seq-jepa-dense`.

9 Acknowledgments

The present contribution is supported by the Helmholtz Association under the program “Natural, Artificial and Cognitive Information Processing (NACIP)” and the joint research school “Helmholtz Information & Data Science School for Health (HIDSS4Health)”.

During the preparation of this work, the authors used the AI language models Claude Sonnet 5, Claude Opus 5 and 5.5 (Anthropic) and GPT-5.6 Sol (OpenAI) to assist with literature search, software development, and drafting and revising the manuscript text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of this manuscript.

References

[1] T. Chen *et al.*, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research,

H. Daumé III and A. Singh, Eds., vol. 119. PMLR, Jul. 2020, pp. 1597–1607.

[2] S. Azizi *et al.*, "Big self-supervised models advance medical image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 3478–3488.

[3] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9650–9660.

[4] F. Pérez-García *et al.*, "Exploring scalable medical image encoders beyond text supervision," *Nature Machine Intelligence*, vol. 7, no. 1, pp. 119–130, Jan. 2025.

[5] K. He *et al.*, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 16 000–16 009.

[6] O. Kraus *et al.*, "Masked autoencoders for microscopy are scalable learners of cellular biology," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024, pp. 11 757–11 768.

[7] M. Assran *et al.*, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 15 619–15 629.

[8] A. A. H. Khan *et al.*, "RadJEPA: Radiology encoder for chest X-rays via joint embedding predictive architecture," arXiv:2601.15891, 2026.

[9] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, Jul. 2021, pp. 8748–8763.

[10] S. Zhang *et al.*, “A multimodal biomedical foundation model trained from fifteen million image–text pairs,” *NEJM AI*, vol. 2, no. 1, p. AIoa2400640, 2025.

[11] M. Gadermayr and M. Tschuchnig, “Multiple instance learning for digital pathology: A review of the state-of-the-art, limitations and future potential,” *Computerized Medical Imaging and Graphics*, vol. 112, p. 102337, 2024.

[12] V. van Veldhuizen *et al.*, “Foundation models in medical imaging: A review and outlook,” arXiv:2506.09095, 2025.

[13] A. H. Song *et al.*, “Artificial intelligence for digital and computational pathology,” *Nature Reviews Bioengineering*, vol. 1, no. 12, pp. 930–949, Oct. 2023.

[14] H. R. Tizhoosh, “Rethinking foundation models in pathology,” *Nature Biomedical Engineering*, vol. 10, no. 8, pp. 1513–1518, Jun. 2026.

[15] M. Caron *et al.*, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Advances in Neural Information Processing Systems*, H. Larochelle *et al.*, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924.

[16] M. Oquab *et al.*, “DINOv2: Learning robust visual features without supervision,” *Transactions on Machine Learning Research*, 2024.

[17] O. Siméoni *et al.*, “DINOv3,” arXiv:2508.10104, 2025.

[18] W. R. Quinones Robles *et al.*, “Leveraging spatial context for positive pair sampling in histopathology image representation learning,” arXiv:2503.05170, 2026.

[19] J. Zbontar *et al.*, “Barlow Twins: Self-supervised learning via redundancy reduction,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, Jul. 2021, pp. 12 310–12 320.

[20] J.-B. Grill *et al.*, "Bootstrap Your Own Latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, H. Larochelle *et al.*, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 271–21 284.

[21] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *International Conference on Learning Representations*, 2022.

[22] V. Belagali *et al.*, "TICON: A slide-level tile contextualizer for histopathology representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, Jun. 2026, pp. 5440–5451.

[23] H. Ghaemi, E. B. Muller, and S. Bakhtiari, "seq-JEPA: Autoregressive predictive learning of invariant-equivariant world models," in *Advances in Neural Information Processing Systems*, D. Belgrave *et al.*, Eds., vol. 38. Curran Associates, Inc., 2025, pp. 32 943–32 973.

[24] R. H. Wurtz, "Corollary discharge contributions to perceptual continuity across saccades," *Annual Review of Vision Science*, vol. 4, pp. 215–237, 2018.

[25] S. Xu *et al.*, "Next-embedding prediction makes strong vision learners," arXiv:2512.16922, 2025.

[26] X. He *et al.*, "DSeq-JEPA: Discriminative sequential joint-embedding predictive architecture," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

[27] Y. Li *et al.*, "A world model of radiologist reading for medical image representation learning," arXiv:2605.23992, 2026.

[28] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv:1807.03748, 2019.

[29] A. El-Nouby *et al.*, "Scalable pre-training of large autoregressive image models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research,

R. Salakhutdinov *et al.*, Eds., vol. 235. PMLR, Jul. 2024, pp. 12 371–12 384.

[30] Q. Garrido *et al.*, "Learning and leveraging world models in visual representation learning," arXiv:2403.00504, 2024.

[31] A. Linardos *et al.*, "DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 12 919–12 928.

[32] Y. Yue *et al.*, "CheXWorld: Exploring image world modeling for radiograph representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2025, pp. 20 778–20 788.

[33] Z. Yang *et al.*, "X-WIN: Building chest radiograph world model via predictive sensing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2026, pp. 6920–6930.

[34] Y. Wang and Q. Chen, "3D MRI image pretraining via controllable 2D slice navigation task," arXiv:2605.06487, 2026.

[35] J. Kömen *et al.*, "Towards robust foundation models for digital pathology," *Nature Communications*, vol. 17, no. 1, Jun. 2026.

[36] Y. Tolkach *et al.*, "Artificial intelligence for tumour tissue detection and histological regression grading in oesophageal adenocarcinomas: a retrospective algorithm development and validation study," *The Lancet Digital Health*, vol. 5, no. 5, pp. e265–e275, 2023.

[37] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[38] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 750–15 758.

[39] Q. Garrido, L. Najman, and Y. LeCun, “Self-supervised learning of split invariant equivariant representations,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause *et al.*, Eds., vol. 202. PMLR, Jul. 2023, pp. 10 975–10 996.

[40] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform manifold approximation and projection for dimension reduction,” arXiv:1802.03426, 2020.

[41] A. Sharafeldin, N. Imam, and H. Choi, “Active sensing with predictive coding and uncertainty minimization,” *Patterns*, vol. 5, no. 6, p. 100983, Jun. 2024.

Safe Reinforcement Learning for Power Converters with Hardware in the Loop

Benjamin Schwabe[1], Zhi Li[2] and Xaver Höchst[1]

[1]Infineon Technologies AG
Am Campeon 1-15, 85579 Neubiberg
E-Mail: {benjamin.schwabe,xaver.hoechst}@infineon.com

[2]Technische Universität München
Arcisstraße 21 80333 München
E-Mail: zhi.li@tum.de

1 Introduction

Switch-mode power supplies (SMPS) constitute a key component of most electronic systems, providing efficient voltage conversion to match the requirements of downstream loads to the available power source. By combining high efficiency with high power density, SMPS enable a broad range of applications that would otherwise be infeasible, ranging from portable computing devices and compact chargers to large-scale data center infrastructures supporting computationally demanding workloads such as the training of large language models, [1]. Furthermore, they enable power grids to handle different energy sources, like solar panels producing DC current in AC power grids, [2].
Despite their widespread adoption, the operating principle of SMPS introduces substantial challenges for analysis and control. Switch-mode power converters regulate energy transfer by repeatedly commutating the circuit between a finite set of discrete topological states through the gating of semiconductor devices such as MOSFETs and IGBTs. This inherently switched operation yields strongly nonlinear, and in many cases hybrid, system dynamics. In practice, however, regulation is most commonly implemented using linear

control structures such as PID controllers designed around linearized small-signal models obtained at a nominal operating point, [1, 3].
Such linear controllers perform adequately when the converter operates near the design point, but their limitations become apparent under the increasingly stringent requirements imposed by data-center environments. In these settings, the converter must maintain stability and high conversion efficiency across a wide range of operating points, from light-load conditions during server idle periods to heavy-load operation under compute-intensive workloads. Moreover, it must accommodate rapid transitions between these regimes while preserving tight output regulation and robust dynamic performance. Because the converter's dynamics vary significantly with operating point and switching behavior, a PID controller designed around a fixed small-signal model exhibits degraded regulation performance and reduced robustness under such wide-ranging and fast-changing conditions. Accordingly, more advanced control strategies — including model predictive control and neural network–based controllers — have received considerable attention for these applications.
The sequential and decision-making nature of converter regulation makes reinforcement learning (RL) a natural framework for training neural network–based controllers in this context. However, this training paradigm introduces a critical safety challenge: during exploration, the controller may encounter parameter configurations that drive the system into unsafe operating regimes. Certain network parameter sets, for instance, may induce large undamped oscillations that exceed the voltage or current ratings of converter components such as capacitors, potentially causing irreversible hardware damage. At the same time, visiting and identifying such unsafe regions is inherently necessary for the RL training process to effectively discriminate between beneficial and detrimental parameter configurations. Resolving this tension between the exploratory demands of RL and the safety requirements of power converter operation is therefore a central challenge in applying this paradigm to the present setting.
This work investigates several approaches for incorporating safety constraints into the training and deployment of neural network–based controllers for power converter applications. The analysis is conducted using a buck converter as a representative example throughout. Section 2 presents the operating principles of power converters and their associated control requirements, and details the limitations of linear regulation in this context. Section 3 introduces the relevant

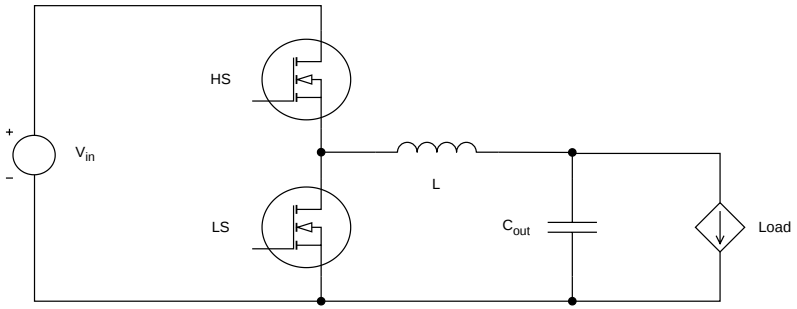


Figure 1: Schematic of a Synchronous Buck Converter.

background on neural network–based control and reinforcement learning. The setup for using actual measurement data from hardware in the loop (HiL) for reinforcement learning is examined in Section4. Subsequently, Section 5 then addresses the safety aspects that must be considered when applying these methods to power converter control.

2 Power Converter and Control

Buck converters, a class of DC/DC converters that step down input voltage to a lower output level, are frequently employed as the final conversion stage prior to the load, thereby earning the designation of point-of-load (PoL) converters [4]. The schematic of a synchronous buck converter is illustrated in Figure 1. In this configuration, a half-bridge comprising two metal-oxide-semiconductor field-effect transistors (MOSFETs) converts the DC input into a switched voltage waveform oscillating between 0 V and the input voltage. This high-frequency signal is subsequently filtered by an LC low-pass filter (inductor and capacitor) to produce a stable DC output [1].

Functionally, the buck converter operates as a closed-loop control system: the output voltage V_{out} is sensed and compared against a reference value V_{ref} to generate an error signal V_{err}. This error voltage is processed by a controller, which generates a control signal u. This signal is fed into a modulator, which produces a binary switching signal. The resulting signal is then conditioned

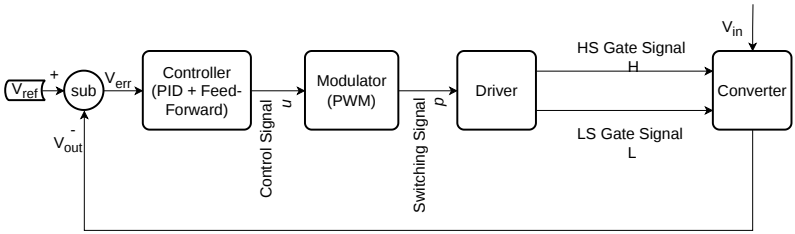


Figure 2: The Buck Converter as a Closed-Loop Controlled System.

by an inverter and amplifier stage to drive the gate terminals of the half-bridge MOSFETs (see Figure 2). The most commonly used modulator in such converters is the pulse-width modulator (PWM), [4]. In this context, the control signal u is interpreted as the duty cycle of the modulator; consequently, the control signal must satisfy the constraint $u(t) \in [0,1] \forall t$.
A PID controller, together with a feed-forward term, is employed in most power converters. The feed-forward term is derived from the steady-state conversion ratio

$$V_{out} = D \cdot V_{in}, \tag{1}$$

where $D \in [0,1]$ denotes the steady-state duty cycle, [1]. This linearization is based on state-space averaging and does not account for the converter's switching behavior, rendering the system nonlinear and time-varying on short time scales [4]. In data-center applications, point-of-load (PoL) converters are exposed to a wide range of load conditions and rapid load transitions. Consequently, this approximation approaches its limits in such scenarios, prompting both industry and academia to investigate the applicability of more elaborate and flexible control strategies, such as model predictive control (MPC) and neural-network-based control, which are capable of handling nonlinear dynamics and large operating point variations more effectively [5, 6].

3 Neural Network Based Control and Reinforcement Learning

Artificial neural networks have been investigated as feedback controllers in a variety of application domains [7, 8]. Due to their limited transparency, interpretable neural-network architectures have been proposed for power converter control [5]. Relative to classical control schemes, such approaches typically involve a substantially larger set of tunable parameters, which renders manual tuning and classical loop-shaping procedures impractical. Consequently, reinforcement learning (RL) constitutes a systematic framework for controller parameter optimization via interaction with the plant.
Following [9], the converter control problem can be formulated as a Markov Decision Process (MDP). For the buck converter, the continuous state space is defined by the inductor current and output voltage (equal to the capacitor voltage),

$$\mathcal{S} = \{s = (i_L, v_{out}) \in \mathbb{R}^2\},$$

and the action space is given by the controller output u, interpreted as the duty cycle command to the PWM modulator,

$$\mathcal{A} = \{a = u \in [0,1]\}.$$

The stochastic state evolution is captured by the transition kernel

$$P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$$

and each state–action pair is assigned an instantaneous reward according to

$$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}.$$

Given a discount factor $\gamma \in [0,1]$, the objective is to determine an optimal policy π^* that maximizes the expected cumulative discounted reward,

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{i=0}^{T} \gamma^i r_i \mid \pi\right]. \tag{2}$$

In the present context, the reward is chosen to reflect regulation quality and switching-related penalties. A commonly used quadratic form based on the error voltage $V_{err} = V_{ref} - V_{out}$:

$$r_t = -\left(V_{err}(t)\right)^2 - \lambda_I i_L^2(t) - \lambda_u \left(u_t - u_{t-1}\right)^2 , \tag{3}$$

where the weights $\lambda_I \in \mathbb{R}_0^+$ and $\lambda_u \in \mathbb{R}_0^+$ represent the penalties on inductor current magnitude and control-signal variation, respectively. The latter term discourages high-frequency modulation of the duty cycle, which may manifest as jitter and can degrade closed-loop stability and electromagnetic compatibility. Numerical simulations of buck converters generally provide only an approximation of the behavior of a physical implementation. To reduce computational complexity, parasitic effects are often neglected or simplified; for instance, the resistive losses of passive components and interconnects may be imperfectly known and can vary with temperature [10]. Moreover, accurately resolving the switching transients of the half-bridge MOSFETs and the associated gate-driver dynamics may require sub-nanosecond time resolution, which would result in prohibitively long simulation runtimes. These effects are therefore commonly omitted and idealized or reduced-order component models are used. Using measured data for the reinforcement training not only may speed up the process, but it should also yield better optimization results. In the context of robotics, the combination of simulation-based pre-training and real-work refinement has been established as sim-to-real approaches, which can be considered a special case of transfer learning, [11–14].

4 Reinforcement Learning and Hardware in the Loop

While simulation environments afford unrestricted access to all internal signals, this is not generally the case in hardware implementations. In most digital power converter controllers, the PWM modulator is integrated as a hardware peripheral, rendering the control signal u inaccessible as an explicit variable. Similarly, the error voltage V_{err} is not directly observable; rather, the output voltage V_{out} is digitized by an analog-to-digital converter (ADC) and a digital representation

of the reference voltage V_{ref} is subsequently subtracted to compute the error voltage. Furthermore, the sampling frequency of the output voltage is typically significantly higher than the switching frequency, the latter being defined by the period between two consecutive PWM pulses of the converter. Internal states of the controller, like the differential and integral term of a PID regulator are therefore not directly observable. Limited access to data has been identified as one of the main challenges for reinforcement learning in real-world applications, [16].
In contrast, the state variables of the power converter, the inductor current and output voltage can be observed. The output voltage V_{out} can be measured very accurately by an oscilloscope and power converters often contain circuitry, like a DCR current-sensing circuit, which gives access to the inductor current i_L, [15]. Consequently, the error terms based on the state variables in Equation (3) can be obtained. The last error term, however, cannot be assessed this way.
In order to overcome these limitations, a prototype equipped with a powerful field programmable gate array (FPGA) as controller together with an analog front-end board including ADC, modulation peripherals and a power board providing the power converter circuit with measurement ports. This prototype is depicted in Figure 3. Unlike a power converter control integrated chip, the FPGA is equipped with significant amount of memory and a high-speed interface, such that the controller input and output can be stored and transferred to a computer for further processing and provided to a reinforcement learning algorithm. A high-level diagram of such a hardware-in-the-loop test setup is shown in Figure 4. Ensuring the interface between the test manager and the real-world measurement and the interface between the test manager and the simulation model are compatible, a seamless transition between hardware-base training and simulation-based training can be achieved.

5 Safety aspects of Reinforcement Learning in Power Conversion

Beyond signal accessibility, the safe operation of the power converter during and after training represents a critical concern. The exploratory nature of

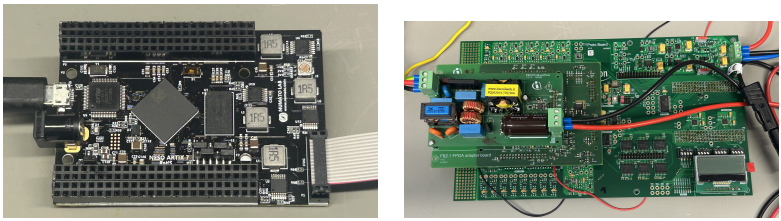


Figure 3: Pictures of an FPGA based prototype representing a power converter. The FPGA board is shown on the left. It is mounted to the bottom side of the large board providing the analog front end onto which the power board is mounted (right plot). While the elements mentioned in the left column are all hardware devices, the middle indicates the FPGA, the elements in the right column are realized with software only (e.g. MATLAB or Python).

reinforcement learning necessitates that suboptimal parameter configurations are applied to the hardware in order to evaluate their effect on the voltage response; only upon observing a low reward can the algorithm infer that the explored region of the parameter space is unlikely to yield satisfactory performance. This inherent exploration-exploitation trade-off introduces an unavoidable risk of damaging the converter or the connected load. Safety considerations of this nature have been addressed in broader control contexts, and a generalized safety framework has been proposed in [17].

Power converters typically incorporate a set of integrated protection mechanisms: an independent state machine continuously monitors input and output voltages, output current, and junction temperature against predefined limits. Upon violation of any of these boundaries, the controller may issue a fault warning or initiate a protective shutdown to prevent hardware damage. However, these fault conditions do not cover all failure modes that an ill-configured controller may induce. For instance, an admissible control parameter set may cause the converter to respond to a load transient with a sustained oscillatory output voltage that remains within the monitored safety bounds. Such periodic voltage fluctuations subject the output capacitors to repeated charge and discharge cycles within a short time interval, potentially causing thermal runaway, reducing lifetime of the capacitor or even resulting in a catastrophic capacitor failure without triggering any of the fault conditions monitored by the state machine, [18].

By extending the concept of transfer learning from a one directional sim-to-real

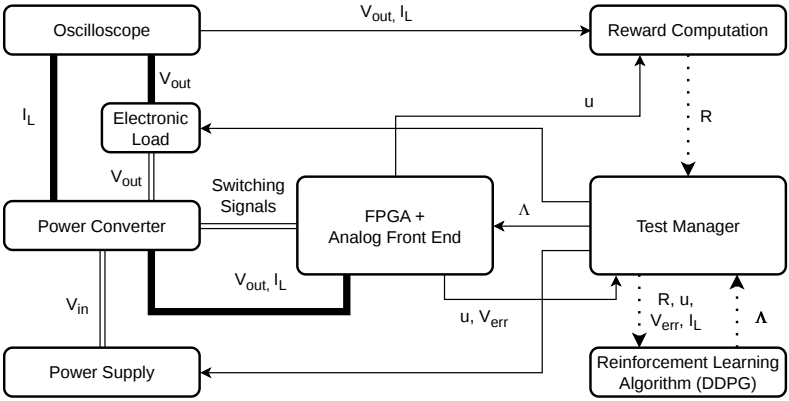


Figure 4: Diagram of a hardware setup enabling reinforcement with hardware in the loop. Double lined connections indicate an electric connection, bold connections indicate electrical measurement connections, while dashed lines simply indicate an entirely software based transfer of digital numbers. Arrows with solid lines indicate any form of control lines or digital transfer between devices.

approach to a bidirectional transfer between simulations and real world for each iteration of the training process, the safety concerns can be mitigated almost entirely using the approach sketched in Figure 5: For each iteration of the RL algorithm a candidate of the regulation parameters Λ^k is provided and a test execution manager decides if this test is being executed in simulation or using the real hardware. The simulation or the measurement equipment then produces the output o^k of the controller defined by

$$o^k = (u, v_{out}, i_L) \, .$$

Due to a limited temporal resolution of the simulation and measurement equipment, o^k would be a vector. Based on the output result o^k the accumulated reward R^k can be computed, which then completes the input required for all the most important reinforcement learning algorithms, like the deep deterministic policy gradient (DDPG) algorithm, [19].

The key component is the logic inside the test execution manager deciding when a simulation is selected over a real-world measurement or vice versa. The function in Algorithm 1 provides one possible implementation: The first

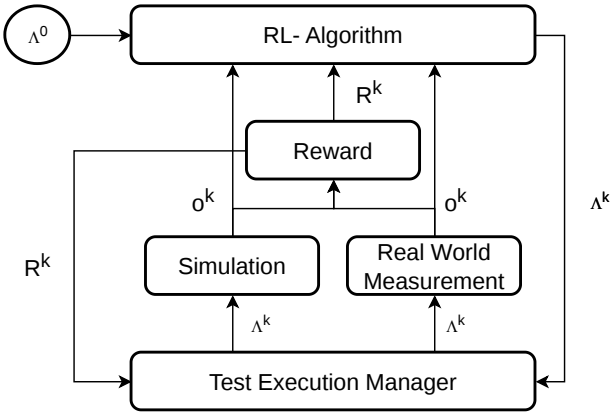


Figure 5: Flowchart of a safe reinforcement learning approach combining simulations and real data as input.

iteration has no information about the reward R^{k-1}. Hence, this value is set as a large negative value, specifically $R^{k-1} << R^{thres}$. This initialization ensures that the algorithm always chooses a simulation at the beginning. It will continue selecting the simulation until a parameter set was found for which the regulation is considered to be good enough to be safe for testing with the hardware. Mathematically speaking, the condition $R^{k-1} \geq R^{thres}$ is fulfilled. Then, the next iteration may be performed in hardware, if the next parameter set is close enough and therefore meets the second criterion: $||\Lambda^k - \Lambda^{k-1}|| \leq \Delta\Lambda^{thres}$.
This way only the risky tests are investigated with simulations, and only tests that are considered safe will be assigned to the hardware. For this sketched approach to converge, a good match of the simulation with the hardware is vital. More specifically, the reward value of a test performed in hardware must be close enough to a matching simulation.

6 Summary and Outlook

The stringent requirements of modern data centers pose significant challenges for power converter control, particularly regarding rapid and large-scale load transients. While classical PID regulation remains prevalent, its performance is

Algorithm 1.: Pseudo code describing the safe reinforcement learning approach using both simulations and real data as input. The threshold values R^{thres} and $\Delta\Lambda^{thres}$ are given parameters. For the first iteration, R^{-1} is set to a large negative value.

```
function TestExecutionManager(Λ^k, Λ^{k-1}, R^{k-1})
    Simulation ← ExecutionTarget      ▷ Initialize ExecutionTarget with simulation
    if R^{k-1} ≥ R^{thres} then       ▷ High reward indicates good regulation behaviour
        if ||Λ^k − Λ^{k-1}|| ≤ ΔΛ^{thres} then    ▷ step from k-1 to k is not an exploration
            Measurement ← ExecutionTarget
```

increasingly constrained by the nonlinear and time-varying dynamics inherent to point-of-load applications. Neural network-based controllers offer superior flexibility and adaptability. However, their practical deployment is hindered by the complexity of tuning weights and biases in a manner that is both efficient and safe. To address this, this work introduced a reinforcement learning (RL) framework, leveraging the structural similarities between RL and sequential control problems.

Although system simulations provide a viable initial training environment, the discrepancy between simulated and real-world dynamics necessitates transfer learning techniques to bridge the "reality gap." Furthermore, the use of real hardware during training introduces critical safety concerns, as in principal admissible parameter settings can lead to catastrophic failure of the converter or the load. To mitigate these risks, a hybrid training strategy is proposed that combines simulation-based exploration to ensure safety bounds with hardware-in-the-loop (HIL) exploitation to refine performance using real-world measurements. This approach balances the need for robust and fast policy optimization with the imperative of hardware protection.

Future work will focus on quantifying the impact of measurement noise and model-to-reality discrepancies on the convergence properties of RL-based control policies. Additionally, mitigation strategies to enhance the robustness and reliability of neural controllers under uncertain and noisy operating conditions need to be developed, in order to facilitate their broader adoption in industrial power electronics applications.

References

[1] R. W. Erickson and D. Maksimovic. “Fundamentals of Power Electronics”. Third Edition. Springer. ISBN 078-3-030-43879-1.

[2] M. Paramasivan “Soft-Switching Power Converter for Energy-Efficient Renewable Power Systems”. In: *9th International Conference on Circuit, Power & Computing Technologies (ICCPCT)*, pp. 2225-2230. 2026.

[3] C. P. Basso. “Switch-Mode Power Supplies: SPICE Simulations and Practical Designs”. Second Edition. Mc Graw Hill Education. ISBN 978-0-07-192346-3.

[4] M. K. Kazimierczuk. “Pulse-width Modulated DC-DC Power Converters”. Wiley. ISBN: 978-0-470-77301-7.

[5] B. Schwabe, A. Bach and T. Senjab.“Interpretable AI for Power Conversion”. In: *2023 Applied Power Electronic Conference and Exhibition (APEC)*. Orlando, Florida, USA.

[6] N. Snehal, W. Pooja and K. Sonam. “Model Predictive Control for DC-DC Buck Converter using Enumeration”. In: *2020 52nd North American Power Symposium (NAPS)*, pp. 1-6. Tempe, Arizona, USA.

[7] L. Böttcher “Control of dynamical systems with neural networks”. Online: *https://arxiv.org/html/2510.12810v1*

[8] G. Norris, G. J. J. Ducard and C. Onder. “Neural Networks for Control: A Tutorial and Survey of Stability-Analysis Methods, Properties, and Discussions”. In: *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1-6. Mauritius, Mauritius, 2021.

[9] S. K. Jayaweera, “Markov Decision Processes”. In: *Signal Processing for Cognitive Radios*. pp.207-268. Wiley. 2015.

[10] J .Wallace. “What’s Not in the Power MOSFET Data Sheet, Part 1: Temperature Dependency”. Technical Paper by Texas Instruments . URL: https://www.ti.com/lit/ta/sszt206/sszt206.pdf .

[11] W. Zhao, J. P. Queralta and T. Westerlund “Sim-to-real transfer in deep reinforcement learning”. URL: https://arxiv.org/abs/2009.13303

[12] M. Picerno, L. Koch, K. Badalian, M. Wegener, J. Schaub, C. R. Koch and J. Andert “Transfer of Reinforcement Learning-Based Controllers from Model- to Hardware-in-the-Loop”. URL: https://arxiv.org/abs/2310.17671

[13] F. Zhuang ,Z. Qi ,K. Duan,D. Xi,Y. Zhu, H. Zhu,H. Xiong and Q. He “A comprehensive survey on transfer learning”. URL: https://ieeexplore.ieee.org/abstract/document/9134370

[14] M.E. Taylor and P. Stone “Transfer Learning for Reinforcement Learning Domains: A Survey”; In: *2009 Journal of Machine Learning Research*.

[15] S. Tian, W. Ballar, X. Chen, Y. Yan, Yingyi, Z. Liu, R. Ying and E. Beville. “"A Novel DCR Current Sensing Scheme for Accurate Current Readback in Power uModule Applications"”. In: *2020 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 2763-276. New Orleans, LA, USA, 2020.

[16] G. Dulac-Arnold, D. Mankowitz, T. Hester. “Challenges of real-world reinforcement learning”. URL: https://arxiv.org/abs/1904.12901

[17] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, A. Knoll. “A review of safe reinforcement learning: Methods, theory and applications”. URL: https://arxiv.org/abs/2205.10330

[18] M. Novak, W. Hang and F. Blaabjerg “Experimental Validation of Capacitor Damage Accumulation in Varying Operating Conditions”. In: *2024 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 7186-7191.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra “Continuous Control with deep Reinforcement Learning”. URL: https://arxiv.org/pdf/1509.02971 .

Comparing Extensions of DeePC: A Simulation Study

Marcel Zehner, Alessio Cavaterra, Steven Lambeck

FB Elektrotechnik und Informationstechnik, Hochschule Fulda
Leipziger Str. 123, 36037 Fulda
E-Mail: {marcel.zehner,alessio.cavaterra,steven.lambeck}@et.hs-fulda.de

1 Introduction

Data-Enabled Predictive Control (DeePC) [1] is a data-driven counterpart to Model Predictive Control (MPC) [2] and has gained interest in academia in recent years. DeePC uses a system's past input and output trajectories to compute the optimal control sequences while minimising a cost function and taking into account input and output constraints. Unlike MPC, the DeePC approach does not require a parametric model to be determined in advance using offline system identification. DeePC is based on the *behavioral systems theory* [3] and Willems' *fundamental lemma* [4], established for linear time-invariant (LTI) plants. Within this theoretical framework, a LTI system is defined as a set of past input and output trajectories. Furthermore, any possible trajectory of an LTI system can be constructed from a finite number of past trajectories of the same LTI system. Based on this idea, input and output data can be utilised as a nonparametric model in an optimal control problem (OCP). Although there is a solid theoretical basis for data-driven predictive control of linear systems, the practical implementation requires extensions that also take nonlinear processes into consideration. In this regard, several nonlinear extensions have been published in recent years. In [5], a nonlinear extension based on basis functions is presented. The Basis-functions DeePC or ϕ-DeePC transforms the data into a feature space where the behaviour of the system is approximately linear and is then used in an OCP. Furthermore, there are nonlinear DeePC approaches [6, 7]

which use the data in the OCP that is closest to the actual operating point at each optimisation step. In [8], the DeePC was extended using Takagi-Sugeno fuzzy systems (TS). The TS variants weight the data set according to fuzzy rules, either on a regular grid or by Fuzzy c-Means (FCM) clustering and blend the weighted local nonparametric predictors. It has been shown that the weighting of the data has led to an improvement in control performance compared with the standard DeePC formulation. The work cited from [9] extends DeePC to linear parameter-varying (LPV) systems, thereby enabling a nonlinear version of DeePC known as LPV-DeePC. This extension utilises a global scheduling-dependent predictor from input-output and scheduling data. For an overview of nonlinear extensions of DeePC, interested readers are referred to [10]. All these extensions are validated on their own benchmarks, ranging from quadcopters and pendulums to autonomous racing and chemical reactors with individually tuned parameters, individually generated data sets and different solvers, in some cases on noise-free data only. However, a systematic benchmark under identical conditions for all previously mentioned DeePC approaches is still missing. We try to close this gap in this work. Consequently, it remains unclear for a practitioner what benefits an extension brings in terms of accuracy and what the costs are in terms of incorporating process knowledge, data, tuning and computation time.

This work contributes a comparison under a unified implementation framework. Seven controllers are considered: (i) standard linear DeePC [1], (ii) Basis-functions DeePC [5], (iii) Contextual-sampling DeePC [6], (iv) G-DeePC and (v) FCM-DeePC [8], together with its column-selecting variant (vi) FCM-Select-DeePC, as well as (vii) LPV-DeePC [9]. The sixth method FCM-Select-DeePC represents the second main contribution of this work. All of the methods are supplied with the same excitation data and face identical constraints, reference trajectories and the resulting convex quadratic programs (QP) are solved by the same solver, namely qpOASES [11]. The methods differ only in how the nonlinearity enters the multi-step predictor, so that the observed differences can be attributed to the predictor construction alone. As a benchmark a simulated SISO process is used: a mechanical oscillator with a nonlinear and position-dependent stiffness as well as a nonlinear damping characteristic. The controllers are then compared in terms of tracking accuracy, computation time, noise robustness as well as regularisation sensitivity and

incorporating process knowledge. The paper is structured as follows: Section 2 introduces the nonparametric system model and the standard DeePC formulation, and section 3 presents the nonlinear extensions. The benchmark system and the simulation setup is described in section 4. The results of the simulations and the discussion are shown in section 5. Section 6 concludes the paper with an outlook.

2 Data-Enabled Predictive Control

2.1 Nonparametric System Representation

Unlike MPC, DeePC does not use parametric model structures such as linear state space models. Data-driven methods employ nonparametric models based on the *behavioral systems theory* [3] and Willems' *fundamental lemma* [4]. Thus, it is possible to represent the following controllable discrete-time LTI system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k + Du_k, \end{aligned} \tag{1}$$

with the state $x_k \in \mathbb{R}^n$, the input $u_k \in \mathbb{R}^m$ and the output $y_k \in \mathbb{R}^p$ as a nonparametric model using only input-output data from (1). To construct a predictor for subsequent use in the OCP, the input and output data are organised into Hankel matrices. For an input signal $u = \{u_i\}_{i=1}^{T_d}$ with length T_d and $T_d \geq L$, the following Hankel matrix is obtained:

$$\mathcal{H}_L(u) := \begin{pmatrix} u_1 & u_2 & \cdots & u_{T_d-L+1} \\ u_2 & u_3 & \cdots & u_{T_d-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L+1} & \cdots & u_{T_d} \end{pmatrix}. \tag{2}$$

Definition 1 (Persistence of excitation [1,4])**.** The input signal u_d is said to be persistently exciting of order L if $\mathcal{H}_L(u_d)$ has full row rank.

Based on Definition 1, Willems' *fundamental lemma* states that a single sufficiently exciting input-output trajectory spans the set of all trajectories of an LTI system.

Lemma 1 (Fundamental lemma [4])**.** *Let* (u_d, y_d) *be an input-output trajectory of length* T_d *of the controllable LTI system* (1) *and let* u_d *be persistently exciting of order* $L+n$*. Then a sequence* (u, y) *with* $u \in \mathbb{R}^{m \cdot L}$ *and* $y \in \mathbb{R}^{p \cdot L}$ *is an input-output trajectory of* (1) *if and only if there exists a vector* $g \in \mathbb{R}^{T_d - L + 1}$ *such that*

$$\begin{pmatrix} \mathcal{H}_L(u_d) \\ \mathcal{H}_L(y_d) \end{pmatrix} g = \begin{pmatrix} u \\ y \end{pmatrix}. \tag{3}$$

To put it in other terms, the result from [4] states that any trajectory of an LTI system can be constructed from a finite number of past trajectories. In this case, the columns of the Hankel matrices, which each represent a system trajectory of length L, are linearly combined. This results in a possible trajectory of the system. The vector g can thus be understood as the weighting of the individual columns of the Hankel matrices. It is important to highlight that this exact representation only holds for LTI systems and noise-free data. Noisy data are addressed by the regularised formulation in Subsection 2.2, while nonlinear dynamics motivate the extensions in Section 3. For a nonparametric prediction and its usage in the OCP, the trajectory window is divided into a past window of length T_{ini}, which implicitly encodes the initial condition, and a future window of length N_P, the prediction horizon. Choosing $L = T_{\text{ini}} + N_P$, the Hankel matrices in (3) are split as follows:

$$\begin{pmatrix} U_\text{p} \\ U_\text{f} \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N_P}(u_d), \quad \begin{pmatrix} Y_\text{p} \\ Y_\text{f} \end{pmatrix} := \mathcal{H}_{T_{\text{ini}}+N_P}(y_d). \tag{4}$$

Here, $U_\text{p} \in \mathbb{R}^{m \cdot T_{\text{ini}} \times (T_d - T_{\text{ini}} - N_P + 1)}$ and $Y_\text{p} \in \mathbb{R}^{p \cdot T_{\text{ini}} \times (T_d - T_{\text{ini}} - N_P + 1)}$ consist of the first $m \cdot T_{\text{ini}}$ and $p \cdot T_{\text{ini}}$ rows of $\mathcal{H}_L(u_d)$ and $\mathcal{H}_L(y_d)$, respectively. By analogy, U_f and Y_f are made up of the remaining $m \cdot N_P$ and $p \cdot N_P$ rows. If we now introduce $(u_{\text{ini}}, y_{\text{ini}}) = \{u_{k-i}, y_{k-i}\}_{i=T_{\text{ini}}}^{1}$ as the currently measured data of length T_{ini}, (4)

can be expressed as follows:

$$\begin{pmatrix} U_\mathrm{p} \\ Y_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{f} \end{pmatrix} g = \begin{pmatrix} u_\mathrm{ini} \\ y_\mathrm{ini} \\ u \\ y \end{pmatrix}. \tag{5}$$

This nonparametric system representation in (5) can therefore be used in data-driven predictive control approaches. Given the measured signals $(u_\mathrm{ini}, y_\mathrm{ini})$ and a candidate input u, every g that satisfies the first three block rows of (5) yields a purely data-driven prediction $y = Y_\mathrm{f}\, g$.

2.2 The Regularised DeePC Problem

The nonparametric model given by (5) represents the system model in the data-driven predictive control approach and thus replaces parametric models such as those found in MPC approaches. The optimisation problem (6), proposed in [1], can be used for the predictive control of LTI systems without measurement noise:

$$\min_{g,u,y} \sum_{k=0}^{N_P-1} \ell(y_k, u_k) \tag{6a}$$

$$\text{s.t. (5)} \tag{6b}$$

$$u_k \in \mathcal{U}, \forall k \in \{0, \ldots, N_P - 1\}, \tag{6c}$$

$$y_k \in \mathcal{Y}, \forall k \in \{0, \ldots, N_P - 1\}. \tag{6d}$$

Here, $u = (u_0, \ldots, u_{N_P-1})$ and $y = (y_0, \ldots, y_{N_P-1})$ are the future input and output trajectories from (5), where the index k counts the prediction steps relative to the current sampling instant. The cost term $\ell(y_k, u_k)$ in (6a) penalises the deviation of the controlled variable y_k from the reference vector r_k, as well as the manipulated variable u_k:

$$\ell(y_k, u_k) = \|y_k - r_k\|_Q^2 + \|u_k\|_R^2. \tag{7}$$

In (7), $\|x\|_M := \sqrt{x^T M x}$ denotes the weighted 2-norm, $r_k \in \mathbb{R}^p$ is the reference at prediction step k, and $Q \in \mathbb{R}^{p\times p}$, $Q \succeq 0$, and $R \in \mathbb{R}^{m\times m}$, $R \succ 0$, are positive semidefinite and positive definite weighting matrices, respectively. N_P and T_{ini} are two additional tuning parameters. The input u_k and the output y_k are constrained to the convex sets $\mathcal{U}$ and $\mathcal{Y}$ in (6c) and (6d), respectively. Constraint (6b) embeds the nonparametric system representation (5) as the prediction model. The OCP is solved in a receding-horizon fashion. However, this formulation only applies to LTI systems without measurement noise. As measurement noise is always present in real-world processes, the optimisation problem in (6) must be adapted. The authors of [1] propose the use of regularisation and slack variables. Here, both past block rows of (5) are relaxed by the slack variables $\sigma_u \in \mathbb{R}^{mT_{\text{ini}}}$ and $\sigma_y \in \mathbb{R}^{pT_{\text{ini}}}$, which are penalised in the cost function. Together with the regularisation of g, this leads to the following regularised OCP:

$$\min_{g,u,y,\sigma_u,\sigma_y} \sum_{k=0}^{N_P-1} \ell(y_k,u_k) + \lambda_g \|g\|_2^2 + \lambda_u \|\sigma_u\|_2^2 + \lambda_y \|\sigma_y\|_2^2 \tag{8a}$$

$$\text{s.t.} \begin{pmatrix} U_\text{p} \\ Y_\text{p} \\ U_\text{f} \\ Y_\text{f} \end{pmatrix} g = \begin{pmatrix} u_{\text{ini}} + \sigma_u \\ y_{\text{ini}} + \sigma_y \\ u \\ y \end{pmatrix}, \tag{8b}$$

$$u_k \in \mathcal{U}, \forall k \in \{0,\dots,N_P-1\}, \tag{8c}$$

$$y_k \in \mathcal{Y}, \forall k \in \{0,\dots,N_P-1\}. \tag{8d}$$

The slack variables σ_u and σ_y ensure feasibility in the presence of measurement noise and are kept small by the penalty weights λ_u and λ_y, while the regularisation term $\lambda_g \|g\|_2^2$ mitigates overfitting to noisy data. The weights λ_g, λ_u and λ_y are therefore tuning parameters. Especially the influence of λ_g as regularisation parameter is investigated in Subsection 5.4. (8) is a convex QP whose size scales with the number of columns of the data matrices. The nonlinear extensions presented in Section 3 retain the structure of (8) and differ only in replacing the predictor by method-specific constructions of the data matrices and, where required, in extending the cost function (8a) by additional penalty terms.

3 Extensions of DeePC to Nonlinear Systems

3.1 Basis-functions DeePC

Basis-functions DeePC or ϕ-DeePC [5] handles nonlinearity globally by mapping the recorded data into a feature space in which the nonlinear process is approximately linear. General basis functions are used for this transformation, e.g. radial, polynomial or orthogonal basis functions. The transformation is defined by a basis map $\bar{\phi}$. The argument $\xi \in \mathbb{R}^{(m+p)T_{ini}}$ combines input and output trajectories of length T_{ini} into a vector. The map $\bar{\phi}$ lifts ξ to

$$\bar{\phi}(\xi) = \begin{pmatrix} 1 \\ \xi \\ \phi_1(\xi) \\ \vdots \\ \phi_{n_\phi}(\xi) \end{pmatrix}, \quad \text{where} \quad \phi_j(\xi) = \exp\left(-\frac{\left\|\xi - c_j\right\|_2^2}{2\beta^2}\right). \tag{9}$$

In $\bar{\phi}(\xi)$, the first entry 1 accounts for affine constant terms. The entry ξ corresponds to the linear DeePC. Overall there are n_ϕ Gaussian radial basis functions with centres c_j and width β. The centres are obtained directly from the recorded data, whereby in [5], the centres are learned using a radial basis function (RBF) network. Applying $\bar{\phi}$ to the columns of the data matrices U_p and Y_p results in the lifted data matrix Φ. With the lifted currently measured data $\phi_{\text{ini}} := \bar{\phi}(u_{\text{ini}}, y_{\text{ini}})$ one can replace the predictor in (8b) with the lifted counterpart:

$$\begin{pmatrix} \Phi \\ U_{\text{f}} \\ Y_{\text{f}} \end{pmatrix} g = \begin{pmatrix} \phi_{\text{ini}} + \sigma_\phi \\ u \\ y \end{pmatrix}. \tag{10}$$

The slack variable σ_ϕ substitutes σ_u and σ_y and replaces both terms in the cost function (8a) with $\lambda_\phi \left\|\sigma_\phi\right\|_2^2$. Following [5], the basis is restricted to functions that are linear in the present and future inputs which results in Koopman-DeePC as a special case of Basis-functions DeePC. Therefore, U_{f} and Y_{f} remain

unchanged and the convex QP structure of (8) is preserved. To get consistent predictions we use the dynamic regularisation proposed and recommended in [5]. For further information on the dynamic regularisation, interested readers are referred to [5]. The other tuning parameters for this extension are the number of basis functions n_ϕ, their centres c, their width β and λ_ϕ for the slack variable σ_ϕ and λ_{r1} for the dynamic regularisation.

3.2 Contextual-Sampling DeePC

Contextual-sampling DeePC (CS-DeePC) [6] takes advantage of the fact that a class of nonlinear systems can be approximated at operating points using linear dynamics. For each optimisation step, the CS-DeePC selects the data that are closest to the operating point. This ensures that the prediction and optimisation are carried out using only data points in the vicinity of the actual operating point. A similar idea was proposed in [7]. The columns of the Hankel matrices in (4) contain a system trajectory of length L. Given that $(u_{\mathrm{ini}}, y_{\mathrm{ini}})$ describe the current operating point, the relevant columns are selected using the following distance measure:

$$d_i = \left\| W \left(\begin{pmatrix} u_{\mathrm{ini}} \\ y_{\mathrm{ini}} \end{pmatrix} - \begin{pmatrix} U_{\mathrm{p},i} \\ Y_{\mathrm{p},i} \end{pmatrix} \right) \right\|_2 . \tag{11}$$

$U_{\mathrm{p},i}$ and $Y_{\mathrm{p},i}$ represent the i-th columns of the past data matrices. W is a scaling matrix that ensures that all input and output trajectories contribute equally to the distance measure. The tuning parameter N_s specifies the cardinality of the following set and stores the column indices with the smallest distances:

$$\mathcal{J} = \underset{|\mathcal{J}|=N_s}{\operatorname{argmin}} \sum_{i \in \mathcal{J}} d_i. \tag{12}$$

Choosing the indices from the Hankel matrices in (4) gives us the reduced Hankel matrices $U_{\mathrm{p},\mathcal{J}}$, $Y_{\mathrm{p},\mathcal{J}}$, $U_{\mathrm{f},\mathcal{J}}$ and $Y_{\mathrm{f},\mathcal{J}}$, which replace the predictor in (8b). The cost function (8a) is not modified. The choice of columns does not guarantee the persistence of excitation condition in Definition 1. The number of optimisation

variables is defined by N_s. As a result, the OCP of CS-DeePC has less decision variables than the one in the standard DeePC formulation.

3.3 Takagi-Sugeno Fuzzy DeePC

TS fuzzy systems can be used to approximate the global dynamics of nonlinear processes. This is achieved by using local linear models weighted by membership functions [35, 40]. This approach has been applied to the DeePC formulation in [8], where G-DeePC and FCM-DeePC were described. In this case, the collected input and output data are weighted according to fuzzy rules based on a regular grid structure or FCM, resulting in N_r nonparametric local models consisting of $U_p^{(i)}, U_f^{(i)}, Y_p^{(i)}, Y_f^{(i)}$. The submodels are calculated using the fuzzy basis functions $h_i(\mathbf{z}_k)$, based on the regular grid structure or FCM. $\mathbf{z}_k$ is the premise vector, which contains entries from the initialisation trajectories $(u_{\text{ini}}, y_{\text{ini}})$. Each column j of the Hankel matrices is weighted by $h_i(\mathbf{z}_j)$. This is done using the diagonal weighting matrix $W_i = \text{diag}\left(h_i(\mathbf{z}_1), \ldots, h_i(\mathbf{z}_{T_d-L+1})\right)$. The i-th local nonparametric submodel is calculated using:

$$U_p^{(i)} = U_p W_i, \quad Y_p^{(i)} = Y_p W_i, \quad U_f^{(i)} = U_f W_i, \quad Y_f^{(i)} = Y_f W_i. \tag{13}$$

During control, the resulting nonparametric model, as a weighted sum of the N_r submodels, is calculated as a function of the current premise variable in the following way:

$$\bar{U}_{\text{p}}(\mathbf{z}_k) = \sum_{i=1}^{N_r} h_i(\mathbf{z}_k)\, U_{\text{p}}^{(i)}. \tag{14}$$

The matrices $\bar{Y}_{\text{p}}$, $\bar{U}_{\text{f}}$ and $\bar{Y}_{\text{f}}$ are constructed in the same fashion and replace the data matrices in (8b). The TS-variants differ in how the premise space is partitioned. G-DeePC covers every premise variable with membership functions of a common width [14]. The tuning parameters are the number of membership functions per premise variable and the width α of the membership functions. FCM-DeePC partitions the premise space with Fuzzy c-Means clustering [23, 40]. The tuning parameters are the number of clusters and the fuzziness exponent ν. As part of this work, FCM-Select DeePC is being introduced and is included in the comparison. This variant keeps the N_s

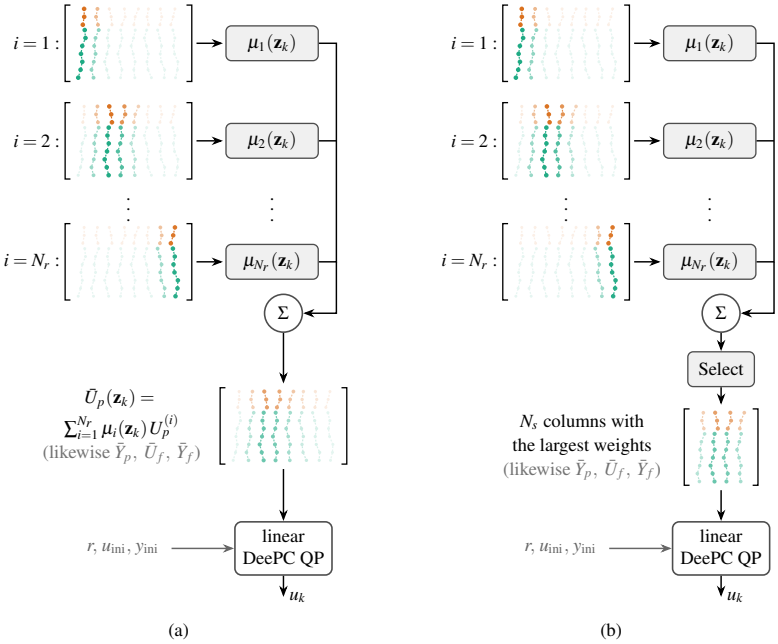


Figure 1: a.) Approach for FCM- or G-DeePC to determine the currently valid nonparametric model. b.) The "Select" variant of FCM- or G-DeePC, which allows the selection of the N_s columns with largest weights.

columns with the largest membership values at the current operating point. The membership of a column is the weight that is assigned to it through the fuzzy rules. This idea is taken from [23], where the memberships act as data weights when the local parametric models of a TS system are estimated. The new variant makes the size of the optimisation problem independent of the number of columns in the Hankel matrices. It is also possible to apply this to the G-DeePC approach. Figure 1 illustrates the approach taken by the TS Fuzzy DeePC methods, whereby the nonparametric submodels are weighted and summed according to the current premise to represent the currently valid model. The Select variant, as a new contribution, selects the N_s columns with the highest membership functions, which reduces the size of the used data matrices and therefore the size of the optimisation problem.

3.4 LPV-DeePC

The LPV framework makes it possible to handle nonlinear processes. Here, a linear relation exists between the input, state and output signals. This relation is varied by a measurable and time-varying scheduling signal ρ, which captures the nonlinear effects of the process. The work in [9] presents an extension of DeePC to LPV systems and, with it, an LPV fundamental lemma, which states that the recorded data span all trajectories of a LPV system under a *generalised persistence-of-excitation* condition. For this, in addition to u and y, the scheduling signal ρ and the pointwise products $u_d^\rho = \{\rho_{d,i}\, u_{d,i}\}_{i=1}^{T_d}$ and $y_d^\rho = \{\rho_{d,i}\, y_{d,i}\}_{i=1}^{T_d}$ must also be expressed as Hankel matrices in the nonparametric system representation (stated for a scalar scheduling signal and SISO dynamics, see [9] for the general case). The predictor (8b) is extended by the following terms:

$$\begin{aligned} \left(\mathcal{H}_L(u_d^\rho) - \mathcal{P}\,\mathcal{H}_L(u_d)\right) g &= \sigma_{\rho u}, \\ \left(\mathcal{H}_L(y_d^\rho) - \mathcal{P}\,\mathcal{H}_L(y_d)\right) g &= \sigma_{\rho y}, \end{aligned} \tag{15}$$

where $\mathcal{P} := \mathrm{diag}(\rho_{\mathrm{ini}}, \hat{\rho})$ contains both the most recent measured values $\rho_{\mathrm{ini}} \in \mathbb{R}^{T_{\mathrm{ini}}}$ and the prediction values $\hat{\rho} \in \mathbb{R}^{N_P}$ of the scheduling signal over the horizon. There are several ways to determine $\hat{\rho}$ [9]. For the scope of this article, $\hat{\rho}$ is obtained in a gain-scheduled fashion by holding the current scheduling value constant over the prediction horizon. The slack variables $\sigma_{\rho u}, \sigma_{\rho y} \in \mathbb{R}^L$ relax the equality constraint in a way analogous to σ_u and σ_y, and are penalised in the cost function (8a) with the term $\lambda_\rho \left(\left\| \sigma_{\rho u} \right\|_2^2 + \left\| \sigma_{\rho y} \right\|_2^2 \right)$. In the noise-free case, the constraints hold with $\sigma_{\rho u} = \sigma_{\rho y} = 0$. To maintain consistency with the OCP in (8), both terms have been relaxed with slack variables. A convex QP results for the LPV-DeePC, as the scheduling signal is known at every optimisation step and the constraints (15) are linear in g. If the scheduling signal is constant, the left-hand sides of (15) vanish and the standard DeePC is recovered as a special case [9]. The single additional tuning parameter is the weight λ_ρ.

4 Benchmark System and Simulation Setup

4.1 Benchmark System

A mass-spring-damper system [16] with a nonlinear spring characteristic and a nonlinear damper characteristic is used as the benchmark system. These nonlinearities represent phenomena that occur in fluid drives. The nonlinear spring can be interpreted as the increase in the stiffness of the air column in the cylinder chambers, caused by the reduction in chamber volume during large piston displacement. This position-dependent stiffness can be expressed as a cubic function. The damping characteristic in servopneumatic or hydraulic drives is caused by the mass flow-pressure relation and viscous friction between the cylinder wall and the piston. Both nonlinear terms can be found in equation (16a). Let m be the mass, c_0 and c_1 the stiffness coefficients, d_0 and d_1 the damping coefficients, x_1 the position x of the mass and x_2 the velocity $\dot{x}$ of the mass, u the input as a force F applied to the mass, then the following system results:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{c_0}{m} & -\frac{d_0}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{c_1}{m} {x_1}^3 - \frac{d_1}{m} \operatorname{sgn}(x_2) \sqrt{|x_2|} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u, \tag{16a}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{16b}$$

The coefficients as well as the working range $|u| \leq 3.5\,\mathrm{N}$ and $|y| \leq 0.19\,\mathrm{m}$ of the system have been taken from [16]. The nonlinearities of the system occur in both x_1 and x_2. The damping characteristic of x_2 is continuous but not differentiable at $x_2 = 0$. The system is globally stable and the excitation data can be collected in open loop.

4.2 Data Acquisition

All methods work with the same data set and face the same control task. The controllers operate at a sampling time of $T_s = 50\,\text{ms}$. Every measurement of x_1 is corrupted by additive white Gaussian noise with a standard deviation of 2 mm. An amplitude-modulated pseudo-random binary sequence (APRBS) with a minimum hold time of two samples is applied for 20 s as an excitation signal, which results in a data trajectory of length $T_d = 400$. With $T_{\text{ini}} = 4$ and $N_P = 10$, the data matrices in (4) have 387 columns. After the excitation phase the controllers run for 40 s, which corresponds to 800 solved QPs per simulation.

4.3 Method Configurations

All methods share the weights $Q = 12$ and $R = 0.01$ as well as the slack penalties $\lambda_u = \lambda_y = 10^6$. The method-specific parameters were tuned by grid searches, so that each method is compared at its own optimum. Table 1 summarises the resulting configurations. All TS-variants use the premise vector $\mathbf{z} = [y_{k-1},\, y_{k-1} - y_{k-T_{\text{ini}}}]^\top$, with measured position and the difference of the position values over the initialisation window. G-DeePC covers the premise space with 12 equidistantly placed Gaussian membership functions. The parameters for FCM-DeePC and FCM-Select DeePC can be taken from Table 1. For ϕ-DeePC, the $n_\phi = 20$ centres are taken from the recorded data columns and the common width β as the median of the distances between all columns and the centres. LPV-DeePC uses a two-component scheduling signal $\rho = (\rho_1, \rho_2)$ with the spring channel $\rho_1 = (x/x_{\text{nom}})^2$ and the damper channel $\rho_2 = \sqrt{\dot{x}_{\text{nom}} / \max(|\dot{x}|, \varepsilon_v)}$, where $x_{\text{nom}} = 0.15$ m and $\dot{x}_{\text{nom}} = 0.3\,\text{m}\,\text{s}^{-1}$ normalise both channels. ε_v is required because the damper channel is singular at zero velocity and inside the band a linear viscous damping law is used.

[1] There is a total of $N_r = 12$ submodels. 6 submodels for the premise variable y_{k-1} and 2 submodels for the premise variable $y_{k-1} - y_{k-T_{ini}}$.

Table 1: Configurations of the compared DeePC extensions

method	specific parameters	λ_g
DeePC	—	10
ϕ-DeePC	$n_\phi = 20$, $\lambda_\phi = 10^6$, $\lambda_{r1} = 10^4$	3
CS-DeePC	$N_s = 75$	0.5
G-DeePC	6×2 grid[1], $\alpha = 0.25$	0.1
FCM-DeePC	$N_r = 4$, $\nu = 1.6$	1
FCM-Select DeePC	$N_s = 160$	1
LPV-DeePC	$\varepsilon_v = 0.2\,\mathrm{m\,s^{-1}}$, $\lambda_\rho = 10^5$	3

5 Simulation Results and Discussion

5.1 Tracking Accuracy

Following the data acquisition, the system enters a 40-second control phase. A sinusoidal signal of the form $r(t) = A\sin(2\pi t/T_{\mathrm{per}})$, where $A = 0.12\,\mathrm{m}$ and $T_{\mathrm{per}} = 20\,\mathrm{s}$, is used as a reference. The *normalised mean squared error* (NMSE) is used as the evaluation criterion for tracking accuracy. To eliminate the settling phase following the data acquisition, the first 2 s of data are discarded. Both the excitation data and the data recorded during control have been corrupted with a noise signal whose standard deviation is $\sigma = 2\,\mathrm{mm}$. The following Table 2 shows the tracking performance achieved by the seven methods measured with the NMSE. It can be seen that all nonlinear extensions outperform the standard DeePC. The LPV-DeePC and the ϕ-DeePC demonstrate the best tracking accuracy, followed by the TS-variants and the CS-DeePC. Figure 2 shows the time series of the controlled variable y and the manipulated variable u for all seven methods over a period of 40 s. Here, too, it is noticeable that the nonlinear extensions outperform DeePC in terms of tracking accuracy. This is most apparent at the peaks of the sine wave, where the standard DeePC exhibits the largest control error. This is because the nonlinear terms of the spring and damper have their most significant effect at these points. The distribution of the control error is illustrated in Figure 3. The error distribution for all controllers is approximately normally distributed. Here, too, the advantage of the nonlinear

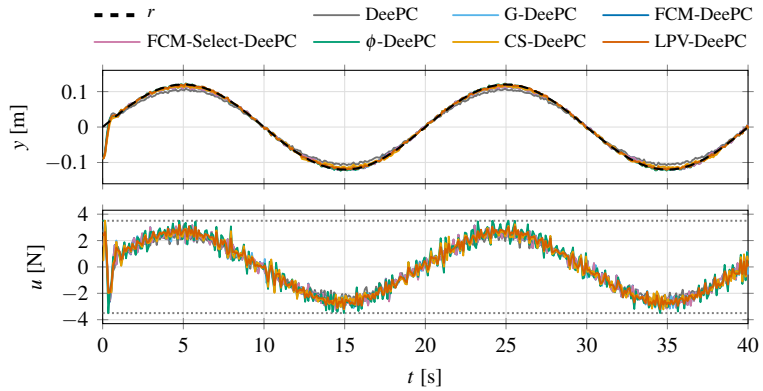


Figure 2: Tracking performance of the different DeePC extensions with noisy data, where $\sigma = 2$ mm. The dotted grey lines represent the constraints for u. The control error can be found in Figure 3 visualised as a violin plot.

extensions in terms of tracking accuracy is noticeable. In addition, the LPV-DeePC and the ϕ-DeePC stand out amongst the nonlinear extensions, as the NMSE had already shown. For both methods the interquartile range (IQR) of the error distributions lies essentially within the measurement noise band.

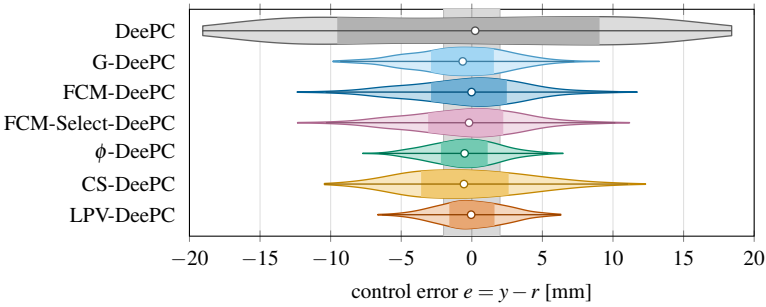


Figure 3: Violin plot of the control error for all DeePC extensions. The white point is the median and the dark shaded areas are the IQR. The grey area indicates the $\pm\sigma$-band of noise.

5.2 Computational Load

Figure 4a shows the boxplot of the QP solve time for all 800 optimisation steps during control, obtained with qpOASES [11] on an *Intel Core i7-1255U* processor. The standard formulation and ϕ-DeePC do not change the data matrices during control and therefore have a fixed Hessian for the optimisation problem. As a result, the QPs can be warm-started, giving median QP solve times of 1.02 and 1.07 ms. G-DeePC, FCM-DeePC and LPV-DeePC all update their data matrices at every sampling step, which means the matrices for the optimisation problem must be rebuilt at every step, resulting in median solve times of 9.15–9.68 ms. FCM-Select-DeePC and CS-DeePC also rebuild their matrices at every step, but the column selection reduces the size of the optimisation problem from 387 to N_s decision variables. With 0.78 and 0.22 ms, both methods are the cheapest in comparison. All median QP times stay below the sampling time of $T_s = 50$ ms. Figure 4b compares the median QP solve time with the tracking accuracy from Subsection 5.1. The FCM-Select-DeePC exhibits a similar tracking accuracy to the FCM-DeePC, but can reduce computation time by an order of magnitude. The ϕ-DeePC and the LPV-DeePC are the two methods with the highest tracking accuracy, but differ by a factor of nine in terms of computational load.

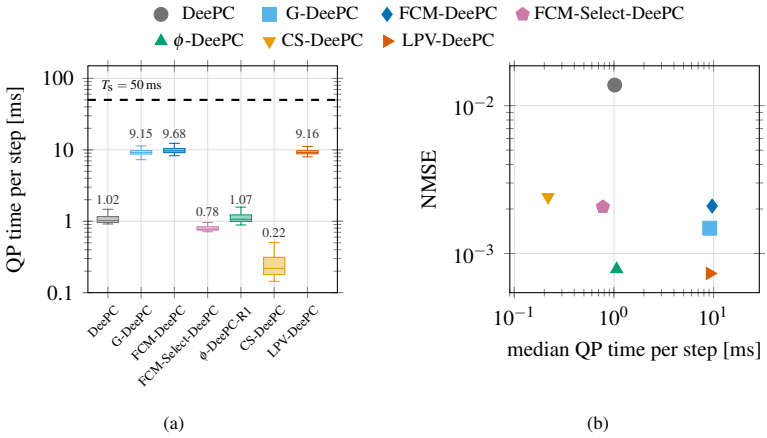


Figure 4: a.) QP time per step over all simulation steps. The whiskers cover 1.5 times the IQR. b.) trade-off between median QP time and tracking accuracy.

5.3 Robustness to Noise

To evaluate the robustness to measurement noise, every controller is simulated with 30 noise realisations at each of the noise levels $\sigma \in \{0.5\,\text{mm}, 2\,\text{mm and } 6\,\text{mm}\}$, where the noise corrupts both the excitation data and the measurements during control. The excitation signal and the configurations in Table 1 remain unchanged. Figure 5 shows the NMSE distributions. Here the grey band marks the noise level $\sigma = 2\,\text{mm}$ of the previous subsections. The distributions confirm the single-run comparison, where all nonlinear extensions outperform the standard DeePC in terms of the median NMSE at every noise level. The variance of the standard DeePC distributions is comparable to most of the extensions. The ranking within the top group depends on the noise level. LPV-DeePC achieves the lowest median NMSE at 0.5 and 2 mm, but its spread widens at 6 mm, where the damper scheduling channel has to reconstruct the velocity from the noisy position signal. There, ϕ-DeePC takes the lead and is the most robust method overall. CS-DeePC shows the opposite behaviour at 6 mm, with an upper quartile more than one order of magnitude above its median. The TS-variants remain in the midfield with moderate spread at all levels.

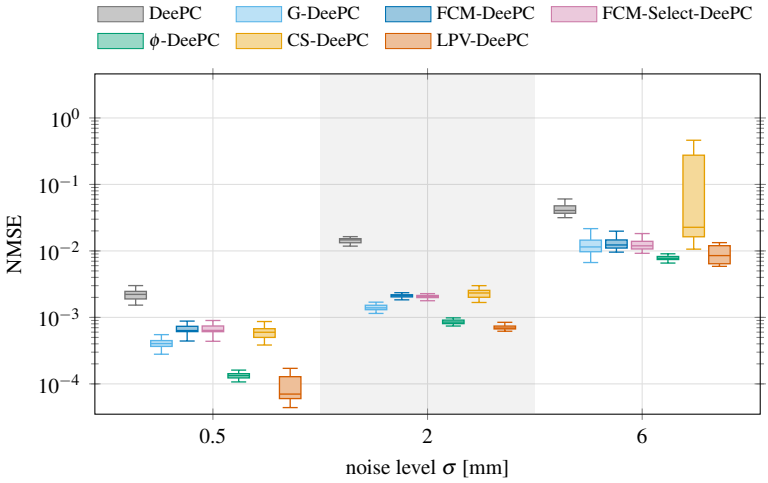


Figure 5: NMSE of the DeePC extensions for 30 noise realisations per noise level σ. The grey band marks the noise level of Subsection 5.1. The whiskers cover 1.5 times the IQR.

5.4 Sensitivity of Regularisation Parameters

The regularisation weight λ_g is the central tuning parameter for many data-driven control methods and all mentioned formulations in Section 3. To quantify its influence, λ_g is varied over eight orders of magnitude between 10^{-4} and 10^4, while the data set, the noise realisation and all remaining parameters of Table 1 are kept fixed. The additional dynamic regularisation for the ϕ-DeePC with $\lambda_{r1} = 10^4$ remains active. Figure 6 shows the resulting NMSE. The tuned values of Table 1 lie at or directly adjacent to the minimum of the respective curve, confirming that each method is compared at its own optimum. The valleys around the optima are narrow. Too small as well as too large values for λ_g result in a decreased tracking accuracy. This is observable for all methods. ϕ-DeePC is the exception, as its curve remains nearly flat down to a value of $\lambda_g = 10^{-4}$, since λ_{r1} takes over the regularising role of λ_g. Furthermore, the optima of the individual methods are located between $\lambda_g = 0.1$ (G-DeePC) and $\lambda_g = 10$ (DeePC). A regularisation weight shared by all methods would therefore distort the comparison, which justifies the method-wise tuning in Subsection 4.3.

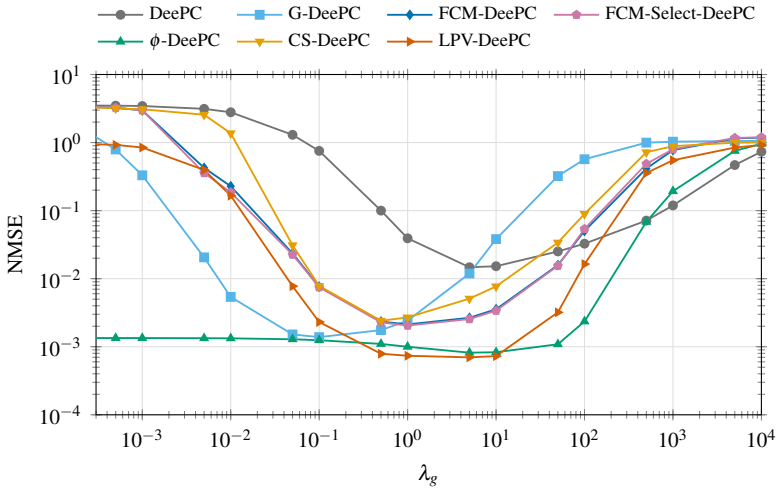


Figure 6: NMSE of the DeePC extensions as a function of the regularisation parameter λ_g.

5.5 Discussion

Table 3 summarises the findings of Section 5 in a qualitative evaluation. LPV-DeePC and ϕ-DeePC form the top group in terms of tracking accuracy, but obtain it in different ways. LPV - DeePC incorporates explicit process knowledge through a scheduling signal, which yields the best accuracy, but places the method in the computationally most expensive group. ϕ-DeePC achieves nearly the same accuracy without process knowledge, using generic radial basis functions. Together with its low computational load, high noise robustness (here the additional dynamic regularisation could also play a role) and insensitivity to λ_g, this yields the most balanced profile in this study. Note, however, that [5] suggests learning the RBF parameters offline using an RBF network. This step involves additional tuning effort, which was not explicitly covered in this study. The TS-variants form the midfield and differ in how the partition is obtained. On this benchmark the equidistant grid of G-DeePC outperforms the clustering of FCM-DeePC (NMSE: 1.48×10^{-3} vs. 2.10×10^{-3} and lower medians at all three noise levels). A possible explanation is that the G-DeePC uses more local nonparametric submodels in comparison to the FCM-DeePC ($N_r = 12$ vs. $N_r = 4$). For both TS methods the computational load is comparatively high because both the data matrices and the QP matrices are rebuilt at every sampling step (median solve times around 9 ms). In contrast, FCM-Select-DeePC, proposed in this paper, removes the computational burden. Truncating the data matrices to the N_s columns of highest membership shrinks the QP and the median solve time to 0.78 ms without degrading the tracking accuracy, changing the noise distribution significantly or shifting the optimum of λ_g. The proposed truncation method can be applied unchanged to G-DeePC. Furthermore, all TS-variants can incorporate process knowledge through the premise variables and the corresponding partitioned product space. CS-DeePC remains the controller with the lowest computational load and is favourable where computation time dominates, but it is the least noise-robust extension. At $\sigma = 6$ mm the tracking accuracy deteriorates in many of the simulation runs, as the wide IQR shows. A possible cause is that at this noise level the distance measure (11) over the short initialisation trajectory is dominated by noise, so the selected columns no longer represent the current operating point.

6 Conclusion and Outlook

This paper compared seven data-driven predictive control methods under identical conditions on a nonlinear oscillator benchmark. All nonlinear extensions of DeePC outperformed the standard DeePC on this process, especially in terms of tracking accuracy. Furthermore all extensions keep the optimisation problem a convex QP and differ solely in how the data-driven predictor is constructed. LPV-DeePC, who incorporates process knowledge, as well as ϕ-DeePC shows the best tracking accuracy. ϕ-DeePC additionally offers a low computation time and the best noise robustness in this study. The TS-variants show their strengths in noise robustness and incorporating a-priori process knowledge with G-DeePC outperforming the FCM-DeePC on this benchmark. Notably, the proposed FCM-Select-DeePC retained the accuracy and robustness of the FCM-DeePC at one twelfth of its computation time. The CS-DeePC is the controller with the lowest computation time but also most sensitive to high noise levels in this study.
Future work should include multi-input multi-output (MIMO) processes as well as different classes of nonlinear systems in the comparative study. In addition, the practical implementation of the methods and their performance on a laboratory test setup should be investigated in order to draw conclusions regarding their practical applicability. Finally, the TS approach should be continued with a theoretical stability analysis aiming at closed-loop guarantees.

Statement on the Use of AI Tools

During the preparation of this work, the authors used the AI coding assistant *Claude Code* (Anthropic) to assist with spelling, grammar, phrasing and with the generation of plots using *TikZ* and *Pgfplots*. The authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] J. Coulson, J. Lygeros and F. Dörfler. “Data-Enabled Predictive Control: In the Shallows of the DeePC”. 2019 European Control Conference (ECC), pp. 307-312. 2019.

[2] J. Maciejowski “Predictive Control with Constraints”. Prentice Hall, England. 2002.

[3] J. W. Polderman and J. C. Willems “Introduction to mathematical systems theory: a behavioral approach”. Springer-Verlag. 1997.

[4] J. C. Willems, P. Rapisarda, I. Markovsky and B. L. M. De Moor. “A note on persistence of excitation”. Systems & Control Letters, 54(4), pp. 312-329. 2005.

[5] M. Lazar “Basis-Functions Nonlinear Data-Enabled Predictive Control: Consistent and Computationally Efficient Formulations”. 2024 European Control Conference (ECC), pp. 888-893. 2024.

[6] J. Beerwerth and B. Alrifaee. “Less is more: contextual sampling for nonlinear data-enabled predictive control”. *arXiv:2503.23890*. 2025.

[7] J. Näf, K. Moffat, J. Eising and F. Dörfler. “Choose Wisely: Data-driven Predictive Control for Nonlinear Systems Using Online Data Selection”. *arXiv:2503.18845*. 2025.

[8] M. Zehner, A. Cavaterra and S. Lambeck. “Direkte datenbasierte prädiktive Regelung nichtlinearer Prozesse unter Verwendung von Takagi-Sugeno-Fuzzy-Systemen”. *Proc. 35. Workshop Computational Intelligence*, pp. 175-194. Berlin. 2025.

[9] C. Verhoek, J. Berberich, S. Haesaert, R. Tóth and H. S. Abbas. “A linear parameter-varying approach to data predictive control”. *IEEE Transactions on Automatic Control*, 71(4), pp. 2512-2523. 2026.

[10] J. Berberich and F. Allgöwer “An Overview of Systems-Theoretic Guarantees in Data-Driven Model Predictive Control”. *Annual Review of Control, Robotics, and Autonomous Systems* 8, pp. 77-100. 2025.

[11] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock and M. Diehl. “qpOASES: a parametric active-set algorithm for quadratic programming”. Mathematical Programming Computation, 6(4), pp. 327-363. 2014.

[12] O. Nelles, “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”. Springer. 2020.

[13] A. Kroll, “Computational Intelligence. Probleme, Methoden und technische Anwendungen.” De Gruyter Studium. 2016.

[14] A. Cavaterra, S. Lambeck and U. Schwalbe “Approximative Modellierung eines LLC-Resonanzwandlers mit Takagi-Sugeno-Modellen” at - Automatisierungstechnik, 71(10), pp. 853-866. De Gruyter Oldenbourg. 2023.

[15] R. Babuška “Fuzzy Modeling for Control”. Springer Science. 1998.

[16] H. Schulte, “Approximative Modellierung, Systemidentifikation und Reglerentwurf mittels gewichteter Kombination lokaler Zustandsraummodelle am Beispiel fluidischer Antriebe”. Kassel University Press. 2005.

Table 2: Tracking performance of the DeePC methods measured with the NMSE

method	NMSE $\times 10^3$
DeePC	13.78
ϕ-DeePC	0.78
CS-DeePC	2.41
G-DeePC	1.48
FCM-DeePC	2.10
FCM-Select-DeePC	2.07
LPV-DeePC	0.73

Table 3: Qualitative evaluation of the compared methods on this benchmark, from $(++)$ very favourable to $(--)$ very unfavourable with $(\circ)$ acceptable. Columns: tracking accuracy, computational load, noise robustness, sensitivity of the regularisation parameter λ_g, and the ability to incorporate a-priori process knowledge.

method	accuracy	comp. load	noise	reg. sens.	a-priori
DeePC	$-$	$+$	$\circ$	$\circ$	$--$
ϕ-DeePC	$++$	$+$	$++$	$+$	$-$
CS-DeePC	$\circ$	$++$	$-$	$\circ$	$--$
G-DeePC	$+$	$-$	$+$	$\circ$	$+$
FCM-DeePC	$\circ$	$-$	$+$	$\circ$	$+$
FCM-Select-DeePC	$\circ$	$++$	$+$	$\circ$	$+$
LPV-DeePC	$++$	$-$	$+$	$\circ$	$++$

On Interpretability and Scheduling Space Choice for System Identification with Dynamic Takagi–Sugeno Models

Konrad Lange[1], Ralf Mikut[2], Andreas Kroll[1]

[1]Measurement and Control Research Group, University of Kassel
Mönchebergstr. 7, 34125 Kassel, Germany
E-Mail: {konrad.lange, andreas.kroll}@mrt.uni-kassel.de

[2]Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT)
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
E-Mail: ralf.mikut@kit.edu

Abstract

Dynamic Takagi–Sugeno models are often motivated by their interpretability. However, the choice of the scheduling variables with respect to interpretability is comparatively little discussed for auto-regressive models. In contrast to many neural network and kernel method approaches, Takagi–Sugeno models can expose where in the operating space the behavior changes. The scheduling variables directly influence how the nonlinear behavior can be approximated by affine local models. Here, interpretability primarily means that the partitioning of the scheduling space follows physical expectation, as opposed to other measures such as interpretability of the local models.

This paper examines how the choice of scheduling variables affects the interpretability and accuracy of auto-regressive Takagi–Sugeno models with cluster-based membership functions. Both a synthetic and a real-world example show that interpretability and accuracy are not necessarily a trade-off under a limited budget of scheduling dimensions and local models.

1 Introduction

Takagi–Sugeno (TS) models [1] achieve the approximation of nonlinear functions and systems by weighting the output of simpler local models by fuzzy basis functions. Unlike piecewise-affine (PWA) models, TS models with continuously differentiable basis functions provide smooth transitions between the local models [2]. Compared with many neural network approaches, TS models have an internal structure that can allow for controller design and better interpretation. Furthermore, established and interpretable methods can be applied to construct the linear or affine local models. Moreover, because their number of parameters is moderate, TS models can often be identified from less training data.

In the recent machine learning literature, a growing focus lies on interpretability [3–6]. The choice of TS models is often explained by interpretability of the models. Fuzzy systems, such as Mamdani systems, originally used rule bases with inherent linguistic interpretability incorporating expert knowledge [7]. TS models introduced data-driven identification of the rule consequents [1]. Although interpretability of fuzzy systems has been of interest for a considerable time [8, 9], it remains an active research topic [10].

Whereas some works focus on *local* interpretability [11–13], there is a multitude of interpretability measures [8, 9, 14]. Among others, a smaller number of inputs, linguistic terms and rules can be considered better interpretable [9]. Interpretability issues of TS models identified from data have been addressed [9, 12]. Many papers address interpretability and accuracy in classification [9, 15] or (static) regression with TS models [16]. Specifically, a trade-off is for example mentioned in [15, 16] and examples that implement multi-objective optimization approaches on the Pareto front of accuracy and parsimony or interpretability include [17–20].

In data-driven identification of dynamic discrete-time TS models, the scheduling variables are often chosen to be identical to, or as a subset of, the regressor variables (examples include [21]). This choice is not necessarily effective or physically interpretable when the nonlinear behavior depends on functions of the regressors. An example is nonlinear friction, which may depend on the

velocity rather than simply on the position alone. In this work, interpretability therefore refers to whether the partitioning follows physical expectation.

This work aims to show that choosing the scheduling variables based on physical insight can be beneficial. If the system's behavior depends on an operating condition such as high and low velocity, the identification procedure should (explicitly) incorporate this physically meaningful quantity for partitioning. Interpretable membership functions distinguish TS models within the broader class of mixture-of-experts (MoE) approaches [22]. Those approaches can also use linear submodels, but typically schedule them with a less interpretable gating network.

The idea of using physically meaningful scheduling variables is by no means new. Examples from the literature call this *semi-mechanistic fuzzy modeling* [23, 24]. In [25], a TS state-space model of a servo-pneumatic actuator is derived from a nonlinear physical model, with premise variables chosen based on the actuator physics. This contribution investigates in more depth how the choice of the scheduling space affects both the accuracy and the interpretability of the resulting TS model.

Related work uses principal component analysis (PCA) in the scheduling space of linear parameter-varying (LPV) models [26] and dynamic TS models [27]. Hierarchical local model trees (HILOMOT) are used for nonlinear regression in [28], where the split direction is based on the curvature of the residuals. This makes nonlinear effects interpretable along the split directions.

The rest of the paper is structured as follows: Section 2 introduces the specific TS model class considered here. Section 3 contains the main investigation and considerations of this work regarding the scheduling space choice. Test systems are introduced in Section 4, whereas the results are presented and discussed in Section 4.

2 Dynamic Takagi–Sugeno Models

This work is concerned with discrete-time local affine Takagi–Sugeno models originally introduced in [1]. Here, multi-dimensional membership functions are

used [2]. In general, the prediction $\hat{y}$ of TS models is a sum of the c local model predictions $\hat{y}_i$ weighted by the fuzzy basis functions ϕ_i. For the single-input single-output (SISO) models considered here, the prediction is:

$$\hat{y}(k) = \sum_{i=1}^{c} \phi_i(\boldsymbol{z}(k), \boldsymbol{\theta}_{\mathrm{MF}}) \cdot \hat{y}_i(\boldsymbol{x}(k), \boldsymbol{\theta}_{\mathrm{LM},i}). \tag{1}$$

Here, the local models are auto-regressive with exogenous inputs (ARX) or output error (OE) models [29], resulting in a TS model of nonlinear ARX (NARX) or nonlinear OE (NOE) type. The TS-NARX model is optimized with respect to the one-step-ahead (OSA) prediction error, whereas the TS-NOE model is optimized with respect to the (recursive) simulation error.

In the case of TS-NOE models, the local model predictions are:

$$\begin{aligned} \hat{y}_i(\boldsymbol{x}(k), \boldsymbol{\theta}_{\mathrm{LM},i}) &= \boldsymbol{x}(k)^{\mathrm{T}} \cdot \boldsymbol{\theta}_{\mathrm{LM},i} \\ &= \begin{bmatrix} \hat{y}(k-1) & \cdots & \hat{y}(k-n) & u(k-\tau-1) & \cdots & u(k-\tau-m) & 1 \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} -a_{1,i} & \cdots & -a_{n,i} & b_{1,i} & \cdots & b_{m,i} & c_i \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{2}$$

The regressor vector $\boldsymbol{x}(k) \in \mathbb{R}^p$ includes $m \in \mathbb{N}$ time-delayed inputs and $n \in \mathbb{N}$ time-delayed predicted outputs, with the discrete dead time τ. In the case of local ARX models, delayed measurements $y(k-j)$ are used instead of the predicted outputs $\hat{y}(k-j)$. The local parameter vectors are combined in $\boldsymbol{\theta}_{\mathrm{LM}}^{\mathrm{T}} = [\boldsymbol{\theta}_{\mathrm{LM},1}^{\mathrm{T}} \cdots \boldsymbol{\theta}_{\mathrm{LM},c}^{\mathrm{T}}]$.

The fuzzy basis functions are cluster-based membership functions of fuzzy c-means (FCM) [30] type, with the fuzziness parameter $\nu > 1, \nu \in \mathbb{R}$:

$$\phi_i(\boldsymbol{z}(k), \boldsymbol{\theta}_{\mathrm{MF}}) = \mu_i(\boldsymbol{z}(k)) = \left[\sum_{j=1}^{c} \left(\frac{\|\boldsymbol{z}(k) - \boldsymbol{v}_i\|_2}{\|\boldsymbol{z}(k) - \boldsymbol{v}_j\|_2} \right)^{\frac{2}{\nu-1}} \right]^{-1}. \tag{3}$$

The variables $\boldsymbol{z}(k) \in \mathbb{R}^q$ used in determining the membership degree are called scheduling variables and span the scheduling space. The membership function parameters $\boldsymbol{\theta}_{\mathrm{MF}}^{\mathrm{T}} = [\boldsymbol{\theta}_{\mathrm{MF},1}^{\mathrm{T}} \cdots \boldsymbol{\theta}_{\mathrm{MF},c}^{\mathrm{T}}]$ are the locations of the c cluster centers $\boldsymbol{v}_j$

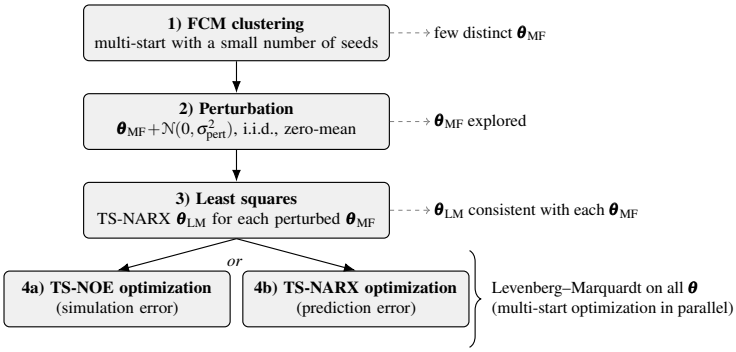


Figure 1: Overview of the multi-start optimization strategy to create initial parameter vectors through FCM clustering 1), perturbation of the cluster centers 2) and least-squares estimation of the local model parameters 3) for the subsequent gradient-based optimization 4a) or 4b).

in the scheduling space. The TS model parameter vector is therefore:

$$\boldsymbol{\theta}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{\theta}_{\mathrm{MF}}^{\mathrm{T}} & \boldsymbol{\theta}_{\mathrm{LM}}^{\mathrm{T}} \end{bmatrix}. \tag{4}$$

The complete parameter vector $\boldsymbol{\theta}$ of the TS-NARX and TS-NOE models is optimized with the gradient-based Levenberg–Marquardt algorithm [31]. To reduce the risk of converging to poor local minima, a multi-start (MS) strategy with initial values obtained from TS-NARX estimates is used [32] (Fig. 1). First, $\boldsymbol{\theta}_{\mathrm{MF}}$ is determined by FCM clustering with the Euclidean norm, using 8 random initializations of the clustering algorithm. To generate 128 MS initializations, the cluster centers are then perturbed by zero-mean, independent and identically distributed (i.i.d.) Gaussian noise with standard deviation σ_{pert}. For each perturbed $\boldsymbol{\theta}_{\mathrm{MF}}$, the local model parameters $\boldsymbol{\theta}_{\mathrm{LM}}$ are estimated globally by least squares. Starting from these initializations, all parameters $\boldsymbol{\theta}$ are optimized with respect to the prediction or simulation error. A separate validation dataset is used for early stopping and for selecting the best MS solution. The influence of the number of local models c for different choices of $\boldsymbol{z}$ is investigated in this work.

3 Scheduling Space Choice

3.1 Effect on Approximation of Nonlinearities

Similar to neural networks, TS models are often introduced as universal approximators. In practice, the theoretical conditions for universal approximation are usually not met, for example because the scheduling variables are only a subset of the regressors and the number of local models is limited. In general, universal approximation results [33, 34] only apply if all regressor variables $\boldsymbol{x}$ are part of the scheduling variables $\boldsymbol{z}$ or if the elements in $\boldsymbol{x}$ can be reconstructed from $\boldsymbol{z}$ (meaning there exists an injective mapping). This of course includes the case $\boldsymbol{z} = \boldsymbol{x}$.

For the sake of the following argument, the regressor vector defined in (2) is now split as

$$\boldsymbol{x}(k)^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{N}}(k)^{\mathrm{T}} & \boldsymbol{x}_{\mathrm{L}}(k)^{\mathrm{T}} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{x}_{\mathrm{N}}(k)$ is the subset of the regressors on which the system's output $y(k)$ depends nonlinearly, whereas the output only shows linear dependence on $\boldsymbol{x}_{\mathrm{L}}(k)$. With $c \to \infty$, an arbitrarily fine clustering in the space spanned by the selected scheduling variables

$$\boldsymbol{z}(k) = \boldsymbol{x}_{\mathrm{N}}(k) \tag{6}$$

enables universal approximation. In system identification practice, the regressors are often highly correlated and their full inclusion in the scheduling vector is not necessary [35].

Baseline results for well-known nonlinear benchmarks show that auto-regressive models often perform well with a relatively large number of lagged inputs and outputs [36]. In the study, the lag structures were selected based on the Akaike information criterion (AIC) evaluated with a linear ARX model and include a total of 17 to 23 regressors [36]. Using this number of scheduling dimensions would be neither effective nor interpretable. In applications, the number of local models is limited by computational resources. Moreover, the amount of training data required to avoid overfitting generally grows with model size [37].

This work shows that, instead of (6), it can be advantageous to select

$$\boldsymbol{z}(k) = \boldsymbol{W}\boldsymbol{x}_{\mathrm{N}}(k) \tag{7}$$

to span the scheduling space based on information about the (physical) system. This can reduce the order if no strong dependence on the individual components of $\boldsymbol{x}_{\mathrm{N}}(k)$ is observed. This can for example be the case when there are many regressors and/or the regressors are highly collinear. This approach could therefore reduce the curse of dimensionality that arises when a limited number of clusters is used in a high-dimensional clustering space. This benefit is conditional on the assumption that $\boldsymbol{W} \in \mathbb{R}^{q\times p}$ captures the directions along which the system is genuinely nonlinear. If this assumption is violated, reducing q introduces an irreducible error that does not vanish as $c \to \infty$. Even without dimension reduction, $\boldsymbol{W}$ can be beneficial if it aligns the cluster boundaries with the directions in which the dynamics change, so that fewer affine local models are needed. However, interpretability is only achieved when $\boldsymbol{W}$ has a small number of non-zero entries and is constructed using prior physical knowledge. Using a covariance matrix (or another data-driven estimate) to construct $\boldsymbol{W}$ does not automatically result in an interpretable scheduling space. Another way to improve interpretability is to merge compatible clusters [24]. However, this results in an accuracy–interpretability trade-off if many local models are needed to capture the direction of the nonlinear dependence.

3.2 Dependence on the Clustering Algorithm

Widely used clustering algorithms for dynamic TS models are fuzzy c-means (FCM) [30], Gustafson–Kessel (GK) [38], and Gath–Geva (GG) [39]. They differ mainly in their distance measure. FCM uses a fixed norm shared by all clusters. Usually this is either the Euclidean norm, which produces hyperspherical clusters, or the Mahalanobis norm, which produces hyperellipsoidal clusters of identical shape and orientation aligned with the principal axes of the data. GK instead adapts a cluster-specific norm based on the fuzzy covariance matrix, which allows arbitrarily oriented ellipsoidal clusters. GG additionally allows clusters of different volumes and densities.

The importance of an informed scheduling space depends on the clustering algorithm. It is particularly beneficial for FCM with the Euclidean distance norm, because this algorithm is sensitive to scaling. For example, because $y(k-1)$ and $y(k-2)$ are highly correlated, their difference has a small range, and FCM will not place clusters along this direction with the Euclidean distance norm. GK and GG are less affected because they can detect clusters elongated along hyperplanes [35]. However, both algorithms use more strategy parameters, are more likely to converge to local minima, and are more computationally expensive [40]. The ability to detect these clusters sacrifices axis-aligned interpretability and requires more parameters to be estimated. Furthermore, the clustering results are only used as initialization of the parameters $\boldsymbol{\theta}_{\mathrm{MF}}$. Regardless of which clustering algorithm is used to obtain this initialization, the subsequent gradient-based optimization uses FCM-type membership functions here, and the dimensions of the scheduling space remain unchanged during optimization for the considered models. Therefore, without also adopting more flexible membership functions with corresponding parameters, any additional information gained from a more flexible clustering algorithm such as GK or GG would be lost in the subsequent TS model optimization. Accordingly, only FCM is used here, consistent with the chosen FCM-type membership functions.

4 Test Systems for Case Studies

4.1 Simulated Test System

The first example is a well-known test system proposed by Narendra and Parthasarathy [41]:

$$y(k) = \frac{y(k-1)y(k-2)\big(y(k-1)+2.5\big)}{1+y(k-1)^2+y(k-2)^2} + u(k-1). \tag{8}$$

The nonlinearity depends on $y(k-1)$ and $y(k-2)$, whereas $u(k-1)$ enters linearly. The system is excited with an amplitude-modulated pseudo-random binary signal (APRBS) and the resulting dataset is displayed in Fig. 2. Since the true system is known and regressor selection is not the focus of this study,

the regressors are set to $\boldsymbol{x}(k) = [y(k-1), y(k-2), u(k-1)]^{\mathrm{T}}$. The fuzziness parameter $\nu = 1.3$ is chosen following [42]. The investigated choices for the scheduling space for the Narendra system are listed in Tab. 1.

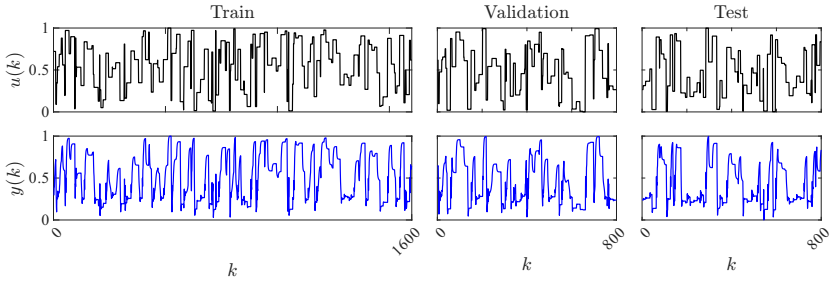


Figure 2: Scaled input and output signal of the Narendra system with the used APRBS excitation signal. The training, validation, and test datasets are shown.

4.2 Servo-Pneumatic Linear Drive

The servo-pneumatic linear drive (SPLD) and the dataset (Fig. 3) are described in detail in [32]. The input u is the voltage applied to the valve, while the position of the carriage is the measured output y. Both u and y are normalized to $[0, 1]$. Unlike the simulated system, the real-world data contains measurement noise and stochastic behavior caused by stick-slip friction. The system exhibits nonlinear behavior due to several sources, including:

- valve characteristics depending on u,
- compressibility of air and chamber thermodynamics depending on y,
- and the friction depending on y and $\dot{y}$.

Therefore, an informed choice of the scheduling space includes $u(k-\tau-1)$, $y(k-1)$ as well as $y(k-1)-y(k-2)$ as an approximation of the velocity $\dot{y}$ at time step $k-1$. The difference quotient $\big(y(k-1)-y(k-2)\big)/T_{\mathrm{s}}$ differs from the plain difference only by the constant $1/T_{\mathrm{s}}$, which is absorbed by the normalization of the $\boldsymbol{z}(k)$ to $[0, 1]$.

The TS model structure was taken from an investigation on the same dataset, where it was selected from the Pareto front of the validation MSE and the number of model parameters [32]. The regressors are $\boldsymbol{x}(k) = [y(k-1),\ldots,y(k-4), u(k-\tau-1),\ldots,u(k-\tau-6),1]^{\mathrm{T}}$, with $\tau = 2$, and the fuzziness parameter is chosen as $\nu = 1.3$.

The different choices for the scheduling space for the SPLD system are listed in Tab. 1. The reduced variants are chosen as minimal examples: one including only $[y(k-1), u(k-3)]$ and one additionally including $y(k-2)$. This allows evaluation of whether the additional information of $y(k-2)$ is beneficial. The input $u(k-3)$ is included in all scheduling space choices, as a dependence of the nonlinear behavior on the input cannot be ruled out for the uninformed variants, and is known to exist for the physically informed choice. Further variants could be considered, but the selection is kept concise to maintain the clarity of the study.

5 Results

5.1 Accuracy

Narendra System:

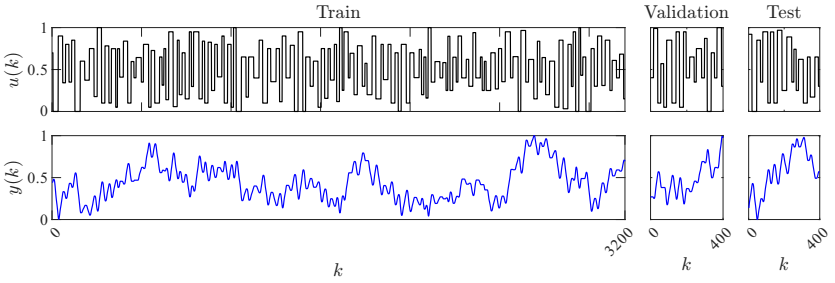


Figure 3: Scaled input and output signal of the servo-pneumatic linear drive (SPLD). The training, validation, and test datasets are shown.

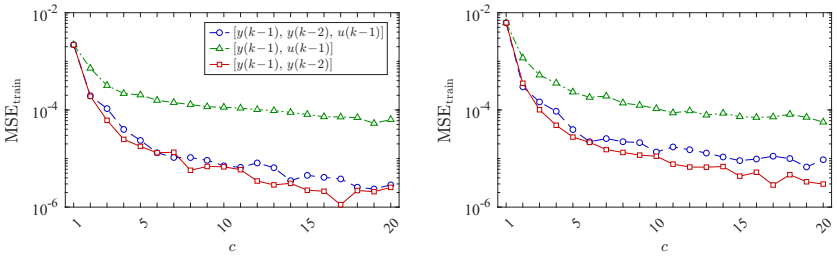
Variant	Scheduling space z
Narendra system	
full regressors	$[y(k-1), y(k-2), u(k-1)]$
reduced regressors	$[y(k-1), u(k-1)]$
informed	$[y(k-1), y(k-2)]$
Servo-pneumatic linear drive (SPLD)	
full regressors	$[y(k-1), \ldots, y(k-4), u(k-3), \ldots, u(k-8)]$
reduced regressors	$[y(k-1), y(k-2), u(k-3)]$
reduced regressors	$[y(k-1), u(k-3)]$
informed, without $y(k-1)$	$[y(k-1)-y(k-2), u(k-3)]$
informed	$[y(k-1), y(k-1)-y(k-2), u(k-3)]$

Table 1: Investigated scheduling space choices.

For the Narendra system, a linear model ($c = 1$) cannot adequately approximate the nonlinear behavior, and all TS models with $c > 1$ perform significantly better (Fig. 4). The OSA prediction and simulation errors for increasing c support choosing the scheduling space according to the known nonlinear dependencies of the system. Omitting $y(k-2)$ degrades the accuracy significantly, whereas omitting $u(k-1)$ improves it. This holds for all investigated numbers of local models. As expected for noise-free data, the TS-NOE simulation results are similar to the TS-NARX prediction results.

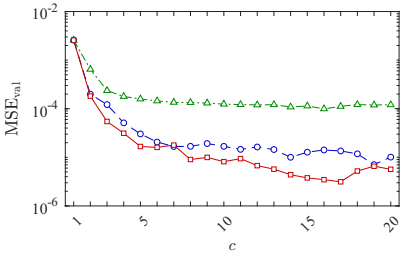
SPLD:

The results of the different scheduling space choices for the SPLD are shown in Fig. 5. For OSA prediction, the MSE is very small for all scheduling spaces, but the informed choice still performs best. Including only the difference $y(k-1) - y(k-2)$ and omitting $y(k-1)$ performs similarly, except on the validation dataset. Unlike the Narendra system, the TS-NOE simulation results differ considerably from the TS-NARX results. TS models with a moderate number of local models and a suitable $\boldsymbol{z}$ outperform the linear OE model, which again indicates nonlinear behavior. This is expected, as the TS-NARX models are trained and evaluated on the prediction error using measured past outputs, so their errors do not accumulate. The TS-NOE test error is therefore the

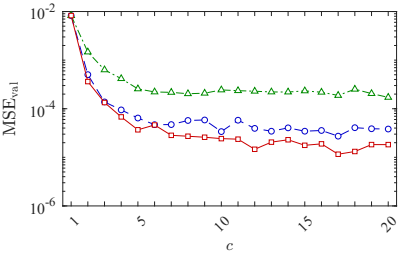


(a) TS-NARX, prediction MSE, training dataset.

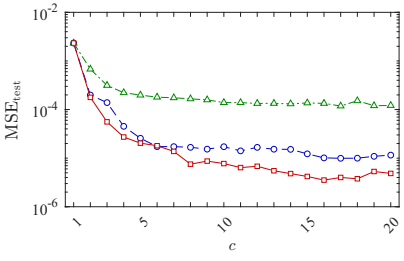
(b) TS-NOE, simulation MSE, training dataset.



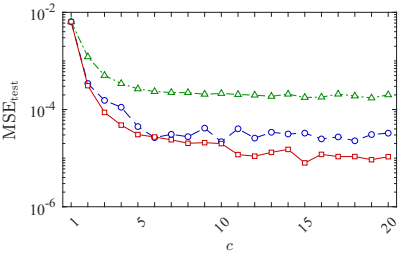
(c) TS-NARX, prediction MSE, validation dataset.



(d) TS-NOE, simulation MSE, validation dataset.



(e) TS-NARX, prediction MSE, test dataset.



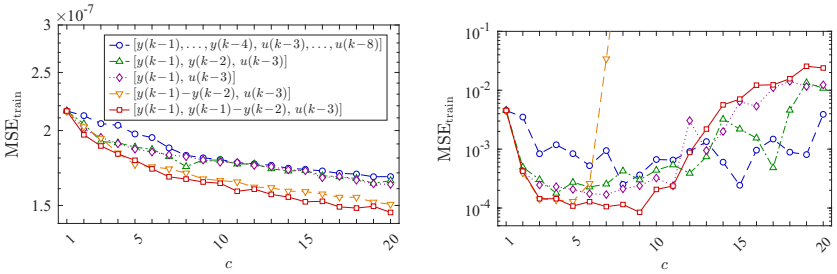
(f) TS-NOE, simulation MSE, test dataset.

Figure 4: Prediction and simulation error (Narendra system) depending on the choice of the scheduling space and the number of local models c. Left column: TS-NARX (prediction-error optimized), one-step-ahead prediction MSE. Right column: TS-NOE (simulation-error optimized), simulation MSE. Rows: training (top), validation (middle) and test (bottom) datasets. The informed choice of the scheduling space is shown in red.

more realistic indicator of model quality for practical deployment, since it reflects performance in a free-run simulation. Using all regressors as scheduling variables is generally not advisable. Without explicitly including $y(k-1) - y(k-2)$, the information of $y(k-2)$ is not beneficial to the model accuracy. For small c, the physically informed choice is clearly beneficial. In contrast to the simulated case study, the results of the MSE on the training and test datasets differ significantly and are less clear for increasing numbers of local models.

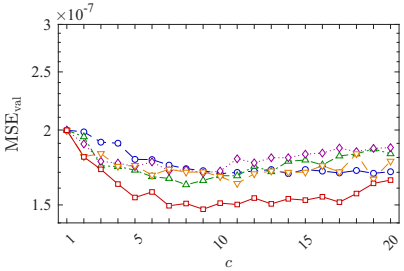
On the training data, including $y(k-1)-y(k-2)$ clearly improves performance for moderate c. In contrast, adding $y(k-2)$ itself to $[y(k-1), u(k-3)]$ has no significant effect. This supports the hypothesis that the informed choice improves accuracy by separating the nonlinear behavior along this direction. Up to $c = 8$ local models, the test results are mostly consistent with the training results. For higher numbers of local models, the results can only be interpreted as a rough trend. An explanation could be that the amount of training data per local model decreases as more local models are used. This also applies to the data available for the transitions between local models, which can lead to overfitting. The problem is further aggravated by the stochastic system behavior caused by the stick-slip effect.

With more local models, the cluster centers lie closer together and each cluster spans a smaller region of high membership degree. Since only predicted outputs are used in $\boldsymbol{z}(k)$, the scheduling variables deviate from their values based on measured outputs. An error of the same magnitude is therefore more likely to shift $\boldsymbol{z}(k)$ into the region of a neighboring cluster, activating a local model identified for a different operating region. This applies in particular to the difference $\hat{y}(k-1) - \hat{y}(k-2)$, which makes the scheduling more susceptible to errors. The scheduling variables $\boldsymbol{z}$ are normalized, and the lagged outputs $y(k-j)$ are highly correlated. Consequently, even small errors in the model outputs $\hat{y}(k)$ can lead to the activation of different local models. In simulation, this effect is amplified by error propagation. This explanation is supported by the results for scheduling with $[y(k-1)-y(k-2),\, u(k-3)]$, where the absolute output information is lost. For this choice, competitive results for small c are followed by a drastic increase in MSE.

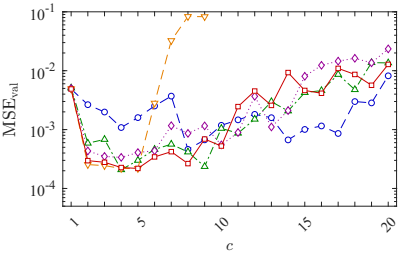


(a) TS-NARX, prediction MSE, training dataset.

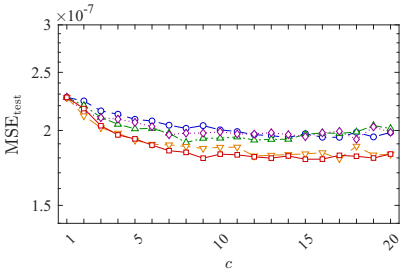
(b) TS-NOE, simulation MSE, training dataset.



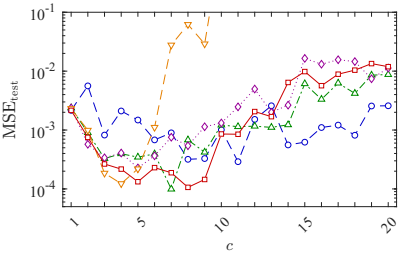
(c) TS-NARX, prediction MSE, validation dataset.



(d) TS-NOE, simulation MSE, validation dataset.



(e) TS-NARX, prediction MSE, test dataset.



(f) TS-NOE, simulation MSE, test dataset.

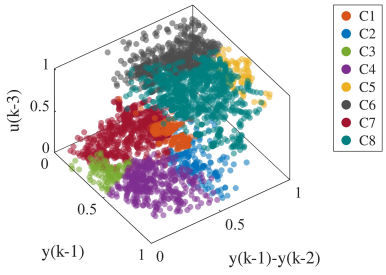
Figure 5: Prediction and simulation error (SPLD) depending on the choice of the scheduling space and the number of local models c. Left column: TS-NARX (prediction-error optimized), one-step-ahead prediction MSE. Right column: TS-NOE (simulation-error optimized), simulation MSE. Rows: training (top), validation (middle) and test (bottom) datasets. The physically informed choice of the scheduling space is shown in red.

Complete results for Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC), which weigh model accuracy against model complexity [35], are not reported here, as a meaningful comparison would also require varying the number of regressors used by each local model. Preliminary results on the SPLD training data, obtained by varying the scheduling variables and the number of local models while keeping the regressor variables fixed, indicate that the physically informed choice yields lower AIC and BIC values.

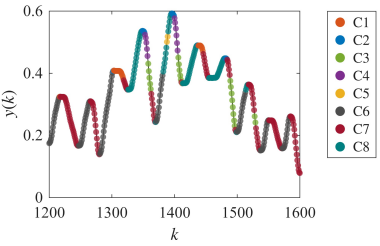
5.2 Interpretability

To assess interpretability, the membership degrees are visualized both in the scheduling space and over a section of the time series. Each sample is colored according to its highest membership degree. This loses interpretation of the transitions but improves visibility. For the uninformed choice (Fig. 6c), the data lies close to a diagonal because $y(k-1)$ and $y(k-2)$ are strongly correlated. The velocity enters the scheduling only implicitly, if at all. Using $y(k-1)-y(k-2)$ as a velocity proxy spreads the data more widely in the scheduling space, and the resulting partitioning persists after gradient-based optimization.

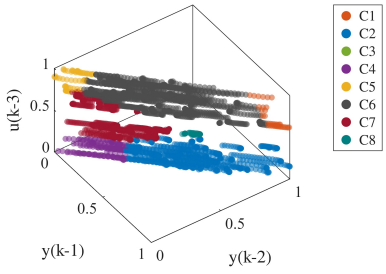
Because the difference makes changes in position explicit, the clusters partition the data along this direction. Fig. 6a depicts differences in clusters along the direction of the velocity (e.g., $C3$ and $C5$) that are not only due to correlation with the input. The clusters in Fig. 6a can be assigned to physical operating regions. For example, $C1$ corresponds to a slowly moving carriage near the center of the stroke with a small absolute input (as positive/negative u are normalized to $[0,1]$). In contrast, $C5$ represents fast motion with a large input, largely independent of the absolute position. The resulting difference is also visible when the fuzzy membership to the clusters is highlighted in the time series. Fig. 6b shows much clearer separation based on velocity than Fig. 6d and Fig. 6f. The apparent velocity-based differentiation in Fig. 6d and Fig. 6f is largely explained by the correlation between $u(k-3)$ and $y(k-1)-y(k-2)$ at most data points, rather than by explicitly separating different velocities.



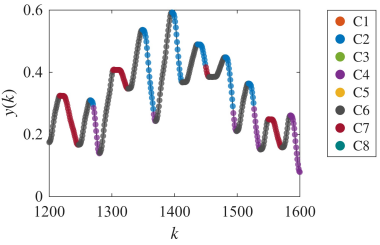
(a) Informed choice.



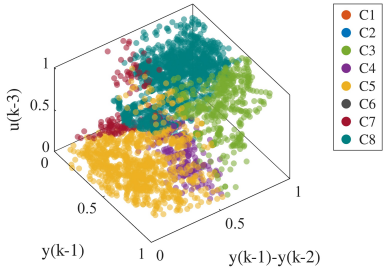
(b) Informed choice.



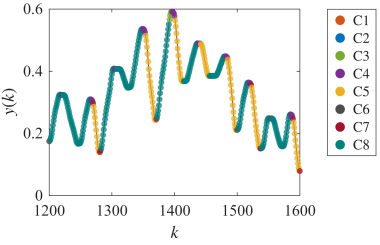
(c) Uninformed choice (reduced regressors).



(d) Uninformed choice (reduced regressors).



(e) Uninformed choice (all regressors used): membership projected to physically informed dimensions.



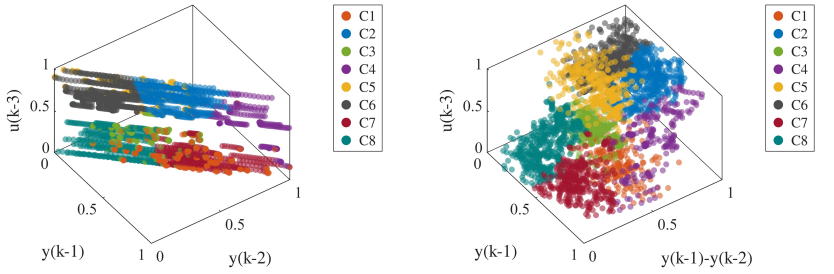
(f) Uninformed choice (all regressors used).

Figure 6: Comparison of the physically informed and uninformed choice of scheduling variables for the SPLD case study. Initial clustering is carried out with the FCM algorithm and the Euclidean distance norm after normalization of the input features. The membership shown is the result after gradient-based optimization for TS-NOE. Eight local affine models are used and the highest fuzzy membership degree to the clusters *Ci* is depicted by crisp colors. The results are displayed in the scheduling space (left) and over a part of the time series (right).

5.3 Comparison to Clustering with the Mahalanobis Norm

FCM clustering with the Euclidean norm is sensitive to scaling and restricted to hyperspherical clusters. In contrast, FCM clustering with the Mahalanobis norm can detect clusters that are aligned with the principal directions of the data and have a hyperellipsoidal shape [40]. This is achieved by calculating the covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{q \times q}$ of the data in the scheduling space. To preserve the cluster shape information determined from the data scattering during TS model optimization, the distance norm in (3) has to be changed to the Mahalanobis norm. The inverse $\mathbf{\Sigma}^{-1}$ of the covariance matrix is used as the form matrix. In the simplest case $\mathbf{\Sigma}$ is kept fixed, and the number of model parameters to be optimized with respect to prediction or simulation error remains the same as with the Euclidean norm. It increases slightly if a shared matrix $\mathbf{\Sigma}$ is optimized, but considerably if the shape of each cluster is optimized individually during gradient-based TS optimization. Preliminary results for the SPLD dataset using a fixed matrix $\mathbf{\Sigma}$ indicate a numerical problem when optimizing for the simulation error. The large eigenvalue of $\mathbf{\Sigma}^{-1}$ associated with the difference direction amplifies errors in $\hat{y}(k-1) - \hat{y}(k-2)$ direction. These errors then propagate recursively through the simulation. Strongly regularizing $\mathbf{\Sigma}$ mitigates this amplification. However, it also removes the separation into clusters along $y(k-1) - y(k-2)$. Figure 7 therefore shows the result after FCM clustering with the Mahalanobis norm (without a subsequent gradient-based TS optimization with respect to the model prediction error).

In the coordinates of the scheduling space (Fig. 7a), several clusters overlap at a similar position and input, e.g., $C1$ and $C7$ or $C3$ and $C8$. Projected onto the physically informed dimensions (Fig. 7b), the same clusters form regions that are clearly separated along $y(k-1) - y(k-2)$. This is consistent with Section 3: since $y(k-1)$ and $y(k-2)$ are highly correlated, one principal axis of $\mathbf{\Sigma}$ approximately corresponds to their difference, and the Mahalanobis norm amplifies this low-variance direction. Through clustering with the Mahalanobis norm, the scaling that the physically informed choice achieves through the normalization of $y(k-1) - y(k-2)$ is determined from data. However, two limitations remain apart from the described problem in simulation error optimization. First, the partitioning becomes interpretable only after projection onto the physically



(a) FCM clustering results (Mahalanobis norm), displayed in scheduling space.

(b) FCM clustering results (Mahalanobis norm) projected to physically informed dimensions.

Figure 7: Fuzzy membership degree using $[y(k-1), y(k-2), u(k-3)]$ as the scheduling space on the SPLD data. FCM clustering results with the Mahalanobis norm are shown without gradient-based TS-NOE optimization. The membership is shown in the scheduling space's coordinates (left) and projected onto the physically informed dimensions $[y(k-1), y(k-1)-y(k-2), u(k-3)]$ (right). The highest membership degree to each cluster Ci is depicted by crisp colors; 8 clusters are used.

informed dimensions. Second, the Mahalanobis norm weights the low-variance directions, regardless of whether the system is nonlinear along them.

5.4 Discussion

For the limited case studies with two systems, the physically informed and therefore more easily interpretable scheduling space can also result in better model accuracy. Interestingly, while the prediction and simulation errors decrease steadily for the simulated example without noise (Fig. 4), there is an increase for $c > 8$ local models for the SPLD example. Here, the more accurate models are more interpretable as well. For example, $c = 8$ fuzzy clusters in a 3-dimensional scheduling space can be interpreted visually (Fig. 6) because the dimensions allow for mapping to system behavior. Analyzing a 3-dimensional scheduling space also makes it possible to identify clusters with a small number of training data points. The steeper MSE increase with $c > 8$ for TS-NOE in Fig. 5 could be explained by the fact that errors of past predicted outputs propagate into the membership degrees.

The use of the Euclidean distance norm is closely related to the observed improvement. Possible improvements when using more geometrically flexible distance norms, such as the Mahalanobis norm, must be investigated with different membership functions to preserve the benefit during gradient-based TS-NOE optimization. Although the fixed fuzziness parameter $\nu = 1.3$ for all clusters does not appear to limit the model performance here, it reduces the generality of the observations. Other values of ν could be examined in future work. It is important to note that this contribution does not argue that including physical knowledge is necessary to achieve accuracy in data-driven system identification with the considered modeling approach. Other contributions, such as the work [43], show that the automatic approximation of the nonlinearity by tree-based local model networks can deliver competitive performance for benchmark problems.

6 Conclusion and Outlook

Interpretability and accuracy of dynamic cluster-based TS models depend heavily on an adequate choice of the scheduling space, and they do not always form a trade-off. With a limited number of scheduling dimensions and local models, it is crucial to choose physically interpretable features on which the output depends nonlinearly, so that the model can effectively approximate the nonlinear dynamic behavior. Results from two case studies support this. If the true nonlinearity is not fully captured by the chosen scheduling space, an irreducible bias is introduced that no increase in the number of local models can remove. This is unlike fully data-driven alternatives, which (in principle, given enough data and successful optimization) could still discover the missing direction when all regressors are used.

In future work, the investigations have to be extended to a broader range of test systems and benchmarks. It is also necessary to investigate gradient-based parameter optimization for simulation error with membership functions with flexible cluster shapes.

7 Acknowledgment

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 541311230.

References

[1] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.

[2] A. Kroll, “Identification of functional fuzzy models using multidimensional reference fuzzy sets,” *Fuzzy Sets and Systems*, vol. 80, no. 2, pp. 149–158, 1996.

[3] D. Materassi, S. Warnick, C. Rojas, M. Schoukens, and E. Cross, “Explaining complex systems: a tutorial on transparency and interpretability in machine learning models (Part I),” in *20th IFAC Symposium on System Identification (SYSID 2024)*, vol. 58, no. 15, 2024, pp. 492–496.

[4] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, “A Survey on Neural Network Interpretability,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, “A Survey of Methods for Explaining Black Box Models,” *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, 2018.

[6] K. Zhang, T. Shao, Y. Sun, X. Xu, X. Zhang, X. Zhou, K. Ding, and S. Huang, “Interpretable research of fuzzy methods: A literature survey,” *Information Fusion*, vol. 126, p. 103524, 2026.

[7] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[8] S. Guillaume, “Designing fuzzy inference systems from data: An interpretability-oriented review,” *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 426–443, 2001.

[9] R. Mikut, J. Jäkel, and L. Gröll, “Interpretability issues in data-based learning of fuzzy systems,” *Fuzzy Sets and Systems*, vol. 150, no. 2, pp. 179–197, 2005.

[10] L. Pickering, K. Cohen, and B. De Baets, “A Narrative Review on the Interpretability of Fuzzy Rule-Based Models from a Modern Interpretable Machine Learning Perspective,” *International Journal of Fuzzy Systems*, vol. 28, no. 4, pp. 1313–1332, 2026.

[11] J. Yen, L. Wang, and C. W. Gillespie, “Improving the interpretability of TSK fuzzy models by combining global learning and local learning,” *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 530–537, 1998.

[12] T. A. Johansen, R. Shorten, and R. Murray-Smith, “On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models,” *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 3, pp. 297–313, 2000.

[13] J. Abonyi and R. Babuska, “Local and global identification and interpretation of parameters in Takagi-Sugeno fuzzy models,” in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000*, vol. 2, 2000, pp. 835–840.

[14] M. J. Gacto, R. Alcalá, and F. Herrera, “Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures,” *Information Sciences*, vol. 181, no. 20, pp. 4340–4360, 2011.

[15] D. Nauck and R. Kruse, “How the learning of rule weights affects the interpretability of fuzzy systems,” in *1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence*, vol. 2, 1998, pp. 1235–1240.

[16] M. Setnes, R. Babuska, and H. B. Verbruggen, “Rule-based modeling: precision and transparency,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 1, pp. 165–169, 1998.

[17] T. A. Johansen and R. Babuska, “Multiobjective identification of Takagi-Sugeno fuzzy models,” *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 847–860, 2003.

[18] M. J. Gacto, R. Alcalá, and F. Herrera, “Integration of an Index to Preserve the Semantic Interpretability in the Multiobjective Evolutionary Rule Selection and Tuning of Linguistic Fuzzy Systems,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 515–531, 2010.

[19] C. Fuchs, U. Kaymak, and M. S. Nobile, “Building Interpretable and Parsimonious Fuzzy Models using a Multi-Objective Approach,” in *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2022, pp. 1–8.

[20] Q. Chang, Y. Chen, J. Liang, C. Liu, and J. Wang, “Constructing Concise Instance-Based Takagi–Sugeno–Kang Fuzzy Systems via Multiobjective Particle Swarm Optimization,” *IEEE Transactions on Fuzzy Systems*, vol. 33, no. 11, pp. 4111–4123, 2025.

[21] D. Aleksovski, D. Dovžan, S. Džeroski, and J. Kocijan, “A comparison of fuzzy identification methods on benchmark datasets,” in *4th IFAC Conference on Intelligent Control and Automation Sciences (ICONS 2016)*, vol. 49, no. 5, 2016, pp. 31–36.

[22] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive Mixtures of Local Experts,” *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[23] R. Babuska, *Fuzzy Modeling for Control*. Dordrecht: Springer, 1998.

[24] J. Abonyi and B. Feil, “Fuzzy Clustering for System Identification,” in *Cluster Analysis for Data Mining and System Identification*. Birkhäuser Basel, 2007, pp. 141–224.

[25] H. Schulte and H. Hahn, “Fuzzy state feedback gain scheduling control of servo-pneumatic actuators,” *Control Engineering Practice*, vol. 12, no. 5, pp. 639–650, 2004.

[26] A. Kwiatkowski and H. Werner, “PCA-Based Parameter Set Mappings for LPV Models With Fewer Parameters and Less Overbounding,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 781–788, 2008.

[27] D. Dovžan and I. Škrjanc, “Fuzzy Space Partitioning Based on Hyperplanes Defined by Eigenvectors for Takagi-Sugeno Fuzzy Model Identification,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5144–5153, 2020.

[28] M. Schöne, M. Kohlhase, and O. Nelles, “Incorporation of structural properties of the response surface into oblique model trees,” *at - Automatisierungstechnik*, vol. 73, no. 10, pp. 727–739, 2025.

[29] L. Ljung, *System identification: theory for the user*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[30] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY: Springer, 2006.

[32] K. Lange and A. Kroll, “Multi-Start Initialization for Simulation Error Optimization of Takagi–Sugeno Model Parameters,” in *30th International Conference on System Theory, Control and Computing (ICSTCC)*, 2026, accepted.

[33] B. Kosko, “Fuzzy systems as universal approximators,” *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1329–1333, 1994.

[34] H. Ying, “Sufficient conditions on uniform approximation of multivariate functions by general Takagi-Sugeno fuzzy systems with linear rule consequent,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 4, pp. 515–520, 1998.

[35] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*, 2nd ed. Cham: Springer, 2020.

[36] M. Champneys, G. I. Beintema, R. Tóth, M. Schoukens, and T. J. Rogers, "Baseline Results for Selected Nonlinear System Identification Benchmarks," in *20th IFAC Symposium on System Identification (SYSID 2024)*, vol. 58, no. 15, 2024, pp. 474–479.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[38] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, 1978, pp. 761–766.

[39] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773–780, 1989.

[40] A. Kroll, *Computational Intelligence: Probleme, Methoden und technische Anwendungen*, 2nd ed. Berlin: De Gruyter Oldenbourg, 2016.

[41] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[42] A. Kroll, "On choosing the fuzziness parameter for identifying TS models with multidimensional membership functions," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 1, no. 4, pp. 283–300, 2011.

[43] J. Belz, T. Münker, T. O. Heinz, G. Kampmann, and O. Nelles, "Automatic Modeling with Local Model Networks for Benchmark Processes," in *20th IFAC World Congress*, vol. 50, no. 1, 2017, pp. 470–475.

Joint Structure and Dynamics Optimization in Local Model Networks with Orthonormal Basis Functions

Timo Pütz, Oliver Nelles

Universität Siegen, Department Maschinenbau
Institut für Mechanik und Regelungstechnik – Mechatronik
Paul-Bonatz-Str. 9–11, 57068 Siegen, Germany
E-Mail: {timo.puetz, oliver.nelles}@uni-siegen.de

1 Introduction

Local model networks (LMNs) enable nonlinear system identification by combining local linear models, providing both approximation power and interpretability [3]. The local linear model tree (LOLIMOT) algorithm [6] is a widely used method for identifying an LMN's structure, i.e., the partitioning of the operating space into regions of local validity together with their local models.

Widely used approaches rely on ARX (autoregressive with exogenous input) local models [6]. This is computationally convenient but has well-known drawbacks: equation error estimates are biased under output noise, the local order must be fixed a priori or found by expensive subset selection, and stability of the resulting nonlinear ARX model is hard to establish due to output feedback.

Orthonormal basis function (OBF) models – Laguerre [8], Kautz [9], and their generalization to arbitrary pole sequences (GOBF) [2] – avoid these issues, since they are output error models and typically need far fewer coefficients than a comparable finite impulse response (FIR) expansion, provided the poles are

close to the true dynamics. Their price is a nonconvex, nonlinear dependence on the pole location(s).

Combining OBF local models with LOLIMOT was initially proposed as a sequential procedure [6]: an ARX-LMN is trained first, poles are extracted from its local models, and the ARX regressors are then replaced by OBF filters with fixed poles, while only the linear coefficients are re-estimated. Although this removes the ARX-specific drawbacks from the final models, the partition is determined before the OBF poles are considered.

In this contribution, we instead optimize the OBF poles *nonlinearly and directly within* the LOLIMOT structure search. To our knowledge, this direct coupling is new. For the noise-free case, we show why this coupling matters: the partition favored by an equation error (ARX) criterion is not necessarily the one an output error (OBF) criterion would choose, since it may be influenced by parts of the local dynamics that no OBF filter, regardless of pole choice, can represent. Under noisy conditions, we investigate the relevance of this mechanism empirically. Section 2 reviews the necessary background, Section 3 develops the argument, and Sections 4–4 describe the compared variants and experimental study.

2 Theoretical Background

2.1 Local Model Networks

Let $u(k)$ and $y(k)$ denote the input and output of a time-discrete dynamic system at sampling instant k. The input-output behaviour of a nonlinear dynamic system of order m can be approximated by a static nonlinear function $f(\cdot)$ acting on an appropriate regressor:

$$\underline{x}(k) = \left[u(k-1)\cdots u(k-m)\; y(k-1)\cdots y(k-m)\right]^{\mathsf{T}} \tag{1}$$

where the output is given by $y(k) = f(\underline{x}(k))$ [7]. An LMN approximates $f(\cdot)$ by a set of M local models (LMs), each valid within a bounded region of an input

space. Following [7], the network output is computed as

$$\hat{y}(k) = \sum_{j=1}^{M} \mathrm{LM}_j\left(\underline{x}(k), \underline{\theta}_j\right) \cdot \Phi_j\left(\underline{z}(k)\right), \tag{2}$$

where $\underline{\theta}_j$ collects the parameters of the j-th LM. Commonly, LMNs use a local ARX model for LM_j, i.e., an affine function of the regressor (1), $\mathrm{LM}_j\left(\underline{x}(k), \underline{\theta}_j\right) = \underline{\theta}_j^{\mathsf{T}} \underline{x}(k) + \theta_{0,j}$; Section 2.3 introduces an alternative choice for LM_j. The input space $\underline{x}(k)$ of the local models (the *x-space*) and the input space $\underline{z}(k)$ of the validity functions Φ_j (the *z-space*) need not coincide [3, 5]. The *x-space* determines the regressor used by each local dynamic model. The *z-space* determines which local model is responsible for a given operating point. As validity functions, normalized Gaussians centered at $\underline{c}_j$ with width $\underline{\sigma}_j$ are commonly used, forming a partition of unity for every $\underline{z}$. The choice of the *z-space* strongly influences model quality and smoothness [3], so we distinguish *x-space* and *z-space* explicitly throughout.

2.2 Structure and Parameter Identification via LOLIMOT

The partition of the *z-space* into M regions is determined by the LOLIMOT algorithm [6, 7]. It is a heuristic, incremental tree-construction procedure that partitions the *z-space* into axis-orthogonal hyper-rectangles. Starting from a single region covering the entire *z-space*, it iterates:

1. Identify the worst-performing local model, i.e., the region contributing the largest share of the global loss.
2. For every input dimension of the *z-space*, split this region's hyperrectangle in half, place a candidate membership function in each half, construct all validity functions, and estimate the local models of the two regions affected by the split.
3. Commit the split that yields the largest reduction of the global loss; discard the remaining candidates.

For every split candidate, the local parameters $\underline{\theta}_j$ are obtained by locally weighted least squares,

$$\hat{\underline{\theta}}_j = \left(\underline{X}^{\mathsf{T}} \underline{Q}_j \underline{X}\right)^{-1} \underline{X}^{\mathsf{T}} \underline{Q}_j \underline{y}, \tag{3}$$

where $\underline{X}$ stacks the regressor $\underline{x}(k)^{\mathsf{T}}$ over all samples, and $\underline{Q}_j$ weights each sample by its validity for region j [7]. Because (3) is a closed-form estimator, evaluating a split only requires solving a few weighted least-squares problems, not a full nonlinear optimization.

2.3 Orthonormal Basis Functions as Local Models

Instead of the raw, delayed input and output samples of (1), the input $u(k)$ can be passed through a bank of n orthonormal, stable filters $F_i(q)$, $i = 1, \ldots, n$, where q is the forward shift operator. Here, orthonormality means that the filters' impulse responses $f_i(k)$ satisfy $\sum_{k=0}^{\infty} f_i(k)\, f_j(k) = \delta_{ij}$. As with any local model in an LMN [7], an offset $\theta_{0,j}$ is included, so the local model output is

$$\hat{y}_j(k) = \theta_{0,j} + \sum_{i=1}^{n} \theta_{i,j}\, \varphi_i(k), \qquad \varphi_i(k) = F_i(q)\, u(k). \tag{4}$$

As (4) is again linear in $\underline{\theta}_j$, the local parameters are still obtained by (3), now with the filtered regressor $\underline{x}(k) = [\varphi_1(k), \cdots, \varphi_n(k)]^{\mathsf{T}}$.

Such orthonormal filter banks are usually built as a cascade of first-order all-pass sections with poles $\underline{\eta} = \{\xi_1, \ldots, \xi_n\}$ inside the unit circle [2] (Figure 1 shows such a cascade for an alternating sequence of two real poles). Three special cases are relevant here. The Laguerre model [8] uses a single pole of multiplicity n. The Kautz model [9] uses a single complex-conjugate pole pair of multiplicity n, well suited to lightly damped, oscillatory dynamics. The GOBF model [2] allows an arbitrary, possibly mixed sequence of distinct real and complex-conjugate poles; Laguerre and Kautz are its special cases. In every case, the basis depends on the pole(s) $\underline{\eta}$ *nonlinearly*, but remains *linear* in $\underline{\theta}_j$ for any fixed $\underline{\eta}$.

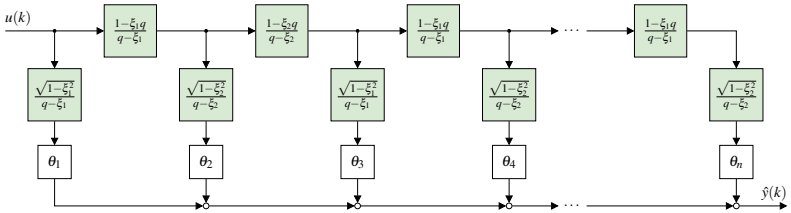


Figure 1: GOBF filter cascade realizing the orthonormal basis functions $\varphi_i(k)$ for an alternating sequence of two real poles ξ_1, ξ_2: first-order all-pass sections (top) propagate $u(k)$ along the chain, first-order tap filters (below) generate $\varphi_i(k)$, which are weighted by θ_i and summed to $\hat{y}(k)$.

This separable structure makes pole optimization within the LOLIMOT structure search computationally efficient. Optimizing $\underline{\eta}$ jointly with $\underline{\theta}_j$ as one nonlinear least-squares problem would add many free parameters and increase the risk of poor local minima. Instead, for any fixed $\underline{\eta}$ the optimal $\underline{\theta}_j$ is already given in closed form by (3), so only $\underline{\eta}$ needs to be refined by gradient-based search, with $\underline{\theta}_j$ solved from (3) at each step and detached from the computational graph before differentiating with respect to $\underline{\eta}$. By separable nonlinear least squares (*variable projection*) [1,4], this detached gradient is exact: the dependence of the optimal linear parameters on $\underline{\eta}$ does not contribute to the gradient, as stated by the envelope theorem. The detached formulation is therefore computationally cheaper than differentiating through the closed-form solve, while yielding the same gradient. The search consequently reduces to a low-dimensional optimization over $\underline{\eta}$ alone.

3 Why Coupling Matters

As shown at the end of Section 2.3, pole optimization can be efficiently coupled to the LOLIMOT structure search. Here, we explain why this coupling matters: a partition obtained by minimizing the local ARX equation error is not necessarily suitable for a subsequent replacement by an OBF model.

As an alternative, in [6], an ARX-LMN is first trained with LOLIMOT until the desired complexity is reached. The local ARX coefficients are then extracted, and each local model is replaced by a GOBF filter constructed from these

coefficients, while the partition itself remains unchanged. This procedure is fundamentally decoupled: the partition is fixed before the OBF representation or its poles are considered.

As a running example, consider a second-order process with input-dependent coefficients,

$$y(k) = -a_1(u(k))\,y(k-1) - a_2(u(k))\,y(k-2) + b_1\,u(k-1) + b_2\,u(k-2), \quad (5)$$

with

$$a_j(u) = a_{j,0} + a_{j,1}\,u, \quad j = 1, 2. \quad (6)$$

Around the center $\bar{u}_i$ of region i, with $\delta u(k) = u(k) - \bar{u}_i$ and $\bar{a}_{j,i} := a_j(\bar{u}_i)$, the model can be rewritten exactly as

$$y(k) = -\bar{a}_{1,i}\,y(k-1) - \bar{a}_{2,i}\,y(k-2) + b_1\,u(k-1) + b_2\,u(k-2) + w_i(k), \quad (7)$$

with

$$w_i(k) := -\delta u(k)\,(a_{1,1}\,y(k-1) + a_{2,1}\,y(k-2)). \quad (8)$$

The term $w_i(k)$ describes the local modeling error caused by the variation of u within region i. For a local ARX model, this error directly enters the equation error residual. Since $|\delta u(k)|$ is bounded by the half-width of the region, refining the partition reduces the equation error criterion. Equation error identification therefore favors splitting along $u(k)$.

Now consider replacing the local ARX model by a GOBF filter constructed from the coefficients $(\bar{a}_{1,i}, \bar{a}_{2,i})$, i.e., from the poles of $\bar{D}_i(z) = z^2 + \bar{a}_{1,i}\,z + \bar{a}_{2,i}$. Any local model of the form (4) consists of an offset and a dynamic response to u alone,

$$\hat{y}_i(k) = \theta_{0,i} + G_i(q)\,u(k). \quad (9)$$

Thus, $\hat{y}_i(k)$ depends on $u(k), u(k-1), \ldots$ regardless of whether u has recently been inside or outside region i.

Writing the true local dynamics (7) in the z-domain, where $Y(z), U(z), W_i(z)$ denote the z-transforms of $y(k), u(k), w_i(k)$, gives

$$Y(z) = \underbrace{\frac{N(z)}{\bar{D}_i(z)} U(z)}_{\text{representable by } G_i(q)} + \underbrace{\frac{z^2}{\bar{D}_i(z)} W_i(z)}_{\text{not representable}}, \qquad N(z) = b_1 z + b_2. \tag{10}$$

The first term is exactly the part that a GOBF filter with the poles of $\bar{D}_i(z)$ can reproduce. The second term, $\frac{z^2}{\bar{D}_i(z)} W_i(z)$, is the response of the same local dynamics to $w_i(k)$ instead of $u(k)$. According to (7), $w_i(k)$ is directly linked to the true output history. Since any local model of the form (4) consists only of a constant and an LTI response to u and therefore has no dependence on the actual y, no choice of OBF order, pole, or offset can produce a term that matches it. A local ARX model does not have this limitation: its regressor directly contains $y(k-1), y(k-2)$, so it can absorb $w_i(k)$ by adjusting its own coefficients away from $\bar{a}_{1,i}, \bar{a}_{2,i}$.

ARX- and GOBF-based partitioning therefore pursue different goals. Finer splits along $u(k)$ reduce $w_i(k)$ for both local model types, but only the ARX model can directly use this effect: its regressor allows it to absorb the smaller $w_i(k)$ by adjusting its own coefficients away from $\bar{a}_{1,i}, \bar{a}_{2,i}$, so its equation error loss benefits twice. A GOBF model only gets the first of these benefits.

There is a further disadvantage: since $G_i(q)$ is not reset, the error actually captured by the loss of region i, $\frac{z^2}{\bar{D}_i(z)} W_i(z)$, is an accumulation of $w_i(k)$ over the entire past. Time points at which u was far from $\bar{u}_i$, and therefore $w_i(k)$ was large, have a particularly strong effect. The error therefore does not depend only on the size of $w_i(k)$ while $u(k)$ is currently inside region i. An ARX-based structure search, which never evaluates $G_i(q)u(k)$, cannot see this accumulation and therefore cannot select splits that reduce it. Only forming the partition directly under the local OBF output error criterion can account for this effect. If the partition from an ARX-optimized structure search is used, the GOBF model is therefore tied to splits whose benefit it can only partly exploit, while the mechanism to which it is most sensitive remains unaddressed.

Additive output noise reinforces this asymmetry. In equation error (ARX) estimation, output noise enters the regressors through past outputs. This

typically biases the estimated poles toward the origin. The bias becomes stronger as the partition is refined, whereas output error GOBF estimation is structurally much less affected.

4 Variants Compared and Experimental Setup

This contribution compares the following variants. All share the same locally weighted estimator (3) for $\underline{\theta}_j$. They differ only in how the local pole(s) are obtained and coupled to the LOLIMOT structure search.

ARX-GOBF Decoupled. An ARX-LMN is trained first. Its partition and region-dependent poles are then frozen, and only the linear coefficients are re-estimated for the GOBF replacement.

ARX-GOBF Coupled. A GOBF local model is used throughout the structure search. For each candidate region, an ARX fit provides a pole *proposal*, which is refined by gradient-based optimization (Section 2.3). After each split, child regions repeat this procedure on their respective data. Figure 2 contrasts this single-phase procedure with the two-phase Decoupled variant.

Laguerre and Kautz Coupled. A Laguerre or Kautz local model is used throughout the structure search. The pole parameters are refined by gradient-based optimization.

ARX (diagnostic). The unmodified ARX-LMN of the decoupled variant, without OBF replacement. It is included to illustrate the well-known gap between equation error prediction and output error accuracy in ARX models.

All gradient-based pole refinements (Section 2.3) use the Adam optimizer for 50 steps with a learning rate of 10^{-2}. ARX-GOBF Coupled initializes its poles from the ARX proposal (Section 4); Laguerre and Kautz Coupled have no such proposal and instead start from a fixed initial pole $\xi_0 = 0.5$ (Laguerre) or a complex-conjugate pair with initial radius $r_0 = 0.5$ and angle $\omega_0 = 0.8$ (Kautz).

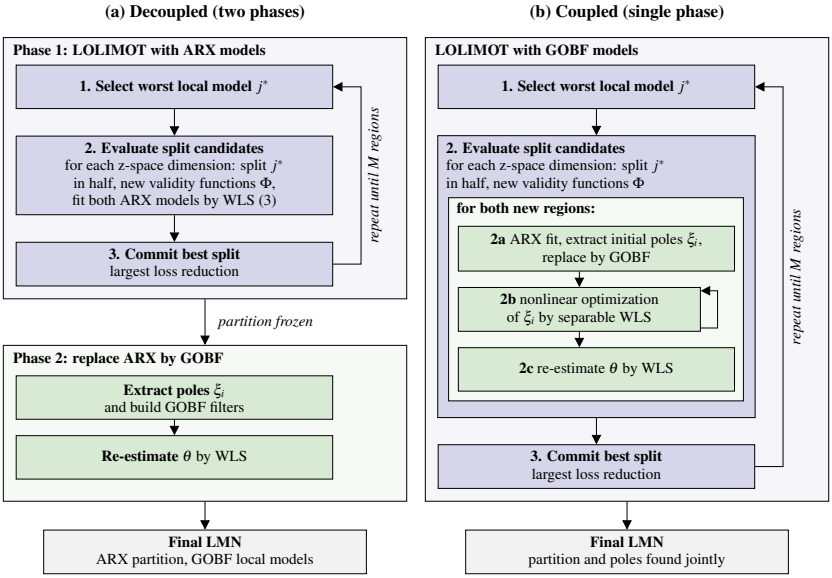


Figure 2: LOLIMOT structure search for ARX-GOBF Decoupled (a: two phases, ARX structure search then GOBF replacement) and ARX-GOBF Coupled (b: single phase, poles proposed by an ARX fit and refined by nonlinear optimization within the search).

4.1 Example Process and Experimental Design

The example process is

$$\begin{aligned} y(k) = &\big(1.5 + 0.1\,u(k)\big)\,y(k-1) - \big(0.7 + 0.2\,u(k)\big)\,y(k-2) \\ &+ 0.133\,u(k-1) - 0.0667\,u(k-2). \end{aligned} \tag{11}$$

This instantiates the general input-dependent-coefficient structure of (5) introduced in Section 3, on which the coupling argument was developed. The parameters are chosen so that the operating range spans both a complex-conjugate pole pair and two real poles. It is excited by an amplitude-modulated pseudo-random binary signal (APRBS) with hold time $T = 25$ and input range $u \in [-1.5, 0.5]$. The local ARX models used throughout this study (ARX-GOBF Decoupled, the ARX pole proposal in ARX-GOBF Coupled, and the ARX diagnostic) are second-order and with direct feedthrough, i.e.,

Table 1: Experimental design of the main study.

Parameter	Value
Number of local models M	5 (fixed)
Training / test length	10 000 / 5 000 samples
GOBF / Laguerre / Kautz order	2, 4, 6, 8
z-space	full ARX regressor vs. $u(k)$ alone
Independent APRBS realizations	10 per configuration

$\underline{x}(k) = [u(k), u(k-1), u(k-2), y(k-1), y(k-2)]^\mathsf{T}$, matching (1) with $m = 2$ extended by $u(k)$. The full ARX-regressor *z-space* referred to throughout coincides with this regressor, $\underline{z}(k) = [u(k), u(k-1), u(k-2), y(k-1), y(k-2)]^\mathsf{T}$. Table 1 summarizes the experimental design.

Every configuration (order $\times$ *z-space*) is repeated over 10 independently generated training/test realizations, giving $4 \times 2 \times 10 = 80$ runs per variant. We evaluate both one-step-ahead prediction error (true y fed back, equation error) and output error (only u is used as input, with the nonlinear model output recursively fed back) on the test set. We study this design in two settings: first noise-free (Sections 4–5.2), isolating the mechanism of Section 3, and then under additive output noise on the training data (Section 5.3).

5 Results

5.1 Overall Comparison

Figure 3 shows the full distribution (over 10 realizations, order 4, full ARX-regressor *z-space*) of prediction error and output error.

ARX-GOBF Decoupled is clearly the worst in output error (median 0.174). All three structure-coupled variants reach medians of 0.072–0.082, a factor of roughly 2.1–2.4 lower. The raw ARX diagnostic shows a large gap between

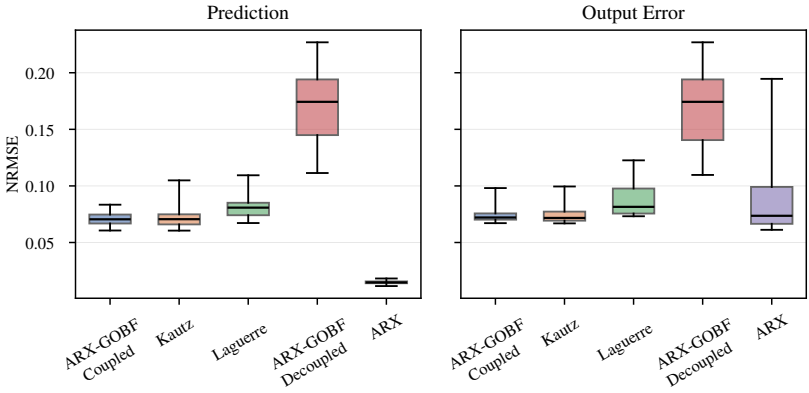


Figure 3: Prediction Error and Output Error by variant, noise-free (all OBFs: order 4, full ARX-regressor *z-space*, 10 realizations).

prediction error (median 0.015) and output error (median 0.074), consistent with Section 3: the true y-feedback in prediction hides the actual output error accuracy. Its output error distribution is markedly right-skewed. In contrast ARX-GOBF Decoupled shows almost no prediction/output error gap, because the replaced local models ignore y entirely by construction. But its own distribution is wide for both metrics, a first sign that a partition formed independently of the OBF criterion is less robust. The complete output error results, broken down by basis order and *z-space*, are given in Table 2 in the following section.

5.2 Effect of Order, z-Space, and Partition Structure

Table 2 shows both effects at once. All three structure-coupled variants improve with increasing basis order. Laguerre improves most, from output error 0.118 to 0.075, while Kautz and ARX-GOBF Coupled saturate from order 4 onward. ARX-GOBF Decoupled shows only a small improvement with order (0.180 to 0.166). This matches Section 3: higher order only enriches the input-driven response that a GOBF filter can already represent. It does not address the disturbance term that Section 3 identifies. ARX-GOBF Decoupled also has

Table 2: Output error (mean over 10 realizations) by OBF order and *z-space*.

Order, z-space	ARX-GOBF Coupled	Kautz	Laguerre	ARX-GOBF Decoupled
2, full ARX regressor	0.079	0.078	0.118	0.180
2, $u(k)$ alone	0.140	0.141	0.165	0.193
4, full ARX regressor	0.075	0.076	0.088	0.169
4, $u(k)$ alone	0.139	0.140	0.145	0.186
6, full ARX regressor	0.074	0.074	0.076	0.167
6, $u(k)$ alone	0.140	0.140	0.141	0.184
8, full ARX regressor	0.074	0.073	0.075	0.166
8, $u(k)$ alone	0.140	0.140	0.141	0.183

by far the largest spread across realizations (standard deviation ≈ 0.04, versus ≈ 0.01 for the other variants).

Restricting the *z-space* to $u(k)$ alone (second row of each order block in Table 2) costs the structure-coupled variants an order of magnitude more output error accuracy (+0.047 to +0.067 NRMSE) than it costs ARX-GOBF Decoupled (+0.016) or raw ARX (+0.001). The freedom to schedule on lagged input and output information, not only on $u(k)$, is exactly the lever that helps the coupled variants, while ARX-GOBF Decoupled already prefers splitting in $u(k)$ instead. Table 3 confirms this: ARX-based structure search places 78 % of its splits in $u(k)$, against only 10–14 % for the OBF-based variants. Given the freedom to choose, OBF-based structure search picks a systematically different and empirically better partition.

This preference is not absolute, but it strengthens with more training data. A dedicated noise-free sweep with much larger training sets (n_{train} from 20,000 up to 200,000 samples, ARX/ARX-GOBF Decoupled only) shows the $u(k)$-fraction converging monotonically toward 100 %: 0.825, 0.925, 0.950, 0.975, and 1.000 – every single split – at $n_{\text{train}} = 200{,}000$. This confirms that the residual, non-$u(k)$ splits observed at moderate training-set sizes are a finite-

Table 3: Fraction of LOLIMOT splits placed in $u(k)$ (full ARX-regressor *z-space*).

Variant	Fraction of splits on $u(k)$
ARX / ARX-GOBF Decoupled	0.775
ARX-GOBF Coupled	0.144
Laguerre	0.144
Kautz	0.100

sample artifact. With few samples left in a narrow region, an non-$u(k)$ split can reduce the local loss estimate even though $u(k)$ remains the asymptotically preferred choice.

5.3 Noise Study

The noise-free study isolates the mechanism of Section 3 from a second, noise-driven mechanism. To quantify the latter, we repeat the same design (order 4, both *z-spaces*, 10 realizations) with additive output noise on the training data only, swept over SNR $\in \{30, 20, 10, 5, 3, 2, 1\}$ dB. The test set stays noise-free.

Figure 4 shows that all variants degrade as the noise level increases, but the gap between ARX-GOBF Decoupled and the best coupled variant narrows only to a positive value of about 0.03 NRMSE, reached at around 5 dB, rather than vanishing.

A second finding concerns the partition itself. ARX's $u(k)$-split preference collapses abruptly between the noise-free case and 30 dB (78 % to 25 %) and then stays near chance down to 1 dB – much faster than the NRMSE gap closes, so split *selection* loses discriminative power well before pole *estimates* degrade. At mild noise, more training data shifts this preference back toward $u(k)$, as in the noise-free case, but an SNR-specific residual that additional samples alone do not resolve remains. This is an open question so far. ARX and the OBF-based variants keep splitting differently at mild to moderate noise (30–20 dB),

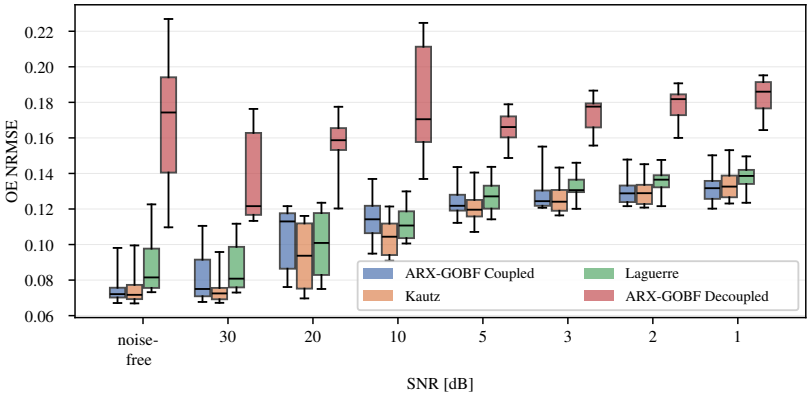


Figure 4: Output Error vs. SNR (all OBFs: order 4, full ARX-regressor *z-space*, 10 realizations per box).

Table 4: Mean complex pole-estimation distance, noise-free vs. pooled over all SNR levels.

Variant	Noise-free	With noise	Factor
ARX / ARX-GOBF Decoupled	0.065	0.492	7.5×
ARX-GOBF Coupled	0.139	0.462	3.3×
Kautz	0.152	0.187	1.2×
Laguerre	0.289	0.314	1.1×

converging below 10 dB to a common pattern dominated by $u(k-2)$ (72–89 %). Since the NRMSE gap persists even after partition choice converges, at least part of it must reflect a mechanism other than partitioning.

Table 4 reports the mean complex pole-estimation distance between estimated and true poles for the noise-free and noise case. Kautz and Laguerre, each restricted to a single fixed pole family, can never match the true poles exactly, since the process alternates between real poles and a complex-conjugate pair. Hence they have nonzero distance already without noise. This distance barely changes under noise (1.1–1.2×), unlike ARX's, which degrades by 7.5×.

ARX-GOBF Coupled fits this picture less cleanly. Its pole distance already exceeds ARX's in the noise-free case (0.139 vs. 0.065), because it does not split purely on $u(k)$, leaving wider regions in u (and hence a larger disturbance term $w_i(k)$) than ARX. Under noise, ARX's own $u(k)$-split preference also collapses, leaving it with similarly wide regions, and its pole distance converges close to Coupled's (0.492 vs. 0.462). So wider regions and noise-driven attenuation bias likely both contribute, without the data separating them. Yet Coupled's output error accuracy remains better than Decoupled's even where pole distances and partitions have converged, so the coupled variants' persisting advantage under strong noise is not fully explained by either mechanism, and remains an open question.

6 Conclusion

This contribution shows that nonlinear OBF pole optimization should be coupled *directly* to the LOLIMOT structure search. If the partition is instead formed independently of the OBF criterion, as in the decoupled variant, it is optimized for reducing the local equation error mismatch, which is not the same as reducing the local output error mismatch a GOBF local model actually faces. A partition chosen for equation error alone does not target this limitation, and no amount of later pole or order refinement can fix it. On a representative second-order process with input-dependent poles, letting the structure search proceed directly under the OBF criterion reduces output error by a factor of roughly 2.1–2.4 relative to the decoupled variant. The coupled variants retain an advantage even under noise, but it is not yet clear to what extent partitioning and pole estimation alone account for it, and this remains open. Local ARX information nonetheless remains a useful pole initializer. Confirming these findings on further processes, and extending the comparison to other scheduling-variable (z-)spaces – including OBF-based ones – are important next steps, with noise sensitivity remaining a key aspect to track throughout.

References

[1] G. H. Golub, V. Pereyra, “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate”. In: *SIAM J. Numer. Anal.* 10.2, pp. 413–432. 1973.

[2] P. S. C. Heuberger, P. M. J. Van den Hof, B. Wahlberg (Eds.), “Modelling and Identification with Rational Orthogonal Basis Functions”. London, U.K.: Springer. 2005.

[3] C. Illg, T. Kösters, O. Nelles, “On the choice of the scheduling variables for dynamic local model networks with local regularized FIR models”. 2024.

[4] L. Kaufman, “A variable projection method for solving separable nonlinear least squares problems”. In: *BIT Numer. Math.* 15.1, pp. 49–57. 1975.

[5] T. Kösters, O. Nelles, “Nonlinear dynamic system identification of separation unit for raw materials”. In: *Proc. IEEE Conf. Control Technology and Applications (CCTA)*, pp. 992–997. 2023.

[6] O. Nelles, “Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT)”. In: *Proc. 11th IFAC Symp. System Identification*, Kitakyushu, Japan, pp. 639–642. 1997.

[7] O. Nelles, “Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy and Gaussian Process Models”. 2nd ed., Berlin, Germany: Springer. 2020.

[8] B. Wahlberg, “System identification using Laguerre models”. In: *IEEE Trans. Autom. Control* 36.5, pp. 551–562. 1991.

[9] B. Wahlberg, “System identification using Kautz models”. In: *IEEE Trans. Autom. Control* 39.6, pp. 1276–1282. 1994.

An Interpretable LLM-Assisted Framework for Passenger Safety Incident Analysis and Mixed Reality Scenario Development in Automated Shuttle

Daryna Sytnychenko, Torsten Bertram

TU Dortmund
Institute of Control Theory and Systems Engineering
Technische Universität Dortmund
Otto-Hahn-Str. 8, 44227 Dortmund
E-Mail: {daryna.sytnychenko, torsten.bertram}@tu-dortmund.de

1 Introduction

Automated public transport (APT), particularly automated buses (ABs) and automated shuttles (ASs), introduces challenges that extend beyond conventional transportation safety to include both the safety and security of passengers inside the vehicle. Safety primarily concerns protection against unintentional harm arising from operational or technical hazards, whereas security concerns protection against intentional threats and potentially harmful passenger behavior [1].

The absence of a human driver and the public nature of the transport environment create new requirements for managing incidents and emergencies. Ensuring that passengers feel safe and secure, while enabling APT systems to respond to emergencies at least as effectively as conventional buses with human drivers, is essential for fostering public acceptance of APT [2]. Although existing research has proposed safety mechanisms for specific hazards, a comprehensive approach addressing the wide range of incidents that may occur in APT remains lacking [3].

The development and evaluation of appropriate safety and security mechanisms requires realistic incident scenarios that can be used to investigate human-machine interaction (HMI) between automated vehicles, passengers, and other road users [4]. This, in turn, requires a systematic understanding of the incidents that may occur in and around ABs and ASs. However, existing datasets and taxonomies provide only limited coverage of passenger-related incidents in public transport. Existing classifications often focus on specific event types, such as vehicle collisions, without comprehensively considering incidents occurring both inside and outside the vehicle [5]. This is particularly relevant for APT, where incidents may originate from the surrounding environment as well as from within the vehicle. Such incidents may influence passengers' perceived safety and fear of crime, potentially affecting their acceptance of public transport [6, 7].

To address the identified gap, this paper investigates the following research questions:

- **RQ1:** How can real-world incidents relevant to passenger safety as well as security be systematically collected and structured into a comprehensive incident database?
- **RQ2:** Which clustering configuration most effectively reveals meaningful groups of bus-related incidents with similar characteristics?
- **RQ3:** What feature relationships characterize the identified incident clusters based on association rule mining (ARM)?
- **RQ4:** Can the identified patterns serve as an interpretable basis for generating representative incident scenarios for future HMI development and evaluation?

By addressing these questions, the proposed approach aims to establish a systematic data-driven basis for analyzing bus-related incidents and supporting the development of safety and security mechanisms for APT.

The remainder of this paper is structured as follows. Section 2 provides a brief overview of existing taxonomies of bus-related incidents, data mining methods, and related approaches. Section 3 describes the proposed framework

for passenger safety and security incident analysis, including the creation of the incident database, clustering, and cluster characterization. The framework is evaluated in Section 4. Finally, Section 5 summarizes the work presented in this paper and discusses the main findings and directions for future work.

2 State of the Art

Existing research shows that even harmless unexpected operational events, such as unplanned stops without injuries, raise questions regarding the protection of passengers and other road users without direct driver intervention [3]. A systematic characterization of potential incidents can therefore provide a basis for developing appropriate responses and safety mechanisms by identifying the range of possible incidents, their contributing factors, the circumstances under which they occur, and their potential consequences.

Incidents involving buses have been classified from different perspectives in the existing literature. The predominant focus lies on collision-related incidents, while other types of events occurring in or around buses receive comparatively less attention [8]. One type of classifications distinguishes between incidents occurring during crashes and non-crash incidents [9]. Other studies specifically address non-collision incidents and the associated casualties; however, these classifications generally address particular incident categories rather than providing a comprehensive overview of the possible causes and circumstances of such events [10].

To address these aspects, structured incident datasets can be analyzed using data mining methods to identify recurring patterns and relationships within the collected information. Various data mining techniques have been applied in transportation and traffic incident research [11]. Of particular relevance to this work are unsupervised clustering and ARM, which can be combined to first identify groups of similar incidents without predefined classes and subsequently characterize the patterns and relationships associated with these groups.

A closely related approach was presented by Esenturk et al. [11], who used clustering followed by ARM to derive patterns from the UK STATS19 road-

traffic accident dataset [12]. Their analysis focuses on road-traffic data covering different vehicle types and uses COOLCAT for clustering. The selection of COOLCAT was motivated by the categorical nature of the dataset and its entropy-based clustering principle. However, the approach is specifically designed for categorical data, whereas incident data collected for the present study contain a combination of numerical (e.g., number of injured people), categorical (e.g., incident type), and Boolean features (e.g., whether a driver was affected).

Clustering provides a means of organizing incidents into groups with similar characteristics and can reveal differences in the distribution of individual features between clusters. However, clustering primarily describes the similarity structure of the data and does not necessarily provide an interpretable characterization of the specific feature combinations that distinguish individual clusters. ARM can therefore be used as a complementary approach to identify recurring combinations of feature values that characterize individual clusters [13]. In this context, a combination such as

$$(\textit{Driver Status} = \textit{intoxicated}) \wedge (\textit{Weather} = \textit{fog}) \Rightarrow (\textit{Incident Type} = \textit{collision})$$

indicates an association between these feature values and the corresponding incident type, rather than a causal relationship.

Apriori [14] is one of the established algorithms for ARM. It identifies frequent itemsets by iteratively generating and evaluating candidate itemsets, which can result in considerable computational and memory requirements for large or complex datasets [15]. The algorithm has been applied to transportation safety research, including the analysis of traffic accidents on urban roads [16], the investigation of factors associated with major traffic accidents [17], and the identification of factors related to bus-crash severity [18]. In particular, Samerei et al. [18] combined clustering and association rule discovery to investigate factors affecting the severity of bus-involved crashes, demonstrating the potential of this combination for transportation safety analysis.

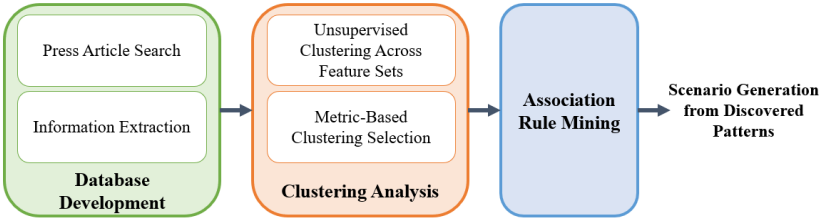


Figure 1: Proposed framework for the systematic analysis and characterization of bus-related incidents.

Another popular algorithm for ARM is Frequent Pattern Growth (FP-Growth) [19] which provides an alternative to Apriori by using an FP-tree structure to represent frequent patterns and avoiding the explicit generation of candidate itemsets. This can reduce the number of database scans and improve computational efficiency, particularly for datasets with a large number of frequent patterns [20]. Comparative studies have also reported advantages of FP-Growth in terms of computational or memory efficiency under specific datasets and parameter settings [15]. Compared with Apriori, FP-Growth has been less frequently applied in transportation safety research, although recent studies demonstrate its potential. For example, Wang et al. [22] applied FP-Growth to investigate relationships between traffic accidents and traffic violations.

Both Apriori and FP-Growth have also been combined with other analytical methods to investigate interactions between multiple risk factors contributing to severe traffic accidents [22].

Overall, the reviewed studies demonstrate the applicability of clustering and ARM to transportation incident analysis. However, existing approaches are predominantly based on conventional traffic-accident datasets and focus primarily on crash-related events.

3 Methodology

The proposed framework for identifying patterns in bus-related incidents and supporting the development of passenger safety mechanisms is divided into three main steps: database development (Section 3.1), clustering analysis (Section

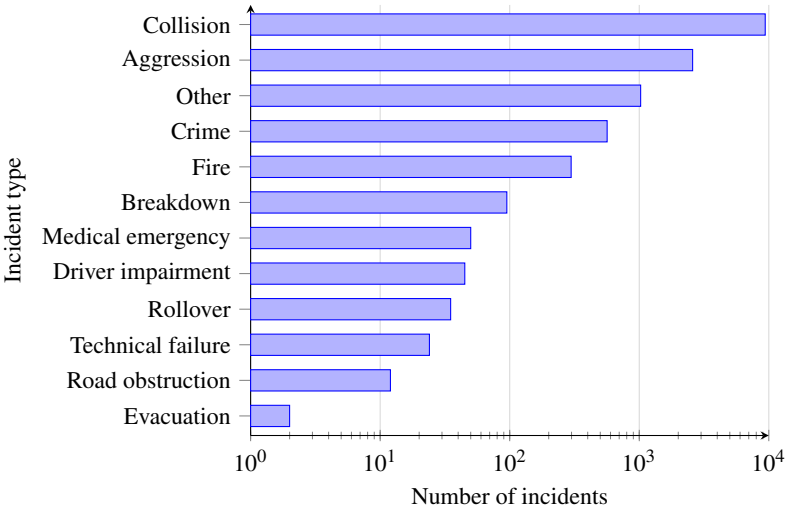


Figure 2: Distribution of incidents by *Incident Type*.

3.2), and ARM (Section 3.3). An overview of the framework is provided in Figure 1.

3.1 Database Development

The first part of the framework focuses on *creating a database of incidents*. For this purpose, articles published on Presseportal Deutschland [23] over a ten-year period, from 10 July 2016 to 10 July 2026, were collected and analysed. Presseportal was selected as the primary data source because police reports provide detailed information on the circumstances and characteristics of incidents and cover a broad range of event types. In contrast to hospital or health insurance records and official statistical datasets, which primarily focus on specific aspects such as injuries, healthcare outcomes, or traffic statistics, police reports can provide information on both traffic-related and non-traffic-related incidents [24]. In total, 1,144,063 articles were collected, of which 14,157 were identified as reporting bus-related incidents.

The database creation process consists of two main stages. First, the collected articles were analyzed to identify bus-related incidents and to extract the relevant text. Second, predefined features were extracted from the identified incident descriptions.

Several approaches were considered for the automated extraction and processing of article content from Presseportal. The first approach consisted of a C#-based implementation relying on predefined patterns in the structure of the web pages and articles. However, this approach was not suitable for processing the complete dataset due to the variable structure and free-form nature of the published content. In addition, individual articles may describe multiple incidents which further complicates rule-based extraction.

Therefore, a more adaptable approach based on Large Language Model (LLM) was employed, whereas the outline for possible recreation or extension of the creating of dataset was created with C#. The Ollama framework (version 0.34.0) [25] was used to run the Qwen3-14B model for automated information extraction from the article content. This approach was selected based on its reported high performance for automated information extraction processes [26].

As no comparable dataset was identified in the existing literature, the definition and selection of the features were based on two main considerations: the requirements of the framework for generating representative incident scenarios and the characteristics identified in existing taxonomies of collision and non-collision incidents. Since the extraction of information from the articles into the database was performed using an LLM, categorical features were defined as enumerations wherever possible. This approach was used to reduce duplicate entries and inconsistencies caused by different terms describing similar concepts, such as *"hit"* and *"strike"*.

The created dataset was stored using SQLite, a lightweight database management system that provides the required SQL functionality [27]. For each identified incident, a set of predefined features was extracted, resulting in 41 features in total. The complete set of features and their descriptions is provided in Table 3.

One of the most important extracted features is *Incident Type*, as it characterizes the nature of the incident (Figure 2). The resulting dataset shows that more than one third of the identified incidents were not caused by a collision. This distribution demonstrates that 33% of incidents cannot be addressed solely through collision-related safety mechanisms and therefore motivates the consideration of a broader range of incident types in passenger safety systems.

Another noteworthy feature is *Outcome*. Defined as an enumeration, this feature describes the consequences of an incident and can subsequently be used to assess its severity and associated risk. For example, an incident resulting only in property damage may indicate a lower level of risk, whereas an incident involving passenger injury may indicate a higher level of risk. The corresponding risk level is stored in the *Risk* feature.

A correlation analysis was conducted to identify potentially redundant features. No feature pairs exhibited an absolute Pearson correlation coefficient of $|r| \geq 0.90$, indicating that no highly correlated pairs were identified according to the predefined threshold. Nevertheless, the feature space contained a substantial number of low-frequency variables, with several attributes being present in less than 1% of incidents. For example, low-frequency variables were particularly common among attributes describing road conditions and lighting. Due to their low occurrence, these variables were removed from the respective feature sets prior to the clustering analyses.

3.2 Clustering Analysis

The resulting correlation analysis indicated the need to preprocess the dataset before applying the clustering algorithms. Further investigation revealed that several features contain a high proportion of *"unknown"* values, with some exceeding 70%, such as *Driver Status*. In addition, the features exhibit different distributions and characteristics. Furthermore, some features are more important for scenario description than others. To account for these inconsistencies and the high dimensionality of the dataset, the clustering algorithms were applied to three predefined feature sets, as presented in Tables 5, 6, and 7. The first feature set comprises Primary Incident and Driver Characteristics

Table 1: Overview of the chosen clustering algorithms

Method	Supported data types	Clustering principle	Key characteristic
K-means [28]	Numerical	Centroid-based	Mean-based cluster representation
K-medoids [29]	Numerical, categorical, ordinal	Dissimilarity-based	Medoid-based cluster representation
K-prototypes [30]	Numerical, categorical	Prototype-based	Combines means and data types
Hierarchical clustering [31]	Numerical, categorical, ordinal	Hierarchical, dissimilarity-based	Produces a dendrogram
HDBSCAN [34]	Numerical	Density-based	Detects clusters and noise
COOLCAT [32]	Categorical	Entropy-based	Designed for categorical data

(PIDC), the second represents Incident Causes and Consequences (ICC), and the third provides an Extended Incident Characteristics (EIC) representation incorporating a substantially broader range of incident-related features. Feature Sets PIDC and ICC capture distinct groups of incident characteristics, whereas EIC combines almost all available features, excluding those with less than 3% non-*unknown* observations across the dataset.

To identify potential hidden clustering patterns, unsupervised clustering algorithms were selected. The selection of the algorithms was based on the characteristics of the database, including mixed data types (numerical, categorical, and binary), the potential presence of outliers, and uneven feature completeness. Based on the literature review, the selected algorithms are presented in Table 1.

K-means was selected as a reference algorithm due to its computational efficiency and relatively simple implementation. Since the algorithm supports only numerical data, categorical and boolean features were transformed before

applying the algorithm. Categorical features were converted using one-hot encoding, while the numerical features were normalized to ensure comparable feature scales. The Euclidean distance used by K-means is defined as

$$d(x,y) = \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2} \tag{1}$$

where x and y represent different observations in the dataset and p denotes the total number of features.

K-medoids is a modification of the K-means algorithm in which the cluster centers are represented by actual observations from the dataset, referred to as *medoids*, rather than by the mean of the observations. The clusters are subsequently formed around these representative observations. To better handle mixed-type data, Gower distance was used:

$$d_{\text{Gower}}(x,y) = \frac{1}{p} \sum_{j=1}^{p} d_j(x_j, y_j) \tag{2}$$

The feature-level distance $d_j(x_j, y_j)$ is defined as

$$d_j(x_j, y_j) = \begin{cases} \dfrac{|x_j - y_j|}{R_j}, & \text{if feature } j \text{ is numerical,} \\ \mathbb{I}(x_j \neq y_j), & \text{if feature } j \text{ is categorical.} \end{cases} \tag{3}$$

where x represents one observation, y represents a medoid or another observation, and R_j denotes the range of numerical feature j. For categorical features, the distance is set to 1 when the categorical values differ and to 0 when they are identical.

K-prototypes was selected as another extension of K-means for handling mixed-type data. In addition to the number of clusters, the algorithm uses γ as a weighting parameter to balance the influence of numerical and categorical features on the distance:

$$d(x,q) = \sum_{j \in \mathcal{N}} (x_j - q_j)^2 + \gamma \sum_{j \in \mathcal{C}} \mathbb{I}(x_j \neq q_j) \tag{4}$$

where x denotes an observation and q denotes the prototype of a cluster. A prototype is a representative of a cluster consisting of the mean values of its numerical features and the most frequent values of its categorical features [30]. Furthermore, $\mathcal{N}$ denotes the set of numerical features and $\mathcal{C}$ the set of categorical features.

After selecting algorithms from the K-means family, hierarchical clustering was also considered. Hierarchical clustering can overcome some limitations of flat or partition-based algorithms, such as requiring the number of clusters k to be specified in advance [33]. An agglomerative hierarchical clustering approach with complete linkage and a Gower distance matrix was applied. Starting from individual observations, the closest clusters are successively merged according to the complete-linkage criterion, resulting in a hierarchical structure represented by a dendrogram. The dendrogram can subsequently be used to obtain different cluster partitions without rerunning the clustering algorithm. However, hierarchical clustering can be sensitive to outliers, which is particularly relevant for the created dataset due to its heterogeneous nature.

For these reasons, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) was also investigated. In contrast to partition-based and hierarchical clustering approaches, HDBSCAN is a density-based method that explicitly identifies noise and does not require the number of clusters to be specified in advance [34]. The algorithm constructs a hierarchy of density-based clusters and extracts a final clustering based on cluster stability.

The final selected algorithm, COOLCAT, was included because it was specifically developed for clustering categorical data and has previously been applied to an incident dataset in the road-traffic domain [11]. Since COOLCAT is designed for categorical data, the dataset was additionally preprocessed for this algorithm. Integer-valued features were converted into categorical values using the categories "0", "1", "2", "3", "4", "5", and "6_plus". These categories were selected because the corresponding integer features represent the number of injured or affected persons, and values greater than six occur only rarely in the

dataset. Boolean values were treated as categorical values with the categories "0" and "1", representing *false* and *true*, respectively.

The resulting clusters and their evaluation are presented in Section 4.1.

3.3 Association Rule Mining

Since a key objective of the proposed framework is to obtain an interpretable representation of the identified incident groups, clustering alone is insufficient. While clustering partitions incidents into groups based on their similarity, it does not explicitly describe the characteristics that distinguish the resulting clusters. Therefore, a subsequent step is required for the *characterization of the resulting clusters using ARM*. The objective of this step is not primarily to derive predictive rules, but to identify combinations of incident characteristics that are frequent, statistically supported, and particularly representative of a given cluster relative to the remaining incidents [47]. In this way, the cluster assignments are complemented by human-readable descriptions of their characteristic feature combinations, thereby providing an interpretable representation of the identified incident groups.

In the proposed framework, each bus incident is represented as a transaction containing feature-value items selected according to a predefined feature configuration. For example, an incident may be represented conceptually as

$$\begin{aligned} &\texttt{PartOfDay} = \texttt{morning}, && \texttt{IncidentType} = \texttt{collision}, \\ &\texttt{RootCause} = \texttt{driver\_error}, && \texttt{Driver\_Braking} = \texttt{true}. \end{aligned} \tag{5}$$

Three different feature sets were defined, and each clustering algorithm was applied to these feature sets (see Tables 5, 6, and 7). For the subsequent association rule analysis, the features included in the respective clustering process are referred to as *internal* features, whereas features that were not used during clustering are referred to as *external* features. The remaining features, such as *Article Title* and *Incident Date*, are referred to as *redundant* features. The analysis can therefore be performed using internal and external features.

Algorithm 1.: Proposed Association Rule Mining

Input: Incident transactions $\mathcal{D}$, clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$, characterization modes $\mathcal{M}$
Output: Stable cluster-specific patterns $\{\mathcal{P}_1^{stable}, \ldots, \mathcal{P}_K^{stable}\}$

1: **for** each mode $m \in \mathcal{M}$ **do**
2: $\mathcal{T} \leftarrow$ Build transactions from $\mathcal{D}$ according to mode m
3: $\mathcal{F} \leftarrow$ Extract frequent itemsets from $\mathcal{T}$ using FP-Growth
4: $\mathcal{F}_{closed} \leftarrow$ Extract closed itemsets from $\mathcal{T}$
5: $\mathcal{S} \leftarrow \emptyset$
6: **for** each closed itemset $f \in \mathcal{F}_{closed}$ **do**
7: Determine transactions containing f using vertical TID-set intersection
8: **for** each cluster $C_k \in \mathcal{C}$ **do**
9: Determine occurrences of f inside C_k
10: Determine occurrences of f outside C_k
11: Calculate cluster coverage, global and outside-cluster support
12: Calculate confidence and lift
13: Perform Fisher's exact test and obtain the p-value
14: Store the resulting pattern statistics in $\mathcal{S}$
15: Adjust all p-values in $\mathcal{S}$ using the Benjamini-Hochberg procedure
16: $\mathcal{S}_{sig} \leftarrow$ patterns satisfying adjusted $p \leq 0.05$
17: Calculate growth rate for each pattern in $\mathcal{S}_{sig}$
18: Calculate WRAcc for each pattern in $\mathcal{S}_{sig}$
19: **for** each cluster $C_k \in \mathcal{C}$ **do**
20: $\mathcal{S}_k \leftarrow$ significant patterns associated with C_k
21: $\mathcal{P}_k \leftarrow$ Filter $\mathcal{S}_k$ using: adjusted $p \leq 0.05$, coverage, and growth rate
22: Rank patterns according to WRAcc, growth rate, and coverage
23: Remove redundant patterns using itemset containment and quality/growth comparison
24: Analyze pattern stability using repeated resampling
25: $\mathcal{P}_k^{stable} \leftarrow$ patterns with stability ≥ 0.80
26: **return** $\{\mathcal{P}_1^{stable}, \ldots, \mathcal{P}_K^{stable}\}$

The proposed cluster characterization procedure is presented in pseudocode in Algorithm 1. Its inputs consist of incident transactions, clusters, and the mode based on the selected feature set (*internal* or *external*).

The algorithm can be divided into four main parts: closed itemset extraction from frequent patterns (lines 2–4 in the algorithm); cluster-specific pattern characterization and statistical significance testing (lines 6–14); multiple-testing correction (lines 15–16); and pattern selection and redundancy checking (lines 17–25) [35].

In the first part, frequent itemsets are extracted from the incident transactions using FP-Growth, an efficient alternative to the Apriori algorithm [36, 37]. This stage provides the set of frequent combinations satisfying the specified support threshold, rather than attempting to determine immediately which patterns are most characteristic of a particular cluster. Algorithms such as FP-Growth and Apriori can produce a very large number of patterns because the same underlying relationship may be represented by many subsets [38]. To obtain a more compact representation, frequent itemsets are reduced to frequent closed itemsets [39].

In the second part, each closed itemset is investigated using several techniques, including contrast-set analysis and cluster-specific characterization based on measures such as confidence and lift. Confidence [40] assesses the proportion of incidents containing a pattern that belong to the target cluster, whereas lift [41] compares this proportion with the baseline prevalence of the target cluster, thereby quantifying the strength of the pattern–cluster association. These descriptive measures are complemented by statistical significance testing using Fisher's exact test [42]. For each pattern-cluster pair, a 2×2 contingency table is constructed from the occurrences of the pattern inside and outside the target cluster. Fisher's exact test was chosen because individual itemsets may have low support, resulting in sparse contingency tables. Unlike the chi-square test, it does not rely on large-sample approximations and therefore provides an exact significance test for rare patterns [42].

The third part focuses on the Benjamini-Hochberg procedure [43]. It is applied to account for the large number of statistical tests performed across the mined itemsets. The procedure controls the false discovery rate by limiting the expected proportion of false discoveries among the itemsets identified as statistically significant.

After statistical testing, the final selection stage, evaluates the remaining patterns based on their descriptive properties, including support, coverage, growth rate, and weighted relative accuracy (WRAcc). Support represents the relative frequency of an itemset in the whole dataset, whereas coverage measures the proportion of the target cluster to which an itemset applies [40]. Growth rate measures how much more prevalent an itemset is in the target cluster than outside

the target cluster [40]. WRAcc [48], also known as unusualness, evaluates the trade-off between the coverage of a subgroup and its deviation from the overall population. This stage is necessary because, although the Benjamini-Hochberg procedure controls the false discovery rate of the statistical tests, it does not address redundancy among the statistically significant patterns. Therefore, redundant patterns that provide little additional information compared with stronger patterns are removed [49]. This produces a compact set of cluster characteristics that balances descriptive usefulness, statistical evidence, and non-redundancy.

The final output consists of a set of interpretable cluster characteristics expressed as combinations of incident features and their corresponding values. These characteristics provide a human-readable description of the distinctive circumstances associated with each bus-incident cluster and can subsequently be used to generate textual descriptions of the identified clusters.

4 Evaluation

To determine the appropriate number of clusters for algorithms that require a predefined number of clusters, values of $k = 2, \ldots, 20$ were investigated. This resulted in 19 different models for the K-means and K-medoids algorithms. For K-prototypes, five different values of the γ parameter were investigated for each value of k, resulting in 95 different models. For hierarchical clustering, different threshold distances were defined, and one clustering model was generated for each threshold, resulting in 57 different models.

For HDBSCAN, the minimum cluster size and the minimum number of points can be varied [50]. In this study, 18 different values of the minimum cluster size and 18 different values of the minimum number of points were investigated, resulting in $18 \times 18 = 324$ different models. For COOLCAT, different values of k and the percentage of reprocessed observations were investigated. The percentage of reprocessed observations, i.e., the proportion of observations that are reprocessed after the initial clustering, was varied between 0% and 40%, resulting in 76 different models.

The best model for each feature set was subsequently selected based on the clustering evaluation metrics. ARM was then applied to the selected clustering result.

4.1 Metric-Based Clustering Selection

Since the selected algorithms have different underlying principles and characteristics, using only a single metric to identify the best clustering model would not provide a sufficiently comprehensive basis for model selection. Therefore, the selection of the best model for each feature set was divided into three stages: hard filtering, quality calculation and evaluation, and k-neighborhood stability analysis.

In the hard-filtering stage, the number of clusters and the cluster sizes were examined. The minimum cluster size was required to exceed 1% of all incidents to avoid generating clusters that are overly specific to individual cases. In the quality calculation and evaluation stage, a combined quality score was calculated using the normalized metrics. Before calculating the combined score, all metrics were normalized such that higher values consistently indicate better clustering quality.

In the clustering quality assessment stage, the performance of the candidate clustering solutions was evaluated using a composite validity measure based on three complementary internal clustering validity indices: the Silhouette Score (SS), the Davies-Bouldin Index (DBI), and the Dunn Index (DI). Prior to their aggregation, the indices were normalized to a common scale such that higher values consistently corresponded to better clustering quality. The normalized indices were subsequently combined using the Cohesion-Dispersion (CD) index [51]:

$$CD = w_S S_{\text{norm}} + w_{DB} DB_{\text{norm}} + w_D D_{\text{norm}}, \tag{6}$$

where S_{norm}, DB_{norm}, and D_{norm} correspond to the normalized values of the Silhouette Score, Davies-Bouldin Index, and Dunn Index, respectively, while

Table 2: Clustering results and evaluation of the models for all feature sets

Method	k	k_{min}	k_{max}	SS	DBI	DI
Feature Set PIDC						
K-means	**20**	**363**	**1322**	**0.20**	**1.76**	**0.04**
K-medoids	6	1566	3308	0.21	1.41	0.02
K-prototypes	5	1527	4488	0.25	1.17	0.00
Hierarchical Clustering	20	143	1957	0.05	1.61	0.02
HDBSCAN	20	152	8718	-0.02	1.21	0.01
COOLCAT	20	317	1458	0.16	2.38	0.00
Feature Set ICC						
K-means	20	293	1102	0.32	1.42	0.03
K-medoids	**20**	**188**	**1940**	**0.36**	**1.19**	**0.01**
K-prototypes	13	554	1667	0.27	1.51	0.00
Hierarchical Clustering	2	1053	13104	0.27	1.31	0.14
HDBSCAN	35	155	4201	0.30	1.16	0.00
COOLCAT	20	271	1610	0.16	2.91	0.00
Feature Set EIC						
K-means	2	5008	9149	0.17	2.09	0.14
K-medoids	2	4778	9379	0.28	1.44	0.06
K-prototypes	**20**	**372**	**1322**	**0.19**	**1.86**	**0.00**
Hierarchical Clustering	2	3544	10613	0.25	1.45	0.03
HDBSCAN	3	217	13647	-0.28	1.38	0.06
COOLCAT	20	271	1410	0.09	3.78	0.00

Note: k_{min} denotes minimum cluster size, k_{max} denotes maximum cluster size.

w_S, w_{DB}, and w_D indicate the respective weighting coefficients. The weights satisfy

$$w_S + w_{DB} + w_D = 1. \tag{7}$$

Because a lower DBI indicates better cluster separation and compactness, it was inversely normalized prior to aggregation. Thus, all three normalized components follow the same interpretation, with larger values indicating better clustering quality.

To examine the sensitivity of the clustering evaluation to the relative importance assigned to the individual validity criteria, three weighting schemes were considered. First, an equal-weighted CD-index was defined by assigning the same weight to all three indices (Equation 8). Second, a Silhouette-focused CD-index assigned greater weight to the SS (Equation 9). Third, a separation-focused CD-index assigned greater weight to the DI and DBI, emphasizing inter-cluster separation (Equation 10). The corresponding formulations were defined as follows:

$$CD_{\text{Equal}} = \frac{1}{3}S_{\text{norm}} + \frac{1}{3}DB_{\text{norm}} + \frac{1}{3}D_{\text{norm}}, \tag{8}$$

$$CD_{\text{Sil}} = 0.50S_{\text{norm}} + 0.25DB_{\text{norm}} + 0.25D_{\text{norm}}, \tag{9}$$

$$CD_{\text{Sep}} = 0.2S_{\text{norm}} + 0.40DB_{\text{norm}} + 0.40D_{\text{norm}}. \tag{10}$$

The resulting CD-index values were used to compare the clustering solutions obtained from the different clustering algorithms and feature sets. Higher CD-index values indicate clustering solutions with stronger overall internal validity under the respective weighting scheme (Table 4).

Finally, k-neighborhood stability was evaluated to determine whether nearby values of k also resulted in good clustering solutions. This multi-stage procedure was used because the individual evaluation criteria address different aspects of clustering quality, including cluster compactness and separation between clusters, as well as the relationship between within-cluster variation and between-cluster separation [44–46].

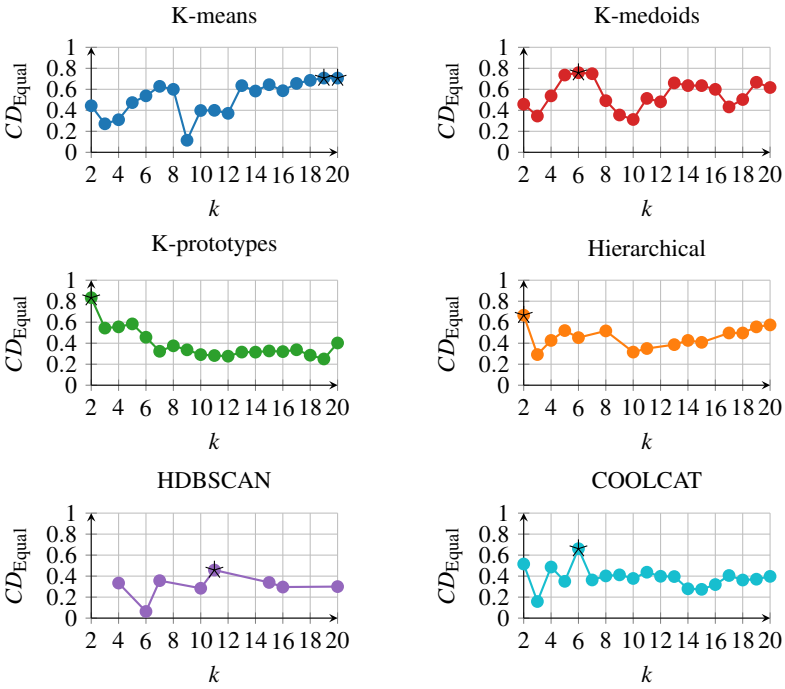


Figure 3: Clustering quality score CD_{Equal} across cluster numbers k for the six clustering methods for Feature Set PIDC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of CD_{Equal} for the corresponding method.

For hierarchical clustering, the selection procedure was extended by analyzing the dissimilarity threshold. The highest threshold value was selected for the final clustering.

COOLCAT represents a special case because it is based on entropy. Therefore, the best COOLCAT model was first selected using the same hard-filtering procedure as the other algorithms and was then evaluated based on the resulting entropy, with lower entropy indicating a preferred solution.

The figures 3, 4, 5 show the clustering quality score CD_{Equal} for each clustering method across the evaluated numbers of clusters k. The corresponding results for the other two CD-indexes are provided in Appendix figures 8, 9, and 10. These

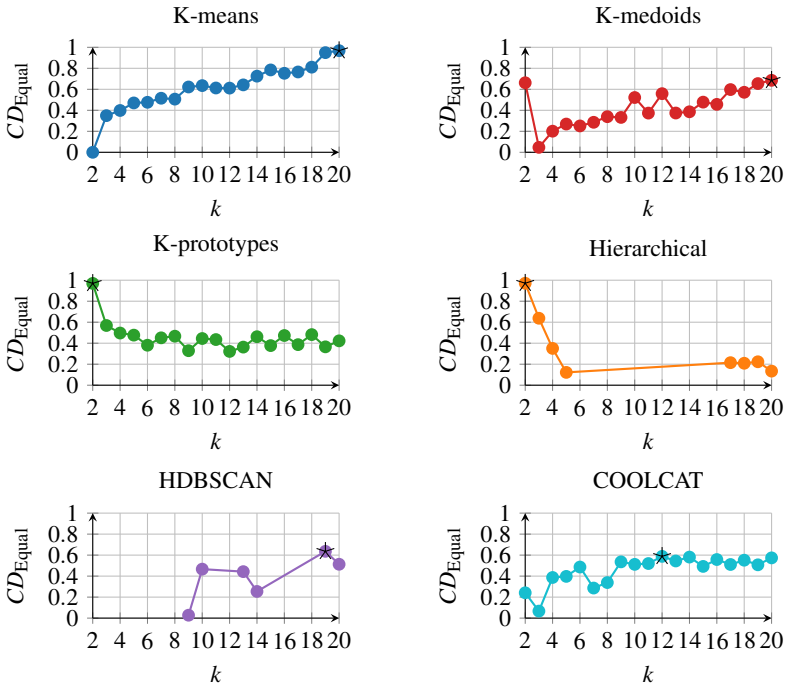


Figure 4: Clustering quality score CD_{Equal} across cluster numbers k for the six clustering methods for Feature Set ICC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of each CD_{Equal} for the corresponding method.

results are presented separately to improve readability and avoid visual clutter caused by overlapping index curves. For each clustering method, the graphs show the three CD-index variants and indicate the value of k at which each index reaches its maximum. These graphs therefore support the selection of the most appropriate number of clusters within each individual clustering method. The evaluation results of the resulting clustering solutions are then presented in Table 2, while the corresponding CD-index values under the different weighting schemes are reported in Table 4. In contrast to the method-specific graphs, these tables enable comparison of the clustering methods with each other based on their clustering characteristics and validity metrics. The tables also show that even when different clustering algorithms produce the same number of

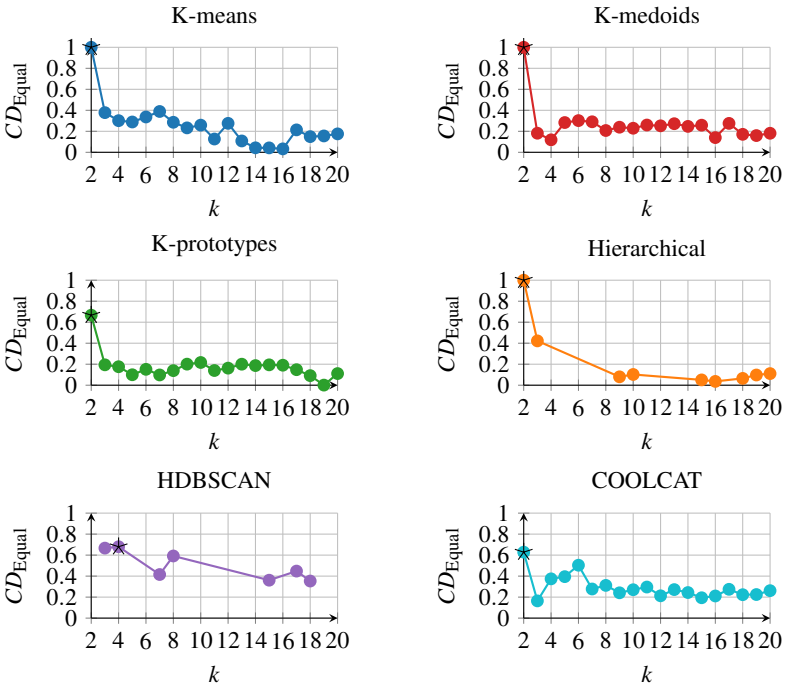


Figure 5: Clustering quality score CD_{Equal} across cluster numbers k for the six clustering methods for Feature Set EIC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of each CD_{Equal} for the corresponding method.

clusters, the resulting cluster structures can differ substantially, as reflected by the different clustering validity metrics.

For Feature Set PIDC, K-medoids and K-prototypes produced solutions with relatively small numbers of clusters, resulting in a greater concentration of incidents within individual clusters. Such coarse partitioning may limit the level of detail available for the subsequent ARM analysis. In contrast, the remaining four algorithms achieved their highest evaluated clustering quality with 20 clusters. The evaluation of these solutions was based on the CD-index, while entropy was additionally considered for COOLCAT. Among the evaluated solutions, K-means achieved the highest CD-index values across all three weighting schemes. Although COOLCAT produced a relatively low

entropy value ($E = 5.38$), this value was not sufficiently low to justify its selection over the competing clustering solutions. Therefore, the K-means solution with 20 clusters was selected for the subsequent ARM analysis of Feature Set PIDC.

For Feature Set ICC, hierarchical clustering was the only algorithm that produced a solution with a relatively small number of clusters. Although this solution achieved relatively high values for both CD_{Equal} and CD_{Sep}, it did not satisfy the predefined number-of-clusters requirement and was therefore not considered for the final selection (Figure 4, 9). In contrast, HDBSCAN produced 35 clusters, although this number was constrained by setting the minimum cluster size to 1% of the total number of incidents. Based on the favorable CD-index values obtained for K-medoids, an additional K-medoids solution with $k = 35$ was investigated to assess whether increasing the number of clusters could provide an improved partitioning. The additional solution resulted in a higher SS ($SS = 0.41$); however, its minimum cluster size was only 9 observations and therefore did not satisfy the predefined hard-filtering criterion. Based on the CD-index evaluation and the predefined cluster-size requirements, the K-medoids solution with $k = 20$ was selected for the subsequent ARM analysis of Feature Set ICC.

Feature Set EIC presented the greatest challenge for the clustering algorithms because it contained substantially more features than the other feature sets. Consequently, most algorithms achieved their best evaluated solutions with relatively small numbers of clusters, between $k = 2$-3. K-prototypes and COOLCAT were exceptions, with both algorithms achieving their best evaluated solutions with $k = 20$. The K-prototypes solution achieved substantially more favorable SS and DBI values than COOLCAT, resulting in higher CD-index values across all three weighting schemes. Furthermore, COOLCAT produced an entropy value of $E = 12.06$, which was not sufficiently low to support its selection. Based on the combined CD-index evaluation and entropy assessment, the K-prototypes solution with $k = 20$ was selected for the subsequent ARM analysis of Feature Set EIC.

4.2 Analysis of Association Rules

After selecting the clusters for each feature set, ARM was applied. Since the complete set of extracted rules is extensive, only selected representative results are presented below. For each feature set, the *internal* and *external* characterization modes are presented in the following subsections.

For Feature Set PIDC (Table 5), the **internal rules** reveal distinct combinations of incident type, root cause, time of occurrence, fatalities, and affected-person characteristics. Traffic-related patterns characterize Clusters 0 and 6, with *RootCause* = *traffic* covering 90.3% of Cluster 0 and the combination of afternoon occurrence and traffic covering 75.5%. In Cluster 6, the corresponding morning-and-traffic combination covers 83.4%. Unknown root cause is prominent in Clusters 3, 4, and 8, covering the entirety of Cluster 4, while collision with unknown root cause covers 85.4% of the same cluster. In Cluster 8, morning occurrence combined with unknown root cause covers 90.1%, whereas unknown root cause with no affected persons outside the bus covers 94.2% of Cluster 3.

Other clusters are differentiated by temporal and causal characteristics. Evening collision incidents cover 96.8% of Cluster 5, afternoon occurrence with loss of control covers 92.3% of Cluster 7, and morning occurrence with loss of control covers the entirety of Cluster 15. Bus-driver-error patterns cover 85.3% and 100% of Clusters 9 and 11, respectively. Aggression-related patterns are prominent in Clusters 2, 17, 18, and 19; for example, aggression covers 88.4% of Cluster 2 and 93.5% of Cluster 18.

The **external characterization** provides information about collision partners, driver actions, and affected-person characteristics. Car collision partners are prominent in Clusters 0, 6, 9, 11, 12, and 16, whereas unknown collision partners occur frequently in Clusters 1, 2, 17, 18, and 19. Driver braking and lane changes further distinguish several clusters, with the combination of a car collision partner and driver lane change covering 27.7% of Cluster 11 and achieving a lift of 5.38. Intoxication-related patterns occur in Clusters 2, 17, and 18, with coverages of 21.5%, 18.8%, and 18.8%, respectively. The retained internal patterns have a stability of 1.00 and adjusted p-values below 0.05.

Overall, Feature Set PIDC provides interpretable cluster profiles combining temporal, causal, and incident characteristics. For example, Cluster 0 can be summarized as predominantly involving afternoon, traffic-related incidents, frequently involving car collisions and low-risk situations.

Feature Set ICC (Table 6) produces a similar **internal structure**, with traffic, unknown root cause, aggression, loss of control, and temporal characteristics providing the main cluster distinctions. Traffic-related patterns characterize Clusters 0 and 6, while the combination of unknown root cause and no persons affected outside the bus covers 91.4%, 94.2%, and 77.2% of Clusters 1, 3, and 4, respectively. Evening collision incidents cover 96.8% of Cluster 5, night-time occurrence characterizes Cluster 13, and afternoon occurrence with unknown root cause is prominent in Cluster 14. Bus-driver-error patterns distinguish Clusters 9 and 11, while morning collision incidents characterize Cluster 12.

The **external rules** provide additional information on collision partners and driver actions. Car collision partners are prominent in Clusters 0, 6, 11, 12, and 16, whereas unknown collision partners occur in Clusters 1, 2, 3, 8, 13, 14, 17, 18, and 19. Driver braking is particularly associated with Clusters 5, 7, 12, and 15, while driver lane changes distinguish Clusters 9 and 11.

Overall, Feature Set ICC provides cluster descriptions similar to those obtained for Feature Set PIDC, with a strong emphasis on incident type, root cause, temporal characteristics, and driver-related circumstances.

Feature Set EIC due to its nature (Table 7) reveals distinct **internal patterns** (Figure 6) related mainly to incident type, time, root cause, driver involvement, affected passengers, and persons affected outside the bus (Table 8). Morning collision incidents with no affected driver, passengers, or persons outside the bus characterize Cluster 0, with the most specific pattern covering approximately 87.0% of the cluster and achieving a lift of 10.48. Collision-related patterns similarly characterize Clusters 1, 2, and 4, but differ in additional characteristics such as loss of control, afternoon occurrence, car collision partners, and driver braking. For example, collision incidents with no persons affected outside the bus cover approximately 90.0% of Cluster 2 and 89.8% of Cluster 4.

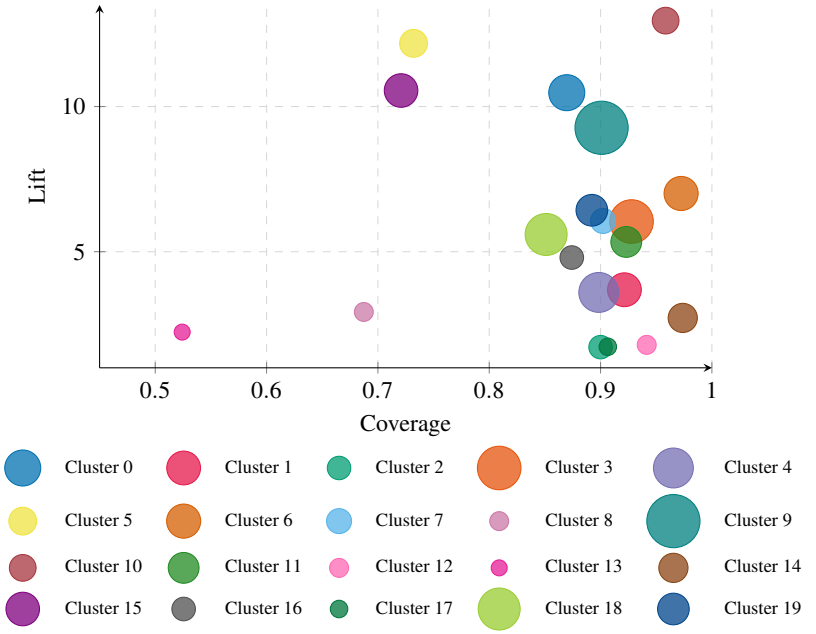


Figure 6: Selected association rules for the internal characterization of Feature Set EIC, with one representative rule selected for each cluster. The selected rules, together with their Coverage, Lift, and normalized WRAcc values, are reported in Table 10. Coverage and Lift are shown on the x- and y-axes, respectively, while bubble size represents normalized WRAcc. Bubble color denotes the corresponding cluster, with cluster numbers reported only in the legend.

Driver involvement further differentiates several clusters. Bus-driver-error incidents cover approximately 73.2% of Cluster 5 with a lift of 12.18, while affected-driver patterns cover approximately 90.2% and 68.7% of Clusters 7 and 8, respectively. Similarly, collision incidents involving an affected driver cover approximately 92.3% of Cluster 11. Clusters 6 and 19 are distinguished by incidents involving one person affected outside the bus, covering approximately 97.2% and 100% of the respective clusters. Cluster 15 is characterized by collision incidents involving one affected passenger, whereas Clusters 16 and 18 show aggression-related patterns.

The **external characterization** (Table 9) primarily differentiates the clusters by risk level and affected-person state. Low-risk incidents are prominent in

Clusters 0, 1, 4, 5, 7, and 11, with coverages ranging from 68.3% to 88.4%. High-risk incidents characterize Clusters 2, 6, 12, 14, 15, 16, 17, and 19, while medium-risk incidents are prominent in Clusters 3, 9, 10, and 18. Intoxication is particularly distinctive in Cluster 10, covering 39.3% of the cluster with a lift of 7.80 and a growth rate of 9.92. Unconscious affected persons provide additional differentiation in Clusters 10, 11, 12, 15, 16, and 19. Most retained external patterns have a stability of 1.00; the lowest reported values are 0.96 for the medium-risk pattern in Cluster 9 and 0.93 for the unconscious-state pattern in Cluster 10.

For the subsequent generation of incident scenarios from the extracted association rules, Feature Set EIC was selected as the primary feature configuration. Among the three evaluated feature sets, it provides the most comprehensive representation of the identified rule structure when considering rule coverage, association strength, and stability. In addition to capturing the core characteristics of the incidents, Feature Set EIC retains a broader range of attributes that can be used to construct more detailed and diverse incident scenarios. The extracted rules also generally exhibit high stability and statistical significance, supporting their use as a basis for subsequent rule-based scenario generation. Feature Sets A and B remain applicable where the generation process is intended to focus more specifically on causal, temporal, or incident-type characteristics. However, given the broader range of characteristics captured by its rules, Feature Set EIC was selected for the subsequent generation of incident scenarios.

4.3 Example of Incident in MR Bus Emulator

As described previously, the overall objective of this research is to support the development of effective HMI for incidents in APT. The association rules identified in this study provide a basis for generating representative incident scenarios. However, to develop and evaluate HMI concepts, these scenarios must also be reproducible in a controlled and safe experimental environment. Virtual Reality (VR) and Mixed Reality (MR) offer suitable approaches for this purpose, enabling realistic incident scenarios to be recreated while supporting the evaluation of different HMI concepts. Future work will therefore

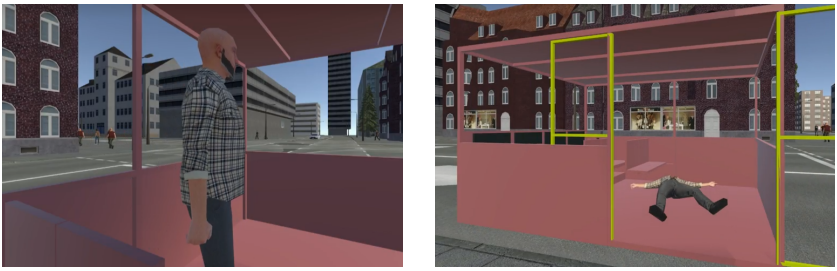


Figure 7: Example of an incident scenario implemented in the VR bus emulator.

focus on transforming the identified incident clusters and associated rules into reproducible VR/MR-based scenarios.

As a preliminary example, Figure 7 illustrates an incident scenario derived from the rules associated with Cluster 15:

"The passenger stood up while the bus was moving and intended to exit at the next bus stop. The bus subsequently collided with another object, causing the passenger to fall and sustain an injury."

This example illustrates how the extracted association rules can serve as a basis for transforming abstract statistical associations into concrete incident scenarios.

5 Conclusion

This work established a framework for extracting structured information from real-world bus incident reports and using it as a basis for generating representative incident scenarios. A taxonomy derived from articles published on Presseportal provided an overview of more than 14,157 bus-related incidents. Based on the resulting dataset, unsupervised clustering was performed using six algorithms representing different clustering principles, including distance-, density-, and entropy-based approaches, across three defined feature sets. The clustering methods were selected and evaluated according to the characteristics

of each feature set, after which ARM was applied to the resulting cluster structures.

Among the evaluated configurations, Feature Set EIC provided the most comprehensive association rule structure. Although its larger number of internal features made the clustering task more challenging, it also allowed a broader range of incident characteristics to be represented in the resulting rules. This made Feature Set EIC the most suitable configuration for the subsequent generation of incident scenarios.

The analysis also revealed several limitations and opportunities for further rule mining. Across all feature sets, the extracted rules were predominantly associated with aggression and collision incidents, while other incident types and features were less represented. This can primarily be attributed to the high proportion of collision incidents, which account for more than 50% of the dataset, as well as the considerable number of unknown or missing attribute values. Applying the ARM procedure separately to individual incident types could reduce this imbalance and enable the identification of more incident-specific patterns.

Finally, the potential application of the proposed framework was demonstrated through the generation of an example incident scenario for a VR/MR-Reality environment. In future work, this process will be further automated to transform extracted rules into reproducible VR/MR-based scenarios. These scenarios can subsequently support the development and evaluation of HMI concepts for APT.

References

[1] C. Bolzmacher, S. Panëels, C. Louison, C. Hidalgo, M. Marcano, S. Diaz and M. Anastassova, "Safety and security perception in shared autonomous shuttles: A user-centred evaluation of interface design and passenger behaviour," *Frontiers in Mechanical Engineering*, vol. 11, 2025, p. 1677376.

[2] A. O. Salonen, "Passenger's subjective traffic safety, in-vehicle security and emergency management in the driverless shuttle bus in Finland," *Transport Policy*, vol. 61, 2018, pp. 106–110, doi: 10.1016/j.tranpol.2017.10.011.

[3] A. G. Mirnig, M. Gärtner, E. Füssl, K. Ausserer, A. Meschtscherjakov, V. Wallner, M. Kubesch and M. Tscheligi, "Suppose your bus broke down and nobody came," *Personal and Ubiquitous Computing*, vol. 24, no. 6, 2020, pp. 797–812, doi: 10.1007/s00779-020-01454-8.

[4] T. Brandt, M. Wilbrink and M. Oehl, "Transparent internal human-machine interfaces in highly automated shuttles to support the communication of minimal risk maneuvers to the passengers," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 107, 2024, pp. 275–287, doi: 10.1016/j.trf.2024.09.006.

[5] P. Albertsson and T. Falkmer, "Is there a pattern in European bus and coach incidents? A literature analysis with special focus on injury causation and injury mechanisms," *Accident Analysis & Prevention*, vol. 37, no. 2, 2005, pp. 225–233, doi: 10.1016/j.aap.2004.03.006.

[6] A. D. Newton, S. D. Johnson and K. J. Bowers, "Crime on bus routes: An evaluation of a safer travel initiative," *Policing: An International Journal*, vol. 27, no. 3, 2004, pp. 302–319, doi: 10.1108/13639510410553086.

[7] A. Millonig and P. Fröhlich, "Where autonomous buses might and might not bridge the gaps in the 4 A's of public transport passenger needs: A review," in *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2018, pp. 291–297, doi: 10.1145/3239060.3239079.

[8] D. Roy, V. Deshpande and M. H. Linder, “A cluster-based taxonomy of bus crashes in the United States,” *Computational Statistics*, vol. 36, 2021, pp. 1621–1638, doi: 10.1007/s00180-021-01073-8.

[9] U. Björnstig, P.-O. Bylund, P. Albertsson, T. Falkmer, J. Björnstig and J. Petzäll, “Injury events among bus and coach occupants: Non-crash injuries as important as crash injuries,” *IATSS Research*, vol. 29, no. 1, 2005, pp. 79–87, doi: 10.1016/S0386-1112(14)60121-7.

[10] A. Kirk, R. Grant and R. Bird, “Passenger casualties in non-collision incidents on buses and coaches in Great Britain,” in *Proceedings of the 18th International Technical Conference on the Enhanced Safety of Vehicles*, Nagoya, Japan, 2003, 10 pp.

[11] E. Esenturk, A. G. Wallace, S. Khastgir and P. Jennings, “Identification of traffic accident patterns via cluster analysis and test scenario development for autonomous vehicles,” *IEEE Access*, vol. 10, 2022, pp. 6660–6675, doi: 10.1109/ACCESS.2021.3140052.

[12] U.K. Department for Transport, *Road Safety Data – STATS19*, 2020. Available: `https://www.data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data`. [Accessed] 09/25/2026.

[13] E. Han, G. Karypis, V. Kumar and B. Mobasher, “Clustering Based on Association Rule Hypergraphs,” University Digital Conservancy, 1997.

[14] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, “Fast discovery of association rules,” in *Advances in Knowledge Discovery and Data Mining*, American Association for Artificial Intelligence, 1996, pp. 307–328.

[15] A. E. Widjaja, “Analysis of Apriori and FP-Growth algorithms for market basket insights: A case study of the Bread Basket bakery sales,” *Journal of Digital Market and Digital Currency*, vol. 1, no. 1, 2024, pp. 63–83, doi: 10.47738/jdmdc.v1i1.2.

[16] Q. Cai, “Cause analysis of traffic accidents on urban roads based on an improved association rule mining algorithm,” *IEEE Access*, vol. 8, 2020, pp. 75607–75615, doi: 10.1109/ACCESS.2020.2988288.

[17] S. Huang, C. Jin, T. Chen, Z. Wang and J. Wang, “Analysis of major road traffic accident causes using a combined method of association rule and complex network,” *Journal of Advanced Transportation*, vol. 2025, no. 1, 2025, p. 8714444, doi: 10.1155/atr/8714444.

[18] S. A. Samerei, K. Aghabayk, A. Mohammadi and N. Shiwakoti, “Data mining approach to model bus crash severity in Australia,” *Journal of Safety Research*, vol. 76, 2021, pp. 73–82, doi: 10.1016/j.jsr.2020.12.004.

[19] J. Han, J. Pei and Y. Yin, “Mining frequent patterns without candidate generation,” *ACM SIGMOD Record*, vol. 29, no. 2, 2000, doi: 10.1145/335191.335372.

[20] A. A. Aldino, E. D. Pratiwi, Setiawansyah, S. Sintaro and A. D. Putra, “Comparison of market basket analysis to determine consumer purchasing patterns using FP-Growth and Apriori algorithm,” in *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, 2021, pp. 29–34, doi: 10.1109/ICOMITEE53461.2021.9650317.

[21] H. Wang and G. Liang, “Association rules between urban road traffic accidents and violations considering temporal and spatial constraints: A case study of Beijing,” *Sustainability*, vol. 17, 2025, p. 1680, doi: 10.3390/su17041680.

[22] S. Sarkar, S. Biswas, S. Koramati, A. K. Sinha and B. B. Majumdar, “Determination of various risk elements’ association causing fatal crashes in heterogeneous traffic conditions,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2679, no. 5, 2025, pp. 863–880, doi: 10.1177/03611981241312221.

[23] News aktuell GmbH, *Presseportal*, `https://www.presseportal.de/pressemitteilungen`. [Accessed] 09/16/2026.

[24] Statistisches Bundesamt (Destatis), *GENESIS*, https://genesis.destatis.de/datenbank/online/. [Accessed] 09/25/2026.

[25] Ollama, *Ollama*, https://ollama.com/. [Accessed] 09/16/2026.

[26] J. Brünnhäußer, X. Zhang, E. P. Konietzko and S. Kehl, "Automated Information Extraction from Unstructured Documents Using Large Language Models and Different Questioning Strategies," in *2025 5th ICECCME*, Zanzibar, Tanzania, United Republic of, 2025, pp. 1–6, doi: 10.1109/ICECCME64568.2025.11277979.

[27] SQLite, *SQLite*, https://sqlite.org/. [Accessed] 09/16/2026.

[28] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, 1979, pp. 100–108, doi: 10.2307/2346830.

[29] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, 2009, pp. 3336–3341, doi: 10.1016/j.eswa.2008.01.039.

[30] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 1997.

[31] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, 2012, pp. 86–97, doi: 10.1002/widm.53.

[32] D. Barbará, Y. Li and J. Couto, "COOLCAT: An entropy-based algorithm for categorical clustering," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM '02)*, McLean, VA, USA, 2002, pp. 582–589, doi: 10.1145/584792.584888.

[33] C. K. Reddy and B. Vinzamuri, "A survey of partitional and hierarchical clustering algorithms," pp. 87–110, 2018, doi: 10.1201/9781315373515-4.

[34] L. McInnes, J. Healy and S. Astels, “hdbscan: Hierarchical density based clustering,” *The Journal of Open Source Software*, vol. 2, 2017, p. 205, doi: 10.21105/joss.00205.

[35] C. Esnault, M.-L. Gadonna, M. Queyrel, A. Templier and J.-D. Zucker, “Q-Finder: An algorithm for credible subgroup discovery in clinical data analysis—An application to the International Diabetes Management Practice Study,” *Frontiers in Artificial Intelligence*, vol. 3, 2020, p. 559927, doi: 10.3389/frai.2020.559927.

[36] M. Hossain, A. H. M. Sattar and M. K. Paul, “Market basket analysis using Apriori and FP Growth algorithm,” in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–6, doi: 10.1109/ICCIT48885.2019.9038197.

[37] C. Borgelt, “An implementation of the FP-growth algorithm,” in *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, Chicago, IL, USA, 2005, pp. 1–5, doi: 10.1145/1133905.1133907.

[38] M. Z. Ashrafi, D. Taniar and K. Smith, “An efficient compression technique for frequent itemset generation in association rule mining,” in *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2005, pp. 125–135, doi: 10.1007/11430919_16.

[39] M. J. Zaki and C.-J. Hsiao, “CHARM: An efficient algorithm for closed itemset mining,” in *Proceedings of the 2002 SIAM International Conference on Data Mining (SDM)*, 2002, pp. 457–473, doi: 10.1137/1.9781611972726.27.

[40] J. Han, M. Kamber and J. Pei, *Data Mining: Concepts and Techniques*, 2nd ed., Elsevier, Amsterdam, 2006, ISBN 978-0-12-381479-1, Chapter 6.

[41] S. D. Bay and M. J. Pazzani, “Detecting group differences: Mining contrast sets,” *Data Mining and Knowledge Discovery*, vol. 5, 2001, pp. 213–246, doi: 10.1023/A:1011429418057.

[42] M. A. Fernández-Cásseres and D. Russi-Pulgar, “Categorical variable analyses: Chi-square, Fisher exact, Mantel–Haenszel,” in *Translational Sports Medicine*, A. E. M. Eltorai, J. A. Bakal, S. F. DeFroda and B. D. Owens, Eds., Academic Press, 2023, pp. 151–156, doi: 10.1016/B978-0-323-91259-4.00043-6.

[43] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: A practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, 1995, pp. 289–300, doi: 10.1111/j.2517-6161.1995.tb02031.x.

[44] M. Gagolewski, M. Bartoszuk and A. Cena, “Are cluster validity measures (in) valid?” *Information Sciences*, vol. 581, 2021, pp. 620–636, doi: 10.1016/j.ins.2021.10.004.

[45] A. M. Ikotun, F. Habyarimana and A. E. Ezugwu, “Cluster validity indices for automatic clustering: A comprehensive review,” *Heliyon*, vol. 11, no. 2, 2025, p. e41953, doi: 10.1016/j.heliyon.2025.e41953.

[46] R. Ünlü and P. Xanthopoulos, “Estimating the number of clusters in a dataset via consensus clustering,” *Expert Systems with Applications*, vol. 125, 2019, pp. 33–39, doi: 10.1016/j.eswa.2019.01.074.

[47] R. Agrawal, T. Imieliński and A. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 1993, pp. 207–216, doi: 10.1145/170035.170072.

[48] C. J. Carmona, M. J. del Jesus and F. Herrera, “A unifying analysis for the supervised descriptive rule discovery via the weighted relative accuracy,” *Knowledge-Based Systems*, vol. 139, 2018, pp. 89–100, doi: 10.1016/j.knosys.2017.10.015.

[49] J. Li, J. Liu, H. Toivonen, K. Satou, Y. Sun and B. Sun, “Discovering statistically non-redundant subgroups,” *Knowledge-Based Systems*, vol. 67, 2014, pp. 315–327, doi: 10.1016/j.knosys.2014.04.030.

[50] Doxakis, *HdbscanSharp*, NuGet package, version 3.0.1, 2025. Available at: `https://www.nuget.org/packages/HdbscanSharp`. [Accessed] 09/25/2026.

[51] C. R. Caro-Martín, J. M. Delgado-García, A. Gruart and R. Sánchez-Campusano, "Spike sorting based on shape, phase, and distribution features, and K-TOPS clustering with validity and error indices," *Scientific Reports*, vol. 8, no. 1, 2018, p. 17796, doi: 10.1038/s41598-018-35491-4.

6 Appendix

Table 3: Features in created dataset

Field	Type	Required	Description	Allowed Values
			Overall Information	
Incident Id	String	Yes	Unique incident ID. Extracted from the article.	Any string
Article Url	String	Yes	URL of the article.	Any string
Article Title	String	Yes	Title of the article.	Any string
Relevant Article Text	String	Yes	Extracted text from the article.	Any string
Publish Date	Date	Yes	Date on which the incident was published.	Any date within the considered time span
Incident Date	Date	Yes	Date on which the incident occurred.	Any date within the considered time span
City	String	No	City where the incident happened.	Any string

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
			Details	
Incident Type	Enum	Yes	Describes what happened.	collision, rollover, fire, medical_-emergency, technical_failure, aggression, evacuation, breakdown, road_obstruction, driver_impairment, crime, other
Part Of Day	Enum	No	Part of the day when the incident happened.	morning, afternoon, evening, night
Root Cause	Enum	Yes	Primary reason for the incident.	technical, medical, driver_-medical_event, weather, traffic, bus_driver_error, driver_error, driver_impairment, aggression, passenger_behavior, uncontrol-lable_person, infrastructure, fire, robbery_attempt, loss_of_control, wildlife, pedestrian_caused, unknown

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Secondary Causes	List <Enum>	No	Additional contributing causes. Always sorted.	Same values as `RootCause`.
Weather	List <Enum>	No	Weather conditions contributing to the incident.	clear, sunny, cloudy, rain, snow, ice, fog, hail, storm, wind, flood
Road Condition	Enum	No	Road conditions that could potentially have caused the incident.	icy, wet, slippery, snowy, unknown
Lighting	Enum	No	Lighting conditions that could potentially have caused the incident.	dark, daylight, street_lighting, unknown
Bus Type	Enum	No	Type of bus involved in the incident.	line_bus, city_bus, school_bus, coach, long_distance_bus, other, unknown
Passengers Affected	Integer	No	Number of passengers directly affected.	Any non-negative integer
Driver Affected	Integer	No	Number of affected drivers, usually 0 or 1.	Any non-negative integer

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Pedestrians Affected	Integer	No	Number of affected pedestrians.	Any non-negative integer
Cyclists Affected	Integer	No	Number of affected cyclists.	Any non-negative integer
Motorcyclists Affected	Integer	No	Number of affected motorcyclists.	Any non-negative integer
Car Occupants Affected	Integer	No	Number of affected car occupants.	Any non-negative integer
Emergency Responders Affected	Integer	No	Number of affected emergency responders.	Any non-negative integer
Others Affected	Integer	No	Number of other affected persons.	Any non-negative integer
Minor Injuries	Integer	No	Number of persons with minor injuries.	Any non-negative integer

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Serious Injuries	Integer	No	Number of seriously injured persons.	Any non-negative integer
Fatalities	Integer	No	Number of fatalities.	Any non-negative integer
Driver Status	Enum	Yes	Final condition of the driver.	unaffected, injured, seriously_injured, deceased, medical_emergency, assaulted, unknown
Driver Actions	List <Enum>	No	Actions performed by the driver.	braking, changed_lane, stopped_safely, called_emergency_services, helped_passengers, overtaking, evacuated_bus, turning, accelerating, fled_scene, prevented_collision, first_aid

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Affected Person State	List <Enum>	No	Physical or behavioral state of affected passengers, pedestrians, cyclists, or other involved persons before or during the incident. Multiple values are allowed. Store in canonical order.	seated, standing, boarding, alighting, waiting_at_stop, crossing_road, walking, cycling, driving, riding_motorcycle, wearing_seatbelt, holding_handrail, using_wheelchair, using_walker, using_stroller, medical_emergency, collapsed, unconscious, intoxicated, panic, none, unknown

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Affected Person Actions	List <Enum>	No	Actions performed by affected passengers, pedestrians, cyclists, or other involved persons during the incident. Multiple values are allowed. Store in canonical order.	helped_driver, helped_passengers, provided_first_aid, called_emergency_services, restrained_aggressor, evacuated_bus, warned_others, attacked_driver, attacked_passenger, attacked_pedestrian, threatened, used_weapon, vandalized_bus, fare_evasion, violent, uncontrollable, running, distracted, using_phone, ignored_signals, entered_road, left_scene, none, unknown
Collision Partner	List <Enum>	No	All participants in the collision.	car, truck, bus, pedestrian, cyclist, motorcycle, object, animal, tram, unknown, other
Aggressor Weapon	String	No	Weapon used by the aggressor, if present.	Any string

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
Aggressor Present	Boolean	No	Whether the aggressor remained at the crime scene.	true, false
Aggressor Type	Enum	No	Type of aggressor, if present.	person, group, unknown
Aggressor Arrested	Boolean	No	Whether the aggressor was arrested.	true, false
Emergency Services Called	Boolean	No	Whether emergency services were called.	true, false
Fire Department Present	Boolean	No	Whether the fire department responded to the incident.	true, false
Police Present	Boolean	No	Whether the police responded to the incident.	true, false
Medical Services Present	Boolean	No	Whether medical services responded to the incident.	true, false

Continued on next page

Table 3 continued from previous page

Field	Type	Required	Description	Allowed Values
			Outcome	
Summary	String	Yes	Short description of the incident.	Any string
Outcome	List <Enum>	Yes	Overall consequences.	property_damage, service_delay, road_closed, bus_disabled, evacuation, injuries, fatalities, criminal_investigation
Risk	Enum	Yes	Risk level derived from the *Outcome*.	low, medium, high, very high

Table 4: Normalized clustering validity indices and CD-index scores under different weighting schemes.

Features	**Clustering Method**	k	**Normalized validity indices**			**CD-index**		
			SS	**DBI**	**DI**	CD_{Equal}	CD_{Sil}	CD_{Sep}
Set A	**K-means**	20	0.83	0.51	1.00	**0.78**	**0.79**	**0.77**
	K-medoids	6	0.86	0.80	0.45	0.70	0.74	0.67
	K-prototypes	5	1.00	1.00	0.01	0.67	0.75	0.61
	Hierarchical Clustering	20	0.25	0.63	0.69	0.53	0.46	0.58
	HDBSCAN	20	0.00	0.97	0.17	0.38	0.28	0.45
	COOLCAT	20	0.65	0.00	0.00	0.22	0.32	0.13
Set B	K-means	20	0.80	0.85	0.23	0.63	0.67	0.59
	K-medoids	20	1.00	0.98	0.08	**0.69**	**0.77**	**0.62**
	K-prototypes	13	0.53	0.80	0.00	0.44	0.46	0.43
	Hierarchical Clustering	2	0.55	0.91	1.00	**0.82**	0.76	**0.88**
	HDBSCAN	35	0.67	1.00	0.03	0.57	0.59	0.55
	COOLCAT	20	0.00	0.00	0.00	0.00	0.00	0.00

Continued on next page

Table 4 – continued from previous page

Features	**Clustering Method**	k	**Normalized validity indices**			**CD-index**		
			SS	**DBI**	**DI**	CD_{Equal}	CD_{Sil}	CD_{Sep}
	K-means	2	0.80	0.71	1.00	**0.83**	0.83	**0.84**
Set C	K-medoids	2	1.00	0.98	0.44	0.81	**0.86**	0.77
	K-prototypes	20	0.83	0.80	0.00	**0.55**	**0.62**	**0.49**
	Hierarchical Clustering	2	0.95	0.97	0.20	0.71	0.77	0.66
Set C	HDBSCAN	3	0.00	1.00	0.44	0.48	0.36	0.57
	COOLCAT	20	0.66	0.00	0.00	0.22	0.33	0.13

Note: SS = Silhouette Score; DBI = Davies–Bouldin Index; DI = Dunn Index. Higher normalized values indicate better clustering quality.

Table 5: Feature Set PIDC

Internal	External
Part Of Day	Affected Person State
Incident Type	Road Condition
Root Cause	Lighting
Passengers Affected	Bus Type
Driver Affected	Collision Partner
Total Affected Persons Not In Bus	Weather
Total Injuries	Affected Person Actions
Fatalities	Aggressor Weapon
Driver Status	Second Causes
Driver Actions	Risk

Table 6: Feature Set ICC

Internal	External
Part Of Day	Affected Person State
Incident Type	Road Condition
Root Cause	Lighting
Passengers Affected	Bus Type
Driver Affected	Collision Partner
Total Affected Persons Not In Bus	Weather
Minor Injuries	Affected Person Actions
Serious Injuries	Aggressor Weapon
Fatalities	Risk

Continued on next page

Table 6 continued from previous page

Internal	External
Second Causes	Driver Status

Table 7: Feature Set EIC

Internal	External
Part Of Day	Affected Person State
Incident Type	Road Condition
Root Cause	Lighting
Passengers Affected	Risk
Driver Affected	
Total Affected Persons Not In Bus	
Minor Injuries	
Serious Injuries	
Fatalities	
Second Causes	
Bus Type	
Collision Partner	
Weather	
Affected Person Actions	
Aggressor Weapon	
Driver Status	
Driver Actions	

Table 8: Descriptions of the clusters based on the extracted **internal** rules for Feature Set EIC

Cluster	Description of the cluster
0	Morning **collision** incidents, with patterns indicating no affected driver, no affected passengers, and no persons affected outside the bus. The extracted rules also include line buses, unknown driver status, and zero fatalities.
1	**Collision** incidents, with patterns indicating no affected driver, no affected passengers, and no persons affected outside the bus. The extracted rules also include loss of control, unknown driver status, and a car as the collision partner.
2	**Collision** incidents with no persons affected outside the bus. The extracted rules additionally include line buses, car collision partners, zero fatalities, and driver braking.
3	Line-bus incidents with an unknown collision partner and no affected driver. The extracted rules also include unknown driver status, zero fatalities, no persons affected outside the bus, and **aggression** as the root cause.
4	**Collision** incidents, with patterns indicating no affected driver, no affected passengers, and no persons affected outside the bus. The extracted rules also include afternoon occurrences, unknown driver status, line buses, and zero fatalities.
5	Incidents associated with **bus driver error** and no affected passengers. The extracted rules also include no persons affected outside the bus, collision incidents, zero fatalities, and unknown driver status.
6	Incidents involving one **person affected outside** the bus. The extracted rules additionally include collision incidents, no affected passengers, no affected driver, and zero fatalities.
7	Incidents involving an **affected driver** and no affected passengers. The extracted rules also include unaffected driver status, zero fatalities, and no persons affected outside the bus.

Continued on next page

Cluster	Description of the cluster based on the extracted rules
8	Incidents involving an **affected driver**. The extracted rules additionally include **collision** incidents, line buses, zero fatalities, and no persons affected outside the bus.
9	Incidents involving an **unknown collision partner**, unknown driver status, no affected passengers, and an unknown root cause. The extracted rules also include no affected driver, zero fatalities, and no persons affected outside the bus.
10	Incidents involving an unknown collision partner and an affected driver, with zero fatalities. The extracted rules also include aggression incidents, **aggression** as the root cause, no affected passengers, and no persons affected outside the bus.
11	**Collision** incidents involving an **affected driver**. The extracted rules additionally include zero fatalities, no affected passengers, no persons affected outside the bus, and a **car** as the collision partner.
12	**Collision** incidents with no persons affected outside the bus. The extracted rules also include **loss of control**, line buses, zero fatalities, no affected passengers, and a **car** as the collision partner.
13	Incidents involving an **affected driver outside the bus**. The extracted rules additionally include zero fatalities, no persons affected outside the bus, unaffected bus driver status, and collision incidents.
14	**Collision** incidents with no affected driver, zero fatalities, and no affected passengers. The extracted rules also include a **car** as the collision partner, unknown driver status, and **traffic** as the root cause.
15	**Collision** incidents involving **one affected passenger**. The extracted rules additionally include zero fatalities, no persons affected outside the bus, no affected driver, line buses, and driver braking.
16	**Aggression** incidents with zero fatalities. The extracted rules also include aggression as the root cause, an unknown collision partner, and no persons affected outside the bus.

Continued on next page

Cluster	Description of the cluster based on the extracted rules
17	**Collision** incidents with no persons affected outside the bus. The extracted rules additionally include an affected driver, morning occurrences, zero fatalities, and a **car** as the collision partner.
18	Incidents involving **aggression** as the root cause and no affected passengers. The extracted rules also include unknown driver status, an unknown collision partner, no affected driver, and zero fatalities.
19	Incidents involving **one person affected outside the bus**. The extracted rules additionally include collision incidents, no affected passengers, no affected driver, and zero fatalities.

Table 9: Descriptions of the clusters based on the extracted **external** rules for Feature Set EIC

Cluster	Description
0	**Low risk** is the main distinguishing characteristic, occurring in 85.2% of the cluster. The rule has a lift of 1.70 and a growth rate of 1.77, indicating that low risk is more represented within this cluster than outside it.
1	This cluster is characterized primarily by **low risk**, with a cluster support of 83.5%. The associated lift (1.67) and growth rate (1.74) indicate an increased representation of low-risk incidents within the cluster.
2	**High risk** provides the principal distinguishing characteristic, occurring in 44.5% of the cluster. It has a lift of 1.74 and a growth rate of 1.83, showing that high risk is more frequently represented within this cluster than outside it.

Continued on next page

Cluster	Description
3	Two characteristics distinguish this cluster: **medium risk** and **intoxicated affected persons**. Medium risk has a cluster support of 64.7%, while intoxication occurs in 11.0% of the cluster and has a comparatively high lift of 2.19.
4	A clear **low-risk profile** characterizes this cluster, with low risk occurring in 86.6% of cases. The corresponding lift of 1.73 and growth rate of 1.84 indicate an increased representation relative to outside the cluster.
5	**Low risk** is strongly represented, covering 88.4% of the cluster. The characteristic has a lift of 1.77 and a growth rate of 1.83, indicating that low-risk incidents occur more frequently within this cluster than outside it.
6	**High risk** is the distinguishing characteristic, appearing in 69.6% of the cluster. Its lift of 2.73 and growth rate of 2.99 indicate a substantially higher representation of high-risk incidents within the cluster.
7	The cluster is primarily associated with **low risk**, which occurs in 74.0% of cases. Its lift is 1.48 and its growth rate is 1.50, indicating a higher representation of low risk within the cluster.
8	**No strong and stable distinguishing characteristics** were identified for this cluster based on the supplied rules.
9	**Medium risk** is the identified characteristic, occurring in 23.6% of the cluster. With a lift of 1.32 and a growth rate of 1.37, medium risk is represented more frequently within the cluster than outside it.
10	The cluster combines **medium risk** with **intoxicated and unconscious affected persons**. Medium risk occurs in 60.5% of cases, while intoxication has a particularly high lift of 7.80 and growth rate of 9.92.

Continued on next page

Cluster	Description
11	**Low risk** characterizes 68.3% of the cluster, while **unconscious affected persons** occur in 16.2%. The latter has a lift of 2.30 and a growth rate of 2.46, indicating increased representation within the cluster.
12	**High risk** is present in 60.8% of the cluster, accompanied by **unconscious affected persons** in 16.4% of cases. Both characteristics have lifts above 2, indicating higher representation within the cluster than outside it.
13	**No strong and stable distinguishing characteristics** were identified for this cluster based on the supplied rules.
14	**High risk** is the identified distinguishing characteristic, occurring in 41.2% of the cluster. Its lift of 1.61 and growth rate of 1.67 indicate that high risk is more represented within the cluster than outside it.
15	The cluster is characterized by **high risk** and **unconscious affected persons**. High risk occurs in 47.4% of cases, while unconsciousness occurs in 20.1% and has a lift of 2.86.
16	**High risk** characterizes 75.1% of the cluster, alongside rules for **intoxicated** and **unconscious affected persons**. High risk has a lift of 2.94, while intoxication and unconsciousness have lifts of 3.14 and 2.28, respectively.
17	**High risk** is the main distinguishing characteristic, occurring in 54.0% of the cluster. The characteristic has a lift of 2.12 and a growth rate of 2.19, indicating increased representation within the cluster.
18	**Medium risk** is strongly represented, occurring in 69.7% of the cluster, while **intoxicated affected persons** occur in 15.9%. Their lifts are 3.90 and 3.15, respectively, indicating increased representation within the cluster.
19	The cluster is characterized by **high risk** and **unconscious affected persons**. High risk occurs in 52.3% of cases, while unconsciousness occurs in 13.0%, with corresponding lifts of 2.05 and 1.85.

Table 10: Representative association rules for the internal characterization of Feature Set EIC (Figure 6). Coverage, Lift, and normalized WRAcc are reported.

Cluster	Selected rule	Coverage	Lift	Normalized WRAcc
0	DriverAffected is 0, IncidentType is collision, TimeOfDay is morning, PassengersAffected is 0, and TotalAffectedPersonsNotInBus is 0	**0.87**	**10.48**	**0.54**
1	DriverAffected is 0, IncidentType is collision, PassengersAffected is 0, and TotalAffectedPersonsNotInBus is 0	**0.92**	**3.69**	**0.48**
2	IncidentType is collision and TotalAffectedPersonsNotInBus is 0	**0.90**	**1.71**	**0.20**
3	VehicleType is line bus, OtherParty is unknown, and DriverAffected is 0	**0.93**	**6.04**	**0.74**
4	DriverAffected is 0, IncidentType is collision, PassengersAffected is 0, and TotalAffectedPersonsNotInBus is 0	**0.90**	**3.60**	**0.64**
5	PassengersAffected is 0, RootCause is bus-driver error, and TotalAffectedPersonsNotInBus is 0	**0.73**	**12.18**	**0.32**

Continued on next page

Table 10 – continued

Cluster	Selected rule	Coverage	Lift	Normalized WRAcc
6	TotalAffectedPersonsNotInBus is 1	**0.97**	**7.01**	**0.49**
7	DriverAffected is 1 and PassengersAffected is 0	**0.90**	**6.06**	**0.24**
8	DriverAffected is 1	**0.69**	**2.92**	**0.08**
9	OtherParty is unknown, DriverStatus is unknown, PassengersAffected is 0, and RootCause is unknown	**0.90**	**9.27**	**1.00**
10	OtherParty is unknown, DriverAffected is 1, and Fatalities is 0	**0.96**	**12.96**	**0.29**
11	DriverAffected is 1 and IncidentType is collision	**0.92**	**5.34**	**0.40**
12	IncidentType is collision and TotalAffectedPersonsNotInBus is 0	**0.94**	**1.79**	**0.08**
13	DriverAffected is 1	**0.52**	**2.23**	**0.00**
14	DriverAffected is 0, Fatalities is 0, IncidentType is collision, and PassengersAffected is 0	**0.97**	**2.72**	**0.36**
15	IncidentType is collision and PassengersAffected is 1	**0.72**	**10.55**	**0.47**
16	Fatalities is 0 and RootCause is aggression	**0.87**	**4.80**	**0.20**

Continued on next page

Table 10 – continued

Cluster	Selected rule	Coverage	Lift	Normalized WRAcc
17	IncidentType is collision and TotalAffectedPersonsNotInBus is 0	**0.91**	**1.72**	**0.04**
18	DriverStatus is unknown, PassengersAffected is 0, and RootCause is aggression	**0.85**	**5.59**	**0.70**
19	TotalAffectedPersonsNotInBus is 1	**0.89**	**6.43**	**0.42**

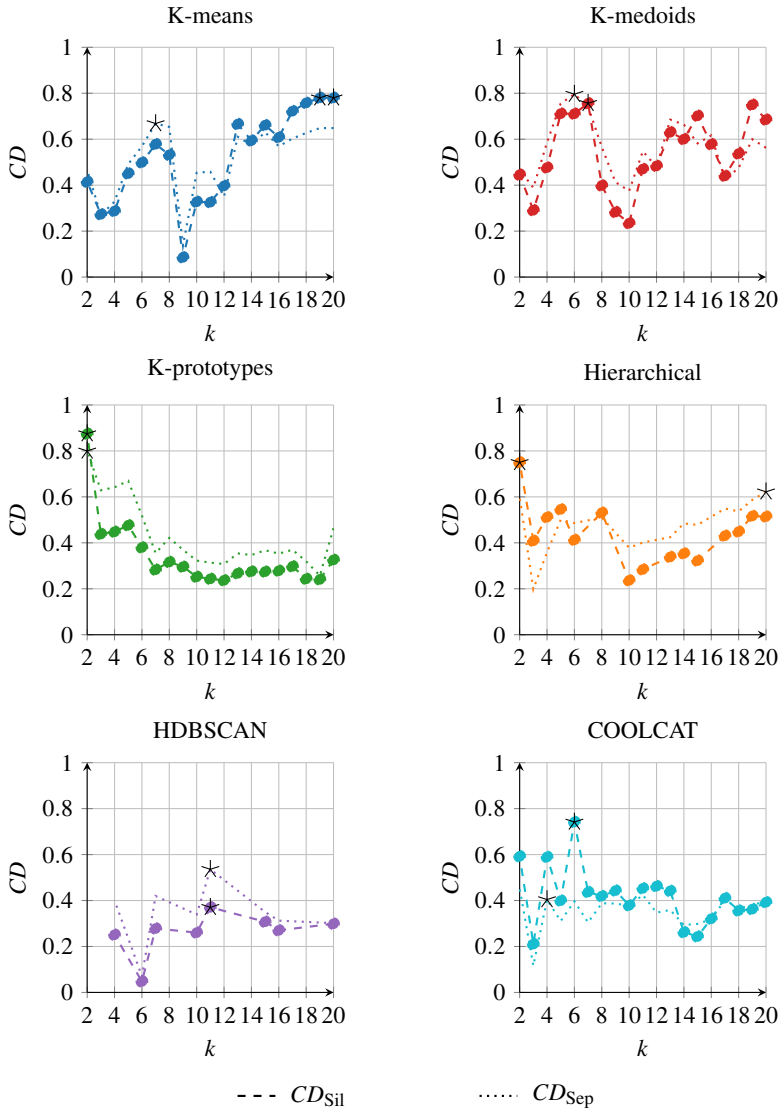


Figure 8: Clustering quality scores CD_{Sil} and CD_{Sep} across cluster numbers k for the six clustering methods for Feature Set PIDC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of each CD index for the corresponding method.

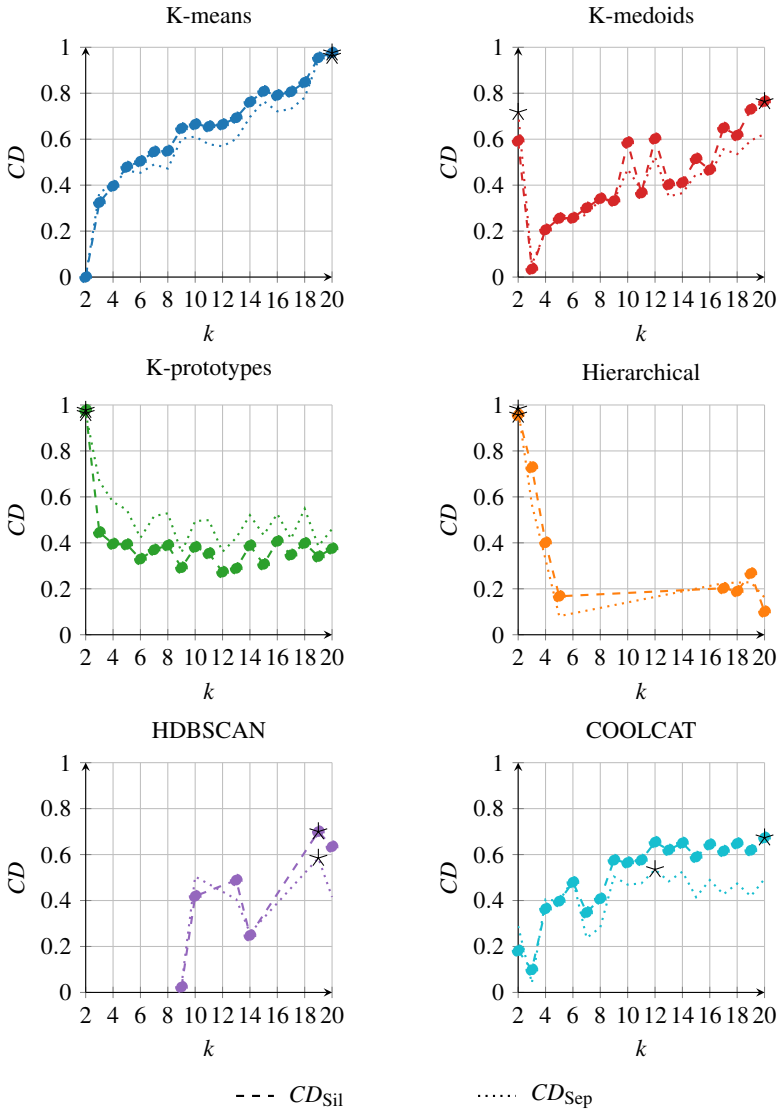


Figure 9: Clustering quality scores CD_{Sil} and CD_{Sep} across cluster numbers k for the six clustering methods for Feature Set ICC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of each CD index for the corresponding method.

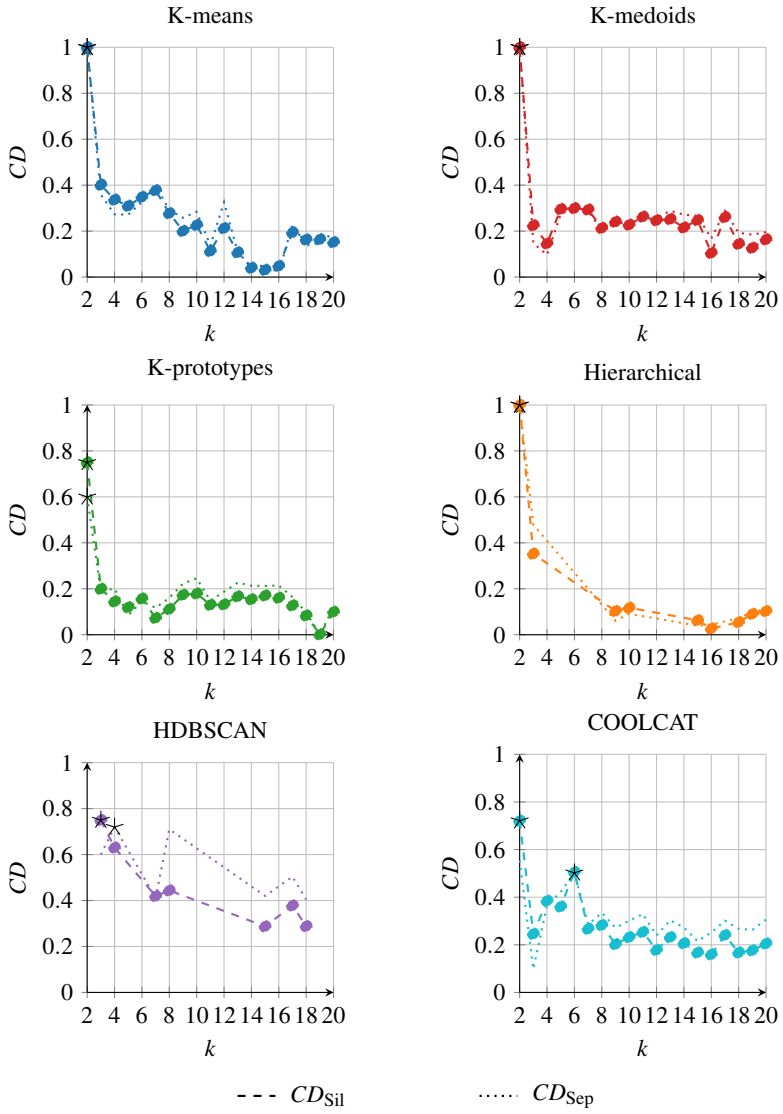


Figure 10: Clustering quality scores CD_{Sil} and CD_{Sep} across cluster numbers k for the six clustering methods for Feature Set EIC. The underlying SS, DBI, and DI are normalized separately within each clustering method across the evaluated values of k. Stars indicate the maximum value of each CD index for the corresponding method.

Cold-Start Active Learning on Tabular Data: A Systematic Benchmark

Bjarne Jaster, Martin Kohlhase

Hochschule Bielefeld, Center for Applied Data Science Gütersloh
Interaktion 1, 33619 Bielefeld
E-Mail: {bjarne.jaster, martin.kohlhase}@hsbi.de

1 Introduction

Machine learning (ML) is key to modern industrial operations, where tabular data remains an essential backbone despite the increased use of unstructured data [4]. However, deploying ML models is often hindered by a scarcity of labeled samples despite an abundance of unlabeled data. *Active learning* (AL) mitigates this bottleneck by iteratively selecting the most informative data points for labeling, which significantly reduces labeling time and expert costs [19].

Despite these promising advantages and a surge in academic research, AL is rarely deployed in real-world industrial settings [2]. A primary reason for this is that selecting an AL strategy prior to deployment is challenging because evaluating an AL pipeline is nearly impossible without spending the very labeling budget AL aims to minimize. In standard supervised ML, model selection and hyperparameter tuning are usually performed by evaluating and comparing different pipelines for a specific task. In contrast, AL strategies cannot be selected and evaluated this way without annotating dedicated hold-out test sets, which often already defeats the cost savings of AL [12]. This inherent validation bottleneck makes deploying AL in real-world scenarios essentially a blind choice. Additionally, practitioners must select not only the AL strategy and its underlying base model, but also an initialization strategy to provide the initial training set required by the base model.

In this work, we focus specifically on the selection of the initialization strategy. Because no trained model is available at this stage, initialization methods operate in a model-free way, by leveraging geometric properties like data density, distance, or coverage to construct an initial training set. In contrast, most established AL strategies are model-based and typically rely on the uncertainty of the base models predictions, either exclusively or combined with geometric measures. For model-based sampling to be effective, the base model must accurately reflect the underlying data distribution, as a misspecified model can generate misleading query signals [5]. This phenomenon is illustrated in Figure 1. A poorly initialized active learner (top left) is aware of only two out of three classes before starting the model-based AL part (here Uncertainty Sampling [15]). Consequently, it focuses its queries primarily on refining the decision boundary between the known classes while overlooking the unobserved class for most of the AL process. In contrast, the active learner that has knowledge of all three classes after the initialization can focus on refining all three class boundaries. Although missing an entire class represents an extreme scenario, less obvious coverage gaps can cause similarly biased query selections. Therefore, selecting a good initial set is critical for the success of AL.

Despite its practical significance, most AL research neglects the selection of the initial set. Historically, fewer than 6% of AL papers used a targeted approach to construct the initial pool [11] and revert to using naive random sampling, which is a practice that is also applied in most recent AL benchmarks. While this suffices as an equal baseline for academic comparisons of AL strategies, it falls short in real-world scenarios where a good initialization strategy can potentially improve the initial model quality and with that the overall label efficiency.

Although the initialization problem is largely overlooked for tabular data, it is actively studied in Deep Active Learning (Deep AL) as the *cold-start problem* (e.g. [6]). In deep learning problems, differences in raw feature values (e.g. pixel intensities) often carry little semantic meaning, which makes it hard for geometric initialization strategies to capture the dataset distribution. *Self-supervised learning* (SSL) methods tackle this challenge by learning a semantically meaningful representation which is called embedding. They exploit domain-specific structures (e.g. spatial relations for vision tasks or

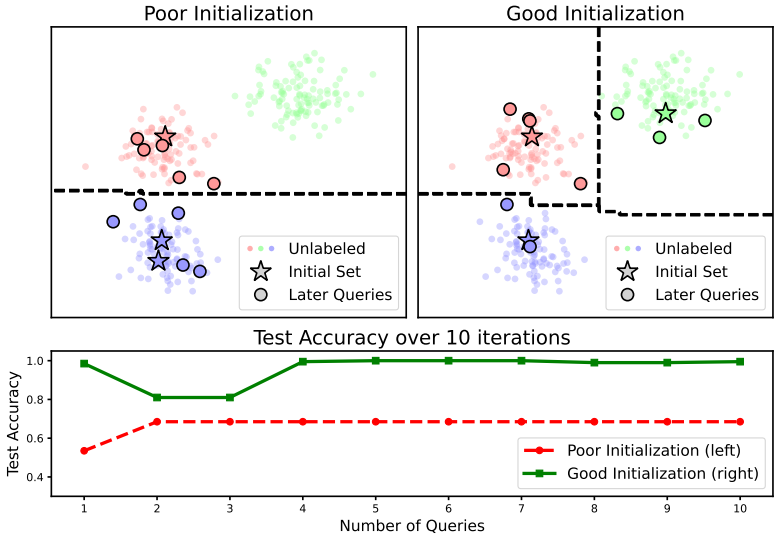


Figure 1: Illustrative example. The top-left panel shows an initial set (★) containing only two of three classes, whereas the top-right panel shows an initial set that includes all three classes. After the initialization an uncertainty-based active learner queries 10 additional samples (•). The lower panel illustrates the accuracy of a Random Forest trained on the queried points over the iterations.

semantic relations in language tasks) to learn a function that transforms the features into another space (embedding) where geometric proximity reflects true semantic similarity [22]. For tabular data, however, the challenge is different: features already carry explicit semantics but are highly heterogeneous across mixed types and scales. Consequently, it remains an open question whether SSL pretext tasks can construct an embedding space that offers benefits for initialization methods on tabular datasets.

To address the lack of systematic benchmarks and practical guidance for the initialization of AL on tabular data, we conduct a comprehensive empirical study. Our evaluation compares model-free initialization strategies developed originally for tabular data as well as for Deep AL with and without SSL-based feature representations across diverse classification and regression tasks. Specifically, our study is driven by three central research questions (RQs):

- **RQ1:** Is there a specific initialization strategy or general initialization paradigm that consistently outperforms others on tabular data?
- **RQ2:** What proportion of the overall label budget should be allocated to initialization, and does a larger initial set enhance the subsequent model-based AL process?
- **RQ3:** Do representations derived from SSL pretext tasks improve initialization quality for tabular data?

2 Related Work

AL is a subfield of ML designed to reduce labeling costs while maximizing model performance. Similar to semi-supervised ML, it uses labeled and unlabeled data. The difference is that the unlabeled data is not included with the purpose of improving the models performance but to select new samples for labeling by an oracle (e.g. domain expert or industrial machine). For this most AL strategies train a base model on a small labeled initial set and use its information to select one or more samples. After these new samples are labeled and added to the training set, the model is retrained and the process continues until some stopping-criterion is met.

Depending on the availability of the unlabeled data, AL is traditionally categorized into *membership query synthesis*, *stream-based*, and *pool-based* settings [19]. Pool-based AL, where samples are sequentially selected from an unlabeled pool, is the most common scenario for real-world tabular datasets. Consequently, we focus exclusively on this setting.

2.1 Initialization Methods

Initialization strategies fall into three different paradigms: *random sampling*, *space-filling or diversity-based* methods, and *representativeness-based* methods. In the following, we review the ideas of these paradigms and present common algorithms before examining their theoretical trade-offs in AL workflows.

Random sampling ignores all geometric and structural properties of the unlabeled pool, but follows the original data distribution $P(X)$. Despite its naive approach, random sampling serves as an important baseline, that is sometimes surprisingly hard to beat.

Space-filling or diversity-based methods aim at selecting samples that are as different as possible or cover as much of the input space as possible. Historically, this approach was used for Design of Experiment (DoE), which deals with a similar but slightly different problem than AL. In DoE, the task is to define a set of samples in a predefined space, which is then labeled, with the goal of learning as much as possible about the correlation between independent and dependent variables (e.g. by training an ML-model). Since in DoE typically no pool is available and the samples can be selected from anywhere, there are a few methods that adapt the idea of a diverse and space-filling initialization for the pool-based setting. First is *Greedy Sampling Input* (GSx) [23], which iteratively selects the unlabeled datapoint with the largest distance to its nearest already labeled neighbor. Second, developed for Deep AL, is *CoreSet* [18]. CoreSet approaches a space-filling data coverage by defining a hypersphere around every labeled sample. New samples are selected with the objective of minimizing the hypersphere-radius under the condition that all unlabeled samples are covered by a sphere.

Representativeness-based methods follow the idea that an initial set should cover the general structure of the whole dataset. They select samples that are representative for many other samples, by clustering or density estimation. Many methods belonging to this category are dedicated AL strategies and are not used exclusively as initialization strategy but to fully select all samples in their original work. In the following we shortly present the ideas behind important methods, that are also used for our benchmark. Cluster-based sampling [11] clusters the unlabeled samples and selects the data point closest to the centroid from each cluster. *Iterative Representativeness and Diversity Maximization* (iRDM) [16] additionally maximizes the distance to already selected points while maintaining close distances to the cluster-centers. *ProbCover* [24] maximizes data distribution coverage by greedily selecting data points whose fixed-radius neighborhoods (δ-balls) cover the largest possible portion of the unlabeled space. *TypiClust* [9] specifically targets the low-budget regime by

clustering the feature space and preferentially selecting typical, representative examples from dense data regions for annotation. Importantly, all methods either consider diversity implicitly by selecting only one sample per cluster or explicitly, like iRDM and ProbCover. This is necessary to avoid redundant queries from dense regions.

While the idea of a space-filling initial set may seem intuitively appealing, they have disadvantages when applied to unlabeled data-pools. By maximizing sample distances or coverage, these algorithms often query extreme values or isolated samples in low-density regions, which provide minimal informational value and may even degrade the initial model performance [21]. Furthermore, attempting to cover a multidimensional feature space with a constrained initial budget is often ill-posed, due to the curse of dimensionality.

Conversely, representativeness-based methods avoid outliers by targeting dense regions or clusters. However, they do not attempt to cover the whole input space, which can lead to potentially crucial information being missed in sparse regions. This establishes a fundamental trade-off, as each paradigm directly addresses the primary weakness of the other. Which of these fundamentally opposing approaches is better suited is also discussed conversely in other works. On one hand, theoretical frameworks for space-filling methods, like CoreSet, guarantee error reduction under increasing sample sizes [18]. On the other hand, for low budget scenarios (like the initialization), querying representative samples to learn the main characteristics of the data distribution is provably superior [9, 24]. The resolution to this apparent contradiction lies in the label budget, as the optimal choice shifts from representativeness in low-budget regimes to space-filling coverage as the sample size increases.

However, these theoretical proofs often rely on specific model choices and idealized assumptions, as is common in AL theory. This gap between theory and practice is further demonstrated by a comprehensive survey on initial pools in Deep AL, which revealed no consistent advantage for either paradigm [6]. Consequently, it remains to be seen whether these mathematical findings hold during empirical evaluation on real-world tabular data.

2.2 Self-supervised Learning

To construct high-quality initial sets, recent Deep AL research increasingly employs SSL methods. The primary objective of SSL is to learn a transformation of the features into an embedding space where geometric proximity reflects true semantic similarity better than the distances in original space. While most SSL frameworks are tailored for deep learning domains like computer vision and natural language processing, specialized pretext tasks have been developed to handle the heterogeneity and mixed datatype characteristic of tabular data. Tabular SSL architectures can be broadly categorized into three paradigms based on their learning objectives [22]:

Predictive Learning methods corrupt the input features, e.g. through random masking or additive noise, and train an encoder-decoder model to reconstruct the original data points. Typical architectures include VIME [25] and TabNet [1]. *Contrastive Learning* methods directly learn to rate the similarity between two inputs. For this they are trained to pull augmented views of the same instance closer together in the embedding space while pushing different instances apart. The augmented views are typically generated via subset feature masking or noise injection, as seen in SCARF [3] and STab [10]. *Hybrid methods* combine predictive reconstruction losses with contrastive similarity objectives to leverage the strengths of both paradigms.

When integrated into AL, SSL embeddings are used in two ways: First, initialization strategies can directly work in the embedding space. By using this potentially more meaningful representation, geometric measures can better identify representative or diverse samples during the initialization process. Second, instead of returning to the usage of the original features after the initialization for training the base model, the embedding learnt for the initial set selection can also be used as the input representation. This can improve the model performance and consequently yield better uncertainty estimates for selecting new samples.

3 Experimental Procedure

This section describes the datasets, the compared initialization and SSL methods alongside their implementation details, and the overall evaluation framework.

3.1 Datasets

Our benchmark builds upon the curated tabular dataset suite introduced by Grinsztajn et al. [8], comprising 23 binary, balanced classification tasks and 36 regression tasks. Most of these tasks come from real-world applications, with only a few of them being simulated. To ensure that the initial set selection meaningfully impacts the follow-up AL performance, we filtered the original suite according to three criteria: We exclude trivial tasks, that achieve an Accuracy > 0.85 or $R^2 > 0.90$ with a random 1% initial set. Furthermore, we exclude tasks failing to reach Accuracy > 0.50 or $R^2 > 0.50$ at the end of the full AL cycle, as the model misspecification over the full AL process masks the impact of different initialization methods. Finally, some tasks have a low number of unique samples ($n_{\text{unique}}/n_{\text{total}} < 0.50$). Because all methods (except random sampling) ignore identical points, this reduces the amount of available samples and therefore these tasks are also excluded.

This filtering results in a final benchmark suite of 40 diverse, real-world tabular datasets (19 classification, 21 regression). Missing values are not present, categorical features are one-hot encoded, and datasets exceeding 10,000 samples are randomly subsampled to 10,000. Subsampling seeds are identical across all initialization methods to guarantee fair comparisons. The number of features ranges from 5 to 613 (with only two tasks exceeding 100 features), providing a diverse coverage of real-world tabular datasets.

3.2 Initialization Methods and SSL Architectures

To systematically evaluate the impact of the different initialization paradigms, we select strategies from each category identified in Section 2 and pair methods developed for tabular data with methods from Deep AL initialization: Random

Sampling serves as the naive baseline. For the space-filling / diversity-based category we select GSx [23] (tabular-native) and CoreSet [18] (Deep AL). The representativeness-based methods are iRDM [16] (tabular-native) as a cluster-based method and ProbCover [24] and TypiClust [9] (Deep AL) as density-based methods.

All initialization strategies are integrated via the `scikit-activeml` framework [14], with a custom implementation added for iRDM. Hyperparameters follow the default values of the framework and original paper recommendations. To make ProbCover applicable to regression tasks the continuous targets are binned into two clusters, as the neighborhood radius δ is tuned via pseudo-labeling (via clustering with the number of clusters mirroring the number classes).

Model based AL Loop: Once the initial set is selected, subsequent AL iterations use *Query-By-Committee* (QBC) [20] with a Random Forest base model (50 trees, default `scikit-learn` parameters). The disagreement of the Committee is measured using vote-entropy for classification and empirical variance for regression.

SSL Methods: We compare two prominent SSL frameworks: *VIME* [25] (predictive learning) and *SCARF* [3] (contrastive learning). For both, we again follow the recommendations of the authors: VIME uses a feature corruption rate of 60% and an embedding-dimension equal to the input feature size. This is done to avoid memorization of the instances while reconstructing the corrupted versions. SCARF projects the features into a fixed 256-dimensional space. Memorization is less likely with SCARF as no reconstruction is done but similarity based on the InfoNCE loss [17] is learnt. Both architectures are trained using the Adam optimizer (learning rate 0.001, batch size 128) for up to 50 epochs with early stopping (patience of 5) on a 10% validation split.

3.3 Evaluation Process

The general settings for the benchmark are as follows: Following the selection of the initial set, QBC is used to sequentially query unlabeled samples until the labeled training set reaches 50% of the total dataset pool. To answer RQ2,

we evaluate three distinct initial set sizes: 1%, 10%, and 50%[1] of available instances. For each combination of task, method, and initial set size, we conduct 50 independent, reproducible runs. The performance of the model-based part of the AL process is evaluated every 50 iterations using a Random Forest model (100 trees) trained on the currently labeled subset and evaluated on a fixed 30% hold-out test set. To ensure comparability, the train-test split and all other random seed are identical across all initialization strategies within a given run. Additionally, we evaluate a fully random baseline where both the initial set creation and all subsequent query iterations are using uniform random sampling. We evaluate the *Root Mean Squared Error* (RMSE) for regression tasks and *Accuracy* for classification tasks.

Each initialization strategy is evaluated across three feature space settings: original input features, VIME-, and SCARF-embeddings. Combined with the three initial budget sizes, this yields $3 \times 3 = 9$ different experimental configurations per method and dataset. Crucially, SSL embeddings are used exclusively during the initial set selection phase and all subsequent model-based AL iterations and model evaluations operate only on the original features. This is done to focus the evaluation on the initial set creation and not distort it with a differences in the model-based part.

Because the performance metrics across the 40 tasks are influenced by different scales and performance ceilings (e.g., maximum accuracy varying from 60% to 90% depending on task difficulty), averaging the metrics across all datasets would introduce strong task-specific biases. To aggregate the performance fairly across datasets, we rank the different methods per task and run and then average these ranks across tasks. Alternatively, the relative improvements of each method to a baseline (e.g. fully random baseline) are averaged and compared.

To assess statistical significance and identify meaningful results, we apply a paired non-parametric Wilcoxon signed-rank test on the mean performance scores per dataset. To control the Family-Wise Error Rate across pairwise comparisons, p-values are adjusted using the Holm-Bonferroni post-hoc procedure at significance level $\alpha = 0.05$ [7]. Adjusted p-values below 0.05 indicate

[1] Since all of the budget is spent on the initial set no model-based AL is performed for this case.

a statistically significant, task-agnostic and consistent advantage for a given strategy.

4 Results

This section presents the results, structured by the RQs presented in the introduction. We explain why we present the results of this large benchmark in the way we do, analyze the results and use them to answer the RQs.

4.1 Comparative Performance of Initialization Methods (RQ1)

To answer RQ1, we evaluate the performance of initialization methods in two ways: the immediate quality of the selected initial set and its long-term impact on the subsequent model-based AL process. Here, we do not use SSL embeddings, as these are analyzed in RQ3.

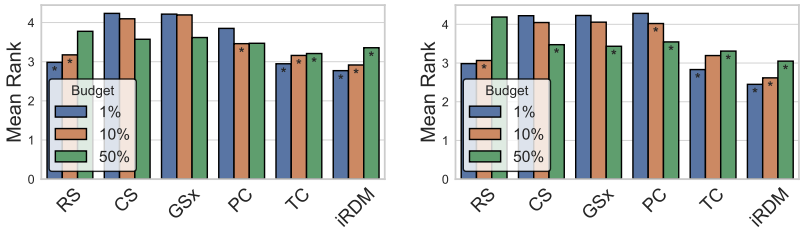
We first analyze the immediate quality of the initial sets. Figure 2 depicts the mean ranks of the compared methods. For each dataset, initial set size and run, the methods are ranked from 1 (best) to 6 (worst) based on their Accuracy (classification) or RMSE (regression) and averaged across all datasets within the respective task type. A clear pattern emerges for both classification (Figure 2a) and regression (Figure 2b): representativeness-based approaches, specifically iRDM and TypiClust, achieve significantly lower (better) mean ranks than all other methods. Conversely, diversity-based approaches (CoreSet and GSx) perform worst, being consistently outperformed even by Random Sampling. The only notable exceptions occur at the largest initial set size (50%), where performances become statistically indistinguishable for the regression tasks. Interestingly, ProbCover, which also falls under the representativeness-based paradigm, has an intermediate position. While its average rank is slightly better than diversity-based approaches in classification and similar to them for regression, it becomes statistically indistinguishable from the top-performing methods at larger initial set sizes (10% and 50%).

Because the AL process continues after initial set creation, we next investigate whether a better initial set provides a lasting advantage throughout the subsequent model-based AL process. To assess performance across the complete querying process rather than at a single iteration, we calculate the Area Under the Learning Curve (AULC) relative to the fully Random baseline (where all samples across all iterations are selected uniformly at random). Table 1 presents the performances relative to this baseline.

For classification tasks, the initial advantage of representativeness-based selection vanishes once the model-based AL phase begins. This can be seen by the similar and statistically indistinguishable AULC scores (except ProbCover) for the smaller initial set sizes (1% and 10%). The selection of the initial set only has a significant impact when 50% of the available samples are spent on initialization, as no subsequent model-based AL iterations follow.

In contrast, for regression tasks, initial set selection has a stronger influence on the model-based AL process. Random Sampling and ProbCover eventually catch up to iRDM and TypiClust, which is demonstrated by their similar AULCs. But both diversity-based methods (CoreSet and GSx) have a significantly higher (worse) AULC, which shows that their initial disadvantage remains present throughout the model-based AL phase. At a 10% initial set size, this effect is even more pronounced: only the active learner using the Random Sampling initialization catches up on the consistent performance of the iRDM initialization, whereas all other methods initial disadvantages stays permanently.

In summary regarding RQ1, representativeness-based initialization methods (iRDM and TypiClust) consistently outperform all other paradigms in terms of immediate initial set quality on tabular data. However, whether this initial advantage impacts the subsequent model-based AL process depends on the task type. While the advantage quickly vanishes for classification, it provides a persistent performance gain in regression tasks. Here, iRDM in particular stands out as the best and most influential strategy, making the overall best choice for both task types.



(a) Classification

(b) Regression

Figure 2: Average performance ranks of initialization methods (1 = best, 6 = worst) for initial set sizes of 1%, 10%, and 50%. The * denotes the best method and all statistically indistinguishable methods per budget. Abbreviations: Random Sampling (RS), CoreSet (CS), ProbCover (PC), TypiClust (TC).

Table 1: AULC for Classification (accuracy - higher is better) and Regression (RMSE - lower is better) relative to the fully random baseline. Bold entries show the best method, while the * denotes all methods that are not significantly worse than the best one.

	Classification (Budget)			Regression (Budget)		
Init. Method	1%	10%	50%	1%	10%	50%
Random	1.016*	1.013*	1.000	0.968*	**0.954***	1.000
CoreSet	1.016*	1.014*	1.001	0.985	0.970	0.992*
GSx	1.016*	1.013*	1.001	0.984	0.970	0.992*
ProbCover	1.015	1.012	1.001	0.973*	0.960	0.997*
TypiClust	**1.016***	1.014	**1.003***	0.968*	0.958	0.990*
iRDM	1.016*	**1.014***	1.002*	**0.968***	0.956*	**0.988***

4.2 Effects of Different Initialization Budgets (RQ2)

To answer RQ2, we investigate what proportion of the available pool should be used for the initial set selection. To evaluate this, Figure 3 depicts the Top-33% rate of all method-budget combinations. To prevent misinterpretation of the results, two distinct percentage values must be distinguished:

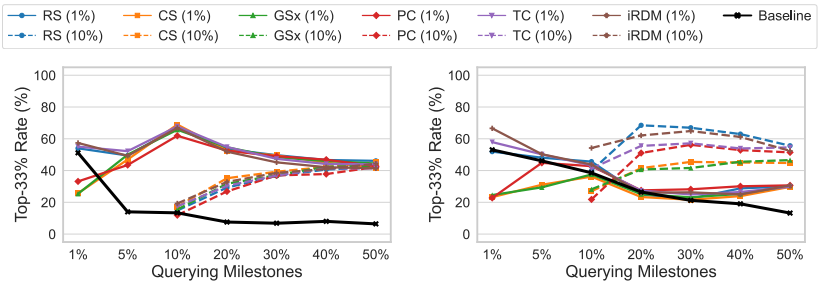
1. **Initial set size:** The proportion of the total pool spent exclusively on the initialization method (1%, 10%, or 50%).

2. **Querying milestones:** The percentage of the labeled samples of the pool so far (1%, 5%, 10% to 50% in steps of ten).

At each querying milestone, we rank all unique method and initial set size combinations whose initial set size does not exceed the current milestone. For instance, at the 5% milestone, only combinations with a 1% initial set size are ranked (6 combinations), whereas at the 20% milestone, combinations with both 1% and 10% initial set sizes are ranked simultaneously (12 combinations). Figure 3 plots the percentage of datasets in which a specific combination ranks within the top third (upper 33%) of all active combinations.

The top-33% rate trajectories reveal different behaviors for classification and regression tasks once multiple initial set sizes can be compared past the 10% milestone: For classification (Figure 3a), smaller initial set sizes (1%) yield a clear advantage at early milestones ($\leq$ 20%). At the 10% milestone, all combinations using a 1% initial set size have a Top-33% rate of 60-70%, while the combinations using a 10% initial set size have a rate of only 10-20%. However, for the 30-50% milestones, this initial advantage diminishes, and the choice of initial set size becomes less influential, with all combinations having a Top-33% rate of 40-45%

For regression (Figure 3b), different observations can be made. While 10% initial set size combinations already partially outperform their 1% counterparts at the 10% milestone, they all surpass their counterparts by the 20% milestone. This advantage holds up to the 50% milestone, proving that AL for regression profits from broader initial data coverage.



(a) Classification

(b) Regression

Figure 3: Percentage of tasks where a specific combination of initialization method and initial set size ranks among the Top-33% of all active combinations at each querying milestone. Continuous lines represent the combinations that use a 1% initial set size, dashed lines the combinations that use a 10% initial set size. Abbreviations: Random Sampling (RS), CoreSet (CS), Greedy Sampling Input (GSx), ProbCover (PC), TypiClust (TC).

To fortify these observations with statistical significance and compare the case of spending the entire budget on initialization (50% initial set size), Table 2 presents the relative performance improvements of larger initial set sizes over smaller ones.

The statistical tests confirm the visual findings from Figure 3: For classification, allocating a 10% initial set size causes a statistically significant decrease in accuracy at early milestones compared to a 1% initial set size. At later milestones, performance differences become statistically insignificant across all methods. This is again different for regression: While a 1% initial set size appears superior at early stages (e.g. up to 17% improvement for GSx), these differences are not statistically significant. From the 30% milestone onward, however, a 10% initial set size provides statistically significant performance gains over a 1% initial set size across nearly all methods. The combinations using 50% initial set size (no model-based AL) demonstrate a statistically significant degradation in performance across both task types compared to both 1% and 10% initial set sizes. This confirms and highlights the importance of model-based AL.

In conclusion for RQ2, the optimal initial set size is highly dependent on the task type. For classification tasks, a very small initial set size (1%) should be

Table 2: Relative improvement of larger initial budgets in %. Results are given as improvements for Classifcation / Regression (`accuracy` / `rmse`). Bold values indicate statistically significant differences ($p_{adj} < 0.05$, Holm-Wilcoxon). Abbreviations: Milestone (MS), otherwise the same as for previous Figures.

MS	RS	CS	GSx	PC	TC	iRDM
Improvement of Initial Set Size 10% over 1%						
10%	**-1.6** / -3.1	**-3.6** / -15.1	**-3.7** / -17.0	**-1.7** / -5.5	**-1.5** / -5.1	**-1.4** / -3.8
20%	**-0.4** / **+2.0**	-0.2 / +0.5	**-0.3** / +0.4	**-0.4** / +1.1	**-0.3** / +1.3	**-0.2** / +1.5
30%	-0.1 / **+1.5**	-0.0 / **+0.9**	-0.1 / **+0.8**	**-0.2** / **+1.2**	-0.1 / **+1.2**	-0.1 / **+1.4**
40%	-0.0 / **+0.9**	+0.0 / **+0.6**	-0.0 / **+0.6**	-0.1 / **+0.8**	-0.0 / **+0.7**	+0.0 / **+0.9**
50%	-0.0 / **+0.5**	+0.1 / +0.3	+0.0 / **+0.3**	-0.1 / **+0.5**	-0.0 / **+0.4**	+0.0 / **+0.4**
Improvement of Initial Set Size 50% over 1%						
50%	**-1.4** / **-3.2**	**-1.2** / **-2.4**	**-1.3** / **-2.4**	**-1.2** / **-2.9**	**-1.1** / **-2.2**	**-1.1** / **-2.0**
Improvement of Initial Set Size 50% over 10%						
50%	**-1.3** / **-3.8**	**-1.3** / **-2.7**	**-1.3** / **-2.7**	**-1.2** / **-3.4**	**-1.1** / **-2.6**	**-1.1** / **-2.4**

preferred. This holds particularly for low total labeling budgets, as the early model-based AL feedback is crucial for successful AL. For regression tasks, allocating a larger initial set size (10%) is recommended. Establishing a broader initial representation allows the base model to produce more reliable uncertainty estimates during subsequent iterations, which ultimately leads to higher overall AL performance.

4.3 Impact of SSL Pretext Representations (RQ3)

RQ3 investigates whether transforming tabular features into SSL embeddings improves the quality of the initial sets. We first evaluate whether SSL pretext tasks alter the geometric structure of the feature space at all, because the original feature space often already presents semantically similarities by spatial proximity, which potentially renders the embedding useless. To keep computational costs manageable while covering all paradigms, we restrict this analysis to three representative initialization methods: CoreSet (representing space-filling approaches, as GSx yields near-identical performance), iRDM (representing top-performing cluster-based representativeness), and ProbCover (representing density-based coverage). Uniform Random Sampling is excluded since its selection is feature-agnostic.

Figure 4 displays the Jaccard similarity between initial sets selected on the three feature representations (original features versus VIME embedding versus SCARF embedding). All methods have low overlap, confirming that SSL pretext tasks construct newly structured feature spaces. ProbCover exhibits the lowest Jaccard similarity, which implies that SSL embeddings change local neighborhoods which impacts the local density estimations. In contrast, CoreSet exhibits a higher overlap, indicating that boundary points, which are often selected by diversity-based methods, in the original feature space remain boundary points within the embedding. Overall, VIME and SCARF induce similarly strong structural transformations, and their resulting embeddings also differ substantially from each other (as shown in the last columns (VIME vs. SCARF).

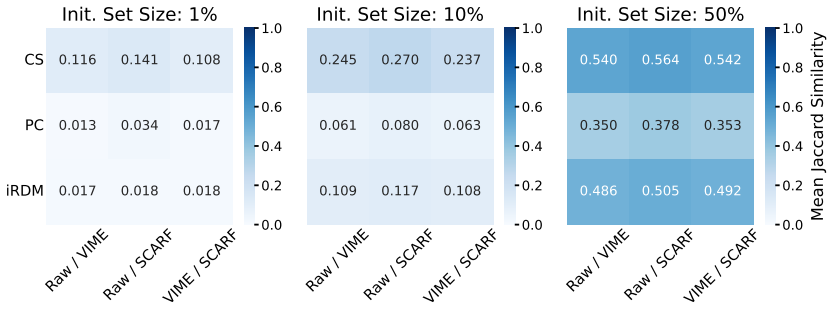
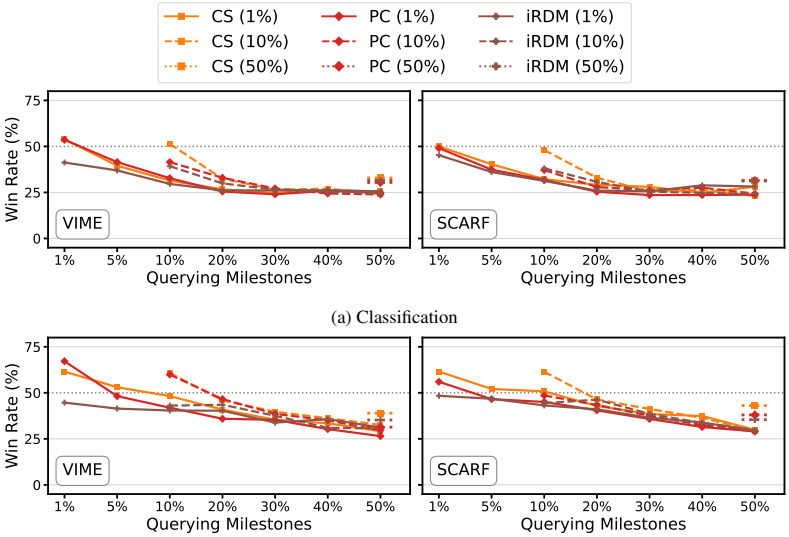


Figure 4: Jaccard similarity of selected initial sets between original features (Raw) and SSL embeddings (VIME and SCARF) for all compared methods and across initial set sizes. Excpected Jaccard similarity for two randomly drawn sets: set size 1% $\rightarrow$ 0.005, 10% $\rightarrow$ 0.053, 50% $\rightarrow$ 0.333. Abbreviations: CoreSet (CS), ProbCover (PC).

Having established that SSL embeddings alter initial set selection, we next evaluate whether this restructuring also leads to superior AL performance. Figure 5 presents pairwise win rates at different querying milestones of the AL process, comparing each method-budget combination directly against its identical counterpart operating on the original features. A 50% win rate indicates performance parity, whereas values above 50% denote an advantage for SSL embeddings.

The pairwise evaluation reveals a largely positive initial effect followed by a performance collapse during the model-based AL part. The initial win rate increase is less notable for classification tasks (Figure 5a) than for regression tasks (Figure 5b), where win rates reach up to 70%, and more prominent when employing VIME embeddings instead of SCARF embeddings. However, it is important to note, that this relative gain is limited to methods that performed poorly on original features, such as CoreSet and ProbCover. For the top-performing strategy on the original features, iRDM, SSL embeddings yield no initial advantage, with win rates remaining slightly below 50%. As soon as the model-based AL process begins, any initial advantage rapidly vanishes. Across all task types, methods, initial set sizes, and SSL architectures, the win rates decline, eventually converging to approximately 25%. At the largest initial set size (50%), SSL-guided selections underperform the selections on original features with win rates between 30 and 40%.



(a) Classification

(b) Regression

Figure 5: Pairwise win rates of SSL-guided initializations (VIME and SCARF) against their respective counterparts on original features across querying milestones. The 50% dashed line represents equal performance. Continuous lines represent the combinations that use a 1% initial set size, dashed lines the use of a 10% initial set size and dotted lines the use of a 50% initial set size. Abbreviations: Random Sampling (RS), CoreSet (CS), Greedy Sampling Input (GSx), ProbCover (PC), TypiClust (TC).

These findings provide a clear answer to RQ3: leveraging SSL pretext tasks for the selection of the initial set is not recommended for tabular AL. While SSL embeddings may offer marginal improvements for weaker methods directly after the initialization, they hinder the subsequent model-based AL process. This can likely be attributed to the representation mismatch between selection and base model training. Potentially, SSL embeddings encourage selection from localized, complex regions over a broad global coverage, which can boost the initial performance, but causes miscalibrated uncertainty estimates. However, validating this hypothesis would require a follow-up study executing the entire AL loop within the embedding space.

Additionally, the answer to RQ3 is limited in another way. We evaluated two basic, untuned SSL frameworks (VIME and SCARF). While extensive

hyperparameter tuning is often not possible in real-world AL scenarios, which makes default configurations a realistic baseline, exploring whether task-specific or fine-tuned SSL architectures could yield better representations remains an open question. Nevertheless, within the tabular AL setting evaluated here, the original feature space remains the superior basis for initial set selection.

5 Conclusion

In this paper, we presented a comprehensive benchmark evaluating initialization strategies for AL on tabular data across 40 diverse classification and regression tasks. By systematically isolating the effects of different initialization paradigms, initial set sizes, and SSL representations, our work provides clear empirical insights into the cold-start problem in tabular AL.

We find that representativeness-based methods are generally preferable over diversity-based and random approaches, with iRDM emerging as a strong default candidate for both classification and regression tasks. In line with the findings of [13], we observe that a larger initial set size is beneficial for regression (e.g., 10%), whereas for classification tasks, it is advisable to only spend a small budget on initialization (e.g., 1%) to quickly transition to model-based AL. First insights into the use of SSL embeddings demonstrate a potential initial advantage that, however, cannot be sustained during the subsequent model-based process. Therefore, without further more detailed research, we cannot recommend the use of SSL pretext tasks.

Future work should explore whether end-to-end embedding-based AL loops can eliminate the observed performance drop, as well as adaptive budget strategies that dynamically adjust initial set sizes based on dataset properties.

Acknowledgements

The author acknowledges financial support by the project “SAIL: SustAInable Life cycle of Intelligent Socio Technical Systems” (Grant ID NW21 059B),

which is funded by the program “Netzwerke 2021” of the Ministry of Culture and Science of the State of Northrhine Westphalia, Germany.

References

[1] Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:6679–6687, 5 2021.

[2] Josh Attenberg and Foster Provost. Inactive learning? difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12:36–41, 3 2011.

[3] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. SCARF: Self-supervised contrastive learning using random feature corruption. In *International Conference on Learning Representations (ICLR)*, 2022.

[4] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024.

[5] Robert Burbidge, Jem J. Rowland, and Ross D. King. Active learning for regression based on query by committee. In *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, pages 209–218, 2007.

[6] Akshay L. Chandra, Sai Vikas Desai, Chaitanya Devaguptapu, and Vineeth N. Balasubramanian. On initial pools for deep active learning. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, volume 148, pages 14–32. PMLR, 11 Dec 2021.

[7] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[8] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on tabular data? In

Proceedings of the 36th International Conference on Neural Information Processing Systems, pages 507–520. Curran Associates Inc., 7 2022.

[9] Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 8175–8195. PMLR, 7 2022.

[10] Ehsan Hajiramezanali, Nathaniel Lee Diamant, Gabriele Scalia, and Max W Shen. STab: Self-supervised learning for tabular data. In *NeurIPS 2022 First Table Representation Workshop*, 2022.

[11] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. In *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference*. AAAI Press, 5 2010.

[12] Bjarne Jaster and Martin Kohlhase. Trust issues in active learning and their impact on real-world applications. In *2025 IEEE Symposium on Trustworthy, Explainable and Responsible Computational Intelligence (CITREx Companion)*, pages 1–5. IEEE, 3 2025.

[13] Bjarne Jaster, Alaa Tharwat, Eiram Mahera Sheikh, Martin Kohlhase, and Wolfram Schenck. Low query budget active learning for classification and regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2025*, pages 5–21. Springer Nature Switzerland, 2026.

[14] Daniel Kottke, Marek Herde, Tuan Pham Minh, Alexander Benz, Pascal Mergard, Atal Roghman, Christoph Sandrock, and Bernhard Sick. scikit-activeml: A library and toolbox for active learning algorithms, 3 2021.

[15] David D. Lewis and William A. Gale. *A Sequential Algorithm for Training Text Classifiers*, pages 3–12. Springer London, 1994. Uncertainty Sampling.

[16] Ziang Liu, Xue Jiang, Hanbin Luo, Weili Fang, Jiajing Liu, and Dongrui Wu. Pool-based unsupervised active learning for regression using iterative

representativeness-diversity maximization (irdm). *Pattern Recognition Letters*, 142:11–19, 2 2021.

[17] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[18] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, 2008.

[20] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 7 1992.

[21] Alaa Tharwat and Wolfram Schenck. Active learning for handling missing data. *IEEE Transactions on Neural Networks and Learning Systems*, 36:3273–3287, 2 2025.

[22] Wei-Yao Wang, Wei-Wei Du, Derek Xu, Wei Wang, and Wen-Chih Peng. A survey on self-supervised learning for non-sequential tabular data. *Machine Learning*, 114:16, 1 2025.

[23] Dongrui Wu, Chin-Teng Lin, and Jian Huang. Active learning for regression using greedy sampling. *Information Sciences*, 474:90–105, 2 2019.

[24] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[25] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self- and semi-supervised learning to tabular domain. In *Advances in Neural Information Processing Systems*, volume 33, pages 11033–11043. Curran Associates, Inc., 2020.

Foundation Models Versus Machine-Learning Algorithms: Results from a Load-Forecasting Challenge Under EU AI Act Requirements

Thomas Bartz-Beielstein, Marcel Samuel Klimm, Alexander Hinterleitner, Alex Prediger

Institute for Data Science, Engineering, and Analytics (IDE+A), TH Köln
Steinmüllerallee 1, 51643 Gummersbach, Germany
E-Mail: thomas.bartz-beielstein@th-koeln.de

Abstract

Short-term load forecasting (STLF) is essential for predicting the electricity demand of the next hours to days. It is currently under discussion which components of the electricity grid are considered critical infrastructure under the European Union's AI Act (AIA). Even if STLF is not a high-risk AI system, we claim that STLF is no longer a pure accuracy problem: determinism, fail-safe behaviour, a minimal attack surface, and full auditability should become standard software-engineering requirements. This paper presents results from a 41-day live challenge held at TH Köln from 10 June to 20 July 2026, in which eleven student teams and eight organiser-operated reference models forecast the 24 hourly values of the aggregated German transmission-grid load one day ahead. It shows how AIA requirements can be mapped onto code-development and process rules of an open-source forecasting pipeline. It also evaluates whether such a pipeline can compete with the official day-ahead forecast of ENTSO-E, and compares small, local models with the pretrained foundation model Chronos-2. The rule-compliant reference pipeline, a deterministic recursive LightGBM forecaster built on `spotforecast2-safe` and tuned by `spotoptim`, reduced the mean absolute error (MAE) of the official ENTSO-E forecast by 34.7 % on the jointly evaluated days ($p = 0.0001$). Furthermore, ten of the eleven student

teams were more accurate than ENTSO-E. Transparent local models (MACL2L) led the leaderboard and were at least competitive with Chronos-2.

Keywords: short-term load forecasting, EU AI Act, foundation models, in-context learning, gradient boosting, hyperparameter tuning, reproducibility

1 Introduction

Short-term load forecasting (STLF) is a core task in the operation of electricity grids [1, 2]. Two recent developments add to this classical problem. First, the methodological landscape has widened: next to statistical models and time-series forecasting with machine learning (TSFML) [3, 4], in which gradient-boosted decision trees such as LightGBM [5] dominate many competitions [6–8] and are competitive with deep-learning architectures [9], pretrained time-series *foundation models* such as Chronos-2 [10] and tabular *in-context* learners [11] promise accurate forecasts without task-specific training. Second, the regulatory landscape has changed. The EU Artificial Intelligence Act (AIA) [12] classifies AI systems used as safety components in the supply of electricity as high-risk, and the NIS-2 Directive [13], the Cyber Resilience Act (CRA) [14], and the revised Product Liability Directive [15] add cybersecurity and liability obligations for operators and suppliers of software in critical infrastructure. Under these recent developments, the role of determinism, fail-safe behaviour, minimal attack surface, and auditability should be newly evaluated. We claim that these properties gain importance and are no longer nice-to-have features but software-engineering requirements.

These two developments pull in different directions. Huge, pretrained foundation models import weights whose training data and procedure lie outside the operator's audit boundary. They are black boxes, which typically require substantial computational resources. Classical, deterministic, locally trained pipelines are auditable by construction, but it is unclear whether they remain competitive. We address this conflict empirically with three research questions (RQ):

- *RQ1:* How can EU AI Act requirements be implemented in a STLF pipeline?
- *RQ2:* Can an open-source pipeline developed under these requirements compete with the official day-ahead forecast of the European Network of Transmission System Operators for Electricity (ENTSO-E)?
- *RQ3:* Can small, local models compete with large pretrained foundation models?

To answer these questions, we conducted a comprehensive empirical study under real operating conditions: a 41-day live forecasting challenge was held at TH Köln in the summer term of 2026 [16]. Eleven student teams and eight organiser-operated reference models submitted daily forecasts of the 24 hourly values of the aggregated German transmission-grid load on the basis of the ENTSO-E Transparency Platform [17, 18]. To ensure replicability, every team result was independently reproduced by another team, and model cards and security scorecards were mandatory.

The contributions of this work are (i) a mapping of AIA requirements onto verifiable code-development and process rules of the open-source forecasting library `spotforecast2-safe` [19], (ii) a statistically controlled comparison of classical machine learning methods (recursive gradient boosting), student-developed pipelines, a pretrained foundation model, and local models developed by an industrial partner against the operational ENTSO-E forecast, and (iii) a reproducible evaluation protocol in which every result is recomputed from the published challenge data. The next section derives the requirements. The following sections describe the forecasting pipeline and the competing models, the challenge design and the evaluation protocol, and the results, which are then discussed and summarised.

2 Regulatory Requirements for Load Forecasting

Whether a day-ahead forecaster of the German transmission-grid load *is* a high-risk AI system in the sense of the AIA is currently under discussion and not obvious. The AIA ties its notion of critical infrastructure to the annex of the Critical Entities Resilience (CER) Directive [20] (Recital 55) and considers

an energy-related AI system high-risk only when it acts as a *safety component* (Annex III, point 2) [12]. An STLF pipeline that supports scheduling and trading decisions is located *alongside* the safety perimeter of the grid rather than *inside* it [21]. It is, however, an IT system of a critical-installation operator and thus subject to the German BSI-Gesetz, which transposes NIS-2 [22]. Once the forecast feeds automated decisions, or once the software is integrated into a commercial product that falls under the CRA and the revised Product Liability Directive, the requirements tighten considerably. We therefore adopt the high-risk requirements voluntarily.

Table 1 summarises our proposals to answer RQ1. The relevant AIA articles for high-risk systems demand data governance (Art. 10), technical documentation (Art. 11), automatic record-keeping (Art. 12), transparency towards deployers (Art. 13), human oversight (Art. 14), and accuracy, robustness, and cybersecurity (Art. 15). The open-source software library `spotforecast2-safe` [19] is developed under the eight rules from Table 1 that translate these abstract obligations into verifiable properties. Four code-development rules (CR-1 to CR-4) restrict what the source code may contain and are enforced at commit time. Four process rules (PR-1 to PR-4) restrict how the package is developed, shipped, and operated. They are applied at review or release time, for example by continuous-integration workflows on GitHub Actions or GitLab CI/CD. We extended the process rules so that they can be run locally on a machine that does not depend on external services such as GitHub. One important design decision of the pipeline follows directly from these rules: *gaps and corruptions are annotated, not hidden* (Art. 10, CR-3). A flagged value may be repaired to complete a series, but it is never trusted as training evidence.

3 Forecasting Pipeline and Competing Models

3.1 Data and Target

The target is the ENTSO-E *Actual Total Load* of the German country aggregation "DE", measured in megawatts at hourly resolution after aggregation of the native 15-minute values [17]. A forecast consists of the 24 hourly values

$\hat{y}_{T+1|T}, \ldots, \hat{y}_{T+24|T}$ of the target day, issued at forecast origin T on the preceding day. The official ENTSO-E *day-ahead total load forecast*, referred to as the *ENTSO-E baseline* in the following, competes on the leaderboard alongside two seasonal-naive benchmarks that repeat the load of the previous day (seasonal period $m = 24$ h) or week ($m = 168$ h). Möbius et al. [23] reported systematic under-prediction and autocorrelated errors of the ENTSO-E forecast for the German–Luxembourg bidding zone from 2016 to 2019. In our study, only a weakened version of this pattern remains.

3.2 Reference Pipeline: Recursive LightGBM

Every team received a reference pipeline built on `spotforecast2-safe` [19, 24]. The pipeline is based on a similar architecture as described by Chagnet [18]. Its data preparation has three stages. (1) *Target-corruption checks* on the 15-minute data test the intra-hour range, reject implausible jumps between consecutive slots, and flag excessive shortfalls below a reference series, namely the ENTSO-E baseline. A configurable policy then aborts the run (the default), marks the slots for repair, or truncates the corrupted tail. (2) *Anomaly flagging* on the hourly series combines hard plausibility bounds, which set values to missing, with an advisory Isolation Forest [25] that logs suspicious points only. (3) *Gap imputation* completes the series by nearest-neighbour copying or linear interpolation. It assigns a training weight of zero to every imputed hour and to all subsequent hours whose lag features look back into the gap. Gaps at the recent end of the series are never filled. The series is truncated to the last observed hour instead.

The exogenous feature vector $\mathbf{x}_t$ contains 30 columns: sine/cosine encodings of hour, weekday, week of year, month, sunrise, and sunset; six integer holiday and day-type indicators derived from the German holiday calendar, including bridging days (“Brückentage”); and twelve weather variables from Open-Meteo [26], taken from the archive for history and from the numerical weather forecast for the target day. Every covariate must be known, or itself forecast, over the whole horizon, because $\mathbf{x}_{T+h}$ is needed at every step $h = 1, \ldots, 24$.

The forecaster reduces the multi-step problem to tabular regression, following the recursive strategy of `skforecast` [27], which `spotforecast2-safe` vendors as a reviewed subset. A LightGBM regressor f [5] is trained on lagged loads $y_{t-\ell}$ for the lags ℓ of a lag set $\mathcal{L}$, a rolling mean, and $\mathbf{x}_t$. The 24 hours are then predicted recursively [28],

$$\hat{y}_{T+h|T} = f\big(\tilde{y}_{T+h-\ell},\ \ell \in \mathcal{L};\ \mathbf{x}_{T+h}\big), \qquad \tilde{y}_s = \begin{cases} y_s, & s \le T, \\ \hat{y}_{s|T}, & s > T, \end{cases} \tag{1}$$

so that each prediction is fed back as an input for the following hours. The model is refitted daily on a three-year window with a fixed seed. XGBoost [29] and CatBoost [30] variants were operated as additional reference entries.

3.3 Hyperparameter Tuning

Eight LightGBM hyperparameters (number of leaves, maximum depth, learning rate in $[10^{-4}, 0.3]$, number of trees in $[10, 4000]$, bagging and feature fractions, and L_1/L_2 regularisation) and, as a ninth categorical dimension, the lag set $\mathcal{L}$ (six candidates between $\{1,\dots,24\}$ and a set including weekly lags 167 and 168) span the search space Θ. The hyperparameter vector $\boldsymbol{\theta}$ is tuned by minimising the cross-validated mean absolute error (MAE),

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} \ \text{CV-MAE}(\boldsymbol{\theta}), \tag{2}$$

where CV-MAE is estimated by ten-fold rolling-origin cross-validation (CV) [31, 32] with 24-hour blocks and weekly refits on the training window. The primary tuner is the surrogate-model based optimiser `spotoptim` [33], a Python implementation of sequential parameter optimisation [34], with an initial design of 50 and a total budget of 200 evaluations. The resulting entry is listed as *spotoptim lgbm*. As a comparison, the identical search space and budget were handed to the hyperparameter optimiser Optuna [35] (*optuna lgbm*).

3.4 Foundation and Local Models

Two families of models that perform no gradient training at submission time were operated by the organisers.

Chronos-2 [10, 36] is a pretrained time-series foundation model with more than 100 million parameters. It was applied zero-shot and univariately: the actual load series is its only input, with no covariates, no calendar features, and no training or tuning. From a context of the trailing 180 days it emits quantiles of the next 24 hours in one direct multi-step pass from which the 0.5 quantile was submitted as the point forecast.

MACL2L (Mac Learning to Learn) denotes small, locally trained models that run efficiently and do not require high-performance hardware. At submission time, the model conditions in-context on the most recent 540 to 600 day-rows of the data without updating its weights. Its output is averaged with that of a ridge regression fitted on the same feature rows and passed through a median safety guard. Two variants differ in exactly one input: *MACL2L* uses renewable-generation forecasts and the day-ahead price as covariates, whereas *MACL2L (ENTSO-E)* additionally receives the ENTSO-E baseline as a fifth covariate, which enters as a feature column, through the ridge regression, and as the third voter of the median guard. It therefore carries the `entsoe` label required by the rules.

3.5 Student Team Pipelines

All teams started from the reference pipeline provided by the organiser. The teams' own accounts of their forecasters are collected in the challenge report [21]. The teams were allowed to submit that configuration unchanged, extend it with further covariates, a different regressor, or a different tuning strategy, or replace it entirely with a forecaster of its own design. Whichever approach a team selected, the code-development rules CR-1 to CR-4 remain mandatory, and the process rules PR-1 to PR-4 governed how a package was developed and shipped. The best student team, *Hot Rod*, replaced the recursive strategy by a direct 24-hour LightGBM ensemble of five seeds while retaining

`spotforecast2-safe` for data access and preprocessing. It was trained on 2018–2019 and 2024–2026 data (excluding the pandemic and energy-crisis years), with population-weighted weather from six German regions averaged over three numerical weather prediction products and a nowcaster that fills only those short lags which were not yet published at submission time. Changes were adopted only if a live-replica backtest showed a paired MAE reduction whose block-bootstrap 95 % interval excluded zero, with no regression in the worst ten days or the worst rolling seven-day window. *Team Neura* kept the recursive LightGBM of `spotforecast2-safe` but used 168 hourly lags and four-city weather features with manually set hyperparameters.

4 Challenge Design and Evaluation Protocol

The challenge ran as a public leaderboard repository [16]. For every target day, the 24 forecasts were due at 23:59 Coordinated Universal Time (UTC) of the preceding day. Each day was scored after ENTSO-E published the actual load, and scores were recomputed whenever ENTSO-E revised the actuals within a 21-day look-back window. A team that missed a day had its most recent submission carried forward, and each team held one joker to replace one already-scored day. After a preliminary testing phase, the live phase comprised the 41 target days from 10 June to 20 July 2026. Besides the 11 student teams from two courses, 8 reference models operated by the organisers (the `spotoptim`- and Optuna-tuned LightGBM, XGBoost, CatBoost, and a causal variant of the reference pipeline, Chronos-2, and the two MACL2L variants), the ENTSO-E baseline, and the two seasonal naives were scored. Reference models joined the live phase on different dates.

Every team had to publish a *model card* documenting data, features, model, and known limitations. Furthermore, every team's result had to be *reproduced independently* by another team. This peer reproduction was certified for all eleven teams.

Metrics. For each target day, the MAE, root mean squared error (RMSE), mean absolute percentage error (MAPE), mean bias $\bar{e} = \frac{1}{24}\sum_h (\hat{y}_{T+h|T} - y_{T+h})$, and under-prediction rate (UPR, share of hours with $\hat{y} < y$) are computed over the

24 hours. The mean absolute scaled error (MASE) [37] divides the daily MAE by the mean absolute one-step change of the load before the target day. Entries are ranked by their mean daily MAE, and the number of scored days breaks ties.

Statistical testing. Because entries cover different days, mean scores are not directly comparable [38]. Pairs of entries are therefore compared by two-sided paired t-tests on the daily MAE of their shared days. Each daily MAE is itself a mean of 24 hourly errors. Samples comprise 26 to 41 days, so the normality assumption is unproblematic. For the student field, a repeated-measures analysis of variance (ANOVA), blocked by target day, is followed by all pairwise paired t-tests with Holm–Bonferroni correction.

Reproducibility. The analysis of the following section is computed from the frozen data bundle of the companion report [21], which contains the hourly matrix of every scored forecast, the ground truth, and the ENTSO-E baseline.

5 Results

5.1 Leaderboard

Table 2 lists the final leaderboard of the 20 ranked entries. Figure 1 visualises the mean MAE per entry. Every entry beat the weekly seasonal naive (2502 MW) on its scored days. The daily naive trails the field by a wide margin (3995 MW), so the challenge separated forecasting skill from persistence. Ten of the 11 student teams were more accurate than the ENTSO-E baseline. The best full-coverage entry was the student team Hot Rod with an MAE of 1228 MW and a MAPE of 2.42 % over all 41 days, 43.0 % below the ENTSO-E baseline (2156 MW and 4.34 %). Only the two MACL2L variants, which joined the live phase later and cover 26 and 29 days, ranked higher. The spotoptim-lgbm reference pipeline reached rank 6 with 1369 MW over 35 days, directly behind Chronos-2 and its Optuna-tuned sibling.

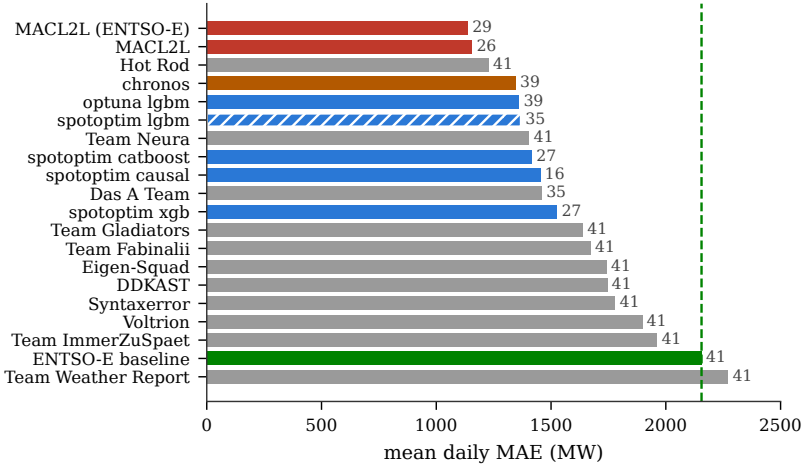


Figure 1: Mean MAE per entry on the final leaderboard: student teams (grey), the EU-AI-Act-compliant reference pipelines (blue, *spotoptim lgbm* hatched), the foundation model Chronos-2 (orange), the ENTSO-E baseline (green, dashed line), and the local in-context models MACL2L (red), which lead the field. Numbers give the scored days.

5.2 Reference Pipeline Versus the ENTSO-E Baseline (RQ2)

On the 35 days scored for both, the spotoptim-lgbm pipeline was more accurate than the ENTSO-E baseline on 26 days, with mean MAE 1369 versus 2098 MW, a reduction of 34.7 % ($p = 0.0001$). The Optuna-tuned sibling was statistically indistinguishable (1369 versus 1372 MW, $p = 0.98$), so the choice of tuner did not matter within the daily budget. Hot Rod's nominal advantage over the reference pipeline on their shared days (1217 versus 1369 MW) is not significant ($p = 0.13$).

The improvement does not stem from removing a constant offset. Over all 41 days, the ENTSO-E baseline's mean bias is only -38.4 MW, nearly two orders of magnitude below its MAE. Figure 2 (top) shows instead that the baseline's mean error follows a pronounced daily pattern, from an under-forecast of up to -1.8 GW at night to an over-forecast of up to 0.9 GW during the morning ramp, a weaker form of the pattern reported by Möbius et al. [23]. The reference pipeline keeps the mean error of every hour between -812 and 393 MW. Its

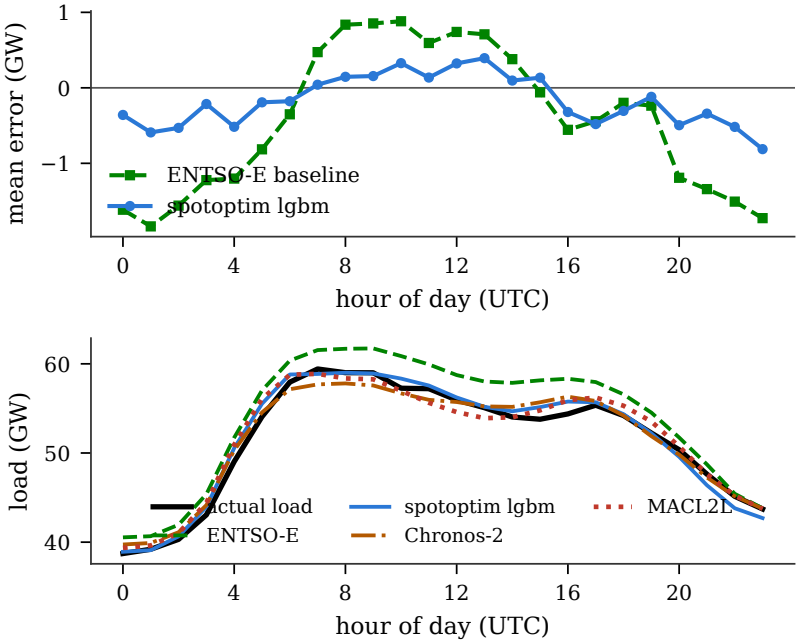


Figure 2: Top: mean hourly forecast error (forecast minus actual) by hour of day over the 35 days scored for both the spotoptim-lgbm pipeline (blue) and the ENTSO-E baseline (green dashed); negative values are under-forecasts. Bottom: forecasts and measured load (black) on 20 July 2026, the final challenge day, including the foundation model Chronos-2 (dark orange, dash-dotted) and the local in-context model MACL2L (red, dotted). Hours in UTC.

residual bias of -176 MW (UPR 56.0 %) is a mild but persistent under-forecast, which the MAE-only ranking gives no incentive to remove.

5.3 Statistical Significance and Rank Stability

The complete panel of all 11 student teams, the ENTSO-E baseline, and both naives covers the 35 days scored for every member. The repeated-measures ANOVA rejects the equality of the 14 entries decisively ($F(13, 442) = 9.21$, $p < 10^{-16}$). Of the 91 Holm-adjusted pairwise comparisons, the following are

relevant. Three student teams, Das A Team, Hot Rod, and Team Neura, are significantly more accurate than the ENTSO-E baseline (adjusted $p < 10^{-4}$ for Hot Rod, $p = 0.0017$ for Team Neura, $p = 0.016$ for Das A Team). Seven of the remaining eight student teams are nominally better, but do not separate from the ENTSO-E baseline after correction, and the baseline itself is indistinguishable from the weekly naive at the day level ($p = 1.0$). With 35 days, the field separates at its ends but not between neighbours, because of the small sample size. This effect is described by Hewamalage et al. [38].

Although only the MAE determines the official ranking, re-ranking all entries by RMSE, MAPE, and MASE changes little: Kendall's τ between the MAE ranking and the three alternatives is 0.979, 0.968, and 1.000, and every movement is a swap of adjacent or near-adjacent entries. The ranking is therefore robust to the choice of metric.

5.4 Local In-Context Models Versus a Foundation Model (RQ3)

Table 3 compares the small local and the huge foundation models with each other and with the recursive pipelines on shared days. The two MACL2L variants are statistically indistinguishable on their 26 shared days ($p = 0.80$), with the variant without ENTSO-E input nominally ahead. The first place of MACL2L (ENTSO-E) is probably an artefact of three additional scored days. Interestingly, the ENTSO-E input bought no measurable accuracy. MACL2L (ENTSO-E) is significantly more accurate than the spotoptim-lgbm pipeline on 29 shared days ($p = 0.019$). Zero-shot Chronos-2 is statistically tied with the daily-retuned Optuna LightGBM over 39 shared days (1346 versus 1361 MW, $p = 0.90$). Between the two in-context families, plain MACL2L leads Chronos-2 on their 26 shared days by 1156 versus 1283 MW. The margin is not significant ($p = 0.22$). The small local model is therefore at least competitive with the large foundation model. Figure 2 (bottom) illustrates the forecasts on the final challenge day.

6 Discussion

RQ1: implementing AIA requirements. The challenge shows that abstract high-risk requirements can be made operational as a small set of verifiable rules (Table 1) that are checked automatically by continuous integration. The rules did not prevent methodological diversity[1].

RQ2: competing with ENTSO-E. A rule-compliant, deterministic open-source pipeline beat the operational ENTSO-E forecast by about a third. Ten of eleven student teams achieved a lower mean MAE than the ENTSO-E baseline. We assume that the gain stems from correcting the *shape* of the daily error profile, particularly the night-time under-forecast that matters most to a load-serving operator. Compliance and accuracy were thus not in conflict.

RQ3: local versus foundation models. In-context models topped the board. Plain MACL2L led Chronos-2 nominally, and MACL2L (ENTSO-E) significantly outperformed the tuned LightGBM, whereas zero-shot Chronos-2 matched the Optuna-tuned LightGBM. Together with the evidence of Hollmann et al. [11] and Ansari et al. [10], this marks in-context regression as a genuine challenger to daily-refit gradient boosting. From the audit perspective of the section "Regulatory Requirements for Load Forecasting", however, these models are not interchangeable. Chronos-2 imports more than 100 million pretrained parameters whose training data and procedure lie outside the operator's audit boundary and whose inference is energy-intensive, whereas MACL2L runs locally, is transparent, and needs no high-performance hardware. If auditability and energy efficiency are requirements, the small local model is preferable at no loss of accuracy. The provenance and auditability of in-context models nevertheless have to be resolved before either of them can be considered for safety-critical deployment [21].

Threats to validity. The evaluation covers 41 summer days in a single bidding zone with no winter peaks or seasonal variations, so the results say nothing about the seasons in which load forecasting is hardest. Entries joined on different

[1] Auditability extended to the evaluation itself: the full submission history is public, and every number in this paper is recomputed from versioned data with a render-time gate against the published leaderboard.

dates. Although all cross-entry claims rest on shared-day paired tests, the partial-coverage means on the board remain sample-dependent [38]. The carry-forward penalty rule mixes submission discipline with model quality, as the last place of a team with ten carried-forward days shows.

7 Conclusion

This paper reported a 41-day live challenge on day-ahead forecasting of the aggregated German transmission-grid load under EU AI Act requirements. AIA obligations were translated into four code-development and four process rules of the open-source library `spotforecast2-safe` that every student entry and every explicitly designated rule-compliant reference entry had to satisfy. Under these rules, a deterministic recursive LightGBM pipeline tuned by `spotoptim` reduced the MAE of the official ENTSO-E day-ahead forecast by 34.7 % ($p = 0.0001$), ten of eleven student teams outperformed ENTSO-E, and the rankings were robust across metrics. Transparent local models (MACL2L) led the leaderboard and were at least competitive with the foundation model Chronos-2, which has more than 100 million parameters.

Future work covers a multi-season evaluation including winter load regimes (the challenge continues[2]), probabilistic forecasts, richer weather covariates, and the formalisation of requirements for safety-critical machine learning together with regulators, operators, and working groups such as ExTraSafe [39]. The evaluation pipeline is not restricted to this challenge and can be applied to other forecasting tasks [21].

Data, Code, and Acknowledgements

The leaderboard and submission history are archived online[3]. The packages `spotforecast2-safe` [24] and `spotoptim` [33] are open source under the AGPL-3.0-or-later licence. Load data stem from ENTSO-E [17] and weather

[2] `https://advm1.gm.fh-koeln.de/~bartz/sf2-forecast`
[3] `https://bartzbeielstein.github.io/challenge-leaderboard/`

data from Open-Meteo [26]. We thank all student teams of the summer term 2026 at TH Köln for their forecasts and the peer reproduction of each other's results.

Competing Interests

The first author develops the open-source packages `spotforecast2-safe` and `spotoptim` used in this paper. The MACL2L models were developed by the industrial partner of this project, namely the Bartz & Bartz GmbH. No financial competing interests are declared.

Use of AI Tools

Transparency notice: parts of this paper were researched, drafted, and translated with the support of artificial intelligence. This information is provided voluntarily for complete transparency, as no legal requirement mandates its disclosure.

References

[1] K. Ullah, M. Ahsan, S. M. Hasanat, M. Haris, H. Yousaf, S. F. Raza, R. Tandon, S. Abid, and Z. Ullah, "Short-term load forecasting: A comprehensive review and simulation study with cnn-lstm hybrids approach," *IEEE Access*, vol. 12, pp. 111858–111881, 2024.

[2] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[3] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21,*

Table 1: Code-development (CR) and process (PR) rules of `spotforecast2-safe` [19] and the AIA articles [12] they primarily address. CR-1 to CR-4 were mandatory for every student entry and for reference entries explicitly designated as rule-compliant, and PR-1 to PR-4 governed how the corresponding packages were developed and shipped.

Rule	Requirement and implementation in `spotforecast2-safe`	AIA
CR-1	*No dead code:* tests and executable docstring examples for functions and classes	Art. 11, 15
CR-2	*Deterministic transformations:* the same input results in the same bit-level output; fixed random seeds	Art. 12, 15
CR-3	*Fail-safe handling:* invalid inputs fail explicitly; missing values are rejected by default or passed to an explicit, logged repair or truncation policy; repaired target values receive zero training weight	Art. 10, 15
CR-4	*Minimal attack surface:* short, versioned blocklist of forbidden dependencies; vendored, reviewed subset of `skforecast`	Art. 15
PR-1	*Traceability:* documentation coverage; a release guard fails when a public symbol is undocumented	Art. 11, 13
PR-2	*Documented threat model:* STRIDE table (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) kept beside the code it describes	Art. 15
PR-3	*Supply-chain integrity:* software bill of materials	Art. 15
PR-4	*Structured audit log:* every operational action emits a record using a pinned, versioned logging schema	Art. 12, 14

Table 2: Final leaderboard of the live phase (41 target days, 10 June to 20 July 2026), ranked by mean daily MAE. MAE, RMSE, and bias in MW, MAPE in %. Asterisks mark organiser-operated reference models; Days counts scored days with carried-forward days in parentheses. The seasonal naives are unranked benchmarks.

Rank	Entry	MAE	RMSE	MAPE	Bias	Days
1	MACL2L (ENTSO-E)*	1137.1	1403.7	2.20	54.2	29
2	MACL2L*	1156.0	1399.7	2.26	339.3	26
3	Hot Rod	1227.8	1452.4	2.42	−90.1	41
4	chronos*	1345.8	1606.8	2.61	−482.6	39
5	optuna lgbm*	1361.1	1615.5	2.67	−253.1	39
6	spotoptim lgbm*	1369.1	1626.0	2.70	−175.8	35
7	Team Neura	1404.8	1631.3	2.79	−378.5	41
8	spotoptim catboost*	1417.3	1690.1	2.77	−166.2	27
9	spotoptim causal*	1453.5	1745.7	2.95	−34.1	16
10	Das A Team	1459.8	1723.0	2.86	−207.3	35
11	spotoptim xgb*	1526.3	1793.7	3.03	230.1	27
12	Team Gladiators	1639.1	1882.9	3.28	−151.8	41
13	Team Fabinalii	1672.7	1957.2	3.33	−387.5	41
14	Eigen-Squad	1742.5	2009.2	3.49	584.6	41 (1)
15	DDKAST	1745.2	2018.2	3.44	−103.8	41
16	Syntaxerror	1779.5	2096.6	3.53	347.7	41 (1)
17	Voltrion	1897.6	2185.6	3.78	591.1	41
18	Team ImmerZuSpaet	1962.4	2362.3	3.84	−373.8	41 (1)
19	ENTSO-E baseline	2155.6	2434.8	4.34	−38.4	41
20	Team Weather Report	2271.5	2613.4	4.52	4.8	41 (10)
–	Seasonal naive (168 h)	2501.7	2823.5	4.94	−503.4	41
–	Seasonal naive (24 h)	3994.8	4502.7	7.95	58.4	41

Table 3: Paired comparisons of the in-context and foundation models on shared scored days: mean MAE (MW) of each side over exactly those days, number of days on which entry A was more accurate (Wins), and two-sided paired t-test p-value.

A vs B	Days	MAE A	MAE B	Wins	p
MACL2L vs MACL2L (ENTSO-E)	26	1156.0	1171.3	13/26	0.804
MACL2L vs chronos	26	1156.0	1283.1	15/26	0.218
MACL2L (ENTSO-E) vs spotoptim lgbm	29	1137.1	1391.9	19/29	0.019
MACL2L vs spotoptim lgbm	26	1156.0	1377.3	17/26	0.066
chronos vs spotoptim lgbm	35	1250.4	1369.1	17/35	0.238
chronos vs optuna lgbm	39	1345.8	1361.1	20/39	0.901
MACL2L vs spotoptim xgb	26	1156.0	1539.5	18/26	0.025

2012, Tutorial Lectures (M.-A. Aufaure and E. Zimányi, eds.), pp. 62–77, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[4] T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, and L. Callot, "Criteria for classifying forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 167–177, 2020.

[5] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.

[6] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020. M4 Competition.

[7] T. Januschowski, Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, and J. Gasthaus, "Forecasting with trees," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1473–1481, 2022.

[8] S. Ben Taieb and R. J. Hyndman, “A gradient boosting approach to the Kaggle load forecasting competition,” *International Journal of Forecasting*, vol. 30, no. 2, pp. 382–394, 2014.

[9] S. Elsayed, D. Thyssens, A. Rashed, H. S. Jomaa, and L. Schmidt-Thieme, “Do we really need deep learning models for time series forecasting?,” 2021.

[10] A. F. Ansari, O. Shchur, J. Küken, A. Auer, B. Han, P. Mercado, S. S. Rangapuram, H. Shen, L. Stella, X. Zhang, M. Goswami, S. Kapoor, D. C. Maddix, P. Guerron, T. Hu, J. Yin, N. Erickson, P. Mutalik Desai, H. Wang, H. Rangwala, G. Karypis, Y. Wang, and M. Bohlke-Schneider, “Chronos-2: From univariate to universal forecasting.” arXiv, 2025.

[11] N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirrmeister, and F. Hutter, “Accurate predictions on small data with a tabular foundation model,” *Nature*, vol. 637, no. 8045, pp. 319–326, 2025.

[12] European Parliament and Council of the European Union, “Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).” Official Journal of the European Union, 2024.

[13] European Parliament and Council of the European Union, “Directive (EU) 2022/2555 of the European Parliament and of the Council on measures for a high common level of cybersecurity across the Union (NIS-2 Directive).” Official Journal of the European Union, 2022.

[14] European Parliament and Council of the European Union, “Regulation (EU) 2024/2847 of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements (Cyber Resilience Act).” Official Journal of the European Union L 2024/2847, 2024.

[15] European Parliament and Council of the European Union, “Directive (EU) 2024/2853 of the European Parliament and of the Council of 23 October 2024 on liability for defective products and repealing Council

Directive 85/374/EEC.” Official Journal of the European Union, L series, 18 November 2024, 2024.

[16] T. Bartz-Beielstein, “challenge-leaderboard: Public leaderboard repository of the day-ahead load-forecasting challenge,” June 2026.

[17] ENTSO-E, “ENTSO-E Transparency Platform.” European Network of Transmission System Operators for Electricity, 2024.

[18] N. Chagnet, “Energy demand forecaster for France.” Open-source repository, 2025. Recursive LightGBM/XGBoost forecasting pipeline on ENTSO-E data, built with skforecast.

[19] T. Bartz-Beielstein and E. Bartz, “Time-Series Forecasting in Safety-Critical Environments: An Open-Source Package for EU-AI-Act-Compliant Development / Zeitreihenprognose in sicherheitskritischen Umgebungen: Ein Open-Source-Paket für die KI-VO-konforme Entwicklung,” 2026. arXiv v2 is a bilingual twin paper: English version first, German original below (98 pages total). Checked 2026-08-10, on the day of the v2 replacement upload.

[20] European Parliament and Council of the European Union, “Directive (EU) 2022/2557 of the European Parliament and of the Council on the resilience of critical entities (CER Directive).” Official Journal of the European Union, 2022.

[21] T. Bartz-Beielstein, I. Akiev, and L. Mohamad, “Short-term load forecasting under EU-AI Act requirements in safety-critical environments: Results from a 41-day live challenge on the aggregated German transmission-grid load.” arXiv, 2026.

[22] Deutscher Bundestag, “Gesetz über das Bundesamt für Sicherheit in der Informationstechnik und über die Sicherheit in der Informationstechnik von Einrichtungen (BSI-Gesetz – BSIG).” BGBl. 2025 I Nr. 301, ausgefertigt am 2. Dezember 2025, 2025. Recast by the act implementing Directive (EU) 2022/2555 (NIS2); in force since 6 December 2025. Governs the information security of critical infrastructure, alongside the physical resilience regime of the KRITIS umbrella act.

[23] T. Möbius, M. Watermeyer, O. Grothe, and F. Müsgens, "Enhancing energy system models using better load forecasts," *Energy Systems*, vol. 16, no. 2, pp. 573–602, 2025. Published online 1 July 2023; preprint arXiv:2302.11017.

[24] T. Bartz-Beielstein, "spotforecast2-safe: Safety-critical subset of spotforecast2," 2026. Version 28.0.1, AGPL-3.0-or-later, accessed 23 September 2026. Also on PyPI: https://pypi.org/project/spotforecast2-safe/.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

[26] P. Zippenfenig, "Open-Meteo.com Weather API." Zenodo, 2024.

[27] J. Amat Rodrigo and J. Escobar Ortiz, "skforecast," July 2026.

[28] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 6638–6648, 2018.

[31] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International Journal of Forecasting*, vol. 16, no. 4, pp. 437–450, 2000. The M3- Competition.

[32] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012. Data Mining for Software Trustworthiness.

[33] T. Bartz-Beielstein, "Optimization with SpotOptim," *arXiv e-prints*, p. arXiv:2604.13672, Apr. 2026.

[34] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, *Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide*. Singapore: Springer, 2022.

[35] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019.

[36] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. Pineda Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the language of time series." arXiv, 2024. Transactions on Machine Learning Research.

[37] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, Australia: OTexts, 3rd ed., 2021.

[38] H. Hewamalage, K. Ackermann, and C. Bergmeir, "Forecast evaluation for data scientists: common pitfalls and best practices," *Data Mining and Knowledge Discovery*, vol. 37, no. 2, pp. 788–832, 2023.

[39] Fachbereich Künstliche Intelligenz der Gesellschaft für Informatik, "Arbeitskreis Explainability, Transparency, and Safety (ExTraSafe)," 2026. Working group of the AI section of the German Informatics Society.

Synthetic Sensor Data Generation for Sensor Network Metrology Using Large Language Models

Anshuman Singh[1], Anupam Prasad Vedurmudi[2]

[1]Physikalisch-Technische Bundesanstalt
Bundesallee 100, 38116 Braunschweig
E-Mail: anshuman.singh@ptb.de
beginequation1ex] [2]Physikalisch-Technische Bundesanstalt
Abbestr. 2-12, 10587 Berlin
E-Mail: anupam.vedurmudi@ptb.de

Abstract

Real-world sensor datasets often suffer from missing observations and noise causing imbalances that hinder the development of reliable machine-learning models. This study investigates the use of open-source Large Language Models (LLMs) to generate realistic augmentations of multivariate sensor time series data. Using data from an indoor enviromental monitoring system in an office building, issues such as strong temporal persistences of the sensor readings and inter-sensor dependencies were addressed. A framework was developed using Low-Rank Adaptation (LoRA) to predict normalized multivariate transitions followed by the retrieval of physically observed joint transitions via a retrieval-based decoder. The study reveals that, while LLMs are promising numerical transition learners, model scale alone cannot fully reproduce certain sensor regimes and complex cross-sensor covariances. Temporal conditioning is essential to model realistic persistence and high-fidelity data synthesis requires the integration of explicit temporal, regime, and privacy structures beyond mere model capacity.

Keywords: Internet of Things, synthetic sensor data, large language models, time-series generation, parameter-efficient fine-tuning, LoRA, Unsloth, Qwen2.5, Llama 3.1, temporal conditioning, privacy-preserving synthesis.

1 Introduction

The proliferation Internet-of-Things (IoT) systems in the environmental sensor networks has led to the generation of massive volumes of heterogeneous data, both in outdoor [1] and indoor [2] environments. In the case of indoor systems, measurements of temperature, relative humidity, ambient pressure, carbon dioxide (CO_2), illuminance, ultraviolet (UV) radiation and other atmospheric variables are critical for building monitoring and energy management. However, the utility of these datasets is often hindered by real-world constraints such measurement noise, missing observations, and limited sampling periods, resulting in imbalanced datasets due to a lack of examples for particular environmental regimes. Furthermore, the sharing and redistribution of such data is frequently restricted by privacy concerns and operational security, further complicating the collaborative development of robust machine-learning and calibration models [4].

Synthetic data generation offers a promising alternative by reproducing the statistical and temporal properties of observed data without compromising original records. Traditional approaches for synthetic time series data generation include probabilistic models, variational autoencoders (VAEs), diffusion models and specialized time-series architectures using generative adversarial networks (GAN). These methods often struggle to capture the intricate dynamics of indoor environments. Specifically, indoor sensor streams exhibit [5] issues such as strong temporal persistence, daily periodicity, and complex cross-sensor dependencies. Treating these observations as independent tabular rows, as seen in many baseline synthesis methods, fails to capture the underlying physical dynamics and renders the synthetic output unrealistic. Recent advances in transformer-based Large Language Models (LLMs) [3] have demonstrated a surprising capacity for structured data generation. While preliminary work has explored the use of LLMs for tabular and sequence synthesis, adapting these

models to multivariate sensor data remains challenging due to heterogeneous measurement scales and complex regime transitions.

To address these challenges, this study proposes a transition-based LLM framework for the generation of realistic multivariate indoor sensor time series. The primary objective is to demonstrate how open-source LLMs can learn the latent structure of environmental observations sufficiently to produce high-fidelity synthetic datasets. Rather than requiring the LLM to hallucinate raw numerical values independently, the proposed approach treats the generation problem as the prediction of a normalized multivariate transition, which is subsequently mapped to a physically observed joint transition via a retrieval-based decoder. This architecture, optimized via parameter-efficient fine-tuning (LoRA) [13] as well a chronological (as opposed to a random) data splitting, explicitly incorporates temporal and regime-specific conditioning to preserve the inherent persistence and covariance of the sensor network. This is achieved through a rigorous evaluation framework that moves beyond simple discriminator performance (such as AUC), instead prioritizing marginal distributions, cross-sensor correlations, temporal fidelity, and privacy-preserving constraints. By combining generative modeling with a multi-dimensional quantitative assessment, this study provides insights into the capabilities and limitations of LLMs as numerical transition learners for environmental sensor network metrology. In this manner, the framework goes beyond independent row generation toward a model that explicitly understands the transition dynamics of a physical system.

1.1 Related Work

Large language models have increasingly been investigated for structured sensor and tabular data. REaLTabFormer [3] demonstrated an autoregressive transformer-based generation of tabular and relational data and introduced mechanisms for detecting overfitting and copying. GReaTER (Generate Realistic Tabular data after data Enhancement and Reduction) subsequently introduced semantic enhancement and cross-table connections for improving LLM-based tabular synthesis [10]. The SDForger framework is particularly relevant as it directly addresses synthetic time-series generation using autoregressive LLMs

and reports preservation of statistical and temporal characteristics [11]. SensorLLM demonstrated explicit alignment between numerical sensor trends and language-model representations for human activity recognition [12]. The present study instead focuses on generation of continuous multivariate environmental observations.LLM-based sensor applications have also been investigated for few-shot optimization and fault diagnosis. Ronando and Inoue used numerical distance followed by LLM-based contextual selection for sensor examples, illustrating the usefulness of combining numerical similarity with LLM representations [15]. Zheng et al. [16] systematically examined fine-tuned LLMs for complex-system fault diagnosis and explicitly studied factors including normalization and missing data . Parameter-efficient fine-tuning is essential when adapting large models to small domain datasets. LoRA, which is part of the current framework, freezes the pretrained weights and introduces low-rank trainable matrices, substantially reducing the number of trainable parameters and memory requirements [13]. A recent survey provides a broader taxonomy of parameter-efficient adaptation methods [6]. Finally, synthetic time-series evaluation requires multiple complementary dimensions rather than a single score. TimeGAN [9] established the use of discriminative, predictive, and visualization-based evaluation, while more recent surveys emphasize distributional, temporal, and downstream utility measurements.

2 Dataset and Exploratory Analysis

The original hourly dataset contained 2,497 chronological observations and 13 numerical sensor variables plus a timestamp. The sensors were part of a standard environmental measurement setup measuring parameters such as carbon dioxide concentration, pressure, relative humidity and temperature. Additionally optical sensors such as measuring total illuminance and UV radiation were also present. The details of the constituent sensors are provided in Tab 1. Values for relative humidity (RH) and temperature were measured by multiple sensors with some being attached to a sensor board with an external power supply (BME, SHT), while others were on wireless battery operated boards (MHZ). The data was further divided chronologically so as to ensure that temporally adjacent observations from the same regime do not appear

Group	Sensors
Atmospheric	CO_2-CCS, CO_2-MHZ
Thermodynamic	Pressure, RH-BME, RH-SHT, Temp-BME, Temp-MHZ, Temp-SHT
Optical	Color temperature, Lux, UV
Activity/occupancy-related	Dust count, Noise

Tabelle 1: Grouping of sensors by category.

in both training and evaluation sets, resulting in overly optimistic estimates; see Tab. 2. The exploration of the hourly data revealed a persistent, periodic,

Split	Observations	Period
Training	1,747	1 May–12 July 2026
Validation	375	12–28 July 2026
Test	375	28 July–13 August 2026

Tabelle 2: Chronological split of the dataset.

regime-switching multivariate dynamical system rather than independent tabular observations. Finally, a three-hourly subset of the full dataset was used to evaluate whether the proposed representation and fine-tuning approach remained effective when observations were more widely separated in time. The three-hourly dataset contained 833 observations and the same 13 sensor variables. A 70/15/15 chronological division in this case corresponds approximately to 583 training, 125 validation, and 125 test observations.

2.0.1 Preprocessing

Records were first sorted by timestamp and duplicate timestamps were removed. No random permutation was applied before splitting in order to prevent information from future observations entering the training process. Hourly continuity was explicitly verified within training, validation, and test partitions as the

transition-based modeling assumes that single time step $t \rightarrow t+1$ represents a one-hour transition. Moreover, the timestamp was augmented with a "daylight regime" indicator which was defined as

$$R(h) = \begin{cases} \text{night} & 0 \leq h < 5 \quad \text{morning} \quad 5 \leq h < 9 \\ \text{day} & 9 \leq h < 17 \quad \text{evening} \quad 17 \leq h < 21 \\ \text{night} & 21 \leq h < 24 \end{cases} \tag{1}$$

This representation reflects the strong daily periodicity observed during EDA.

3 Methods

3.1 Formulation of the problem

Formally, the task of synthetic sensor data generation can be described as the estimation of the probability distribution $p(\mathbf{x}_t)$ of the d-dimensional vector, $\mathbf{x}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,d}]^\top$ corresponding to the measurements of all sensors in the network at time t. A conventional row-wise generator attempts to learn the conditional distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, where the distribution of the present sensor vector is estimated using only the vector at the previous time step. In the present study, we instead estimate the conditional transition distribution

$$p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{z}_t) \,, \tag{2}$$

where z_t contains additional temporal and contextual information. This distinction is central as the exploratory analysis demonstrated extremely strong temporal persistence quantified high values of lag-one autocorrelations ($> .95$) for pressure, Temp-SHT, RH-SHT and CO2-CCS. In this manner, we avoid the situation where a model that produces statistically plausible independent rows that are physically implausible.

Training-only normalization: For each sensor the normalized values are given by,

$$z_{(t,j)} = \frac{x_{(t,j)} - \mu_j^{\text{train}}}{\sigma_j^{\text{train}}} \ , \tag{3}$$

i.e., the mean and standard deviation were calculated exclusively from the training partition. This is necessary because direct normalization using the entire dataset would introduce information from validation and test periods into model construction.

Transition representation: Instead of asking the LLM to produce an unconstrained 13-dimensional state, the model predicts a normalized transition:

$$\Delta x_{(t,j)} = x_{(t+1,j)} - x_{(t,j)}, \tag{4}$$

followed by,

$$\Delta \tilde{x}_{(t,j)} = \frac{\Delta x_{(t,j)} - \mu_{(\Delta x,j)}}{\sigma_{(\Delta x,j)}}, \tag{5}$$

where $\mu_{(\Delta x,j)}$ and $\sigma_{(\Delta x,j)}$ represent the mean and standard deviation of Δx. This substantially reduces the numerical burden on the LLM because highly persistent state values need not be regenerated from scratch.

Regime representation: Carbon dioxide exhibited particularly strong regime persistence. The CO_2-CCS transition matrix demonstrated probabilities as high as 0.955 for remaining in the highest regime and approximately 0.91 for remaining in the lowest regime. Consequently, the current CO_2 regime was introduced as explicit conditioning information.

Joint transition retrieval: The generated transition is not independently decoded sensor by sensor. Instead, it is used as a key for selecting a joint observed transition from the training set. Candidate selection first considers similarity in the current state and temporal context and then evaluates transition similarity. This design preserves combinations of sensor values that actually occurred together in the training data.

Privacy-aware retrieval: A source-state distance constraint was introduced in the final privacy-aware variant. Candidate transitions whose source state

was too close to the current state were excluded where sufficient alternatives existed. This reduces the possibility that synthetic records become direct copies of training observations.

3.1.1 Architecture of the models used

4.1 Llama-3.1-8B: Llama-3.1-8B is an open-weight 8-billion-parameter autoregressive Transformer model from the Llama 3 family. The Llama 3.1 family introduced extended context and multilingual capabilities relative to earlier Llama generations [17]. In this work, Llama-3.1-8B was evaluated on the three-hourly indoor sensor representation using the same fundamental transition-generation principle. Its final training and evaluation losses were:

$$L_{\text{train}} = 0.69 \, , \quad L_{\text{eval}} = 0.61 \, . \tag{6}$$

The relatively close training and evaluation losses indicate that the model did not exhibit severe conventional overfitting during the final training stage.

Qwen2.5-7B: Qwen2.5 is a family of open-source large language models designed for general language and reasoning applications. The technical report describes substantial improvements in both pretraining and post-training relative to previous Qwen generations [14]. The 7-billion-parameter model provided an intermediate computational scale between the Llama-3.1-8B and Qwen2.5-32B models. The model was adapted to numerical transition generation rather than conventional natural-language generation.

Qwen2.5-32B: Qwen2.5-32B was used for the hourly dataset and represents the largest model evaluated in this study. Its use tests whether increased model capacity provides benefits for the highly structured multivariate temporal problem. The best validation configuration obtained a score of ~ 0.2.

3.2 Fine-Tuning Method and Implementation

The central architecture can be represented as:

$$\begin{aligned}(x_{(t-1)}, x_t, \Delta x_t, z_t \rightarrow \text{LLM} \rightarrow &\Delta\hat{x}_{(t+1)}) \\ &\rightarrow \text{joint transition retrieval} \rightarrow \hat{x}_{(t+1)}) \ ,\end{aligned} \tag{7}$$

where $z_t = [\text{hour, month, daylight regime, } CO_2 \text{ regime, sensor groups}]$. Five conditioning configurations were evaluated sequentially.

- State and time: The first configuration supplied the current normalized sensor state and temporal information to the LLM.
- State and time recent-change: The second configuration additionally supplied $\Delta x_t = x_t - x_{(t-1)}$, which provides local velocity information.
- Addition of carbon-dioxide regime: The third configuration introduced the current CO_2regime to distinguish qualitatively different environmental states.
- Addition of sensor-group representation: The fourth configuration grouped sensors into atmospheric, thermodynamic, optical, and activity-related subsystems.
- Privacy-constrained retrieval: The fifth configuration introduced a minimum source-state retrieval distance to reduce direct training-record reproduction.

The validation results showed that adding information does not automatically improve synthesis. For the final hourly study, the best validation configuration was the third conditioning level rather than the most complicated privacy/group configuration. Additional conditioning variables can increase model complexity without necessarily improving distributional or temporal fidelity. The resulting fine-tuning parameters are shown in Tab. 3. The effective training batch size is $B_{\text{eff}} = B_{\text{device}} \times G = 44 = 16$. Finally, a deliberately conservative learning rate of 210^{-5} was chosen as the datasets are small relative to the scale of the pretrained models. The 2,048-token context provides sufficient room for numerical state,

Tabelle 3: Fine-tuning results

Parameter	Value	Parameter	Value
Max seq. length	2,048	LoRA rank	8
Epochs	8	LoRA α	32
Train batch size	4	LoRA dropout	0
Eval. batch size	4	LoRA bias	none
Grad. accumulation	4	Model loading	4-bit
Effective batch size	16	Gen. max tokens	350
Learning rate	2×10^{-5}	Gen. temperature	0.20
Warm-up steps	20	Top-p	0.90
Random seed	42	Repetition penalty	1.05

temporal descriptors, conditioning information, and the target transition without unnecessarily increasing computational cost.

3.2.1 Optimization Algorithm

The models were optimized using gradient-based supervised fine-tuning with the AdamW-based optimizer implemented through the Hugging Face training stack. For model parameters θ, the generic update is

$$\theta_{(k+1)} = \theta_k - \eta \hat{m}_k / (\sqrt{\hat{v}_k + \varepsilon}) \,, \tag{8}$$

where η is the learning rate and $\hat{m}_k$,$\hat{v}k$ are bias-corrected first- and second-moment estimates. Only LoRA parameters are trainable. The pretrained Transformer weights remain frozen. Gradient accumulation makes four micro-batches equivalent to an effective batch size of 16.

3.2.2 Loss Function

The primary optimization objective is autoregressive token-level cross entropy:

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(y_i|y_{<i}x) \ . \tag{9}$$

For a numerical transition represented as a structured textual sequence, this loss encourages the LLM to assign high probability to the correct serialized transition representation. The final Llama-3.1-8B training run resulted in a training loss of $L_{\text{train}} = 0.69$ and an evaluation loss of $L_{\text{eval}} = 0.61$. The Qwen-based runs similarly converged without evidence that increasing the number of epochs alone would resolve the principal synthesis errors.

3.2.3 Unsloth-Based Fine-Tuning

Unsloth [7] is an open-source framework for efficient local fine-tuning and inference of LLMs. It provides optimized model-loading, parameter-efficient fine-tuning (PEFT), memory-management, and inference functionality and supports models including Llama and Qwen. Its documentation describes LoRA and quantized fine-tuning as memory-efficient alternatives to full fine-tuning [6]. The implementation used the FastLanguageModel interface for model loading and PEFT configuration, Hugging Face TrainingArguments for optimization, and the TRL SFTTrainer for supervised fine-tuning. The main components were: FastLanguageModel. Loads the pretrained model and tokenizer and prepares the model for efficient fine-tuning. 4-bit model loading. The base model was loaded with 4-bit quantization toreduces memory requirements and allows larger models to be fine-tuned on available GPU hardware. LoRA. The pretrained model weights remain frozen while trainable low-rank matrices are inserted into selected Transformer projections. LoRA can dramatically reduce trainable parameters and memory requirements compared with full fine-tuning[7]. SFTTrainer. Supervised fine-tuning was performed using input-output transition examples. Automatic BF16/FP16 selection. BF16 was enabled when supported by the GPU; otherwise FP16 was used. Inference optimization. FastLanguageModel.for_inference() was used after training. The

implementation targeted $\{q_{\text{proj}}, k_{\text{proj}}, v_{\text{proj}}, o_{\text{proj}}, \text{gate}_{\text{proj}}, \text{up}_{\text{proj}}, \text{down}_{\text{proj}}\}$. This covers both attention and major feed-forward projections.

3.3 Accuracy Function

A conventional classification accuracy is not an appropriate primary metric for continuous multivariate sensor generation. For this reason, the study does not treat an arbitrary token-level or row-level accuracy as a measure of synthetic-data quality. Instead, the quality of a generated dataset is assessed through:

- distributional similarity
- correlation preservation
- temporal profile similarity
- lag autocorrelation
- transition similarity
- one-step prediction error
- collapse and drift
- privacy distance
- post-hoc discriminator AUC

This follows the broader synthetic-time-series literature, where discriminative and predictive evaluations complement distributional analysis [9]. A descriptive summary of the evaluation metrics used in the present study is provided in Tab. 4. The entire methodology described so far is illustrated schematically as a flowchart in Fig. 1.

4 Results

The overall training and validation losses for the Qwen models is shown in Fig. 2 and the best validation configuration used current state, temporal information, recent change, and current CO_2 regime. Adding sensor-group representations, however, did not improve the validation objective, while the privacy-constrained configuration produced a substantially poorer validation score. This demonstrates that model complexity and conditioning richness do not necessarily translate into higher synthetic-data fidelity. The transition-based

Tabelle 4: Summary of Performance Metrics for Synthetic Data Evaluation

Metric (Ideal Value)	**Formula/Definition**	**Purpose**
JSD (Lower)	$\frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M)$ where, $M = \frac{P+Q}{2}$	Measures symmetric distributional similarity.
Wasserstein (Lower)	$W_1(P,Q) = \inf_{\gamma \in \Pi(P,Q)} E_{(x,y)\sim\gamma}[\|x-y\|]$	Measures minimum transport cost to transform P into Q.
KS Test (Lower (Pass))	$D_{KS} = \sup_x \|F_R(x) - F_S(x)\|$	Tests if two samples come from the same distribution.
Corr. Error (Lower)	$E_{corr} = \frac{1}{N_p}\sum_{(i,j)} \|\rho_R(i,j) - \rho_S(i,j)\|$	Evaluates preservation of multivariate dependence.
Disc. AUC (≈ 0.5)	$AUC = P(s(x_R) > s(x_S))$	Ability of a classifier to distinguish real from synthetic.
DCR (Higher)	$DCR(s) = \min_{r\in R} \|z(s) - z(r)\|_2$	Checks for direct reproduction of training records.
Lag-1 ACF (Lower)	$E_{\text{ACF}} = \frac{1}{d}\sum_{j=1}^{d} \|\text{ACF}_{1,R}(j) - \text{ACF}_{1,S}(j)\|$	Measures preservation of temporal persistence/-autocorrelation.
Temporal Error (Lower)	$\mu_{R,j,h}$ vs $\mu_{S,j,h}$	Compares mean values per sensor per hour of the day.
Transition Error (Lower)	$\Delta x_t = x_t - x_{t-1}$ (via Wasserstein)	Evaluates if generator reproduces dynamics/changes.
Privacy Penalty (Lower)	$S_{\text{priv}} = S_{\text{fid}} + 0.10[0.5 f_{<0.05} + 0.5 f_{<0.10}]$	Balances fidelity vs. excessive training data proximity.

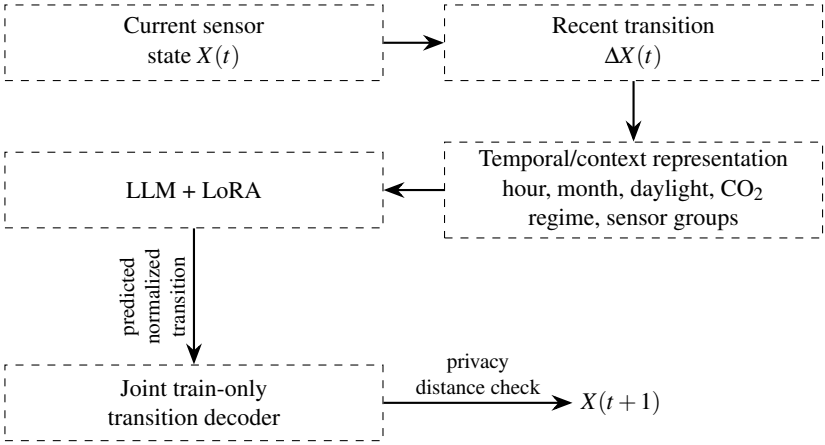


Bild 1: Flowchart describing the overal data generation framework

architecture substantially improves temporal realism compared with direct independent state generation. The final Qwen2.5-32B result obtained a test lag-one ACF error of only 0.15, demonstrating that short-term temporal persistence is substantially captured. The normalized transition Wasserstein error of 0.1X also indicates that the distribution of hourly changes is much better preserved than the raw sensor-level distributions alone would suggest.

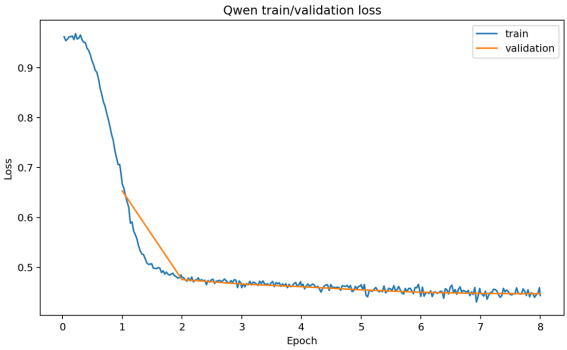


Bild 2: Overall training and validation loss for the Qwen models

Key limitation: CO_2-CCS The dominant remaining error is the high-amplitude CO_2-CCS behavior. The real data contain a large dynamic range, including high nighttime/evening CO_2 states; see Fig. 3. The synthetic generator tends to compress these states toward lower amplitudes. This explains why a model can simultaneously obtain good transition statistics, areasonable temporal persistence as well asacceptable average correlation error while still obtaining a relatively large aggregate Wasserstein distance.

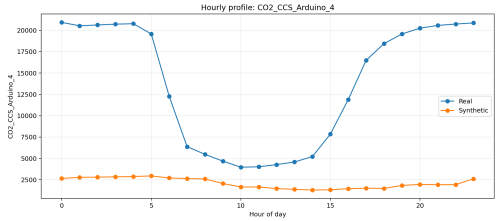


Bild 3: Hourly profile of CO_2 values measured by the CO_2-CCS sensor.

The principal hourly result obtained with Qwen2.5-32B is shown in Tab. 5.

Tabelle 5: Detailed Performance Metrics Comparison for Qwen2.5-32B

Metric	**Validation**	**Test**	**Metric**	**Validation**	**Test**
Overall score	**0.20**	—	Normalized W	0.25	0.59
JSD	0.27	**0.33**	Temp. norm. error	0.29	0.62
Wasserstein	234.21	**898.77**	ACF error	0.21	**0.15**
KS pass rate	7.69	7.69	Delta W	0.14	**0.15**
Corr. error	0.14	0.22	Delta std error	0.16	0.19
AUC	0.93	0.96	Collapse penalty	0	0
	—	—	Drift rate	0.36	0.46

Note: KS pass rate is expressed as a percentage.

4.1 Cross-sensor covariance problem

Several important relationships remain poorly reproduced. For example, the real relationship between pressure and temperature is negative, while the generated relationship can become positive. For the Qwen-based final model, representative errors include pressure–Temp-SHT and pressure–Temp-BME covariance discrepancies. This indicates that the LLM learns local numerical transition behavior more easily than the complete physical covariance structure of the indoor environment.

4.2 Distribution shift

The chronological test period extends beyond the training period. Therefore, the test set represents a more challenging temporal extrapolation problem rather than a conventional independently sampled test set. This is scientifically important: the relatively high test AUC should partly be interpreted as evidence of distributional shift plus incomplete modeling, rather than simply as poor generative quality.

4.3 Three-hourly results

The three-hourly studies with Llama-3.1-8B and Qwen2.5-7B demonstrated the same general phenomenon: the models could learn structured transitions and achieve substantially better temporal behavior than unconstrained numerical generation, but high-dimensional marginal amplitudes and cross-sensor dependencies remained difficult. For the Llama-3.1-8B run, the final training and evaluation losses were around 0.69 and 0.61, respectively, demonstrating stable optimization. Overall, the experiments indicate that temporal conditioning is more important than simply increasing model size. The Qwen2.5-32B model produced the strongest hourly validation result, but increasing model capacity did not eliminate the fundamental CO_2 and covariance limitations. The overall results of the study are summarized in Tab. 6.

Tabelle 6: Overall interpretation of the completed study

Property	**Result**
Marginal distributions	Partially reproduced
Short-term persistence	**Strongly improved**
Transition distributions	**Good improvement**
Daily temporal structure	Partially reproduced
Cross-sensor covariance	**Major remaining limitation**
High-amplitude CO regimes	**Major remaining limitation**
Mode collapse	Not observed in final selected model
Direct record copying	Reduced by retrieval/privacy constraint
Validation-based model selection	Maintained
Test leakage	None by design
AUC optimization	Not performed

5 Conclusion

This study investigated the use of open-source large language models – Llama-3.1-8B, Qwen2.5-7B, and Qwen2.5-32B – for synthetic multivariate indoor sensor time-series generation. The analysis began with a characterization of the underlying sensor system and demonstrated that the dataset cannot reasonably be treated as independent tabular observations. Extremely high lag-one autocorrelations, strong daily periodicity, sensor floors, CO_2 regimes, and substantial cross-sensor relationships require explicit temporal and multivariate modeling. A transition-based LLM architecture was therefore developed in which the language model predicts a normalized multivariate transition conditioned on the current state and temporal context. A train-only joint-transition decoder then converts this predicted transition into a physically observed multivariate transition. Parameter-efficient fine-tuning using LoRA and Unsloth enabled adaptation of 8-billion- and 32-billion-parameter open-source models despite the relatively small dataset. The completed experiments demonstrate several important conclusions. First, temporal conditioning is essential. Modeling transitions produces substantially more realistic persistence than independent

state generation. Second, larger language models alone do not solve numerical synthesis. Qwen2.5-32B produced the strongest hourly validation result, but important physical relationships remained difficult to reproduce. Third, additional conditioning information must be justified by validation performance. Adding sensor groups and privacy constraints did not automatically improve fidelity. Fourth, AUC should remain diagnostic rather than an optimization target. A synthetic dataset can have useful temporal and transition properties while remaining distinguishable from real data because of distribution shift or difficult sensor-specific marginal behavior. Fifth, privacy and fidelity must be considered jointly. Retrieval-based generation provides a natural mechanism for preserving realistic joint sensor transitions while imposing minimum distances from training observations. The most important remaining limitation is the inability to reproduce the full amplitude and regime structure of CO_2CCS and several cross-sensor covariance relationships. This indicates that future improvements should focus on physically structured representations rather than simply increasing model size or training duration. Promising avenues for future work that can build on the current study are the investigation of explicit multi-scale transition representations and second-order dynamics, the incorporation of physics-informed constraints and the evaluation of the downstream utility of the generated datasets.

Literatur

[1] M. Van Poppel et al. „SensEURCity: A multi-city air quality dataset collected for 2020/2021 using open low-cost sensor systems“. In: *Scientific Data*, 10.1. p. 322. 2023. 10.1038/s41597-023-02135-w

[2] I. A. Campodonico-Avendano et al. „COLLECTiEF dataset: A high-resolution indoor environmental dataset from European buildings across diverse climates supporting thermal, air-quality, and visual-comfort assessments“. In: *Data in Brief*, 65, 112486. 2026. doi:10.1016/j.dib.2026.112486

[3] A. V. Solatorio, O. Dupriez. „REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers“. In: *arXiv* 2023. 10.48550/arXiv.2302.02041

[4] S. Tabandeh et al.„Sensor network metrology: Current state and future directions“. In: *Measurement: Sensors*, 38, 101798. 2025. doi:10.1016/j.measen.2024.101798

[5] A. Rusch and T. Rösgen. „An Internet of Things Sensor Array for Spatially and Temporally Resolved Indoor Climate Measurements“. In: *Sensors*, 31, 22.12, 4377. 2022 doi:10.3390/s22124377

[6] L. Wang et al. „Parameter-efficient fine-tuning in large language models: a survey of methodologies“, In: *Artif Intell Rev*, 58, 227. 2025 doi:10.1007/s10462-025-11236-4

[7] M. Han, D. Han and Unsloth team. Unsloth [Online]. Available: https://github.com/unslothai/unsloth. 2023.

[8] L. von Werra et al. TRL: Transformers Reinforcement Learning [Online]. Available: https://github.com/huggingface/trl 2020.

[9] J. Yoon, D. Jarrett and M. Schaar. „Time-series Generative Adversarial Networks“. In: *Advances In Neural Information Processing Systems*, 32 2019

[10] T. Kwok, C. Wang. and G. Cheng. „GReaTER: Generate Realistic Tabular data after data Enhancement and Reduction“. In: *2025 IEEE 41st International Conference On Data Engineering Workshops (ICDEW)* , pp. 218–229. doi:10.1109/ICDEW67478.2025.00032 2025

[11] C. Rousseau et al. „Forging Time Series with Language: A Large Language Model Approach to Synthetic Data Generation“, In: *Advances In Neural Information Processing Systems*, 38, 78402-78432. 2025

[12] Z. Li et al. „SensorLLM: Aligning Large Language Models with Motion Sensors for Human Activity Recognition“. *Proceedings Of The 2025 Conference On Empirical Methods In Natural Language Processing*. pp. 354-379 https://aclanthology.org/2025.emnlp-main.19/ 2025

[13] J. Hu et al. „LoRA: Low-Rank Adaptation of Large Language Models“. *arXiv* abs/2106.09685 2021

[14] A. Yang et al.„Qwen2.5 Technical Report“, *arXiv* 2025, doi:10.48550/arXiv.2412.15115

[15] E. Ronando, E. and S. Inoue. „Few-Shot Optimization for Sensor Data Using Large Language Models: A Case Study on Fatigue Detection“. *Sensors*, 25.11, 3324. 2025 doi:10.3390/s25113324.

[16] S. Zheng et al. „Empirical Study on Fine-tuning Pre-trained Large Language Models For Fault Diagnosis of Complex Systems“, *Reliability Engineering & System Safety*, 252. pp. 110382. 2024 10.1016/j.ress.2024.110382

[17] A. Grattafiori et al. „The Llama 3 Herd of Models“. *arXiv* 2024 abs/2407.21783

[18] T. Adamet al. „Alzheimer's Disease Neuroimaging Initiative. On the fidelity versus privacy and utility trade-off of synthetic patient data“. *iScience*, 14;28(5), 112382. 2025 10.1016/j.isci.2025.112382.

Clustering-Based Rare-Event Detection Beyond Singlets in Flow Cytometry

Yejin Kwak[1,2,3,*], Roberto Würth[4,5,8], Angelo Jovin Yamachui Sitcheu[1,3], Daniel Hübschmann[2,6,7,8,9], Ralf Mikut[1]

[1]Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT)

[2]Innovation and Service Unit for Bioinformatics and Precision Medicine, German Cancer Research Center (DKFZ)

[3]Helmholtz Information and Data Science School for Health (HIDSS4Health)

[4]Heidelberg Institute for Stem Cell Technology and Experimental Medicine (HI-STEM gGmbH)

[5]Division of Stem Cells and Cancer
German Cancer Research Center (DKFZ) and DKFZ-ZMBH Alliance

[6]National Center for Tumor Diseases (NCT), NCT Heidelberg, a partnership between DKFZ and Heidelberg University Hospital

[7]Pattern Recognition and Digital Medicine Group, Heidelberg Institute for Stem Cell Technology and Experimental Medicine (HI-STEM gGmbH)

[8]German Cancer Consortium (DKTK), Core Center Heidelberg

[9]Division of Translational Precision Medicine, Institute of Human Genetics, Heidelberg University

[*]E-mail: yejin.kwak@kit.edu

Abstract

Physical cell interactions are biologically relevant, but they are routinely discarded as doublets or multiplets in standard flow cytometry workflows. A recent clustering-based cytometry framework has instead retained multiplets while

using non-uniform sampling, or sketching, to preserve rare cell types during large-scale analysis. However, it remains unclear how well such sampling strategies support downstream recovery of extremely rare events and how reliably singlets can be distinguished from multiplets. To address this, we compare leverage-score and geometric sketching for recovering circulating tumor cells (CTCs) among singlet events, and evaluate whether adding side scatter (SSC) ratio to forward scatter (FSC) ratio improves singlet/multiplet discrimination across all events. The analysis comprises 4 blood samples from 3 patients, ranging from 116,507 to 1,361,065 events each. Geometric sketching yielded higher CTC recovery on average, particularly in recall, though performance remained sensitive to sample identity and clustering resolution. FSC ratio alone provided strong singlet/multiplet separation, while adding SSC ratio yielded no consistent improvement. These findings highlight the importance of both sampling strategy and feature selection for extending clustering-based cytometry to extremely rare events while retaining information beyond singlets.

1 Introduction

Cellular interactions are important for understanding development, homeostatic function, and pathophysiology of multicellular organisms, shaping processes from immune activation to tumor progression. Characterizing the frequency, composition, and dynamics of specific interaction types can reveal disease-associated alterations, inform patient stratification, and support the monitoring of therapeutic response. Representative examples include circulating tumor cell-neutrophil clusters, whose physical association enhances metastatic competence and has been proposed as a therapeutic target [1], and T cell-tumor cell conjugates, which are therapeutically induced by bispecific T cell engagers to promote tumor cell killing [2].

Despite the widespread adoption of single-cell genomics and spatial transcriptomics for dissecting cellular heterogeneity, direct analysis of physically interacting cells remains limited. Computational approaches such as CellPhoneDB [3] and CellChat [4] infer cell-cell communication indirectly, from the expression of ligands, receptors, and associated signaling components

across cell populations, without confirming physical contact between interacting cells. Spatial transcriptomics methods [5, 6] preserve spatial context and can identify neighboring cells, but spatial proximity alone does not establish direct physical interaction. Moreover, transcriptome-based inference is constrained by prior knowledge of signaling relationships, whereas high-resolution spatial approaches involve trade-offs among gene coverage, cell throughput, acquisition time, and cost.

Flow cytometry [7], by contrast, offers the potential to capture and characterize physical cell interactions through light scatter properties and surface marker co-expression at high throughput and comparatively low cost. Standard flow cytometry workflows, however, are generally centered on the analysis of individual singlet cells. This is traditionally achieved through hierarchical gating, which evaluates one or two parameters at each step [8], or, as panel sizes have grown, through clustering on the joint feature space [9]. In either case, doublets and multiplets are routinely excluded as technical artifacts prior to analysis, discarding potential interaction signal before it is examined. Historically, flow-cytometric analysis of cell-cell conjugates has instead been applied in targeted assays designed to quantify specific interaction types [10].

More recently, interact-omics [11] has begun to recover the discarded multiplet signal. We extend this line of work specifically to the rare-event regime, where events of interest constitute only a small fraction of all measured events. Because cytometry datasets can contain large numbers of events, computational pipelines often perform clustering on a downsampled subset, using sampling strategies designed to retain rare events, with cluster assignments subsequently extended to the full event pool [11, 12]. However, it remains unclear how effectively different downsampling strategies preserve extremely rare events in clinical flow cytometry.

To examine an extreme rare-event setting, we focus on circulating tumor cells (CTCs), which can occur at frequencies as low as approximately one cell per 10^9 blood cells, yet are implicated in metastatic spread and therapy resistance [13]. We therefore compare downsampling strategies for rare CTC recovery among singlets, and assess whether augmenting clustering with an additional scatter-derived feature can improve singlet/multiplet discrimination.

2 Background and Related Work

Hierarchical gating has long been the standard approach for cell population identification in flow cytometry [14]. However, as the number of measured parameters increases, hierarchical gating becomes increasingly time-consuming and difficult to scale [8]. The number of possible bi-axial marker combinations grows quadratically with panel size. Spectral flow cytometry has enabled panels of 50 or more fluorescent parameters with instruments recording fluorescence across dozens to more than one hundred detection channels [15]. At this scale, comprehensive analysis through sequential bi-axial gating becomes increasingly impractical. Consequently, high-dimensional cytometry analyses have increasingly adopted clustering-based analytical tools that consider multiple markers simultaneously [9].

Among the widely used clustering algorithms for this purpose are FlowSOM [16] and PhenoGraph [17], both of which have been implemented in commonly used cytometry analysis software. FlowSOM applies a self-organizing map followed by meta-clustering, whereas PhenoGraph constructs a nearest-neighbor graph and applies Louvain community detection. Both approaches, however, have been shown to vary substantially in their ability to resolve rare cell populations [18].

Cytometry pipelines have long incorporated downsampling strategies to better preserve rare populations during analysis. SPADE [12], for example, uses density-dependent downsampling to reduce differences in local cell density, thereby increasing the relative representation of rare populations prior to clustering. More recently, sketching approaches adopted for large-scale single-cell genomics have begun to be applied to cytometry data. Interact-omics [11], for instance, uses leverage-score sketching [19].

Leverage-score sketching [19] and geometric sketching [20] have both been applied in single-cell genomics to preserve rare populations during downsampling, but rely on contrasting criteria for determining which observations to retain. Leverage-score sketching, as implemented in Seurat v5 [19], assigns sampling probabilities according to statistical leverage, prioritizing high-leverage observations and thereby enriching rare populations. Geometric sketching [20],

in contrast, covers the feature space with equal-volume regions and samples approximately evenly across them, reducing the dominance of densely populated regions and increasing the relative representation of sparse populations. Whether these contrasting sketching strategies reliably preserve extremely rare events in clinical flow cytometry has not been systematically evaluated.

Beyond this uncertainty in sketching performance, the clustering algorithms discussed above can also show reduced performance in resolving rare populations [18]. Moreover, these algorithms are typically applied within cytometry workflows that exclude doublets and multiplets through upstream singlet gating. Interact-omics [11] is a closely related prior approach that recovers cell-interaction events from cytometry data through clustering, addressing this multiplet-exclusion gap. The method jointly clusters events using surface marker expression, light scatter properties, and the forward scatter (FSC) ratio, defined as FSC-A (area) divided by FSC-H (height). Multiplet events tend to exhibit higher FSC ratios than singlets, and clusters are subsequently classified based on the proportion of events above an FSC-ratio threshold.

This framework was primarily validated in immune-cell interaction settings in which interacting events were considerably more abundant than the CTC events examined here. Because downstream analysis depends on the events retained during sketching, it remains unclear whether extremely rare CTC events are retained sufficiently during sketching to enable reliable downstream detection after clustering and classification. It has also not been established how FSC ratio-based classification performs in CTC-containing clinical samples and whether it can be further refined using additional scatter-ratio features. To address these gaps, we compare leverage-score and geometric sketching for rare CTC recovery at both the sketch and full-pool levels. Separately, we test whether FSC ratio-based singlet/multiplet discrimination is effective in CTC-containing samples and whether incorporating an additional side scatter (SSC) ratio improves this discrimination.

3 Methodology

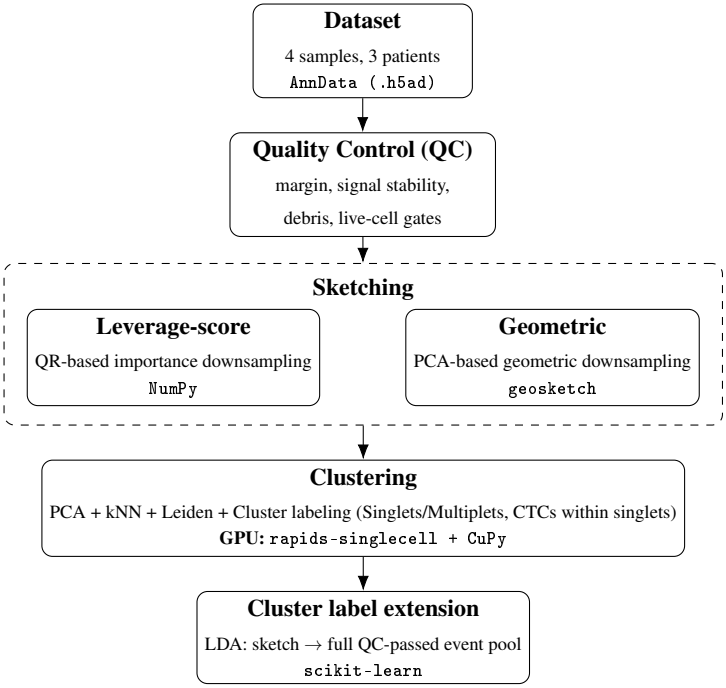


Figure 1: Overview of the analysis pipeline implemented in `Python 3.11`. After QC, samples are downsampled by leverage-score or geometric sketching, projected using principal component analysis (PCA), represented by a k-nearest-neighbor (kNN) graph, and clustered using Leiden clustering. Clusters are labeled as singlets or multiplets based on FSC ratio, with CTCs assigned only within singlet clusters, and the resulting labels are extended to the full QC-passed pool using linear discriminant analysis (LDA).

3.1 Dataset

The dataset comprises 4 conventional flow cytometry blood samples from 3 patients. These are non-spiked clinical samples, with CTCs present at naturally occurring frequencies rather than experimentally introduced or enriched. Up to two longitudinal timepoints per patient were available, spanning several months to years of clinical follow-up. Samples were acquired on a BD FACSAria

III Cell Sorter using four lasers (405, 488, 561, and 633 nm). Acquisition parameters included FSC and SSC area, height, and width measurements (FSC-A/H/W and SSC-A/H/W), together with three fluorescence channels (CD45-APC (allophycocyanin), EpCAM-FITC (fluorescein isothiocyanate), and DAPI). The subsequent analysis workflow, including QC, sketching, clustering, and cluster label extension, is summarized in Fig. 1.

3.2 Quality Control (QC)

Quality Control (QC) comprised margin dynamic-range event QC, signal stability QC, a debris gate, and a live cell gate. Margin dynamic-range event QC and signal stability QC are custom Python procedures inspired by PeacoQC [21], enabling integration into our analysis workflow.

Margin dynamic-range event QC is applied to untransformed data. Events within the upper 0.1% of each channel's range, as specified by the FCS `$PnR`, are removed as upper-range events potentially affected by saturation or clipping. For scatter channels, negative values are also removed, as they are not expected for raw scatter measurements. For fluorescence channels, negative values are retained, as they can arise from compensation spread and are required to estimate the Logicle [22, 23] width parameter (W).

Signal stability QC is applied to transformed data across all channels ($\log(1+x)$ for scatter parameters, Logicle for fluorescence markers). Events are binned by equal event count to maintain comparable statistical support despite variation in acquisition rate. Bins whose median differs from the global median by more than a threshold defined using the median absolute deviation (MAD) are removed. This captures signal drift and abrupt perturbations associated with acquisition anomalies such as clogs.

Debris gating excludes events with low forward-scatter signals, which are more likely to represent cellular fragments than intact cells. In this dataset, debris exclusion was implemented through an instrument-level FSC threshold ($\mathrm{FSC} > 5000$) applied during acquisition rather than through an additional post-acquisition gate.

Live cell gating retains events with low viability-marker signal. In this dataset, DAPI was used as the viability marker, with DAPI^- events considered live. The cutoff was determined per sample by one-dimensional kernel density estimation on the Logicle-transformed DAPI channel and set at the density valley between the dominant peak and the next sufficiently separated peak.

No further morphological cell gate is applied to avoid removing rare or morphologically atypical events, including CTCs and multiplets, before clustering.

3.3 Sketching

Sketching downsamples the QC-passed pool for each sample to 10,000 events. This sketch size was chosen to retain sufficient rare events while maintaining a manageable dataset size for downstream clustering, given QC-passed pool sizes of up to 1,361,065 events in this dataset. Baseline features comprised **raw scatter parameters** (FSC-A/H/W and SSC-A/H/W) and **Logicle-transformed fluorescence markers** (CD45-APC and EpCAM-FITC). All baseline features were standardized (z-scored and clipped to ± 10) prior to sketching.

Leverage-score sketching assigns sampling weights based on each event's contribution to the linear structure of the data. For the standardized event-by-feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n and d denote the numbers of events and features, respectively, a reduced QR decomposition is computed as $\mathbf{X} = \mathbf{QR}$, where $\mathbf{Q}$ has orthonormal columns and $\mathbf{R}$ is upper triangular.

The leverage score h_i of event i is the squared Euclidean norm of the corresponding row of $\mathbf{Q}$, and the normalized sampling weight p_i is defined as

$$h_i = \|\mathbf{Q}_{i,:}\|_2^2, \qquad p_i = \frac{h_i}{\sum_{j=1}^{n} h_j}, \tag{1}$$

where j indexes all n events. Events are sampled without replacement according to the weights p_i. Leverage scores were computed directly from the reduced QR decomposition using a custom NumPy-based implementation.

Geometric sketching aims to provide a more balanced representation of sparsely and densely populated regions of the data. For the standardized feature matrix $\mathbf{X}$,

PCA is performed using full singular value decomposition (full SVD), retaining all 8 available components (baseline features) in this study. The resulting embedding is covered by non-empty, equal-sized axis-aligned boxes,

$$\mathcal{C} = \{B_1, B_2, \ldots, B_K\}, \tag{2}$$

where B_k denotes a covering box and K is the number of non-empty boxes. For the target sketch size of 10,000 events, events are sampled approximately evenly across the covering boxes, with events selected uniformly within each box. This reduces the dominance of densely populated regions in the resulting sketch. Geometric sketching is implemented using the `Geosketch` package.

3.4 Clustering

3.4.1 Clustering algorithm

Based on the clustering benchmark in Interact-omics [11], **Leiden clustering** was selected and implemented in `rapids-singlecell` with a fixed random seed for reproducibility.

Features of the sketched events (baseline features + FSC ratio *(raw FSC-A / raw FSC-H)*) are standardized (z-scored, clipped to ± 10) and a PCA embedding is computed using full SVD. Given the low-dimensional feature space, all 9 PCs are retained to avoid discarding low-variance structure that may be relevant to rare populations. PCA therefore acts only as an orthogonal transformation, preserving Euclidean distances for kNN graph construction.

A k-nearest-neighbor (kNN) graph is then constructed ($k = 15$, brute-force search) on this embedding, followed by Leiden clustering with a fixed random seed. The Leiden resolution parameter controls clustering granularity, with higher values generally producing a larger number of smaller clusters. For each sample and sketching method, the same 10,000-event sketch was reused across three Leiden resolutions ($r = 0.5$, 1.0, 1.2) to assess how downstream results depend on clustering granularity.

3.4.2 Cluster labeling

Singlets and multiplets are distinguished following the FSC ratio-based strategy of Interact-omics [11]. Otsu thresholding is applied to the event-level FSC ratio distribution across the sketch. In this implementation, a cluster is classified as a multiplet cluster when at least 50% of its member events exceed this threshold, and as a singlet cluster otherwise.

CTCs within singlets are identified based on their relative CD45 and EpCAM expression. Rather than relying on conventional manual inspection of marker expression [11,16,17], each singlet cluster is assigned a reproducible, rule-based score derived from its median CD45 and EpCAM expression. Because the two markers are measured on different fluorescence intensity scales, cluster medians for each marker are z-scored across singlet clusters within each sample, and a score is computed as score $= z_{\mathrm{EpCAM}} - z_{\mathrm{CD45}}$.

The highest-scoring singlet cluster, representing the strongest relative high-EpCAM/low-CD45 profile, is designated as the CTC cluster. Using relative comparisons across singlet clusters within the same sample instead of an absolute threshold makes the approach less sensitive to sample-to-sample variation in fluorescence intensity while remaining consistent with the $\mathrm{CD45}^{-}/\mathrm{EpCAM}^{+}$ CTC phenotype used in this study.

3.5 Cluster label extension

Leiden cluster labels are extended from the sketched pool ($n = 10,000$) to the full QC-passed pool ($n = N$) using **linear discriminant analysis (LDA)** as in Interact-omics [11]. The clustering features (scatter channels, marker channels, and FSC ratio) are standardized using the sketch mean and standard deviation (z-score, clipped to ± 10), and a multiclass LDA classifier is trained to predict the Leiden cluster labels and is then applied to the non-sketched events. Together with the original sketch labels, these predictions assign cluster labels across the full QC-passed pool. Each event inherits the singlet/multiplet label of its assigned cluster and, within singlet clusters, its CTC label where applicable.

3.6 Evaluation

3.6.1 Pseudo ground truth

Because no orthogonal confirmation (e.g., imaging-based validation) is available, hierarchical manual gating in FlowJo (v11) is used as a pseudo ground truth. The gating hierarchy used to define CTCs and multiplets is shown in Figure 2.

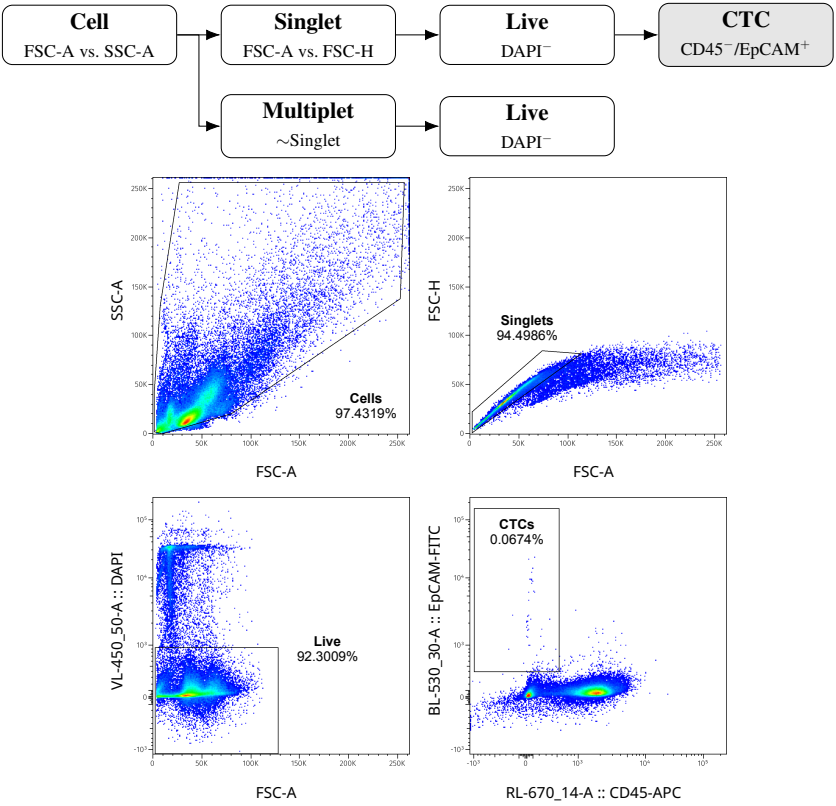


Figure 2: Manual gating hierarchy for P3 (T1), with representative gating plots for each step: an initial scatter-based cell gate, followed by singlet gating based on FSC-A vs. FSC-H, with events outside the singlet gate classified as multiplets. Live cells were subsequently selected using $DAPI^-$ within both the singlet and multiplet branches, and $CD45^-/EpCAM^+$ singlet events were used to define the CTC population.

4 Results and Discussion

4.1 CTC recovery under QC

The QC pipeline (Section 3.2) is designed to remove technical noise while avoiding the exclusion of rare or heterotypic events. To assess this, we examine whether manually gated CTCs are preserved after QC (Table 1). Overall event retention ranged from 79.8% to 83.4% across the 4 samples, whereas manually gated CTCs were retained at a higher rate, from 86.9% (73 out of 84) to 100% (553 out of 553). This indicates that the QC procedure preserved the majority of manually gated CTCs. Manually gated CTC prevalence ranged from 0.054% to 0.238% of the QC-passed pool across samples, showing that CTCs constitute a rare population even after QC.

Table 1: Per-sample event and manually gated CTC counts before and after QC, with retention rates and post-QC CTC prevalence. (P: patient; T: timepoint; prev.: prevalence)

Sample	**Events**		**CTCs**		**Prev.** %
	Raw	**Post-QC** (%)	**Manual**	**Post-QC** (%)	
P1 (T1)	659,308	526,348 (79.8)	553	553 (100.0)	0.105
P1 (T2)	864,009	712,824 (82.5)	410	387 (94.4)	0.054
P2 (T1)	1,703,663	1,361,065 (79.9)	3,556	3,244 (91.2)	0.238
P3 (T1)	139,716	116,507 (83.4)	84	73 (86.9)	0.063

4.2 CTC recovery through sketching

We compare leverage-score sketching (lev-sketch) and geometric sketching (geo-sketch) across all 4 samples and 3 Leiden resolutions ($r = 0.5, 1.0, 1.2$), yielding 24 scenarios. For each scenario, CTC recovery within singlet clusters is evaluated after extending sketch-derived cluster labels to the full QC-passed pool (Section 3.5).

Cluster-based CTC predictions exceeded manual CTC counts in most scenarios (20 of 24), although underprediction was observed in four scenarios. The frequent overprediction may reflect both the conservative nature of hierarchical manual gating (Figure 2), which requires events to pass a sequence of strict gating boundaries, and possible overestimation by the clustering approach when CTC and non-CTC events are grouped within the same cluster. Given this uncertainty and the rare-event setting, recall and F_2 are emphasized to prioritize successful CTC recovery, while precision and F_1 are also reported to capture potential overprediction. F_1 and F_2 are defined as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \qquad F_2 = 5 \cdot \frac{\text{precision} \cdot \text{recall}}{4 \cdot \text{precision} + \text{recall}}, \tag{3}$$

where F_1 balances precision and recall equally, whereas F_2 places greater emphasis on recall and is therefore more sensitive to successful CTC recovery.

Geo-sketch achieved higher full-pool mean and median values than lev-sketch across every summary metric (Table 2): mean F_1 (**0.445** vs. 0.370), mean precision (**0.410** vs. 0.338), and mean recall (**0.719** vs. 0.465). The gap was most pronounced for recall, where the median for geo-sketch exceeded the median for lev-sketch (**0.771** vs. 0.363) by more than a factor of two. This recall-driven advantage was reflected in the larger between-method difference in median F_2 score (**0.631** vs. 0.280) than in median F_1 score (**0.488** vs. 0.227).

Table 2: Mean and median CTC recovery metrics across 12 sample-resolution combinations (4 samples $\times$ 3 resolutions), for the sketch and full pool.

		Precision	**Recall**	**F_1**	**F_2**
Full pool	lev-sketch (mean)	0.338	0.465	0.370	0.409
	geo-sketch (mean)	0.410	0.719	0.445	0.541
	lev-sketch (median)	0.210	0.363	0.227	0.280
	geo-sketch (median)	0.366	0.771	0.488	0.631
Sketch	lev-sketch (mean)	0.322	0.516	0.375	0.433
	geo-sketch (mean)	0.347	0.708	0.430	0.536
	lev-sketch (median)	0.200	0.488	0.236	0.314
	geo-sketch (median)	0.391	0.717	0.483	0.618

Despite geo-sketch's higher full-pool averages, **lev-sketch** showed more extreme performance, with higher maxima for F_1 (**0.882** vs. 0.836) and precision (**0.857** vs. 0.789), but lower minima for both (F_1: **0.000** vs. 0.078, precision: **0.000** vs. 0.041) across all scenarios (Table 3). For recall, **geo-sketch** had both a higher maximum (**0.953** vs. 0.914) and a higher minimum (**0.297** vs. 0.000), although its range remained substantial. Thus, its higher mean and median recall partly reflected avoidance of very low-recall scenarios rather than uniformly stable performance.

Pronounced dependence of recall on resolution was observed more often for lev-sketch in these samples. In P1, lev-sketch's recall varied sharply across resolutions (from 0.604 to 0.119 for T1, and from 0.054 to 0.912 for T2, between $r = 0.5$ and $r = 1.0$), while geo-sketch's recall remained stable in both (swings of 0.024 and 0.015, respectively). In P3, lev-sketch's recall instead declined monotonically toward zero across resolutions (0.110, 0.068, 0.000), a modest absolute change in magnitude but a complete loss of recall. At $r = 1.2$, this corresponded to complete recovery failure in P3, despite 48 manually gated CTCs being retained in the lev-sketch (Figure 3). This suggests that rare-event retention during sketching alone does not guarantee cluster-based recovery. In P2, however, the pattern was reversed. Geo-sketch recall dropped from 0.906 to 0.297, while lev-sketch remained stable (0.897–0.914).

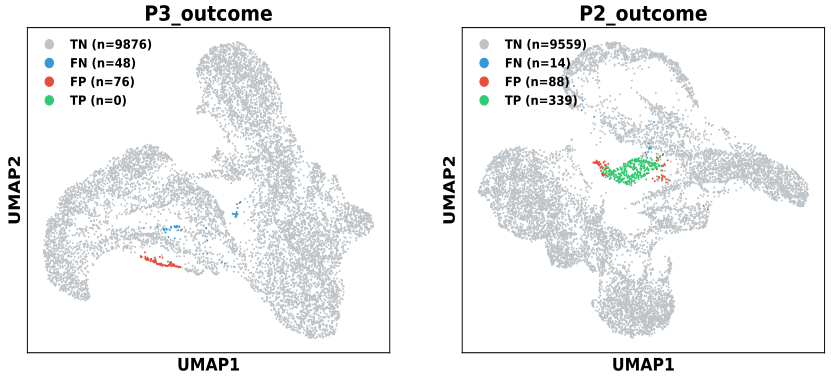


Figure 3: Comparison of sketch-level classification outcomes for P3 (left) and P2 (right) using lev-sketch at $r = 1.2$, highlighting the complete loss of recall observed in P3 compared with P2 under the same analysis settings.

Sketch-only performance is also examined as a secondary comparison to assess whether the relative performance of the two sketching methods is preserved after label extension. For precision and F_1, the relative ranking was preserved. Whichever method scored higher within the sketch also scored higher after cluster extension, in all 12 paired scenarios. Recall was the exception. In 4 of the 12 scenarios, lev-sketch had the higher sketch-only recall but was overtaken by geo-sketch after extension, with no instances of the reverse.

Table 3: CTC recovery performance of geo-sketch and lev-sketch by sample and resolution (Res.).

Sample	Sketch	Res.	TP	FP	FN	Precision	Recall	F_1	F_2
P1 (T1)	geo	0.5	420	716	133	0.370	0.759	0.497	0.627
		1.0	433	763	120	0.362	0.783	0.495	0.635
		1.2	421	575	132	0.423	0.761	0.544	0.656
	lev	0.5	334	1350	219	0.198	0.604	0.299	0.429
		1.0	66	232	487	0.221	0.119	0.155	0.131
		1.2	67	393	486	0.146	0.121	0.132	0.125
P1 (T2)	geo	0.5	369	781	18	0.321	0.953	0.480	0.684
		1.0	368	655	19	0.360	0.951	0.522	0.716
		1.2	363	489	24	0.426	0.938	0.586	0.756
	lev	0.5	21	2493	366	0.008	0.054	0.014	0.026
		1.0	353	231	34	0.604	0.912	0.727	0.828
		1.2	335	851	52	0.282	0.866	0.426	0.613
P2	geo	0.5	2939	845	305	0.777	0.906	0.836	0.877
		1.0	1021	273	2223	0.789	0.315	0.450	0.358
		1.2	964	260	2280	0.788	0.297	0.432	0.339
	lev	0.5	2910	485	334	0.857	0.897	0.877	0.889
		1.0	2965	517	279	0.852	0.914	0.882	0.901
		1.2	2957	501	287	0.855	0.912	0.882	0.900
P3	geo	0.5	57	1338	16	0.041	0.781	0.078	0.169
		1.0	43	199	30	0.178	0.589	0.273	0.403
		1.2	43	446	30	0.088	0.589	0.153	0.275
	lev	0.5	8	1873	65	0.004	0.110	0.008	0.018
		1.0	5	182	68	0.027	0.068	0.038	0.052
		1.2	0	99	73	0.000	0.000	0.000	0.000

UMAP visualizations qualitatively illustrate differences in the sampled distributions between the two sketching methods (Figure 4). In the separately

computed UMAP embeddings, geo-sketch points appeared more evenly distributed, whereas lev-sketch points formed more compact and visually distinct groupings.

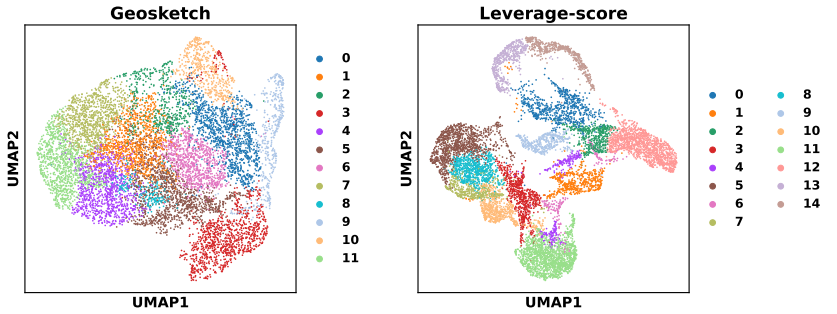


Figure 4: UMAP embeddings for geo-sketch (left) vs. lev-sketch (right), P2 at $r = 0.5$. Numbers indicate Leiden cluster labels.

4.3 Singlet/multiplet discrimination with SSC ratio

Next, we assess whether SSC ratio (raw SSC-A / raw SSC-H) improves singlet/multiplet discrimination within the QC-passed pool. For P2 at $r = 0.5$, we compare four conditions: lev-sketch with FSC ratio alone, lev-sketch with FSC and SSC ratios, geo-sketch with FSC ratio alone, and geo-sketch with both ratios. This setting was chosen because both sketching methods achieved high CTC F_2 scores (Table 3). The same sketch is reused within each sketching method, so within-method differences between the FSC-only and FSC+SSC conditions arise solely from the added clustering feature.

Adding SSC ratio had opposite effects for the two sketching methods (Table 4). For lev-sketch, it substantially improved specificity (0.789 to 0.940) and balanced accuracy (0.894 to 0.960), at a small cost to recall (0.999 to 0.980) and F_1 (0.997 to 0.989) (Figure 5). For geo-sketch, the same addition instead reduced specificity (0.720 to 0.660) and balanced accuracy (0.860 to 0.830), while recall (both near 1.0) and F_1 (0.996 to 0.995) remained comparably high.

Because singlets are treated as the positive class and substantially outnumber multiplets, precision and recall primarily characterize singlet classification

and do not directly quantify multiplet recovery. Specificity and balanced accuracy are therefore also reported to capture performance on the multiplet class. Because the manual singlet/multiplet gate is itself defined on FSC-A versus FSC-H, the strong performance of FSC ratio should not be interpreted as independent validation. The more informative comparison is whether SSC ratio provides additional discriminatory value beyond FSC ratio. Overall, SSC ratio did not provide a consistent improvement across the two sketching strategies. Specificity and balanced accuracy improved for lev-sketch but decreased for geo-sketch. This method-dependent effect suggests that the impact of SSC ratio depends on the sampled event composition and subsequent clustering structure, although these contributions were not separately evaluated here.

Table 4: Singlet/multiplet classification against the manual gate for P2 at $r = 0.5$, using FSC ratio alone or combined with SSC ratio. (Spec.: specificity; Bal. acc.: balanced accuracy)

Sketch	**Features**	**Precision**	**Recall**	**F_1**	**Spec.**	**Bal. acc.**
lev-sketch	FSC ratio	0.9937	0.9994	0.9965	0.7892	0.8943
lev-sketch	FSC + SSC ratio	0.9982	0.9798	0.9889	0.9404	0.9601
geo-sketch	FSC ratio	0.9916	0.9999	0.9958	0.7197	0.8598
geo-sketch	FSC + SSC ratio	0.9899	0.9996	0.9947	0.6601	0.8298

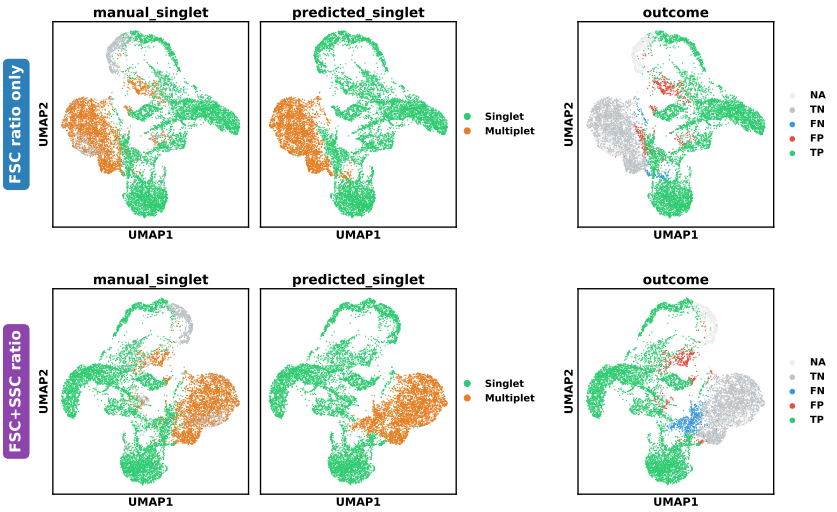


Figure 5: Manual singlet/multiplet (left), predicted singlet/multiplet (center), and classification outcome (right) for lev-sketch, P2 at $r = 0.5$, using FSC ratio only (top) and FSC and SSC ratios combined (bottom). Events without a corresponding manual singlet/multiplet label are shown as NA and were excluded from metric calculation.

5 Conclusion

In this study, we investigated whether clustering-based cytometry analysis can be extended to an extreme rare-event setting while retaining multiplets that are conventionally excluded during singlet gating. Using CTCs in clinical blood samples, we compared geometric and leverage-score sketching prior to Leiden clustering and extended sketch-derived cluster labels to the full QC-passed event pool. Geometric sketching showed higher average CTC recovery from singlet clusters, particularly in recall, and avoided the complete recovery failure observed for leverage-score sketching. However, performance remained sample- and resolution-dependent, showing that sketching strategy can substantially affect rare-event recovery. Adding SSC ratio to FSC ratio did not consistently improve singlet/multiplet discrimination in the evaluated P2 setting at $r = 0.5$, indicating that additional scatter-derived features should be evaluated in the context of the sampling and clustering strategy.

Several limitations constrain the interpretation of these findings. No orthogonal ground truth, such as imaging-based confirmation of CTCs and physical cell interactions, was available. Manual gating was therefore used as a pseudo ground truth, although this means that the clustering results are evaluated against labels derived from the same fluorescence and scatter measurements. The data were originally generated to isolate CTCs for downstream molecular profiling, rather than rare-event clustering or multiplet identification. Consequently, the acquisition panel, available markers, and reference annotations were not optimized for the present task. The dataset comprised only 4 blood samples from 3 patients, limiting the scope for inferential statistical analysis.

The present analysis selected the single most CTC-like singlet cluster based on relative CD45 and EpCAM expression. CTC labeling was restricted to singlets because $CD45^{+}$/$EpCAM^{+}$ multiplets could represent CTC-leukocyte conjugates, but their cellular composition could not be unambiguously determined with the available marker panel. Moreover, phenotypically heterogeneous CTCs may occupy multiple clusters or form a minority subpopulation within larger mixed clusters, which may lead to underestimation of CTC abundance. Conversely, assigning an entire mixed cluster as CTC-positive may overestimate CTC abundance by including non-CTC events. With more reliable ground truth, future studies should therefore evaluate multiple CTC-cluster assignments and downstream subclustering of CTC-enriched or mixed clusters.

The LDA-based extension of sketch-derived labels could also be compared with non-linear classifiers, such as kernel support vector machines [24] or random forests [25], which can capture more complex decision boundaries than linear discriminant analysis. Beyond label extension, alternative clustering strategies may also be useful for rare-event recovery. In particular, noise clustering explicitly introduces a noise class for observations that are poorly represented by the main clusters [26]. Further examination of these events could provide a complementary strategy for identifying rare events, although the noise class may also contain technical noise and debris.

Finally, the analysis was implemented in `Python 3.11` using AnnData (`.h5ad`) as the central data structure, with GPU acceleration using `rapids-singlecell` and `CuPy`. In contrast to R-based Interact-omics, the present framework operates

within the Python/AnnData ecosystem, allowing these GPU-enabled tools to support analysis of larger event pools. The use of `.h5ad` as the central data structure also provides a natural route for integrating `.h5`-based imaging flow cytometry features into the same event-level representation. Such morphological information could complement scatter and marker measurements and provide a more direct basis for distinguishing singlets from multiplets.

Acknowledgements

The present contribution is supported by the Helmholtz Association under the joint research school "HIDSS4Health - Helmholtz Information and Data Science School for Health" and the program "Natural, Artificial and Cognitive Information Processing (NACIP)". The authors used the AI language model Claude Sonnet 5 to improve the language and style of this manuscript. The authors reviewed all AI-assisted text and take full responsibility for the content.

References

[1] B. M. Szczerba, F. Castro-Giner, M. Vetter, I. Krol, S. Gkountela, J. Landin, M. C. Scheidmann, C. Donato, R. Scherrer, J. Singer, et al. Neutrophils escort circulating tumour cells to enable cell cycle progression. *Nature*, 566(7745):553–557, 2019.

[2] H. Kantarjian, A. Stein, N. Gökbuget, A. K. Fielding, A. C. Schuh, J.-M. Ribera, A. Wei, H. Dombret, R. Foà, R. Bassan, et al. Blinatumomab versus chemotherapy for advanced acute lymphoblastic leukemia. *New England Journal of Medicine*, 376(9):836–847, 2017.

[3] M. Efremova, M. Vento-Tormo, S. A. Teichmann, and R. Vento-Tormo. CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature Protocols*, 15(4):1484–1506, 2020.

[4] S. Jin, M. V. Plikus, and Q. Nie. CellChat for systematic analysis of cell–cell communication from single-cell transcriptomics. *Nature Protocols*, 20(1):180–219, 2025.

[5] K. H. Chen, A. N. Boettiger, J. R. Moffitt, S. Wang, and X. Zhuang. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, 348(6233):aaa6090, 2015.

[6] M. F. d. Oliveira, J. P. Romero, M. Chung, S. R. Williams, A. D. Gottscho, A. Gupta, S. E. Pilipauskas, S. Mohabbat, N. Raman, D. J. Sukovich, et al. High-definition spatial transcriptomic profiling of immune cell populations in colorectal cancer. *Nature Genetics*, 57(6):1512–1523, 2025.

[7] J. P. Robinson and M. Roederer. Flow cytometry strikes gold. *Science*, 350(6262):739–740, 2015.

[8] E. Lugli, M. Roederer, and A. Cossarizza. Data analysis in flow cytometry: the future just started. *Cytometry Part A*, 77(7):705–713, 2010.

[9] J. P. Robinson, G. B. Gmyrek, and B. Rajwa. Flow cytometry: advances, challenges and trends. *Bioessays*, 48(1):e70091, 2026.

[10] G. Iserentant, C. Seguin-Devaux, and J. Zimmer. Flow cytometry conjugate formation assay between natural killer cells and their target cells. In *Methods in Cell Biology*, volume 193, pages 213–228. Elsevier, 2025.

[11] D. Vonficht, L. Jopp-Saile, S. Yousefian, V. Flore, I. Simó Vesperinas, R. Teuber, B. Avanesyan, Y. Luo, C. Röthemeier, F. Grünschläger, et al. Ultra-high-scale cytometry-based cellular interaction mapping. *Nature Methods*, 22(9):1887–1899, 2025.

[12] P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs Jr, R. V. Bruggner, M. D. Linderman, K. Sachs, G. P. Nolan, and S. K. Plevritis. Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature Biotechnology*, 29(10):886–891, 2011.

[13] Z. Myrkhiyeva, K. Seitkamal, Z. Ashikbayeva, A. Taizhanova, D. Tosi, and A. Bekmurzayeva. Circulating tumor cells: overcoming challenges

of detecting a needle in a haystack. *Exploration of Targeted Anti-tumor Therapy*, 6:1002321, 2025.

[14] A. Cossarizza, H. D. Chang, A. Radbruch, S. Abrignani, R. Addo, M. Akdis, I. Andrä, F. Andreata, F. Annunziato, E. Arranz, et al. Guidelines for the use of flow cytometry and cell sorting in immunological studies. *European Journal of Immunology*, 51(12):2708–3145, 2021.

[15] E. Astakhova, A. Gubaeva, D. Naumova, A. Egorova, A. Maznina, I. Rybkina, I. Osmanov, D. Tabakov, O. Mityaeva, and P. Y. Volchkov. Spectral flow cytometry: the current state and future of the technology. *International Journal of Molecular Sciences*, 26(12):5911, 2025.

[16] S. Van Gassen, B. Callebaut, M. J. Van Helden, B. N. Lambrecht, P. Demeester, T. Dhaene, and Y. Saeys. FlowSOM: using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A*, 87(7):636–645, 2015.

[17] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015.

[18] K. Märtens, M. Bortolomeazzi, L. Montorsi, J. Spencer, F. Ciccarelli, and C. Yau. Rarity: discovering rare cell populations from single-cell imaging data. *Bioinformatics*, 39(12):btad750, 2023.

[19] Y. Hao, T. Stuart, M. H. Kowalski, S. Choudhary, P. Hoffman, A. Hartman, A. Srivastava, G. Molla, S. Madad, C. Fernandez-Granda, et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 42(2):293–304, 2024.

[20] B. Hie, H. Cho, B. DeMeo, B. Bryson, and B. Berger. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Systems*, 8(6):483–493, 2019.

[21] A. Emmaneel, K. Quintelier, D. Sichien, P. Rybakowska, C. Marañón, M. E. Alarcón-Riquelme, G. Van Isterdael, S. Van Gassen, and Y. Saeys.

PeacoQC: peak-based selection of high quality cytometry data. *Cytometry Part A*, 101(4):325–338, 2022.

[22] D. R. Parks, M. Roederer, and W. A. Moore. A new “logicle” display method avoids deceptive effects of logarithmic scaling for low signals and compensated data. *Cytometry Part A: The Journal of the International Society for Analytical Cytology*, 69(6):541–551, 2006.

[23] W. A. Moore and D. R. Parks. Update for the logicle data scale including operational code implementations. *Cytometry. Part A: the journal of the International Society for Analytical Cytology*, 81(4):273, 2012.

[24] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[25] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[26] R. N. Dave. Robust fuzzy clustering algorithms. In *IEEE International Conference on Fuzzy Systems*, pages 1281–1286. IEEE, 1993.

Closing the Loop: Bridging Deep Learning-Based Multiview Defect Detection and Information Fusion for Injection Molding Optimization

Michael Hieb[1], Volker Lohweg[2]

[1]Phoenix Contact GmbH & Co. KG.
Flachsmarktstraße 8, 32825 Blomberg
E-Mail: mhieb@phoenixcontact.com

[2]Technische Hochschule Ostwestfalen-Lippe
Campusallee 12, 32657 Lemgo
E-Mail: volker.lohweg@th-owl.de

Abstract

Visual defect detection using vision systems is a key enabler of product quality assurance in Cyber-Physical Production Systems. Recent advances in Deep Learning, Vision Transformers, and Multimodal Vision Models have significantly improved automated inspection. However, most approaches focus on defect detection and classification without supporting systematic process optimization. This paper presents a closed-loop AI framework that combines multiview defect detection, continuous learning, and semantic information fusion. The framework integrates scalable data acquisition, efficient annotation and model refinement, robust Deep Learning models, and a knowledge-driven reasoning layer for decision support. The approach is demonstrated using a ten-camera inspection system for injection-molded Sensor Actuator Cable heads. To link inspection results with process improvement, a domain-specific Knowledge Graph connects detected defects with process parameters, manufacturing knowledge, root causes, and corrective actions. By combining data-driven learning

with explicit domain knowledge, the framework enables explainable root-cause analysis and closes the loop between quality inspection and manufacturing optimization. The presented methodology provides a transferable blueprint for adaptive and knowledge-driven quality assurance in modern manufacturing.

1 Introduction

1.1 Motivation

The ongoing digital transformation of manufacturing and the emergence of Cyber-Physical Production Systems (CPPS) have significantly increased the demand for intelligent and adaptive quality assurance systems. Automated visual inspection, supported by advanced machine learning techniques, has become a key technology for maintaining product quality in high-volume manufacturing environments such as injection molding.

Injection molding processes are characterized by complex interactions between material properties, tool design, machine settings, and environmental conditions. These interactions may lead to a wide variety of visual defects, including flash, sink marks, burn marks, incomplete filling, and surface irregularities, whose root causes are often difficult to determine from visual inspection alone [10]. Additionally, Process-related measurements alone do not necessarily provide sufficient information about the actual product quality of an injection-molded component [9]. Therefore, visual inspection as well as process related data are required to detect quality deviations that otherwise would remain unobservant in pure process-related quality indicators.

Recent developments in Computer Vision (CV), Deep Learning (DL), Vision Transformers (ViTs), and Multimodal Vison Models (MVM) in both 2D and 3D modalities have substantially improved the ability of automated systems to identify, localize, and classify defects with high precision and robustness. These techniques are particularly beneficial in applications where conventional rule-based CV inspection systems exhibit limited performance or where the complexity and variability of defects make manual rule programming impractical.

Despite these technological advances, most existing visual inspection systems remain focused on defect detection, localization, classification, or segmentation. Consequently, defect predictions are often treated as isolated outputs, providing limited support for understanding the underlying causes of quality deviations or for improving manufacturing processes. While recent advances in deep learning have significantly improved inspection accuracy, systematic integration of process knowledge, domain expertise, and causal reasoning remains largely unexplored in manufacturing applications [4]. Knowledge Graphs have therefore emerged as a promising approach for connecting heterogeneous information sources, including inspection results, process parameters, expert knowledge, and technical documentation, thereby enabling explainable root-cause analysis and knowledge-driven decision support [2, 3]. Furthermore, recent studies highlight the need to combine industrial data with structured domain knowledge to move beyond defect detection toward actionable process optimization and fault diagnosis [1].

As a result, process engineers must still rely heavily on expert knowledge and manual investigations to identify root causes and derive corrective actions. To fully exploit the potential of Artificial Intelligence (AI) in manufacturing, visual inspection systems must evolve from standalone defect detectors to knowledge-driven decision support systems. This requires the integration of data acquisition, continuous learning, defect detection, domain knowledge and process information within a unified framework to close the loop. Such an approach can transform inspection results into actionable insights, enabling explainable root-cause analysis and supporting continuous process optimization.

This paper addresses these challenge by proposing a closed-loop AI framework that combines multiview Deep Learning-based defect detection with semantic information fusion. By linking detected defects to manufacturing knowledge, process parameters, causal relationships, and corrective actions, the proposed methodology closes the gap between quality inspection and process optimization, thereby contributing to more adaptive, explainable, and knowledge-driven manufacturing systems.

1.2 Research Contibution

This work contributes to the field of AI-enabled quality assurance in CPPS through four key developments. First, a scalable multiview data acquisition and annotation architecture is introduced, enabling **complete 360° inspection of injection-molded Sensor Actuator Cable (SAC) heads** using ten synchronized cameras. Second, a **continuous learning pipeline** is developed that integrates data annotation, model retraining, and performance monitoring to support the continuous improvement of inspection models in production environments. Third, a **Deep Learning-based defect detection framework** is proposed for robust multiview defect classification under challenging industrial conditions, including high product variety and class imbalance. Finally, a **Knowledge Graph-based information fusion** approach is presented that semantically links detected defects with process parameters, manufacturing expertise, causal relationships, and corrective actions. By combining these components, the proposed framework extends conventional visual inspection towards explainable root-cause analysis and closed-loop manufacturing process optimization.

2 Related Work

2.1 Visual Inspection in Manufacturing

Visual inspection plays a critical role in manufacturing quality assurance by ensuring compliance with functional and visual quality requirements. Traditionally, such inspections are carried out by trained human operators. However, manual inspection is labor-intensive, difficult to scale, and inherently susceptible to human error. For simple inspection tasks to decide between a good and a defective part, human error rates have been reported to be approximately 10^{-3} [11]. However in contrast, complex visual inspection tasks may result in defect detection rates decreasing substantially, with studies reporting missed-defect rates between 20% and 30% [12]. Moreover, factors such as fatigue, reduced attention, workload, and subjective judgment can further impact inspection performance. Consequently, automated optical inspection (AOI) systems have

emerged as promising technologies for achieving more reliable, objective, and scalable quality control in manufacturing.

Traditional AOI-systems utilize industrial cameras, controlled illumination, and rule-based image-processing algorithms to inspect products and detect defective parts. These systems typically rely on established machine vision libraries, such as those provided by MVTec or OpenCV, and employ techniques including thresholding, edge detection, blob analysis, or template matching to verify product characteristics and identify quality deviations [13, 14]. As a result, AOI-systems have become widly adopted in modern productions environments and inspect product at inline speed. While AOIs are effective for stable and well-defined inspection tasks, they often struggle with complex product variations, unpredictable defect patterns and real-world manufacturing environments.

Recent advances in AI, particularly DL, have significantly improved AOI capabilities by enabling the inspection of complex products and defect patterns that are difficult or economically impractical to address using traditional rule-based image-processing approaches.

2.2 Deep Learning for Visual Defect Detection in Manufacturing

DL has significantly enhanced the capabilities of AOI systems by enabling advanced approaches such as object detection, anomaly detection, semantic segmentation, and defect classification. In contrast to traditional rule-based image processing, DL models automatically learn discriminative visual features directly from image data, reducing the need for manual feature engineering and extensive parameter tuning. As a result, DL-based inspection systems can reliably identify complex, subtle, and highly variable defect patterns under challenging industrial conditions. These capabilities have accelerated the adoption of DL across manufacturing applications, including injection molding, where visual inspection systems are increasingly used to detect surface defects and quality deviations.

Recent studies have successfully demonstrated the application of object detection techniques for defect localization [15, 16], anomaly detection methods

for identifying previously unseen defects [17, 18], and transfer-learning-based classification models for automated quality inspection in injection molding processes [19]. These developments have established DL as a key technology for scalable, accurate, and automated quality assurance. However, most existing approaches are based on Convolutional Neural Networks (CNNs), which typically require large quantities of labeled training data. Kim et al. demonstrated that while classification accuracies exceeding 97% can be achieved with approximately 1,000 labeled samples, more than 5,000 samples are required to obtain stable model performance and minimize the risk of overfitting [22]. In industrial environments, acquiring such datasets is challenging due to the rarity of certain defect classes and the significant effort associated with data annotation. Consequently, the availability of large, balanced datasets remains a major limitation for many industrial inspection applications.

To address these challenges, recent research has increasingly shifted toward ViTs, self-supervised learning, and foundation models. Compared to conventional CNNs, transformer-based architectures have demonstrated improved generalization capabilities and can effectively leverage knowledge from large-scale pretraining. In the context of injection molding, transformer-based approaches such as Real-Time Detection Transformer have shown that generalized models trained on diverse defect and component combinations can achieve performance comparable to, or even exceeding, that of specialized models while reducing computational and maintenance effort [20]. These findings indicate that transformer architectures can improve the scalability, adaptability, and economic viability of automated visual inspection systems in highly variable manufacturing environments.

Furthermore, foundation-model-based industrial inspection has recently emerged as a promising research direction. By leveraging large-scale pretraining on vast image and vision-language datasets, foundation models provide rich semantic representations and offer considerable potential for few-shot and zero-shot defect detection [4]. Such capabilities are particularly attractive in manufacturing environments, where labeled defect samples are often scarce and expensive to acquire. Recent surveys highlight foundation models as a promising approach to overcoming data scarcity while improving generalization across different inspection tasks, products, and industrial domains [4]. However, despite these

advances, the application of genuinely multimodal foundation models that jointly integrate visual data, process parameters, expert knowledge, production data, and inspection results remains largely unexplored in injection molding. Such models could enable not only defect detection but also root-cause analysis, process optimization, and knowledge-driven decision support, representing an important direction for future research.

2.3 Knowledge Graphs in Manufacturing

In parallel to the rapid advances in AI-based visual defect detection, several approaches have been proposed to support knowledge management and defect analysis in injection molding. Traditional troubleshooting and process optimization activities are largely based on expert knowledge accumulated through years of practical experience. This knowledge is often documented in process manuals, troubleshooting guides, and best-practice repositories, yet it frequently remains fragmented, unstructured, and difficult to reuse across production sites and product variants. Example of such troubleshooting guides are provided by BASF or by the Kunststoff-Institut Lüdenscheid [10, 23].

More recently, Knowledge Graphs (KGs) have emerged as a promising approach for representing these complex relationships in a structured and machine-readable manner. By injection modeling entities such as defects, process parameters, materials, machines, and corrective actions as interconnected nodes, KGs enable the integration of heterogeneous knowledge sources, including expert knowledge, technical documentation, experimental studies, and production data [3, 25]. Consequently, KGs provide a semantic foundation for organizing manufacturing knowledge and supporting defect analysis, root-cause investigation, and decision-making processes.

Despite significant progress in both visual defect detection and knowledge representation, these research areas are still largely addressed independently. Current AI-based inspection systems primarily focus on the reliable detection and classification of defects, whereas knowledge-based approaches concentrate on capturing and reasoning about process knowledge. As a result, limited research has investigated how defect predictions from visual inspection systems

can be systematically linked to manufacturing knowledge and expert reasoning. Therefore, integrated approaches that combine automated visual defect detection with semantic knowledge representations are required for practical industrial applications.

3 Framework Architecture and Design

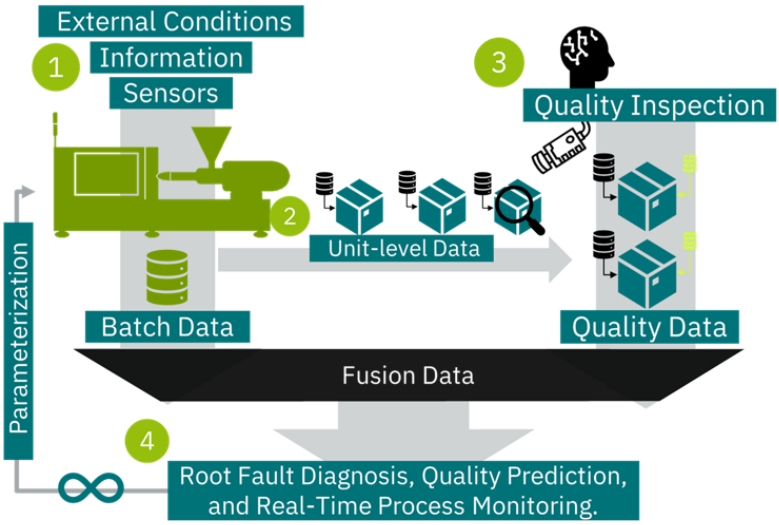


Figure 1: Proposed closed-loop AI framework for intelligent quality assurance and process optimization in manufacturing.

Figure 1 illustrates the proposed closed-loop framework for intelligent quality assurance and process optimization in injection molding. The framework consists of four interconnected stages. (1) Data acquisition captures external conditions, machine parameterization data, process signals, and sensor information generated during production. These data are aggregated at the batch level and provide a comprehensive representation of the manufacturing process. (2) Unit-level data generation associates process information with individual produced parts, enabling traceability and the creation of digital product instances. (3)

Quality inspection is performed using automated optical inspection techniques to acquire quality-related information, including visual defects and product-specific quality characteristics. The resulting quality data are linked with the corresponding unit-level and process data. Through data fusion, process parameters and inspection results are integrated into a unified information space. (4) Advanced analytics and reasoning methods leverage the fused data for root-cause analysis, quality prediction, and real-time process monitoring. The derived insights support continuous process optimization by feeding corrective actions back into machine parameterization, thereby establishing a closed-loop learning cycle that continuously improves manufacturing quality and process stability.

4 Multiview Inspection System

4.1 Industrial Use Case Sensor Actutor Cable

The industrial use case investigated in this work focuses on the manufacturing of Sensor Actuator Cable (SAC) (see Figure 2) heads using injection molding technology. SAC heads are widely used in industrial automation and connectivity systems and must fulfill strict functional, dimensional, and visual quality requirements. The manufacturing process involves the encapsulation of electrical contacts and wires with thermoplastic material to ensure mechanical protection, electrical insulation, and environmental sealing. During production, a large number of process variables, including injection pressure, holding pressure, melt temperature, mold temperature, cooling conditions, cycle times, and machine states, are continuously recorded by the injection molding machine and associated sensors.



Figure 2: Sensor Actuator Cable (SAC)

Despite extensive process monitoring, various quality defects can occur as a result of process instabilities, material fluctuations, tool wear, or environmental influences. Typical defect types include short shots (see Figure 3), flash, burn marks, sink marks, weld lines, contamination, surface scratches, discoloration, deformation, and other surface irregularities. Many of these defects directly affect both the visual appearance and functional performance of the product and cannot be reliably identified solely from process data.

Consequently, visual quality inspection remains an essential component of quality assurance. Currently, the inspection of SAC heads is performed manually by trained operators through visual examination. Since defects may occur on any side of the component, a complete 360° inspection is required, making the process time-consuming, labor-intensive, and susceptible to human variability. Furthermore, SAC heads are manufactured in multiple colors and geometrical variants, including straight and angled designs, which significantly increase the complexity of the inspection task. These challenges impose demanding industrial requirements on quality assurance systems, including high detection accuracy, low false-positive rates, real-time inspection capability, robustness against product and process variations, and complete traceability between process and quality data. As a result, SAC head manufacturing represents a highly relevant industrial application for evaluating AI-based visual inspection, multimodal data fusion, root-cause analysis, and closed-loop process optimization approaches in modern manufacturing environments.



Figure 3: Good Part vs. Short Shot Defect Part

4.2 Hardware Architecture

To enable reliable 360° visual inspection of SAC heads, a dedicated multi-camera acquisition system was developed, as illustrated in Figure 4. The hardware architecture comprises ten industrial cameras (Basler ace Classic acA1920-25gc) strategically positioned around the inspection chamber to capture all relevant surfaces and geometric features of the component without requiring mechanical repositioning. This multi-view configuration enables comprehensive inspection of both straight and angled SAC head variants while minimizing the risk of missed defects due to occlusions. To ensure consistent image quality and reduce the impact of environmental influences, the camera system is integrated with a controlled illumination setup specifically designed to enhance the visibility of surface structures, contours, and defect characteristics.

Image acquisition is initiated through a hardware-based triggering mechanism synchronized with the production process, ensuring that each SAC head is captured at a precisely defined position and time. Furthermore, all cameras operate in a fully synchronized manner, enabling simultaneous image acquisition from multiple viewing angles and guaranteeing temporal consistency across the complete image set. To enhance the robustness and long-term stability of the inspection system, the vision station is enclosed and equipped with an overpressure ventilation concept incorporating air filters and cooling fans. This design minimizes dust ingress and contamination of optical components while maintaining stable environmental conditions within the inspection chamber.

Image acquisition and processing are executed on an edge computing platform based on the NVIDIA Jetson Orin NX, which provides sufficient computational resources for real-time image processing, deep learning inference, and system

control directly within the production environment. The combination of multi-view imaging, controlled illumination, precise triggering, camera synchronization, environmental protection, and edge-based processing establishes a robust foundation for automated defect detection, traceable quality inspection, and future closed-loop process optimization in industrial manufacturing environments.

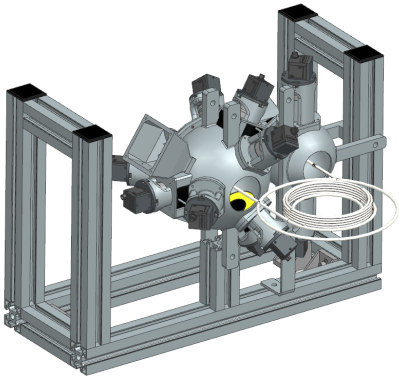


Figure 4: 360° Visual Inspection System

4.3 Dataset Generation

To evaluate the proposed AI-based inspection framework, a large-scale industrial dataset was collected from the production of injection-molded SAC heads. The dataset comprises green SAC heads and more that 26.167 defect images acquired by the ten-camera inspection system. Since each product is captured simultaneously from multiple viewing angles, the dataset provides complete 360° visual coverage and enables the detection of defects that may only be visible from specific perspectives. The collected data include both defect-free and defective products and represent the variability encountered under real production conditions, including differences in product geometries, colors, and manufacturing batches.

In total, 20 defect classes were identified and annotated by domain experts using a developed multiview anotation tool (see Figure 5). These classes

encompass a wide range of manufacturing defects, including geometric deviations, surface irregularities, contamination, flash, short shots, and other quality-relevant anomalies affecting the functionality or appearance of the SAC heads. The resulting multi-class dataset reflects the complexity of industrial visual inspection tasks, where defects can vary significantly in size, location, shape, and visual appearance.



Figure 5: Multiview Annotation Tool

A key challenge of the dataset is the presence of class imbalance, which is characteristic of industrial quality control applications. While defect-free products and certain defect categories occur frequently, several critical defect classes are only rarely observed due to stable manufacturing processes and low defect rates. Consequently, the number of samples per class differs substantially, resulting in a long-tailed distribution. Additionally, defect sample parts were produced to even the number of class imbalance.

5 Continuous Learning Pipeline

5.1 Annotation Environment and Continuous Learning Dashboard

The development of robust deep learning models for industrial defect detection requires accurately labeled training data. To support this process, a dedicated annotation environment was developed for the SAC inspection system. The

labeling workflow enables quality experts to review multi-view image data captured by the ten-camera setup and assign the corresponding defect classes. To improve efficiency, the system follows a human-in-the-loop approach in which preliminary predictions generated by trained models are presented as annotation suggestions and subsequently verified, corrected, or refined by domain experts.

In addition, a production monitoring dashboard serves as the central interface for continuous quality assurance and model improvement. The dashboard allows operators to digitize defective parts directly on the shop floor, ensuring that defect occurrences and associated production data are systematically recorded. At the same time, classification results generated by the deployed models are visualized in real time, providing transparency regarding model performance under production conditions. In cases of incorrect detections or misclassifications, operators can relabel the affected samples and correct the assigned defect classes. Furthermore, the system supports the annotation of defects across multiple camera perspectives, enabling precise localization of defect regions within the complete 360° image set. By integrating defect reporting, relabeling, multi-view annotation, and real-time model monitoring, the annotation environment establishes a continuous feedback loop between human expertise and machine learning, facilitating ongoing model refinement and improving the robustness and accuracy of defect detection in industrial production environments.



Figure 6: Production Dashboard with Continuous Learning Framework

6 Deep Learning-Based Defect Detection

6.1 Model Architecture

The model architecture is based on CNNs trained on the annotated multi-view SAC head dataset described above. Owing to the heterogeneous visual appearance of the considered defect classes, a dedicated ResNet18 model was trained for each of the 20 defect types, formulating the problem as a binary classification task between defective and non-defective samples. ResNet18 was selected due to its favorable trade-off between classification performance, computational efficiency, and real-time deployment capabilities on the NVIDIA Jetson Orin NX edge platform. Preliminary experiments with a single multi-class model showed inferior performance, particularly for rare defect categories and visually similar defect patterns, resulting in reduced defect detection rates. By training specialized models for each defect class, higher detection accuracy and improved robustness could be achieved. This modular approach further enables independent retraining and optimization of individual defect classifiers as new defect samples become available during operation.

Although Vision Transformers (ViTs) and large-scale Vision Foundation Models (VFMs) have recently demonstrated promising results in various computer vision tasks, they are not currently employed within the presented system. The primary reasons are the limited amount of defect data available for certain classes, the computational constraints of industrial edge devices, and the strong performance already achieved by CNN-based approaches.

The multi-view nature of the inspection system enables the application of ensemble learning and multi-view fusion approaches, where predictions from multiple camera perspectives are combined to improve defect detection performance. However, experiments with various fusion and ensemble strategies did not yield significant improvements over the best-performing individual ResNet18 classifiers. Therefore, the final system relies on defect-specific CNN models operating on the most informative camera views, providing the best trade-off between detection accuracy, computational efficiency, and deployment complexity.

6.2 Training Strategy and Experimental Setup

All defect classifiers (ResNet18) were trained on the annotated multi-view SAC head dataset using a supervised learning approach. To improve model generalization and reduce overfitting, data augmentation techniques such as defined rotations, brightness variations, color jitter, and geometric transformations were applied during training. Given the significant class imbalance inherent to industrial quality inspection datasets, where defect-free products substantially outnumber defective samples, class balancing techniques were employed to improve the learning of minority defect classes. The models were trained using optimized hyperparameters, including learning rate, batch size, number of training epochs, and regularization settings, which were determined empirically.

The experiments were conducted on an trained on azure Deep Learning Environment and tested on the NVIDIA Jetson Orin NX edge platform and supporting inference hardware. The dataset was divided into independent training, validation, and test sets to ensure unbiased performance evaluation. Model performance was assessed using Macro F1-Score, Balanced Accuracy, and Matthews Correlation Coefficient (MCC), as these metrics provide a robust evaluation under class imbalance and better reflect the practical performance of defect detection systems in industrial production environments.

6.3 Experimental Results

Table 1: Performance of the defect-specific classification models on the validation datasets.

Defect Class	True Positive	False Positive	False Negative	True Negative	Matthews Correlation Coefficient	Accuracy	F1 Score	Precision
1. Ridge	3298	48	28	420	0.9059	98.00%	98.86%	98.57%
2. Ridge in Parting Plane	2332	25	26	584	0.9473	98.28%	98.92%	98.94%
3. Flow Bath	192	12	4	46	0.8157	93.70%	96.00%	94.12%
4. Overfill Injection Nose Formation	348	20	7	435	0.9331	96.67%	96.27%	94.57%
5. Underfill	328	4	3	43	0.9142	98.15%	98.94%	98.80%
6. Dip Formation	1833	223	36	584	0.7662	90.32%	93.40%	89.15%
7. Deformed Logo	828	30	35	221	0.8341	94.17%	96.22%	96.50%
8. Cable Adequately Visible	2747	10	64	279	0.8737	97.61%	98.67%	99.64%
9. Cold Spots	2337	30	66	146	0.7363	96.28%	97.99%	98.73%
10. Faulty Injection Point	2516	81	67	262	0.7514	94.94%	97.14%	96.88%
11. Pollution/Dirty Spots	3552	150	113	418	0.7256	93.79%	96.43%	95.95%
12. Metal Splinter	3145	19	29	201	0.8860	98.59%	99.24%	99.40%
13. Shiny Spots	267	27	29	205	0.7849	89.39%	90.51%	90.82%
14. Burn Spots	2455	28	22	243	0.8967	98.18%	98.99%	98.87%
15. Color Errors	505	10	18	197	0.9070	96.16%	97.30%	98.06%
16. Streaking	2754	117	48	402	0.8036	95.03%	97.09%	95.92%
17. Color Dosing Error	344	1	1	359	0.9943	99.72%	99.71%	99.71%
18. Cable Crush	1336	31	18	123	0.8169	96.75%	98.20%	97.73%
19. Nut Damages	3472	92	87	530	0.8304	95.72%	97.49%	97.42%
20. Not Overmolded	396	0	0	72	1.0000	100.00%	100.00%	100.00%

7 Knowledge Graph Design

To bridge the gap between automated defect detection and actionable process optimization, a domain-specific Knowledge Graph (KG) was developed for the injection molding of SAC heads. The KG provides a structured semantic representation of manufacturing knowledge by linking visual defects with process parameters, machine configurations, materials, root causes, and corrective actions. As illustrated in Figure 7, the ontology comprises the core entity classes Product, Defect, Process Parameter, Machine, Material, Root Cause, Corrective Action, and Plain Text Knowledge extracted from technical documentation, operating procedures, and expert reports. These entities are connected through semantic relationships such as causes, affects, detected_by, resolved_by, and associated_with, enabling the explicit representation of dependencies between manufacturing conditions and product quality. By formalizing expert knowledge in a machine-readable structure, the KG transforms implicit operator expertise

into a reusable and scalable digital knowledge base. The graph integrates heterogeneous information sources, including historical production data, domain expertise, process documentation, and quality inspection results, creating a comprehensive representation of manufacturing knowledge.

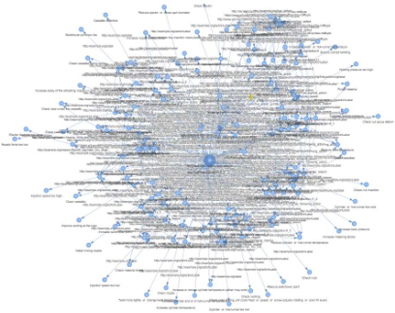


Figure 7: Knowledge Graph Representation

To operationalize the KG within the proposed framework, predictions generated by the multi-view deep learning models are mapped to corresponding defect nodes in the graph. These detected defects serve as entry points for graph traversal, enabling the automatic retrieval of relevant process parameters, potential root causes, and recommended corrective actions. Figure 8 illustrates this concept using the defect class Ridge as an example. In contrast to conventional inspection systems that rely heavily on operator experience, the proposed approach transforms visual defect detection into an explainable decision-support system for root-cause analysis and process optimization. Furthermore, the knowledge base continuously evolves by incorporating the outcomes of mitigation actions, enabling continuous learning and improvement. Initial validation in the industrial SAC production environment indicates that the integration of the KG can reduce troubleshooting effort, accelerate response times, and contribute to more stable and adaptive injection molding processes.

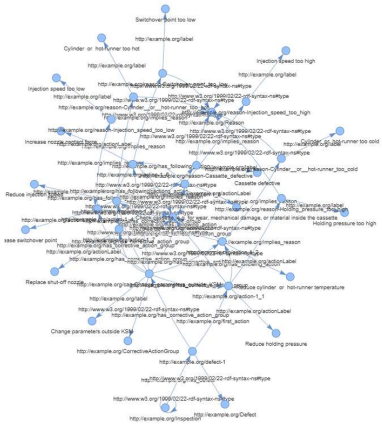


Figure 8: Knowledge Graph Representation for the Error Class Ridge

8 Discussion

The experimental results demonstrate that the proposed multiview inspection framework is capable of detecting a wide range of injection molding defects with high accuracy under real industrial conditions. Across the 20 investigated defect classes, most defect-specific ResNet18 models achieved F1-scores above 95% and Matthews Correlation Coefficients exceeding 0.80, indicating robust classification performance despite varying defect appearances and manufacturing conditions. Particularly strong results were obtained for defect classes such as Ridge, Ridge in Parting Plane, Underfill, Metal Splinter, Burn Spots, and Color Dosing Error, which achieved F1-scores close to or above 99%. These results confirm that dedicated defect-specific classifiers can effectively learn the visual characteristics of individual defect categories and provide reliable defect identification in production environments.

The study further demonstrates the advantages of a multiview inspection approach for complex injection-molded components. Since defects may only be visible from specific perspectives, complete 360° image acquisition significantly reduces the risk of missed detections and improves inspection coverage compared to conventional single-view systems. Although different ensemble

and multiview fusion strategies were evaluated, no significant performance improvements were observed compared to the best-performing individual camera views. This finding suggests that, for the investigated SAC head use case, carefully selected viewpoints contain sufficient information for reliable defect classification while reducing computational complexity for edge deployment.

A major contribution of this work extends beyond defect detection itself. Most existing AI-based visual inspection systems terminate at the classification result and provide limited support for process engineers seeking to identify defect causes and define corrective actions. The proposed framework addresses this limitation by linking defect predictions with process knowledge through a domain-specific KG. By connecting defects, process parameters, machine states, materials, root causes, and corrective actions, the framework transforms isolated defect predictions into actionable manufacturing knowledge. Consequently, the proposed approach enables explainable root-cause analysis and supports systematic troubleshooting, reducing the dependency on individual operator expertise. The integration of continuous learning mechanisms represents another important aspect of the framework. Industrial production environments are characterized by evolving products, changing process conditions, and the occurrence of previously unseen defect patterns. The developed annotation environment and feedback loop allow operators to continuously improve model performance through relabeling and retraining activities. This capability is essential for maintaining model performance over time and represents a practical prerequisite for long-term industrial deployment.

Despite the promising results, several limitations remain. The current implementation relies on independently trained binary classification models and focuses primarily on visual defect detection. While the proposed KG provides structured decision support, the causal relationships currently depend largely on expert-defined knowledge and technical documentation. Furthermore, the investigated dataset is restricted to a specific SAC manufacturing process and product family. Additional validation across different injection molding applications and product variants is required to assess the generalizability of the proposed framework. Future research should therefore focus on tighter integration of process data and visual inspection results, enabling automated causal discovery and data-driven root-cause analysis.

In addition, modern Vision Transformers, multimodal foundation models, and large-scale vision-language models offer considerable potential for combining image data, process parameters, documentation, and expert knowledge within a unified architecture. Such approaches could further reduce the dependence on manual knowledge engineering while improving reasoning capabilities and adaptability. Ultimately, the combination of multimodal AI, continuous learning, and semantic knowledge representation may enable fully autonomous closed-loop quality control systems capable of continuously optimizing manufacturing processes based on observed product quality.

9 Conclusion and Future Work

This paper presented a closed-loop AI framework for intelligent quality assurance and process optimization in injection molding. The proposed approach combines multiview visual inspection, continuous learning, deep learning-based defect detection, and knowledge-driven information fusion within a unified architecture. A ten-camera inspection system was developed to enable complete 360° inspection of SAC heads under real industrial conditions. Based on a large-scale industrial dataset comprising 20 defect classes, dedicated ResNet18 classifiers achieved high classification performance across a broad range of manufacturing defects, demonstrating the feasibility of robust AI-based visual inspection in production environments.

Beyond defect detection, the presented framework extends conventional automated optical inspection by integrating a domain-specific Knowledge Graph that links detected defects with process parameters, manufacturing knowledge, root causes, and corrective actions. This integration enables explainable root-cause analysis and transforms defect predictions into actionable insights for process engineers. Furthermore, the continuous learning pipeline establishes a feedback mechanism between operators and AI models, supporting the long-term adaptation and improvement of inspection performance. The results demonstrate that combining data-driven learning with explicit domain knowledge represents a promising step toward adaptive, explainable, and knowledge-driven quality assurance systems in Cyber-Physical Production Systems.

Future work will focus on extending the framework toward fully automated closed-loop process optimization. In particular, process data from injection molding machines, sensor systems, and manufacturing execution systems will be more tightly integrated with inspection results to enable automated root-cause analysis and data-driven process parameter recommendations. Furthermore, Vision Transformers, multimodal foundation models, and vision-language models will be investigated to improve generalization capabilities while reducing the dependency on large defect datasets. Another important research direction is the automated expansion and refinement of the Knowledge Graph through the integration of historical production records, technical documentation, and operator feedback. Finally, future studies will evaluate the proposed methodology across additional product families, production sites, and manufacturing processes to assess its scalability and general applicability. The long-term vision is the realization of self-improving manufacturing systems capable of continuously learning from production data and autonomously optimizing process quality based on observed product outcomes.

10 Acknowledgment

Generative AI tools were used to improve the language quality, grammar, and readability of this manuscript. The authors remain responsible for all scientific content and conclusions.

References

[1] J. Chen, J. Qian, X. Zhang, and Z. Song, "Root-KGD: A Novel Framework for Root Cause Diagnosis Based on Knowledge Graph and Industrial Data," *arXiv preprint arXiv:2406.13664*, 2024.

[2] J. Martinez-Gil et al., "Examining the Adoption of Knowledge Graphs in the Manufacturing Industry: A Comprehensive Review," in *Artificial Intelligence in Manufacturing*, Springer, 2024. doi: 10.1007/978-3-031-46452-2_4.

[3] Z.-W. Zhou, Y.-H. Ting, W.-R. Jong, S.-C. Chen, and M.-C. Chiu, "Development and Application of Knowledge Graphs for the Injection Molding Process," *Machines*, vol. 11, no. 2, p. 236, 2023. doi: 10.3390/machines11020236.

[4] T. Yang, L. Chang, J. Yan, J. Li, Z. Wang, and K. Zhang, "A Survey on Foundation-Model-Based Industrial Defect Detection," *arXiv preprint arXiv:2502.19106*, 2025.

[5] K. Papageorgiou, T. Theodosiou, A. Rapti, E. I. Papageorgiou, N. Dimitriou, D. Tzovaras, and G. Margetis, "A Systematic Review on Machine Learning Methods for Root Cause Analysis Towards Zero-Defect Manufacturing," *Frontiers in Manufacturing Technology*, vol. 2, 2022. doi: 10.3389/fmtec.2022.972712.

[6] D. Pietsch, M. Matthes, U. Wieland, S. Ihlenfeldt, and T. Munkelt, "Root Cause Analysis in Industrial Manufacturing: A Scoping Review of Current Research, Challenges and the Promises of AI-Driven Approaches," *Journal of Manufacturing and Materials Processing*, vol. 8, no. 6, p. 277, 2024. doi: 10.3390/jmmp8060277.

[7] Y. Cheng, Y. Cao, H. Yao, W. Luo, C. Jiang, H. Zhang, and W. Shen, "A Comprehensive Survey for Real-World Industrial Defect Detection: Challenges, Approaches, and Prospects," *arXiv preprint arXiv:2507.13378*, 2025.

[8] H. Zhou, G. Li, Y. Wang, et al., "Causal Deep Learning for Explainable Vision-Based Quality Inspection under Visual Interference," *Journal of Intelligent Manufacturing*, 2024. doi: 10.1007/s10845-023-02297-9.

[9] C. Jung, T. Magyar, S. Strommer, T. Glück, and W. Kemmetmüller, "Data-Driven Cycle-to-Cycle Quality Control of Injection Molding Using Inline Visual Inspection and Part Weight," *Journal of Process Control*, 2026. doi: 10.2139/ssrn.6880994.

[10] Kunststoff-Institut Lüdenscheid, *Guide to Surface Defects on Thermoplastic Injection-Moulded Parts*. Kunststoff-Institut Lüdenscheid, 2023. Available: `https://kimw.shop/de/produkte/2104319-`

guide-to-surface-defects-on-thermoplastic-injection-moulded-parts.html. Accessed: Oct. 10, 2026.

[11] A. D. Swain and H. E. Guttmann, *Handbook of Human-Reliability Analysis with Emphasis on Nuclear Power Plant Applications*. Sandia National Laboratories, Albuquerque, NM, USA, NUREG/CR-1278, 1983.

[12] C. G. Drury and J. G. Fox, "The Imperfect Inspector," in *Human Reliability in Quality Control*, London, UK: Taylor & Francis, 1975, pp. 11–16.

[13] MVTec Software GmbH, *HALCON: Machine Vision Software*, 2024. Available: https://www.mvtec.com/products/halcon.

[14] A. Daniilidis, G. Bradski, et al., "OpenCV Library," *ACM Computing Surveys*, 2019. Available: https://opencv.org.

[15] B. Wang and F. Huang, "A Lightweight Deep Network for Defect Detection of Insert Molding Based on X-ray Imaging," *Sensors*, vol. 21, no. 16, p. 5612, 2021. doi: 10.3390/s21165612.

[16] J. Liu, F. Guo, H. Gao, M. Li, Y. Zhang, and H. Zhou, "Defect Detection of Injection Molding Products on Small Datasets Using Transfer Learning," *Journal of Manufacturing Processes*, vol. 70, pp. 400–413, 2021. doi: 10.1016/j.jmapro.2021.08.034.

[17] V. Ketonen and J. O. Blech, "Anomaly Detection for Injection Molding Using Probabilistic Deep Learning," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, Victoria, BC, Canada, 2021, pp. 70–77.

[18] S. Lee, Y. Yun, S. Park, S. Oh, C. Lee, and J. Jeong, "Two Phases Anomaly Detection Based on Clustering and Visualization for Plastic Injection Molding Data," *Procedia Computer Science*, vol. 201, pp. 519–526, 2022. doi: 10.1016/j.procs.2022.03.067.

[19] M. El Ghadoui, A. Mouchtachi, and R. Majdoul, "Smart Defect Detection Using Transfer Learning in Injection Molding: A Comparative

Exploration Study of Deep Learning Architectures," *The International Journal of Advanced Manufacturing Technology*, vol. 133, no. 1–2, pp. 625–639, 2024. doi: 10.1007/s00170-024-13768-5.

[20] A. Pieressa, M. Sorgato, and G. Lucchetta, "Towards Efficient Defect Detection in Injection Molding with Transformer Models," in *XVII AITeM Conference*, *Materials Research Proceedings*, vol. 57, Materials Research Forum LLC, 2025, pp. 600–607. doi: 10.21741/9781644903735-70.

[21] A. Pieressa et al., "A DINOv2-Based Self-Supervised Learning Framework for Automated Detection of Surface Defects in Injection Molding," *The International Journal of Advanced Manufacturing Technology*, 2026. doi: 10.1007/s00170-026-17386-1.

[22] D. Kim, S. B. Seo, N. H. Yoo, and G. Shin, "A Study on Sample Size Sensitivity of Factory Manufacturing Dataset for CNN-Based Defective Product Classification," *Computation*, vol. 10, no. 8, p. 142, 2022. doi: 10.3390/computation10080142.

[23] BASF SE, *Injection Molding Problems in Engineering Thermoplastics: Causes and Solutions*. BASF SE, 2024. Available: `https://download.basf.com/p1/8a8082587fd4b608017fd6631d5a24b1/de/Injection-Molding_Problems_in_Engineering_Thermoplastics_-_Causes_and_Solutions_Brochure_German.pdf`. Accessed: Oct. 11, 2026.

[24] Z.-W. Zhou, Y.-H. Ting, W.-R. Jong, S.-C. Chen, and M.-C. Chiu, "Development and Application of Knowledge Graphs for the Injection Molding Process," *Machines*, vol. 11, no. 2, p. 271, 2023. doi: 10.3390/machines11020271.

[25] S. Bachhofner, K. Kurniawan, E. Kiesling, K. Revoredo, and D. Bayomie, "Knowledge Graph Supported Machine Parameterization for the Injection Moulding Industry," in *Knowledge Graphs and Semantic Web*, B. Villazón-Terrazas, F. Ortiz-Rodriguez, S. Tiwari, M.-A. Sicilia, and D. Martín Moncunill, Eds., Cham: Springer International Publishing, 2022, pp. 106–120.

Partikelverfolgung im Schmelzbad von Directed-Energy-Deposition-Bauteilen mittels In-situ-Röntgenbildgebung

T. Dommaschk[1], M. Nobach[1], C. Lauhoff[1], A. Bolender[1], H. Chapman[2,3], Y. Chen[4], F. Sket[5], P. D. Lee[2,3], T. Niendorf[1]

[1]Institut für Werkstofftechnik, Universität Kassel, Mönchebergstraße 3, 34125 Kassel, Deutschland
E-Mail: timon.dommaschk@uni-kassel.de
[2]Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, Vereinigtes Königreich
[3]Research Complex at Harwell, Rutherford Appleton Laboratory, Oxfordshire OX11 0FA, Vereinigtes Königreich
[4]Centre for Additive Manufacturing, School of Engineering, Royal Melbourne Institute of Technology, Melbourne 3000, Australien
[5]IMDEA Materials Institute, C/ Eric Kandel 2, 28906 Getafe, Madrid, Spanien

1 Motivation

Mithilfe der Additiven Fertigung lassen sich Endkonturnahe Bauteile ohne große Nachbearbeitung herstellen [1]. Gerade große Bauteile werden mit dem Directed Energy Deposition Verfahren (DED) hergestellt, bei dem Metallpulver über eine Düse in ein Schmelzblad befördert wird [2]. Hierbei kann vorlegiertes oder elementar gemischtes Pulver verwendete werden [3] . Die Verwendung elementar gemischter Pulver ist aus wirtschaftlicher Sicht vorteilhaft, da sich unter anderem ihre Zusammensetzung lokal verändern lässt [3]. Dadurch ist es leichter neue Legierungen für die additive Fertigung (AM) zu untersuchen [4]. Die Verwendung solcher Mischpulver erhöht aber die Komplexität der Schmelzbaddynamik [4]. Gerade bei der Verwendung von Elementen mit sehr unterschiedlichen Schmelzpunkten kann es zum Zurückbleiben von

unaufgeschmolzenen Pulverpartikeln im erstarrten Material kommen, die Rissinitierungsstellen darstellen und zu einem vorzeitigen Versagen der Bauteile führen [5, 6]. In der letzten Zeit rücken additiv gefertigte Titan-Refraktärmetallbildlegierungen in den Fokus [7–9]. Gerade Titan-Tantal-Legierungen stehen im Fokus, da diese neben ihrem geringen E-modul und guten Korrosionsbeständigkeit eine gute Biokompatibilität aufweisen [7, 10, 11]. Allerdings stellen die strak unterschiedlichen Schmelztemperaturen der Legierungselemente einer Herausforderung bei der Herstellung homogener Bauteile da [7]. Aufgrund dessen und des unbekannten Schmelzbadverhaltens wird die Anwendung der in situ Legierungsbildung dieses Legierungssystems behindert [12]. Es gibt bereits erste Ansätze die Schmelzbadströmung durch die Verwendung von Hochgeschwindigkeits-Röntgenbildgebungsverfahren zu untersuchen [13–16]. Allerdings fehlt ein Umfangreiches Verständnis über den Einfluss des Energieeintrags auf das Schmelzbadverhalten bei der in situ Legierungsbildung unter Verwendung von Refraktärmetallen. Deshalb wird in dieser Arbeit die in situ Legierungsbildung in einem Pulvergemisch aus Reintantal (Ta) und einer Titan-Tantal-Aluminium (Ti-Ta-Al) Legierung im DED Verfahren unter Zuhilfenahme von Hochgesbildchwindigkeits-Röntgenbildgebung untersucht. Hierbei wird die Bewegung der hochschmelzenden Tantalpartikel mithilfe eines Erkennungsalgorithmus verfolgt und mit den Ergebnissen die vorliegenden Schmelzbadströmungen und Partikelverteilungen untersucht.

2 Versuchsaufbau und Methoden

Zur Untersuchung des Auflöseverhaltens der Tantal Partikel wurden in situ Hochgeschwindigkeits-Röntgenaufnahmen in einem DED-Prozessreplikator an der ID 19 Beamline des Synchrothronrings der European Synchrotron Radiation Facility (ESRF) durchgeführt [17]. Hierbei wurden mithilfe eines Lasers Schweißbahnen aus gemischtem Ta und Ti-25Ta-5Al (gew.-%) Pulver auf Ti-6Al-4V Bauplatten gedruckt. Der Reintantalanteil im vorgemischten Pulver beträgt 1 gew.-%. Die Partikelgrößenverteilung liegt bei beiden Pulvern zwischen 60-105 µm. Während des Druckprozesses wurden Hochgeschwindigkeitsaufnahmen mithilfe einer Photron FASTCAM SA-Z mit einer Aufnahmerate von 20000 Hz getätigt. Hierbei wurden Schweißbahnen mit 350 W

und 450 W Laserleistung gedruckt. Die aufgenommenen Bilder wurden in einem Vorverarbeitungsschritt mit dem Photron FASTCAM viewer eigenen HDR-Filter verändert und Bildausschnitte, die das Schmelzbad beinhalten ausgeschnitten. Des Weiteren wurde von jedem Bild der Mittelwert der 32 das jeweilige Bild umgebenden Bilder abgezogen, wodurch Änderungen innerhalb dieser Bildfolge hervorgehoben werden. Durch die höhere Dichte und damit auch höhere Röntgenstrahlabsorption erscheinen die reinen Tantalpartikel dunkler in den Röntgenaufnahmen als die vorlegierten Partikel [18]. Zudem weisen die Tantalpartikel eine annähernd kreisförmige Geometrie auf, weshalb sich bei der Erkennung dieser Partikel die Circle Hough Transform (CHT) anbietet. Diese Methode berechnet für alle starken Helligkeitsgradienten im Bild alle möglichen Positionen eines dazugehörigen Kreiszentrums [19]. Die Partikelpositionen werden anschließend an den akkumulierten Stimmen für mögliche Kreiszentren erkannt [19]. Im Zuge dieser Arbeit wird die MATLAB-Funktion `imfindcircles` verwendet. Um die erkannten Partikel zwischen Bildern zu verfolgen wird der Kalman Filter verwendet. In jedem Zeitschritt wird hierbei der aktuelle Zustand des Systems und dessen Unsicherheit mithilfe eines mathematischen Modells bestimmt und mithilfe der aktuellsten Messwerte korrigiert [20]. Die Bestimmung der für die Verwendung benötigten Messrauschkovarianzmatrix R und die Prozessrauschkovarianzmatrix Q ist schwierig, aber diese sind essentiell für die richtige Leistungsfähigkeit das Kalman-Filters [21, 22]. Die festzulegende Prozessrausch-Kovarianzmatrix Q beschreibt wie stark der Filter dem Model traut und wird in dieser Arbeit, da keine genauen Kenntnisse vorliegen, über eine Diagonalmatrix mit einer Unsicherheit von jeweils 0,25 Pixel abgeschätzt. Zudem wird der verbesserte adaptive Kalman Filter verwendet [23]. Diese Version des Kalman-Filters macht die Filterleistung weniger empfindlich gegenüber einer nicht optimal gewählten Messrauschkovarianzmatrix R, indem zusätzliche Parameter eingeführt werden, die die Gewichtung der Modellvorhersage gegenüber den aktuellen Messungen adaptiv anpassen [23]. Als Abschätzung der Messrauschkovarianzmatrix R wurde eine Diagonalmatrix mit den Einträgen 0,25 verwendet. Das zugrundeliegende Modell nimmt an, dass sich die Partikel mit konstanter Geschwindigkeit bewegen. Der Kalman-Filter prognostiziert die Position der Partikel im nächsten Bild und kombiniert anschließend die Prognose mit den Messwerten, um die genaueste Schätzung der tatsächlichen Partikelposition zu ermitteln. Ausgehend

von den vorhergesagten Positionen der Partikel im Folgebild wird mithilfe der Ungarischen Methode die nächstgelegene detektierte Partikelposition demselben Partikel zugeordnet. Die genannten Methoden werden zudem durch Erweiterungen ergänzt, die den Umgang mit neuen, zwischenzeitlich nichterkannten und verschwindenden Partikeln festlegen. Zudem werden die Bilder in Blöcken von 150 Bildern ausgewertet, um die entstehenden Matrixgrößen zu reduzieren und damit die Leistungsfähigkeit des Algorithmus zu erhöhen.

3 Ergebnisse

Die mit den Partikelverfolgungsdaten erstellen Bilderserien zeigen, dass größere Partikel vom Algorithmus besser erkannt und verfolgt werden, als Kleinere. Zudem ist erkennbar, dass gerade Partikelüberlappungen ein Problem darstellen, da hier sich Partikelansammlungen rotationsförmig umeinander bewegen und dadurch Partikel zeitweise durch Überlappung nicht mehr erkannt werden. Dies führt zur erneuten Initialisierung einer Partikelidentifikationsnummer für Partikel, die bereits eine eigene Nummer besaßen. In Abhängigkeit des untersuchten Datensatzes beträgt die Berechnungszeit des Algorithmus 30,18 ms Bild^{-1} oder 47,23 ms Bild^{-1}. Die aus den Verfolgungsdaten berechneten Geschwindigkeitskarten sind in der Abbildung 1 dargestellt. Diese zeigen, dass die Tantalpartikel im Schmelzbad entlang zweier Vortexe bewegen, die jeweils an der Oberfläche nach außen strömen. Bei einer Laserleistung von 450 W tritt eine signifikante Zunahme der Schmelzbadabmessungen auf, begleitet von erhöhten

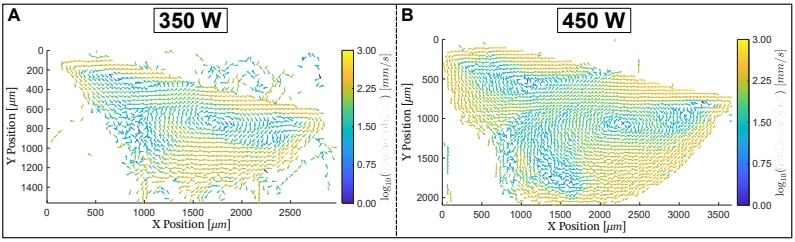


Bild 1: Strömungsrichung und Geschwindigkeit der Tantalpartikel im Schmelzbad bei verschiedenen Laserleisungen

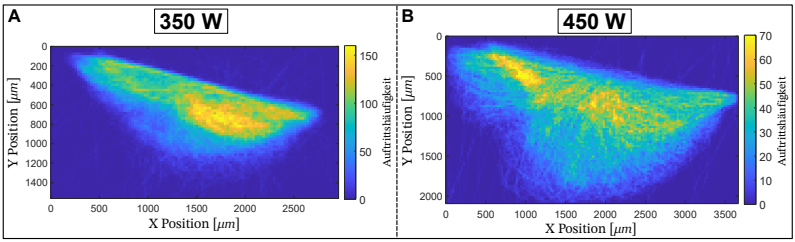


Bild 2: Auftretenshäufigkeiten der Tantalpartikel im Schmelzbad bei verschiedenen Laserleisungen

Partikelgeschwindigkeiten von bis zu 633 mm s^{-1}. Bei der Bewertung der Unsicherheit der Strömungsdaten helfen die Partikelauftretenshäufigkeitskarten die in Abbildung 2 gezeigt sind. Diese zeigen zusätzlich Bereiche im Schmelzbad in den mehr Partikel während des Druckens vorliegen. So zeichnet sich klar ab, dass bei einer Laserleistung von 350 W sich mehr Partikel im vorderen unteren Vortex aufhalten. Da die Circle Hough Transform auch die Größe jedes Partikels erkennt, ist eine Segmentierung der Verfolgungsergebnisse nach Partikelgröße möglich, wodurch die Bereiche getrennt voneinander ausgewertet werden können. Dies zeigt, dass sich bei 350 W Laserleistung kleiner Partikel eher im Zentrum des untere Vortex aufhalten, wohingegen sich die größeren Partikel in beiden Vortizes bewegen.

4 Diskussion und Zusammenfassung

Die Ergebnisse zeigen, dass der Verfolgungsalgorithmus nicht perfekt ist und Probleme bei kleinen und überlappenden Partikeln zeigt. Dies liegt zum einen an der Partikelerkennung und zum Anderen an dem zugrundeliegenden Bewegungsmodel, das hierbei nicht berücksichtigt, dass es sich bei den Daten um eine zweidimensionale Projektion des 3-dimensionalen Schmelzbades handelt und Partikel ihre Geschwindigkeit ändern können. Allerdings haben diese Effekte einen geringen Einfluss auf die erstellten Strömungsgeschwindigkeits- und Auftretenshäufigkeitskarten. Die Ergebnisse der Strömungskarten zeigen einen durch Marangoni-Konvektion geprägte Strömungsverlauf mit deutlicher Größenzunahme mit ansteigender Laserleistung. Die Partikelauftretenshäufig-

keitskarten zeigen bei 350 W Laserleistung eine höhere Konzentration kleiner Partikel im Bereich des Zentrums des unteren vorderen Vortex.

Literatur

[1] M. H. Mosallanejad, B. Niroumand, A. Aversa und A. Saboori. „In-situ alloying in laser-based additive manufacturing processes: A critical review“. In: *Journal of Alloys and Compounds*, 872, S. 159567. 2021. doi: 10.1016/j.jallcom.2021.159567.

[2] J.-P. Chen, S.-C. Xie und H. Huang. „In-situ powder mixing for laser-based directed energy deposition of functionally graded materials“. In: *Advances in Manufacturing*, 12(1), S. 150–166. 2024. doi: 10.1007/s40436-023-00460-2.

[3] P. C. Collins, R. Banerjee, S. Banerjee und H. L. Fraser. „Laser deposition of compositionally graded titanium–vanadium and titanium–molybdenum alloys“. In: *Materials Science and Engineering: A*, 352(1–2), S. 118–128. 2003. doi: 10.1016/S0921-5093(02)00909-7.

[4] D. Xu, W. Yang und P. Cao. „In situ multi-metal alloying in laser-based additive manufacturing: A concise review“. In: *Composites Part B: Engineering*, 299, S. 112443. 2025. doi: 10.1016/j.compositesb.2025.112443.

[5] N. Kang, Y. Li, X. Lin, E. Feng und W. Huang. „Microstructure and tensile properties of Ti-Mo alloys manufactured via using laser powder bed fusion“. In: *Journal of Alloys and Compounds*, 771, S. 877–884. 2019. doi: 10.1016/j.jallcom.2018.09.008.

[6] Z. Wang, M. Xie, Y. Li, W. Zhang, C. Yang, L. Kollo, J. E. Eckert und K. G. Prashanth. „Premature failure of an additively manufactured material“. In: *NPG Asia Materials*, 12(1). 2020. doi: 10.1038/s41427-020-0212-0.

[7] C. Lauhoff, T. Arold, A. Bolender, M. W. Rackel, F. Pyczak, M. Weinmann, W. Xu, A. Molotnikov und T. Niendorf. „Microstructure of an additively manufactured Ti-Ta-Al alloy using novel pre-alloyed

powder feedstock material“. In: *Additive Manufacturing Letters*, 6, S. 100144. 2023. doi: 10.1016/j.addlet.2023.100144.

[8] L.-L. Xing, C.-C. Zhao, H. Chen, Z.-J. Shen und W. Liu. „Microstructure of a Ti–50 wt% Ta alloy produced via laser powder bed fusion“. In: *Acta Metallurgica Sinica (English Letters)*, 33(7), S. 981–990. 2020. doi: 10.1007/s40195-020-01052-w.

[9] E. G. Brodie, K. J. Robinson, E. Sigston, A. Molotnikov und J. E. Frith. „Osteogenic Potential of Additively Manufactured TiTa Alloys“. In: *ACS Applied Bio Materials*, 4(1), S. 1003–1014. 2021. doi: 10.1021/acsabm.0c01450.

[10] E. G. Brodie, A. E. Medvedev, J. E. Frith, M. S. Dargusch, H. L. Fraser und A. Molotnikov. „Remelt processing and microstructure of selective laser melted Ti25Ta“. In: *Journal of Alloys and Compounds*, 820, S. 153082. 2020. doi: 10.1016/j.jallcom.2019.153082.

[11] B. Dutta und F. H. Froes. „The Additive Manufacturing (AM) of titanium alloys“. In: *Metal Powder Report*, 72(2), S. 96–106. 2017. doi: 10.1016/j.mprp.2016.12.062.

[12] Y. Liu, J. Su, Y. Li, R. Han, R. C. W. Wong, J. H. P. Hui und S. L. Sing. „In-situ alloying modulation in additive manufacturing of titanium-tantalum alloy: From melt pool modelling to process development“. In: *Materials Science and Engineering: R: Reports*, 166, S. 101082. 2025. doi: 10.1016/j.mser.2025.101082.

[13] Y. Chen, S. J. Clark, D. M. Collins, S. Marussi, S. A. Hunt, D. M. Fenech, T. Connolley, R. C. Atwood, O. V. Magdysyuk, G. J. Baxter, M. A. Jones, I. Todd, C. L. Alex Leung und P. D. Lee. „Correlative Synchrotron X-ray Imaging and Diffraction of Directed Energy Deposition Additive Manufacturing“. In: *Acta Materialia*, 209, S. 116777. 2021. doi: 10.1016/j.actamat.2021.116777.

[14] Y. Chen, S. J. Clark, Y. Huang, L. Sinclair, C. L. Alex Leung, S. Marussi, T. Connolley, O. V. Magdysyuk, R. C. Atwood, G. J. Baxter, M. A. Jones, I. Todd und P. D. Lee. „In situ X-ray quantification of melt pool

behaviour during directed energy deposition additive manufacturing of stainless steel“. In: *Materials Letters*, 286, S. 129205. 2021. doi: 10.1016/j.matlet.2020.129205.

[15] L. Aucott, H. Dong, W. Mirihanage, R. Atwood, A. Kidess, S. Gao, S. Wen, J. Marsden, S. Feng, M. Tong, T. Connolley, M. Drakopoulos, C. R. Kleijn, I. M. Richardson, D. J. Browne, R. H. Mathiesen und H. V. Atkinson. „Revealing internal flow behaviour in arc welding and additive manufacturing of metals“. In: *Nature Communications*, 9(1), S. 5414. 2018. doi: 10.1038/s41467-018-07900-9.

[16] X. Fan, T. G. Fleming, S. J. Clark, K. Fezzaa, A. C. M. Getley, S. Marussi, H. Wang, C. L. A. Leung, A. Kao und P. D. Lee. „Magnetic modulation of keyhole instability during laser welding and additive manufacturing“. In: *Science*, 387(6736), S. 864–869. 2025. doi: 10.1126/science.ado8554.

[17] I. Cowley, H. E. Chapman, S. Marussi, X. Fan, D. Rees, T. Fleming, Y. Chen, A. Rack, R. C. Atwood, M. A. Jones, S. J. Clark, C. L. A. Leung und P. D. Lee. „A second-generation apparatus for in situ synchrotron characterisation of directed energy deposition additive manufacturing with multi-modal monitoring and external forces“. In: *Materials & Design*, 262, S. 1–16. 2026. doi: 10.1016/j.matdes.2026.115495.

[18] J. H. Hubbell und S. M. Seltzer. „Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients“. National Institute of Standards and Technology, Gaithersburg. 1995. doi: 10.18434/T4D01F.

[19] T. J. Atherton und D. J. Kerbyson. „Size invariant circle detection“. In: *Image and Vision Computing*, 17(11), S. 795–803. 1999. doi: 10.1016/S0262-8856(98)00160-7.

[20] R. E. Kalman. „A New Approach to Linear Filtering and Prediction Problems“. In: *Journal of Basic Engineering*, 82(1), S. 35–45. 1960. doi: 10.1115/1.3662552.

[21] V. A. Bavdekar, A. P. Deshpande und S. C. Patwardhan. „Identification of process and measurement noise covariance for state and parameter

estimation using extended Kalman filter“. In: *Journal of Process Control*, 21(4), S. 585–601. 2011. doi: 10.1016/j.jprocont.2011.01.001.

[22] M. Perneel, I. Adriaens, B. Aernouts und J. Verwaeren. „Consistent multi-animal pose estimation in cattle using dynamic Kalman filter based tracking“. In: *Smart Agricultural Technology*, 11, S. 101014. 2025. doi: 10.1016/j.atech.2025.101014.

[23] Y. Yang und W. Gao. „An Optimal Adaptive Kalman Filter“. In: *Journal of Geodesy*, 80(4), S. 177–183. 2006. doi: 10.1007/s00190-006-0041-0.

Data-driven Identification of VSC Control Modes in Active Distribution Grids

Mohammadamin Rahimzadeh Gevari, Horst Schulte

HTW Berlin, University of Applied Sciences,
School of Engineering – Energy and Information, Control Systems Group
E-Mail: mo.a.rahimzadeh@gmail.com, schulte@htw-berlin.de

1 Introduction

Future distribution networks with high renewable penetration will likely require voltage source converters (VSCs) to operate in both grid-forming and grid-following modes to ensure secure and stable operation. To date, no universally applicable optimal ratio of grid-forming to grid-following VSCs has been established for active distribution networks. Instead, the appropriate ratio depends on network characteristics, renewable penetration levels, short-circuit strength, stability requirements, and control objectives. However, the network operator must be able to monitor the operating modes of all distributed VSCs using voltage measurements at selected network nodes. This paper presents an approach for identifying the operating modes of distributed VSCs from measurements at selected non-injection nodes.

2 Problem Formulation

2.1 VSC Control Modes

Voltage Source Converters (VSCs) provide a controllable power-electronic interface between DC energy resources and the AC grid [1, 2]. Under balanced three-phase conditions, the AC-side dynamics across the coupling filter (R_c,

L_c), expressed in a synchronous dq reference frame aligned with the point of common coupling (PCC) voltage ($v_d = 0$, $v_q = V_{\text{peak}}$), are given by [1]

$$L_c \frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} e_d - v_d \\ e_q - v_q \end{bmatrix} - R_c \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \omega L_c \begin{bmatrix} i_q \\ -i_d \end{bmatrix}, \tag{1}$$

where e_{dq} denotes the synthesized converter voltage, v_{dq} is the PCC voltage, and ω is the network angular frequency. The DC-link dynamics of capacitor C_{dc} are described by

$$C_{dc} v_{dc} \frac{dv_{dc}}{dt} = P_{in} - (v_d i_d + v_q i_q) - R_L i_{dc}^2, \tag{2}$$

where P_{in} is the input DC power and $R_L i_{dc}^2$ represents internal losses. With this orientation, active and reactive power of a three-phase system are expressed as $p_{3\phi}(t) = v_q i_q$ and $q_{3\phi}(t) = v_q i_d$, respectively, using the power-invariant Clarke-Park transformation [1].

A grid-following (GFL) converter synchronizes with the host grid using a synchronous reference frame phase-locked loop (SRF-PLL). The SRF-PLL regulates v_d toward zero while using nominal frequency $\omega_n = 2\pi f_0$ as a feed-forward term [3, 4]. The objective is to track the reference currents i_{dq}^* normally realized by two separate inner-loop decoupled PI current controllers. The outer power-control loops calculate i_{dq}^* from active and reactive power setpoints.

A grid-forming converter (GFM) synthesizes its terminal voltage magnitude and frequency without relying on a PLL in steady state, operating behind a virtual impedance [2–4]. The internal angle θ_g and quadrature reference voltage $V_{q,ref}$ are derived from frequency and voltage droop characteristics. Cascaded voltage-control loops generate current references $I_{qd,ref}$ across the filter capacitance C_{ac}, where the output control commands are constrained to $I_{sat} = 1.15$ p.u. during overcurrent conditions. The main converter parameters for GFL and GFM modes used in this study are summarized in Table 1.

Table 1: Benchmark Converter and System Parameters

Parameter	Sym.	Value	Parameter	Sym.	Value
Base Power	S_n	15 MVA	Current Loop τ	τ_s	1.0 ms
AC Voltage (L-L)	V_n	480 V	Outer PQ τ	τ_p, τ_q	0.2 s
DC Voltage	V_{dc}	1200 V	GFL Droop	$K_{droop,f}$	0.06 p.u.
Grid Frequency	f_0	60 Hz	GFM Droop P-f	$K_{d,f}$	0.01 p.u.
Filter Inductance	L_c	20.37 μH	GFM Droop Q-V	$K_{d,u}$	10^{-4} p.u.
Filter Resistance	R_c	0.77 mΩ	Current Limit	I_{sat}	1.15 p.u.
Filter Capacitance	C_{ac}	5.87 mF	Machine Inertia	H	8.0 s

2.2 Multi-Node Network Benchmark and VSC Integration

A modified IEEE 13-node distribution feeder serves as the simulation benchmark [5]. Following [6], the feeder system illustrated in Figure 1 is parameterized symmetrically to eliminate phase imbalances and focus on positive-sequence dynamic cross-couplings and disturbance propagation. The feeder operates at $U_{\mathrm{LL,eff}} = 4.16$ kV and $f_n = 60$ Hz. An aggregated transmission grid equivalent connected at Node 632 (Fig. 1) represents upstream inertia and primary frequency control. Four photovoltaic (PV) plants of identical design form a dynamic virtual power plant (DVPP) that provides fast frequency reserve (FFR), under nominal irradiance ($G = 186$ W/m^2), interface via 480 V$/$4.16 kV transformers at Nodes 634, 672, 675, and 680 to balance the nominal transmission inflow $P_{\mathrm{grid}} = 2.0$ MW [6]. These units constitute a dynamic virtual power plant (DVPP) providing fast frequency reserve (FFR) coordinated by decentralized participation filters $m_i(s)$ presented in [7]. A reconfigurable VSC connects PV1 to Node 634 (Fig. 1) operating in either:

- Grid-Following (GFL, Class 0): Synchronizes via SRF-PLL; controls decoupled active/reactive currents [1, 3].
- Grid-Forming (GFM, Class 1): Establishes AC voltage/frequency behind virtual impedance; provides immediate inertial response [2, 3].

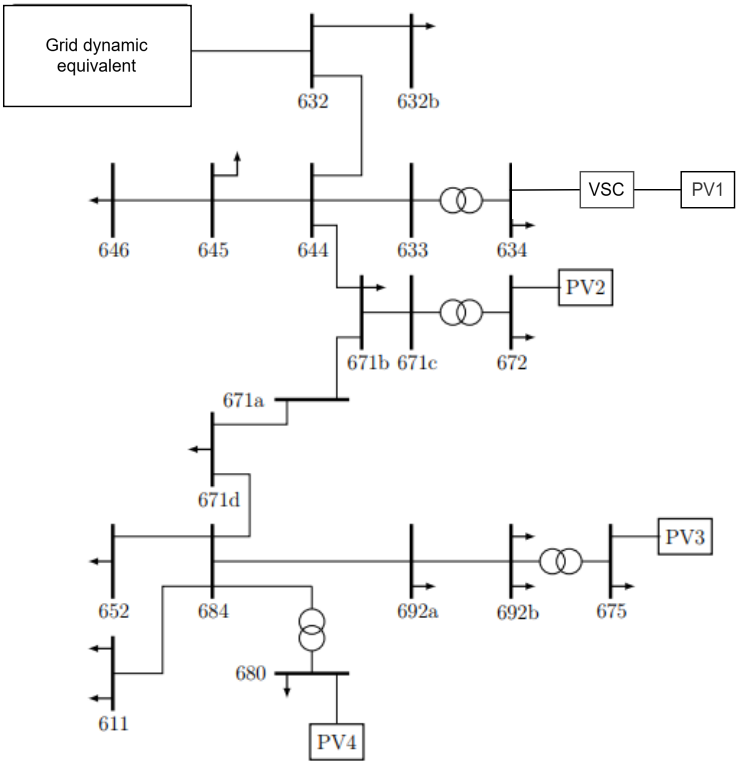


Figure 1: Modified IEEE 13-bus feeder incorporating the reconfigurable VSC interface at Node 634 and distributed PV assets.

3 Feature Extraction and Configuration

Feature extraction is carried out using a Python-based processing pipeline. The transient time-series data are evaluated over the time window $t \in [4.9, 8.0]$s, which covers the disturbance step applied at $t_{\text{step}} = 5.0, \text{s}$. The window contains a short interval before the disturbance as well as the full transient response after the disturbance [1, 2]. For each monitored bus $k \in \{634, 675, 672, 680\}$, the instantaneous three-phase active power is calculated from the corresponding phase voltages and currents with $P_k(t) = V_{k,a}(t)I_{k,a}(t) + V_{k,b}(t)I_{k,b}(t) + V_{k,c}(t)I_{k,c}(t)$. Fourteen transient features are extracted from the post-disturbance response

($t \geq 5.0, \mathrm{s}$) for each scenario. These features form the input data for the machine learning models and describe the frequency response, DVPP control actions, and electrical quantities measured at the monitored nodes.

- **Frequency Nadir (**Freq_Nadir**):** Minimum recorded frequency $f_{\min}$
- **Peak Frequency Deviation (**Max_Delta_f_Signal**):** Maximum absolute value of the frequency deviation signal, $\max(|\Delta f(t)|)$.
- **Peak DVPP Control References (**Max_dP1**–**Max_dP4**):** Maximum absolute values of the power-reference signals $dP_i(t)$ dispatched to the four DVPP units
- **Node Peak Active Powers (**Max_P_634**–**Max_P_680**):** Maximum instantaneous active power recorded at buses 634, 675, 672, and 680 following the disturbance
- **Node Minimum Voltages (**Min_V_634**–**Min_V_680**):** Minimum phase-A voltage magnitude recorded at buses 634, 675, 672, and 680 during the transient. The three phases remain symmetric, so the phase-A voltage is used as the representative voltage measurement

Three supervised machine learning approaches are used to identify the converter control mode: Decision Trees (DT), Support Vector Machines (SVM), and Random Forests (RF). Each approach uses the extracted transient features to distinguish between the two control classes, Grid-Forming (GFM) and Grid-Following (GFL) [4].

Six feature configurations are evaluated for each of the three machine learning classifiers to examine the contribution of localized and global information [2, 4]. Four configurations use measurements from individual monitored nodes. Each node-specific configuration combines the corresponding DVPP control-reference signal with the local maximum active power and minimum voltage: **Node 634:** Max_dP1, Max_P_634, Min_V_634; **Node 672:** Max_dP2, Max_P_672, Min_V_672; **Node 675:** Max_dP3, Max_P_675, Min_V_675; **Node 680:** Max_dP4, Max_P_680, Min_V_680.

A fifth configuration, denoted as global frequency, contains only the two global frequency features:

$$\mathbf{x}_{\text{freq}} = [\text{Freq\_Nadir}, \text{Max\_Delta\_f\_Signal}]\,. \tag{3}$$

The final configuration, all features, contains the complete set of extracted global, control-reference, and localized electrical features. Each classifier family is therefore evaluated with the same six feature configurations. This makes it possible to examine the effect of localized measurements, global frequency information, and the combination of multi-node features without changing the classification algorithm [4].

4 Results and Discussion

The classification results for the six feature configurations are summarized for the Decision Tree (DT), Support Vector Machine (SVM), and Random Forest (RF) classifiers. All models were evaluated on the same test set of 17 samples, allowing the effect of the feature configuration to be compared across the three classifiers.

Figure 2 compares the test accuracy and training time of the three classifier families across six feature configurations. The accuracy varies with both the classifier and the available feature information. With the complete feature set, the SVM reaches the highest accuracy at 100.0%, followed by the DT at 94.1% and the RF at 88.2%. The training-time comparison shows that the RF models require more computational time than the DT and SVM models. Across the evaluated feature configurations, the RF training times are above 100 ms, while the DT and SVM training times remain much lower. The classifier choice therefore depends on both the classification accuracy and the computational requirements for online implementation.

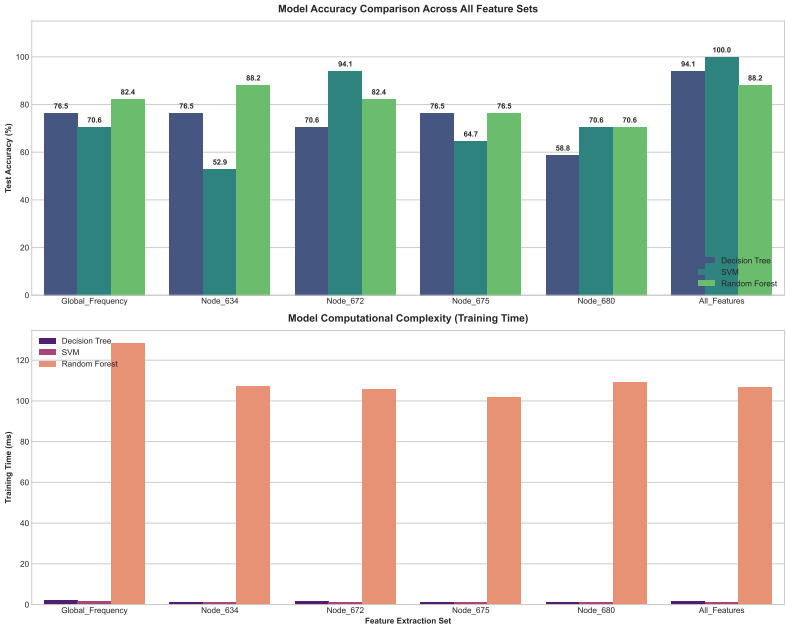


Figure 2: Comparison of test accuracy and training time for the DT, SVM, and RF classifiers across the six feature configurations.

The results show that converter control-mode classification benefits from combining transient measurements from multiple network locations with global frequency information. Among the evaluated configurations, the SVM achieves 100.0% test accuracy on the 17-sample test set, while its training time remains substantially lower than that of the RF models. The results are used in the following analysis to examine the classification framework under more complex network conditions.

References

[1] A. Yazdani and R. Iravani. “Voltage-Sourced Converters in Power Systems: Modeling, Control, and Applications”. Hoboken, NJ, USA: IEEE Press / John Wiley & Sons, Inc., 2010. doi: 10.1002/9780470551578.

[2] H. Schulte. “Advanced Control of Grid-Integrated Renewable Energy Power Plants: LMI-Based Design in the Takagi-Sugeno Framework”. IEEE Press, John Wiley & Sons, Inc., 2024. doi: 10.1002/9781119875525.

[3] J. Rocabert, A. Luna, F. Blaabjerg, and P. Rodríguez. “Control of Power Converters in AC Microgrids”. *IEEE Transactions on Power Electronics*, vol. 27, no. 11, pp. 4734–4749, Nov. 2012. doi: 10.1109/TPEL.2012.2199334.

[4] R. Ouali, M. Legry, J.-Y. Dieulot, P. Yim, X. Guillaud, and F. Colas. “Convolutional Neural Network for the Classification of the Control Mode of Grid-Connected Power Converters”. *Energies*, vol. 17, no. 24, p. 6458, Dec. 2024. doi: 10.3390/en17246458.

[5] W. H. Kersting. “Radial Distribution Test Feeders”. In: *Proc. IEEE Power Engineering Society Winter Meeting*, vol. 2, pp. 908–912, 2001. doi: 10.1109/PESW.2001.917000.

[6] M. Wypych. “Dezentrale Regelungsstrategien zur Deckung der Bedarfe an Momentanreserve in Übertragungsnetzen mit Erzeugern aus Verteilnetzen”. Master’s thesis, Hochschule für Technik und Wirtschaft Berlin, Berlin, Germany, Jan. 2026.

[7] H. Schulte, J. Brunner, M. Wypych, Frequency Control of Low-inertia Power Grids via DVPP-based support from Distribution Networks: An IEEE 13-Node Feeder Case Study, In Proc. IEEE PES ISGT, Innovative Smart Grid Technologies Europe 2026, 19 to 22 Oct. 2026, Budapest, Hungary.

[8] W. Du and Y. Chen. “Machine Learning Applications for Power System Dynamic Security Assessment”. *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4531–4541, 2021.

Evaluation of Training Curricula for Collaborative 3D Object Detection Under Communication Latency

Felix Glatzel, Stefan Schütte, Timo Osterburg, Torsten Bertram

Institute for Control Theory and Systems Engineering, TU Dortmund University
E-Mail:
{felix.glatzel|stefan.schuette|timo.osterburg|torsten.bertram}@tu-dortmund.de

Abstract

Collaborative perception can improve 3D object detection by combining observations from multiple vehicles, but communication latency introduces temporal inconsistencies between the exchanged sensor measurements. This work investigates how the organization of latency exposure during training affects the robustness of collaborative 3D object detection under delayed information.

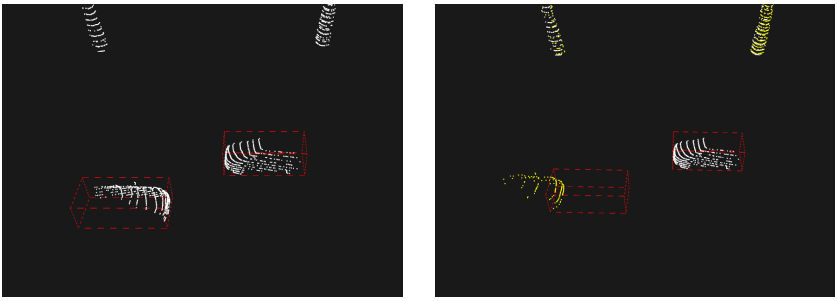
We evaluate six latency curricula using a collaborative perception model based on OpenPCDet. Lidar measurements from two vehicles are fused at the point-cloud level, while the temporal offset between asynchronous measurements is provided to the detector through a frame-based temporal encoding. To enable a fine-grained analysis of communication delay, a custom CARLA dataset is recorded at 20 Hz. The evaluated curricula comprise three fixed-delay schedules that use the same discrete delay levels in varying orders and three latency-model curricula based on exponentially distributed communication delays. After the respective training curricula, all models are additionally subjected to a common fine-tuning stage using the same stochastic latency distribution.

Before fine-tuning, the ordering of the fixed-delay curricula produces distinct robustness profiles, with models generally performing better at delays encountered later during training, while two latency-model curricula involving an unchanged

latency distribution achieve the highest overall mean average precision (mAP). The common fine-tuning stage substantially changes these relative performances. All fixed-delay models improve, with one fixed-delay model achieving mAP improvements of 0.11 to 0.15 across the evaluated delays, whereas the two models previously trained with the unchanged latency distribution decrease in performance. These results show that the temporal organization of latency exposure has a lasting influence on the learned robustness profile and that subsequent training with a common latency distribution does not uniformly remove these differences or improve performance.

1 Introduction

Collaborative perception can enhance the field of view of autonomous vehicles by allowing traffic participants to exchange sensor information. However, communication introduces effects such as latency, which can cause temporal inconsistencies between the observations of collaborating vehicles. Figure 1 illustrates the effect of communication latency on collaborative perception. Collaborative perception systems therefore need to remain robust under delayed information. The concept of curriculum learning [1] has been applied to collaborative perception under communication latency [2]. However, to the best of our knowledge, the effect of different latency-training curricula on the resulting model robustness has not been investigated in detail. This work addresses this gap by evaluating different training curricula for collaborative 3D object detection under communication latency. To enable a fine-grained analysis of delay effects, we use a custom dataset generated with the CARLA simulator [3] at a temporal resolution of 20 Hz. The evaluated model is based on OpenPCDet [4] and combines lidar measurements from two collaborating vehicles. The resulting models are evaluated across fixed communication delays both before and after a common additional training epoch using a stochastic latency distribution.



(a) Synchronous setting.

(b) Asynchronous setting.

Figure 1: Comparison of ideal and latency-affected scenarios; Two cars drive in opposite directions on a country road. The left car is the ego vehicle, which receives sensor data from the right, collaborating vehicle. The red boxes show the ground truth bounding boxes of the vehicles. White points are lidar points without latency, while the yellow points are latency-affected. Figure 1a shows the ideal case, where all points are sensed in the corresponding bounding boxes. Figure 1b shows the same scenario with latency, the ego vehicle receives a delayed recording of itself. As the yellow points show in relation to the bounding box, the car has driven further, so that the points do not align with the actual car anymore. Static objects, such as the trees near the upper edge of the image, are not affected by latency, as they do not move between the two recordings.

2 Related Work

Collaborative perception has been investigated extensively for automotive applications, including approaches based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) information exchange [5–7]. Several works additionally consider communication latency and asynchronous sensor information as part of the collaborative perception problem [2, 6, 11–13].

2.1 Curriculum Learning in Collaborative Perception

Curriculum learning [1] has been applied to collaborative perception before. Lei et al. [2] progressively introduce communication latency during training to improve robustness to delayed information and subsequently fine-tune the model using an exponential latency distribution. Other works use curriculum-based training to expose collaborative perception models to additional communication-related disturbances, such as communication interruptions [15] or spatial mis-

alignment between collaborating vehicles [14]. In contrast, the work at hand focuses specifically on how different temporal organizations of latency exposure affect the resulting model robustness.

2.2 Datasets for Collaborative Perception

Several datasets for collaborative perception with lidar data are available, such as V2X-Sim [8], DAIR-V2X [9], and OPV2V [10]. However, these datasets provide temporal resolutions of only 5 or 10 Hz, resulting in comparatively coarse increments for representing communication latency. To enable a finer-grained analysis of latency effects, we therefore use a custom dataset recorded at 20 Hz.

3 Experimental Setup

The evaluated model is assembled from components of the OpenPCDet library [4]. The model consists of a PointPillars backbone [17], feeding an OpenPCDet BaseBEV backbone and a CenterPoint detection head [18]. Based on this model, we implement a collaborative perception experiment, the procedure of which is illustrated in Figure 2.

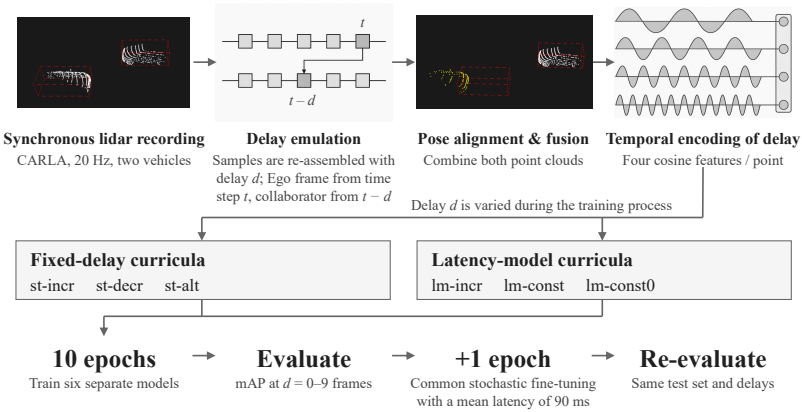


Figure 2: Overview of the experimental procedure. The lidar data of two vehicles is recorded synchronously in a simulation, and later combined with a configurable delay. All lidar points are augmented with a temporal encoding, depending on their delay. During training, the delay is configured according to the respective curriculum. An evaluation is performed on the test set with fixed delays. An additional fine-tuning stage is applied to all models after their respective curricula, using a common latency distribution, before the fine-tuned models are evaluated again.

3.1 Dataset

For the means of the work at hand, a custom dataset is generated using the CARLA simulator [3]. The data base for the dataset consists of 1,000,000 samples, which are generated across 20 different scenarios. The scenarios are designed to cover a wide range of traffic situations, including urban and rural environments, intersections, and highways. Traffic participants include cars, trucks, buses, vans, motorcycles, and, in depending on the scenario, pedestrians and bicycles. The traffic participants navigate the environment using the CARLA autopilot. Each scenario consists of a number of start positions for vehicles and pedestrians, with a probability for each position to spawn an object at all. Among the vehicles, two collaborating vehicles are placed, which are equipped with a lidar sensor. The sensor is configured to 32 channels, and a range of 200 meters, incorporating a vertical field of view of $-25°$ to $+15°$ and up to 1,200,000 points per second.

The collaborative samples are recorded at a constant temporal sampling interval of $\Delta t = 0.05\,\mathrm{s}$. Following this, we express communication latency either as a continuous time delay τ in seconds or as an integer frame delay d. A delay of d frames therefore corresponds to a temporal offset of $d\Delta t$.

The recording of the data is performed synchronously, meaning that both collaborating vehicles record their data at the same simulation step. Parallel to the data recording, the ground truth bounding box and orientation of the objects in the scene is recorded, which is used for the training and evaluation of the model. After a warmup period to allow the vehicles to settle, 410 frames are recorded consecutively for each scenario, allowing for a later variation of the communication delay between the two collaborating vehicles. From these recorded frames, 400 samples are usable as samples, with the first 10 frames being used for the emulation of communication delay for the first samples.

The used dataset is subsampling the generated data with a pre-defined, ramdomly-created sequence of sample numbers, which is used for all curricula to ensure comparability. The training set consists of 25,000 samples from the 14 training scenarios, while the validation set consists of 3,600 subsamples from the 3 validation scenarios, and the test set consists of 3,600 subsamples from the 3 held-out test scenarios.

3.2 Collaborative Point Cloud Fusion

The collaboration is performed at the unprocessed sensor-data level. A data loader combines the pre-generated samples of the two collaborating vehicles. At loading time, the frame delay described in Section 3.1 is applied according to the respective dataset configuration. The point cloud of the collaborating vehicle is transformed using the recorded sensor poses and combined with the synchronous point cloud of the primary ego vehicle. The resulting concatenated point cloud is subsequently forwarded to the detection pipeline. No intermediate or BEV-level features are exchanged between the vehicles.

Communication latency is simulated by selecting an earlier lidar frame from the collaborating vehicle while taking the primary ego vehicle sensor data from the

current time step. Consequently, the combined point cloud may contain measurements acquired at different times. The recorded sensor poses are used when assembling the combined point cloud, such that the spatial relationship between the two sensor observations is taken into account before concatenation. This spatial alignment accounts for the change in sensor poses between acquisition times, but does not compensate for the motion of independently moving objects. Based on this configuration, the different training curricula described in Section 3.4 are applied.

3.3 Temporal Encoding of Asynchronous Measurements

To provide the detector with information about the temporal offset between the combined measurements, each point is augmented with a temporal encoding. Instead of absolute sensor timestamps, the temporal offset is represented by the discrete frame delay d introduced in Section 3.1. Ego-vehicle points receive an offset of zero, whereas points from the collaborating vehicle receive the applied delay d.

The frame offset is encoded using a sinusoidal positional encoding and appended to the lidar sensor features of each point before PointPillars processing. Four temporal features are added per point, using cosine functions at four frequencies. No additional agent-specific fusion is performed in the subsequent processing stages.

3.4 Evaluated Curricula

For the evaluated curricula, the training data is modified to simulate different communication latencies. The curricula can be split into two groups, *fixed-delay* curricula and *latency model* curricula. Across all curricula and all epochs, the Adam optimizer [19] is used with a constant learning rate of 5×10^{-4}, to increase comparability. Each epoch is implemented as an individual training run, with the model weights being saved after each epoch and loaded for the next epoch. Due to this, all optimizer states are reset after each epoch.

3.4.1 Fixed-Delay Curricula

The group of fixed-delay curricula varies the delay in discrete steps, either increasing, decreasing or alternating the delay over time, with each epoch using one constant delay d between the two collaborating vehicles. Three curricula are evaluated, *st-incr*, *st-decr*, and *st-alt*. They differ only in the ordering of the same delay levels: increasing, decreasing, or alternating between low and high delays (Table 1).

Table 1: Delay d_e for fixed-delay curricula in each epoch e.

Epoch e	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
d_e (st-incr)	0	1	2	3	4	5	6	7	8	9
d_e (st-decr)	0	9	8	7	6	5	4	3	2	1
d_e (st-alt)	0	9	1	8	2	7	3	6	4	5

For each epoch, a new random seed is used to ensure the shuffling of the training data across the epochs. To increase comparability, the seed is connected to the delay, focussing the variation across the epochs onto the different curricula.

3.4.2 Latency Model Curricula

In contrast to the fixed-delay curricula, the latency-model curricula sample an individual communication latency for each training sample. While more realistic approaches to communication latency have been proposed [16], we use a simplified exponential latency model, similar to the fine-tuning used in [2]. For each training sample, a latency $\tilde{\tau}$ is drawn from an exponential distribution with mean μ. The corresponding probability density function is

$$f_{\tilde{\tau}}(t;\mu) = \begin{cases} \frac{1}{\mu}\exp\left(-\frac{t}{\mu}\right), & t \geq 0, \\ 0, & t < 0. \end{cases}$$

and subsequently limited to a predefined maximum delay $\tau_{\max}$,

$$\tau = \min(\tilde{\tau}, \tau_{\max}).$$

Thus, sampled latencies exceeding $\tau_{\max}$ are assigned the maximum delay rather than being discarded or resampled. For all samples evaluated in this work, $\tau_{\max} = 0.5\,\mathrm{s}$ is used, corresponding to a maximum frame delay of $d_{\max} = 10$ at the dataset's temporal resolution of 20 Hz. This sampling procedure is applied independently to every training sample.

Since the dataset is available only at discrete time steps, the sampled continuous latency is converted into an integer frame delay according to

$$d = \left\lceil \frac{\tau}{\Delta t} \right\rceil.$$

The resulting delayed sample from the collaborating vehicle is taken from d frames before the current ego-vehicle sample.

Three latency-model curricula are evaluated: *lm-incr*, *lm-const*, and *lm-const0*. They differ in how the parameters of the sample-wise latency distribution are configured over the course of training. The corresponding parameter values for each epoch are given in Table 2. All latency-model curricula use the same random seed order as *st-incr*.

Table 2: Mean μ_e of the continuous exponential latency distribution used in each training epoch e, in seconds.

Epoch e	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
μ_e (lm-incr)	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
μ_e (lm-const)	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
μ_e (lm-const0)	0	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09

3.4.3 Common Latency-Distribution Fine-Tuning Stage

Similar to the curriculum used in [2], a final fine-tuning stage is applied to all models after their respective curricula. After their respective ten-epoch curricula, all six models are trained for one additional epoch using an exponential latency distribution with mean latency $\mu^* = 0.09\,\mathrm{s}$. This corresponds to the maximum mean latency used in the latency-model curricula and provides a common fine-tuning stage for all training strategies.

4 Results

For the evaluation of the different curricula, the mean average precision (mAP) is calculated for each delay. The calculation of the mAP is performed using the calculator from [20], following a center-distance based assignment method in the 2D plane. Across all object classes, four individual values mAP_c are calculated, corresponding to the four center-distance thresholds of 0.5 and 1.0, 2.0 and 4.0 meters. These values are then averaged to obtain the overall mAP:

$$\mathrm{mAP} = \frac{1}{4} \sum_{c \in \{0.5,1.0,2.0,4.0\}} \mathrm{mAP}_c \tag{1}$$

The test set is evaluated with different constant delays, considering both moving and static objects, as well as varying speeds of the collaborating vehicles. For the measurement of each delay, the model is evaluated on the complete test set.

4.1 Effect of the Training Schedule Before Common Fine-Tuning

We first evaluate the models immediately after their respective training schedules, before the fine-tuning stage. This comparison isolates the performance differences produced by the different latency schedules. Figure 3 shows the mAP obtained at fixed evaluation delays from 0 to 9 frames.

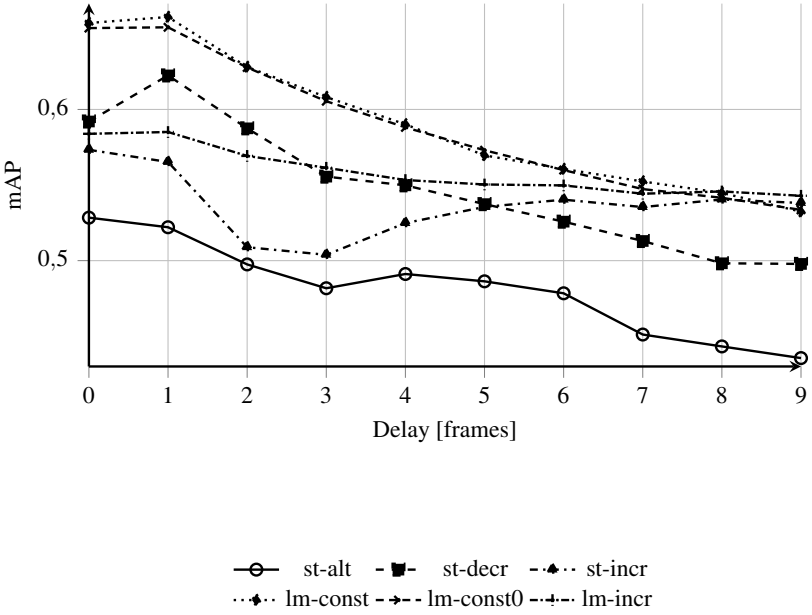


Figure 3: mAP as a function of delay for the evaluated curricula before fine-tuning.

lm-const and *lm-const0* show the highest mAP overall, ranging between 0.66 at delays of 0 and 1 to 0.53 at a delay of 9, with *lm-const* scoring slightly higher than *lm-const0* at most delays. Notably, *lm-const* not only performs good at a delay of 0, but it even performs better than *lm-const0*. This happens, even though *lm-const* never sees a latency of 0 during training, and *lm-const0* does see this delay in the first epoch.

For the fixed-delay curricula, *st-incr* starts with an mAP of 0.57 at a delay of 0, which drops to 0.50 at a delay of 3, then increases again to 0.54 at a delay of 8 and 9. *st-decr* shows a high mAP at low delays, with the highest mAP of 0.62 at a delay of 1, followed by a gradual decline. The mAP decreases with increasing delay. *st-alt* shows the lowest mAP among all models, with values ranging from 0.53 at a delay of 0 to 0.44 to a delay of 9. It shows a higher mAP at delays 4 and 5 than at a delay of 3.

4.2 Effect of Common Latency-Distribution Fine-Tuning

We next evaluate the same models after the additional fine-tuning epoch. Changes between Figure 3 and Figure 4 indicate how strongly the performance differences induced by the previous schedules persist after common stochastic latency exposure. Figure 4 shows the mAP over fixed evaluation delays with the fine-tuned models.

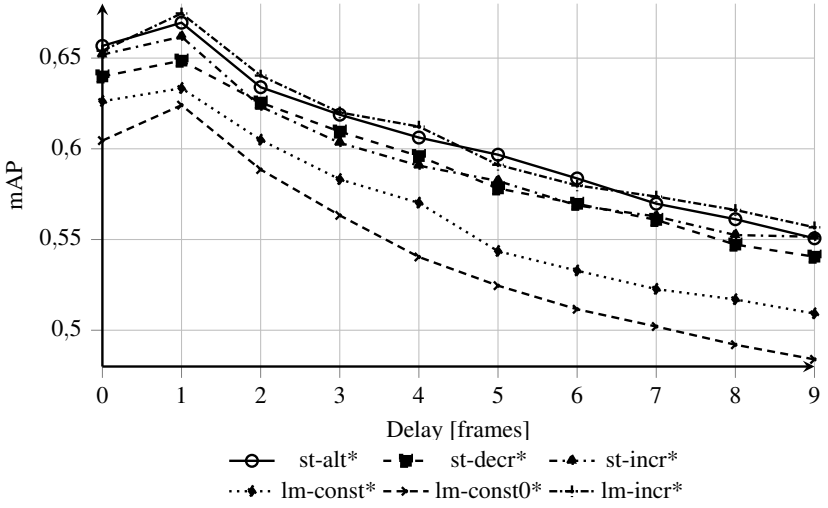


Figure 4: mAP as a function of delay for the evaluated curricula after fine-tuning.

The additional epoch substantially changes the robustness profiles of the models trained with fixed-delay schedules. Their performance increases, so that their mAP after fine-tuning is higher than the mAP of *lm-const** and *lm-const0**, across all delays. The effect is especially pronounced for *st-alt*, which scores the worst mAP before fine-tuning but substantially improves after the additional training. *st-alt** achieves the highest overall mAP at a delay of 0 and its best performance is at a delay of 1, ranging from 0.66 to 0.67. Then its mAP gradually decreases to 0.55 at a delay of 9. *st-decr** and *st-incr** also improve, reaching maximum mAP values of 0.65 and 0.66, respectively, at a delay of 1, before gradually decreasing to 0.54 and 0.55 at a delay of 9. *lm-incr**

shows a similar trend, reaching the overall highest mAP of 0.67 at a delay of 1, before gradually decreasing to 0.56 at a delay of 9. *lm-const** and *lm-const0** perform worst among the fine-tuned models, being outperformed by all fine-tuned fixed-delay curricula. Figure 5 shows the change in mAP, from before to after fine-tuning.

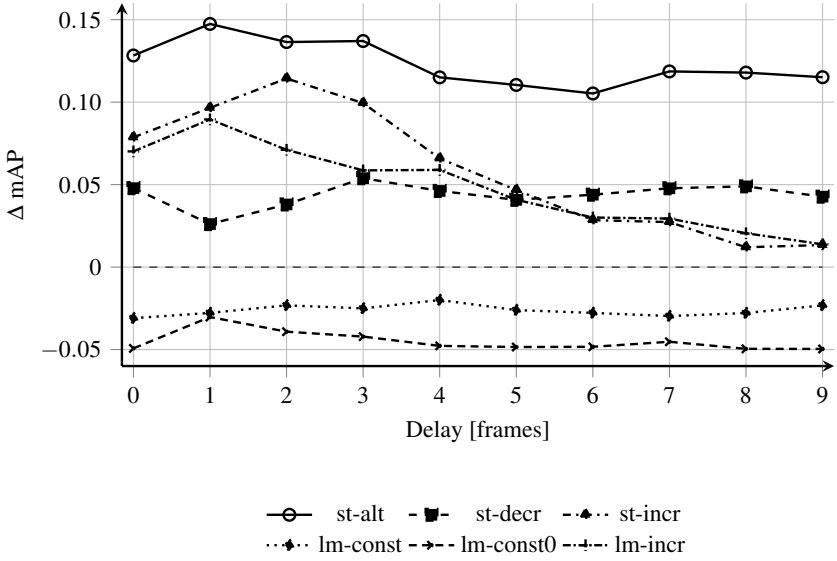


Figure 5: Change in mean average precision (Δ mAP) after the common additional epoch with the latency distribution as a function of evaluation delay. Positive values indicate an improvement over the corresponding model before the additional epoch.

The effect of fine-tuning depends strongly on the preceding training strategy. *st-alt* shows the largest improvement, reaching a mean increase of 0.12 mAP. *st-decr* and *st-incr* also show improvements, but to a smaller extent, with mean gains of 0.04 and 0.06 mAP. The effect on the latency-model curricula is considerably different. Among them, the only improvement is observed for *lm-incr* with a mean inrease of 0.05 mAP. In contrast, *lm-const0* and *lm-const* decrease in mAP across all evaluated delays, with *lm-const0* showing a mean decrease of 0.04 mAP and *lm-const* showing a mean decrease of 0.03 mAP.

4.3 Discussion

The results indicate that the temporal organization of latency exposure during training has an impact on the resulting robustness profile of the trained model.

For the fixed-delay curricula before fine-tuning, the ordering of the same set of delays produces different robustness profiles. It is observable that the models perform better on the delays they encounter later in training, which is particularly visible for *st-decr* and *st-incr*. Even though *st-alt* performs worst among the models, the effect is also visible, with a higher mAP at delays 4 and 5 than at delay 3. Under the assumption that samples with larger delays are more difficult to learn, this is noteworthy, as models can perform better on harder-to-learn samples than on easier-to-learn samples, depending on the order they learn them. However, the use of a constant learning rate across all epochs may increase this effect, as learned weights may be *overwritten* by later training stages, instead of being fine-tuned by using a decreasing learning rate.

When it comes to the fine-tuned models, compared performance shows a different picture. All fine-tuned models achieve their highest mAP at a delay of 1 frame, which also represents the most frequently sampled delay during fine-tuning. For larger delays, the mAP gradually decreases across all models, indicating a reduction in detection performance with increasing temporal offsets. For the delays of 0, 5, and 6, the model formerly scoring the lowest mAP, *st-alt**, now scores the highest mAP values among all. In contrast, the models formerly scoring the highest mAP, *lm-const** and *lm-const0**, now score the lowest mAP values over all delays. *lm-incr** now performs best among the latency-model curricula, even though it performed worst among them before the additional epoch.

The decrease in mAP after the additional epoch for *lm-const** and *lm-const0** is noteworthy, as it indicates that the additional epoch does not provide a uniform performance gain across all curricula. The two models have trained on the same latency distribution for 10, or 9 epochs, respectively. The performance before fine-tuning suggests that the models have learned the detection task well. Still, they are exposed to the additional epoch and the constant learning rate. Therefore, a possible explanation for the decrease in mAP is that the additional

epoch leads to an overfitting to the scenarios used in the training set. This explanation indicates the possibility that the earlier models score a higher mAP, even before the last epoch before fine-tuning. However, the examination of models after every epoch exceeds the scope of the work at hand.

5 Conclusion and Outlook

This work investigated how the temporal organization of latency exposure during training affects collaborative 3D object detection under communication delay. The results show that training curricula containing the same delay levels can lead to substantially different robustness profiles depending on their ordering. Before the common fine-tuning stage, the fixed-delay models tend to perform better at delays encountered later in training, suggesting an influence of the most recently encountered latency conditions. At the same time, the constant latency-model curricula achieve the highest mAP values across the evaluated delays before fine-tuning.

After the common fine-tuning stage, all models exhibit similar delay-dependent performance trends, but differences in their absolute mAP values remain. The effect of fine-tuning depends strongly on the preceding curriculum. The fixed-delay curricula benefit substantially from the additional latency-distribution training, particularly the alternating schedule, whereas the models already trained for most or all epochs with the same latency distribution exhibit a reduction in mAP. Consequently, latency robustness depends not only on the set or distribution of delays observed during training, but also on how this exposure is organized over time. The results demonstrate that additional training with a common latency distribution does not necessarily improve detection performance.

Future work may investigate whether this behavior results from recency or forgetting effects and whether similar behavior occurs across different detector architectures, datasets, and latency distributions.

Acknowledgement

This work has been funded by the Federal Ministry of Research, Technology and Space (BMFTR) via the 6GEM+ transfer hub under grant number 16KIS2412. The contribution at hand was created with assistance from the OpenAI ChatGPT 5.6 and Microsoft 365 Copilot language models.

References

[1] Y. Bengio, J. Louradour, R. Collobert and J. Weston. “Curriculum Learning”. In: Proceedings of the 26th Annual International Conference on Machine Learning (ICML). Association for Computing Machinery. 2009. doi: 10.1145/1553374.1553380.

[2] Z. Lei, S. Ren, Y. Hu, W. Zhang and S. Chen. “Latency-Aware Collaborative Perception”. In: European Conference on Computer Vision (ECCV). 2022. doi: 10.1007/978-3-031-19824-3_19

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “CARLA: An Open Urban Driving Simulator”. In: Proceedings of the 1st Annual Conference on Robot Learning. 2017.

[4] OpenPCDet Development Team. “OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds”. `https://github.com/open-mmlab/OpenPCDet`. 2020.

[5] Q. Chen, S. Tang, Q. Yang and S. Fu. “Cooper: Cooperative Perception for Connected Autonomous Vehicles Based on 3D Point Clouds”. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). 2019. doi: 10.1109/ICDCS.2019.00058

[6] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng and R. Urtasun. “V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction”. In: European Conference on Computer Vision (ECCV). 2020. doi: 10.1007/978-3-030-58536-5_36.

[7] Y. Hu, S. Fang, Z. Lei, Y. Zhong and S. Chen. “Where2comm: Communication-Efficient Collaborative Perception via Spatial Confidence Maps”. In: Advances in Neural Information Processing Systems (NeurIPS). 2022.

[8] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen and C. Feng. “V2X-Sim: Multi-Agent Collaborative Perception Dataset and Benchmark for Autonomous Driving”. In: IEEE Robotics and Automation Letters (Volume 7, Issue 4). 2022.

[9] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan and Z. Nie. “DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022.

[10] R. Xu, H. Xiang, X. Xia, X. Han, J. Li and J. Ma. “OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication”. In: 2022 International Conference on Robotics and Automation (ICRA). 2022. doi: 10.1109/ICRA46639.2022.9812038.

[11] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang and J. Ma. “V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer”. In: European Conference on Computer Vision (ECCV). 2022. doi: 10.1007/978-3-031-19842-7_7.

[12] J. Wang and T. Nordström. “Latency Robust Cooperative Perception using Asynchronous Feature Fusion”. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV). 2025.

[13] W. Wang, J. Wang, Y. Deng and G. Tan. “BEVSync: Asynchronous Data Alignment for Camera-based Vehicle-Infrastructure Cooperative Perception Under Uncertain Delays”. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 14. 2025. doi: 10.1609/aaai.v39i14.33611.

[14] J. Yoo, Z. Feng, T.-Y. Pan, Y. Sun, C. P. Phoo, X. Chen, M. Campbell, K. Weinberger, B. Hariharan and W.-L. Chao. “Learning 3D Perception

from Others' Predictions". In: International Conference on Learning Representations (ICLR). 2025.

[15] S. Ren, Z. Lei, Z. Wang, M. Dianati, Y. Wang, S. Chen and W. Zhang. "Interruption-Aware Cooperative Perception for V2X Communication-Aided Autonomous Driving". In: IEEE Transactions on Intelligent Vehicles, vol. 9, no. 4. 2024. doi: 10.1109/TIV.2024.3371974.

[16] H. Zhou, E. Xie, W. Shao, D. Gao, M. Dong and J. Zhang. "EI-Drive: A Platform for Cooperative Perception With Realistic Communication Models". In: IEEE Internet of Things Journal, vol. 12, no. 13. 2025. doi: 10.1109/JIOT.2025.3552554.

[17] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom. "PointPillars: Fast Encoders for Object Detection from Point Clouds". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

[18] T. Yin, X. Zhou, and P. Krähenbühl. "Center-based 3D Object Detection and Tracking". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021.

[19] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: International Conference on Learning Representations (ICLR). 2015.

[20] T. Osterburg, F. Albers, C. Diehl, R. Pushparaj, and T. Bertram. "HiLO: High-Level Object Fusion for Autonomous Driving using Transformers". In: 2025 IEEE Intelligent Vehicles Symposium (IV). 2025.

Dieser Tagungsband enthält die Beiträge des 36. Workshops Computational Intelligence und präsentiert aktuelle Forschungsarbeiten zu KI-Methoden in Ingenieur- und Steuerungssystemen. Themen reichen von Anwendungen in der Mechanik, Energietechnik und Medizin, über Computational Intelligence in der Automatisierungs- und Regelungstechnik, Large Language Models bis hin zu Objekterkennung und Tracking.