

Karlsruher Schriften  
zur Anthropomatik

Band 70



Jürgen Beyerer, Tim Zander (Eds.)

**Proceedings of the 2024 Joint  
Workshop of Fraunhofer IOSB  
and Institute for Anthropomatics,  
Vision and Fusion Laboratory**



Jürgen Beyerer, Tim Zander (Eds.)

**Proceedings of the 2024 Joint Workshop  
of Fraunhofer IOSB and Institute for  
Anthropomatics, Vision and Fusion Laboratory**

Karlsruher Schriften zur Anthropomatik

Band 70

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe  
erschienenen Bände finden Sie am Ende des Buchs.

# **Proceedings of the 2024 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory**

by  
Jürgen Beyerer, Tim Zander (Eds.)

## Impressum



Karlsruher Institut für Technologie (KIT)  
KIT Scientific Publishing  
Straße am Forum 2  
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark  
of Karlsruhe Institute of Technology.  
Reprint using the book cover is not allowed.

[www.bibliothek.kit.edu/ksp.php](http://www.bibliothek.kit.edu/ksp.php) | E-Mail: [info@ksp.kit.edu](mailto:info@ksp.kit.edu) | Shop: [www.ksp.kit.edu](http://www.ksp.kit.edu)



*This document – excluding the cover, pictures and graphs – is licensed  
under a Creative Commons Attribution 4.0 International License  
(CC BY 4.0): <https://creativecommons.org/licenses/by/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>*

Print on Demand 2025 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489 (Schriftenreihe)

ISSN 2510-7259 (Tagungsband)

ISBN 978-3-7315-1423-7

DOI 10.5445/KSP/1000179597







## Preface

In 2024, the annual joint workshop of the Fraunhofer Institute of Optronics, System Technologies, and Image Exploitation (IOSB) and the Vision and Fusion Laboratory (IES) of the Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT) was once again hosted in a Black Forest house near Triberg.

For a week from the 28th of July to the 3rd of August, the PhD students of both institutions delivered extended reports on the status of their research and participated in heated discussions on topics ranging from computer vision, industrial production, optimisation, control theory, security to large language models. Most results and ideas presented at the workshop are collected in this book in the form of detailed technical reports. This volume provides a comprehensive and up-to-date overview of some of the research programs of the IES Laboratory and the Fraunhofer IOSB.

The editors thank Jonas Vogl and Zeyun Zhong for their efforts resulting in a pleasant and inspiring atmosphere throughout the week. We would also like to thank the doctoral students for writing and reviewing the technical reports, as well as for responding to the comments and suggestions of their colleagues.

*Jürgen Beyerer & Tim Zander*



# Contents

<b>Preface</b> .....	I
Jürgen Beyerer and Tim Zander	
<b>Quantification metrics for the maturity of production processes</b> .....	1
Negar Arabizadeh	
<b>Uncertainty Quantification and Monte Carlo and RNG</b> .....	19
Ali Darijani	
<b>Localization of latent influences in cyber physical systems</b> .....	33
Frank Doehner	
<b>Outdoor Semantic Occupancy Mapping with 3D-LiDARs</b> .....	49
Raphael Hagmanns	
<b>Shape Models in EOT for Radar-ITS</b> .....	71
Longfei Han	
<b>Multi-stage process modeling using Gaussian Processes</b> .....	85
Saksham Kiroriwal	
<b>Evaluating the Realism of Lateral Movement Modeling</b> .....	97
Nicole Neis	
<b>Towards Quantifying Simulated Image Sensor Data</b> .....	121
Anne Sielemann	

**Advancing Adaptive Learning in Dynamic Environments** ..... 149  
Benedikt Stratmann

**Registration as a warping method for optotypes** ..... 179  
Oliver Veitl

**Attention Few-Shot Learning for Vehicle Classification** ..... 201  
Stefan Wolf

**Causality-Driven AI for Manufacturing Systems** ..... 217  
Shahenda Youssef

# Quantification metrics for measuring the maturity of production processes

*Negar Arabizadeh*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
negar.arabizadeh@kit.edu

## Abstract

Having metrics to evaluate progress and verify solution approaches is a significant advantage in all fields of science and engineering. Measuring the maturity of a manufacturing process, considering the possible changes that can be made is essential for assessing progress toward a more mature production process. Suppose that we add sensors or actuators to the manufacturing process or implement software changes, such as modifying process parameters, adjusting control policies, or altering initial conditions. In that case, we need formal definitions for measuring the process's maturity after these modifications. To address this, we applied definitions from control theory to develop formal mathematical metrics to evaluate process maturity. We introduce the concepts of *Elucidability*, *Forcability*, and *Supervisability*, which are ostensive, interpretable, and based on quantifiable or estimable measures. These metrics provide a formalized approach to assessing process maturity by integrating both technical and economic considerations. Finally, we showcase the effectiveness of our definitions by assessing the maturity of an Electric Arc Furnace system.

# 1 Introduction

Having metrics to measure the maturity of a production process is essential as it gives an overview of the status of the process compared to the desired goals of the mature process. "Metric is a vehicle for understanding and providing a standard unit to be used as a basis for comparison among like items or characteristics." [22]. In [22] the metric plane is explained, which describes five levels of maturity of a process and the type of metric required for each of them. The maturity metrics developed in the literature are categorized in qualitative and quantitative metrics. Qualitative metrics such as [19] have been developed in which a maturity assessment model was developed to measure the maturity of social and technical systems. Furthermore, due to the lack of knowledge on how to design theoretically sound and widely accepted maturity assessment models, the paper discussed the typical phases of maturity model development and application by taking a design science research perspective. In the literature, different fields have developed quantitative measures for their values of interest. In [4] a quantitative definition of risk for safety and security was developed. In [3] a metric, Bayesian decision theory, for evaluating the relative worth of additional knowledge within the measurement context was tried. In this paper we introduce a mathematical maturity model which relates the features of the internal dynamic of the system to the quality of the eventual production process.

# 2 Related Work

In the literature, there exist models of maturity, but no mathematical formulas to measure it [36]. In [19] the definition of maturity was released and the concept of maturity assessment models for social systems was contextualized. There are different models to measure the maturity of the system. In the literature, sometimes the words maturity and readiness are used in a similar way, but there are some differences between them [23].

### **2.0.0.1 Qualitative Maturity Models**

In [21] a five-level qualitative maturity model was defined for software systems processes that measures maturity in an ordinal scalar. In [34], the authors tried to use some methods to develop the improvement system assessment tool (ISAT). One of the methods that are used examines the approach, deployment, and evaluation review to assess the maturity of key management systems' ability to create a high-performing organization. They described the toll that was used to assess the maturity and effectiveness of the enterprise performance measurement system. Their methods are not mathematical, but qualitative, based on measuring the maturity of different levels such as the Approach, Deployment, Study, and Refinement.

[38] proposes a theoretical model to measure the degree of readiness of an innovation process in critical industries such as the aerospace-defense sector in Colombia, this model is applicable to decision-making processes, development centers, and the country's leading sectors. It offers a theoretical basis and a tool to be applied in the decision-making process to help minimize the inherent risks to the innovation process in the sector. First, a review of the state of the art and a scientometric analysis was carried out to determine factors subsequently validated by experts, which gave rise to variables with a high degree of relevance in generating innovative products or services. These were modeled through structural equations that consider variables covering the innovation cycle, which consider market, industry, and technology criteria, which minimize risk until the absorption of the innovative product is integrated. This work also contributes with a tool for decision making through a process flow chart, which will allow measuring from an external perspective, the maturity level of the process against the sector in order to define resource allocation, prioritization levels, and the state of the technology or process contextualized in local conditions.

### **2.0.0.2 Having degree for controllability and observability**

There are different methods in the literature, such as calculating the rank of the controllability matrix, the controllability Gramian matrix [10] and the PBH [8] that indicates whether the system is controllable or not. These methods give us a binary answer.

Methods like [10, 7, 9, 31, 35] measure the degree of controllability of a system not in a binary way, without providing any physical meaning. So, mostly measures based on the minimum input energy are used. [20] proposed an infinite set of possible scalars to measure the degree of controllability and observability for linear dynamical systems. The physical interpretation of these measures is the minimum input energy to regulate a system from the initial condition in a finite time interval. If the system requires less energy, it is considered more controllable. In an application, it was used to optimize certain structural parameters to optimize the proposed measures of quality. [12] proposed a method to measure the degree of controllability of a system with unstable modes by the physical meaning of measuring the minimum energy required to change the state from an arbitrary initial condition to an arbitrary final condition in the positive time domain. As the measure is related to the initial and final condition, the appropriate initial and final condition that correspond to the control objective should be used.

[17, 16] proposed Gramian-based methods to determine optimal sensor and actuator locations for a descriptor system. [28, 29] extended these methods to the nonlinear system. In [11] the degree of controllability is defined as the minimum input energy required to make the final state zero in the presence of persistent external disturbance. It is made up of Gramian controllability and sensitivity matrices. In [24], VCS (volumetric controllability score) and AECS (average energy controllability score) are defined, which are two mathematical definitions based on the Gramian controllability matrix and are the unique solutions of a convex optimization problem to produce the degree of measurement of controllability of a system. In other words, the VCS of each state node indicates its importance in enlarging the controllability ellipsoid, and the AECS of each state node indicates its importance in steering overall states to a point in the unit sphere. This metric measures the energy input needed to move the state from one condition to another. The characteristic function of the VCS is based on the log-determinant of the controllability Gramina matrix, and the AECS is the trace of the inverse of this matrix. By the rule of duality, they also give some answers for the observability.

[37] introduced a measure for the degree of controllability of a linear system with the random initial condition with disturbance, by solving the fixed-time



expected minimum energy transfer control problem and the use of the method in the turbine. In [27], they produced a new data-driven method to measure the degree of controllability of a system.

### **2.0.0.3 Maturity vs Readiness Level**

Readiness is "the state of being ready or prepared, as for use or action." [6]. "Readiness of a system, technology, or integration implies how ready it is to be deployed on a numerical scale"[25].

"Maturity is the characterization of physical development that is quantified by readiness" [25]. With regard to these definitions, readiness and maturity might seem similar, but they have some differences. However, in some literature research, they are used synonymously [1, 30]. "The lower the maturity, or readiness, of an incoming technology," for example, is a quote from [32], which is an example of using these two words similarly. [30] states that readiness and maturity are often used synonymously, but the authors argue that there is a difference. A mature system or technology may not be ready for a particular environment. The authors will address the readiness level of the technology, subsystem, and system, which is not the maturity level. In addition, we assert that the system that is ready might not be mature and has the potential to be improved.

Technology Readiness Levels (TRLs) developed by NASA is commonly used to assess the maturity of a particular technology and the consistent comparison of maturity between different types of technology [15]. It is a 9-level evaluation method with limited quantified meaning to evaluate the readiness of a technology to be used in a space program.

Integration Readiness Level (IRL) appeared to measure the readiness when technologies interact[14]. Having TRL and IRL for each subsystem is not sufficient to define the readiness of the system [26]. The system readiness level (SRL) is defined as a function of the TRL and IRL of the subsystems that are components of the whole system [26] [18]. [18] quantified the SRL, they produced some properties that are necessary to consider to be able to measure SRL of a system. The authors presented four desirable mathematical properties

that are necessary properties for an SRL model which are the building blocks of a mathematical framework for defining the readiness metric of the system.

### 3 Definitions for measuring the maturity

#### Elucidability

For the process instance, Elucidability which is similar to observability is approached by defining the minimum observation time  $\tau$  that is necessary to achieve a certain precision of estimation for the state  $s$  and parameters  $\theta$ . We find the probability of having the estimation error less than a constant precision  $\eta_\theta$  and  $\eta_s$ . The internal states of the process can be estimated from the time series  $X_T, Y_T$  during the finite observation in duration  $T$ , so  $X_T := \{x(0), \dots, x(T)\}$ ,  $Y_T := \{y(0), \dots, y(T)\}$ . The formula for Elucidability is:

$$E(T, \eta_s, \eta_\theta) := \Pr(\|s(T) - \hat{s}(X_T, Y_T)\| < \eta_s \wedge \|\theta - \hat{\theta}(X_T, Y_T)\| < \eta_\theta | s(0)) \quad (3.1)$$

In which  $\hat{s}(X_T, Y_T)$  is the estimation of state  $s(T)$  at time  $T$ , with time series  $X_T$  and  $Y_T$  in duration  $T$ . It is the same for  $\hat{\theta}(X_T, Y_T)$  which is the estimation of  $\theta$ . If we do not have the time duration  $T$  we can find it with the smallest time to achieve a specific error  $\eta_s$  and  $\eta_\theta$ , with a probability larger than  $1 - \delta$ ,

$$\tau(\eta_s, \eta_\theta, \delta) := \min\{T > 0 | E(T, \eta_s, \eta_\theta) > 1 - \delta\}. \quad (3.2)$$

#### Forcability

Forcability  $F$  that is similar to controllability is defined to measure the ability to steer the values of  $y$  and  $s$  through  $x$  toward the target values  $\check{y}$  and  $\check{s}$ . The most effective way to steer the process is to establish a closed-loop control to react instantaneously to the dynamic answers of the process. To quantify Forcability of a process instance with a given controller  $\pi$ , the probability of being able to force the process after a time  $T$  into the neighborhood of target values  $\check{y}$  and  $\check{s}$  is

defined as follows,

$$F_{\pi}(T, \eta_s, \eta_y) := \Pr(\|y(T) - \check{y}\| < \eta_y \wedge \|s(T) - \check{s}\| < \eta_s \mid s(0), x(t) = \pi(Y_t, \check{y}, \check{s})) \quad (3.3)$$

In close loop control,  $X_T$  is determined from the control policy  $\pi$ . In the case that we do not have  $y(T)$  and  $s(T)$  and we need to estimate them, we can replace the estimated  $\hat{y}(T)$  and  $\hat{s}(T)$  with the real value of them, so the modified formula for Forcability is:

$$F_{\pi}(T, \eta_s, \eta_y) := \Pr(\|\hat{y}(T) - \check{y}\| < \eta_y \wedge \|\hat{s}(T) - \check{s}\| < \eta_s \mid s(0), x(t) = \pi(Y_t, \check{y}, \check{s})) \quad (3.4)$$

If we implement an optimal controller our formula will be changed to

$$F(T, \eta_s, \eta_y) := F_{\pi^*}(T, \eta_s, \eta_y) \quad (3.5)$$

If the time duration  $T$  is not fixed for a given process instance, we can use the probably approximately correct (PAC) [33] in a way to find the minimum time we need to have a specific error  $\eta_s$  and  $\eta_y$  with the probability larger than  $1 - \delta$ ,

$$\tau(\eta_s, \eta_y, \delta) = \min\{T > 0 \mid F(T, \eta_s, \eta_y) > 1 - \delta\}. \quad (3.6)$$

## Supervisability

Supervisability  $S$  is defined to measure the maturity of the quality of the product instance. The formula is written to find the probability of having quality in the desired set  $Q$  when considering that we have set our controller  $\pi$  and the input signal  $x$ , the state  $s$ , and the parameter  $\theta$  are in their desired set. The formula for Supervisability is defined as:

$$S(P_i^j) := \Pr(q \in Q \mid x \in X_{adm}, s \in S_{adm}, \theta \in \Theta_{adm}, T < T_{max}). \quad (3.7)$$

In which  $X_{adm}$ ,  $S_{adm}$ , and  $\Theta_{adm}$  are our desired set for  $x$ ,  $s$ , and  $\theta$ , and the throughput time  $T$  necessary for producing a product instance does not exceed  $T_{max}$ .  $S$  measures the maturity of a process only from the technical point of view. To find out whether the resulting optimized process  $P^*$  is economically viable, the cost of the process must also be considered. Applying the economic part is beyond the scope of this project, but it can be considered in the future.

## 4 Numerical example: electric arc furnace model

The electric arc furnace system is the steel-making process that uses the heat of the arc to melt iron. According to literature studies [5], in which three-phase electric arc furnace systems with three electrodes were modeled, the general model of the system is in this form:  $M_I : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , where the inputs are the positions of each electrode and the outputs are the currents of each electrode.

$$[I_i] = [G_{ij}][x_i] + [B_i] \quad (4.1)$$

In which  $i = 1, 2, 3$  corresponded to the values of each electrode,  $I_i$  is the current of each electrode,  $x_i$  is the position of each electrode, and  $G_{ij}$  is the  $3 \times 3$  conductance matrix and  $B_i$  is the  $3 \times 1$  constant matrix.  $G_i$  is the slag to matte conductance and its relation with the position of each electrode is:

$$G_i = c_i x_i + G_s \quad (4.2)$$

where  $c_i$  is the conductance coefficient (in Siemens/m),  $x_i$  is the position of each electrode (in m) and  $G_s$  is the conductance of the slag when the electrodes are positioned at the surface of the slag (in S). In modeling the system from the physical model to the electric circuit the matte is considered as the virtual ground so we represent the voltage of this node in the electric circuit as  $V_m = 0$ . In [5] the electric arc furnace is modeled as a three-phase electric circuit and the amplitude of each voltage is 500 V with  $120^\circ$  phase difference. Also, it is considered that the inter-electrode conductance is the same in three electrodes and is shown by  $G$ . The dynamic equation  $\mathbf{GG}$  of the system is

$$\begin{bmatrix} 2V_1c_1(G_s + G) - V_2c_1(G_s + G) & V_1c_2(G_s + 2G) - V_2c_2(G_s + G) & V_1c_3(G_s + 2G) - V_2c_3G \\ -V_3c_1(G_s + G) + c_1c_2x_2(V_1 - V_2) & -V_3c_2G & -V_3c_3(G_s + G) \\ \quad \quad \quad + c_1c_3x_3(V_1 - V_3) & & \\ -V_1c_1(G_s + G) + V_2c_1(G_s + 2G) & -V_1c_2(G_s + G) + 2V_2c_2(G_s + G) & -V_1c_3G - V_2c_3(G_s + 2G) \\ \quad \quad \quad -V_3c_1G & -V_3c_3(G_s + G) + c_2c_3x_3(V_2 - V_3) & -V_3c_3(G_s + G) \\ \quad \quad \quad \quad \quad \quad \quad + c_1c_2x_1(V_2 - V_1) & & \\ -V_1c_1(G_s + G) - V_2c_1G & -V_1c_2G - V_2c_2(G_s + G) & -V_1c_3(G_s + G) - V_2c_3(G_s + G) \\ \quad \quad \quad -V_3c_1(G_s + 2G) & +V_3c_2(G_s + 2G) & 2V_3c_3(G_s + G) + c_1c_3x_1(V_3 - V_1) \\ & & \quad \quad \quad + c_2c_3x_2(V_3 - V_2) \end{bmatrix}$$

$$\begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{1}{G_{tot}} \mathbf{GG} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \frac{G_s^2 + 3GG_s}{G_{tot}} \begin{bmatrix} 2V_1 - V_2 - V_3 \\ -V_1 + 2V_2 - V_3 \\ -V_1 - V_2 + 2V_3 \end{bmatrix}$$

**Table 4.1:** Table of variables in the electric arc furnace system

Variables	Acceptable Range	Chosen Value
$v_1, v_2, \text{ and } v_3$	100–1000 V	500 V
$c_1, c_2, \text{ and } c_3$	1–100 S/m	20 S/m
$g_s$	5–25 S	10 S
$g$	$\approx 0$ S	0.1 S

In which the  $v_1, v_2, v_3$  are the voltage of each electrode and  $G_{\text{tot}} = c_1x_1 + c_2x_2 + c_3x_3 + 3G_s$ . Table 4.1 shows variables for this electric arc furnace system as a three-phase electric circuit the chosen values for the amplitude of the voltage with the  $120^\circ$  phase difference for each electrode is,  $v_1 = v_2 = v_3 = 500$  V,  $c_1 = c_2 = c_3 = 20$  S/m,  $g_s = 10$  S and the  $g = 0.1$  S.

#### 4.1 Temperature as feature for quality in the electric arc furnace system

One of the features considered for the quality of the electric arc furnace system is the temperature. It is assumed that heat transfer in the liquid metal is fast enough so that the temperature of the liquid metal is uniform throughout [2]. A similar situation applies to liquid slag.

There are differences in temperature between the melted part of the furnace, the scrap, and the solid part of the slag. The melted part heats both the scrap and the solid part of the slag. The heating rate is proportional to the difference between the temperature of the solid part and the temperature of the liquid part [2].

We consider that the liquid part is heated by the heat that it receives from the arc of the electrodes, so the formula for the rate of change of temperature is:

$$\frac{d\text{Temp}}{dt} = \frac{Q_{\text{arc}}}{mc_p} \quad (4.3)$$

In which the  $Q_{\text{arc}}$  is the power received from the electrodes, and  $m$  is the mass of the liquid steel. and  $c_p$  is the heat capacity of the liquid steel. Regarding [13],

$c_p = 0.84 \text{ kJ/kg}^\circ \text{ C}$ , we consider the value of  $m = 80 \text{ t}$  and the temperature starts at  $25^\circ \text{ C}$  which is the temperature of the room at this temperature  $c_p = 0.47 \text{ KJ/Kg}^\circ \text{ C}$ , we produce  $Q_{arc}$  by calculating the formulas in the next section.

### **Energy calculation:**

The formula for the power for one electrode is  $Q_{arc} = VI$  and

$$E_{arc} = \int_0^T Q_{arc} dt + Q_{start},$$

in which  $V$  and  $I$  are the root mean square (RMS) magnitude of the voltage and current for each electrode and  $E_{start}$  is calculated with this formula  $E_{start} = mc\Delta\text{Temp} = 80 \times 10^3 \times .47 \times 25 = 940\text{MJ}$ , where  $\Delta\text{Temp}$  shows the temperature change. The temperature grows from  $0^\circ \text{ C}$  to  $25^\circ \text{ C}$ .

## **4.2 Design a controller for the system**

We designed two types of controller for the system. One of them steers the position of each electrode to achieve the desired current, and the other steers the position of each electrode in the furnace to achieve the desired temperature. Since temperature is a key quality feature in the electric arc furnace system, we expect that directly controlling the position to achieve the desired quality feature (second controller) will increase the maturity of the system. We will calculate  $F$  and  $S$  in practice for this furnace system. In the future, we can compare  $E, F$ , and  $S$  for these two controllers to determine which increases the maturity of the system and which one is better to implement.

### **Position controller for achieving the desired current**

We consider that in the furnace model explained in the previous section, there is noise from the sensors. Therefore, we added sensor noise  $\mathcal{N}(0, 500)$  to the measurement of the output current. In addition, we have initial condition noise  $\mathcal{N}(0, 0.1)$  for the electrode positions. Therefore, we solved the stochastic differential equation (SDE) for the relationship between the output current and the position of the electrodes.

We designed a controller, in order to steer the position of the electrodes to their desired position, where we have the desired current. To find the desired position

for each electrode, we solved the optimization problem and found the optimal position for the electrodes to minimize the error between the desired current and the output current of the electric arc furnace model.

### **Position controller for achieving the desired current**

The optimum position of each electrode is calculated by solving this optimization problem:

$$\min_{x \in X_{adm}} \|I_{des} - M_I(x)\|$$

where each element of  $I_{des}$  is 10 kA RMS, and  $M_I(x)$  A RMS is the magnitude of the current when the electrodes are in the position  $x$ .

### **Position controller for achieving the desired temperature**

For the next phase, we designed a controller to control the positions of the electrodes, in order to achieve the desired temperature.

The optimization problem that gives the optimum position of each electrode to achieve the desired temperature is

$$\min_{x \in X_{adm}} \|\text{Temp}_{des} - \text{Temp}(x)\|$$

where  $\text{Temp}_{des}$  is the desired temperature at time  $T$ , which is 1500° C, and  $\text{Temp}(x)$  gives the temperature of the whole furnace when the electrodes are in the position  $x$ .

## **4.3 Implementing the maturity metrics in electric arc furnace model**

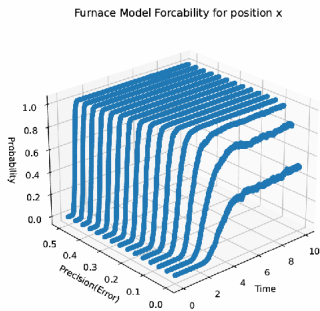
### **Forcability Plot**

We have the furnace arc model and find the error between the desired current  $I_{des}$  and its value at time  $T$ . In the simulation part, the Furnace Model is run for a time duration of  $T = 10$  hours. We produced the probability plot empirically by running the simulation 1000 times. For the specific error  $\eta_x$ , at each iteration, we checked if the error  $|\check{x} - x(T)| < \eta_x$ . When the error was less than the specified  $\eta_x$ , we incremented the count variable corresponding to this specific error. In the end, we divided the count value corresponding to each  $\eta_x$  at each specific time

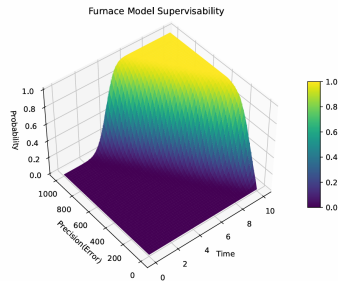
by 1000 to find the probability of having an error less than  $\eta_x$ . We plotted the probability distribution for this error. Figure 4.1 shows the Forcability plot for the system with the controller to achieve the desired current. As represented in the figure, the probability of having an error around 0.5 is higher. The variance in the lines shows the uncertainty caused by sensor noise and the output noise of the system. In other words, the plot shows the probability:

$$F_\pi(T, \eta_s, \eta_y) := Pr(\|x_1(T) - \check{x}_1\| < \eta_{x_1} \mid x_1(0), x(t) = \pi(Y_t), \check{x}_1) \quad (4.4)$$

In which the  $\check{x}_1 = 0.95$  for the first electrode.



**Figure 4.1:** The Forcability plot for the position of the first electrode of the electric arc furnace model ( $x_1$ ) illustrates the probability of steering  $x_1$  to its desired value with specific precision, as presented in (4.4).



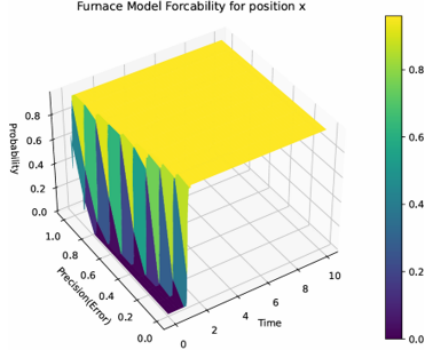
**Figure 4.2:** Supervisability plot for achieving the desired temperature as illustrated in (4.5)

## Supervisability Plot

Figure 4.2 shows the Supervisability plot for the system with a controller to achieve the desired temperature. As shown, the probability of having a precision of 1000 at the early stage of the time duration is higher. Also, the probability of having an error around 200 is possible at the end of the simulation. Calculating the probability plot for the Supervisability is empirical, analogous to the method of plotting the Forcability plot.

$$S := Pr(\|\text{Temp}(T) - \check{\text{Temp}}(T)\| < \eta_T \mid x \in X_{adm}). \quad (4.5)$$





**Figure 5.1:** Forcability plot for the system with learned model

In which  $\text{Temp}(T)$  shows the temperature at time  $T$ .

## 5 Surrogate model of the system

In the previous example we considered that we have the complete physical equation of the system. In this section, we consider that we do not have the exact physical model of the system and we try to implement our metrics on this example. We estimate the state of the system at time  $t + 1$  from its' state at time  $t$ . We consider that the Neural Network is deterministic but we have output noise and initial condition noise in the system as uncertainty.

Figure 5.1 shows the Forcability plot for the system with the learned model. Comparison of 5.1 and 4.1, the probability started to grow earlier for the system with exact physical model.

## 6 Summery and Outlook

In this paper, we introduced novel mathematical metrics to quantify the maturity of production processes. We introduced Elucidability and Forcability, which

are analogous to the classical definitions in control theory: observability and controllability.  $E$  quantifies the ability to observe the states and parameters of the system, while  $F$  quantifies the ability to steer the outputs and states of the system, which are related to the internal dynamics of the system.  $S$  represents the probability of achieving the desired quality for the final product. Using the example of the electric arc furnace system, we demonstrated  $E$ ,  $F$ , and  $S$  in practice. The probability plots presented in the example can be compared to assess how changes in the process can increase the maturity of the production process and provide value. In the future, we plan to investigate the impact of hardware and software changes on  $E$ ,  $F$ , and  $S$ , and explore whether there is a monotonic relationship between increases in these variables.

## References

- [1] Nazanin Azizian et al. “A framework for evaluating technology readiness, system quality, and program performance of US DoD acquisitions”. In: *Systems Engineering* 14.4 (2011), pp. 410–426.
- [2] Johannes Gerhardt Bekker, Ian Keith Craig, and Petrus Christiaan Pistorius. “Modeling and simulation of an electric arc furnace process”. In: *ISIJ international* 39.1 (1999), pp. 23–32.
- [3] Jürgen Beyerer. “The value of additional knowledge in measurement–a Bayesian approach”. In: *Measurement* 25.1 (1999), pp. 1–7.
- [4] Jürgen Beyerer and Jürgen Geisler. “A framework for a uniform quantitative description of risk with respect to safety and security”. In: *European Journal for Security Research* 1 (2016), pp. 135–150.
- [5] Benoit Boulet, Gino Lalli, and Mark Ajersch. “Modeling and control of an electric arc furnace”. In: *Proceedings of the 2003 American Control Conference, 2003*. Vol. 4. IEEE. 2003, pp. 3060–3064.
- [6] American Heritage Dictionaries. *American Heritage Dictionary of the English Language*. Houghton Mifflin Harcourt, 2015.

- [7] AMA Hamdan and AH Nayfeh. “Measures of modal controllability and observability for first-and second-order linear systems”. In: *Journal of guidance, control, and dynamics* 12.3 (1989), pp. 421–428.
- [8] Malo LJ Hautus. “Controllability and observability conditions of linear autonomous systems”. In: *Ned. Akad. Wetenschappen* 72 (1969), pp. 443–448.
- [9] PC Hughes and RE Skelton. “Controllability and observability of linear matrix-second-order systems”. In: *Journal of applied mechanics* 47.2 (1980), pp. 21–39.
- [10] Rudolf Kalman. “Controllability of linear dynamical systems”. In: *Contributions to differential equations* (1963), pp. 189–213.
- [11] Okhyun Kang et al. “New measure representing degree of controllability for disturbance rejection”. In: *Journal of guidance, control, and dynamics* 32.5 (2009), pp. 1658–1661.
- [12] Haemin Lee and Youngjin Park. “Degree of controllability for linear unstable systems”. In: *Journal of Vibration and Control* 22.7 (2016), pp. 1928–1934.
- [13] Vito Logar, Dejan Dovžan, and Igor Škrjanc. “Modeling and validation of an electric arc furnace: Part 1, heat and mass transfer”. In: *ISIJ international* 52.3 (2012), pp. 402–412.
- [14] Jennifer M Long. “Integration readiness levels”. In: *2011 Aerospace Conference*. IEEE. 2011, pp. 1–9.
- [15] John C Mankins. “Technology readiness levels: A white paper”. In: <http://www.hq.nasa.gov/office/codeq/trl/trl.pdf> (1995).
- [16] Benoît Marx, Damien Koenig, and Didier Georges. “Optimal sensor and actuator location for descriptor systems using generalized gramians and balanced realizations”. In: *Proceedings of the 2004 American Control Conference*. Vol. 3. IEEE. 2004, pp. 2729–2734.
- [17] Benoît Marx, Damien Koenig, and Didier Georges. “Optimal sensor/actuator location for descriptor systems using Lyapunov-like equations”. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*. Vol. 4. IEEE. 2002, pp. 4541–4542.

- [18] Eileen McConkie et al. “Mathematical properties of system readiness levels”. In: *Systems Engineering* 16.4 (2013), pp. 391–400.
- [19] Tobias Mettler. “Maturity assessment models: a design science research approach”. In: *International Journal of Society Systems Science* 3.1-2 (2011), pp. 81–98.
- [20] PC Müller and HI Weber. “Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems”. In: *Automatica* 8.3 (1972), pp. 237–246.
- [21] Mark C Paulk et al. “Capability maturity model, version 1.1”. In: *IEEE software* 10.4 (1993), pp. 18–27.
- [22] Shari Lawrence Pfleeger. “Maturity, models, and goals: How to build a metrics plan”. In: *Journal of Systems and Software* 31.2 (1995), pp. 143–155.
- [23] Hossein Saiedian and Laura M McClanahan. “Frameworks for quality software process: SEI Capability Maturity Model versus ISO 9000”. In: *Software Quality Journal* 5 (1996), pp. 1–23.
- [24] Kazuhiro Sato and Shun Terasaki. “Controllability scores for selecting control nodes of large-scale network systems”. In: *IEEE Transactions on Automatic Control* (2024).
- [25] B Sauser et al. “Development of systems engineering maturity models and management tools”. In: *Stevens Institute of Technology. Report No. SERC-2011-TR-014* (2011).
- [26] Brian Sauser et al. “From TRL to SRL: The concept of systems readiness levels”. In: *Conference on Systems Engineering Research*. Vol. 5. 0002. Citeseer. 2006, pp. 5–7.
- [27] Hamid Reza Shaker and Sanja Lazarova-Molnar. “A new data-driven controllability measure with application in intelligent buildings”. In: *Energy and Buildings* 138 (2017), pp. 526–529.
- [28] Abhay K Singh and Juergen Hahn. “Determining optimal sensor locations for state and parameter estimation for stable nonlinear systems”. In: *Industrial & engineering chemistry research* 44.15 (2005), pp. 5645–5659.

- [29] Abhay K Singh and Juergen Hahn. “Sensor location for stable nonlinear dynamic systems: Multiple sensor case”. In: *Industrial & engineering chemistry research* 45.10 (2006), pp. 3615–3623.
- [30] James D Smith. “An alternative to technology readiness levels for non-developmental item (NDI) software”. In: *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE. 2005, 315a–315a.
- [31] M Tarokh. “Measures for controllability, observability and fixed modes”. In: *IEEE Transactions on Automatic Control* 37.8 (1992), pp. 1268–1273.
- [32] Ricardo Valerdi and Ron J Kohl. “An approach to technology risk management”. In: *Engineering Systems Division Symposium*. Vol. 3. Citeseer. 2004, pp. 29–31.
- [33] Leslie G Valiant. “A theory of the learnable”. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.
- [34] Eileen M Van Aken et al. “Assessing maturity and effectiveness of enterprise performance measurement systems”. In: *International Journal of Productivity and Performance Management* 54.5/6 (2005), pp. 400–418.
- [35] CN Viswanathan, RW Longman, and PW Likins. “A degree of controllability definition-fundamental concepts and application to modal systems”. In: *Journal of Guidance, Control, and Dynamics* 7.2 (1984), pp. 222–230.
- [36] Thomas Wettstein and Peter Kueng. “A maturity model for performance measurement systems”. In: *WIT Transactions on Information and Communication Technologies* 26 (2002).
- [37] Yaping Xia et al. “A new measure of the degree of controllability for linear system with external disturbance and its application to wind turbines”. In: *Journal of Vibration and Control* 24.4 (2018), pp. 739–759.
- [38] Jimmy Anderson Flórez Zuluaga et al. “Model for measuring technological maturity for critical sector industries”. In: *Journal of Open Innovation: Technology, Market, and Complexity* 10.1 (2024), p. 100194.



# Uncertainty Quantification and The Need for Better Understanding of Monte Carlo and Random Number Generation

*Ali Darijani*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
ali.darjani@kit.edu

## Abstract

Uncertainty Quantification (UQ) is critical in scientific modeling and decision making, addressing the inherent uncertainty and lack of precision in computational modeling. Monte Carlo methods, one of the two prominent approaches in UQ, rely on random number generation to simulate complex systems and estimate probability density functions. However, the quality of these methods is naturally tied to the integrity of the random numbers used. Poorly designed random number generators (RNGs) can lead to biased or unreliable results, undermining the validity of UQ. Despite advancements in RNG algorithms, challenges remain in balancing computational efficiency, reproducibility, and randomness quality. A deeper understanding of these generators, their limitations, and their integration with Monte Carlo methods is essential to advancing UQ practices. This technical report tries to highlight keywords in the random number generation field and give a couple of examples showcasing the pitfalls.

# 1 Introduction

Mathematicians mostly categorize the uncertainty into epistemic, degree of knowledge when compared to an absolute truth, and aleatoric, something inherently nondeterministic[15], which is something that the physicists also support. While quantum mechanics is still young compared to its counterparts and there is still not consensus in some parts; the Bell inequality and The Einstein-Podolsky-Rosen(EPR) Argument show that quantum mechanics is nondeterministic in nature and there are no “local hidden variables” as opposed to what Einstein thought[14]. As models in molecular dynamics require knowledge from quantum mechanics the uncertainty will carry over and is present at the molecular scale[16]. The same happens when moving from the molecular scale to the continuum scale, and therefore we have uncertainty which we can not get rid of, in the macro scale.

Uncertainty Quantification (UQ) field is gaining attraction which focuses on characterizing, reducing, and managing uncertainty in mathematical models that are based on physical systems. In both forward and backward modes, uncertainty quantification approaches the problem of determining how uncertainties in inputs, models, or parameters affect outputs or vice versa[10].

Monte Carlo Simulation sample inputs from their distributions, run the model multiple times, and analyze the outputs to obtain insights into their distributions[13].

Random number generation from arbitrary distributions typically depends on uniform random number generators because they provide a standardized and versatile base. Transformations like the inverse transform sampling, rejection sampling, or Box-Muller method convert uniform random numbers into samples from desired distributions.

True uniform random number generation is hard to achieve, instead, Pseudo Random Number Generators (PRNGs) are used to produce sequences that approximate true randomness. PRNGs are practical, reproducible, and efficient for most applications, despite being deterministic and periodic over long sequences. Common pitfalls in using them include correlations in sequences (lack of independence), and small periodicity in pseudo-random number generators (PRNGs).



These issues can lead to inaccuracies in simulations or statistical applications if the generator is not carefully chosen.

## 2 Sample Generation

### 2.1 Uniform Random Number Generator

John von Neumann jokingly stated: ‘‘Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin’’, but still there exist the notion of a good PRNG. A good PRNG is a balance between different factors depending on the application. Some of these factors are but not limited to[13]:

- Good performance in statistical tests to justify the *independently and identically distributed* assumption which is of course too inaccurate and not feasible practically.
- Reproducible for correctness checking with a small memory footprint.
- Fast and efficient to the degree that is required of them.
- Large period to reduce the chance of correlation. The rule of thumb is that to produce  $N$  random number the period has to be at least  $10N^2$ .
- The ability to produce multiple independent streams.
- Cheap and easy to implement and use as opposed to having expensive physical equipments to capture nature noise as random numbers.
- Does not produce 0 and 1 for technical reasons.

**Remark 2.1.1.** Some might need speed over large periods. Some might need irreproducible sequences for example for Cryptography applications therefore the need for physics based generators as opposed to PRNG which brings us back to the *There is no solution that fits all.*

## 2.2 Sample Generation from Arbitrary Distribution

Unfortunately not every probability distribution function can be sampled easily. Different methods exist for different distributions in different dimensions and different regularities. Some with their pros and cons are:

- (Inverse-Transform Method):
  - Works if you know the inverse of the cumulative distribution function. Naturally it does not work for several variables as the probability density function is not bijective. Gets discarded if there exist no analytical inverse; for example in the case of fifth order polynomials.
- (Acceptance-Rejection Method):
  - Works if you can bound the probability density function by another probability density function with known sampling procedure. Unfortunately it is not an efficient method as a lot of samples have to be thrown away as the *Rejection* in the name implies. It is also not a black box method which makes it not usable for those who would like an abstraction layer for the random generation.
- (Composition Method):
  - The overall distribution must be written as a convex hull of distributions with known sample generation procedures. One famous example is the Gaussian Mixture.

## 3 The Curse of Dimensionality

**Definition 3.0.1** (Curse of Dimensionality). The higher the dimension the more difficult solving the problem becomes.

**Remark 3.0.2.** The term got coined by Richard E. Bellman[2].

**Remark 3.0.3.** The dimensionality is not always an adversary and there exist ‘‘Blessing of Dimensionality’’[11][4][6][7] as well.

**Remark 3.0.4.** The curse of dimensionality can and will happen Sampling Methods[13], Differential Equations[8][5][12], Tensor Based Methods[9] and many other disciplines.

### 3.1 Concentration of Measure

**Definition 3.1.1.** We call:

$$B_R^n(c) = \{x, c \in \mathbb{R}^n : \sum_{i=1}^n (x_i - c_i)^2 \leq R^2\}$$

the  $n$ -ball and its points are at the distance  $R$  from the point  $c$  with respect to the Euclidean distance in the  $n$ -dimensional space. We show the volume of the said ball by  $\text{Vol}(B_R^n(c))$ .

**Proposition 3.1.2.**

$$\text{Vol}(B_R^1(c)) = 2R.$$

*Proof.* Move the center of the ball to zero as the volume is invariant under translation and then:

$$x_1 = R \sin \theta, \quad -\pi/2 \leq \theta \leq +\pi/2$$

$$\int_{B_R^1} dx = \int_{-\pi/2}^{+\pi/2} |\det(J)| d\theta = \int_{-\pi/2}^{+\pi/2} R \cos \theta d\theta = R \sin \theta \Big|_{-\pi/2}^{+\pi/2} = 2R$$

where  $J$  is the Jacobian of the mapping:

$$J = \left( \frac{\partial x_1}{\partial \theta} \right) = R \cos \theta$$

□

**Proposition 3.1.3.**

$$\text{Vol}(B_R^2(c)) = \pi R^2.$$

*Proof.* Move the center of the balls to zero as the volume is invariant under translation and then:

$$\begin{aligned} x_1 &= r \cos \theta, & 0 \leq \theta < 2\pi \\ x_2 &= r \sin \theta, & 0 \leq r \leq R, \end{aligned}$$

$$\int_{B_R^1} dx = \int_0^R \int_0^{2\pi} |\det(J)| d\theta dr = \int_0^R \int_0^{2\pi} r d\theta dr = \pi R^2$$

where  $J$  is the Jacobian of the mapping:

$$J = \begin{pmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_1}{\partial \theta} \\ \frac{\partial x_2}{\partial r} & \frac{\partial x_2}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{pmatrix}$$

□

**Proposition 3.1.4.**

$$\text{Vol}(B_R^3(c)) = \frac{4}{3}\pi R^3.$$

*Proof.* Move the center of the ball to zero as the volume is invariant under translation and then:

$$\begin{aligned} x_1 &= r \sin \theta \cos \varphi, & 0 \leq \varphi < 2\pi \\ x_2 &= r \sin \theta \sin \varphi, & 0 \leq \theta \leq \pi, \\ x_3 &= r \cos \theta, & 0 \leq r \leq R, \end{aligned}$$

$$\begin{aligned} \int_{B_R^3} dx &= \int_0^R \int_0^\pi \int_0^{2\pi} |\det(J)| d\varphi d\theta dr \\ &= \int_0^R \int_0^\pi \int_0^{2\pi} r^2 \sin \theta d\varphi d\theta dr = 4/3\pi R^3 \end{aligned}$$

□

where  $J$  is the Jacobian of the mapping:

$$J = \begin{pmatrix} \frac{\partial x_1}{\partial r} & \frac{\partial x_1}{\partial \theta} & \frac{\partial x_1}{\partial \varphi} \\ \frac{\partial x_2}{\partial r} & \frac{\partial x_2}{\partial \theta} & \frac{\partial x_2}{\partial \varphi} \\ \frac{\partial x_3}{\partial r} & \frac{\partial x_3}{\partial \theta} & \frac{\partial x_3}{\partial \varphi} \end{pmatrix} = \begin{pmatrix} \sin \theta \cos \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \sin \theta \sin \varphi & r \cos \theta \sin \varphi & r \sin \theta \cos \varphi \\ \cos \theta & -r \sin \theta & 0 \end{pmatrix}$$

**Proposition 3.1.5.**

$$\text{Vol}(B_R^n(c)) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)} R^n.$$

*Proof.* Move the center of the ball to zero as the volume is invariant under translation and then:

$$\begin{aligned}x_1 &= r \cos \theta_1, \\x_2 &= r \sin \theta_1 \cos \theta_2, \\&\vdots \\x_{n-1} &= r \sin \theta_1 \cdots \sin \theta_{n-2} \cos \theta_{n-1}, \\x_n &= r \sin \theta_1 \cdots \sin \theta_{n-2} \sin \theta_{n-1},\end{aligned}$$

for:

$$\begin{aligned}r &\in [0, R], \theta_1 \in [0, \pi], \theta_2, \dots, \theta_{n-1} \in [0, 2\pi) \\ \Omega &= [0, R] \times [0, \pi] \times \cdots \times [0, 2\pi], d\Theta = \prod_{i=1}^{i=n-1} d\theta_i \\ \int_{\mathbb{B}_R^n} dx &= \int_{\Omega} |\det(J)| dr d\theta_1 d\theta_2 \cdots d\theta_{n-1} \\ &= \int_{\Omega} r^{n-1} \sin^{n-2} \theta_1 \sin^{n-3} \theta_{n-2} \cdots \sin \theta_{n-1} dr d\Theta \\ &= \frac{\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)} R^n\end{aligned}$$

where  $\Gamma$  is the gamma function. □

**Remark 3.1.6.**

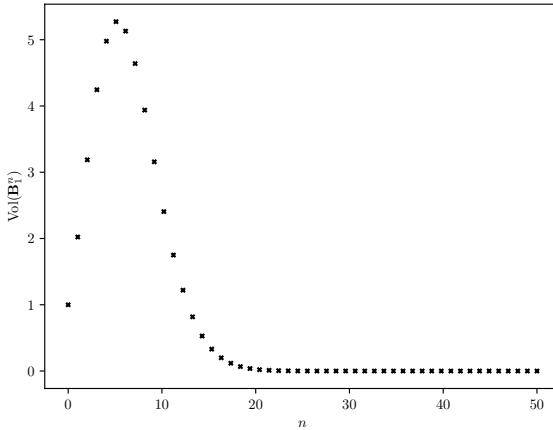
- The Gamma function[1]
- Non-geometric approach for the  $n$ -ball[3]

**Proposition 3.1.7.**

$$\lim_{n \rightarrow \infty} \text{Vol}(\mathbb{B}_1^n(c)) = 0$$

**Remark 3.1.8.** While it is possible to prove the proposition by recalling properties of the Gamma function[1] and do asymptotic analysis, taking a look at the (3.1) is beneficial.

**Figure 3.1:** Volume of the unit  $n$ -ball against its dimension.



**Proposition 3.1.9.** For small  $\varepsilon$ :

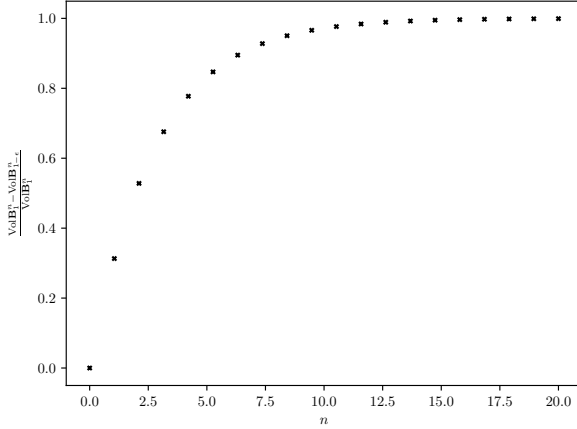
$$\lim_{n \rightarrow \infty} \frac{\text{Vol}(B_1^n(c)) - \text{Vol}(B_{1-\varepsilon}^n(c))}{\text{Vol}(B_1^n(c))} = 1$$

**Remark 3.1.10.** While it is possible to prove the proposition by recalling properties of the Gamma function[1] and do asymptotic analysis, taking a look at the (3.2) is beneficial.

**Remark 3.1.11.** The phenomenon is known as the Concentration of Measure.

### 3.2 $\pi$ Calculation by Monte Carlo on the $n$ -Ball

Monte Carlo methods can estimate  $\pi$  by leveraging the relationship between the volume of an  $n$ -ball and  $\pi$ . To estimate  $\pi$ , random points are generated within an  $n$ -dimensional cube that bounds an  $n$ -ball. The fraction of points that fall inside the  $n$ -ball, determined by checking if their distance from the origin is less than

**Figure 3.2:** Concentration of measure for the unit  $n$ -ball.


or equal to the radius, used to approximate the ratio of the volume of the  $n$ -ball to the  $n$ -cube. Results can be seen in (3.3), (3.4) show that it is quite effective in low dimensions, but unfortunately as it can be seen in (3.5) and (3.6) no matter the number of points, estimating  $\pi$  proves to be difficult in high dimensions.

**Theorem 3.2.1** (Nyquist-Whitaker-Shannon Theorem). *Let  $B > 0$  and  $f$  be in  $L_2(\mathbb{R}, \mathbb{C})$  be such that  $\mathcal{F}f(\omega) = 0$  for  $|\omega| > B$ . Then  $f$  is continuous (more precisely, has a continuous representative) and is uniquely determined by the values  $(f(k\pi/B))_{k \in \mathbb{Z}}$ . Moreover,*

$$u(x) = \sum_{k \in \mathbb{Z}} u\left(\frac{k\pi}{B}\right) \operatorname{sinc}\left(\frac{B}{\pi}\left(x - \frac{k\pi}{B}\right)\right).$$

**Remark 3.2.2.** If the correct sampling rate indicated by the Nyquist-Whitaker-Shannon Theorem is not met it might lead to undersampling (3.7) or oversampling which are undesirable.

Figure 3.3: Calculating  $\pi$  using Monte Carlo on the unit 2-ball.

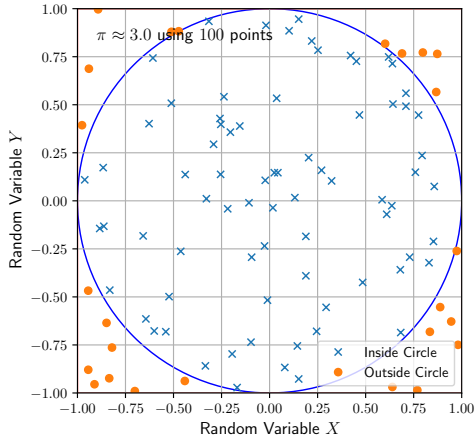
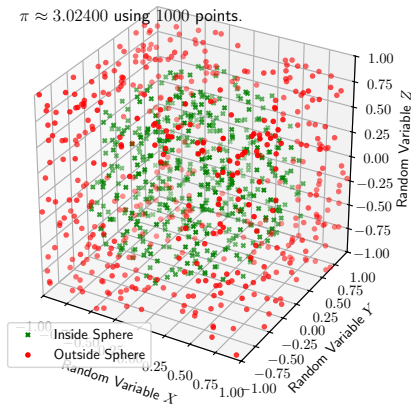
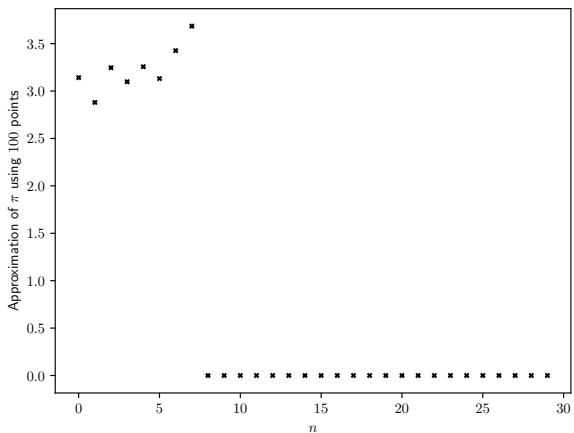


Figure 3.4: Calculating  $\pi$  using Monte Carlo on the unit 3-ball.

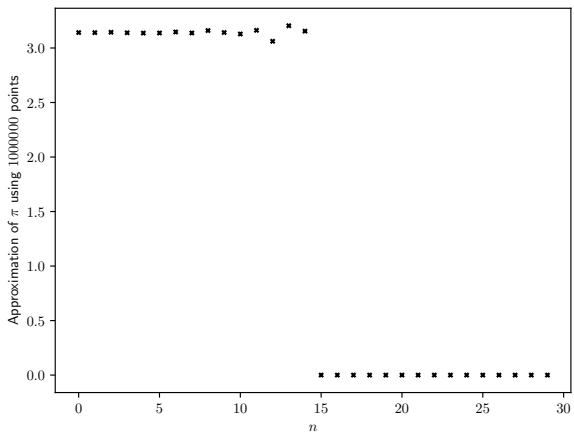


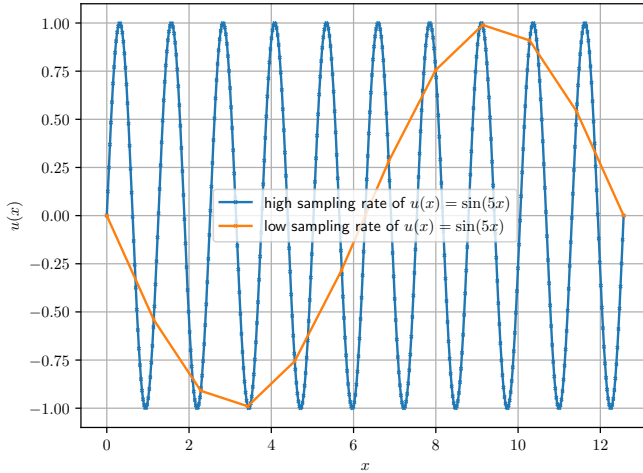


**Figure 3.5:** Calculating  $\pi$  using Monte Carlo using 100 points in different dimensions.



**Figure 3.6:** Calculating  $\pi$  using Monte Carlo using 1000000 points in different dimensions.



**Figure 3.7:** The effect of undersampling

## 4 My Plans

I am interested in studying Monte Carlo methods more thoroughly to gain a deeper understanding of their mathematical foundations and practical applications. By diving deep into the principles of randomness, sampling, and estimation that underpin these methods, I hope to gain the upper hand when dealing with Monte Carlo solvable problems. Additionally, I aim to explore specialized Monte Carlo techniques designed for specific scenarios, such as variance reduction methods, importance sampling, or Markov Chain Monte Carlo (MCMC), to better apply these tools to targeted challenges in fields like physics, finance, and machine learning.

## References

- [1] Emil Artin. *The Gamma Function*. Dover, 2015.

- [2] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Dec. 2010. ISBN: 9781400835386. DOI: 10.1515/9781400835386.
- [3] L. E. Blumenson. “A Derivation of n-Dimensional Spherical Coordinates”. In: *The American Mathematical Monthly* 67.1 (Jan. 1960), p. 63. ISSN: 0002-9890. DOI: 10.2307/2308932.
- [4] David L. Donoho. “High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality”. In: (2000).
- [5] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer New York, 2021. ISBN: 9781071613443. DOI: 10.1007/978-1-0716-1344-3.
- [6] A. N. Gorban and I. Y. Tyukin. “Blessing of dimensionality: mathematical foundations of the statistical physics of data”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2118 (Mar. 2018), p. 20170237. ISSN: 1471-2962. DOI: 10.1098/rsta.2017.0237.
- [7] Alexander N. Gorban, Valery A. Makarov, and Ivan Y. Tyukin. “High-Dimensional Brain in a High-Dimensional World: Blessing of Dimensionality”. In: *Entropy* 22.1 (Jan. 2020), p. 82. ISSN: 1099-4300. DOI: 10.3390/e22010082.
- [8] Wolfgang Hackbusch. *Elliptic Differential Equations*. Springer Berlin Heidelberg, 2017. ISBN: 9783662549612. DOI: 10.1007/978-3-662-54961-2.
- [9] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer International Publishing, 2019. ISBN: 9783030355548. DOI: 10.1007/978-3-030-35554-8.
- [10] *Handbook of Uncertainty Quantification*. 2017. DOI: 10.1007/978-3-319-12385-1.
- [11] Paul C. Kainen. “Utilizing Geometric Anomalies of High Dimension: When Complexity Makes Computation Easier”. In: *Computer Intensive Methods in Control and Signal Processing*. Birkhäuser Boston, 1997, pp. 283–294. ISBN: 9781461219965. DOI: 10.1007/978-1-4612-1996-5\_18.

- [12] Peter Knabner and Lutz Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations: With contributions by Andreas Rupp*. Springer International Publishing, 2021. ISBN: 9783030793852. DOI: 10.1007/978-3-030-79385-2.
- [13] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo methods*. Wiley Series in Probability and Statistics. Includes bibliographical references and index. Hoboken, NJ: Wiley, 2011. ISBN: 0470177934.
- [14] Franz Schwabl. *Quantum Mechanics*. Fourth Edition. Springer Berlin Heidelberg, 2007. ISBN: 9783540719328. DOI: 10.1007/978-3-540-71933-5.
- [15] Tim Sullivan. *Introduction to Uncertainty Quantification*. Springer International Publishing, 2015. ISBN: 9783319233956. DOI: 10.1007/978-3-319-23395-6.
- [16] Mark E. Tuckerman. *Statistical Mechanics*. Fourth Edition. Oxford Graduate Texts. Description based upon print version of record. Oxford: Oxford University Press, Incorporated, 2023. ISBN: 0192559613.

# **Localization of latent influences in immature cyber physical systems**

*Frank Doehner*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
frank.doehner@kit.edu

## **Abstract**

Immature processes are often plagued by latent noise sources which prevent the system from performing with the required productivity and product quality. Locating them within the system in order to then instrument, remove or control for them is usually a costly and time consuming endeavor. Therefore, there is a need for a unifying methodology and techniques to optimize such systems and cut down on development costs. Depending on the system at hand, on varying amounts of initial knowledge about the system as well as the characteristics of the latent influence, different approaches for locating these latent influences need to be considered. In this work, we conceptually explore different approaches for locating latent influences with regards to different sets of prerequisites. Furthermore, we provide a general methodological approach which can be customized to fit to specific systems.

## **1 Introduction**

A manufacturing process is immature if it cannot be operated with the required productivity and product quality. One cause for a process to be immature can be

hidden noise sources. These latent variables (LVs) within the system can cause high defect rates in production processes and unstable or poorly controllable process settings, all of which are undesired. Finding these LVs within a system can be costly in time and money, while usually relying on experts to identify them. One advantage immature processes have over matured processes is that they are still in development and can therefore be changed. Most approaches in the current literature on root cause localization [18, 5, 15] assume a fixed process structure where faults are out of distribution outliers. The addition of sensors and actuators to a system can be key factors in quickly maturing a process and thereby opens up room for novel approaches on the matter.

Localizing LV in physical systems is challenging due to several aspects. With each distinct physical system comes a unique domain of possible LVs, each with a separate locality within the system and different characteristics. Another vital aspect that needs to be considered is the initial knowledge about the system. Localizing a LV in a White Box system is generally easier than in a Grey Box or even Black Box system. All these aspects contribute to the plethora of different possible problem settings, making it a tall task to find a universally applicable methodology for LV localization.

The following work is composed of the following chapters: First, we categorize the facets of the possible problem settings in 2 and discuss how and in what form information about a LV's location within a system can be obtained. In 3 we first introduce a coarse methodology for iteratively narrowing down a LV's location within a physical system. We follow up by providing a set of possible approaches for extracting the maximum amount of information from the system. Finally, we finish this work with 4, providing our view on the next necessary steps to further advance this research topic.

## **2 Problem Settings**

Immature physical systems come in many different forms, since the number of different use cases is basically limitless. These go hand in hand with a number of design aspects and fundamental restrictions. Understanding them is crucial for finding means to localize LVs in order to optimize performance, improve

quality, and enhance overall system efficiency. In this section, we provide an introduction to various aspects of the problem setting and highlight some initial angles of attack.

## 2.1 Structural Causal Model

Causality offers frameworks for describing such physical systems on a logical level, allowing for reasoning on cause and effect. One prominent framework are structural causal models (SCMs) [16] where every input parameter and measured quantity is a variable.

**Definition 1:** *A structural causal model (SCM) is a tuple  $\mathcal{M} := \{U, V, F\}$  consisting of:*

1.  $U$ , the set of exogenous variables that are determined by factors outside the model.  $P(u)$  is the joint probability defined over the features in  $U$ .
2.  $V$ , the set of endogenous variables that are determined by variables in  $U \cup V$ .
3.  $F$ , the set of functions  $\{f_1, \dots, f_n\}$  where for each  $X_i \in V$ , a function  $f_i$  is a mapping from  $U \cup Pa(X_i)$  to  $X_i$  and  $Pa(X_i) \subseteq (V \setminus \{X_i\})$  are the parents of  $X_i$ .

The causal structure of a SCM can be depicted via a directed acyclic graph (DAG). DAGs and consequently SCMs do not allow for cycles, and therefore for bidirectional edges between variables. The acyclicity assumption is a key aspect for analytical tractability. It prevents feedback loops and allows for a clear ordering of cause and effect. In real life systems, this acyclicity assumption is not always feasible. In thermodynamic systems, temperature and pressure for example can mutually be the cause of each other. In some cases, modeling a process over time can approximate the real process well enough while upholding the acyclicity assumption. While SCMs can cover a broad variety, there is not one single ideal model for all possible problem settings.

LVs  $X_{LV}$  are endogenous variables, influenced by some exogenous noise, within a system that are unobservable. We further assume that, due to the problem setting at hand, it is not known where they are situated within a SCM. As a result, a SCM might be incomplete and therefore not able to accurately describe the system at hand. The residual  $\mathbf{R} = \mathbf{X} - \hat{\mathbf{X}}$  between the real data  $\mathbf{X}$  and the SCM prediction  $\hat{\mathbf{X}}$  holds a majority of the available information regarding any LVs in the system, as the residual is the direct cause of any uncertainty within the system. The prediction  $\hat{\mathbf{X}}$  can come from a physical model of the system or a regression network trained on the data. Extracting information encoded in the residuals via regression analysis techniques [8, 7] and effectively utilizing this information are the challenges at hand.

## 2.2 Physical Systems

One aspect which needs to be considered, are the spatial geometries of a system. Locating a LV in a 1-dimensional space is a different task from locating a LV in a 2- or even 3-dimensional space. Causal graphs are inherently one dimensional. While a causal graph either has or does not have an edge between two variables, a physical system might have multiple paths from one measuring point to another or the path might be spatially expanded. It is also not apparent if two different causal edges correspond to different physical paths or share a single one. Without LVs in the system this might be not of importance but due to the task of locating them, having an understanding of where these causal paths lie within the physical system becomes important.

## 2.3 Noise Characteristics

Not only the structure of the physical system and the position of the LVs within it are important but the LVs themselves have different characteristics which can hold valuable information. LVs, in our context, are noise sources within a physical system. To start off, these exogenous influences  $U_{LV}$  have different probability distributions  $D$ . Prominent noise distributions are, among others, the normal distribution and the uniform distribution. The noise  $U_{LV}$ , or in



a probabilistic sense its probability distribution  $P(U_{LV})$ , is then propagated through the corresponding SCM producing the measured residuals  $R_i$  at variable  $X_i$  with probability distribution  $P(R_i)$  in the outputs:

$$P(R_i) = (f_i(Pa(X_i), \cdot))_* P(U_{LV}) \quad (2.1)$$

$$\text{with: } X_i = f_i(Pa(X_i), U_{LV}) \quad (2.2)$$

2.1 describes the pushforward distribution of a random variable through a SCM, where the function  $f_i$  maps the Parents of  $X_i$  and any latent noise variables to  $X_i$  (2.2). Different initial distributions  $P(U_{LV})$  therefore generally result in different distributions  $P(R_i)$  of the systems residual.

If a LVs probability distribution is time independent, meaning it has a constant mean  $\mathbb{E}[U_{LV}(t)]$  and a constant variance  $\text{Var}[U_{LV}(t)]$ , it is considered stationary. Is the distribution stationary and its ensemble average equal to the time average  $\mathbb{E}_{\text{Ensemble}}[U_{LV}] = \mathbb{E}_{\text{Time}}[U_{LV}(t)]$  for any single realization, it is considered to be ergodic. Furthermore, the LV can be autocorrelated. The autocorrelation function  $\mathcal{R}_{U_{LV}, U_{LV}} = \mathbb{E}[U_{LV}(t_1) \cdot U_{LV}(t_2)]$  is a mathematical tool for measuring the similarity of a signal at time  $t_1$  with itself at a later time  $t_2$ . Autocorrelation may give concrete hints to the physical nature of the noise source. A periodic noise for example could vastly reduce the possible type of noise sources and therefore the LVs possible locations. A stationary noise would indicate a stable LV location and an ergodic noise would further suggest a LV that is completely independent of the process itself.

Scedasticity is another form of correlation that characterizes the change of a random variables variance [13, 17]. Homoscedastic variables have a constant variance, and heteroscedastic variables have a varying variance. This change of variance may be with respect to time but also can be with respect to other variables in the system. Take a 3D printer with an adjustable printing speed and the room temperature as a LV for example. If the room temperature is low, the printed part will be precise even at fast printing speeds. Is the room temperature high on the other hand, fast printing speeds might lead to a less precise print. Not only the variance, similarly the mean or higher modes of the LVs distributions can be correlated with variables in the system. This correlation can then be

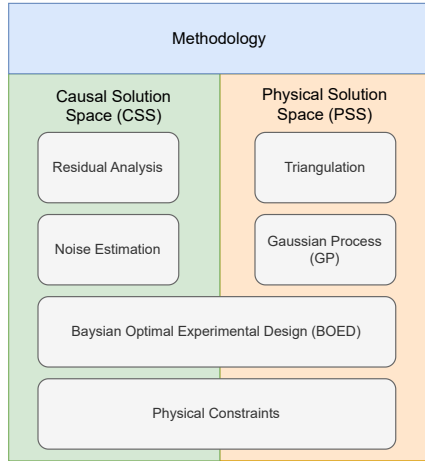
inferred from the systems residual.

## **2.4 Experimental Design**

Locating a LV within a system is by design an iterative process. The information obtained from an initial system's residual is in most cases not going to be conclusive. Leveraging this information, the process is then equipped with additional sensors or actuators to form a new process instance. These steps are then repeated until the LV has been localized. Keeping the process iterations to a minimum minimizes development time and cost. Therefore, the design of the experiment is another factor in order to maximize the gained information. The physical system receives a set of input parameters or variables. The former are fixed for one experiment, the latter can be adjusted or controlled during runtime. Finding an optimal set of input parameters or steering input variables to maximize the information extracted from a time series are topics of interest.

## **2.5 Process Restrictions**

A final aspect in need of consideration is the available information of the system at hand. As instrumenting the process is a key aspect for locating any LV, it is important for the process to be instrumentable. While theoretically not an issue, it can be practically very challenging or even impossible to place sensors or actuators in certain locations. Furthermore, some processes or some parts of processes might inherently be black boxes. This makes it very difficult to map a causal system to its physical one as the latter is unobservable. Even if sensors can be placed and a causal graph can be obtained, inferring about the physical system outside of the sensor locations is not possible without further information in such areas.



**Figure 3.1:** Overview of proposed solution propositions for the causal and physical domain.

### 3 Solution Propositions

This chapter contains a collection of ideas and possible research topics of which most have not been tested or verified yet 3.1. As portrayed in 2, the space of possible problem settings for locating LVs in physical systems is large, and finding a single method to cover them all seems ambiguous at best. We therefore intent to provide a sort of tool box that contains different methods that can be applied individually or arbitrarily combined for locating LVs.

#### 3.1 Methodology for Process Instrumentation

When locating a LV, we are constrained by the available information: The knowledge about the physical realization of the system, the known causal dependencies between measured variables and the residual  $\mathbf{R}$  between real data  $\mathbf{X}$  and predicted data  $\hat{\mathbf{X}}$  given the measured or set inputs. Due to the nature of the problem,

this initial information is often not enough to come to a conclusion regarding the LVs location. The system needs to be further instrumented to acquire additional information [1]. Therefore, a methodology which proposes the type and location of an additional instrumentation under consideration of the available information is a necessity.

In the first step information is extracted from the system's residual under consideration of its causal structure. For example, a correlation between the residual  $R_i$  and some input values can be detected. This constrains the solution space within the SCM. This "causal" solution space (CSS) possesses a mapping to the physical system and its physical solution space (PSS).

$$\text{Mapping to physical domain:} \quad f_{C-P} : \text{CSS} \rightarrow \text{PSS} \quad (3.1)$$

$$\text{Mapping to causal domain:} \quad f_{P-C} : \text{PSS} \rightarrow \text{CSS} \quad (3.2)$$

Both functions  $f_{C-P}$  and  $f_{P-C}$  are non bijective and generally do not have an unambiguous mapping. If the PSS is large and the LV cannot be pinpointed, a decision about the location of an additional instrumentation needs to be made. For simplicity sake, we further consider the placement of additional sensors and leave the placement of actuators to future works. The choice of possible sensor locations is constraint by several aspects discussed in 2. Selecting the optimal position from the remaining PSS is an optimization problem in itself which is highly dependent on the specific system in question. A naive approach would be placing the sensor spatially centered within the PSS, splitting it up evenly. Setting a finite limit to the considered sensor locations is beneficial in order to limit the theoretical iterations of any applied search algorithms.

Once an additional sensor has been placed a new set of measurements is taken. Based on the new data, the assumed causal relations of the added variable need to be verified utilizing causal discovery methods [4, 12] in order to exclude potential confounding effects. These steps are then repeated until no further reduction of the PSS is possible or the LV has been located 3.1.

**Algorithm 3.1** A rough algorithm for LV localization

---

```
while LV location is unknown do
  Choose experiment parameters
  Measure  $\mathbf{X}$ 
  Update  $\mathcal{G}$  via causal discovery algorithm
  for each  $X_i$  in  $\mathcal{G}$  do
    Analyze  $R_i$ 
  end for
  Determine CSS
  Infer PSS
  Place additional sensor according to some metric
end while
```

---

### 3.2 Physical constraints

Physical systems adhere to the laws of physics, intrinsically constricting the solution space through conservation laws. If a physical quantity produces a non negligible residual, the cause can often not be arbitrary. If, for example, we take the mixing of two liquids with different temperatures, and we detect a residual in the temperature of the mixture, it is initially not clear if the LV affects the temperature of either or both of the liquids or their volumes. By further taking the volumes of the mixed and unmixed liquids into consideration, the physical effect of the LV on the system becomes apparent.

When estimating the system's output with a regression network, implementing physical constraints directly into the network's structure or loss function [19, 11, 3] could further highlight apparent conservation law violations and increase performance.

### 3.3 Triangulation

As discussed in 2.2, physical systems are generally of higher dimension than their SCM counterparts. Just because the causal edge where a LV impacts the system has been identified, it does not mean that the LV is found. In spatially

extended systems, the PSS can still be large while the CSS is minimal. One way of tackling this setting is triangulation. By measuring multiple distinct points and quantifying the relative magnitudes of the LV's influence, a suggestion for the LV's location or the next sensor placement could be generated. Similarly, if a neural network is used for predicting the measured values, the relative feature importance of the inputs on the prediction could be evaluated. A relatively dampened feature importance could indicate a sort of spatial shielding of the corresponding input variable by the LV. In a non linear environment, standard triangulation would not be applicable but the problem could then be solved via convex optimization algorithms.

### 3.4 Residual Analysis

Before deciding on a location to place a sensor in the PSS, the CSS needs to be determined. Correlation can be a vital tool to do so. In physical systems, the interaction of a LV with the system usually produces correlation with other variables in the system. Even constant LVs can produce correlation. A puncture in a water pipe for example leads to a loss of fluid volume which is also dependent on the pressure within the pipe. Multiple expressions of correlation exist, some of which were discussed in 2.3. Linear Heteroscedasticity for example can be detected using the Breusch-Pagan test [2].

A more general metric is the mutual information (MI). MI quantifies the information content one random variable  $X$  holds about another random variable  $Y$ .

$$I(X; Y) = D_{KL} (p_{(X,Y)}(x, y) || p_X(x) \otimes p_Y(y)) \quad (3.3)$$

$$= \int_{x \in \mathcal{X}} \int_{y \in \mathcal{Y}} p_{(X,Y)}(x, y) \log \left( \frac{p_{(X,Y)}(x, y)}{p_X(x)p_Y(y)} \right) \quad (3.4)$$

$\mathcal{X}$  and  $\mathcal{Y}$  are the respective domains of  $X$  and  $Y$ ,  $p_{X,Y}$  is the joint probability density function (PDF) of  $X$  and  $Y$ ,  $p_X$  and  $p_Y$  are the marginal PDFs of  $X$  and  $Y$  respectively,  $D_{KL}$  is the Kullback-Leibler divergence and  $\otimes$  the outer product.

The MIs between the systems residuals  $\mathbf{R}$  and related variables  $\mathbf{X}$  enable an

ordering of correlation. For better comparison of the MIs, the normalized mutual information (NMI) can be utilized.

$$\text{NMI}_{ik}(R_i; X_k) = \frac{2 \cdot I(R_i; X_k)}{H(R_i) \cdot H(X_k)} \quad (3.5)$$

$H(\cdot)$  is the entropy and  $X_k$  are the parents of  $X_i$  within the corresponding SCM. MI and NMI are absolute correlation measures and therefore unable to distinguished negative and positive correlation. Revisiting the previous example of two liquids of certain temperatures being mixed, it becomes evident that if a LV impacts the temperature  $t_1$  of the unmixed volume  $V_1$  via an additive noise  $\Delta t_{LV}$ , increasing the volume  $V_2$  directly reduces the residual of the mixing temperature  $R_i = T_{\text{noise}} - T_{\text{model}} = \frac{V_1 \cdot \Delta t_{LV}}{V_1 + V_2}$ . One way to distinguish positive correlation from negative correlation would be Pearson's correlation coefficient  $\rho_{R_i, X_k}$ .

$$\rho_{R_i, X_k} = \frac{\text{cov}(R_i, X_k)}{\sigma_{R_i} \cdot \sigma_{X_k}} \quad (3.6)$$

Where  $\text{cov}(\cdot, \cdot)$  is the covariance and  $\sigma_{(\cdot)}$  the variance.

### 3.5 Noise Estimation

Another way to determine the CSS could be noise estimation. Studies on root cause localization model exogenous noise distributions in order to detect outliers [10]. Similarly, given that all  $f_i$  within the SCM are bijective and known, inverting equation 2.1 would yield a set of probability distributions for all possible locations within the SCM. By making assumptions about the LV such as distribution type, mean or variance, a CSS would be obtained. Given a set of accurate functions  $f_i$  the change of variables formula can be used to calculate the original noise distributions of  $U_{LV}$ .

$$p(U_{LV}) = p(R_i) \left| \frac{dR_i}{dU_{LV}} \right| \quad (3.7)$$

$p(U_{LV})$  and  $p(R_i)$  are the PDFs of the LV and the residual and  $\left| \frac{dR_i}{dU_{LV}} \right|$  is the absolute value of the Jacobian of the transformation. The challenge here lies in creating a method which is resistant to model inaccuracies or even works from a solely data driven model.

### 3.6 Optimal Experimental Design

The task at hand is developing methods to quickly and efficiently mature immature processes. This generally leaves sizable room for designing the specific experiments. To start off, running a set of repeat experiments with constant input parameters allows for quantification of unobserved stochasticity within the system. This information further allows the utilization of physical constraints as discussed in 3.2. Bayesian optimal experimental design (BOED) is a powerful approach which maximizes the information gain of each experiment [6]. A corresponding utility function could measure the reduction in uncertainty about the residuals' distributions or in a further step the LV's distribution itself. Here too, MI could be utilized. Optimal sensor placement, as touched on in 3.3, is another design aspect. In the presence of a non trivial PSS Gaussian processes (GPs) can be utilized [14, 9]. The GP models a spatial field and provides a mean  $\mu(x)$  and an uncertainty, in form of a variance  $\sigma(x)$ , at each point  $x$ . MI, entropy reduction and expected improvement (EI) are some options for optimization metrics. The entropy at any point within a GP is given by  $H(x) = \frac{1}{2} \log(2\pi e \sigma^2(x))$ . By calculating the expected entropy for the whole system after placing an additional sensor, a position which maximally reduces this entropy can be found. EI is a method for finding extrema within the spatial field while considering the likelihood and magnitude of improvement:

$$\text{EI}(x) = (\mu(x) - y_{\text{best}}) \Phi \left( \frac{\mu(x) - y_{\text{best}}}{\sigma(x)} \right) + \sigma(x) \phi \left( \frac{\mu(x) - y_{\text{best}}}{\sigma(x)} \right) \quad (3.8)$$

$y_{\text{best}}$  is the highest or lowest previous point,  $\Phi$  is the cumulative distribution function of the standard normal distribution and  $\phi$  the corresponding PDF.

Some processes, or parts of it, might allow for variables to be adjusted or freely changed during runtime. Besides the obvious increase in collected information per experiment and the applicability of BOED, time series data of the residuals could be of interest. Response times and potential decay rates after input changes in the residual might yield further information regarding the LV position.



## 4 Discussion

In this work, we provided an overview of the different challenges and opportunities of this problem setting. The biggest issue being the generally ambiguous mapping from the CSS to the PSS, as it is highly dependent on the specific physical system, and as of now can not be done automatically in most cases. Upsides are the many degrees of freedom for designing experiments and instrumenting the process. Many aspects, as discussed, can be utilized and optimized in order to locate LVs within a physical process. In order to chain all of these together a unifying methodology which governs the iterative maturation cycle is required. Few work in the scientific community has focused on a fast and efficient maturation of immature manufacturing processes. With it comes a lack of available benchmark data sets which makes evaluating and designing novel methods difficult. Therefore, we see a pressing need for both a unifying methodology for detecting and locating LVs in immature cyber physical systems and for benchmark data sets that represent a wide range of cyber physical systems.

## References

- [1] Miguel Bagajewicz. “A review of techniques for instrumentation design and upgrade in process plants”. In: *The Canadian Journal of Chemical Engineering* 80.1 (2002), pp. 3–16.
- [2] Trevor S Breusch and Adrian R Pagan. “A simple test for heteroscedasticity and random coefficient variation”. In: *Econometrica: Journal of the econometric society* (1979), pp. 1287–1294.
- [3] Luiz FO Chamon et al. “Constrained learning with non-convex losses”. In: *IEEE Transactions on Information Theory* 69.3 (2022), pp. 1739–1760.
- [4] Max Chickering. “Statistically efficient greedy equivalence search”. In: *Conference on Uncertainty in Artificial Intelligence*. Pmlr. 2020, pp. 241–249.

- [5] Yoon Sang Cho and Seoung Bum Kim. “Quality-discriminative localization of multisensor signals for root cause analysis”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.7 (2021), pp. 4374–4387.
- [6] Adam Foster et al. “Variational Bayesian optimal experimental design”. In: *Advances in Neural Information Processing Systems* 32 (2019).
- [7] John Fox. *Applied regression analysis and generalized linear models*. Sage publications, 2015.
- [8] Rudolf J Freund, William J Wilson, and Ping Sa. *Regression analysis*. Elsevier, 2006.
- [9] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. “Near-optimal sensor placements in gaussian processes”. In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 265–272.
- [10] Xiao Han et al. “On root cause localization and anomaly mitigation through causal inference”. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2023, pp. 699–708.
- [11] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.
- [12] Thuc Duy Le et al. “A fast PC algorithm for high dimensional causal discovery with multi-core PCs”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 16.5 (2016), pp. 1483–1495.
- [13] Dylan Molenaar. “Heteroscedastic latent trait models for dichotomous data”. In: *Psychometrika* 80 (2015), pp. 625–644.
- [14] Tomoya Nishida et al. “Region-restricted sensor placement based on Gaussian process for sound field estimation”. In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 1718–1733.
- [15] Eduardo e Oliveira, Vera L Miguéis, and José L Borges. “Automatic root cause analysis in manufacturing: an overview & conceptualization”. In: *Journal of Intelligent Manufacturing* 34.5 (2023), pp. 2061–2078.
- [16] Judea Pearl et al. “Models, reasoning and inference”. In: *Cambridge, UK: CambridgeUniversityPress* 19.2 (2000), p. 3.

- [17] Marco S Reis and Pedro M Saraiva. “Heteroscedastic latent variable modelling with applications to multivariate statistical process control”. In: *Chemometrics and Intelligent Laboratory Systems* 80.1 (2006), pp. 57–66.
- [18] Dongjie Wang et al. “Interdependent causal networks for root cause localization”. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, pp. 5051–5060.
- [19] Yinhao Zhu et al. “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data”. In: *Journal of Computational Physics* 394 (2019), pp. 56–81.



# **A Short Survey on Semantic Occupancy Mapping with 3D-LiDARs in Outdoor Environments**

*Raphael Hagmanns*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
raphael.hagmanns@kit.edu

## **Abstract**

Semantic mapping for mobile robots is a crucial aspect for autonomous navigation and interaction with their environment. Map quality, accuracy of semantic labels and runtime are important dimensions for real-world applications. Voxel models have been widely used to represent occupancy in the map. In recent years, Bayesian Kernel Inference (BKI) has emerged as the main technique on top of traditional occupancy maps to produce smooth maps while maintaining an underlying probabilistic model. Methods based on BKI vary widely in their parameterization and can be tailored to different environments. In this report, we aim to develop the methodology for semantic mapping based on BKI, before evaluating different methods on a range of datasets to assess their parameterization effects for different use cases. We are targeting unstructured outdoor environments where the semantic mapping frameworks need to robustly handle uncertainties in perception.

## **1 Introduction**

3D occupancy mapping with known poses describes the process of building a map representation of an environment by encoding the occupancy state of

discretized voxels into a geometric framework. Generating such a map representation is critical for many robotic downstream applications such as autonomous navigation, planning, or inspection. It also supports human-robot interaction, since the map can be utilized by both the operator and the robot. As a globally consistent representation of the environment, other robots can contribute to the generation of an accurate and lifelong environment representation.

Building accurate maps presents several challenges. First, the geometric consistency of integrating potentially noisy and sparse sensor data. Typical sensors used are 3D LiDARs, as they have a large field of view compared to other depth sensors. On the downside, the generated sensor measurements are sparse, discontinuous, and typically not colorized. Another challenge is the runtime and memory efficiency of the sensor integration. Typically, a voxel occupancy state is determined by raycasting the measurements into the map, which is a costly process considering modern sensors that produce millions of points per second.

Many of these conventional occupancy mapping approaches exist, and the use of advanced datastructures allows them to integrate sensor readings in real time while efficiently encoding the free-space, thus maintaining a manageable memory footprint. However, these methods often lack spatial consistency as all voxels are treated independently. This often leads to inconsistencies in the maps, making them unreliable for downstream applications.

In this report, we consider recent methods that extend purely geometric frameworks by integrating environmental semantics into the occupancy map representation. This is often referred to as *semantic mapping*, allowing both robots and humans to take advantage of a more descriptive scene representation. Enriching the geometric occupancy with a specific class probability also allows to smooth the above mentioned inconsistencies with probabilistic reasoning.

Within the topic of semantic mapping, we focus in particular on a number of smoothing approaches based on *Bayesian Kernel Inference* (BKI). We investigate and evaluate different kernel crafting strategies and discuss their evaluation on outdoor benchmark datasets. Since most of the investigated methods are strongly focused on indoor environments, we also discuss potential difficulties arising from complex unstructured environments, where the boundaries between different classes disappear.

The remainder of this report is organized as follows: Related work is presented in Section 2, before we formulate the problem and establish a common base model for semantic mapping in Section 3. Different smoothing approaches based on BKI are introduced in Section 3.3 before we evaluate and discuss the approaches on several datasets in Section 4. We conclude our report and present potential future research directions in Section 6.

## 2 Related Work

Various representations have been used for purely geometric occupancy mapping ranging from meshes [35], TSDFs [29], Gaussians [28] to surfels [4]. Voxel-based methods are popular and widely used for their simplicity, flexibility and efficiency regarding updates and map queries. Octomap [16] is one of the first voxel-based geometric mapping approaches that uses octrees as an efficient free-space encoding, recent methods use hashmaps [29, 15] or more advanced datastructures such as OpenVDB [27] with enhanced update methods [11, 23, 12, 17].

Due to the flexibility of the information stored in the voxels, the above methods and associated datastructures are also a popular choice for semantic mapping approaches. Most of the voxel-based mapping methods utilize an inverse sensor model by raycasting sensor readings, so that an integration ray can hold a semantic class next to its occupancy information. The semantic class typically comes from a segmented RGB image that is projected onto the LiDAR scan. Some methods directly integrate RGB-D sensors [33], which allows to skip the projection of a 2D information onto the 3D scan data.

Within these voxel-based semantic mapping methods, the main distinguishing feature is the strategy to deal with inconsistencies arising from noisy and sparse sensor data. All methods use local spatial information to propagate the semantics to neighboring voxels and thereby relax the assumption that all voxels are independent.

Earlier methods [31, 21] use Conditional Random Fields (CRF) to enforce consistency potentials at different levels of the hierarchical octree. This can be particularly useful for RGB-D sensor integration, as higher-order potentials

can be established that achieve consistency with 2D superpixels [43, 42, 31]. However, the costly CRF optimization takes over the entire set of occupied cells after a few integration iterations making it impractical for filling sparse areas. Other works [38] are based on Gaussian processes and are well suited to predict the occupancy state of these areas. However, due to the high time complexity of  $\mathcal{O}(N^3)$  in the number of measurements [6, 28], the optimization routine is too expensive to run in real time.

Recent methods mostly rely on BKI to handle map inconsistencies. Doherty et al. propose a continuous counting model [7], which is extended by Gan et al. to a semantic counting model (S-BKI) [9]. Both works use a sparse kernel [24] to predict the occupancy probabilities at unoccupied locations. In later works, S-BKI was refined to fit the procedure to a particular data distribution. In [39], class-specific kernels are crafted using a dataset-specific pre-trained network. SEE-CSOM [5] improved the sparse kernel by adapting its size to the entropy of the particular class. They also use contextual entropy to filter out redundant voxels, thus preventing overinflation. Another recent work [19] incorporates semantic class predictions along with their uncertainty. However, this requires an uncertainty-aware semantic segmentation framework, which is currently based on selected and mostly RGB segmentation networks.

In this report, we summarize and compare the semantic mapping methods based on BKI, as they provide a natural mixture of deep learning-based approaches for the segmentation and reliable and mathematically motivated methods for the map integration. However, recently some methods have emerged that use implicit neural representations to map the environment [22, 30, 2]. While those methods are advantageous in terms of memory efficiency, they often cannot guarantee geometric correctness and are currently mostly used for constrained indoor environments. They rarely leverage LiDAR data and tend to require significant amounts of training data.

### 3 Semantic Mapping

We first introduce the conventional geometric mapping pipeline to demonstrate the general data processing procedure coming from traditional occupancy map-



ping [34]. In the following sections, we then refine the pipeline towards the semantic counting model that is used in recent S-BKI-based methods [9, 39, 5, 19].

### 3.1 Geometric Mapping Pipeline

Traditional occupancy mapping with known poses refers to a framework proposed by Elfes [8] that attempts to find the map posterior  $p(m \mid z_{1:t}, x_{1:t})$  given a set of LiDAR measurements  $z_{1:t}$  at corresponding known poses  $x_{1:t}$ . Since it is intractable to compute the global posterior, one assumption in traditional occupancy mapping is that all grid cells are treated independently [8]. This allows the full posterior to be approximated using the marginals of each individual cell

$$p(m \mid z_{1:t}, x_{1:t}) = \prod_{i=1}^N p(m_i \mid z_{1:t}, x_{1:t}) \quad (3.1)$$

to reduce the required computational effort. Assuming binary states for cells  $m_i$ , plugging into a binary Bayes filter [34] leads to a simple additive update in log-odds space

$$l_{1:t,i} = \log \frac{p(m_i \mid z_{1:t}, x_{1:t})}{p(m_i \mid z_{1:t-1}, x_{1:t-1})} = l_{t,i} + l_{1:t-1,i} - l_0 \quad (3.2)$$

with  $l_{t,i}$  being the current sensor reading from the inverse measurement model,  $l_{1:t-1,i}$  the current occupancy probability and  $l_0 = 0.5$  the prior. Few works [31, 20] extend this model to update an additional multiclass semantic belief

$$p(m_i^k \mid z_{1:t}) = \frac{1}{|K_i|} \sum_{k \in K_i} p(k \mid z_{1:t}) \quad (3.3)$$

by averaging the class occurrence at each voxel  $i$ , where  $K_i \subseteq \{1..K\}$  denotes the set of occurring semantic labels (out of a total of  $K$  classes) at that voxel. In this approach, the semantic belief is treated as stochastically independent of the binary occupancy belief.

### 3.2 Semantic Counting Model

The pure geometric mapping and its semantic extension have several drawbacks. First, the occupancy probability tends to converge towards *occupied* or *free* for a series of different readings, for instance for semi-transparent areas [14]. Second, the model is discrete, so the grid cannot be queried at arbitrary resolutions. Due to the voxel independence assumption, many discontinuities appear in sparse regions. Finally, handling the semantics independently of the occupancy is not convenient. The approaches of Doherty et al. and later Gan et al. try to overcome these drawbacks by employing the counting model of [14], extending it to hold semantic classes [9], and applying a Bayesian kernel to relax the voxel independence assumption and make the map continuous [7].

The basic counting model [14, 7] models assigns each grid cell  $m_i$  some probability  $\theta_i = p(m_i | \mathcal{D}_{1:t})$  of being occupied given a set of measurements  $\mathcal{D} = \{(c_j, z_j)\}_{j=1}^t$  up to time  $t$ , with  $z_j \in \{0, 1\}$  indicating a ray that ends up at cell location  $c_j \in \mathbb{R}^3$ . Thus, we model  $m_i \sim \text{Bernoulli}(\theta_i)$  as the occupancy probability  $\theta_i$  of cell  $m_i$ , seeking the posterior  $p(\theta_i | \mathcal{D})$ . This imposes a conjugate prior  $\text{Beta}(\alpha_0, \beta_0)$  on  $\theta_i$ , which leads to an equally  $\text{Beta}$ -distributed posterior  $\text{Beta}(\alpha_i, \beta_i)$  when applying Bayes' theorem. The parameters  $\alpha_i$  and  $\beta_i$  count the number of rays *reflecting* in a cell  $m_i$  and those passing by, respectively. This results in a simple count

$$\alpha_i = \alpha_0 + \sum_{j \in 1, \dots, t | c_j = m_i} z_j \quad (3.4)$$

$$\beta_i = \beta_0 + \sum_{j \in 1, \dots, t | c_j = m_i} (1 - z_j) \quad (3.5)$$

of all measurements. Note, that one ray produces many measurement pairs  $(c_j, z_j)$ , so technically  $\mathcal{D}$  is a multiset of measurements. However, we index  $\mathcal{D}$  with a single time index  $t$  for simplicity. This model has the closed-form expectation and variance

$$\mathbb{E}[\theta_i] = \frac{\alpha_i}{\alpha_i + \beta_i} \quad \text{and} \quad \mathbb{V}[\theta_i] = \frac{\alpha_i \beta_i}{(\alpha_i + \beta_i)^2 (\alpha_i + \beta_i + 1)} \quad (3.6)$$

which provide simple means of estimating occupancy and filtering out high variance measurements. Again, updates typically take place in log space to

avoid numerical instabilities. Gan et al. extended the approach to multiple categories instead of binary measurements [9]. They redefine  $z_j = (z_j^1, \dots, z_j^K)$  with  $\sum_{k=1}^K z_j^k = 1$  for  $K$  one-hot encoded semantic classes. The occupancy probability  $\theta_i$  becomes a class-specific probability with  $\theta_i = (\theta_i^1, \dots, \theta_i^K)$  and  $\sum_{k=1}^K \theta_i^k = 1$ . The likelihood of inferring a measurement

$$p(z_j | \theta_i) = \prod_{k=1}^K (\theta_i^k)^{z_j^k} \quad (3.7)$$

with probabilities  $\theta_i$  is now modeled with a Categorical distribution. This imposes a Dirichlet  $\text{Dir}(K, \alpha_0)$  with concentration parameters  $\alpha_0^1, \dots, \alpha_0^K$  as conjugate prior, giving another Dirichlet  $\text{Dir}(K, \alpha_i)$  as posterior. Again, the update rule for the concentration parameters  $\alpha_i^1, \dots, \alpha_i^K$  is given by

$$\alpha_i^k = \alpha_0^k + \sum_{j \in 1, \dots, t | c_j = m_i} z_j^k, \quad (3.8)$$

which reassembles a measurement count of each semantic class occurrence in a cell. Similar to the binary case, the expectation and variance again have closed forms given by [9]:

$$\mathbb{E}[\theta_i^k] = \frac{\alpha_i^k}{\sum_{k=1}^K \alpha_i^k} \quad \text{and} \quad \mathbb{V}[\theta_i^k] = \frac{\frac{\alpha_i^k}{\sum_{k=1}^K \alpha_i^k} (1 - \frac{\alpha_i^k}{\sum_{k=1}^K \alpha_i^k})}{\sum_{k=1}^K \frac{\alpha_i^k}{\sum_{k=1}^K \alpha_i^k} + 1}. \quad (3.9)$$

### 3.3 Bayesian Kernel Inference

To relax the independence assumption between cells, Bayesian Kernel Inference (BKI) [36] emerged as the main technique. While the likelihood  $f := p(z_j | \theta_j)$  (where  $\theta_j$  indicates the latent occupancy probability at map cell  $c_j$ ) is based only on  $\theta_j$ , the *extended likelihood*  $g := p(z_j | \theta_*, c_j, c_*)$  relates a observation  $(c_j, z_j)$  to a spatially similar query  $c_*$ . Vega-Brown et al. [36] relate these likelihoods by showing that  $g(z) \propto f(z)^{\kappa(c_*, c)}$  is the maximum entropy distribution that satisfies a smoothness constraint by enforcing a bound on the Kullback-Leibler divergence  $D_{\text{KL}}(g || f)$  between the two distributions. In this context,  $\kappa(\cdot, \cdot)$  is an arbitrary kernel function whose choice will be discussed in the next Section 3.4.

The relationship

$$p(z_j | c_j, \theta_*, c_*) \propto p(z_j | \theta_*)^{\kappa(c_*, c_j)}. \quad (3.10)$$

can be used to perform Bayesian inference [9] on the posterior

$$p(\theta_* | c_*, \mathcal{D}) \propto p(\mathcal{D} | \theta_*, c_*)p(\theta | c_*) \quad (3.11)$$

$$\propto \left[ \prod_{j=1}^t p(z_j | c_j, \theta_*, c_*) \right] p(\theta | c_*) \quad (3.12)$$

$$\propto \left[ \prod_{j=1}^t p(z_j | \theta_*)^{\kappa(c_*, c_j)} \right] p(\theta | c_*) \quad (3.13)$$

of an arbitrary (and possibly non-discrete) location  $c_*$ . Plugging in the Categorical likelihood and the Dirichlet prior again allows for incremental Bayesian updates, since the posterior

$$p(\theta_* | c_*, \mathcal{D}) \propto \left[ \prod_{j=1}^t \left( \prod_{k=1}^K (\theta_*^k)^{z_j^k} \right)^{\kappa(c_*, c_j)} \right] \prod_{k=1}^K (\theta_*^k)^{\alpha_0^k - 1} \quad (3.14)$$

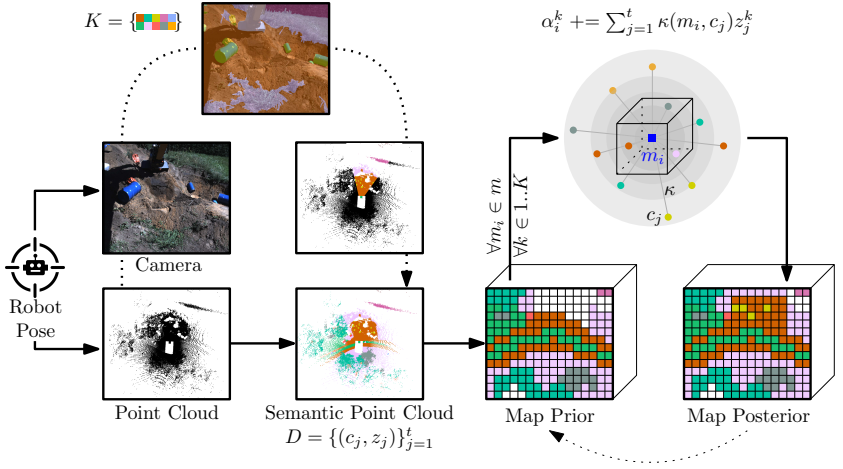
$$\propto \prod_{k=1}^K (\theta_*^k)^{\alpha_0^k - 1 + \sum_{j=1}^t \kappa(c_*, c_j) z_j^k} \quad (3.15)$$

is again a Dirichlet  $\text{Dir}(K, \alpha_t)$ . In this continuous version, the concentration parameters

$$\alpha_*^k = \alpha_0^k + \sum_{j=1}^t \kappa(c_*, c_j) z_j^k \quad (3.16)$$

again reassemble a cell count, but kernel-weighted to take into account the influence of neighboring cells. Therefore, Eq.3.9 still holds for the expectation and variance. At each time step, the concentration parameters are updated for each voxel center as a query point.

The entire processing pipeline is depicted in Figure 3.1. Given a set of measurements  $\mathcal{D}$ , which can be either from a direct semantic point cloud segmentation or from a segmented and projected RGB image, each voxel center performs a



**Figure 3.1:** Overview on semantic mapping pipeline using a kernel-aided semantic counting model. The left part shows the construction of a semantic point cloud, which can be either by direct inference or by RGB segmentation and subsequent projection. The right part shows a single update step to infer the map posterior given the segmented input point cloud.

semantic update according to Eq. 3.16 to derive the map posterior. The decaying influence of the kernel function  $\kappa$  is visualized by grayscale circles around the voxel center.

### 3.4 Choice of Kernels

Several kernel functions have been considered to express an appropriate influence decay of neighboring voxels. The only requirements for a kernel are

$$\kappa(c, c) = 1 \quad \forall c \quad \text{and} \quad \kappa(c, c') \in [0, 1] \quad \forall c, c', \quad (3.17)$$

they do not need to be symmetrical or positive definite [36]. Most approaches are based on the original suggestion [7, 9] of using a sparse kernel [24]

$$\kappa(c, c_*) = \begin{cases} \sigma_0 \left[ \frac{1}{3} \left( 2 + \cos \left( 2\pi \frac{d}{l} \right) \left( 1 - \frac{d}{l} \right) + \frac{1}{2\pi} \sin \left( 2\pi \frac{d}{l} \right) \right) \right] & \text{if } d < l \\ 0 & \text{if } d \geq l \end{cases} \quad (3.18)$$

where  $d = \|c - c_*\|$  is the distance between the query and the training points,  $l$  is a fixed threshold, and  $\sigma_0$  the kernel scale parameter. Since the spatial influence is thresholded by  $l$ , the update time complexity is in  $\mathcal{O}(\log N)$  for each query point, where the influencing  $N$  training points are looked up in a kd-tree [7].

In a follow-up work, Deng et al. [5] propose to adapt the kernel scale  $\sigma_0$  according to different entropy scores obtained at the respective voxel. They suggest a probability entropy  $\mathbf{E}_p$  measuring the proportion between the count of all measurements and the maximum count of measurements for a particular class. In addition, a semantic entropy  $\mathbf{E}_s = \frac{\log k}{\log K}$  reflects the label diversity in a voxel by relating the number of occurring semantic classes  $k$  to the overall number of classes  $K$ . Both subentropies are combined to form a class entropy

$$\mathbf{E}_{\text{class}} = \mathbf{E}_p + \frac{1}{K} \mathbf{E}_s \quad (3.19)$$

for each voxel. To avoid overinflation during kernel inference, they additionally suggest a context entropy  $\mathbf{E}_{\text{con}}$  that indicates how much a voxel would contribute to the completeness of surrounding voxels, allowing to filter out voxels that are potentially outside object boundaries. Finally, they extend the sparse kernel from Eq. 3.18 with a filtering factor for  $\mathbf{E}_{\text{con}}$  and with an adapted kernel scale  $\sigma_0$  based on  $\mathbf{E}_{\text{class}}$ .

Kim, Seo, and Min extend the entropy-based approach in [19] by incorporating an evidential deep learning (EDL) module into the upstream semantic segmentation procedure. This provides uncertainty estimates of the predicted semantic annotations, which they use to update the fixed-length threshold  $l$  within the sparse kernel to be  $l = l_0 \cdot \beta e^{1-\gamma u_*}$ , where  $\beta$  and  $\gamma$  are hyperparameters and  $u_*$  is the uncertainty associated with the query point  $c_*$ .

The approach of Wilson et al. [39] attempts to learn kernels for specific semantic classes and environments. They rewrite the cell-wise update from Eq. 3.16 to a

convolution layer leveraging differentiable kernel functions to create a trainable variant of the BKI update. Since this process allows GPU based inference, it allows for faster runtimes [39]. The custom kernels are again based on the sparse kernel with an adapted length that supports the structure of the object. In addition, they propose a compound kernel composed of horizontal and vertical components to better represent the geometric representations of each semantic class.

### 3.5 Datastructures for Mapping

All methods introduced so far work on voxel grids as dense map representations, although other surface or volume based representations exist (cf. Section 2). In this section, we are concerned with the underlying data structure, which strongly influences the memory consumption and the inference time of the chosen map representation. The main requirement for voxel-based representations is a discrete grid with index coordinates that can be converted to spatial coordinates and vice versa. Naive implementations of such grids can use continuous storage with mappings from grid cell to array indices, more advanced versions leverage hashmaps to retain constant time lookups while reducing memory requirements and the need for reallocation. Popular examples of methods using hashmap structures are FIESTA [15] or the GPU-based OHM [32]. Many modern approaches [16, 3] use hierarchical structures such as octrees or kd-trees, with Octomap [16] being a community standard for many years. Tree-based structures typically allow for a sparse representation in memory at the cost of tree traversals during insert or delete operations. All of the semantic mapping approaches presented [6, 9, 39, 19] also use octrees as the underlying structure but provide a lightweight hashing mechanism for faster access to octree nodes. Advanced data structures such as VDB [27] take this a step further and combine fixed-depth trees and block hashmaps to achieve good tradeoffs between memory efficiency and almost constant time lookup operations for general-purpose mapping [23, 37, 11, 12].

## 4 Experimental Evaluation

We compare BKI-based approaches [9, 39, 19, 5] in terms of computational cost and map accuracy. To do this, we compute class-based and average IoU scores:

$$\text{IoU}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k + \text{FN}_k} \quad \text{mIoU} = \sum_{k=1}^K \text{IoU}_k \quad (4.1)$$

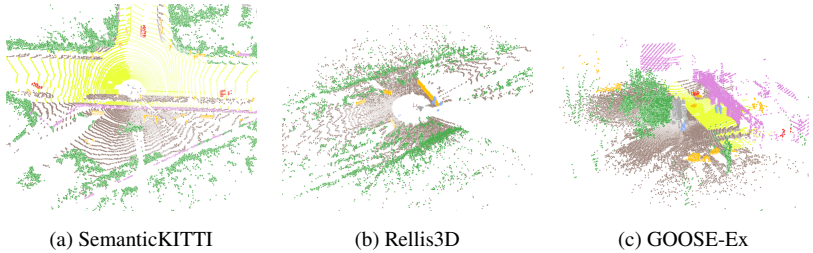
where  $\text{TP}_k, \text{FP}_k$  and  $\text{FN}_k$  are the counts of true positive, false positive and false negative queries of the ground truth point (coming from the ground truth semantic segmentation) queries against the constructed map. To measure computational cost, we evaluate runtime and memory consumption for different sequences.

Since we are primarily interested in outdoor environments, we evaluate the approaches on the SemanticKITTI [1], the Rellis3D [18] as well as the GOOSE and GOOSE-Ex [26, 13] datasets. While the first contains only sequences in structured, urban environments, the others are particularly useful for evaluating the mapping performance in unstructured environments. Accurate mapping in these environments is difficult because classes tend to overlap and merge into each other. In addition, the transition between different terrain classes is often not smooth, as shown in the example image and point cloud pairs depicted in Figure 4.1.

Previous works often reported different results for an apparently similar benchmark set. A fair comparison between semantic mapping approaches seems to be difficult due to the complex pipeline setup. One way to reduce this complexity would be to avoid custom segmentation pipelines, and instead use the ground truth segmentation as input to evaluate the semantic mapping capabilities only. However, this strategy does not allow to measure the performance on noisy inference methods. Another aspect that has emerged in previous evaluations [42, 9] is the exclusion of certain classes or areas from the evaluation.

Instead, we use a simplified and unified label map across all used datasets, consisting of the 8 classes *vegetation*, *artificial ground*, *natural ground*, *artificial structures*, *obstacle*, *vehicle*, *person*, *other*. For SemanticKITTI, we use the same reference labels used in [9] and originally inferred with RangeNet++ [25],





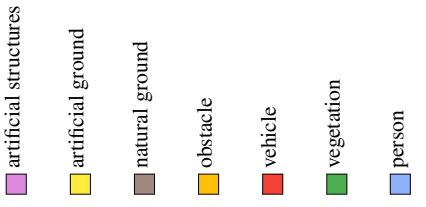
**Figure 4.1:** Example point cloud annotations of the datasets used for evaluating semantic mapping approaches. The SemanticKITTI [1] dataset contains a very long range, but sparse point cloud of urban environments. Both Rellis3D [18] and GOOSE-Ex [13] datasets are recorded in unstructured environments. Rellis3D has some labeling artifacts and the ground truth is obtained by projection of a semantically annotated image, whereas GOOSE contains dense point-wise annotated labels. All datasets have been unified to use the same set of labels.

but mapped to the above classes. For both Rellis3D and GOOSE, we preprocess the dataset labels to map to the above 8 labels and then train a PTv3 [41] semantic segmentation model that works directly on the point clouds. We use the inferred point clouds for all method variants. We ignore labels of the *other* class during calculation of the mIoU due to an insignificant number of occurrences.

We not consider the KITTI odometry benchmark [10] although many of the earlier approaches used it as a benchmark. Since the LiDAR segmentation is only calculated by projection of the front camera segmentation, the field of view is rather small and the evaluation would be different from the other datasets, as the semantic map has to be projected back onto the image to obtain the IoU scores. For the same reason, we did not run the evidential variant of [19], since it would require multimodal fusion of image and point cloud pairs. We only use their baseline implementation of S-BKI and the basic counting model (CSM) to compare with the original implementation. In future work, evidential deep learning could be applied directly to a point cloud segmentation network to simplify the processing pipeline and increase the comparability.

For each evaluated sequence, we use all available frames and for each inserted point cloud, we perform IoU calculation for the previous 10 ground truth clouds.

**Table 4.1:** Quantitative evaluation results of different methods on different datasets. The labels of each datasets have been grouped into eight categories according to the GOOSE ontology [26]. (\*) Note, that we not use the core idea of E-BKI [19] as we directly used segmented point clouds that doesn't come with a uncertainty estimate. We utilize their code for baseline comparison due to a more recent, more efficient and clean implementation of S-BKI [9].

Dataset	Method								mIoU / %
		artificial structures	artificial ground	natural ground	obstacle	vehicle	vegetation	person	
Rellis3D 04	<i>CSM</i> [6] [19]	0.0	53.9	75.0	70.6	60.0	68.1	87.9	59.4
	<i>PTv3</i> [41]	0.0	55.3	74.9	71.5	48.6	67.7	77.8	56.5
	<i>S-BKI</i> [9]	0.0	53.4	75.4	71.1	59.7	68.3	80.1	58.3
	<i>ConvBKI</i> [39]	0.0	53.3	75.3	<b>71.6</b>	<b>69.0</b>	67.4	82.7	<b>59.9</b>
	<i>SEE-CSOM</i> [5]	0.0	53.2	72.9	69.4	62.2	66.0	75.5	57.0
	<i>(E)-S-BKI</i> (*) [19]	0.0	<b>53.6</b>	<b>75.5</b>	70.2	59.5	<b>68.6</b>	<b>87.0</b>	59.2
Semantic- KITTI 04	<i>CSM</i> [6] [19]	80.4	94.7	72.6	38.6	87.7	88.2	29.0	70.2
	<i>RangeNet++</i> [25]	76.2	93.7	67.2	31.5	80.0	84.9	15.7	64.2
	<i>S-BKI</i> [9]	79.9	<b>94.4</b>	71.5	38.6	<b>86.4</b>	87.8	26.0	69.2
	<i>ConvBKI</i> [39]	78.9	94.2	<b>73.7</b>	33.0	85.2	87.5	<b>37.7</b>	<b>70.0</b>
	<i>SEE-CSOM</i> [5]	<b>80.6</b>	93.6	72.7	<b>39.3</b>	80.9	<b>88.3</b>	28.5	69.1
	<i>(E)-S-BKI</i> (*) [19]	77.8	93.5	67.5	38.5	83.6	86.2	21.4	66.9
GOOSE 05	<i>CSM</i> [6] [19]	42.9	65.6	55.9	38.7	2.1	54.2	0.0	37.1
	<i>PTv3</i> [41]	47.2	77.0	58.2	55.2	7.1	56.5	0.0	43.0
	<i>S-BKI</i> [9]	47.0	76.0	57.8	53.2	4.7	55.6	0.0	42.0
	<i>ConvBKI</i> [39]	<b>47.8</b>	<b>77.5</b>	<b>60.0</b>	<b>53.3</b>	<b>7.0</b>	<b>60.5</b>	0.0	<b>43.7</b>
	<i>SEE-CSOM</i> [5]	41.6	65.8	55.8	37.9	0.3	52.4	0.0	36.3
	<i>(E)-S-BKI</i> (*) [19]	46.8	77.4	58.2	52.5	2.6	56.4	0.0	42.0

The evaluation on the GOOSE datasets works slightly different, as the ground truth annotations are not available for every frame, but only for some frames

within a sequence. We still integrate each inferred point cloud, but perform the IoU calculation only for frames with available ground truth.

The input point cloud is downsampled to a grid of size 0.1 m. The rays are then integrated to their maximum length, using a map resolution of 0.3 m. The sampling resolution for generating sensor readings for free space is set to 100. The default occupancy mapping thresholds for freeing and occupying a voxel are set to 0.3 and 0.7, respectively. For the sparse kernel, we use  $\sigma_0 = 0.1$  as scale and  $l = 0.3$  as length threshold for the base variants of S-BKI [9]. For SEE-CSOM [5] we use the default parameters as presented in their work. For ConvBKI [40], we train class-specific compound kernels on the respective test-sets of each dataset. We exclude the sequences that are used for evaluation.

In Table 4.1 we compare different methods with respect to of class IoUs and mIoU. We include the discrete counting model (CSM) as baseline. However, since the evaluation against ground truth labels is inherently performed only on occupied voxels, the results are quite similar compared to continuous methods. For both the GOOSE and Rellis3D datasets, we include the PTV3 semantic segmentation results to make the semantic mapping results more comparable to methods with different segmented inputs. The results show a similar performance of different methods on all three datasets. SEE-CSOMs [5] improvements to prevent overinflation seem to work well especially on sharp-edged classes such as *artificial structures* and *obstacle*, for the more unstructured classes, the gap closes. The ConvBKI [40] approach of learning class-specific kernels seems to be promising, but cannot develop its full potential on our selection of classes, especially for the unstructured scenes. While all methods outperform the segmentation baseline for SemanticKITTI, this is not the case for each class in Rellis3D and GOOSE. Figure 4.2 shows the qualitative mapping results of the three sample sequences using S-BKI [9].

Table 4.2 shows the runtimes and memory consumption of the above methods. For all experiments, we used a 6-core Intel®Core™i7-10850H and 32 GB of memory. CSM and the cleaner S-BKI implementation from [19] exhibit a small memory footprint compared to the other variants. The average runtime per frame increases significantly with long range and dense lidar scans. It can be reduced by filtering the input point cloud as a preprocessing step. ConvBKI [40] is the only method that achieves runtimes in the order of magnitude to keep up with a

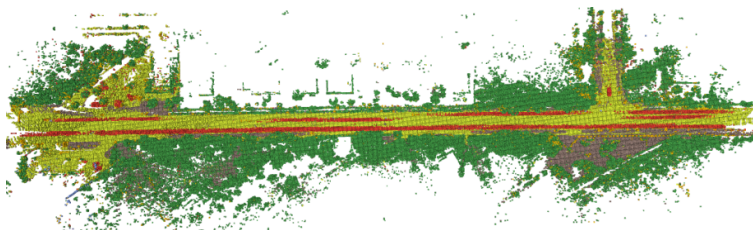
reasonable sensor rate. This fast runtime results from a full GPU implementation of the kernel convolution operations. None of the other variants come close to running at a sensor rate of 10 hz.

**Table 4.2:** Runtime and memory consumption of different methods on different sequences. (\*) Both GPU and CPU memory are included. The required GPU memory is bound to a preselected grid size and resolution.

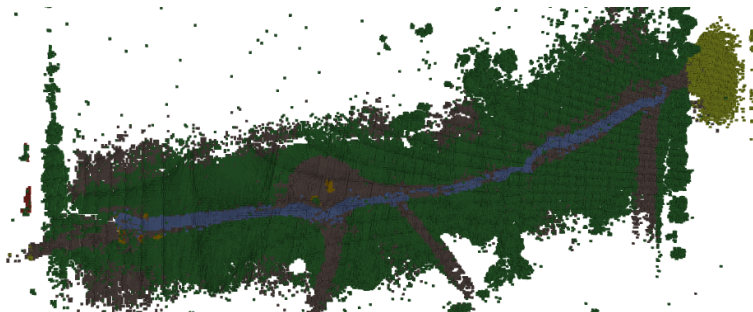
Method	runtime per frame / s			memory / GB		
	Rellis3D	KITTI	GOOSE	Rellis3D	KITTI	GOOSE
<i>CSM</i> [6] [19]	8.5	18.0	221.1	0.35	0.57	1.78
<i>S-BKI</i> [9]	8.2	24.8	38.7	2.08	6.17	6.54
<i>ConvBKI</i> [39] (*)	<b>0.3</b>	<b>0.3</b>	<b>0.3</b>	5.59	5.70	5.86
<i>SEE-CSOM</i> [5]	5.5	16.2	22.7	14.70	8.79	13.16
<i>(E)-S-BKI</i> [19]	7.9	12.4	34.5	<b>0.26</b>	<b>0.40</b>	<b>1.22</b>

## 5 Conclusion and Future Work

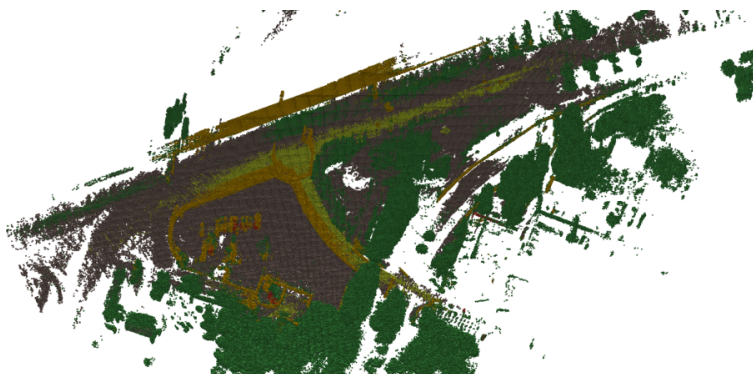
We provide a concise survey of currently available semantic mapping methods that use typical occupancy structures and Bayesian Kernel Inference to provide continuous occupancy information. The main difference between the presented methods lies in their choice of kernels to achieve a good trade-off between map accuracy and runtime. Our evaluations on different datasets with a focus on unstructured environments reveal some potential improvements. Incorporating some kind of uncertainty measure [19] from the inference method seems very useful, but could be extended to work directly on point clouds. For CPU-based variants, more advanced data structures could be used to speed up the geometric integration process. Using GPU-accelerated convolutions as proposed in ConvBKI [40] is a promising idea to achieve sensor-rate performance and could easily be extended to take advantage of uncertainty-based updates.



(a) Map of SemanticKITTI Seq. 04



(b) Map of Rellis3D Seq. 04



(c) Map of GOOSE Seq. 05

**Figure 4.2:** Resulting maps using the standard S-BKI [9] approach on the entire sequence.

## References

- [1] J. Behley et al. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: 2019.
- [2] Alexandra Carlson et al. “CLONeR: Camera-Lidar Fusion for Occupancy Grid-Aided Neural Representations”. In: *IEEE Robotics and Automation Letters* 8.5 (2023), pp. 2812–2819. DOI: 10.1109/LRA.2023.3262139.
- [3] Jing Chen and Shaojie Shen. “Improving octree-based occupancy maps using environment sparsity with application to aerial robot navigation”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3656–3663. DOI: 10.1109/ICRA.2017.7989419.
- [4] Xieyuanli Chen et al. “SuMa++: Efficient LiDAR-based Semantic SLAM”. In: 2019, pp. 4530–4537. DOI: 10.1109/ROSL40897.2019.8967704.
- [5] Yinan Deng et al. “SEE-CSOM: Sharp-Edged and Efficient Continuous Semantic Occupancy Mapping for Mobile Robots”. In: *IEEE Transactions on Industrial Electronics* 71.2 (2024), pp. 1718–1728. DOI: 10.1109/TIE.2023.3262857.
- [6] Kevin Doherty, Jinkun Wang, and Brendan Englot. “Bayesian generalized kernel inference for occupancy map prediction”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3118–3124.
- [7] Kevin Doherty et al. “Learning-Aided 3-D Occupancy Mapping With Bayesian Generalized Kernel Inference”. In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 953–966. DOI: 10.1109/TR0.2019.2912487.
- [8] Alberto Elfes. “Using occupancy grids for mobile robot perception and navigation”. In: *Computer* 22.6 (1989), pp. 46–57.
- [9] Lu Gan et al. “Bayesian Spatial Kernel Smoothing for Scalable Dense Semantic Mapping”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 790–797. DOI: 10.1109/LRA.2020.2965390.
- [10] A. Geiger, P. Lenz, and R. Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: 2012, pp. 3354–3361.

- [11] Marvin Grosse Besselmann et al. “VDB-Mapping: A High Resolution and Real-Time Capable 3D Mapping Framework for Versatile Mobile Robots”. In: Aug. 2021, pp. 448–454. DOI: 10.1109/CASE49439.2021.9551430.
- [12] Raphael Hagmanns et al. “Efficient Global Occupancy Mapping for Mobile Robots using OpenVDB”. In: *Workshop Agile Robotics: Perception, Learning, Planning, and Control at IROS2022* (2022).
- [13] Raphael Hagmanns et al. “Excavating in the Wild: The GOOSE-Ex Dataset for Semantic Segmentation”. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2025).
- [14] Dirk Hähnel. *Mapping with mobile robots*. Jan. 2004.
- [15] Luxin Han et al. “FIESTA: Fast Incremental Euclidean Distance Fields for Online Motion Planning of Aerial Robots”. In: *CoRR* abs/1903.02144 (2019). arXiv: 1903.02144.
- [16] Armin Hornung et al. “OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees”. In: *Autonomous Robots* (2013). Software available at <http://octomap.github.com>. DOI: 10.1007/s10514-012-9321-0.
- [17] Ignacio Martin Vizzo. “Robot Mapping with 3D LiDARs”. PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, May 2024.
- [18] Peng Jiang et al. “Rellis-3d dataset: Data, benchmarks and analysis”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 1110–1116.
- [19] Junyoung Kim, Junwon Seo, and Jihong Min. *Evidential Semantic Mapping in Off-road Environments with Uncertainty-aware Bayesian Kernel Inference*. 2024. arXiv: 2403.14138 [cs.R0].
- [20] Deyvid Kochanov et al. “Scene flow propagation for semantic mapping and object discovery in dynamic street scenes”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 1785–1792. DOI: 10.1109/IR0S.2016.7759285.

- [21] Abhijit Kundu et al. “Joint semantic segmentation and 3d reconstruction from monocular video”. In: *European Conference on Computer Vision (ECCV)*. 2014, pp. 703–718.
- [22] S. Lionar et al. “NeuralBlox: Real-Time Neural Representation Fusion for Robust Volumetric Mapping”. In: 2021, pp. 1279–1289. DOI: 10.1109/3DV53792.2021.00135.
- [23] Steve Macenski, David Tsai, and Max Feinberg. “Spatio-temporal voxel layer: A view on robot perception for the dynamic world”. In: *International Journal of Advanced Robotic Systems* 17.2 (2020). DOI: 10.1177/1729881420910530.
- [24] Arman Melkumyan and Fabio Ramos. “A Sparse Covariance Function for Exact Gaussian Process Inference in Large Datasets”. In: 2009, pp. 1936–1942.
- [25] A. Milioto et al. “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019.
- [26] Peter Mortimer et al. “The GOOSE Dataset for Perception in Unstructured Environments”. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2024).
- [27] Ken Museth et al. “OpenVDB: An Open-Source Data Structure and Toolkit for High-Resolution Volumes”. In: *ACM SIGGRAPH 2013 Courses. SIGGRAPH ’13*. Anaheim, California: Association for Computing Machinery, 2013. ISBN: 9781450323390. DOI: 10.1145/2504435.2504454.
- [28] Simon T O’Callaghan and Fabio T Ramos. “Gaussian process occupancy maps”. In: *The International Journal of Robotics Research* 31.1 (2012), pp. 42–62.
- [29] Helen Oleynikova et al. “Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017.
- [30] Songyou Peng et al. “Convolutional Occupancy Networks”. In: ed. by Andrea Vedaldi et al. 2020, pp. 523–540. ISBN: 978-3-030-58580-8.



- [31] Sunando Sengupta and Paul Sturgess. “Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order MRF”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1874–1879. DOI: 10.1109/ICRA.2015.7139442.
- [32] Kazys Stepanas et al. “OHM: GPU Based Occupancy Map Generation”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11078–11085. DOI: 10.1109/LRA.2022.3196145.
- [33] Jörg Stückler, Nenad Biresev, and Sven Behnke. “Semantic mapping using object-class segmentation of RGB-D images”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 3005–3010.
- [34] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005. ISBN: 9780262201629.
- [35] Julien PC Valentin et al. “Mesh based semantic modelling for indoor and outdoor scenes”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 2067–2074.
- [36] William R Vega-Brown, Marek Doniec, and Nicholas G Roy. “Non-parametric Bayesian inference on multivariate exponential families”. In: *Advances in Neural Information Processing Systems (NeurIPS)* 27 (2014).
- [37] Ignacio Vizzo et al. “VDBFusion: Flexible and Efficient TSDF Integration of Range Sensor Data”. In: *Sensors* 22.3 (2022). ISSN: 1424-8220. DOI: 10.3390/s22031296.
- [38] Jinkun Wang and Brendan Englot. “Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1003–1010.
- [39] Joey Wilson et al. “Convolutional Bayesian Kernel Inference for 3D Semantic Mapping”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 8364–8370. DOI: 10.1109/ICRA48891.2023.10161360.

- [40] Joey Wilson et al. “Convolutional Bayesian Kernel Inference for 3D Semantic Mapping”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 8364–8370. DOI: 10.1109/ICRA48891.2023.10161360.
- [41] Xiaoyang Wu et al. “Point Transformer V3: Simpler, Faster, Stronger”. In: *CVPR*. 2024.
- [42] Shichao Yang, Yulan Huang, and Sebastian Scherer. “Semantic 3D occupancy mapping through efficient high order CRFs”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 590–597. DOI: 10.1109/IROS.2017.8202212.
- [43] Zhe Zhao and Xiaoping Chen. “Building 3D semantic maps for mobile robots using RGB-D camera”. In: 9 (Oct. 2016). DOI: 10.1007/s11370-016-0201-x.

# **A Discussion on Shape Models in Extended Object Tracking for Radar-based Intelligent Transportation Systems**

*Longfei Han*

Fraunhofer IVI, Ingolstadt  
Karlsruhe Institute of Technology (KIT), Germany  
longfei.han@ivi.fraunhofer.de

## **Abstract**

This study evaluates the shape models in Extended Object Tracking (EOT) for Radar (radio detection and ranging)-based Intelligent Transportation Systems (ITS). In the perception systems of ITS, road-side sensors are required to cover large areas and deliver efficient descriptions of the scene. Radar sensors with their long sensing distance, make a good candidate for the task. Their high resolution, combined with the extended object tracking, which estimates the position, orientation and shape of the object, could provide a detailed description of the scene. The discussion will address the shapes models employed in such tracking applications, along with the requirements for their implementation. Furthermore, the coexistence of sensors within a network has the potential to enhance the accuracy of multi-object tracking results. The study also considers scenarios, particularly those involving a decentralized fusion scheme, in which the tracking results are combined. The study culminates in the proposal of a concept for a shape model for radar-based ITS.

# 1 Introduction

Intelligent Transportation Systems (ITS) leverage the sensors installed in road infrastructure to enhance perception for autonomous driving applications, smart traffic analytics, and safer interaction between different road users. Radio-detection-and-ranging (radar) sensors further benefit these scenarios by providing independent sensing ability that is less affected by weather conditions and by covering a large area of interest with a single unit. However, despite advancements in radar resolution, the current radar measurement data is not distributed in a manner that allows humans to semantically label the sensor data, hindering the development of learning-based algorithms for object detection and tracking. Furthermore, alternative sensors capable of labeling radar data are limited in their ability to discern distant objects which are only visible to radars. Nonetheless, the community of researchers of sensor data fusion has long engaged with low-resolution radar data for target state estimation. The recent advancements in 3D sensing resolution have precipitated the rapid evolution of Extended Object Tracking (EOT), a model-based target state estimation approach that encompasses shape and orientation estimation. This development has enabled the advancement of radar perception without the need for a substantial number of manual labels. Concurrently, these models have deep roots in probabilistic theories, which can provide uncertainty measures for decision-making processes. In this study, our objective is to analyse EOT problem to estimate the states of the existing objects in a given scene.

Moreover, when incorporating roadside sensors into a network for the purpose of information sharing, many approaches adhere to a centralized scheme [5]. It is imperative to explore decentralized schemes that exhibit enhanced robustness in ITS, where sensor data is processed locally and the results are then shared. However, there is a paucity of research addressing decentralized radar perception in ITS and the EOT community. Consequently, we consider decentralized fusion of posterior estimations as we discuss the shape model employed in conjunction with EOT.

The remainder of the paper is organized as follows: Section 2 provides a concise overview of related works in the field of EOT along with decentralized schemes and tracking in Lie groups. Section 3 provides a detailed analysis of requirements

of EOT algorithms for radar perception. It also compares main shape models in EOT approaches with this respect and introduces a new concept for the EOT tracking. Section 4 concludes the paper and provides a road map for realization of the concept.

## 2 Related Work

Extended object tracking is a process that estimates the spatial extent, denoted by  $\underline{x}_f$ , of the object along with its kinematic states, denoted by  $\underline{x}_k$ . This estimation is made possible by leveraging the increasing resolution of the sensor data [10]. The extended object can be initially represented by elliptic shapes. The shape of the extended object can be represented by a matrix with randomness spanning an inverse Wishart distribution, denoted  $\mathcal{IW}(X; v, V)$  [15],  $v$  being the degrees of freedom and  $V$  the scale matrix. Alternatively, the ellipse can also be parametrized directly by the semi-axis lengths and the orientation angle of the ellipse  $[l_1, l_2, \alpha]$  [27]. It is further possible to convert elliptic models into a rectangle representation [21]. In the pursuit of a more detailed object model, star convex models are introduced [2]. The models utilize radial functions to represent the distances from center of the object to the points along its contour. The radial function can further be parameterized with Fourier coefficients [2] and basis points in recursive Gaussian process regression [25]. The measurement points on the surface can then be represented by the hypersurfaces with a scaling of the contour between 0 and 1. Another way to model the object shape is to use the spline curves, which have a strong foundation in the computer graphics. Spline curves have been demonstrated to be effective in representing geometric shapes while exhibiting robustness against numerical instabilities [20]. They can be used freely to represent shapes even non-star-convex ones. In [14, 6, 3], Bsplines are used to represent rectangles with varying width and length scales to fit the top view contour of a vehicle. They also have been used in [28] to represent the elongated vehicles.

In addition to the shape models, distributed fusion of estimated extended object has also been addressed in [9, 26], where covariance intersection [13] is used. The possible correlation is then considered in a sub-optimal way, since the

cross correlation between sensors is unknown and treated with conservatism. This could be further exploited in radar-based ITS. In theory, the methods for tracking the 2D ground vehicles can be extended to 3D. Given 3D or 4D radars (3D position along with radial doppler velocity), 3D shape models are to get more attention. Some dedicated work use the Gaussian process to model the radial function in a spherical coordinate system [16, 8] or NURBS(non-uniform rational B splines) surfaces to model the shape with predefined control points [22] and varying scales and weights.

In the domain of 3D tracking, it is imperative to employ a model that is capable of handling the object pose with six degrees of freedom. Lie groups and the Lie theory are of particular relevance. A Lie group  $(G, \circ)$  is a smooth manifold with a composition operation  $\circ : G \circ G \rightarrow G$  [24]. The operation further satisfies axioms of identity, inversity and associativity. Lie theory provides rigorous calculus foundation for uncertainty, derivatives and integral for tracking in relevant Lie groups like  $SE(2)$  and  $SE(3)$ . It assists in enhancing the precision of shape orientation determination. The application of Lie theory in conjunction with extended object models has been demonstrated in the context of tracking space debris [19, 18, 17]. It is noteworthy that not only does the tracker operate within the  $SE(3)$  framework, but a gaussian distribution within  $SE(3)$  further facilitates shape representation. In addition, although their presence has not been observed, integration of Lie group based variants for Kalman filters, such as the Invariant EKF [1] and the Iterated EKF [4] into the EOT trackers should result in improved orientation management.

### 3 Concepts

The following list enumerates the requirements for EOT in radar-based ITS:

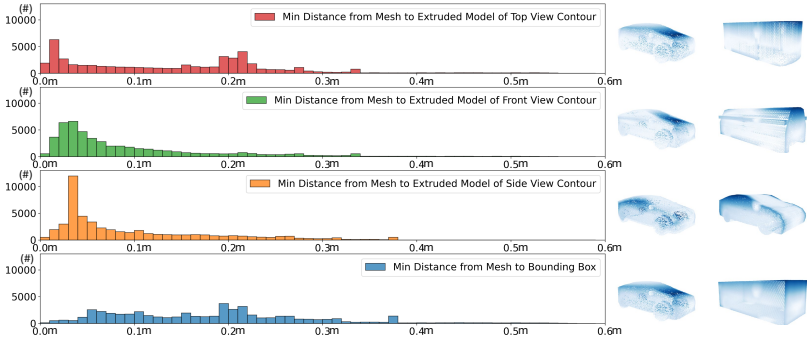
1. The 3D state estimation is accurate with respect to the kinematic state, the orientation, and the spatial extent. This involves establishing shape models that suit the tracked traffic participants correctly. In order to handle the pose correctly, Lie theory-based filters should be used.

2. The formation of a network has the potential to expand the monitoring area and enhance the observability of each object. In a connected system, it is essential to limit the data volume to ensure that communication costs remain affordable and transmission speeds are optimized. When fusing estimations from multiple sensors, the correlation of the data should be taken into consideration.

A direct approach involves the utilization of three-dimensional models to represent road users [16, 8, 22]. These models typically need a significantly greater number of parameters to describe the model in comparison to their two-dimensional counterparts. This increased complexity leads to an escalation in both computation time and transmission cost. Consequently, we advocate for the adoption of the extruded model of the two-dimensional view contours.

Before the extrusion of the 2D models introduced in Section 2, the initial discussion focuses on identifying the optimal view of the processed object for effective representation. A comparative analysis is conducted between extruded models of the top, front, and side views of the object. The mesh of a vehicle model from the CARLA simulator [7] is utilized as an example, with 50,000 points sampled randomly from the mesh. The minimum distance from each sampled point to the nearest point in the extruded models is calculated, indicating the maximum permissible error in the modeling process. The margin should be kept small. The extruded model of the side view yields the smallest range of errors. When using other extruded models as well as a bounding box, the error ranges are found to be larger. It should be noted that the distribution is based on a uniformly distributed sample on the mesh surface. However, if the sensor detects particular components of the vehicle, the distribution may be subject to variation. In cases where the vehicle approaches and departs from the sensor, it is expected that the opposite side in the width direction will be obscured from view. However, the error distribution should exhibit a mirrored pattern with respect to the visible side if it could be seen.

Fig 3.1 further shows on the right side that the greatest distances are calculated on the motor hood. 2D extended shape estimation working with data projected on the x-y plane would result in either the bounding box or an extended model of the top view contour. Both of these cannot avoid the significant error on



**Figure 3.1:** A comparison of extruded 2D contour views. The distances from each point on the true mesh to the model are calculated and organized into bins of histograms, which is shown on the left side of the figure. The right side of the figure depicts the mesh and the extruded models for reference. The color ranges from white indicating small distances, and blue, indicating large distances. From top to bottom, extruded models from top view, front view and side view, as well as a bounding box are shown respectively.

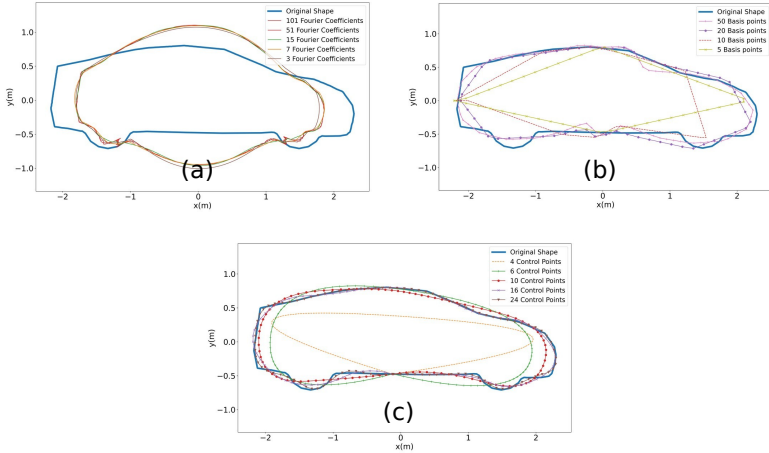
the motor hood. In contrast, side view contour shows smaller error distances on the motor hood, though it has errors on the top edge of the left and right surfaces. The error distances are around  $0.3\text{ m}$ . We draw conclusion here that when extruding 2D shape contour to 3D model, the side view is the preferred option.

We therefore focus on the extruded model of the side-view contour and examine the use of different models to generate it. In comparison to elliptic models, star-convex models and spline models are capable of preserving more details in the modeling. The star-convex model was initially formulated with the Random Hypersurface Model(RHM) [2]. For the points  $p$  on a surface  $\mathcal{S}$ , we have:

$$\mathcal{S}(p) = \{s \cdot r(\underline{b}, \phi) \cdot e(\phi) + \underline{x}_c \mid \phi \in [0, 2\pi] \text{ and } s \in [0, 1]\} \quad (3.1)$$

Here,  $s$  denotes the scaling factor to generate hypersurfaces and to cover the points on the surface. When  $s$  equals 1, the points are located on the contour. The  $e$  is defined as the unit direction vector. The radial function  $r(\underline{b}, \phi)$  maps the angle  $\phi$  with the help of parameters  $\underline{b}$  to the distance between the center of the





**Figure 3.2:** A comparison of different models for representing the side view contour of a vehicle. (a) shows the model based on the radial function and its Fourier coefficients, with the number of coefficients ranging from 3 to 101. (b) shows the model based on the radial function and Gaussian process for it. In the recursive regression process, basis points are used to represent the shape at fixed angles. The model is demonstrated in various shapes, ranging from 5 to 50 basis points. (c) shows the model based on Bsplines. The model is parameterized with 4 to 24 control points.

object and its contour. The  $r(\underline{b}, \phi)$  can be further represented using the Fourier coefficients [2] or Gaussian process [25].  $\underline{x}_c$  denotes the center of the object. As illustrated in Fig 3.2 (a), the reconstructed shape of the side view tends to be a circle using Fourier coefficients. An increase in the number of coefficients does not result in substantial enhancements. This is because that the Fourier serie is conducted on the star-convex formulation. In (b), the recursive GP solver [12] gives good shape estimation only when the basis points are more than 20. However, the profile around the wheels cannot be captured with adequate precision, mainly due to the fact that this component is not star-convex. Another option for modelling the side contour is the B-spline. The radial function is not necessary. The contour is directly represented using control points, a knot vector and a one-dimensional parameter. Main stream work uses least-square method to perform batch optimization in curve-fitting approximation problems [23]. An

example result is shown in Fig. 3.2 (c). Here different numbers of control points are demonstrated. It is evident that reasonable shape models can be obtained with a minimum of 10 control points. Furthermore, an increase in the number of control points can result in a better profile around the wheels.

A preferable concept is therefore based on the Bspline curve recursive approximation. The number of the control points is to be set to a fixed value while their position can vary freely. For some of the feature points, we could also use less parameter like the scaling factor, to keep the number of parameter small while maintaining a greater number of control points. For instance two points at  $(-\frac{\text{length}}{2}, 0)$  and  $(\frac{\text{length}}{2}, 0)$  have only one parameter. The knot vector is set to be uniform. Kalman filter based recursive approximation like [11] hasn't been seen in EOT, but can be integrated. It requires to formulate the Bspline into its matrix formulation [20]: For a Bspline with a knot vector  $\underline{k} = (k_j)_{j=1}^{n+d+1}$ , where  $n$  denotes the number of control points and  $d$  the degree of the Bspline, we use an integer  $\mu$  to represent the index of  $k$ ,  $d + 1 \leq \mu \leq n$ , and use positive integer  $o$  for index of Matrix  $\mathbf{R}_o$ :

$$\mathbf{R}_o(\kappa) = \begin{bmatrix} \frac{k_{\mu+1}-\kappa}{k_{\mu+1}-k_{\mu+1-o}} & \frac{\kappa-k_{\mu+1-o}}{k_{\mu+1}-k_{\mu+1-o}} & 0 & \cdots & 0 \\ 0 & \frac{k_{\mu+2}-\kappa}{k_{\mu+2}-k_{\mu+2-o}} & \frac{\kappa-k_{\mu+2-o}}{k_{\mu+2}-k_{\mu+2-o}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{k_{\mu+o}-\kappa}{k_{\mu+o}-k_{\mu}} & \frac{\kappa-k_{\mu}}{k_{\mu+o}-k_{\mu}} \end{bmatrix} \quad (3.2)$$

We then have for  $\kappa \in [k_{\mu}, k_{\mu+1})$  the  $d + 1$  Bspline function  $\{B_{j,d}\}_{j=\mu-d}^{\mu}$

$$\mathbf{B}_d^{\top} = (B_{\mu-d,d} \ B_{\mu-d+1,d} \ \cdots \ B_{\mu,d}) = \mathbf{R}_1(\kappa)\mathbf{R}_2(\kappa)\cdots\mathbf{R}_d(\kappa). \quad (3.3)$$

Combined with the control points  $\underline{c}_j$  (underline here indicates vector), a Spline curve  $f(\kappa), \kappa \in [k_{\mu}, k_{\mu+1})$  becomes:

$$f(\kappa) = \mathbf{R}_1(\kappa)\mathbf{R}_2(\kappa)\cdots\mathbf{R}_d(\kappa)\underline{c}_d. \quad (3.4)$$

Note  $\underline{c}_d = (\underline{c}_{\mu-d}, \underline{c}_{\mu-d+1}, \cdots, \underline{c}_{\mu})^{\top}$ . This is to be incorporated into the measurement equation of the Kalman filter for the purpose of further development. The concrete algorithm and its validation are beyond the scope of this paper and will be published in the near future.

Working with the side view requires that the object's orientation be managed with high precision. To this end, the implementation of a Lie group-based Kalman filter within the tracking process is preferred. This approach will facilitate the determination of precise uncertainty measures in the pose and the control points of the B-spline. For the decentralized fusion, the employment of a Lie theory-based covariance intersection method is imperative. Additionally, the uncertainty in the control points can be analyzed more explicitly than the parameters like Fourier coefficients or Gaussian process mean values.

In summary, the objective of this study is to develop a methodology for performing 3D extended object tracking for radar-based ITS. The proposed approach involves the utilization of B-spline-based contour description for the side view. To this end, a dedicated tracker with recursive approximation for curve fitting will be developed. This tracker will further leverage Lie theory to properly handle the incremental value in the pose of the object in both tracking and decentralized fusion.

## 4 Conclusion

In this work, we discuss on the shape models for EOT in radar-based ITS. We provide an example comparison of different views to demonstrate its representational ability. The extruded model of a side-view contour could be a suitable candidate for tracking. A further comparison between star-convex models and B-spline models indicates the benefit of explicitly using B-splines. We propose a concept based on modeling the side-view contour and extruding it for a 3D model. The proposed concept integrates a (extended) Kalman filter tracker with Lie theory for tracking, and it fuses pose and control points using a covariance intersection for robust estimation, considering potential correlations between posterior results across multiple sensors.

## References

- [1] Axel Barrau and Silvère Bonnabel. “The Invariant Extended Kalman Filter as a Stable Observer”. In: *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1797–1812. DOI: 10.1109/TAC.2016.2594085.
- [2] Marcus Baum and Uwe D. Hanebeck. “Extended Object Tracking with Random Hypersurface Models”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.1 (2014), pp. 149–159. DOI: 10.1109/TAES.2013.120107.
- [3] Tim Baur et al. “Tracking of Spline Modeled Extended Targets Using a Gaussian Mixture PHD Filter”. In: *2019 22th International Conference on Information Fusion (FUSION)*. 2019, pp. 1–8. DOI: 10.23919/FUSION43075.2019.9011298.
- [4] Guillaume Bourmaud et al. “From Intrinsic Optimization to Iterated Extended Kalman Filtering on Lie Groups”. In: *Journal of Mathematical Imaging and Vision* 55.3 (July 2016), pp. 284–303. ISSN: 1573-7683. DOI: 10.1007/s10851-015-0622-8. URL: <https://doi.org/10.1007/s10851-015-0622-8>.
- [5] Christian Creß, Zhenshan Bing, and Alois C. Knoll. “Intelligent Transportation Systems Using Roadside Infrastructure: A Literature Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 25.7 (2024), pp. 6309–6327. DOI: 10.1109/tits.2023.3343434.
- [6] Karl-Magnus Dahlén et al. “An Improved B-spline Extended Object Tracking Model using the Iterative Closest Point Method”. In: *2022 25th International Conference on Information Fusion (FUSION)*. 2022, pp. 1–8. DOI: 10.23919/FUSION49751.2022.9841363.
- [7] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.
- [8] Felix Ebert and Hans-Joachim Wuensche. “Dynamic Object Tracking and 3D Surface Estimation using Gaussian Processes and Extended Kalman Filter”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 1122–1127. DOI: 10.1109/ITSC.2019.8916891.

- 
- [9] Markus Fröhle, Karl Granström, and Henk Wymeersch. “Decentralized Poisson Multi-Bernoulli Filtering for Vehicle Tracking”. In: *IEEE Access* 8 (2020), pp. 126414–126427. DOI: 10.1109/ACCESS.2020.3008007.
- [10] Karl Granström and Marcus Baum. “A Tutorial on Multiple Extended Object Tracking”. In: (Feb. 2022). DOI: 10.36227/techrxiv.19115858.v1. URL: <http://dx.doi.org/10.36227/techrxiv.19115858.v1>.
- [11] M. Harashima, L.A. Ferrari, and P.V. Sankar. “Spline approximation using Kalman filter state estimation”. In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 44.5 (1997), pp. 421–424. DOI: 10.1109/82.580860.
- [12] Marco F. Huber. “Recursive Gaussian process: On-line regression and learning”. In: *Pattern Recognition Letters* 45 (2014), pp. 85–91. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2014.03.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0167865514000786>.
- [13] Simon Julier and Jeffrey K Uhlmann. “General decentralized data fusion with covariance intersection”. In: *Handbook of multisensor data fusion*. CRC Press, 2017, pp. 339–364.
- [14] Hauke Kaulbersch, Jens Honer, and Marcus Baum. “A Cartesian B-Spline Vehicle Model for Extended Object Tracking”. In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018, pp. 1–5. DOI: 10.23919/ICIF.2018.8455717.
- [15] Johann Wolfgang Koch. “Bayesian approach to extended object and cluster tracking using random matrices”. In: *IEEE Transactions on Aerospace and Electronic Systems* 44.3 (2008), pp. 1042–1059. DOI: 10.1109/TAES.2008.4655362.
- [16] Murat Kumru and Emre Özkan. “3D Extended Object Tracking Using Recursive Gaussian Processes”. In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018, pp. 1–8. DOI: 10.23919/ICIF.2018.8455480.

- [17] S. Labsir et al. “Joint shape and centroid position tracking of a cluster of space debris by filtering on Lie groups”. In: *Signal Processing* 183 (2021), p. 108027. ISSN: 0165-1684. DOI: <https://doi.org/10.1016/j.sigpro.2021.108027>. URL: <https://www.sciencedirect.com/science/article/pii/S0165168421000669>.
- [18] Samy Labsir et al. “A Lie-group based modelling for centroid and shape estimation of a cluster of space debris”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 960–964. DOI: 10.23919/Eusipco47968.2020.9287641.
- [19] Samy Labsir et al. “Tracking a Cluster of Space Debris in Low Orbit by Filtering on Lie Groups”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 5481–5485. DOI: 10.1109/ICASSP.2019.8682409.
- [20] Tom Lyche and Knut Morken. “Spline methods draft”. In: *Department of Informatics, Center of Mathematics for Applications, University of Oslo, Oslo* (2008), pp. 3–8.
- [21] Florian Meyer and Jason L. Williams. “Scalable Detection and Tracking of Geometric Extended Objects”. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 6283–6298. DOI: 10.1109/TSP.2021.3121631.
- [22] Benjamin Naujoks, Patrick Burger, and Hans-Joachim Wuensche. “Fast 3D Extended Target Tracking using NURBS Surfaces”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 1104–1109. DOI: 10.1109/ITSC.2019.8917384.
- [23] Les Piegl and Wayne Tiller. *The NURBS book (2nd ed.)* Berlin, Heidelberg: Springer-Verlag, 1997. ISBN: 3540615458.
- [24] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. *A micro Lie theory for state estimation in robotics*. 2021. arXiv: 1812.01537 [cs.R0]. URL: <https://arxiv.org/abs/1812.01537>.
- [25] Niklas Wahlström and Emre Özkan. “Extended Target Tracking Using Gaussian Processes”. In: *IEEE Transactions on Signal Processing* 63.16 (2015), pp. 4165–4178. DOI: 10.1109/TSP.2015.2424194.

- [26] Jiaye Yang et al. “Distributed Posterior Fusion for Vehicle Tracking with Gaussian Processes”. In: *2023 12th International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2023, pp. 218–223. DOI: 10.1109/ICCAIS59597.2023.10382388.
- [27] Shishan Yang and Marcus Baum. “Tracking the Orientation and Axes Lengths of an Elliptical Extended Object”. In: *IEEE Transactions on Signal Processing* 67.18 (2019), pp. 4720–4729. DOI: 10.1109/TSP.2019.2929462.
- [28] Antonio Zea, Florian Faion, and Uwe D. Hanebeck. “Tracking elongated extended objects using splines”. In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, pp. 612–619.





# Multi-stage process modeling using Gaussian Processes

*Saksham Kiroriwal*

Kognitive Industrielle Systeme (KIS)  
Fraunhofer IOSB, Germany  
saksham.kiroriwal@iosb.fraunhofer.de

## Abstract

In multi-stage processes, dependence on noisy observations of the intermediates is a problem to overcome to predict the outputs accurately. This requires a Multi-Stage Gaussian Process (MSGP)- a modeling idea to incorporate such intermediate observations, considering various observation likelihoods effectively. The MSGP may further boost predictive performance by indirectly observing the multi-stage process by adopting Directed Acyclic Graph (DAG) architecture and Variational Inference (VI) methods. Such a model would use the prior information and increase the accuracy of inference, making Bayesian optimization and prediction effective in situations where one can hardly make direct observations.

## 1 Introduction

In most practical applications, processes are not confined to one stage but usually involve many stages with associated intermediate outputs. This, in turn, calls for a more sophisticated modeling approach to capturing the details of the real-world system. Traditional approaches to modeling nonlinear input-output relationships typically use single black box GPs [12, 18, 4]. More recently, the work by [2, 1, 13, 11] showed improved modeling and optimization results when the intermediate observations were included.

Most existing GPNs adopt a graph structure, where every node represents a process stage with recorded intermediate outputs. Current models often assume independent training of nodes. This is restrictive and can only deal with noise-free observations, not noisy observations. We propose a new conceptual framework, called the Multi-Stage Gaussian Process (MSGP), which will be able to incorporate the noisy intermediate observations robustly to improve inference. By extending established inference methods from Stochastic Variational GP (SVGP) [7]. MSGP will likely facilitate more effective Bayesian optimization. This paper discusses the broader idea of MSGP, why it is required, and how it could be achieved.

## 2 Multi-stage process

In complex systems, a multi-stage stochastic process, denoted as  $\mathbf{M}$ , is composed of  $B$  interconnected subprocesses  $M_{(1)}, \dots, M_{(B)}$ . Each subprocess  $M_{(b)}$  can be expressed using function  $\mathbf{g}_{(b)}$ . Each subprocess accepts some adjustable parameters  $\mathbf{s}_{(b)}$  and outputs from parent subprocesses  $M_{\rho(b)}$ . The process function operation yields outputs  $\tilde{\mathbf{g}}_{(b)}$ . These outputs are further influenced by stochastic noise  $\tilde{\eta}_{(b)}$ . This noise follows a distribution  $p(\eta_{(b)}|\mathbf{s}_{(b)})$ , capturing the inherent randomness of each subprocess. An example process can be shown as

$$\tilde{\mathbf{g}}_{(1)} = \mathbf{g}_{(1)}(\mathbf{s}_{(1)}) + \tilde{\eta}_{(1)}; \tilde{\mathbf{g}}_{(2)} = \mathbf{g}_{(2)}(\mathbf{s}_{(2)}, \tilde{\mathbf{g}}_{(1)}) + \tilde{\eta}_{(2)},$$

where  $\tilde{\eta}_{(b)} \sim p(\eta_{(b)}|\mathbf{s}_{(b)}) \forall \{1, \dots, B\}$ .

Another way to express the stochasticity of the process is using a function space. Using this definition, the process  $\mathbf{M}$  can be redefined as

$$\tilde{\mathbf{g}}_{(1)} \sim p(\mathbf{g}_{(1)}; \mathbf{s}_{(1)}); \tilde{\mathbf{g}}_{(2)} \sim p(\mathbf{g}_{(2)}; \{\mathbf{s}_{(2)}, \tilde{\mathbf{g}}_{(1)}\}).$$

Often, the subprocess outputs are observed indirectly, producing observed outputs  $\tilde{\mathbf{t}}_{(b)}$ , rendering the true outputs latent. This indirect observation can be modeled using a likelihood function. The challenge lies in modeling these final outputs using the adjustable inputs and indirect observations through varied likelihoods while leveraging the known data generation structure of the process  $\mathbf{M}$ .

We wish to incorporate a network-based modeling approach to avoid augmenting intermediate observations with inputs and not letting the input dimensionality blow up. This work proposes a Multi-Stage Gaussian Process (MSGP) to infer outputs using indirect observations by treating the latent outputs as a joint normal distribution.

### 3 Background

In this section, we discuss the existing Gaussian Process framework for a single black-box modeling approach where the inputs  $\{\mathbf{s}_{(b)}\} \forall b \in \{1, \dots, B\}$  are concatenated to model the final output  $\mathbf{t} = \mathbf{t}_{(B)}$ .

#### 3.1 Gaussian Process

A *Gaussian process* (GP) can be viewed as a probability distribution over functions defined on the input domain  $\mathcal{S} \subseteq \mathbb{R}^{D_s}$ . Conventional GP is used to model scalar values observations and use scalar values functions sampled from the function space. Concretely, it assigns a multivariate normal distribution to the set of function values evaluated at any finite collection of inputs. Formally, one writes

$$g(\mathbf{s}) \sim \mathcal{GP}\left(m(\mathbf{s}), k(\mathbf{s}, \mathbf{s}')\right), \quad (3.1)$$

where  $m(\cdot) : \mathbb{R}^{D_s} \rightarrow \mathbb{R}$  is the mean function, and  $k(\cdot, \cdot) : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$  is the covariance (kernel) function.

#### Posterior Inference

Suppose we observe  $R$  data points  $\{\mathbf{s}_r, t_r\}_{r=1}^R$ , where each  $\mathbf{s}_r \in \mathbb{R}^{D_s}$  denotes an input location and  $t_r \in \mathbb{R}$  is the corresponding observed output. Let  $\mathbf{S} = \{\mathbf{s}_r\}_{r=1}^R$  and  $\mathbf{t} = \{t_r\}_{r=1}^R$ . We place a GP prior over the latent function values  $\mathbf{g} = \{g_r\}_{r=1}^R$ , encoded as

$$p(\mathbf{g}; \mathbf{S}) = \mathcal{N}\left(m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}')\right), \quad (3.2)$$

and define the likelihood for each observation  $t_r$  as  $p(t_r | g_r)$ . Using Bayes' rule, the posterior over the latent function values, given all observations, is

$$p(\mathbf{g} | \mathbf{t}; \mathbf{S}) = \frac{p(\mathbf{t} | \mathbf{g}) p(\mathbf{g}; \mathbf{S})}{\int p(\mathbf{t} | \mathbf{g}) p(\mathbf{g}; \mathbf{S}) d\mathbf{g}}. \quad (3.3)$$

This posterior encapsulates how the observed data update the prior GP assumptions.

### Marginal Likelihood and Hyperparameter Optimization

The kernel  $k(\cdot, \cdot')$  and the mean function  $m(\cdot)$  usually include hyperparameters (e.g., length-scale, signal variance). These are often optimized by maximizing the *marginal log-likelihood* (MLL) of the observed data:

$$\mathcal{L}_{\text{GP}} = \sum_{r=1}^R \log \mathbb{E}_{p(g_r; \mathbf{s}_r)} \left[ p(t_r | g_r) \right]. \quad (3.4)$$

In the special case of a Gaussian likelihood with additive noise variance  $\sigma_t^2$ , the marginal distribution  $p(\mathbf{t} | \mathbf{S})$  becomes  $\mathcal{N}(m(\mathbf{S}), k(\mathbf{S}, \mathbf{S}') + \sigma_t^2 \mathbf{I})$ , leading to a closed-form expression for Equation 3.4. However, solving for the exact posterior in this scenario requires  $\mathcal{O}(R^3)$  operations due to the inversion of an  $R \times R$  covariance matrix.

This cubic complexity presents computational challenges when  $R$  is large or non-Gaussian likelihoods are used. In those cases, one typically employs approximations such as sparse GPs or variational methods to reduce the computational load while retaining the desirable properties of Gaussian process models [15].

## 3.2 Stochastic Variational Gaussian Process

When the data size grows large or the observation model is non-Gaussian, optimizing a Gaussian process (GP) by directly maximizing the MLL in Equation 3.4 becomes prohibitive [20]. As a remedy, *Stochastic Variational Gaussian Process* (SVGP) [7, 8] provides a principled approximation scheme that addresses both scenarios.

### Inducing Points and Their Distribution

SVGP introduces a finite set of inducing locations  $\mathbf{J} = \{\mathbf{j}_k\}_{k=1}^K$ , where each  $\mathbf{j}_k \in \mathbb{R}^{D_s}$  is drawn from the same domain as the observed inputs  $\{\mathbf{s}_r\}$ . The corresponding GP function values, called inducing points, at these pseudo-inputs are  $\mathbf{w} = \{w_k\}_{k=1}^K$ . Their prior distribution under the GP is

$$p(\mathbf{w}; \mathbf{J}) = \mathcal{N}\left(m(\mathbf{J}), k(\mathbf{J}, \mathbf{J}')\right), \quad (3.5)$$

which complements the exact GP prior in Equation 3.2. To facilitate variational inference, one also specifies a Gaussian variational distribution for these inducing variables,

$$q(\mathbf{w}) = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}}\right).$$

### Variational Approximation

Because  $\mathbf{w}$  and  $\mathbf{g}$  (the GP values at the observed inputs) come from the same underlying GP, one can write their joint Gaussian distribution by combining Equations 3.2 and 3.5 (see also [14]). Based on this joint model, the SVGP framework defines a variational posterior for  $\mathbf{g}$  as follows:

$$\begin{aligned} q(\mathbf{g}; \mathbf{S}, \mathbf{J}) &= \int p(\mathbf{g} | \mathbf{w}; \mathbf{S}, \mathbf{J}) q(\mathbf{w}) d\mathbf{w} \\ &= \mathcal{N}\left(\boldsymbol{\mu}_{q(\mathbf{g})}, \boldsymbol{\Sigma}_{q(\mathbf{g})}\right), \end{aligned} \quad (3.6)$$

where the closed-form expressions for the mean and covariance are given by the following formulae

$$\begin{aligned} \boldsymbol{\mu}_{q(\mathbf{g})} &= m(\mathbf{S}) + \boldsymbol{\alpha}(\mathbf{S})^\top \left(\boldsymbol{\mu}_{\mathbf{w}} - m(\mathbf{J})\right), \\ \boldsymbol{\Sigma}_{q(\mathbf{g})} &= k(\mathbf{S}, \mathbf{S}) - \boldsymbol{\alpha}(\mathbf{S})^\top \left(k(\mathbf{J}, \mathbf{J}) - \boldsymbol{\Sigma}_{\mathbf{w}}\right) \boldsymbol{\alpha}(\mathbf{S}), \\ \boldsymbol{\alpha}(\mathbf{S}) &= k(\mathbf{J}, \mathbf{J})^{-1} k(\mathbf{J}, \mathbf{S}). \end{aligned} \quad (3.7)$$

By conditioning only on  $K$  inducing points (with  $K \ll R$ ), we are able to keep the computational costs tractable. This is even the case for large datasets or complex likelihood models.

## Variational Objectives

To optimize both the GP hyperparameters and the variational parameters, one maximizes an Evidence Lower BOund (ELBO) [8]:

$$\mathcal{L}_{\text{SVGP, ELBO}} = \sum_{r=1}^R \mathbb{E}_{q(g_r)} [\log p(t_r | g_r)] - \beta \text{KL}(q(\mathbf{w}) \parallel p(\mathbf{w})),$$

where KL is the Kullback–Leibler divergence. Monte Carlo estimation is typically used to approximate the expectation term.

## Predictive Log Likelihood

Alternatively, the Parametric Predictive GP Regressor (PPGPR) [10] proposes maximizing a Predictive Log Likelihood (PLL):

$$\mathcal{L}_{\text{SVGP, PLL}} = \sum_{r=1}^R \log \mathbb{E}_{q(g_r)} [p(t_r | g_r)] - \beta \text{KL}(q(\mathbf{w}) \parallel p(\mathbf{w})),$$

and reports improved predictive performance in some settings. However, the required expectation does not admit a closed-form solution for non-conjugate likelihoods, and various approximations must be employed [8, 9].

## 3.3 Deep Gaussian Process

A *Deep Gaussian Process* (DGP) [3] extends standard Gaussian processes by stacking multiple GPs in a hierarchical structure. Consider a model with  $N$  layers, where the  $j^{\text{th}}$  layer outputs a vector  $\mathbf{g}^{(j)} \in \mathbb{R}^{D_j}$ . Each layer’s outputs serve as inputs to the subsequent layer. Formally, if  $\mathbf{g}_r^{(j-1)}$  denotes the output of layer  $j - 1$  for the  $r^{\text{th}}$  data point, then it becomes the input to layer  $j$ , yielding output  $\mathbf{g}_r^{(j)}$ . Unlike a single-layer GP, this hierarchical setup induces a non-Gaussian marginal distribution at the final layer, making closed-form inference based on the MLL intractable.

### Doubly Stochastic Variational Inference

A well-known technique for approximate inference in DGPs is *doubly stochastic variational inference* [17], which generalizes the SVGP approach to multiple layers. As in single-layer SVGP, one introduces a set of inducing locations

and corresponding inducing points for each layer. For layer  $j$ , let  $\mathbf{J}^{(j-1)} \in \mathbb{R}^{K(j) \times D_{j-1}}$  denote the inducing locations, with  $K(j)$  being the number of inducing points for that layer and  $\mathbf{W}^{(j)} \in \mathbb{R}^{K(j) \times D_j}$  the corresponding function values under the GP prior. By assuming independence across layers [17], each layer’s marginal depends only on the previous layer.

Following Equations 3.6 and 3.7, one writes the variational distribution at the final layer  $N$  as

$$q\left(\mathbf{g}_r^{(N)} \mid \mathbf{g}_r^{(N-1)}, \mathbf{J}^{(N-1)}\right) = \int \prod_{j=1}^N q\left(\mathbf{g}_r^{(j)} \mid \mathbf{g}_r^{(j-1)}, \mathbf{W}^{(j)}, \mathbf{J}^{(j-1)}\right) d\mathbf{g}_r^{(N-1)}, \quad (3.8)$$

where we treat  $\mathbf{g}_r^{(0)} = \mathbf{s}_r$  as the original input, and  $q(\mathbf{g}_r^{(N)})$  is understood as a distribution rather than a single scalar. Because this setup involves a nested composition of distributions, an exact evaluation of the likelihood term is not feasible. Instead, one resorts to Monte Carlo (MC) approximations for the inner expectations [17], sampling  $\tilde{\mathbf{g}}_r^{(j)} \sim q(\mathbf{g}_r^{(j)})$  layer by layer.

### Variational Objective

Given the layered structure, the ELBO for the DGP is derived by summing over all observations and penalizing the divergence between the variational and prior distributions for the inducing points of each layer. Specifically,

$$\begin{aligned} \mathcal{L}_{\text{DGP, ELBO}} &= \sum_{r=1}^R \mathbb{E}_{q(\mathbf{g}_r^{(N)})} \left[ \log p\left(\mathbf{t}_r \mid \mathbf{g}_r^{(N)}\right) \right] \\ &\quad - \beta \sum_{j=1}^N \text{KL}\left(q(\mathbf{W}^{(j)}) \parallel p(\mathbf{W}^{(j)})\right). \end{aligned} \quad (3.9)$$

This objective can be optimized via gradient-based methods, with MC sampling used for the expectation term in non-conjugate settings. By repeatedly sampling through the layers—hence the phrase “doubly stochastic”—one obtains an effective approximate posterior that captures multiple levels of latent structure.

## 4 Gaussian Process Networks and Related Work

### Gaussian Process Networks (GPN)

The so-called *Gaussian Process Network* was first proposed in [5] and extended in [6] for the problem of learning a Bayesian network structure in a Gaussian process framework. These early formulations focus on structure learning about how nodes are connected in a directed acyclic graph (DAG) without considering inference, given noisy observations of intermediate subprocesses. A different line of work, *Gaussian Process Regression Networks* (GPRN), was proposed by [5, 21], where each node in a neural network–like graph and its outputs are modeled via linear combinations of node outputs in conjunction with Gaussian processes. However, GPRNs constitute yet another approach to multi-output regression. Thus, as they do not specifically consider intermediate observations, their problem setting cannot be similar to the one examined here.

A more recent usage of GPNs as *surrogate models* is discussed by [2, 1]. Each subprocess  $M^{(b)}$  is treated as a Gaussian process:

$$g^{(b)}(\cdot) \sim \mathcal{GP}^{(b)}(m^{(b)}(\cdot), k^{(b)}(\cdot, \cdot')),$$

and the associated noisy measurement is given by the likelihood  $p(t^b | g^{(b)})$ . Because each node’s GP is assumed independent given its observed input-output pairs, one can employ standard closed-form marginal log-likelihood [15] to train each GP individually. Monte Carlo (MC) sampling propagates the final predictions through the network.

Despite being computationally convenient, this strategy leads to three main limitations:

1. **Noisy Observations vs. Latent Inputs:** In practice, the child nodes often depend on the *latent* parent outputs, not the noisy observations of the parent. However, the GPN framework provides the child nodes with the observed noisy values, valid only under noise-free conditions.
2. **Deterministic Input Assumption:** Every output of each parent node in GP is a distribution rather than a fixed number. At training, GPNs rely on their closed-form marginal log-likelihood, which depends on a



deterministic input provided to the child GP. For prediction, MC sampling is conducted instead, resulting in a discrepancy between what has been trained and what is inferred.

3. **Restriction to Gaussian Likelihood:** Since the approach relies on exact marginal log-likelihood, it is restricted to Gaussian observation models. This is too restrictive in practice where the noise could be non-Gaussian, and the real-world process might require amortized likelihoods due to high-dimensional intermediate observations.

Models in [19, 13] partially overcome the first issue by explicitly modeling latent values but still relying on independent node inference via exact marginal log-likelihood, leaving the second and third constraints unresolved. Moreover, GPN-based frameworks presented in [1, 19, 2] are mostly surrogate models for Bayesian optimization with intermediate data rather than general-purpose inference methods.

### **Gaussian Process Autoregressive Regression (GPAR)**

An alternative GPN variant, known as *Gaussian Process Autoregressive Regression* (GPAR), has been introduced by [16]. GPAR sequentially models each observed output by treating previous outputs (in a fixed or greedily chosen order) as inputs to downstream GPs in an autoregressive fashion. Although GPAR demonstrates strong multi-output predictive performance, it does not fully resolve the second and third limitations above nor provides a scalable solution for selecting the order of outputs. Furthermore, when the underlying structure is a DAG (rather than a simple chain), it remains unclear how to accommodate more complex dependencies while retaining GPAR’s flexibility.

## **5 Discussion**

Following the variational inference techniques from Section 3, we propose a multi-stage Gaussian process that can mitigate the major limitations of the current state-of-the-art GPN. Unlike the GPN, which uses the exact marginal log-likelihood while directly feeding the noisy observations from the parental

nodes into the child processes, the new model would incorporate a variational formulation capable of naturally handling stochastic outputs from parent nodes and non-Gaussian likelihoods. In particular, a variational posterior would be assigned for each subprocess using inducing points, similar to the case of SVGP or DGP. There are also hierarchical dependencies between layers, which thus allows MSGP to model observation noise disentangled from true latent outputs and use Monte Carlo samples for training and prediction phases. This ensures that the same way of treating uncertainty is followed throughout all the steps of the network. Besides, MSGP naturally could support high-dimensional intermediate observations by extending the VI methods for amortized inference strategies and avoiding rigid assumptions for closed-form marginal likelihood. It would thus provide a flexible, scalable surrogate modeling tool applicable to most real-world multi-stage systems where the intermediate measurements may be noisy, non-Gaussian, or both while keeping the desirable properties of Bayesian inference.

## References

- [1] Virginia Aglietti et al. “Causal bayesian optimization”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 3155–3164.
- [2] Raul Astudillo and Peter Frazier. “Bayesian optimization of function networks”. In: *Advances in neural information processing systems* 34 (2021), pp. 14463–14475.
- [3] Andreas Damianou and Neil D Lawrence. “Deep gaussian processes”. In: *Artificial intelligence and statistics*. PMLR. 2013, pp. 207–215.
- [4] P. Frazier and Jialei Wang. “Bayesian optimization for materials design”. In: *arXiv: Machine Learning* (2015). doi: 10.1007/978-3-319-23871-5\_3.
- [5] Nir Friedman and Iftach Nachman. “Gaussian process networks”. In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. UAI’00. Stanford, California: Morgan Kaufmann Publishers Inc., 2000, pp. 211–219. ISBN: 1558607099.

- [6] Enrico Giudice, Jack Kuipers, and Giusi Moffa. “A Bayesian take on Gaussian process networks”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [7] James Hensman, Nicolo Fusi, and Neil D Lawrence. “Gaussian processes for big data”. In: *arXiv preprint arXiv:1309.6835* (2013).
- [8] James Hensman, Alexander Matthews, and Zoubin Ghahramani. “Scalable variational Gaussian process classification”. In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 351–360.
- [9] Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. “Deep sigma point processes”. In: *Conference on uncertainty in artificial intelligence*. PMLR. 2020, pp. 789–798.
- [10] Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. “Parametric gaussian process regressors”. In: *International conference on machine learning*. PMLR. 2020, pp. 4702–4712.
- [11] Saksham Kiroriwal et al. “Joint Parameter and State-Space Modelling of Manufacturing Processes using Gaussian Processes”. In: *IEEE International Conference on Industrial Informatics* (2024).
- [12] R. Kontar, Shiyu Zhou, and J. Horst. “Estimation and monitoring of key performance indicators of manufacturing systems using the multi-output Gaussian process”. In: *International Journal of Production Research* 55 (2017), pp. 2304–2319. doi: 10.1080/00207543.2016.1237791.
- [13] Shunya Kusakawa et al. “Bayesian optimization for cascade-type multi-stage processes”. In: *Neural Computation* 34.12 (2022), pp. 2408–2431.
- [14] Felix Leibfried et al. “A tutorial on sparse Gaussian processes and variational inference”. In: *arXiv preprint arXiv:2012.13962* (2020).
- [15] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer, 2003, pp. 63–71.
- [16] James Requeima et al. “The gaussian process autoregressive regression model (gpar)”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1860–1869.

- [17] Hugh Salimbeni and Marc Deisenroth. “Doubly stochastic variational inference for deep Gaussian processes”. In: *Advances in neural information processing systems* 30 (2017).
- [18] Syusuke Sano et al. “Application of Bayesian Optimization for Pharmaceutical Product Development”. In: *Journal of Pharmaceutical Innovation* (2019), pp. 1–11. doi: 10.1007/s12247-019-09382-8.
- [19] Scott Sussex, Anastasiia Makarova, and Andreas Krause. “Model-based causal Bayesian optimization”. In: *arXiv preprint arXiv:2211.10257* (2022).
- [20] Michalis Titsias. “Variational learning of inducing variables in sparse Gaussian processes”. In: *Artificial intelligence and statistics*. PMLR. 2009, pp. 567–574.
- [21] Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. “Gaussian process regression networks”. In: *arXiv preprint arXiv:1110.4411* (2011).

# **Evaluating the Realism of Lateral Movement Modeling With and Without the Consideration of Influencing Factors**

*Nicole Neis*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
nicole.neis@student.kit.edu

## **Abstract**

The lateral offset between vehicles is a relevant factor for the range of vision of vehicle sensors and thus influences the output of an automated driving function relying on the sensor input. With simulations being an integral part of the automated driving function validation process, their ability to realistically reflect the lateral movement of vehicles is essential. A model addressing this task has been introduced in earlier work. How to assess the realism of the generated lateral movement is a question of its own. Earlier work bases the evaluation on comparing the distributions of selected features for the real and artificial lateral movement using the Jaccard index. When using this unit as a basis, the model generates not only qualitatively but also quantitatively realistic results and, in particular, outperforms other existing approaches. To gain more insights on the model's abilities and limitations, we explore the use of the Cramér-von Mises distance and the associated test as further means for evaluating the baseline model and, in particular, the enhanced version of the model considering influencing factors. The results show that the Cramér-von Mises distance and test are suitable additional means for comparing the feature and lateral position distributions, underlining the strengths of the model but also revealing its weaknesses. The

introduced *factor-based Cramér-von Mises distance* is in particular able to bring out the effect of considering influencing factors on the model results.

## 1 Introduction

The lateral movement of vehicles within their lane affects the object occlusions that occur, the range of vision of vehicle sensors, and ultimately the objects detected by downstream algorithms. The lateral position of vehicles is also a direct input to automated driving (AD) functions such as a cut-in detection function that uses this information to predict upcoming cut-ins of surrounding vehicles. With classical field tests alone being no longer feasible for the validation of AD functions [31], simulations have become an integral part of it. To ensure their representativeness with respect to reality, they need to be able to realistically model the lateral movement of vehicles. However, currently used tools such as microscopic traffic simulations [10, 16] and scenario-based tests that use the maneuver-based approach [24, 13, 15, 28] simplify or neglect the lateral movement of vehicles. For that reason, earlier work proposed a model addressing this task [20, 22, 23, 21]. How to assess the realism of the so generated lateral movement is a question on its own. The feature-based evaluation strategy presented and adopted in previous work [20, 22, 23, 21] has shown to be a good starting point. Other than the approaches used in comparable work, it is suitable to detect discrepancies in the temporal course of the lateral movement. The present work explores further strategies for model evaluation to gain additional insights on its performance with a particular focus on evaluating the model variants that consider factors influencing the lateral movement.

The remainder of this work is structured as follows: after the introduction given in the current section, related work is presented in Sec. 2. Sect. 3 describes the datasets used. Additional means of evaluation for the baseline model and when taking into account influencing factors are explored and discussed in Sect. 4. A conclusion and an outlook in Sec. 5 close the work.

## 2 Related Work

In previous work, Neis and Beyerer introduced and extended the two-level stochastic model for the lateral movement of vehicles during lane following [20, 25, 23, 22, 21]. It is based on the idea that at any point in time  $t$ , the lateral movement  $x$  can be split into a coarse part  $\kappa$  and a fine part  $\phi$ :

$$x(t) = \kappa(t) + \phi(t). \quad (2.1)$$

The coarse movement represents the systematic and position-dependent part of the lateral movement. The fine movement is the continuous residuum and is assumed to be stochastically independent from the coarse movement. In the following, the general structure of the model is explained. For more details, see the referenced literature.

A Markov model is used to model the coarse movement. It can be extended to a hidden Markov model (HMM) to take into account influencing factors on the lateral movement. The states of the Markov model, respectively hidden states of the HMM, are obtained by separating the lane into  $n_c$  segments of equal width. The (hidden) Markov model is calibrated by using the coarse movement observed in real-world data. It is obtained by rounding the full lateral movement to discrete positions. To compensate for the steps occurring in the output of the (hidden) Markov model, it is smoothed using a kernel  $g_s$  with mean zero and standard deviation  $s$  before adding the noise. The real fine movement is obtained by subtracting the smoothed real coarse movement from the real full movement, and modeled via a noise model. The idea is to describe how far the fine movement of the driver differs from white noise. This difference is reflected by a noise kernel used to convolve white noise to generate noise with the same characteristics as the fine movement of the driver. The results can be improved by parameterizing the noise kernel in every timestep [22] as an exponential function  $k(x) = \max(0, c \exp(-a|x|) + b)$  with three parameters. These are provided by a Markov model with resolution  $M_a \times M_b \times M_c$  calibrated with real data. Suitable meta parameters for the model are derived from the sensitivity analysis performed in [25] and the insights of [22].

The way in which the realism of the generated lateral movement can be shown is a challenge on its own. The strategy developed and applied in earlier work is

based on ten metrics that evaluate the geometric features of lateral offset profiles of constant duration  $T$ . Given a lateral offset profile  $X = [x_0, \dots, x_m] \in \mathbb{R}^{m+1}$  the metrics are the maximum of  $X$   $x_{\max}$ , the minimum of  $X$   $x_{\min}$ , the mean of  $X$   $\bar{x}$ , the standard deviation of  $X$   $\sigma$ , the median of  $X$   $x_{0.5}$ , the 25% and 75% percentile of  $X$   $x_{0.25}$  respectively  $x_{0.75}$ , the range of  $X$   $r$  defined as  $r := |x_{\max} - x_{\min}|$ , the mean difference between two consecutive values in  $X$  times ten  $\bar{x}_{\text{diff},10}$  and standard deviation of the difference between two consecutive values in  $X$  times ten  $\sigma_{\text{diff},10}$ . For each real lateral offset profile of duration  $T$ , an artificial one with the same duration is generated. Then, each of the metrics can be evaluated on each of the lateral offset profiles. The resulting distribution can be compared quantitatively using the so-called Jaccard index  $J_w[f_r, f_a]$  [5]. It measures the overlap of the distributions for a given metric for the *real* and the *artificial* data. As demonstrated in [21], this evaluation strategy can reveal discrepancies in the temporal course of the generated lateral movement. It is therefore superior to other approaches proposed in literature that evaluate the overall lateral offset or speed distribution or their moments only [6, 27]. For the performance of optimizations or condensed evaluations, in [21] the evaluation is further summarized into a fitness function that sums the minimum Jaccard index and the median of the remaining Jaccard indices. Special weight is given to the minimum Jaccard index, as previous work [25, 19] has shown that in particular the performance of the weakest metric is decisive for the overall performance of the model.

Alternative methods for modeling the lateral movement of vehicles within their lane are based on stochastic differential equations [6, 27]. A comparison of these models with the one of Neis and Beyerer shows the qualitative and quantitative superiority of the latter [19]. It creates realistic results on varying time scales and for different driver-specific datasets [21]. In addition, it outperforms other approaches in terms of calibration and computational efficiency [19]. The model as such is use case agnostic, but due to its modular structure it can be adopted based on an envisaged application. For example, one could decide to reduce model realism to increase runtime. In contrast to the other approaches, the model can take into account influencing factors whose relevance on the lateral movement has been shown in numerous studies [4, 29, 3, 8, 18, 14, 7].



Earlier work showed that considering a set of sample influencing factors for the two-level stochastic model can be advantageous to the realism over extended time periods [23]. In addition, taking the longitudinal speed into account can improve the model results [21]. However, a detailed analysis for further influencing factors is pending and lacks an evaluation strategy that takes these into account.

This paper aims to explore additional means to assess the realism of the generated lateral offset profiles for the baseline model, as well as models considering influencing factors.

### 3 Dataset

For the experiments performed within this study, data from three drivers collected on German highways are used. The recording vehicle was a Porsche Cayenne equipped as described in [12] with two additional LiDAR sensors, one on the left and one on the right side. The total amount of data collected on highways for each driver is given in Tab. 3.1. We further specify the amount of data usable for the model after removing lane changes and erroneous measurements.

The lateral position information is directly obtained from the vehicle's bus which provides the distances to the left and the right lane markings. These bus signals are already processed data that trace back to the images of the series camera. To be able to compare the lateral position on lanes of different widths, the lane width is normalized to one, and relative lateral positions are used. These range from  $-0.5$  (vehicle center on left lane marking) to  $0.5$  (vehicle center over right lane marking). The information on the influencing factors is obtained from the inertial measurement unit (longitudinal speed), the vehicle's bus (type and color of lane marking, position, speed, and class of surrounding vehicles), and the LiDAR data (relative position of neighboring vehicles).

The data recorded at frequencies ranging from 10 Hz (LiDAR) over 25 Hz (vehicle bus) to 100 Hz (inertial measurement unit) are synchronized and down-sampled to 5 Hz. It is assumed that below this threshold, vehicle motion is predominantly determined by vehicle dynamics and not by the human driver. Thus, down-sampling is considered permissible.

**Table 3.1:** Data available per driver.

	<i>Driver 1</i>	<i>Driver 2</i>	<i>Driver 3</i>
total duration of highway drives (in h)	9.4	6.9	6.1
usable distance of lane following (in km)	687	616	500
usable duration of lane following (in h)	8.1	6.5	5.4

## 4 Evaluation Extension

Within this section, additional means of model evaluation based on the Cramér-von Mises (CvM) distance and the Cramér-von Mises test are introduced [2]. The CvM distance is a distance measure which can be used to determine the distance between two given sample sets. The CvM test checks whether we can assume the same underlying distribution for them. In its original version, the CvM test works only for the one-dimensional case. However, Hanebeck and Klumpp [11] proposed an extension for the multivariate case which is used within this work. It compares the localized cumulative distributions of two sample sets calculated with various patch sizes.

Alternative approaches to the CvM test are the Kolmogorov-Smirnov test [26], which is, however, commonly considered less powerful [30] and the Anderson-Darling test [1]. The latter places more emphasis on the tails of the distribution [9], while the CvM test equally weights the full distribution. Thus, the latter is selected for this work.

In all experiments, we choose a resolution of  $n_c = 20$  for the (hidden) Markov model and a standard deviation of  $s = 3s$  for the smoothing kernel. The resolutions of the parameter Markov model for the noise model are chosen as  $M_a = 40$ ,  $M_b = 20$ , and  $M_c = 10$ . We evaluate the coarse model at a timestep of one second and interpolate it to be in line with the overall model timestep of  $\Delta\tau = 0.2s$  which is used for the noise model. To verify the null hypothesis that two samples trace back to the same distribution using the CvM test, we select a significance level of 5%.

## 4.1 Baseline Model Evaluation

Within this section, we assess additional evaluation strategies for the baseline model that could potentially extend the current approach explained in Sec. 2. We calculate the CvM distance and apply the CvM test to the metric distributions and the overall lateral offset distribution. The lateral offset distribution is used as an evaluation measure in other papers [6, 27]. Earlier work [21] explained the inferiority of this approach compared to the metric-based evaluation strategy as sole mean of evaluation. However, the interest of this work is to find out in what way it is useful as an additional tool. Due to the stochastic character of the model, the CvM distance and test result vary from one run to the other. Each model is therefore run ten times. We then indicate the mean and standard deviation for the CvM distances as well as the Jaccard indices, and the number of acceptances of the CvM test for these ten runs.

### 4.1.1 Evaluation of lateral position distribution

We start with evaluating the CvM distance and test result, as well as the Jaccard index for the real lateral position distribution and the one resulting from the model. To obtain the model's lateral offset distribution, for each real lane following maneuver present in the data we generate an artificial one with the same duration. For reference, we conduct the same comparison as for the real-world data and the artificial data with the two subsamples of a random split of the real lateral position data. Naturally, this results in a sample size being half the one of the full real data. To ensure comparability of the results, we thus also select half of the real datapoints and half of the model datapoints for their comparison. Moreover, we apply the multivariate CvM distance to compare the distribution of lateral position data on the space and time domain simultaneously. The patch size chosen for the space domain is varied between 0.01 and 0.1 (no unit as we use relative lateral positions), and the one for the time domain between 0.2 s and 5 s. With increasing duration  $T$ , the amount of lane following maneuvers within the real data extending up to that duration reduces, and the space-time distribution of lateral position data becomes sparse. For that reason, the time domain is restricted to  $T = 30$  s. This duration is also frequently used in other

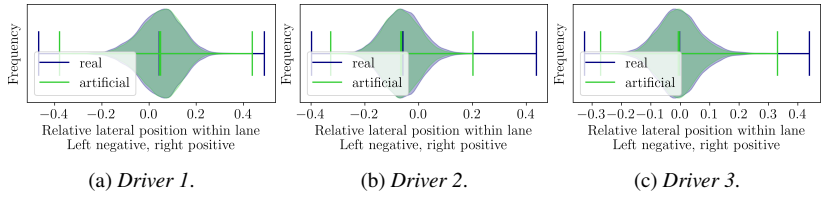
**Table 4.1:** Mean and standard deviation of CvM distances as well as Jaccard indices, and test results for ten runs when comparing distributions of lateral offset data. In the first case we compare the real data and the model data, in the second case two random splits of the real data. To ensure a consistent sample size, for comparing the real data and the model data random samples consisting of half of the data are used for comparison.

	<i>Driver 1</i>		<i>Driver 2</i>		<i>Driver 3</i>	
	data vs model	data vs data	data vs model	data vs data	data vs model	data vs data
acceptance rate	0/10	10/10	0/10	9/10	0/10	10/10
CvM distance	9.719 ± 5.100	0.070 ± 0.039	9.997 ± 6.641	0.184 ± 0.130	4.751 ± 1.935	0.158 ± 0.122
Jaccard index	0.927 ± 0.009	0.987 ± 0.002	0.923 ± 0.020	0.984 ± 0.003	0.931 ± 0.009	0.983 ± 0.002
multivariate CvM distance × 10 <sup>-4</sup>	7.593 ± 1.848	0.705 ± 0.882	4.330 ± 1.404	0.883 ± 0.861	8.112 ± 1.198	1.388 ± 1.796

studies as it is a good compromise between an extended simulation duration and the number of real-world data snippets available. The results are listed in Tab. 4.1. We illustrate the resulting lateral position distributions in Fig. 4.1.

When performing the evaluation based on the two real data samples, the null hypothesis is accepted in almost every case, the mean CvM distances are below 0.2 and the Jaccard indices above 0.98. For the comparison of the real-world data with the one generated by the model, however, the null hypothesis is rejected in every case and the CvM distances are significantly higher, even though the Jaccard indices are still above 0.92. When qualitatively comparing the illustrated lateral offset distributions, one can note slightly different shapes and a small shift of the mean values. For the multivariate case, high standard deviations with respect to the mean value can be observed for the comparison of two real-world data sets. When comparing the real-world data with the model data, the variations of the multivariate CvM distance with respect to the resulting mean value are smaller, but the mean value for the CvM distance is between half and one order of magnitude larger than when comparing the two real-world subsets.

Thus, even though the Jaccard indices reveal a high overlap of the lateral offset distributions observed in reality and generated by the model, due to the high sensitivity of the CvM test, the slight differences visible from the illustrations are sufficient to cause the rejection of the null hypothesis. According to the test, the model is thus not able to reproduce the overall distribution of lateral positions observed in reality and also the significantly larger CvM distances



**Figure 4.1:** Lateral offset distributions for the real data and resulting from the models for the three drivers. The vertical lines in the violins indicate the minimum, mean, and maximum value.

indicate a mismatch between the real-world lateral position distribution and the one generated by the model. The real-world data themselves, however, according to the CvM test result, are able to reproduce the distribution. Also when considering the multivariate case, the two subsets of real-world data lie way closer together than the real-world data and the model data. Thus, if aiming to reproduce the lateral offset distribution of the real-world data, there is space for improvement, not only regarding the overall lateral position but also regarding its evolution over time. If an exact reproduction of the real-world distributions is desirable, however, depends on the use case. One might be interested in generally increasing the variety of the simulations with respect to reality and to reflect in particular rarely appearing behavior that has not been observed in reality. In this case, a model featuring a higher entropy than the real-world data is required, yielding a different lateral offset distribution than the one observed in reality.

#### 4.1.2 Evaluation of metric distributions

Second, the CvM distance and test result are determined for the metric distributions for the three drivers. The chosen snippet length is set to  $T = 30$  s. The results for the three drivers are given in Tab. 4.2 to Tab. 4.4. For reference, we again give the results for comparing two random splits of the real data, and therefore again randomly select half of the model data and half of the real data for their comparison. The applied coloring ranges from green (better results,

**Table 4.2:** Number of acceptances of null hypothesis based on CvM test, mean and standard deviation of CvM distance, and mean and standard deviation of Jaccard indices for ten runs for *Driver 1* for the ten metrics when comparing a randomly selected half of the real data with a randomly selected half of the model data and when comparing two random splits of the real data.

	data vs. model			data vs. data		
	acceptance rate	Jaccard index	CvM distance	acceptance rate	Jaccard index	CvM distance
$\sigma_{\text{diff},10}$	0/10	0.440 ± 0.020	9.272 ± 1.017	10/10	0.903 ± 0.033	0.145 ± 0.130
$\bar{x}_{\text{diff},10}$	9/10	0.806 ± 0.023	0.307 ± 0.107	10/10	0.894 ± 0.026	0.116 ± 0.080
$r$	0/10	0.782 ± 0.034	1.046 ± 0.396	10/10	0.881 ± 0.027	0.177 ± 0.081
$x_{0.75}$	9/10	0.888 ± 0.043	0.182 ± 0.174	10/10	0.901 ± 0.019	0.135 ± 0.082
$x_{0.25}$	8/10	0.884 ± 0.030	0.316 ± 0.228	9/10	0.891 ± 0.021	0.189 ± 0.156
$x_{0.5}$	9/10	0.897 ± 0.036	0.194 ± 0.167	10/10	0.890 ± 0.027	0.169 ± 0.131
$\sigma$	9/10	0.868 ± 0.026	0.242 ± 0.201	9/10	0.882 ± 0.029	0.188 ± 0.131
$\bar{x}$	8/10	0.888 ± 0.039	0.254 ± 0.190	9/10	0.895 ± 0.024	0.165 ± 0.135
$x_{\min}$	10/10	0.886 ± 0.024	0.135 ± 0.087	9/10	0.896 ± 0.024	0.197 ± 0.151
$x_{\max}$	0/10	0.802 ± 0.027	0.957 ± 0.321	10/10	0.897 ± 0.024	0.109 ± 0.058

i.e., lower CvM distances/higher Jaccard indices) to yellow (worse results, i.e., higher CvM distances/lower Jaccard indices).

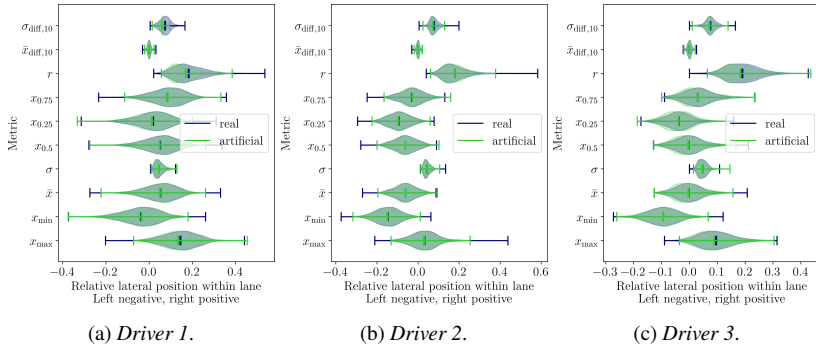
In case of the data split, the null hypothesis is accepted in eight to ten cases for all drivers and metrics with CvM distances smaller than 0.3. With a few exceptions, this is also the performance achieved for most model versus data comparisons. The exceptions are the metric  $\sigma_{\text{diff},10}$  for all drivers,  $r$  and  $x_{\min}$  for *Driver 1* and *Driver 3*, and  $\sigma$  for *Driver 3*. In these cases, the resulting CvM distances are significantly higher and the null hypothesis is accepted in at most half of the cases. Looking at the Jaccard indices and CvM distances together, it can be seen that there is a general correspondence between both. High CvM distances go along with low Jaccard indices, and vice versa. We further consider the illustration of the metric distribution given in Fig. 4.2. A correspondance can be seen between those cases for which the model data show not only a slightly different shape but a significantly shift with respect to the real data, in particular visible from the mean value, and the cases for which the CvM distance is high. The latter also goes along with a frequent rejection of the null hypothesis, relatively low Jaccard indices. The metric  $\sigma_{\text{diff},10}$  for *Driver 1* that shows the visually least agreement is also the one with the lowest Jaccard index and the highest CvM distance.

**Table 4.3:** Number of acceptances of null hypothesis based on CvM test, mean and standard deviation of CvM distance, and mean and standard deviation of Jaccard indices for ten runs for *Driver 2* for the ten metrics when comparing a randomly selected half of the real data with a randomly selected half of the model data and when comparing two random splits of the real data.

	data vs. model			data vs. data		
	acceptance rate	Jaccard index	CvM distance	acceptance rate	Jaccard index	CvM distance
$\sigma_{\text{diff},10}$	0/10	0.676 ± 0.047	1.098 ± 0.453	8/10	0.876 ± 0.046	0.226 ± 0.237
$\bar{x}_{\text{diff},10}$	10/10	0.884 ± 0.026	0.141 ± 0.054	8/10	0.875 ± 0.038	0.213 ± 0.207
$r$	9/10	0.849 ± 0.041	0.199 ± 0.143	9/10	0.879 ± 0.037	0.178 ± 0.141
$x_{0.75}$	10/10	0.855 ± 0.022	0.197 ± 0.090	9/10	0.880 ± 0.033	0.207 ± 0.191
$x_{0.25}$	10/10	0.882 ± 0.024	0.142 ± 0.061	10/10	0.886 ± 0.020	0.165 ± 0.097
$x_{0.5}$	10/10	0.877 ± 0.020	0.138 ± 0.055	9/10	0.890 ± 0.025	0.192 ± 0.149
$\sigma$	10/10	0.885 ± 0.020	0.128 ± 0.057	10/10	0.887 ± 0.029	0.150 ± 0.105
$\bar{x}$	10/10	0.865 ± 0.026	0.154 ± 0.058	9/10	0.890 ± 0.025	0.184 ± 0.145
$x_{\text{min}}$	10/10	0.875 ± 0.017	0.141 ± 0.065	10/10	0.879 ± 0.025	0.199 ± 0.092
$x_{\text{max}}$	9/10	0.840 ± 0.054	0.240 ± 0.132	9/10	0.870 ± 0.042	0.220 ± 0.168

**Table 4.4:** Number of acceptances of null hypothesis based on CvM test, mean and standard deviation of CvM distance, and mean and standard deviation of Jaccard indices for ten runs for *Driver 3* for the ten metrics when comparing a randomly selected half of the real data with a randomly selected half of the model data and when comparing two random splits of the real data.

	data vs. model			data vs. data		
	acceptance rate	Jaccard index	CvM distance	acceptance rate	Jaccard index	CvM distance
$\sigma_{\text{diff},10}$	4/10	0.771 ± 0.030	0.725 ± 0.372	10/10	0.885 ± 0.025	0.121 ± 0.085
$\bar{x}_{\text{diff},10}$	10/10	0.882 ± 0.016	0.157 ± 0.067	9/10	0.876 ± 0.054	0.134 ± 0.128
$r$	3/10	0.760 ± 0.045	0.901 ± 0.444	10/10	0.870 ± 0.028	0.174 ± 0.120
$x_{0.75}$	10/10	0.879 ± 0.039	0.114 ± 0.065	10/10	0.865 ± 0.024	0.198 ± 0.065
$x_{0.25}$	10/10	0.866 ± 0.040	0.150 ± 0.085	8/10	0.860 ± 0.036	0.234 ± 0.178
$x_{0.5}$	10/10	0.879 ± 0.036	0.128 ± 0.064	10/10	0.863 ± 0.029	0.227 ± 0.124
$\sigma$	3/10	0.757 ± 0.048	0.767 ± 0.469	10/10	0.879 ± 0.033	0.168 ± 0.133
$\bar{x}$	10/10	0.887 ± 0.044	0.116 ± 0.085	10/10	0.864 ± 0.022	0.211 ± 0.102
$x_{\text{min}}$	10/10	0.898 ± 0.014	0.080 ± 0.037	10/10	0.872 ± 0.034	0.180 ± 0.113
$x_{\text{max}}$	5/10	0.843 ± 0.037	0.451 ± 0.236	9/10	0.872 ± 0.040	0.185 ± 0.123



**Figure 4.2:** Metric distributions for the real data and resulting from the models for the three drivers. The vertical lines in the violins indicate the minimum, mean, and maximum value.

Consequently, the statement on the model’s performance for those metrics that already show a high Jaccard index can be strengthened to stating that the model is not only able to produce good overlap of the metric distributions, but also results in low CvM distances indicating the strong tendency that the same underlying distribution between the model data and the real data can be assumed. In particular, in these cases the results for the CvM distance and tests are as good as those obtained for two real data splits. Consequently, for almost all features, the model is able to reproduce the feature distribution observed in reality. For the real-world data, however, this applies to all metrics while outliers exist when comparing the model data with the real data. The relatively low Jaccard indices already hint at discrepancies in the distributions which are confirmed by the CvM distances and test results that further indicate that the differences are large enough to assume a different underlying distribution. Consequently, if aiming for better alignment between the model data and the real-world data, the metrics characterized by high mean CvM distances need to be improved.

In the experiments, the CvM distance and test results and the Jaccard index show a high correspondence. However, even though related, they measured different quantities. While the Jaccard index evaluates the overlap of two sample sets, the CvM distance and test evaluate their agreement. A high Jaccard index can thus



be seen as a relevant precondition for a low CvM distance and an acceptance of the null hypothesis, and the acceptance of it is a stronger statement than a high Jaccard index. Vice versa, the rejection of the null hypothesis as a consequence of a large CvM distance is a stronger indicator of a mismatch between the distributions than the value obtained for the Jaccard index. The Jaccard index, due to its fixed value range is easier to interpret than the CvM distance and allows inter-model comparisons. Therefore, we consider both strategies valuable for model evaluation.

## 4.2 Evaluation of Influencing Factors Integration

This section explores how the performance of a model taking into account influencing factor can be assessed using the CvM distance and test. We consider four different types of influencing factors, namely the lane type, traffic in the neighboring right lane, and the front vehicle, referred to as *lead vehicle*. Before presenting the evaluation results, each type of factor (e.g., lane) and the factor configurations considered consisting of the possible factor realizations (e.g., left/center/right lane) are shortly described in Sec. 4.2.1 to Sec. 4.2.3.

Four different evaluation strategies are investigated. Firstly, we give the results for evaluating the model results based on the Jaccard indices for the metric distributions. We again select a snippet length of  $T = 30$  s. For a condensed evaluation, the fitness function is calculated that sums the mean and median of the remaining nine Jaccard indices (see Sec. 2). Secondly, the metric distributions are compared using the CvM distance. Also for this case, we aggregate the results into a fitness function summing the maximum CvM distance and the median of the remaining nine CvM distances. Thirdly, we introduce an evaluation strategy that considers influencing factors. For this, again for each lane following maneuver in the data, we generate an artificial one with the same duration. Then, we derive the lateral position distributions for the real and the artificial data for each influencing factor representation considered for evaluation. Their accordance is measured using the CvM distance. If for example driving on the left lane is one factor realization considered within the evaluation, the lateral position distribution for all real data and all artificial data for which the vehicle was on the left lane is determined and the CvM distance calculated. After having

done so for all influencing factor realizations that are part of the evaluation, we build the weighted sum. Hereby, weighting of the CvM distances is applied based on the number of samples per corresponding factor realization. We refer to this weighted sum of CvM distances as *factor-based CvM distance*. The factor configuration considered for evaluation and the one used in the HMM can potentially differ. For a better comparison of the various model configurations, the factor configuration used for the evaluation is kept the same for each type of influencing factor. It is specified in the section on the corresponding type of influencing factor.

#### **4.2.1 Lane Type**

Regarding the lane type, seven configurations are analyzed. In addition to the factor realizations introduced below for these configurations, there is always an *other* class covering all remaining cases. In the first case, we differentiate between the left or center lane and the right lane, in the second, we summarize the center and right lane and consider the left lane separately, and in the third, we separate all three lane types. By applying this to regular lanes only (and summarizing construction site lanes under *other*) or applying it to construction sites also (e.g., lane type *left lane* applies to regular lanes and construction site lanes), this gives six configurations. In the last case, we differentiate between the three lane positions and whether it is a lane within a construction site or not. This is also the differentiation made for the factor-based evaluation.

#### **4.2.2 Traffic on Right**

Further we investigate the effect of considering passings of vehicles in the neighboring right lane as emission of the HMM. The motivation behind this is the observation of evasive maneuvers to the left during passings in the real data. Each passing can be separated into an approaching phase (twenty seconds until the front of the ego vehicle is reaching the rear of the vehicle to be passed), the passing phase itself, and the leaving phase (fifteen seconds until rear of ego vehicle reaches front of lead vehicle). We can further differentiate the class of vehicle passed (i.e., truck or passenger car) and consider trucks only,

passenger cars only, both while applying the same behavior to each class or both but treating the two classes separately. We request free-flow traffic conditions and thus a longitudinal speed of the passing vehicle of at least 40 km/h. If not restricting the longitudinal speed, the behavior is primarily determined by the long-lasting approaching and leaving phases during traffic jams for which the lateral movement can be assumed to be dominated by other factors, e.g., the building of an emergency lane. When considering also the longitudinal speed as an emission of the Markov model, the speed restriction applied here becomes obsolete; however, for the here performed analysis of the passing effects only, it is necessary. For evaluation, we also distinguish between the passing, approaching, and leaving phase, treat trucks and passenger cars separately, and request free-flow traffic conditions.

### **4.2.3 Lead Vehicle**

Next we analyze the effects of the lead vehicle. There are studies which observed that there are cases in which the lateral offset behavior of the lead vehicle is mirrored by the following vehicle [17]. We thus use the relative lateral position between the lead vehicle and the ego vehicle as emission. We distinguish an offset of less than 0.3 m, more than 0.3 m to the left or more than 0.3 m to the right. Besides the lead vehicle's characteristics as such, we can vary the time headway (thw) threshold up to which the lead vehicle is taken into account. We vary it in steps of 1 s between 1 s and 5 s. As in the case of consideration of the vehicle on the right, only free flow traffic conditions are considered, i.e., cases in which the ego vehicle drives with at least 40 km/h. Also, for the evaluation, we restrict the consideration of lead vehicles to free flow traffic conditions and distinguish lead vehicles with a thw of 1 s, 2 s, 3 s, 4 s, and 5 s as separate factor realizations.

### **4.2.4 Results**

The results for applying the three evaluation strategies introduced in Sec. 4.2 to models considering the various types of influencing factors described previously, are given in Tab. 4.5 to Tab. 4.7. For reference, we also provide the results

**Table 4.5:** Model results for baseline model and extensions considering the lane type as influencing factor for the three drivers using the presented evaluation approaches.

	<i>Driver 1</i>			<i>Driver 2</i>			<i>Driver 3</i>		
	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance
baseline	1.413 ±0.029	9.639 ±1.054	125.289 ±17.182	1.551 ±0.034	1.246 ±0.434	154.061 ±14.045	1.636 ±0.028	1.181 ±0.387	16.819 ±7.179
reg. left or center / reg. right / other	1.413 ±0.025	9.627 ±1.048	135.201 ±27.401	1.551 ±0.034	1.260 ±0.455	175.616 ±34.145	1.655 ±0.025	1.176 ±0.604	20.558 ±9.136
left or center / right / other	1.410 ±0.029	9.659 ±1.070	166.607 ±32.970	1.548 ±0.031	1.266 ±0.483	204.772 ±39.698	1.661 ±0.031	1.319 ±0.544	19.645 ±11.635
reg. left / reg. right or center / other	1.417 ±0.026	9.685 ±1.039	30.867 ±9.238	1.565 ±0.020	1.319 ±0.393	91.390 ±16.290	1.644 ±0.032	1.176 ±0.468	11.520 ±5.287
left / right or center / other	1.418 ±0.026	9.725 ±1.063	28.131 ±9.420	1.565 ±0.021	1.286 ±0.424	78.819 ±17.063	1.641 ±0.018	1.249 ±0.445	9.946 ±5.403
reg. left / reg. center / reg. right / other	1.419 ±0.027	9.710 ±1.063	29.125 ±7.306	1.570 ±0.025	1.296 ±0.402	22.354 ±6.203	1.651 ±0.025	1.194 ±0.532	11.610 ±4.587
left / center / right / other	1.418 ±0.026	9.749 ±1.076	24.286 ±6.213	1.569 ±0.021	1.285 ±0.436	15.880 ±5.359	1.657 ±0.025	1.336 ±0.517	12.136 ±6.123
reg. left / reg. center / reg. right / cs left / cs center / cs right / other	1.418 ±0.025	9.740 ±1.081	26.018 ±7.132	1.569 ±0.023	1.303 ±0.407	20.017 ±6.300	1.658 ±0.020	1.229 ±0.468	11.417 ±6.435

**Table 4.6:** Model results for baseline model and extensions considering the vehicle on the right lane as influencing factor for the three drivers using the presented evaluation approaches.

	<i>Driver 1</i>			<i>Driver 2</i>			<i>Driver 3</i>		
	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance
baseline	1.413 ±0.029	9.639 ±1.054	68.442 ±25.080	1.551 ±0.034	1.246 ±0.434	94.548 ±23.080	1.636 ±0.028	1.181 ±0.387	52.062 ±20.162
truck on right	1.409 ±0.031	9.629 ±1.052	37.703 ±15.552	1.546 ±0.032	1.239 ±0.440	58.016 ±16.610	1.639 ±0.032	1.100 ±0.431	25.806 ±18.062
car on right	1.405 ±0.032	9.649 ±1.036	36.199 ±12.539	1.552 ±0.030	1.252 ±0.407	43.896 ±11.805	1.639 ±0.026	1.340 ±0.477	44.210 ±27.648
any vehicle on right	1.404 ±0.032	9.651 ±1.017	14.564 ±5.632	1.546 ±0.029	1.250 ±0.433	19.282 ±6.867	1.638 ±0.023	1.138 ±0.436	14.175 ±12.173
truck on right or car on right	1.404 ±0.032	9.653 ±1.008	11.837 ±4.655	1.548 ±0.030	1.260 ±0.395	13.611 ±5.362	1.642 ±0.028	1.115 ±0.430	13.716 ±12.241

**Table 4.7:** Model results for baseline model and extensions considering the lead vehicle as influencing factor for the three drivers using the presented evaluation approaches.

	<i>Driver 1</i>			<i>Driver 2</i>			<i>Driver 3</i>		
	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance	fitness Jaccard indices	fitness CvM distance	factor-based CvM distance
baseline	1.413 ±0.029	9.639 ±1.054	21.214 ±5.389	1.551 ±0.034	1.246 ±0.434	15.556 ±4.469	1.636 ±0.028	1.181 ±0.387	7.252 ±1.531
lat. pos lead vehicle, thw <1 s	1.411 ±0.029	9.551 ±1.037	20.407 ±5.368	1.550 ±0.033	1.241 ±0.455	14.992 ±3.406	1.637 ±0.026	1.265 ±0.443	9.008 ±1.732
lat. pos lead vehicle, thw <2 s	1.408 ±0.027	9.643 ±1.069	12.213 ±3.448	1.552 ±0.033	1.235 ±0.460	11.798 ±3.018	1.643 ±0.025	1.393 ±0.653	7.657 ±2.557
lat. pos lead vehicle, thw <3 s	1.410 ±0.029	9.628 ±1.057	9.452 ±2.935	1.551 ±0.032	1.261 ±0.418	7.012 ±1.955	1.637 ±0.027	1.204 ±0.488	7.268 ±2.511
lat. pos lead vehicle, thw <4 s	1.408 ±0.026	9.672 ±1.071	9.552 ±2.485	1.549 ±0.032	1.251 ±0.454	6.210 ±1.917	1.640 ±0.032	1.281 ±0.745	7.029 ±2.455
lat. pos lead vehicle, thw <5 s	1.411 ±0.026	9.659 ±1.034	10.092 ±1.724	1.553 ±0.034	1.249 ±0.457	5.946 ±2.106	1.635 ±0.022	1.362 ±0.757	6.889 ±1.811

achieved for the baseline model, which does not consider influencing factors. When using one of the fitness functions, the evaluation is independent from influencing factors and thus the baseline value is the same across all types of influencing factors. For the factor-based CvM distance, it depends on the factors considered for evaluation described in the respective section, yielding a different baseline value for each influencing factor type.

Due to the stochastic nature of the model, again a slightly different fitness value respectively CvM distance results in each run. We therefore give the mean and standard deviation resulting from ten runs. Coloring is applied based on the mean value with yellow indicating worse and dark green better values.

Throughout all considered influencing factor configurations, if using the Jaccard indices for evaluation, if there is an improvement of the mean fitness value, it is in the range of the standard deviation of the fitness value for the baseline model which is two orders of magnitude smaller than the mean value. Also if the fitness value decreases, this is still within the range of the standard deviation around the baseline's mean value in almost all cases. When using the fitness function based on the CvM distance, the mean values of the model variants considering influencing factors all lie in the corridor spanned by the baseline standard deviation around the baseline mean value.

Thus, when evaluating the model performance based on the accordance of the feature distributions using the Jaccard index or the CvM distance, the models

considering influencing factors perform at most as good as a *lucky* run of the baseline model and no notable changes can be noted. Consequently, the overall distribution of the selected features such as the occurring maximum or mean lateral position stays the same, independent from the chosen influencing factor configuration. However, the metric-based evaluation does not make any statement on whether the characteristic of these features is correct with respect to the present influencing factor. For example, the model might deliver the real-world behavior for the left lane on the right lane and vice versa. When using the metric-based evaluation strategy, the performance of the model can still be high if the feature distribution of the real-world data is reflected, even though the lane-specific behavior is swapped and there is a clear mismatch between model and reality.

When using the factor-based CvM distance for evaluation, the standard deviations are large with respect to the mean values. Nevertheless, notable decreases can be observed for several model variants considering influencing factors with the sum of standard deviation and mean value still being significantly smaller than the mean value minus the standard deviation for the baseline model.

For the lane configurations, significant improvements are achieved if treating the left lane separately from the right and center lane. For the vehicle on the right lane we get the best results if taking into account trucks and cars, regardless of whether they are handled separately or together. Regarding the consideration of the lead vehicle, the results for *Driver 1* and *Driver 2* indicate that at least the vehicles up to a time headway of three seconds should be taken into account. For *Driver 3*, however, no significant improvements over the baseline value can be noted when considering the lead vehicle.

Thus, the factor-based CvM distance is able to reveal the effect of considering influencing factors on the model results and the configuration working best for each type of factor can be identified based on them. If choosing a suitable configuration of influencing factors, the CvM distance significantly decreases, thus, the factor-specific lateral offset distributions of the model fit significantly better to those of the real world data. Due to the factor-dependency of this evaluation strategy, however, the results are always bound to the chosen set of evaluation factors and the comparison of model instances considering different types of influencing factors is hampered. For investigating the relevance of

different types of factors using the factor-based CvM, they need to be analyzed based on a common set of evaluation factors. In general, the evaluation factors and potentially also their weighting need to be selected carefully based on the use case. For example, if it requires accurate behavior with respect to regular lanes only and the behavior within construction sites is of no relevance, the latter should be excluded from the factors used for evaluation and only the regular lanes should be considered.

As every approach based on the lateral position distribution, also the factor-based CvM distance is not able to reveal discrepancies in the temporal course of the lateral offset profiles. Therefore, it is considered not suitable as single mean for evaluation. It is recommended to combine it with a feature-based approach. While the latter can be used to ensure the general realism of the lateral movement's temporal course reflected within the features, the factor-based CvM can be used to track the improvement brought about by a certain set of influencing factors based on a certain set of evaluation factors relevant for the use case.

## **5 Conclusion and Outlook**

Within this paper, potential enhancements to the currently used approach to evaluate the realism of lateral movement modeling comparing the accordance of lateral movement features using the Jaccard index are analyzed. For this, the CvM distance and test are analyzed. The CvM distance is a measure for the difference of two sample sets. The CvM test checks whether the same underlying distribution can be assumed.

In the first case, the evaluation of the baseline model which does not take into account influencing factors is considered. The CvM distance and test are applied to the metric and lateral position distributions. Depending on the driver, for six to nine metrics, the distributions for the real data and the model data do not only show a great overlap (as indicated by the Jaccard indices) but it can be assumed that they follow the same distribution. For the remaining metrics and also for the lateral position distribution, however, the results hint at a mismatch between the underlying distribution for the real data and the model data, even though the Jaccard indices might still be high. Also the multivariate CvM distance on

the space and time domain is significantly larger when comparing the model data with the real-world data than for the two real-world data subsets. Thus, if aiming for a close reflection of real-world behavior, potential for improvement is indicated. In general, the CvM distance and test allow stronger statements on the model performance, in terms of strengths and weaknesses, than the use of the Jaccard index. The latter, however, is more intuitive to interpret and allows for a simple comparison of metric performance, e.g., across different models. Therefore, both strategies in combination are considered valuable.

For the model with influencing factors it turned out that the feature-based strategies evaluating the metric distributions using the Jaccard index or the CvM distance are not able to reveal the effect of integrating influencing factors. They instead assess the general realism of the temporal course of the generated lateral movement which does not change notably. If, however, switching to a different approach evaluating the lateral position distribution under various influencing factors using the factor-based CvM distance, improvements in the model results become visible and the best performing set of given influencing factors can be identified. The results, however, depend on the factor configuration used for evaluation which has to be made carefully and in agreement with the use case. Being a distribution-based approach, the factor-based CvM distance cannot detect discrepancies in the temporal course. It is therefore recommended to combine the factor-based CvM distance with a metric-based evaluation.

With the fixation of the use case, not only the factor-based CvM distance can be refined but more strategies for evaluation might arise. E.g., one could evaluate the sensing results for different levels of traffic congestion and compare the outcome obtained for the real world and from the simulation. Furthermore, also the multivariate CvM distance might be a suitable mean for factor-based evaluation, allowing to compare the generated lateral movement jointly in the space and factor domain for influencing factors having a continuous representation such as the longitudinal speed.



## References

- [1] Theodore. W. Anderson and Donald A. Darling. “Asymptotic theory of certain ”goodness of fit” criteria based on stochastic processes”. In: *The Annals of Mathematical Statistics* 23.2 (1952), pp. 193–212. DOI: 10.1214/aoms/1177729437.
- [2] Dennis D. Boos. “Minimum distance estimators for location and goodness of fit”. In: *Journal of the American Statistical Association* 76.375 (1981), pp. 663–670. ISSN: 01621459, 1537274X. JSTOR: 2287527.
- [3] Alessandro Calvi. “Does Roadside Vegetation Affect Driving Performance?: Driving Simulator Study on the Effects of Trees on Drivers’ Speed and Lateral Position”. In: *Transportation Research Record* 2518.1 (Jan. 2015), pp. 1–8. ISSN: 0361-1981, 2169-4052. DOI: 10.3141/2518-01.
- [4] Harry W Case et al. “Effect of a Roadside Structure on Lateral Placement of Motor Vehicles”. In: *Proceedings of the Thirty-Second Annual Meeting of the Highway Research Board*. Vol. 32. Washington, D.C.: Highway Research Board, Jan. 1953, pp. 364–370.
- [5] Luciano da F. Costa. *Further Generalizations of the Jaccard Index*. Nov. 2021. eprint: 2110.09619 (cs).
- [6] Rafael Delpiano. “Understanding the Lateral Dimension of Traffic: Measuring and Modeling Lane Discipline”. In: *Transportation Research Record* 2675.12 (Aug. 2021), pp. 1030–1042. ISSN: 0361-1981, 2169-4052. DOI: 10.1177/036119812111031884.
- [7] Rafael Delpiano, Juan C. Herrera Maldonado, and Juan E. Coeymans Avaria. “Characteristics of lateral vehicle interaction”. In: *Transportmetrica A: Transport Science* 11.7 (Aug. 2015), pp. 636–647. ISSN: 2324-9935, 2324-9943. DOI: 10.1080/23249935.2015.1059377.
- [8] Chris Dijksterhuis, Karel A. Brookhuis, and Dick De Waard. “Effects of steering demand on lane keeping behaviour, self-reports, and physiology. A simulator study”. In: *Accident Analysis & Prevention* 43.3 (May 2011), pp. 1074–1081. ISSN: 00014575. DOI: 10.1016/j.aap.2010.12.014.

- [9] Patrick J. Farrell and Katrina Rogers-Stewart. “Comprehensive study of tests for normality and symmetry: extending the Spiegelhalter test”. In: *Journal of Statistical Computation and Simulation* 76.9 (Sept. 2006), pp. 803–816. ISSN: 0094-9655, 1563-5163. DOI: 10.1080/10629360500109023.
- [10] Martin Fellendorf and Peter Vortisch. “Microscopic Traffic Flow Simulator VISSIM”. In: *Fundamentals of Traffic Simulation*. Ed. by Jaume Barceló. Vol. 145. New York, NY: Springer New York, 2010, pp. 63–93. ISBN: 978-1-4419-6141-9. DOI: 10.1007/978-1-4419-6142-6\_2.
- [11] Uwe D. Hanebeck and Vesa Klumpp. “Localized Cumulative Distributions and a multivariate generalization of the Cramér-von Mises distance”. In: *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. Seoul: IEEE, Aug. 2008, pp. 33–39. ISBN: 978-1-4244-2143-5. DOI: 10.1109/MFI.2008.4648104. (Visited on 10/25/2024).
- [12] Johann Haselberger et al. “JUPITER – ROS based Vehicle Platform for Autonomous Driving Research”. In: *2022 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. Abu Dhabi, United Arab Emirates: IEEE, Nov. 2022, pp. 1–8. ISBN: 978-1-66548-923-2. DOI: 10.1109/ROSE56499.2022.9977434.
- [13] Aaron Heinz, Wolfram Remlinger, and Johann Schweiger. “Track- / Scenario-based Trajectory Generation for Testing Automated Driving Functions”. In: *8. Tagung Fahrerassistenz*. München, 2017.
- [14] Hanneke Hooft van Huysduynen, Jacques Terken, and Berry Eggen. “The relation between self-reported driving style and driving behaviour. A simulator study”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 56 (July 2018), pp. 245–255. ISSN: 13698478. DOI: 10.1016/j.trf.2018.04.017.
- [15] Nicola Kolb et al. “Automatic Evaluation of Automatically Derived Semantic Scenario Instance Descriptions”. In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. Macau, China: IEEE, Oct. 2022, pp. 1565–1571. ISBN: 978-1-66546-880-0. DOI: 10.1109/ITSC55140.2022.9922013.

- [16] Daniel Krajzewicz. “Traffic Simulation with SUMO – Simulation of Urban Mobility”. In: *Fundamentals of Traffic Simulation*. Ed. by Jaume Barceló. Vol. 145. New York, NY: Springer New York, 2010, pp. 269–293. ISBN: 978-1-4419-6141-9 978-1-4419-6142-6. DOI: 10.1007/978-1-4419-6142-6\_7.
- [17] Daofei Li and Ao Liu. “Personalized highway pilot assist considering leading vehicle’s lateral behaviors”. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* (Feb. 2022). ISSN: 0954-4070, 2041-2991. DOI: 10.1177/\linebreak 09544070221081190.
- [18] Shuo Liu, Junhua Wang, and Ting Fu. “Effects of Lane Width, Lane Position and Edge Shoulder Width on Driving Behavior in Underground Urban Expressways: A Driving Simulator Study”. In: *IJERPH* 13.10 (Oct. 2016). ISSN: 1660-4601. DOI: 10.3390/ijerph13101010.
- [19] N. Neis and J. Beyerer. “Comparison of Vehicle Lateral Movement Models in the Context of Automated Driving Function Validation”. In: In Review.
- [20] Nicole Neis and Jürgen Beyerer. “A Two-Level Stochastic Model for the Lateral Movement of Vehicles Within Their Lane Under Homogeneous Traffic Conditions”. In: *2023 IEEE Intelligent Transportation Systems Conference (ITSC)*. Bilbao, Spain, Sept. 2023.
- [21] Nicole Neis and Jürgen Beyerer. “Efficiently Modeling Lateral Vehicle Movement Including its Temporal Interrelations Using a Two-Level Stochastic Model”. In: *IEEE Open Journal of Intelligent Transportation Systems* 5 (2024), pp. 566–580. DOI: 10.1109/OJITS.2024.3435078.
- [22] Nicole Neis and Jürgen Beyerer. “Improved Modelling of the Position-Independent, Fine-Scale Lateral Movement of Vehicles Within Their Lane”. In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. In press. Edmonton, Canada, Sept. 2024.
- [23] Nicole Neis and Jürgen Beyerer. “Improving the Modelling of the Lateral Driving Behavior of Vehicles Within Their Lane Over Extended Time Periods”. In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. In press. Edmonton, Canada, Sept. 2024.

- [24] Nicole Neis and Jürgen Beyerer. “Literature Review on Maneuver-Based Scenario Description for Automated Driving Simulations”. In: *2023 IEEE Intelligent Vehicles Symposium (IV)*. Anchorage, AK, USA, June 2023.
- [25] Nicole Neis and Jürgen Beyerer. “Sensitivity Analysis and Extended Evaluation of the Two-Level Stochastic Model for the Lateral Movement of Vehicles Within Their Lane”. In: *2024 International Conference on Automation, Robotics and Applications (ICARA)*. Athens, Greece, Feb. 2024.
- [26] William H. Press and Saul A. Teukolsky. “Kolmogorov-Smirnov Test for Two-Dimensional Data”. In: *Computers in Physics 2.4* (July 1988), pp. 74–77. ISSN: 0894-1866. DOI: 10.1063/1.4822753.
- [27] Hongsheng Qi, Yuyan Ying, and Jiahao Zhang. “Stochastic lateral noise and movement by Brownian differential models”. In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. Aachen, Germany: IEEE, June 2022, pp. 98–103. ISBN: 978-1-66548-821-1. DOI: 10.1109/IV51971.2022.9827388.
- [28] Christoph Stadler et al. “A Credibility Assessment Approach for Scenario-Based Virtual Testing of Automated Driving Functions”. In: *IEEE Open J. Intell. Transp. Syst.* 3 (2022), pp. 45–60. ISSN: 2687-7813. DOI: 10.1109/OJITS.2022.3140493. (Visited on 04/03/2024).
- [29] A Taragin and H. G Eckhardt. “Effect of Shoulders on Speed and Lateral Placement of Motor Vehicles”. In: *Proceedings of the Thirty-Second Annual Meeting of the Highway Research Board*. Vol. 32. Washington, D.C.: Highway Research Board, Jan. 1953, pp. 371–382.
- [30] Alison Telford et al. “Properties and approximate  $p$ -value calculation of the Cramer test”. In: *Journal of Statistical Computation and Simulation* 90.11 (July 2020), pp. 1965–1981. ISSN: 0094-9655, 1563-5163. DOI: 10.1080/00949655.2020.1754820.
- [31] Walther Wachenfeld and Hermann Winner. “Die Freigabe des autonomen Fahrens”. In: *Autonomes Fahren*. Ed. by Markus Maurer et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 439–464. ISBN: 978-3-662-45853-2. DOI: 10.1007/978-3-662-45854-9\_21.

# **Towards Quantifying Simulated Image Sensor Data: A Survey and Discussion on GAN Evaluation Metrics**

*Anne Sielemann*

Fraunhofer Institute of Optronics, System Technologies  
and Image Exploitation IOSB  
Department Systems for Measurement, Control and Diagnosis (MRD)  
anne.sielemann@iosb.fraunhofer.de

## **Abstract**

Simulations are capable of solving many of today's problems with collecting real-world data for training and testing machine learning (ML) approaches for mobility applications. However, to effectively utilize synthetic data, it must be ensured that they possess the necessary quality, meaning they are "similar enough" to real-world data and include all characteristics that ML approaches require to learn task-relevant features. As the quantitative quality assessment of image data is a difficult task, the quality of simulated images is in practice mostly determined through cross-dataset tests or performance comparisons after the addition of synthetic data. This has the disadvantage that comparable annotated real-world data are still required, which can only be obtained with significant effort or are rarely available in some domains. In recent years, research in the field of developing metrics to quantify the image quality produced by generative neural networks has made notable progress. But in general, these metrics are not trivially transferable to simulated images, as the desired metric properties diverge between Generative Adversarial Networks (GANs) and simulations. Therefore, this work analyses the differences in requirements and discusses the suitability of established GAN performance metrics for quantifying simulation-based synthetic image sensor data on a theoretical basis. Thereby, a survey on existing GAN evaluation metrics is included.

# 1 Introduction

Simulators possess great potential for generating synthetic sensor data to train and test applications in the field of automated mobility applications: preventing privacy issues, generating highly accurate labels, the ability for systematic updates, targeted testing of predefined scenarios, especially extreme situations, and moreover, enabling a less expensive and time-consuming data generation process. This is also reflected in statistics: according to Li et al. [22], ‘‘over 50 % of methods published in the domain of autonomous driving between 2022 and 2023 were either trained or tested in simulation environments’’. Furthermore, the authors recognized an increase in released simulators (commercial as well as open-source) in the past years. Currently, there is a variety of simulators available, such as commercial systems like NVIDIA DRIVE Sim<sup>1</sup>, Hexagon VTD<sup>2</sup>, IPG CarMaker<sup>3</sup>, or dSPACE ASM<sup>4</sup> / SensorSim<sup>5</sup> (primarily for road vehicles). Also, open-source simulators are widely used, e.g., for the simulation of road vehicles as feasible with CARLA<sup>6</sup> and LG SVL<sup>7</sup>.

Nevertheless, the quantitative evaluation of synthetic image data still poses a challenge. Commonly, published simulated synthetic data are evaluated by measuring task-specific performance metrics in a cross-dataset evaluation [18, 29, 34, 35], i.e., comparing the performance of a network trained on a real-world dataset and a network trained on the simulated synthetic dataset by validating them on another real-world dataset or on the real-world dataset’s test set. Alternatively, some authors [30, 20] determine the task-related suitability of their simulated synthetic datasets by comparing achieved task-specific performance metrics when training only on real-world data versus training on (sometimes a

---

<sup>1</sup> <https://www.nvidia.com/de-de/self-driving-cars/simulation/>

<sup>2</sup> <https://hexagon.com/en/products/virtual-test-drive>

<sup>3</sup> <https://www.ipg-automotive.com/en/products-solutions/software/carmaker>

<sup>4</sup> [https://www.dspace.com/de/gmb/home/products/sw/automotive\\_simulation\\_models.cfm](https://www.dspace.com/de/gmb/home/products/sw/automotive_simulation_models.cfm)

<sup>5</sup> <https://www.dspace.com/en/inc/home/news/dspace-release-2019-a/sensorsim-1-1.cfm>

<sup>6</sup> <https://carla.org/>

<sup>7</sup> <https://www.svl Simulator.com/>

reduced amount of) real-world data in addition to their synthetic dataset. These procedures have a considerable disadvantage: annotated real-world data from the application domain are required for comparison. These can only be obtained with significant (labeling) effort, since, e.g., expert knowledge is required or the data are subject to strict privacy guidelines or relate to rare or critical scenarios, complicating the acquisition of large quantities in the first place. This may result in, on the one hand, relying on poorly suited real-world data from a domain other than the application domain, which in turn skews the evaluation results (a possible cause for effects noted in [34]), or, on the other hand, makes it impossible to evaluate the synthetic data properly. Therefore, it would be desirable to have a quantitative metric that requires only unlabeled real-world data for comparison.

While there is still little research in the area of simulated synthetic data, generative neural networks, especially *Generative Adversarial Networks (GANs)* [11], have been studied for suitable metrics for almost a decade and regular publications have emerged on this topic. However, the proposed metrics cannot be easily transferred to simulated synthetic images, as goals and requirements for metric properties differ between GANs and simulations. For this reason, it is sensible to first familiarize with the differences between the generation of synthetic data using GANs versus simulations (cf. Sec. 2), to analyze the differing desired metric properties (cf. Sec. 2.1 and Sec. 2.2), and to define criteria to conduct the metric comparison (cf. Sec. 2.3) before examining the existing GAN evaluation metrics in detail (cf. Sec. 3) and discussing their suitability for evaluating synthetic simulated image sensor data (cf. Sec. 4).

## 2 Desired Metric Properties

One of the main differences between applying GAN or simulation approaches for image synthesis is the means of establishing the underlying statistical distribution: in the case of GANs, the distribution is estimated by learning from real-world examples. Thus, their strength is the approximation of real-world data distributions, for which large amounts of training data are needed. This requires a high quality of the training data, as they should contain all important data dimensions, as well as include parameter fidelity of the desired target

distribution. In contrast, the underlying statistical distribution of simulation approaches is modeled and defined manually (and possibly in a data-driven way). This leads to better controllability both in terms of generation methods and quality/performance goals, and offers the possibility to cover areas of the abstract data space that might be difficult or laborious to include in its entirety by real-world data. However, manually modeling the distribution(s) requires expert knowledge or at least domain knowledge and is a difficult duty. These differences result in different desires and requirements for an optimal metric.

## 2.1 GANs

Borji conducted two extensive surveys [3, 4], where he compared different GAN performance metrics and worked out the desired metric properties. According to his work [3], an optimal GAN performance metric should consider the following aspects:

**High fidelity:** The generated data should look as realistic as possible, making it difficult to distinguish from real-world data. In the context of GANs, this also means that classes from the training data should not be mixed up in the result, but should be clearly distinguishable.

**Diverse samples:** GAN-generated samples shall be diverse, i.e., cover the learned feature space, both inter and intra classes. In particular, the potential problem of a *mode collapse* (which refers to the restriction to playback a few realistic samples, in the worst case memorized from the training data) should be excluded.

**Disentangled latent space and space continuity:** A disentangled, continuous latent space allows the samples to be controlled and to be specifically modified. GANs with this property exhibit better usability in practice.

**Well-defined bounds:** For good comparability, it is advantageous if the metric's boundaries are well-defined.

**Invariant to image distortions and transformations:** An ideal metric should be invariant to distortions or transformations of the image content that do not change the semantics.



**Aligned with human perceptual judgment:** Ideally, human judgment and the ranking by the metric should match.

**Sample efficiency:** It is desirable that a metric can perform a meaningful quality assessment even for small sets of samples.

**Low computational complexity:** A metric that requires less computational time is preferred over one that requires much computational time.

## 2.2 Simulations

None of today’s widely used and established GAN evaluation metrics is capable of fulfilling all desired aspects [3]. This makes it all the more important when striving for quantifying simulated image sensor data quality to identify those metric properties that are decisive for this purpose. With the experience of previous work on the generation of synthetic sensor simulated data [34, 35], the following characteristics have been identified as desirable for a potential metric:

**High fidelity:** The generated data should look as realistic as possible, making it difficult to distinguish from real-world data. Since 3D models are used in simulations, the depicted objects are typically not mixed from several classes. Due to manual modulation, there is a greater risk that objects have been generated over-simplified and that therefore, e.g., important task-specific features are missing, which can harm the discriminability.

**Diverse samples:** For simulated data, the class distribution can be easily altered, while it is difficult to manually model the variation within a class realistically. Therefore, it is advisable to prioritize a metric that measures intra-class variance. However, incorporating inter-class variance is not disadvantageous.

**Well-defined bounds:** Like for GAN metrics (cf. Sec. 2.1).

**Aligned with human perceptual judgment:** Like for GAN metrics (cf. Sec. 2.1).

**Sample efficiency:** Like for GAN metrics (cf. Sec. 2.1).

**Low computational complexity:** Like for GAN metrics (cf. Sec. 2.1).

In contrast to GANs, a disentangled latent space is not included in the desired metric properties for assessing simulated image data since simulations do not

strictly possess a latent space but usually provide a well-controllable parameter space. Furthermore, we have refrained from listing distortion and transformation invariance among the desired metric properties because the definition of this aspect leaves considerable room for ambiguity and interpretation concerning which transformations are tolerated and how the “semantics” are defined that are to be preserved. To take up Borji’s example [3]: when comparing two single images, it is advantageous if a metric is not oversensitive to a slight pixel shift but when the data generation source itself underlies a systematic distortion or transformation shift (like, e.g., the transformation of RGB to grayscale images), it could be desirable to be considered by a metric.

### 2.3 Implemented Comparison of Metric Properties

To identify suitable metrics for quantifying the quality of simulated image sensor data, all considered metrics should be investigated regarding the desired metric properties for simulation approaches identified in the previous section. Without conducting comprehensive experiments, no comprehensive ranking can be set up. However, to implement the comparison as objectively as possible, the degree of fulfillment of the individual metric properties is categorized into *not measured*, *low*, *moderate*, and *high* (following the example of Borji [3]) based on the criteria defined in Tab. 2.1.

However, there are three deviations from the desired metric properties for assessing simulated image sensor data: since the fidelity is hard to evaluate, the well-assessable discriminability is rated instead. A poor discriminability measurement is an indicator of a poor fidelity rating since the fidelity is not sufficient with indistinguishable objects. However, one should be aware that a good ability to measure discriminability does not automatically imply recognizing high fidelity in general. The fidelity may only be sufficient for task-relevant features.

To avoid misleading oversimplification, no comparison of computational complexity/time effort is provided, as it depends on many factors like the used general hardware, the application-specific hardware, implementation details, and especially the sample efficiency. Furthermore, the effort required in advance is higher if labeled data is presumed. It is thus evident that the assessment of

**Table 2.1:** Defined criteria to categorize the investigated GAN evaluation metrics regarding the desired metric properties for quantifying the quality of simulated image sensor data.

	not measured ○○○	low ●○○	moderate ●●○	high ●●●
<b>Discriminability</b>	Metric does not take discriminability into account	Metric may implicitly consider discriminability (not certain), e.g., via discriminator	Metric assesses the discriminability but can be fooled by confident misclassifications	Metric assesses the discriminability, e.g., by considering classification results or feature embeddings
<b>Diversity</b>	Metric does not take diversity into account	Metric may implicitly consider diversity (not certain), e.g., via discriminator	Metric can detect either inter- or intra-class diversity	Metric can detect intra- and inter-class diversity
<b>Detect Overfitting</b>	The metric is not capable of detecting overfitting and rewards it	The metric is not capable of detecting overfitting	—	The metric is aware of overfitting and penalizes or notifies it
<b>Perceptual Judgment</b>	—	Literature reports a bad alignment to perceptual judgment	Literature reports a medium alignment of perceptual judgement	Literature reports a good alignment to perceptual judgment
<b>Sample Efficiency</b>	—	Applicable to 1 – 5 000 images or more	Applicable to 5 000 – 50 000 images or more	More than 50 000 images (typically ~ 200 000) needed

computational efforts (and related time efforts to establish data) cannot meaningfully be represented in one or a few simple, comparable scales. Consequently, it was deemed that an overseeable classification of computational effort would be extremely difficult to establish formally, while still likely failing grossly to represent the complexity of the considerations needed to select a suitable metric for a particular use case.

The ability to detect overfitting is listed as a separate item. However, in the defined desirable metric properties it is included in the requirement of diverse samples. Since some metrics were designed to be able to assess the diversity but

can be fooled by simple replay of training samples we refrained from assessing both in one criterion.

### 3 Quantitative GAN Evaluation Metrics

Since this work is explicitly focused on quantitative evaluation metrics for simulated synthetic image data, this section only presents a survey of existing quantitative evaluation metrics for GANs. Qualitative evaluation metrics are not considered in this work ([3, 4] give an overview of this topic). The following subsections will discuss the metrics in detail, an overview of all metrics is provided in Tab. 4.1.

Concerning the notation, it should be noted that  $\pi(\cdot)$  is used instead of  $p(\cdot)$  to denote an output distribution of a deep neural network, so as not to give the impression that it is a mathematical correct probability distribution. Works like [12] showed that some neural network architectures are calibrated imprecisely and therefore, the resulting confidences should not be interpreted as true probability distributions.

#### 3.1 Pixel-Level Metrics

This class of metrics operates on pixel-level without incorporating the image content. In practice, content variant metrics are predominantly used. However, for the sake of completeness, the best-known pixel-level metrics are briefly introduced by this section.

##### 3.1.1 Mean Squared Error and Peak Signal-to-Noise Ratio

The *Mean Squared Error (MSE)* [44] calculates the average of pixel-wise differences, i.e., for two images  $\mathbf{x}_1$  and  $\mathbf{x}_2$  the MSE is defined as

$$\text{MSE}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_{1i} - \mathbf{x}_{2i})^2. \quad (3.1)$$

The *Peak Signal-to-Noise Ratio (PSNR)* [44] is based on the MSE while considering the dynamic range  $L$  (255 for an 8-bit grayscale image), so

$$\text{PSNR}(\mathbf{x}_1, \mathbf{x}_2) = 10 \log_{10} \frac{L^2}{\text{MSE}(\mathbf{x}_1, \mathbf{x}_2)}. \quad (3.2)$$

For the general comparison of two (unpaired) datasets, both metrics are rather unsuitable, as they perform pixel-wise comparisons. But these scores might be useful if a reference image is available, e.g., for training conditional GANs using paired data [3].

### 3.1.2 Structural Similarity Index Measure

The *Structural Similarity Index Measure (SSIM)* [46, 37] compares two images (or image patches)  $\mathbf{x}_1$  and  $\mathbf{x}_2$  with regard to their luminance ( $I$ ), contrast ( $C$ ), and structure ( $S$ ). For an image  $\mathbf{x}$ , its mean intensity is defined by  $\mu_{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ , the standard deviation by  $\sigma_{\mathbf{x}} = \left( \left( \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \mu_{\mathbf{x}})^2 \right) \right)^{\frac{1}{2}}$ , and the sample correlation coefficient by  $\sigma_{\mathbf{x}_1 \mathbf{x}_2} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_{1i} - \mu_{\mathbf{x}_1})(\mathbf{x}_{2i} - \mu_{\mathbf{x}_2})$ . Thus,  $I$ ,  $C$ , and  $S$  are defined as

$$I(\mathbf{x}_1, \mathbf{x}_2) = \frac{2\mu_{\mathbf{x}_1}\mu_{\mathbf{x}_2} + C_1}{\mu_{\mathbf{x}_1}^2 + \mu_{\mathbf{x}_2}^2 + C_1}, \quad (3.3)$$

$$C(\mathbf{x}_1, \mathbf{x}_2) = \frac{2\sigma_{\mathbf{x}_1}\sigma_{\mathbf{x}_2} + C_2}{\sigma_{\mathbf{x}_1}^2 + \sigma_{\mathbf{x}_2}^2 + C_2}, \quad (3.4)$$

$$S(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sigma_{\mathbf{x}_1 \mathbf{x}_2} + C_3}{\sigma_{\mathbf{x}_1}\sigma_{\mathbf{x}_2} + C_3}, \quad (3.5)$$

where  $C_1$ ,  $C_2$ , and  $C_3$  are small constants added for numerical stability. The authors choose  $C_1 = (K_1 L)^2$  with a small constant  $K_1 \ll 1$  and  $L$  being the dynamic range of pixel values (255 for an 8-bit grayscale image).  $C_2$  is defined analogously with a small constant  $K_2$ , while  $C_3 = \frac{C_2}{2}$ . These three equations are combined to form the SSIM score defined as

$$\text{SSIM}(\mathbf{x}_1, \mathbf{x}_2) = I(\mathbf{x}_1, \mathbf{x}_2)^\alpha C(\mathbf{x}_1, \mathbf{x}_2)^\beta S(\mathbf{x}_1, \mathbf{x}_2)^\gamma, \quad (3.6)$$

where the parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  should be greater than zero and allow to adjust the impact of the three components. The authors set  $\alpha = \beta = \gamma = 1$ .

The original SSIM score is also referred to as single-scale SSIM, because it operates only at one image scale. Disadvantages are the resulting assumption of a fixed image sampling density and viewing distance, which makes the score only appropriate for a certain range of image scales. Therefore, a *multi-scale SSIM* (*MS-SSIM*) was introduced [45], that operates on multiple by factor 2 iteratively downsampled scales of the original images. The score is given as

$$\text{MS-SSIM}(\mathbf{x}_1, \mathbf{x}_2) = I_M(\mathbf{x}_1, \mathbf{x}_2)^{\alpha_M} \prod_{j=1}^M C_j(\mathbf{x}_1, \mathbf{x}_2)^{\beta_j} S_j(\mathbf{x}_1, \mathbf{x}_2)^{\gamma_j} \quad (3.7)$$

with  $j$  denoting the downsampling stage with maximal stage  $M$ .

## 3.2 Inception Score

The idea behind the *Inception Score* (*IS*) [31] is that a generated high-quality image should be clearly assignable to one single class label, while the whole generated dataset should – in the best case – be maximum diverse, i.e., follow a uniform class distribution. To measure this, an *Inception v3 Network* [39] pre-trained on the *ImageNet* dataset [8] is applied to each of the synthetic images for classification. Compared to pixel-level metrics, this step incorporates the semantics of the images. For each image  $\mathbf{x}$  and the possible labels  $y$  the network outputs the conditional label distribution  $\pi(y|\mathbf{x})$ . Its entropy is expected to be low in contrast to the marginal distribution  $\pi(y)$ , whose entropy should be high. So the score is defined as

$$\text{IS} := \exp(\mathbb{E}_{\mathbf{x}}(\text{KL}(\pi(y|\mathbf{x})||\pi(y))))), \quad (3.8)$$

where KL denotes the *Kullback-Leibler divergence* defined as

$$\text{KL}(P||Q) := \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \quad (3.9)$$

and  $\mathbb{E}_{\mathbf{x}}$  the expected value whereat  $\mathbf{x}$  is assumed to be distributed uniformly. Since the IS measures the (inter-class) diversity, the authors note that it is important to apply the metric to a large enough number of samples. They propose a minimal size of 50 000.

### 3.2.1 Modified Inception Score

The authors of the *Modified Inception Score (m-IS)* [13] aimed to make the original IS with their modification sensitive for intra-class diversity. Therefore, they describe the diversity in a cross-entropy style manner as  $-\pi(y|\mathbf{x}_i)\log(\pi(y|\mathbf{x}_j))$  for  $\mathbf{x}_i$  and  $\mathbf{x}_j$  being samples representing the same class. So for each class, their modified score given as

$$\text{m-IS} := \exp\left(\mathbb{E}_{\mathbf{x}_i}\left(\mathbb{E}_{\mathbf{x}_j}\left(\text{KL}(\pi(y|\mathbf{x}_i)||\pi(y|\mathbf{x}_j))\right)\right)\right) \quad (3.10)$$

can be calculated and averaged to one single score result subsequently.

### 3.2.2 Mode Score

A drawback of the IS is that a good score can still be achieved if the Inception Network systematically misclassifies the generated images, meaning it predicts a false class with high confidence. This results in a low entropy of  $\pi(y|\mathbf{x})$ , the ground truth labels are consequently ignored. This drawback is intended to be solved by the *Mode Score (MS)* [6] defined as

$$\text{MS} := \exp\left(\mathbb{E}_{\mathbf{x}}\left(\text{KL}(\pi(y|\mathbf{x})||p(y_{\text{train}})) - \text{KL}(\pi(y)||p(y_{\text{train}}))\right)\right). \quad (3.11)$$

It compares  $\pi(y|\mathbf{x})$  and  $\pi(y)$  with the label distribution of the training dataset  $p(y_{\text{train}})$  (the ground truth).

### 3.2.3 AM Score

The *AM Score (AMS)* [48] was proposed to counteract the problem of the IS of penalizing an uneven generated data distribution – even if the training data is also unevenly distributed. This is why the authors suggest to also consider the training data label distribution  $\pi(y_{\text{train}})$ . The score is defined as

$$\text{AMS} := \text{KL}(\pi(y_{\text{train}})||\pi(y)) + \mathbb{E}_{\mathbf{x}}(H(y|\mathbf{x})), \quad (3.12)$$

where  $H(\cdot)$  refers to the entropy defined as  $H(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x)$  with  $X$  being a discrete random variable that takes values in the set  $\mathcal{X}$  with  $p: \mathcal{X} \rightarrow [0, 1]$ .

### 3.3 Fréchet Inception Distance

The *Fréchet Inception Distance (FID)* [14] was proposed as an improvement of the IS. It belongs to the feature-based metrics (cf. Sec. 3.4), however, due to its popularity and the many existing adaptations, this metric is described in this separate section. The FID puts synthetic data in context with real-world data. Therefore, each sample is first transformed into a 2048-dimensional feature space vector by extracting the coding layer output of an *Inception v3 Network* [39] pretrained on the *ImageNet* dataset [8]. In total, the dataset contains one million images categorized in 1000 classes. It is assumed that the extracted data points are multivariate Gaussian distributed, so the mean and covariance matrix are estimated for the synthetic data  $(\mu_s, \Sigma_s)$  as well as the real-world data  $(\mu_r, \Sigma_r)$ . The score is then obtained by calculating the Fréchet distance [10] (also known as Wasserstein-2 distance [42]), so

$$\text{FID}^2((\mu_s, \Sigma_s), (\mu_r, \Sigma_r)) = \|\mu_s - \mu_r\|_2^2 + \text{Tr}(\Sigma_s + \Sigma_r - 2(\Sigma_s \Sigma_r)^{\frac{1}{2}}). \quad (3.13)$$

#### 3.3.1 Class-Aware FID

The *Class-Aware Fréchet Distance (CAFD)* [23] uses a *Gaussian Mixture Model (GMM)* to include class-specific information. So for all samples of each of the  $K$  classes in the real-world and synthetic data an own Gaussian distribution  $\mathcal{N}_{ri}(\mu_{ri}, \Sigma_{ri})$  respectively  $\mathcal{N}_{si}(\mu_{si}, \Sigma_{si})$  for  $i = 1$  to  $K$  is estimated and the FID calculated:

$$\begin{aligned} \text{CAFD}(\mathcal{N}_{r1}, \mathcal{N}_{s1}, \dots, \mathcal{N}_{rK}, \mathcal{N}_{sK}) = \\ \frac{1}{K} \sum_{i=1}^K (\|\mu_{si} - \mu_{ri}\|_2^2 + \text{Tr}(\Sigma_{si} + \Sigma_{ri} - 2(\Sigma_{si} \Sigma_{ri})^{\frac{1}{2}})). \end{aligned} \quad (3.14)$$

The authors argue that the modulation by using a GMM is more accurate than estimating one single FID. Furthermore, CAFD is able to combine the advantages of FID with the ability of IS to be sensitive to mode dropping. However, the computing time increases considerably compared to the original FID. Apart from the metric, the authors recommend using a domain-specific encoder instead of an Inception model [39] trained on ImageNet [8].



### 3.3.2 WaM

A similar approach to CAFD is taken by Luzi et al. [24] who also recommend applying Gaussian Mixture Models because they can capture higher order moments and though more information. They estimate a GMM from the extracted feature vectors with  $K$  components where  $K \geq 1$  ( $K = 1$  is the original FID) and  $K \leq 50$  (for a dataset size of 50 000 to prevent overfitting). Then, they calculate the FID for each component combination of a real-world and synthetic component. The WaM score results by solving a discrete optimal transport problem using the Wasserstein-2 distance [42] squared as distance measure.

### 3.3.3 Spatial FID

Additionally to the coding layer feature vectors used for calculating the original FID, the authors of the *spatial FID* (*sFID*) [26] also extract the first seven channels from the intermediate sixth mixed convolutional feature maps of the Inception v3 Network [39] architecture. These feature vectors have a size of  $17 \times 17 \times 7 = 2\,023$ , which the authors argue is a comparable size to the 2 048-dimensional feature vectors resulting from the coding layer. By this addition, the authors want to keep the abstract compressed spatial information of the compressed layer but add the comparison of spatial distributional similarity.

### 3.3.4 Memorization-Informed FID

The FID and IS indicate a good quality for models playing back memorized training data. To overcome this issue, Bai et al. [1] suggest penalizing models producing images too similar to the training dataset. Their presented *Memorization-informed FID* (*MiFID*) takes as inputs a set  $S$  of embedded feature vectors generated by the synthetic data, a set  $R$  of feature vectors from real-world data, and the from the sets estimated Gaussian distributions  $(\mu_s, \Sigma_s)$  and  $(\mu_r, \Sigma_r)$ . The score is defined as

$$\text{MiFID}(S, R, (\mu_s, \Sigma_s), (\mu_r, \Sigma_r)) = \text{FID}((\mu_s, \Sigma_s), (\mu_r, \Sigma_r)) \cdot \text{pen}(S, R) \quad (3.15)$$

where the penalty function is given as

$$\text{pen}(S, R) = \begin{cases} \frac{1}{d_{\text{mem}}(S, R) + \epsilon} \quad (\epsilon \ll 1), & \text{if } d_{\text{mem}}(S, R) < \tau \\ 1, & \text{otherwise} \end{cases} \quad (3.16)$$

with the (asymmetric) memorization distance defined as

$$d_{\text{mem}}(S, R) = \frac{1}{|S|} \sum_{z_s \in S} \min_{z_r \in R} \left( 1 - \frac{|\langle z_s, z_r \rangle|}{|z_s| \cdot |z_r|} \right). \quad (3.17)$$

Thereby, a lower memorization distance means a stronger memorization.  $\tau$  refers to a predefined threshold.

### 3.3.5 Kernel Inception Distance

The FID is biased concerning the evaluated dataset size [7]. The *Kernel Inception Distance KID* [2] wants to mitigate this problem so that it can also be applied to smaller dataset sizes. Instead of estimating Gaussian distributions and calculating their Fréchet Distance, Bińkowski et al. suggest to compute the squared *Maximum Mean Discrepancy (MMD)*, given by

$$\begin{aligned} \text{MMD}^2(S, R) &= \frac{1}{n(n-1)} \sum_{i \neq j}^n k(s_i, s_j) + \frac{1}{m(m-1)} \sum_{i \neq j}^m k(r_i, r_j) \\ &\quad - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(s_i, r_j). \end{aligned}$$

Thereby,  $S$  denotes the set of embedded Inception Network feature vectors resulting from the synthetic data and  $R$  the set of feature vectors from real-world data. The authors recommend using the polynomial kernel  $k(x, y) = (\frac{1}{D}x^T y + 1)^3$ , where  $D$  denotes the representation dimension.

### 3.3.6 Fréchet AutoEncoder Distance

Instead of employing the Inception v3 Network [39] as a feature extractor like in the original FID score, the *Fréchet AutoEncoder Distance (FAED)* [5] suggests

to use a *Vector Quantised-Variational Autoencoder (VQ-VAE)* [41] to generate the feature vectors. The dimension of the feature vectors remains the same. The mean and covariance matrices are estimated and the score is calculated as for the original FID. According to the authors, the experiments showed that the feature space of the VQ-VAE describes a clustering domain-specific representation more intuitively and visually plausible than the feature space of the Inception v3 Network.

### 3.3.7 Other FID Enhancements

This section shortly highlights some additional FID enhancements that either do not represent an own metric or are not intended to be applied to a classical image dataset:

**Fast FID:** Calculating the FID is computationally intensive which makes it impracticable to use for training GANs. Therefore, the *Fast FID* [25] was proposed, whose time complexity is successfully reduced by, among other things, mathematically deriving an efficient alternative way to compute the computational intensive term  $\text{Tr}(\Sigma^s \Sigma^r)$  of the FID.

**Clean FID:** The experiments of Parmar et al. [27] showed that the FID is in practice sensitive to implementation details such as resizing the images, the used image compression technique, and the image pre-processing algorithm. Hence, the authors provide a library called *clean-fid*<sup>8</sup> to be able to quantify the image quality more objectively.

**FID<sub>∞</sub>:** Chong and Forsyth [7] show that the original FID is biased depending on the evaluated model. So they propose an extrapolation approach to compute a bias-free version of FID called FID<sub>∞</sub>.

**Single Image FID:** The *Single Image FID (SIFID)* [32] is a modified version of the FID intended to compare single images to each other. Instead of using the feature vectors from the last pooling layer, the authors propose to estimate the internal distribution of deep features at the output of the convolutional layer just before the second pooling layer.

---

<sup>8</sup> <https://github.com/GaParmar/clean-fid>

**Conditional FID:** Soloveitchik et al. [38] developed a conditional version of FID to evaluate conditional GANs.

**Fréchet Video Distance:** The *Fréchet Video Distance (FVD)* [40] is an enhancement of the FID for evaluating synthetically generated video sequences.

### 3.4 Further Feature-Based Metrics

Apart from FID and its adaptations, there are other metrics that utilize feature embeddings from deep neural networks (DNNs) and determine the distance between them or respectively between the underlying assumed distributions.

#### 3.4.1 Learned Perceptual Image Patch Similarity

The *Learned Perceptual Image Patch Similarity (LPIPS)* [47] was developed to estimate the perceptual distance between two images (or image patches)  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . Therefore, a deep neural convolutional network  $\mathcal{F}$  is used as a feature extractor, whereby one is not restricted to a specific architecture. The authors apply among others SqueezeNet [15], AlexNet [21], and VGG [36]. The metric operates on a feature stack of  $L$  layers which are unit-normalized in the channel dimension. The extracted features from layer  $l$  for  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are denoted as  $\hat{z}_1^l$  and  $\hat{z}_2^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ . The LPIPS metric is defined as

$$\text{LPIPS}(\mathbf{x}_1, \mathbf{x}_2) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\hat{z}_{1hw}^l - \hat{z}_{2hw}^l)\|_2^2. \quad (3.18)$$

Note that the activations get scaled channel-wise by vector  $w^l \in \mathbb{R}^{C_l}$ .

The authors distinguish between different variants of LPIPS, which depend on the training strategy of the network  $\mathcal{F}$ :

**lin:** Keep pre-trained network weights fixed, only learn linear weights  $w$ .

**tune:** Initialize from the pre-trained model, allow all weights to be fine-tuned.

**scratch:** Initialize  $\mathcal{F}$  from random Gaussian weights, train it from scratch.

### 3.4.2 CMMD Metric

For fine-grained or diverse image content, the resolution of 1 000 classes of the original FID may be too coarse. Therefore, Jayasumana et al. [17] recently introduced a new quality measure named *CLIP-MMD (CMMD)* that utilizes *Contrastive Language-Image Pre-training* [28] (*CLIP*) feature embeddings in combination with the squared MMD. CLIP’s image encoder and text encoder are jointly trained for the task of predicting correct image text pairs from a batch of examples. For this kind of training procedure, a high amount of training data is publicly available on the internet. So CLIP is trained on 400 million image-text pairs, making the resulting embeddings according to the authors well-suited for representing diverse and complex scenes. For calculating the distance between the distributions the real-world samples and synthetic samples were sampled from, the squared MMD (cf. Sec. 3.3.5) is used in combination with a Gaussian RBF kernel defined as  $k(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$ , whereby the authors choose  $\sigma = 10$ . Advantages compared to FID are that CMMD does not assume a specific underlying distribution, it is unbiased, and more sample-efficient.

## 3.5 Battle-Based Metrics

Battle-based metrics do not calculate a comparable score like the previously presented metrics. They allow both comparative approaches to compete against each other and determine the winner of the battle.

### 3.5.1 Generative Adversarial Metric

The idea of the *Generative Adversarial Metric (GAM)* [16] is to evaluate two GAN models  $M_1 = (G_1, D_1)$  and  $M_2 = (G_2, D_2)$  with generators  $G_1, G_2$ , and discriminators  $D_1, D_2$  as letting them “battle” against each other by swapping their discriminators. Intuitively, the model is to be assessed more strongly, whose generator is able to fool the other model’s discriminator more often. Therefore, the ratio  $r = \frac{\epsilon(D_1(X_2))}{\epsilon(D_2(X_1))}$  is calculated where  $X_1$  is the set of images generated by  $G_1$ , and  $X_2$  the set generated by  $G_2$ . The function  $\epsilon(\cdot)$  thereby determines the classification error rate. To rule out that one generator is subject

to overfitting and thus also the discriminator, a test ratio  $r_{\text{test}} = \frac{\epsilon(D_1(X_{\text{test}}))}{\epsilon(D_2(X_{\text{test}}))}$  is additionally calculated. So the test ratio aims to determine the output’s validity. Consequently,  $M_1$  is the winner if  $r < 1$  and  $r_{\text{test}} \approx 1$ ,  $M_2$  wins if  $r > 1$  and  $r_{\text{test}} \approx 1$ , otherwise the result is a tie. With this metric, GANs can only be compared pairwise, it can not be applied to general synthetic image datasets.

Durugkar et al. [9] refine GAM to their *Generative Multi-Adversarial Metric (GMAM)* which is capable of evaluating training with multiple discriminators.

## 3.6 Latent Space Metrics

This section presents metrics that operate on the generators ( $G$ ) latent space  $\mathcal{Z}$ . In general, the goal is to train generators so that their latent space is *disentangled*. This means that  $\mathcal{Z}$  consists of linear subspaces, each controlling a variation property without side effects [19].

### 3.6.1 Perceptual Path Length

If a generator’s latent space is disentangled, or at least little curved, the interpolation of latent space vectors should result in perceptually smoother transitions compared to the result generated from a highly curved latent space. So the authors of *Perceptual Path Length (PPL)* [19] propose to measure this effect by comparing perceptually-based image distances between images generated from interpolated latent space vectors. Therefore, they utilize the LPIPS metric (cf. Sec. 3.4.1) with an underlying VGG16 [36]. The PPL metric is defined as

$$\text{PPL} = \mathbb{E}\left(\frac{1}{\epsilon^2} \text{LPIPS}(G(\text{slerp}(z_1, z_2; t)), G(\text{slerp}(z_1, z_2; t + \epsilon)))\right), \quad (3.19)$$

where  $z_1, z_2 \sim p(z)$ ,  $t \sim U(0, 1)$ , and  $\epsilon = 10^{-4}$ . For interpolation, *spherical interpolation (slerp)* [33] is used, given by

$$\text{slerp}(v_1, v_2; t) = \frac{\sin((1-t)\theta)}{\sin(\theta)} v_1 + \frac{\sin(t\theta)}{\sin(\theta)} v_2, \quad (3.20)$$

where  $v_1 \cdot v_2 = \cos(\theta)$ . The authors calculate the expectation by considering 100 000 samples. PPL captures semantics as well as image quality [3].

### 3.6.2 Linear Separability

The authors of PPL introduce another metric in their work: the *Linear Separability* [19]. This metric is intended to measure how well the latent space image representations can be divided into two distinct sets via a linear hyperplane. Thereby, if the latent space is disentangled, each set represents a specific binary image attribute. As the first step, one needs to train  $N$  auxiliary, task-specific classification networks, which are able to classify binary attributes, e.g., for the generation of human faces ‘‘male’’ or ‘‘female’’. The authors trained  $N = 40$  of such binary classifiers in total and applied them on 200 000 synthetic generated images with  $z \sim p(z)$ . For each classifier, they sorted all samples by confidence and fitted a linear *Support Vector Machine (SVM)* on the latent space representations of the more confident half. Then the Linear Separability score is given by

$$\text{LS} = \exp\left(\sum_{i=1}^N H(Y_{\text{CLAS } i} | Y_{\text{SVM } i})\right), \quad (3.21)$$

where  $Y_{\text{CLAS}}$  is the set of classes predicted by the auxiliary classifiers and  $Y_{\text{SVM}}$  the set of classes predicted by the SVM. The function  $H(\cdot)$  refers to the entropy (see Sec. 3.2.3).

### 3.6.3 Adaptive Inversion

The process of inverting an image back into the latent space of a GAN model is called *Adaptive Inversion*. A recently published score [43] to evaluate GANs is based on this procedure: the authors propose to calculate the latent space representation by Adaptive Inversion for all reference images and to reconstruct new synthetic images from them. Then, the distance between each original and reconstructed image is determined by using the LPIPS metric (cf. Sec. 3.4.1). Intuitively, a short distance means, that the GAN’s generator is capable of recreating the corresponding reference image. The resulting score is the average of all image pair distances.

## 4 Discussion

The central question of this work for going towards quantifying the quality of simulated image sensor data is which metric(s) presented in the previous section are suitable for the evaluation of simulated image sensor data. In this regard, the individual metrics are rated in Table 4.2 based on the desired metric properties described in Sec. 2.2 and the derived categorization criteria from Sec. 2.3. Readers are encouraged to apply the provided metric comparison to find metrics that satisfy their requirements from a perspective of capabilities and available data, and then evaluate the suitable selection against the technical requirements of their use case.

Three metric classes under consideration can directly be excluded for the desired use of quantifying simulated image sensor data: these are on the one hand latent space metrics (cf. Sec. 3.6) and battle-based metrics (cf. Sec. 3.5), as these are both (without further enhancements) not applicable to simulated synthetic image data. Battle-based metrics utilize the discriminator of GANs, while latent space metrics make use of the GAN generator’s latent space—both do not exist for a simulation-based image generation approach. On the other hand, pixel-level metrics (cf. Sec. 3.1) are no suitable solution since they presume data pairs of real-world and synthetic data which is a non-negligible limitation and generally are rarely applicable for simulated image data. In addition, pixel-level metrics are not capable of measuring either discriminability nor diversity.

The IS and its adaptations, abbreviated as IS-based metrics, are in principle applicable to simulation-generated data. However, a serious disadvantage of some of the IS-based metrics is the need of a comparable real-world label distribution, which is typically measured from labeled real-world data. In general, labeled real-world data are costly to obtain, cause they are either expensive or time-consuming to collect, which is why it is advisable to prefer metrics operating on unlabeled comparison data. IS-based metrics without this constraint lack of being able to assess the discriminability on a high level.

The most promising metric category is therefore feature-based metrics. All metrics from this category achieve high ratings for discriminability and perceptual judgment. In terms of diversity, CAFD [23] and CMMD [17] stand out



by being able to measure both inter- and intra-class diversity. While all other feature-based metrics are capable of at least measuring the inter-class diversity, the LPIPS [47] metric is insensible for this data(set) property. The reason is that LPIPS was designed for being able to compare single images, which makes it the most sample-efficient feature-based metric. But also CMMD [17] and KID [2] can work with datasets of small size. CAFD [23] in contrast has a high demand for data with needing  $\sim 200\,000$  images per dataset for the score calculation.

Overall, it is not possible to identify the most suitable metric for quantifying simulated image sensor data due to the different strengths and weaknesses. Anyway, according to the conducted analysis, the following metrics have the highest potential for the task and should be further investigated in future experiments:

**LPIPS** [47] for being the most sample-efficient feature-based metric.

**CAFD** [23] for comparing the data class-wise what allows to compare the intra-class variance and to recognize class-specific differences.

**KID** [2] offers a good compromise of sample efficiency and measuring diversity.

**FAED** [5] combines the well-established FID [14] metric with feature embeddings of the state-of-the-art DNN architecture of VQ-VAEs.

**CMMD** [17] seems to have the highest potential based on the theoretical foundation: it is sample efficient while on the same time being able to measure diversity. Furthermore, the authors report a better perceptual judgment than FID and highlight, that CMMD is unbiased and does not assume a specific underlying distribution.

## 5 Conclusion and Future Work

The presented GAN evaluation metrics show the trend of evaluating increasingly “deeper”. The early metrics for assessing GANs [44, 46], originally developed for other use cases like, e.g., quantifying reconstruction quality, operate pixel- and pair-wise on the synthetic image data. While the IS already incorporates semantics by considering classification results from DNNs, feature-based metrics like FID [14], CMMD [17], or LPIPS [47] compare feature embeddings from

**Table 4.1:** Overview of the presented GAN evaluation metrics.

△ metric is aimed to be maximized, ▼ metric is aimed to be minimized

Name	Abbr.	Year	Range	Target Value
Mean Squared Error [44]	MSE	—	$[0, \infty)$	▼
Peak Signal to Noise Ratio [44]	PSNR	—	$[0, \infty)$	△
Structural Similarity Index Measure [46]	SSIM	2004	$[-1, 1]$	△
Inception Score [31]	IS	2016	$[1, \infty)$	△
Modified Inception Score [13]	m-IS	2017	$[1, \infty)$	△
Mode Score [6]	—	2016	$[0, \infty)$	△
AM Score [48]	AMS	2017	$[0, \infty)$	▼
Fréchet Inception Distance [14]	FID	2017	$[0, \infty)$	▼
Class-Aware FID [23]	CAFD	2018	$[0, \infty)$	▼
WaM [24]	WaM	2023	$[0, \infty)$	▼
Spatial FID [26]	sFID	2021	$[0, \infty)$	▼
Memorization-informed FID [1]	MiFID	2021	$[0, \infty)$	▼
Kernel Inception Distance [2]	KID	2018	$[0, \infty)$	▼
Fréchet AutoEncoder Distance [5]	FAED	2023	$[0, \infty)$	▼
Learned Perceptual Image Patch Similarity [47]	LPIPS	2018	$[0, \infty)$	▼
CLIP-MMD [17]	CMMD	2024	$[0, 2]$	▼
Generative Adversarial Metric [16]	GAM	2016	—	—
Perceptual Path Length [19]	PPL	2019	$[0, \infty)$	▼
Linear Separability [19]	—	2019	$[1, \infty)$	▼
Adaptive Inversion [43]	—	2024	$[0, \infty)$	▼

deep layers of DNNs. Recent metrics focus on analyzing the latent space of GAN generators [19, 43].

Despite continuously released further constructive developments in the research field of GAN quality measures, there is still not “the one” metric capable of covering all desired aspects — neither for evaluating synthetic GAN generated image datasets nor for simulation-based image datasets. Therefore, only promising metrics for quantifying simulated image sensor data can be highlighted from the remaining metrics, rather than presenting the best one. Overall, the class of

**Table 4.2:** Analyzed strengths and weaknesses of the described GAN quality metrics, following the structure of Borji [3]. The column “additional sources” provides used sources alongside the original ones given in the first column for the assessment of each row.

●●● high, ●●○ moderate, ●○○ low, ○○○ not measured, ✓ yes, × no  
 P: paired data needed, U: unlabeled data sufficient, L: labeled data needed

Metric	Add. Src.	Discr.	Div.	Detect overfit.	Percep. judg.	Sample effic.	Appl. to sim	Comp. data
MSE [44]	[47]	○○○	○○○	○○○	●○○	●●●	✓	P
PSNR [44]	[47]	○○○	○○○	○○○	●○○	●●●	✓	P
SSIM [46]	[47]	○○○	○○○	○○○	●●○	●●●	✓	P
IS [31]	[3]	●●○	●●○	○○○	●●●	●○○	✓	U
m-IS [13]	[3]	●●○	●●○	○○○	●●●	●○○	✓	U
Mode Score [6]	[3]	●●●	●●○	○○○	●●●	●○○	✓	L
AMS [48]	[3]	●●○	●●○	○○○	●●●	●○○	✓	L
FID [14]	[3]	●●●	●●○	○○○	●●●	●○○	✓	U
CAFD [23]		●●●	●●●	○○○	●●●	●○○	✓	U
WaM [24]		●●●	●●○	○○○	●●●	●○○	✓	U
sFID [26]		●●●	●●○	○○○	●●●	●○○	✓	U
MiFID [1]		●●●	●●○	●●●	●●●	●○○	✓	U
KID [2]		●●●	●●○	○○○	●●●	●●●	✓	U
FAED [5]		●●●	●●○	○○○	●●●	●○○	✓	U
LPIPS [47]		●●●	○○○	○○○	●●●	●●●	✓	U
CMMD [17]		●●●	●●●	○○○	●●●	●●●	✓	U
GAM [16]	[3]	●○○	●○○	●●●	—	●○○	×	—
PPL [19]		○○○	○○○	●○○	—	●●●	×	U
Lin. Sep. [19]		●●●	○○○	●○○	—	●○○	×	L
Adpt. Inv. [43]		○○○	○○○	○○○	—	●●●	×	U

feature-based metrics is identified to have the highest potential for the desired task. From this class, the metrics LPIPS [47], CAFD [23], KID [2], FAED [5], and especially CMMD [17] were identified to offer the highest potential for the task of quantifying simulated image sensor data.

The theoretical findings from the analysis conducted in this work should be practically verified in future follow-up studies. Moreover, it is advisable to regularly research new developments in GAN evaluation metrics, as the research field is very dynamic.

## References

- [1] Ching-Yuan Bai et al. “On Training Sample Memorization: Lessons from Benchmarking Generative Modeling with a Large-scale Competition”. In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 2534–2542.
- [2] Miłkołaj Bińkowski et al. “Demystifying MMD GANs”. In: *arXiv preprint arXiv:1801.01401* (2018).
- [3] Ali Borji. “Pros and cons of GAN evaluation measures”. In: *CoRR* abs/1802.03446 (2018). arXiv: 1802.03446. URL: <http://arxiv.org/abs/1802.03446>.
- [4] Ali Borji. “Pros and cons of GAN evaluation measures: New developments”. In: *Computer Vision and Image Understanding* 215 (2022), p. 103329.
- [5] Lucas F Buzuti and Carlos E Thomaz. “Fréchet AutoEncoder Distance: A new approach for evaluation of Generative Adversarial Networks”. In: *Computer Vision and Image Understanding* 235 (2023), p. 103768.
- [6] Tong Che et al. “Mode Regularized Generative Adversarial Networks”. In: *arXiv preprint arXiv:1612.02136* (2016).
- [7] Min Jin Chong and David Forsyth. “Effectively unbiased fid and inception score and where to find them”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6070–6079.
- [8] Jia Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [9] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. “Generative Multi-Adversarial Networks”. In: *arXiv preprint arXiv:1611.01673* (2016).
- [10] Maurice Fréchet. “Sur la distance de deux lois de probabilité”. In: *Annales de l’ISUP*. Vol. 6. 3. 1957, pp. 183–198.
- [11] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

- [12] Chuan Guo et al. “On calibration of modern neural networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1321–1330.
- [13] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. “DeLiGAN: Generative Adversarial Networks for Diverse and Limited Data”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 166–174.
- [14] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Advances in neural information processing systems* 30 (2017).
- [15] Forrest N Iandola. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).
- [16] Daniel Jiwoong Im et al. “Generating images with recurrent adversarial networks”. In: *arXiv preprint arXiv:1602.05110* (2016).
- [17] Sadeep Jayasumana et al. “Rethinking FID: Towards a Better Evaluation Metric for Image Generation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 9307–9315.
- [18] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, et al. “Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks?” In: *arXiv preprint arXiv:1610.01983* (2016). Sridhar, Sharath Nittur and Rosaen, Karl and Vasudevan, Ram.
- [19] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.
- [20] Tae Soo Kim, Bohoon Shim, Michael Peven, et al. “Learning From Synthetic Vehicles”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Qiu, Weichao and Yuille, Alan and Hager, Gregory D. Jan. 2022, pp. 500–508.

- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in neural information processing systems* 25 (2012).
- [22] Yueyuan Li et al. “Choose Your Simulator Wisely: A Review on Open-Source Simulators for Autonomous Driving”. In: *IEEE Transactions on Intelligent Vehicles* (2024).
- [23] Shaohui Liu et al. “An Improved Evaluation Framework for Generative Adversarial Networks”. In: *arXiv preprint arXiv:1803.07474* (2018).
- [24] Lorenzo Luzi et al. “Evaluating generative networks using Gaussian mixtures of image features”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 279–288.
- [25] Alexander Mathiasen and Frederik Hvilshøj. “Backpropagating through Fréchet Inception Distance”. In: *arXiv preprint arXiv:2009.14075* (2020).
- [26] Charlie Nash et al. “Generating images with sparse representations”. In: *arXiv preprint arXiv:2103.03841* (2021).
- [27] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. “On Aliased Resizing and Surprising Subtleties in GAN Evaluation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11410–11420.
- [28] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.
- [29] Stephan R Richter et al. “Playing for Data: Ground Truth from Computer Games”. In: *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 102–118.
- [30] German Ros, Laura Sellart, Joanna Materzynska, et al. “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vazquez, David and Lopez, Antonio M. June 2016.

- [31] Tim Salimans et al. “Improved Techniques for Training GANs”. In: *Advances in neural information processing systems* 29 (2016).
- [32] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. “SinGAN: Learning a Generative Model from a Single Natural Image”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4570–4580.
- [33] Ken Shoemake. “Animating Rotation with Quaternion Curves”. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. 1985, pp. 245–254.
- [34] Anne Sielemann et al. “Synset Boulevard: A Synthetic Image Dataset for VMMR”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024.
- [35] Anne Sielemann et al. “Synset Signset Germany: A Synthetic Dataset for German Traffic Sign Recognition”. In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. 2024.
- [36] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).
- [37] Jake Snell et al. “Learning to generate images with perceptual similarity metrics”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2017, pp. 4277–4281.
- [38] Michael Soloveitchik et al. “Conditional Frechet Inception Distance”. In: *arXiv preprint arXiv:2103.11521* (2021).
- [39] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [40] Thomas Unterthiner et al. “FVD: A new Metric for Video Generation”. In: (2019).
- [41] Aaron Van Den Oord, Oriol Vinyals, et al. “Neural Discrete Representation Learning”. In: *Advances in neural information processing systems* 30 (2017).

- [42] Leonid Nisonovich Vaserstein. “Markov Processes over Denumerable Products of Spaces, Describing Large Systems of Automata”. In: *Problemy Peredachi Informatsii* 5.3 (1969), pp. 64–72.
- [43] Jianbo Wang, Heliang Zheng, and Toshihiko Yamasaki. “Reference-based GAN Evaluation by Adaptive Inversion”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2024, pp. 910–918.
- [44] Zhou Wang and Alan C Bovik. “Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures”. In: *IEEE signal processing magazine* 26.1 (2009), pp. 98–117.
- [45] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. Ieee. 2003, pp. 1398–1402.
- [46] Zhou Wang et al. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.
- [47] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 586–595.
- [48] Zhiming Zhou et al. “Activation Maximization Generative Adversarial Nets”. In: *arXiv preprint arXiv:1703.02000* (2017).



# **Advancing Adaptive Learning in Non-Stationary Environments: Challenges, Methods, and Future Directions**

*Benedikt Stratmann*

Fraunhofer Institute of  
Optronics, System Technologies and Image Exploitation (IOSB)  
Karlsruhe, Germany  
benedikt.stratmann@iosb.fraunhofer.de

## **Abstract**

This report explores the current landscape and challenges of adaptive learning in non-stationary environments, where systems must adjust to continuous changes in data distribution, known as concept drift. While significant progress has been made, existing methods often remain constrained by narrow applicability to specific domains and require specialized expertise. This report reviews existing techniques, emphasizing the need for universally accessible adaptive learning systems that is applicable to all kind of non-stationary environments. We propose an innovative framework inspired by drift velocity profiles [3], aiming to infer system configurations over time and address the 'what' and 'why' of drifts. Key desirable features for future adaptive learning algorithms are outlined, targeting improved versatility, accuracy, and explainability in industrial settings. By advancing beyond mere detection and adaptation, this work sets the stage for developing robust, model-agnostic solutions capable of proactive drift management in diverse applications.

# 1 Motivation

In today's rapidly evolving industrial landscape, the prevalence of non-stationary environments presents significant challenges for predictive modeling and system analysis. Non-stationary environments, characterized by their dynamic nature, often undergo changes over time due to factors such as equipment degradation, sensor wear, or variations in product and material properties. These changes necessitate the development and implementation of robust methods to detect shifts in the underlying processes, ensuring the reliability and accuracy of predictive models.

Various methodologies exist for identifying and adapting to these changes across different tasks, including regression, classification, clustering, and time series learning, all of which are critical components of big data and machine learning applications. However, the sheer volume of available methods can be overwhelming. To navigate this complexity, it is essential to categorize these methods based on their structural and methodological properties.

While these methods aim to address the symptoms of change through adaptation and detection, approaches that investigate the source or nature of the change are relatively rare. Even when methods extend beyond merely coping with change, they typically focus on defining metrics or classifying different classes of change rather than providing deeper insights. Consequently, the analysis of changes in non-stationary environments reveals a significant gap in research, highlighting the need for more comprehensive studies that explore the underlying mechanisms driving these changes.

## 1.1 Problem Statement

**Non-Stationary environments** are characterized by their time changing data streams. These changes can frequently be expressed as distribution shifts in the data stream. The direction, speed and cause of the process that produces these distribution shifts can be very different and is found under the notion of drift [14]. While there are different types of drift they all have something in common. The data generating process underlying the data stream changes over time.

If we have additional time stamps for a dataset  $D = \{d_i = (x_i, y_i) \in \mathbb{D} | i \in \mathbb{N}\}$  of a data space  $\mathbb{D} = X^n \times Y^m$  we can formulate it as a data stream:

$\Omega = \{(d_i, t_i) \in \mathbb{D} \times \mathbb{R} | i \in \mathbb{N}\}$ .  $X$  and  $Y$  are the feature and target/label spaces respectively.

The data generating process itself can be viewed as sampling from the system concept  $C \in \mathcal{P}(\mathbb{D})$ , which is one of the possible distributions over the viewed data space [64]. If  $C$  is part of a non-stationary environment we can *not* assume that  $\forall t_1, t_2 : C_{t_1} = C_{t_2}$  for  $C_t$  being the concept at a certain point in time  $t$  [23].

This can now be specified to describe specific types of drifts that alter  $C$  in different ways. This leads to drifts that can cause problems for data mining and machine learning applications. The specific types of drifts mentioned in this work are listed and defined in Table 1.1.

The most characteristic distinction being that of Virtual Drift and Actual Drift. Virtual Drift occurs when the relationship between the feature and target space remains stable, but the distribution of the feature domain shifts. This can happen due to changes in basic inputs, such as materials or speed. Although this represents a drift, the data generated does not contradict the model; instead, it reveals previously unseen or rare areas of the dataspace, introducing new information. In contrast, Actual Drift alters the relationship between the feature and target domains while keeping the feature domain's behavior stable. This leads to contradictory information that can complicate modeling efforts and often results from changes not reflected in the feature space, acting as a latent influence. Concept Drift and Real Concept Drift are generalizations of these cases, where less information is available about the system and a further distinction is not possible.

**Adaptive Learning** is concerned with using methods that enable the efficient analysis of non-stationary environments despite their unstable nature. Some detection and adaption methods are designed to handle specific types of drifts [60, 26, 48]. The primary goal of these distinctions is to reduce false detection rates, as well as to only react to drifts that may interfere with a related models performance.

The general goal of adaptive learning can be formulated as follows: Given a data stream  $\Omega$  we want to use a learning algorithm  $A$  that is able to infer a model for

**Table 1.1:** Definitions of drift types with specified names [14].

Drift Name	Definition
Concept Drift	$C_t(x, y) \neq C_{t+\Delta}(x, y)$
Real Concept Drift	$C_t(y x) \neq C_{t+\Delta}(y x)$
Virtual Drift	$C_t(y x) = C_{t+\Delta}(y x) \wedge C_t(x) \neq C_{t+\Delta}(x)$
Actual Drift	$C_t(y x) \neq C_{t+\Delta}(y x) \wedge C_t(x) = C_{t+\Delta}(x)$

a target time  $t$ :

$$A(\Omega, t) = \mathcal{M}_t \quad (1.1)$$

A model  $\mathcal{M} : X^n \rightarrow Y^m$  is able to make predictions for the targets  $y \in Y^m$  given the features  $x \in X^n$ . While the quality of a single model can be measured by the mean absolute error (MAE) or root mean squared error (RMSE), the quality of a learning algorithm that is applied to a data stream should be measured differently. As we know that the concept of the environment can change it could be that a different model is used for every prediction. To make a prediction for a data entry  $d_i$  at time  $t_i$ , we should take into account all information that is available at time  $t_i$ . In an optimal system we could use  $\Omega_{<t_i} = \{(d_j, t_j) \in \mathbb{D} \times \mathbb{R} | i \in \mathbb{N} \wedge t_j < t_i\}$  as the information that is available to prepare a model for a prediction at time  $t_i$  [58]. In reality, we can be faced with labeling or validation delays, limiting the available information to  $\Omega_{<t_i}^* = \{(d_j, t_j) \in \mathbb{D} \times \mathbb{R} | i \in \mathbb{N} \wedge t_j < t_i - \Delta\}$ , where  $\Delta \in \mathbb{R}$  is an application specific delay period.

To measure the performance of a data stream or online learning algorithm, we can also use the MAE and RMSE if we adjust for the changed learning problem:

$$\text{MAE}_{\Omega}(A) = \frac{\sum_{i=0}^{|\Omega|} |y_i - A(\Omega_{t_i}^*, t_i)(x_i)|}{|\Omega|} \quad (1.2)$$

$$\text{RMSE}_{\Omega}(A) = \sqrt{\frac{\sum_{i=0}^{|\Omega|} (y_i - A(\Omega_{t_i}^*, t_i)(x_i))^2}{|\Omega|}} \quad (1.3)$$

### **Regression on time changing industrial data**

Depending on the field of application the problem statement can be specified even more. In this work, we focus on industrial applications in the form of cyber physical systems (CPS). These applications are characterized by their sensor driven data spaces, which results in continuous feature spaces and depending on the task discrete or continuous target spaces. Among all machine learning tasks encountered in CPS, regression (predicting continuous targets) stands out as the most relevant, followed by classification (predicting discrete targets) and other unsupervised tasks [41]. Although there is a multitude of classification methods available for CPS and related fields, the emphasis on regression techniques in the domain is limited [58]. This work therefore focuses on adaptive regression learning algorithms, which operate on the data space  $\mathbb{D} = X^n \times Y^m = \mathbb{R}^n \times \mathbb{R}^m$ .

## **2 Related Work**

This report will mention and compare the currently available methods for adaptive learning, but should not be seen as a systematic literature review, as the associated completeness is out of scope. This section may point the interested reader to related publications that provide a deeper view of the fields of adaptive learning and comparable problems.

### **2.1 Field Analysis and Surveys**

The field of adaptive learning encompasses various topics such as drift detection, adaptation, and analysis techniques. Numerous literature reviews focus on specific subtopics within this vast area. For instance, one work provides an overview of existing approaches for process event mining, emphasizing discrete information spaces represented by event logs and exploring the extraction of causal effects from changes in this context [1]. A related study addresses similar themes [56].

Another survey investigates methods for classification and clustering in non-stationary data streams, particularly in scenarios with sparse labels. This study

highlights the use of unsupervised models to support adaptive classification [26].

Several studies provide functional comparisons of drift detectors, focusing on their speed and accuracy across different variations of concept drift. These comparisons enhance the understanding of available drift detection methods [52] [14].

Some works analyze methods for handling concept drift based on their structure and applications, aiming to clarify the different types of adaptive learning approaches and their respective advantages and disadvantages [48] [2] [23] [47]. Efforts have been made to establish a unified taxonomy for concept drift adaptation methods, aiming to organize the field and enhance the understanding of various approaches. These taxonomies have been widely adopted and remain relevant in contemporary research [30] [14].

Lastly, surveys focused on feature drift address the significance of feature selection in learning algorithms, particularly in large-scale data operations [13] [58].

## 2.2 Related Fields

Learning in non-stationary environments is by its definition associated with comparable fields that seem to be the same, but can be distinguished through closer inspection. The most prominent of course being **continual learning** [45]. This topic is also mentioned by many works referenced in section 2.1 and can be seen as the basic building block of adaptive learning. Continual or incremental learning frequently assumes that not all information is provided with the initial training set, but has to be gathered in continuous deployment. While this is also an assumption of adaptive learning, we can not assume that all information is helpful information that can be included without further introspection. The possibility that future information can contradict information of the provided training set and the task of accessing if this is significant enough to cause a paradigm shift is what differentiates adaptive learning from continual learning. If no changes occur however, continual learning is one of the fundamental methods that enable adaptive learning in the first place. This and comparable relations of fields can also be seen in Figure 2.1.



**Figure 2.1:** Connection of the field of non-stationary environments with adaptive learning and other fields.

**Transfer Learning** is also a term frequently mentioned when discussing adaptive learning. The reasoning being based upon the notion that adaptive learning tries to change a model from one system concept to another. One could say: adaptive learning does transfer learning from one system concept to another. This is not the case, as transfer learning has more information available than adaptive learning. Firstly transfer learning methods know when a transfer is happening, secondly data from the starting and goal domain are usually available and lastly the domain changes are usually not limited in system concept drift but incorporate change in the number of input and output dimensions, or complete new classes. Therefore, adaptive learning is more sophisticated than transfer learning as it has to detect domain transitions online but is less complex in regards to the adaptations themselves.

### 3 Method Review

The body of work that is concerned with handling non-stationary environments in regards to learning is vast [41]. While there are approaches for process mining and unsupervised problems in this work we will focus on supervised learning and therefore prediction tasks. As adaptive learning is often comprised of different interconnecting components, like drift detection, model adaption and memory handling the pure combination of approaches provides a large number of comparable but distinct methods. For example: The mentioned approaches to detect changes in data generating processes can be grouped into: Hypothesis Tests, Change-Point Methods, Sequential Hypothesis Tests, and Change Detection Tests [23]. All of these different detection approaches can then be paired with a suitable adaption method.

An exemplary collection of frequently used methods can be seen in Table 3.1. This table is by no means complete and can be extended by many more works. This collection is based on the set of surveys [1, 56, 26, 52, 48, 2, 23, 47, 30, 14, 13, 58] discussed previously with the addition of a few more recent works. All of the mentioned works present supervised learning solutions and focus on providing accurate drift detection and/or accurate target prediction.



### 3.1 Types of Methods

While the methods are comparable, they can also be categorized and distinguished by the tasks they are able to perform and the paradigms they use.

**Task:** The most important **tasks** covered by these methods is **adaption**, followed by **detection**. Adaption describes the procedure through which the method tries to keep the corresponding ML model up to date, while detection is focused on determining at which point adaption might be useful or required. These are the how and when of adaptive learning. Beyond this, there are other tasks like **learning paradigm** and **system analysis**, the what and why, which are only rarely explored.

**Policy:** The way through which adaption and/or detection is achieved differs between methods, but the following categories can be identified:

**Active:** All methods which distinguish between adaption and detection and do not continuously apply an adaption scheme can be classified as active adaptive methods. These methods only trigger change in their models if they have found significant proof in the incoming data that a drift took place and adaption is necessary. If no drift takes place, these methods can therefore not trigger adaption at all and fully rely on continual learning.

**Passive:** Methods that do not require detection as a trigger for their adaption procedure are named passive adaptive methods. In these approaches the suspicion of drift is always present and adaption is therefore executed at every step possible. This can lead to very fast adaption, but is also data inefficient, as data entries are frequently discarded even though they are not counterfactual to the current presumed system concept.

**Performance:** The first possibility to measure the change in a system is by observing the performance of the current model in predicting the system. Under the assumption of the Probability Approximately Correct (PAC) paradigm the model performance should improve with more available data if the system concept remains stable. If the performance degrades too drastically a concept change can therefore be inferred. This policy is not capable of distinguishing between different kinds of drifts. Model degradation can therefore originate from

a real concept drift or a virtual drift, where the virtual drift could be compensated with retraining.

**Distribution:** The second approach to identify changes is the monitoring of distributions. This approach is independent of the model of the system and solely relies on the incoming data but is heavily dependent on suitable and sufficiently accurate change tests or similarity metrics. These approaches lend more capability to analysis and accuracy but can be comparably expensive to compute. The main advantage of this approach is that it can be applied to both labeled and unlabeled datasets since it only considers the distribution of data points. However, changes in the data distributions do not always affect the predictive performance, potentially leading to alarms for drift even if the model accuracy is not endangered.

**Classification/ Regression:** As the statistical tests and metrics underlying the methods are characteristic to their functionality, adapting a method to a different task can be difficult to impossible. Especially, the transition from classification (discrete target domain) to regression (continuous target domain) can be very challenging. One example is the application of statistical tests for measuring performance degradation. While performance metrics for classification are usually bounded regression metrics can be arbitrarily high. This disqualifies many frequently used statistical tests or requires changes in the regression learning goal itself. Keeping the original domain of the method in mind is therefore paramount to its successful application.

## 4 Problem with the current state of the field

While many methods exist that can handle learning in non-stationary environments, as showcased in the last two sections, many methods are only suited for handling specified drift conditions [14]. Methods can be limited to drifts in singular signals, specific speeds of drift or may depend on a specific model architecture to work as intended. In addition, methods are only designed to detect or handle drift to ensure the validity of corresponding measurements or predictions. The notion of explainability, e.g. where in the data space the drift took place or what caused the drift, is only rarely explored and relies on external

approaches for analysis. The current state of methods is to handle non-stationary effects, not to analyze and understand them.

Given a new use case, the selection of a suitable method becomes difficult as four options present themselves.

**Integrated Adaptive Model:** Choose a learning agent that is already internally constructed to enable adaptive behavior. The adaptive tree learner CVFDT [35] could be such an option. These approaches often rely on internal model architecture to enable detection and or adaption, for example decision tree split criteria. Changing the used model type is not possible in these cases and can heavily limit model expressiveness. If applicable this option is usually the preferred solution.

**Detection & Adaption** Combine a general drift detector and a general adaption strategy with any predictive model. As the functional interfaces between adaption strategies and drift detectors are very lightweight, not all detectors and strategies are compatible and performance of these plug and play systems can be experimental at best. As most detectors only focus on specific drifts unforeseen edge cases in the application can be overlooked.

**Passive Adaption:** Take any predictive model and retrain it over and over again with the most recent data points. While this is computational expensive and inefficient in regards to knowledge management, continuous data generation might also be prohibitively expensive.

**Ensemble Adaption:** Use a combination of drift detector and adaption strategy that is aimed and balancing ensembles of models. One example is the Learn++.NSE [25] approach. The choice of underlying method is usually not limited and adaption is achieved by training multiple basemodel instances on different substrata of the provided data stream and using ensemble weighting to get the best results according to some adaptive loss metric (e.g.  $MAE_{\zeta}$ ). These approaches are either lightweight and just add more and more model instances to the ensemble, which makes the model increasingly expensive to maintain, or they apply active pruning. Both cases are marked with drawbacks. While growing ensembles are hard to analyze the pruning based approaches suffer from the same problems as **passive adaption** and/or **detection & adaption**, depending on how the pruning is realized.

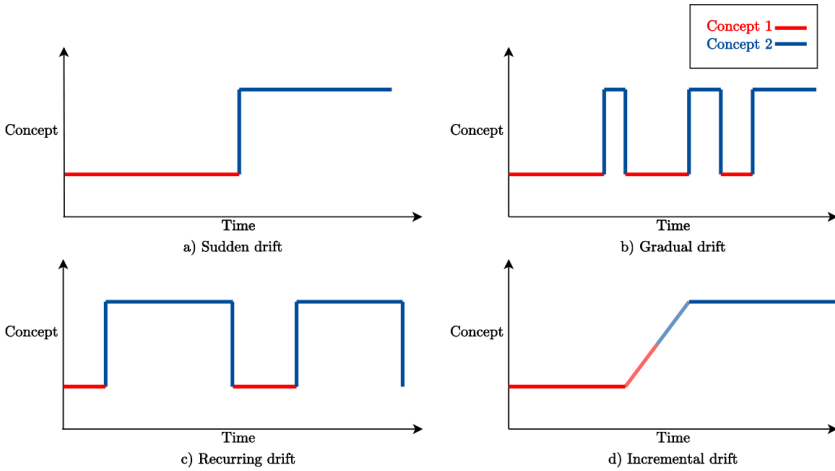


Figure 4.1: Set of frequently referenced drift patterns. Taken from [14]

## 4.1 Limiting assumptions

While a limitation to only real concept drift for performance based detection methods or concept drift/virtual drift for distribution based drift detectors is usual, most methods have further limitations regarding their intended use [45].

**Drift Pattern:** Drifts of any kind can appear in different realizations. One set of frequently used variations can be seen in Figure 4.1. The simplest version is the sudden drift, which all detectors can identify. After that some detectors are aimed at identifying incremental or gradual drifts and after that some might even be able to identify and handle reoccurring drifts.

To handle the whole spectrum for a given use case one might have to combine multiple detectors, which might lead to frequent false positive detections, as all detectors operate independently. This pattern recognition also assumes that two predefined states exist: the concept before the drift and the concept after the drift. This leads us to our next limiting assumption.

**Stationary Sections:** Most detection and adaption methods assume that between drifts there exists a time frame in which the system maintains a stable stationary concept. While this can be the case for use cases where drifts occur through tool changes or something of a similar nature, this is not true if the drifts are caused by something like tool degradation or seasonal temperature changes. For these scenarios the system is in a constant state of drift and no stationary intervals can be found that may be used to refit the model or recalibrate the detector. Some approaches also require a defined beginning and endpoint of the drift to select which information can be used for future identification.

**Binary Detections:** If detector and adapters are used in combination to insure model compliance the interface they most often connect over is the information for which timestamps in the data stream drift was detected. The information is presented in a binary format, either a timestamp is not drifted or it is. This is of course another highly limiting factor, as the speed and amplitude of the concept change may vary, i.e. there are smaller and larger drifts. Drift detectors can be parameterized to increase their sensitivity so that they may also find smaller or slower drifts, but an increase in sensitivity also increases the false detection rate. Passive adaptive methods are not faced with this particular problem, as they can continuously change model parameters to comply and are not limited to decide definitively.

## 5 Methods that go beyond detection and adaption

As described in previous sections, there are tasks in adaptive learning that go beyond detection and adaption. Answering the what and why of drifts, i.e. describing what a drift causes in the behavior of the system and understanding what caused the drift itself.

Here we will shortly describe the related body of work that tries to do deeper analysis of non-stationary environments beyond the handling for learning tasks.

**Understanding** refers to methods that try to make drifts more predictable or unveil their driving factors. Particular emphasis is on the human interpretability of this system analysis. For systems that are in drift [33] propose to measure the

mean direction of the distribution shift and explore the systemic meaning to make drifts more understandable for maintainers and operators. Hinder et al. present a solution that uses an explainable classifier to discern data before and after drift occurrences, enabling the identification of structural disparities between these sample populations [34]. An alternative, system-agnostic approach inspects temporal and spatial drifts in data distribution and extracts *interesting* subspaces to inspect, using appropriate visualizations [3].

Leveraging the ML model itself as a knowledge representation, [57] employ an extended Kalman-Filter to predict the future development of model parameters, thereby enabling preemptive adaptation. Although this approach seems to provide impressive results, it is limited to knowledge representations with continuous parameters and loses efficiency for models with large numbers of parameters.

By distinguishing parameters that model the system at a time point from parameters that model the system over time, the extraction of drift sources can be performed according to [18]. By employing a Naive Bayes classifier where the feature likelihood is dependent not only on the class but also on a term directly influenced by time, the system concept can be differentiated from the system change over time. While this approach is very powerful in theory, it can only extract drifts for singular time points. To make the modeling of drifts continuous, the approach has to be extended significantly, making the inference of the model many times more complex.

**Characterizing** aims at identifying, distinguishing and describing drifts. This can take the form of defining characterizing tuples as in [62], which enable a relative comparison between drifts. Another approach is the development of metrics, measuring the drift magnitude and what attributes of the system are affected most by mapping and measuring the differences in data probability distributions at different points in time [64, 63].

Finally, we would like to highlight the concept introduced by [3], which presents the idea of velocity profiles to describe the temporal and spatial dynamics of data spaces in non-stationary environments. This analysis not only identifies which parts of the data space are changing but also indicates the direction of probability shifts, allowing for the indirect modeling of drifts rather than merely detecting

their symptoms. [3] claim that this method is primarily a visualization technique for the analysis of data streams but the expansion of their concept might lead to a process that can be used to model the drifts instead of the intermediate stationary concepts.

## 6 Discussion

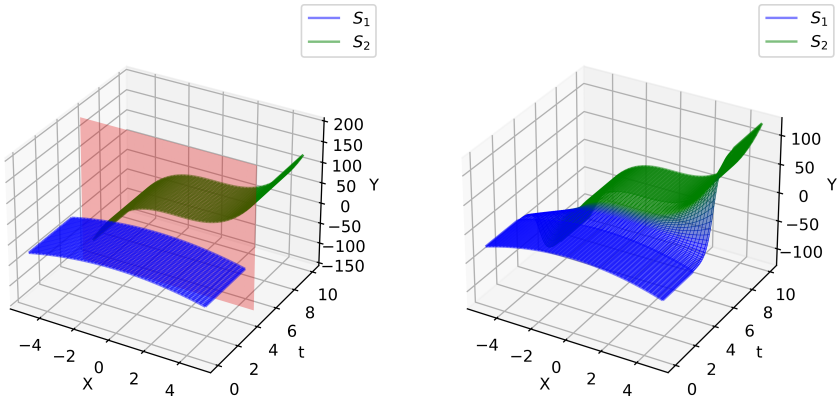
Given the current state of the literature and the known limitations and drawbacks of current adaptive learning approaches, there are numerous avenues for further research and development. Current methods have achieved significant progress, yet many remain constrained by their narrow applicability to specific domains, datasets, or expertise requirements. In fields with limited access to specialized expertise, there is a strong need for adaptive learning systems that are universally accessible and require minimal technical depth for successful implementation. The creation of a robust, adaptable framework could enable broader, easier adoption across diverse industries, particularly in settings where complex change dynamics demand flexible responses.

An approach that extends the drift velocity profiles of [3] might be able to provide such a system and task agnostic approach while being suited for many different forms of non-stationary environments.

### 6.1 Desirable Method

Although there are existing methods capable of adaptive learning within specific applications, several key challenges remain unmet. Below, we outline the primary desirable features for an adaptive learning algorithm, selected based on the limitations of current methods and insights from recent work [23, 26]. These properties, we argue, are essential for a method to be widely applicable in industrial, non-stationary environments and for it to remain effective under varied operational conditions.

First of all: One of the major limitations of passive adaptive methods is their restricted capability to provide insights into the nature of the detected changes,



(a) State of the art approach for adaption of models to a new system configuration.

(b) Proposed approach for the modeling of drift and resulting adaption.

**Figure 6.1:** Visualization of a two system configurations  $S_1$  and  $S_2$  that are present during different time intervals. Comparison of two drift adaption methodologies indicated by the red region. Instead of drift detection with binary outcome and a corresponding adaption based on data from  $S_2$  (left canvas), we propose to model the time dependent transition between  $S_1$  and  $S_2$  (right canvas).

which reduces their ability to inform actionable responses. By contrast, a more active adaptive policy could respond not only by adapting to changes but also by identifying and potentially preempting further challenges associated with the change dynamics. A method that tries to solve the mentioned problems should therefore be oriented on the active adaptive policy.

The following list details the key properties that an optimal adaptive learning algorithm should possess to ensure versatility and resilience in industrial non-stationary environments:

**Detect Concept Drift** The method should continuously monitor the data stream and promptly identify concept drift  $C_t(x, y) \neq C_{t+\Delta}(x, y)$  allowing for timely responses to these changes.



**Adapt for Real Concept Drift** Upon detecting real drift  $C_t(y|x) \neq C_{t+\Delta}(y|x)$  (as opposed to virtual drift), the algorithm should dynamically adjust its model parameters to accommodate the new data distribution.

**Remove Invalid Data** In the case of actual drift  $C_t(y|x) \neq C_{t+\Delta}(y|x) \wedge C_t(x) = C_{t+\Delta}(x)$  an effective adaptive learning system should be capable of recognizing and filtering out data that no longer reflects the current concept, thus preventing obsolete data from contaminating the model and reducing performance.

**Incorporate New Information** In cases of virtual drift  $C_t(y|x) = C_{t+\Delta}(y|x) \wedge C_t(x) \neq C_{t+\Delta}(x)$  the algorithm should retain relevant historical data while adapting to the variability, thus balancing stability and flexibility in model performance.

**High Accuracy** A fundamental requirement, the method must maintain high predictive accuracy despite concept drift and adapt seamlessly to ensure ongoing reliability in real-world applications.

**Explain Source of Drift** Beyond detection, the ideal algorithm should offer insights into the origin or cause of the detected drift. This explanatory power is essential in industrial contexts where understanding the reason behind changes can lead to more informed decision-making and preventive actions.

**Describe Future Development of Drift** The method should aim to not only detect and react to drift but also to forecast potential system evolution, offering insights into trends or recurrent patterns within the drift.

**Compare Different Types of Drifts** The ability to classify and compare different types of drift against a database of previous system changes can facilitate faster or more appropriate real world reactions.

**Model-Agnostic** The method should be model-agnostic, meaning it can work with various underlying learning models without requiring extensive reconfiguration. This quality ensures that the method is versatile and can be integrated into different systems with minimal adjustment and still use the

available state of the art model that is best suited to represent the system in a stable state.

**Non-Binary Detection** Unlike simplistic binary detection systems, the ideal method should offer a nuanced perspective on drift, identifying degrees or types of change instead of a binary drift/no-drift outcome. This approach provides more granular information that can inform a tailored adaptation strategy.

**Limited Model Size** For long-term deployment in industrial settings, the model should maintain a manageable size, balancing adaptability with resource efficiency. Ensuring a compact model size facilitates deployment in resource-constrained environments, such as embedded systems or edge devices, without sacrificing performance.

## 6.2 Vision

One possible framework that is based on the drift velocity notion of [3] is the following:

Given a system that has a set of possible system configurations  $\mathbb{S}$  where a system configuration  $S \in \mathbb{S}$  describes a mapping  $S : X^n \rightarrow Y^m$  with the space of observable system variables  $X$  (feature space) and the space of target values  $Y$  (target space). We propose an approach that is able to infer (or at least approximate) the current configuration of the system in a predefined time interval based on an initial configuration  $S_1$ , which is present for time  $t \in [0, u] \subset \mathbb{R}_{\geq 0}$ . The system then drifts to system configuration  $S_2$  which is present for  $t \in [u + \Delta, \infty]$ . Figure 6.1 depicts an example of one such drifted system, where the system configuration drifts from  $S_1$  in blue to  $S_2$  in green, while the system configuration in the interval  $t \in [4, 6]$  is unknown.

We further propose to infer a transition function  $T : \mathbb{S} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{S}$  that outputs the configuration of the system given a point in time and an initial system configuration. For the depicted example, this relates to  $T(S_1, 0) = S_1$  and  $T(S_1, \Delta) = S_2$ , as shown in the exemplary interpolation of Figure 6.1(b). In addition to these requirements,  $T$  should also accurately describe the data observed during the transition time  $t \in (u, u + \Delta)$ .

If the described method succeeds and is runtime proficient for drift adaption applications, further development could be feasible, in which the whole learning problem could be modeled by this transition description. For this approach,  $T$  would be inferred using all available data with the objective of propagating the initial configuration of the system  $S_1$  through multiple drifts. This would directly include the time domain as a part of the feature space, transforming any type of drift into a learnable system dynamic.

The realization of  $\mathbb{S}$  and  $T$  is as of now uncertain and different possibilities are available. When keeping with the basis of [3]  $\mathbb{S}$  could be the space of all joint probability distributions over  $\mathbb{D}$  while a transition function would transfer one such probability distribution into another. This would express the velocity profiles of [3] as an applicable mapping to describe, model and analyze the drifts in non-stationary environments, while also allowing the use to generate training data for a faster adaption process of the predictive model.

## References

- [1] Jan Niklas Adams et al. “A Framework for Explainable Concept Drift Detection in Process Mining”. In: *Business Process Management* 12875 (2021). Ed. by Artem Polyvyanyy et al., pp. 400–416. DOI: 10.1007/978-3-030-85469-0\_25. URL: [https://link.springer.com/10.1007/978-3-030-85469-0\\_25](https://link.springer.com/10.1007/978-3-030-85469-0_25) (visited on 08/28/2024).
- [2] Adriana Sayuri Iwashita et al. “An Overview on Concept Drift Learning”. In: *IEEE Access* 7 (Jan. 1, 2019), pp. 1532–1547. DOI: 10.1109/access.2018.2886026.
- [3] C. C. Aggarwal. “On Change Diagnosis in Evolving Data Streams”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.5 (Jan. 1, 2005), pp. 587–600. ISSN: 1041-4347. DOI: 10.1109/TKDE.2005.78.
- [4] Charu C. Aggarwal. “On Biased Reservoir Sampling in the Presence of Stream Evolution”. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*. VLDB '06. Seoul, Korea: VLDB Endowment, Sept. 1, 2006, pp. 607–618.

- [5] C. Alippi and M. Roveri. “An Adaptive CUSUM-based Test for Signal Change Detection”. In: *2006 IEEE International Symposium on Circuits and Systems*. 2006 IEEE International Symposium on Circuits and Systems. May 2006, 4 pp.-. DOI: 10.1109/ISCAS.2006.1693942. URL: <https://ieeexplore.ieee.org/document/1693942> (visited on 09/23/2024).
- [6] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. “A Hierarchical, Nonparametric, Sequential Change-Detection Test”. In: *The 2011 International Joint Conference on Neural Networks*. The 2011 International Joint Conference on Neural Networks. July 2011, pp. 2889–2896. DOI: 10.1109/IJCNN.2011.6033600. URL: <https://ieeexplore.ieee.org/document/6033600> (visited on 09/18/2024).
- [7] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. “A Just-in-Time Adaptive Classification System Based on the Intersection of Confidence Intervals Rule”. In: *Neural Networks. Artificial Neural Networks: Selected Papers from ICANN 2010 24.8* (Oct. 1, 2011), pp. 791–800. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2011.05.012. URL: <https://www.sciencedirect.com/science/article/pii/S0893608011001547> (visited on 09/10/2024).
- [8] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. “An Effective Just-in-Time Adaptive Classifier for Gradual Concept Drifts”. In: *The 2011 International Joint Conference on Neural Networks*. The 2011 International Joint Conference on Neural Networks. July 2011, pp. 1675–1682. DOI: 10.1109/IJCNN.2011.6033426. URL: <https://ieeexplore.ieee.org/document/6033426> (visited on 09/18/2024).
- [9] Cesare Alippi, Giacomo Boracchi, and Manuel Roveri. “Just-In-Time Classifiers for Recurrent Concepts”. In: *IEEE Transactions on Neural Networks and Learning Systems* 24.4 (Apr. 2013), pp. 620–634. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2013.2239309. URL: <https://ieeexplore.ieee.org/abstract/document/6425489> (visited on 09/10/2024).
- [10] Cesare Alippi and Manuel Roveri. “Just-in-Time Adaptive Classifiers-Part I: Detecting Nonstationary Changes”. In: *IEEE Transactions on*

- Neural Networks* 19.7 (July 2008), pp. 1145–1153. ISSN: 1941-0093. DOI: 10.1109/TNN.2008.2000082. URL: <https://ieeexplore.ieee.org/document/4470009> (visited on 09/10/2024).
- [11] Cesare Alippi and Manuel Roveri. “Just-in-Time Adaptive Classifiers-Part II: Designing the Classifier”. In: *IEEE Transactions on Neural Networks* 19.12 (Dec. 2008), pp. 2053–2064. ISSN: 1941-0093. DOI: 10.1109/TNN.2008.2003998. URL: <https://ieeexplore.ieee.org/document/4682649> (visited on 09/23/2024).
- [12] Manuel Baena-Garc et al. “Early Drift Detection Method”. In: 2005. URL: <https://www.semanticscholar.org/paper/Early-Drift-Detection-Method-Baena-Garc-Avila/2747577a61c70bc3874380130615e15aff76339e> (visited on 09/09/2024).
- [13] Jean Paul Barddal et al. “A Survey on Feature Drift Adaptation: Definition, Benchmark, Challenges and Future Directions”. In: *Journal of Systems and Software* 127 (May 1, 2017), pp. 278–294. ISSN: 0164-1212. DOI: 10.1016/j.jss.2016.07.005. URL: <https://www.sciencedirect.com/science/article/pii/S0164121216301030> (visited on 09/02/2024).
- [14] Firas Bayram, Bestoun S. Ahmed, and Andreas Kassler. “From Concept Drift to Model Degradation: An Overview on Performance-Aware Drift Detectors”. In: *Knowledge-Based Systems* 245 (June 7, 2022), p. 108632. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2022.108632. URL: <https://www.sciencedirect.com/science/article/pii/S0950705122002854> (visited on 09/05/2024).
- [15] Albert Bifet and Ricard Gavaldà. “Adaptive Learning from Evolving Data Streams”. In: *Advances in Intelligent Data Analysis VIII*. Ed. by Niall M. Adams et al. Berlin, Heidelberg: Springer, 2009, pp. 249–260. ISBN: 978-3-642-03915-7. DOI: 10.1007/978-3-642-03915-7\_22.
- [16] Albert Bifet and Ricard Gavaldà. “Learning from Time-Changing Data with Adaptive Windowing”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, Apr. 26, 2007, pp. 443–448. ISBN: 978-0-89871-630-6. DOI: 10.1137/1.9781611972771.42. URL: <https://>

- //epubs.siam.org/doi/10.1137/1.9781611972771.42 (visited on 09/12/2024).
- [17] Albert Bifet et al. “New Ensemble Methods for Evolving Data Streams”. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. New York, NY, USA: Association for Computing Machinery, June 28, 2009, pp. 139–148. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557041. URL: <https://dl.acm.org/doi/10.1145/1557019.1557041> (visited on 09/18/2024).
- [18] Hanen Borchani et al. “Modeling Concept Drift: A Probabilistic Graphical Model Based Approach”. In: *Advances in Intelligent Data Analysis XIV: 14th International Symposium, IDA 2015, Saint Etienne, France, October 22-24, 2015, Proceedings*. Vol. 9385. Lecture Notes in Computer Science. Cham: Springer International Publishing, Jan. 1, 2015, pp. 72–83. ISBN: 978-3-319-24465-5. DOI: 10.1007/978-3-319-24465-5\_7. URL: [https://link.springer.com/chapter/10.1007/978-3-319-24465-5\\_7](https://link.springer.com/chapter/10.1007/978-3-319-24465-5_7).
- [19] Dariusz Brzezinski and Jerzy Stefanowski. “Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm”. In: *IEEE Transactions on Neural Networks and Learning Systems* 25.1 (Jan. 1, 2014), pp. 81–94. ISSN: 2162-237X. DOI: 10.1109/tnnls.2013.2251352. PMID: 24806646.
- [20] Lior Cohen et al. “Info-Fuzzy Algorithms for Mining Dynamic Data Streams”. In: *Applied Soft Computing*. Soft Computing for Dynamic Data Mining 8.4 (Sept. 1, 2008), pp. 1283–1294. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2007.11.003. URL: <https://www.sciencedirect.com/science/article/pii/S156849460800046X> (visited on 09/18/2024).
- [21] Gregory Ditzler and Robi Polikar. “Hellinger Distance Based Drift Detection for Nonstationary Environments”. In: *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. 2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE). Apr. 2011, pp. 41–48.

- DOI: 10.1109/CIDUE.2011.5948491. URL: <https://ieeexplore.ieee.org/abstract/document/5948491> (visited on 09/11/2024).
- [22] Gregory Ditzler, Gail Rosen, and Robi Polikar. “Discounted Expert Weighting for Concept Drift”. In: *2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. 2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE). Apr. 2013, pp. 61–67. DOI: 10.1109/CIDUE.2013.6595773. URL: <https://ieeexplore.ieee.org/document/6595773> (visited on 09/18/2024).
- [23] Gregory Ditzler et al. “Learning in Nonstationary Environments: A Survey”. In: *IEEE Computational Intelligence Magazine* 10.4 (Jan. 1, 2015), pp. 12–25. ISSN: 1556-603X. DOI: 10.1109/mci.2015.2471196.
- [24] Ryan Elwell and Robi Polikar. “Incremental Learning in Nonstationary Environments with Controlled Forgetting”. In: *2009 International Joint Conference on Neural Networks*. 2009 International Joint Conference on Neural Networks. June 2009, pp. 771–778. DOI: 10.1109/IJCNN.2009.5178779. URL: <https://ieeexplore.ieee.org/document/5178779> (visited on 09/24/2024).
- [25] Ryan Elwell and Robi Polikar. “Incremental Learning of Concept Drift in Nonstationary Environments”. In: *IEEE Transactions on Neural Networks* 22.10 (Oct. 2011), pp. 1517–1531. ISSN: 1941-0093. DOI: 10.1109/TNN.2011.2160459. URL: <https://ieeexplore.ieee.org/document/5975223> (visited on 09/20/2024).
- [26] Conor Fahy, Shengxiang Yang, and Mario Gongora. “Scarcity of Labels in Non-Stationary Data Streams: A Survey”. In: *ACM Comput. Surv.* 55.2 (Jan. 21, 2022), 40:1–40:39. ISSN: 0360-0300. DOI: 10.1145/3494832. URL: <https://dl.acm.org/doi/10.1145/3494832> (visited on 08/30/2024).
- [27] Isvani Frías-Blanco et al. “Online and Non-Parametric Drift Detection Methods Based on Hoeffding’s Bounds”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.3 (Mar. 2015), pp. 810–823. ISSN: 1558-2191. DOI: 10.1109/TKDE.2014.2345382. URL: <https://ieeexplore.ieee.org/document/6871418> (visited on 09/18/2024).

- [28] João Gama and Gladys Castillo. “Learning with Local Drift Detection”. In: *Advanced Data Mining and Applications*. Ed. by Xue Li, Osmar R. Zaiane, and Zhanhuai Li. Vol. 4093. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 42–55. ISBN: 978-3-540-37025-3 978-3-540-37026-0. DOI: 10.1007/11811305\_4. URL: [http://link.springer.com/10.1007/11811305\\_4](http://link.springer.com/10.1007/11811305_4) (visited on 09/09/2024).
- [29] João Gama, Ricardo Fernandes, and Ricardo Rocha. “Decision Trees for Mining Data Streams”. In: *Intelligent Data Analysis 10.1* (Jan. 1, 2006), pp. 23–45. ISSN: 1088-467X. DOI: 10.3233/IDA-2006-10103. URL: <https://content.iospress.com/articles/intelligent-data-analysis/ida00234> (visited on 09/18/2024).
- [30] João Gama et al. “A Survey on Concept Drift Adaptation”. In: *ACM Computing Surveys* 46.4 (Jan. 1, 2014), pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/2523813.
- [31] Maayan Harel et al. “Concept Drift Detection Through Resampling”. In: *Proceedings of the 31st International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, June 18, 2014, pp. 1009–1017. URL: <https://proceedings.mlr.press/v32/harel14.html> (visited on 09/18/2024).
- [32] Douglas M. Hawkins, Peihua Qiu, and Chang Wook Kang. “The Change-point Model for Statistical Process Control”. In: *Journal of Quality Technology* 35.4 (Oct. 1, 2003), pp. 355–366. ISSN: 0022-4065. DOI: 10.1080/00224065.2003.11980233. URL: <https://doi.org/10.1080/00224065.2003.11980233> (visited on 09/18/2024).
- [33] Fabian Hinder, Johannes Kummert, and Barbara Hammer. “Explaining Concept Drift by Mean of Direction”. In: *Artificial Neural Networks and Machine Learning – ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I*. Ed. by Igor Farkaš, Paolo Masulli, and Stefan Wermter. Vol. 12396. Springer eBook Collection. Cham: Springer International Publishing; Imprint Springer, Jan. 1, 2020, pp. 379–390. ISBN: 978-3-030-61609-0. DOI: 10.1007/978-3-030-61609-0\_30.



URL: [https://link.springer.com/chapter/10.1007/978-3-030-61609-0\\_30](https://link.springer.com/chapter/10.1007/978-3-030-61609-0_30).

- [34] Fabian Hinder et al. “Model-Based Explanations of Concept Drift”. In: *Neurocomputing* 555 (Jan. 1, 2023), p. 126640. ISSN: 09252312. DOI: 10.1016/j.neucom.2023.126640.
- [35] Geoff Hulten, Laurie Spencer, and Pedro Domingos. “Mining Time-Changing Data Streams”. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '01*. New York, NY, USA: Association for Computing Machinery, Aug. 26, 2001, pp. 97–106. ISBN: 978-1-58113-391-2. DOI: 10.1145/502512.502529. URL: <https://dl.acm.org/doi/10.1145/502512.502529> (visited on 09/18/2024).
- [36] Elena Ikonomovska, João Gama, and Sašo Džeroski. “Incremental Multi-Target Model Trees for Data Streams”. In: *Proceedings of the 2011 ACM Symposium on Applied Computing. SAC '11*. New York, NY, USA: Association for Computing Machinery, Mar. 21, 2011, pp. 988–993. ISBN: 978-1-4503-0113-8. DOI: 10.1145/1982185.1982402. URL: <https://dl.acm.org/doi/10.1145/1982185.1982402> (visited on 09/18/2024).
- [37] Elena Ikonomovska, João Gama, and Sašo Džeroski. “Learning Model Trees from Evolving Data Streams”. In: *Data Mining and Knowledge Discovery* 23.1 (July 1, 2011), pp. 128–168. ISSN: 1573-756X. DOI: 10.1007/s10618-010-0201-y. URL: <https://doi.org/10.1007/s10618-010-0201-y> (visited on 09/18/2024).
- [38] Elena Ikonomovska, João Gama, and Sašo Džeroski. “Online Tree-Based Ensembles and Option Trees for Regression on Evolving Data Streams”. In: *Neurocomputing. Special Issue on Information Processing and Machine Learning for Applications of Engineering* 150 (Feb. 20, 2015), pp. 458–470. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2014.04.076. URL: <https://www.sciencedirect.com/science/article/pii/S0925231214012338> (visited on 09/18/2024).

- [39] Elena Ikonomovska et al. “Regression Trees from Data Streams with Drift Detection”. In: *Proceedings of the 12th International Conference on Discovery Science*. DS '09. Berlin, Heidelberg: Springer-Verlag, Oct. 7, 2009, pp. 121–135. ISBN: 978-3-642-04746-6. DOI: 10.1007/978-3-642-04747-3\_12. URL: [https://doi.org/10.1007/978-3-642-04747-3\\_12](https://doi.org/10.1007/978-3-642-04747-3_12) (visited on 09/18/2024).
- [40] Elena Ikonomovska et al. “Speeding up Hoeffding-Based Regression Trees with Options”. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Madison, WI, USA: Omnipress, June 28, 2011, pp. 537–544. ISBN: 978-1-4503-0619-5.
- [41] Ziqiu Kang, Cagatay Catal, and Bedir Tekinerdogan. “Machine Learning Applications in Production Lines: A Systematic Literature Review”. In: *Computers & Industrial Engineering* 149 (Nov. 1, 2020), p. 106773. ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106773. URL: <https://www.sciencedirect.com/science/article/pii/S036083522030485X> (visited on 11/04/2024).
- [42] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. “Detecting Change in Data Streams”. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*. VLDB '04. Toronto, Canada: VLDB Endowment, Aug. 31, 2004, pp. 180–191. ISBN: 978-0-12-088469-8.
- [43] Ralf Klinkenberg and Thorsten Joachims. “Detecting Concept Drift with Support Vector Machines”. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., June 29, 2000, pp. 487–494. ISBN: 978-1-55860-707-1.
- [44] Ivan Koychev. “Gradual Forgetting for Adaptation to Concept Drift”. In: ().
- [45] Timothée Lesort, Massimo Caccia, and I. Rish. “Understanding Continual Learning Settings with Data Distribution Drift Analysis”. In: *ArXiv* (Apr. 4, 2021). URL: <https://www.semanticscholar.org/paper/Understanding-Continual-Learning-Settings-with-Data->

- Lesort-Caccia/228a0098d66c0df8532ea37027e3964d35f1030e (visited on 08/28/2024).
- [46] Jing Liu, Xue Li, and Weicai Zhong. “Ambiguous Decision Trees for Mining Concept-Drifting Data Streams”. In: *Pattern Recognition Letters* 30.15 (Nov. 1, 2009), pp. 1347–1355. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2009.07.017. URL: <https://www.sciencedirect.com/science/article/pii/S0167865509001950> (visited on 09/18/2024).
- [47] Jie Lu et al. “Learning under Concept Drift: A Review”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.12 (Dec. 2019), pp. 2346–2363. ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2876857. URL: <https://ieeexplore.ieee.org/document/8496795> (visited on 09/04/2024).
- [48] Meng Han et al. “A Survey of Active and Passive Concept Drift Handling Methods”. In: *Computational Intelligence* (Apr. 10, 2022). DOI: 10.1111/coin.12520.
- [49] Kyosuke Nishida and Koichiro Yamauchi. “Learning, Detecting, Understanding, and Predicting Concept Changes”. In: *2009 International Joint Conference on Neural Networks*. 2009 International Joint Conference on Neural Networks. June 2009, pp. 2280–2287. DOI: 10.1109/IJCNN.2009.5178619. URL: <https://ieeexplore.ieee.org/document/5178619> (visited on 09/25/2024).
- [50] Jan Peter Patist. “Optimal Window Change Detection”. In: *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*. Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). Oct. 2007, pp. 557–562. DOI: 10.1109/ICDMW.2007.9. URL: <https://ieeexplore.ieee.org/document/4476722> (visited on 09/12/2024).
- [51] Russel Pears, Sripirakas Sakthithasan, and Yun Sing Koh. “Detecting Concept Change in Dynamic Data Streams”. In: *Machine Learning* 97.3 (Dec. 1, 2014), pp. 259–293. ISSN: 1573-0565. DOI: 10.1007/s10994-013-5433-9. URL: <https://doi.org/10.1007/s10994-013-5433-9> (visited on 09/18/2024).

- [52] Roberto Souto Maior de Barros et al. “A Large-Scale Comparison of Concept Drift Detectors”. In: *Information Sciences* (July 1, 2018), pp. 348–370. DOI: 10.1016/j.ins.2018.04.014.
- [53] Gordon J. Ross, Dimitris K. Tasoulis, and Niall M. Adams. “Nonparametric Monitoring of Data Streams for Changes in Location and Scale”. In: *Technometrics* 53.4 (Nov. 1, 2011), pp. 379–389. ISSN: 0040-1706. DOI: 10.1198/TECH.2011.10069. URL: <https://doi.org/10.1198/TECH.2011.10069> (visited on 09/18/2024).
- [54] Gordon J. Ross et al. “Exponentially Weighted Moving Average Charts for Detecting Concept Drift”. In: *Pattern Recognition Letters* 33.2 (Jan. 15, 2012), pp. 191–198. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2011.08.019. URL: <https://www.sciencedirect.com/science/article/pii/S0167865511002704> (visited on 09/09/2024).
- [55] Sripirakas Sakthithasan, Russel Pears, and Yun Sing Koh. “One Pass Concept Change Detection for Data Streams”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by Jian Pei et al. Berlin, Heidelberg: Springer, 2013, pp. 461–472. ISBN: 978-3-642-37456-2. DOI: 10.1007/978-3-642-37456-2\_39.
- [56] Denise Maria Vecino Sato et al. “A Survey on Concept Drift in Process Mining”. In: *ACM Comput. Surv.* 54.9 (Oct. 8, 2021), 189:1–189:38. ISSN: 0360-0300. DOI: 10.1145/3472752. URL: <https://dl.acm.org/doi/10.1145/3472752> (visited on 08/30/2024).
- [57] Bai Su, Yi-Dong Shen, and Wei Xu. “Modeling Concept Drift from the Perspective of Classifiers”. In: *2008 IEEE Conference on Cybernetics and Intelligent Systems: September 21-24, 2008, Chengdu, China*. [Piscataway, N.J.]: IEEE, Jan. 1, 2008, pp. 1055–1060. ISBN: 978-1-4244-1673-8. DOI: 10.1109/ICCIS.2008.4670840.
- [58] Andrés L. Suárez-Cetrulo, David Quintana, and Alejandro Cervantes. “A Survey on Machine Learning for Recurring Concept Drifting Data Streams”. In: *Expert Systems with Applications* 213 (Mar. 1, 2023), p. 118934. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2022.118934. URL: <https://www.sciencedirect.com/science/article/pii/S0957417422019522> (visited on 11/04/2024).

- [59] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. “Covariate Shift Adaptation by Importance Weighted Cross Validation”. In: *Journal of Machine Learning Research* 8.35 (2007), pp. 985–1005. ISSN: 1533-7928. URL: <http://jmlr.org/papers/v8/sugiyama07a.html> (visited on 09/18/2024).
- [60] “Towards a Transition Matrix-Based Concept Drift Approach”. In.
- [61] Alexey Tsymbal et al. “Dynamic Integration of Classifiers for Handling Concept Drift”. In: *Information Fusion*. Special Issue on Applications of Ensemble Methods 9.1 (Jan. 1, 2008), pp. 56–68. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2006.11.002. URL: <https://www.sciencedirect.com/science/article/pii/S1566253506001138> (visited on 09/18/2024).
- [62] Pingfan Wang et al. “QuadCDD: A Quadruple-based Approach for Understanding Concept Drift in Data Streams”. In: *Expert Systems with Applications* 238 (Jan. 1, 2024), p. 122114. ISSN: 09574174. DOI: 10.1016/j.eswa.2023.122114.
- [63] Geoffrey I. Webb et al. “Characterizing Concept Drift”. In: *Data Mining and Knowledge Discovery* 30.4 (Jan. 1, 2016), pp. 964–994. ISSN: 1384-5810. DOI: 10.1007/s10618-015-0448-4. URL: <https://link.springer.com/article/10.1007/s10618-015-0448-4>.
- [64] Geoffrey I. Webb et al. *Understanding Concept Drift*. Jan. 1, 2017. DOI: 10.48550/arXiv.1704.00362. Pre-published.
- [65] Gerhard Widmer and Miroslav Kubat. “Learning in the Presence of Concept Drift and Hidden Contexts”. In: *Machine Learning* 23.1 (Apr. 1, 1996), pp. 69–101. ISSN: 1573-0565. DOI: 10.1007/BF00116900. URL: <https://doi.org/10.1007/BF00116900> (visited on 09/18/2024).
- [66] Yibin Ye, Stefano Squartini, and Francesco Piazza. “Online Sequential Extreme Learning Machine in Nonstationary Environments”. In: *Neurocomputing*. Advanced Theory and Methodology in Intelligent Computing 116 (Sept. 20, 2013), pp. 94–101. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2011.12.064. URL: <https://www.sciencedirect.com/science/article/pii/S092523121200728X> (visited on 09/18/2024).

**Table 3.1:** Table of used data stream handling methods. **Tasks** Det: Change Detection ; A: Change Adaption; A\* Change Adaption based on deleting old data; Par: Learning Paradigma; U: Drift Understanding/ System Analysis. **Policies** Ac: Active Learner; Pa: Passive Learner, C: Classification; C\*: Classification with additional assumptions; R: Regression, D: Distribution Based Detection/Adaption; P: Performance based Detection/Adaption

Algorithm	Task	Policy	Reference  Related
DDM	Det, A*	Ac, C, P	[28][27]
EDDM	Det, A*	Ac, C, P	[12]
ECDD	Det, A*	Ac, C, P	[54]
CI-CUSUM (JIT)	Det, A	Ac, C, D	[10, 11][9, 7, 8]
CHANGE	Det, A*	Ac, C, R, D	[42]
HDDDM	Det, A*	Ac, C, R, D	[21]
ADWIN	Det, A	Ac, C*, D	[16][50, 15, 55, 51]
Hierachical CD	Det, A*	Ac, C, R, D	[6]
CVFDT	A	Pa, C, P	[35][46, 17, 20]
FIMT-DD	A	Ac, R, P	[37, 39][38, 36, 40]
VFDTc	A	Ac, C, D	[29]
Adap. CUSUM	Det, A*	Ac, C, R, D	[5]
PERM	Det, A*	Ac, C, R, P	[31]
IWCV	Par	C	[59][4]
SVM-ADWIN	A	Pa, C, R, P	[43]
Learn++.NSE	A	Pa, C, R, P	[25][22, 61, 24, 19]
Forgetting	Par	Pa, C, R	[44]
FLORA	A	Ac, C, P	[65]
LDUP	A, U	Ac, C, P	[49]
CPM	Det	Ac, R, D	[53]
OS ELM TV	A	Pa, R	[66]
UPCP	Det	Ac, D	[32]

# **Registration as a warping method for optotypes towards human vision**

*Oliver Veitl*

Bayrische Motoren Werke AG  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
oliver.ov.veitl@bmw.de

## **Abstract**

The central question guiding this research is whether it is possible to adjust the projection within the field of view (FoV) of the human vision in Head-Up Displays (HUD) towards zero deviations from a design pattern. This paper investigates a method to analyse the quantitative characteristics of a spatial Head-Up Display signal, with the aim of determining whether a best practice of projecting alignment based on a phase-shifting algorithm has any deviations. The analysed signal was derived within a one-factor-at-a-time experimental design for an optimization. Despite the research demonstrating the amount of information that can be extracted from the signal, further questions must be addressed. These include the ability of a human vision system to perceive the residuals of the deviations and the correct scaling of the obtained information by the transformed signal. A possible solution for the calibration and adjustment process is outlined, but further assessment and quantification of the process is required to ensure optimal results.

# 1 Introduction

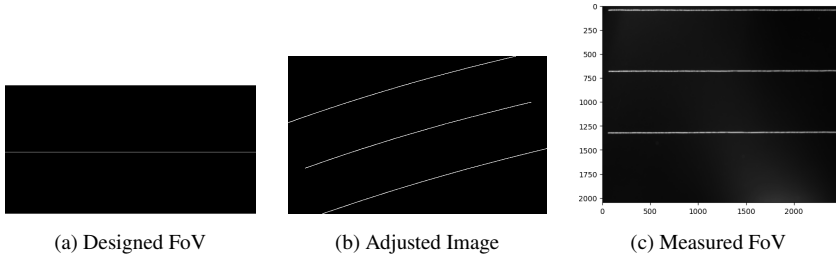
Quality is defined as the ability of a product to satisfy customer needs [2]. However, the quality of images remains ambiguous, possibly due to the distinctiveness of painted pictures or the subjective assessment of images by humans. The challenges encountered in Human-Machine Interfaces (HMI) in Head-Up Displays (HUD) are analogous. The observer's assessment of images is a key factor in this context. The initial inquiry that arises from this statement pertains to the capabilities of the product in terms of quality and the needs that a head-up display is expected to meet in order to ensure customer satisfaction. Based on the findings of Wagner Daniel, these answers are not clarified [6]. In this study, the focus will be on the capabilities of the product. To summarise the current lack of knowledge, there is no possibility for a determination of the minimal deviations in calibrated and warped optotypes of head-up display images. The method of this research is to conduct more concrete investigations on different calibration methods for image data to adjust displays in optical systems and identify the optimal image compared to a designed one. The results should be an optimum warping parameter set in comparable optical setups, meaning that the deviation of the system image compared to the desired images is zero, not perceptible or even not measurable. The following sections are structured as follows: Section 2 discusses the different calibration methods and the one that is investigated, registration based on phase shifting computation. The design of experiments is explained in Section 3, based on the models of Section 2. The results are presented in Section 4, and the outlook is presented in Section 5. In every section measurement based images show horizontal and vertical scales. Whereas, designed images show now scales.

## 2 State of the Art - Image Processing for Human-Machine-Interaction in HUD-Projections

In this proposal it should be clarified whether the registration of observer and imaging pixel in an optical setup based on the phase shifting results is capable of defining an adjustment standard for head-up display images. To derive the



state of the art and the basic for registration data compared to other methods, three different images are introduced.



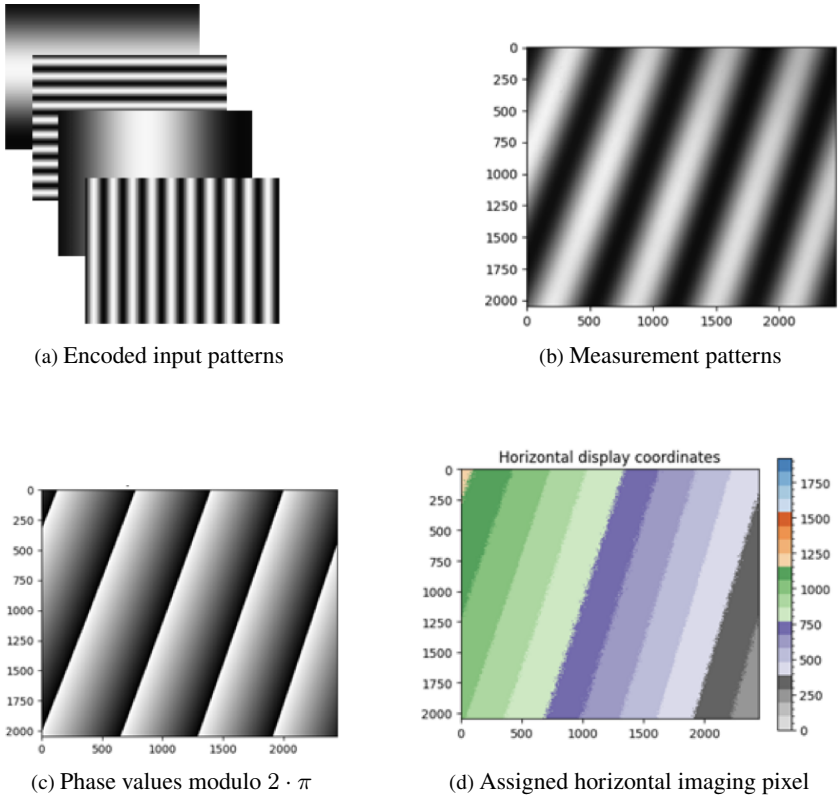
**Figure 2.1:** Image processing steps towards an adjusted field-of-view

The figures in Figure 2.1 demonstrate the process from the image design towards the actual system output of a head-up display. The human interface shall percept three horizontal lines distributed over its field of view which is represented by the designed output, as seen in Figure 2.1(a). Therefore and based on the knowledge of the optical system the machine in this case the projector is calibrated in an appropriate way towards an optimised input as seen in Figure 2.1(b). The comparison of the initial image with the measured output in Figure 2.1(c) leads to our initial issue. The adjusted output shows deviations, for example a lack of sharpness, brightness loss and some geometrical differences. This process is standardised within the automotive industry, the SAE suggests a calibration based on the known geometry to compensate the system design. Additionally, it is possible to fine grain the system with simulations by elaborating the local deviations. [4] Another possibility is to apply image processing algorithms. One proposed method shows predefined patterns in boundary positions to compute the mean position of the projection. [7] Another approach is to use virtual reproduced systems to calculate the calibrated input for the HUD projector. This generated input has multiple degrees of freedom, such as height or rotation, which can be fine-tune by the specific user towards his preferences. One of the latest techniques to computed such an input is based on patterns, which are positioned in front of the vehicle, see [3]. We propose a new process to calibrate and adjust

head-up display images based on the optical system without prior knowledge. The core is about to compute the assignment of the observer sensing element and the image emitting point. The only relevant prior knowledge is the optical system itself. As far as we know it is possible to compute the difference between two images based on feature extraction which is limited to small movements. Comparatively, is the computation of the optical flow, which is an algorithm that is limited to the used measurement data. For the case one uses unwrapped phasemaps based on the phase-shifting methods the optical flow may work, but it still needs the knowledge of the different system components towards each other to assign the adjusted imaging points. Based on this knowledge we suggest a measurement of phase-shifting data and a processing with spatio-temporal unwrapping methods. This makes it possible to assign every imaging point a unique value and assign the two system without any additional information.

## 2.1 Registration - a Phase-Shifting Method

As phase-shifting techniques are well-documented, see [1] [8], this chapter will simply replicate the process and include the most significant equations to derive the mathematical model. Figure 2.2 illustrates the calibration process that is proposed. Firstly, it is necessary to define the input patterns that are compared to the obtained output, as illustrated in Figure 2.2(a). In order to register the data and utilise a higher sensitivity of the method, a second spatial wavelength  $\lambda_n$  is required. The spatial wavelength describes how often the sine-distributed grey values from black to white are repeated along one image axis. The definition of a position in a specific plane necessitates two components. Therefore the patterns get shifted in horizontal the first component and vertical direction the second one. The four different patterns shown in Figure 2.2(a) are to be shifted by  $n_{Steps}$ , what can be described in the following equation 2.1, see [1].



**Figure 2.2:** Steps for the calibration process based on phase shifting methods

$$g_s(x, y) = \beta + \alpha \cdot \sin(\omega \cdot \varphi(x, y) + \varphi_0) \quad (2.1)$$

$$\varphi_0 = 2\pi \cdot \frac{k}{n_{steps}} \quad (2.2)$$

$$k \in 0, \dots, n_{steps} - 1 \quad (2.3)$$

These patterns are observed by a specimen, in this case by the mirroring element that is being compensated for. The observation process transforms the encoded pattern into the measurement pattern illustrated in Figure 2.2(b). The data can be defined by equation 2.4, with the addition of a bias from the surrounding environment  $b_S(x, y)$  and a noise factor  $\delta g$ , see [1].

$$g_K(x, y) = a(x, y) \cdot g_S(x, y) + b_S(x, y) + \delta g \quad (2.4)$$

The greyscales of the pattern are influenced by the modulation term  $a(x, y)$ . This is evaluated within the phase-shifting characteristics, as will be demonstrated subsequently. Following the measurement of  $N_{patterns} = 2 \cdot 2 \cdot n_{steps}$ , the number of deformed patterns, these are wrapped towards modulo  $2 \cdot \pi$ . An example of the by equation 2.5 processed data is shown in Figure 2.2(d) see [8].

$$\varphi_i(x, y) = \arctan\left(\frac{g_{K4}(x, y) - g_{K2}(x, y)}{g_{K1}(x, y) - g_{K3}(x, y)}\right) \quad (2.5)$$

$$\Phi_i(x, y) = \frac{\lambda_{fine}}{\lambda_{coarse}} \cdot \varphi_i(x, y) \quad (2.6)$$

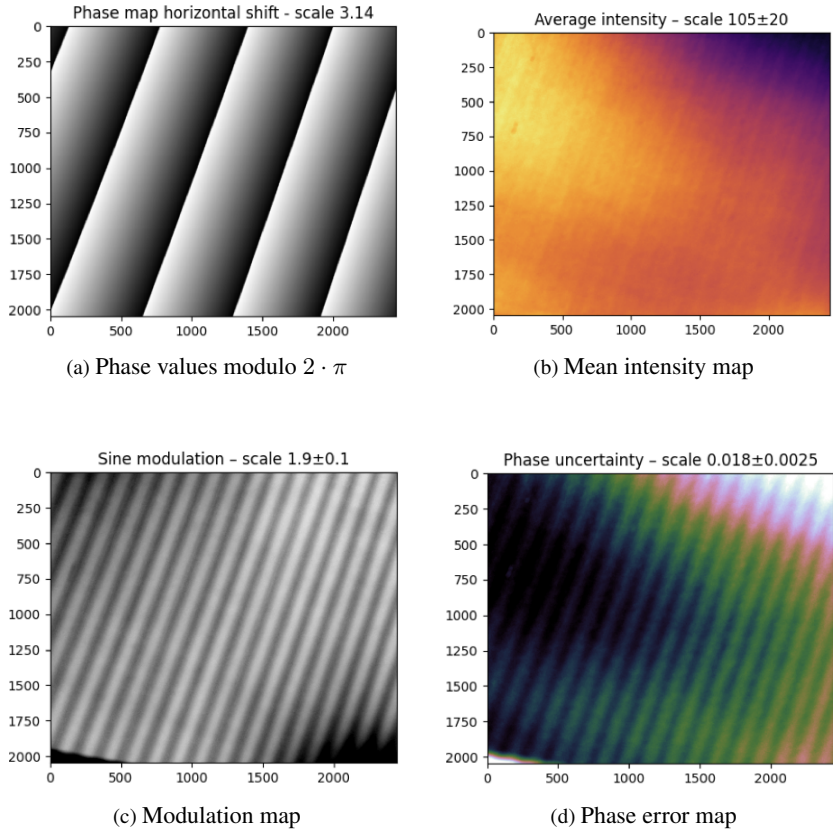
The unwrapping of the modulo  $2 \cdot \pi$  towards a horizontal and vertical phase map, characterised by enhanced sensitivity based on their wavelength relation with the coarse phase map, is demonstrated in equation 2.6. Finally, this unwrapped phase map  $\Phi_i(x, y)$  is scaled in accordance with the specific imaging source to elaborate the specific position in the horizontal and vertical directions, as illustrated in Figure 2.2(d). For a more detailed exposition of this subject, readers are directed to equation 2.6 and the relevant literature [5].

$$p(x, y) = \frac{\pi \cdot \lambda_{fine}}{2 \cdot \pi \cdot \lambda_{coarse} \cdot l_p} \cdot \Phi(x, y) \quad (2.7)$$

As it is not possible to perceive every change within the registration map computed by equation 2.7 in a subjective manner, some characteristics of phase-shifting methods will be utilised.

## 2.2 Characteristics of Phase-Shifting

The quality of phase-shifting is contingent upon the modulation of the measurement data.



**Figure 2.3:** Characteristics of phase measuring processes

Consequently, a perfect measurement would consist of an equal mean intensity map, this case is illustrated in Figure 2.3(b), situated at the midpoint of the

measurable greyscale range, and a modulation, the case of this measurement is depicted in Figure 2.3(c), utilising the entire range. The mean intensity of a measurement series can be trivially calculated, as outlined in equation 2.8, while the modulation is constructed on the intensity, as demonstrated in equation 2.9 [5].

$$g_M(x, y) = \frac{\sum_{i=1}^{n_{steps}} g_i}{n_{steps}} \quad (2.8)$$

$$\Gamma = \frac{\sqrt{(g_{K4} - g_{K2})^2 + (g_{K1} - g_{K3})^2}}{g_M(x, y)} \quad (2.9)$$

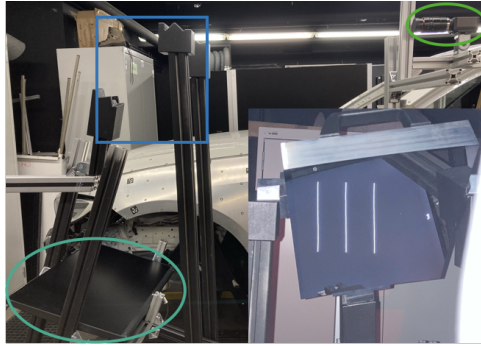
As demonstrated in Figure 2.3(d), which was derived using equation 2.10 [5], the mean intensity map serves as the computational foundation. The modulation map, on the other hand, is capable of highlighting potential substructures within the phase error map.

$$\Delta\varphi = \frac{\Delta I}{\sqrt{2} \cdot g_M(x, y) \cdot \Gamma} \quad (2.10)$$

This characteristic  $\Delta\varphi$  displays a map of potential deviations within the phase maps, which form the foundation of the registration maps depicted in Figure 2.2(d). The map utilises an additional imaging source depending factor  $\Delta I$ , which provides a statistical description of the greyscale noise.

### **3 Design of Experiments - Optimization of calibration process parameters**

This descriptive research has the objective to find an optimum for geometrical manipulation of image data in head-up display projections.



**Figure 3.1:** Test rag to adjust images, see upper left corner, on the display towards a mirroring geometry for specified observer, seen upper right corner.

Therefore a calibration is computed based on phase shifting data gained in the shown test rag of Figure 3.1. As illustrated, the optical configuration comprises a green-marked observer watching an image displayed in turquoise. The observation is achieved through the utilisation of an aspheric aluminium mirror, as depicted in the lower right corner. The mirror is positioned on the blue brackets, as can be seen in the upper left corner of the image. It is important to note that the surface of the mirror is not ideal and is not flat; therefore, a calibration is required to adjust the image on the display. This calibration process enables the observation of an optimal image over the mirror. The definition of an optimal image is determined by two distinct assessments. The first is determined by a subjective observer, while the second is determined by an objective observer. The present study focuses on the latter to develop best practice for geometrical deviations in head-up display projections. Therefore a one-factor-at-a-time experimental plan is used, to optimise every parameter of the mathematical model shown in section 2. This is possible because, the brightness controls the mean level, while the contrast defines the overall measurement range and the gamma correction is relevant for the linearity. The overall experimental process is structured, as follows:

**Table 3.1:** Calibration process for head-up display projections based on phase shifting and quality analysis.

Stage 1: N-Step-Phase-shifting	Stage 2: Calibration/ Adjustment	Stage 3: Objective Assessment
Wavelength 1 horizontal	Spatio-Temporal Unwrapping	Measurement adjusted image
Wavelength 1 vertical	Assignment observer/ display pixel	Deviation computation
Wavelength 2 horizontal	Adjustment display image	
Wavelength 2 vertical		

The process is subdivided into three stages: the phase-shifting stage, in which the calibration data is obtained; the computational stage, in which the warped or adjusted image is defined; and the assessment stage, in which the geometrical deviation is computed between the desired input and measured output. The first stage is primarily the acquisition of the measurement data, in which the luminance camera acquires  $n$  patterns in the first spatial wavelength in the horizontal direction. This step is repeated in the vertical direction, and subsequently, the final steps are repeated for the second spatial wavelength. The measurement phase is succeeded by the calibration phase. In this step, the second one, the wrapped phase maps are computed, unwrapped using a spatio-temporal approach [5], afterwards the absolute phase map is scaled towards a pixel-wise registered calibration map. This map enables a precise alignment of the display pixels with a given observer image. The result of this process is an adjusted display image based on the knowledge of the observed pixels, which is shown in the optical setup and captured via the mirror. This forms are part of the third process step. Finally, the quantification of the deviation between the given image from step two and the captured image in this step is achieved by normalisation and a



wavelet analysis. In this investigation, the process is repeated with its steps one to three for every relevant factor derived in section 2.

**Table 3.2:** Parameters to vary within the optimisation process regarding the display settings.

Parameter display $g_D(x, y)$						
Parameter	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Gamma						
Brighthness						
Contrast						

**Table 3.3:** Parameters to vary within the optimisation process regarding the pattern settings.

Parameter pattern $g_P(x, y)$						
Parameter	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Modulation						
Extrapolation						
Phase-shifting-method						

The factors investigated during the study are summarised in Table 3.2 and 3.3. Based on the knowledge the optical setup contains three different components, and adding the fact the mirror is non-changable. We see Tabel 3.2 is describing the imaging source whereas Table 3.3 shows the relevant parameters of the measuring element. It is important to note that the camera or its settings are optimized for each of the scenarios shown above. Gamma, brightness and contrast are directly link to the imaging source, in this case it is a liquid-cristal display. These settings modify the emitted greyscales within the measurement data, for the case the other parameters are fixed. Another reason for varying these parameters is the mathematical description of the measurement data by Equation 2.4 and its proportions. The subsequent Table 3.3 illustrates the

parameters of the measurement data. The measurement information is conveyed by the deformation of the emitted greyscale patterns. One parameter of Table 3.3 effect this is the modulation, one half of the peak-to-valley value from the measured greyscales. The phase-shifting-method is also greyscale related. It has an impact on the spatial movement of one greyscale value, without the deformation of the mirror. This highlights its influence on the sensitivity to recognize such deformations. The tables contain six varying steps. The overall range of light-relevant parameters is divided in at least five steps.

## **4 Results - Best Practice Horizontal Lines in Head-Up Display Projections**

The shown optical system in section 3 consists of a liquid crystal display, a luminance camera and an aspheric mirror. In this section investigation results and interpretation of the factors in Table 3.2 and 3.3 is presented, as it is the base for identifying the sweet spot, qualitative observations, the decision for the best practice by which the deviation is computed.

### **4.1 Identified Optimal Parameter Set**

Based on the physical and mathematical model from section 2, as well as the derived design of experiments from section 4 the optimal set of parameters is elaborated as seen in the table:

**Table 4.1:** Results of the one-factor-at-a-time optimisation based on the relevant factors.

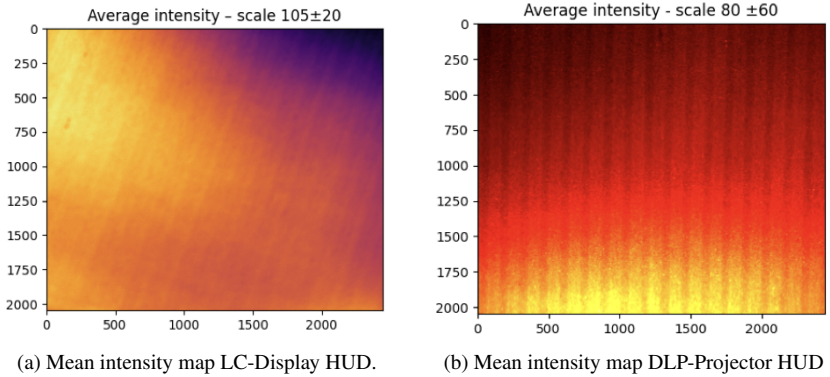
Parameter display $g_D(x, y)$						
Parameter	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Gamma I	0.3	0.8	1.3	1.8		
$\gamma$						
Gamma II	0.7	0.9	1.0	1.2	1.5	-
$\gamma_{Phase}$						
Gamma II	2.4	2.5	2.6	2.7	2.8	-
$\gamma_{Intensity}$						
Brighness I	0	20	40	60	80	100
Brighness II	40	42	45	47	50	-
Contrast I	25	50	75	100	-	-
Contrast II	55	60	65	70	-	-
Parameter pattern $g_P(x, y)$						
Modulation	25	50	75	100	-	-
Extrapolation	Global scalingvalues			Local scalingvalues		
Phase-shifting-method	4 Step		12 Step		20 Step	

As shown in Table 4.1, there is an issue with the rows being segmented differently, especially in the gamma investigation, brightness, and contrast. The reason for this is obvious when it comes to brightness and contrast and we need to do a more detailed investigation to work out the best settings. Looking at the gamma factor there are two different ways of to go. The lower row demonstrates the optimal gamma value  $\gamma_{Intensity}$  for an ideal representation of the mean intensity value, while the phase map error  $\gamma_{Phase}$  is of particular interest in this investigation. A thorough examination of the system behaviour reveals that a highly homogenous error map results in a less disturbed image content, consequently leading to the separation of these directions. An optimal reconstruction of the intensity map is only needed if one wants to inverse the background illumination of the

system. Prior to interpreting the results, it is essential to clarify the colour scale. Red numbers exhibit artefacts in the observed image lines with either high amplitude or disturbing irregularities. Conversely, green numbers indicate settings where these effects are undetectable. Orange numbers demonstrate a relatively imperceptible deviation from neighbouring green numbers. When transferred to an application or interpretation, this solution should be deemed acceptable. In order to quantify this interpretation, it is necessary to determine the sweet spot for a 20-step phase-shifting method. This involves utilising the full dynamic range of images at a mean brightness level  $b_{display} = 50 \cdot \%$  with a higher contrast  $a_{display} = 65 \cdot \%$ . In order to achieve sharper image content, it is possible to apply a mean filter over several measurement images or to apply a Gaussian filter to the map of the registered data. However, it should be noted that no further improvements were observed with more than ten mean filtered measurements. A similar outcome was attained by employing a ninth-order Gaussian filter or higher.

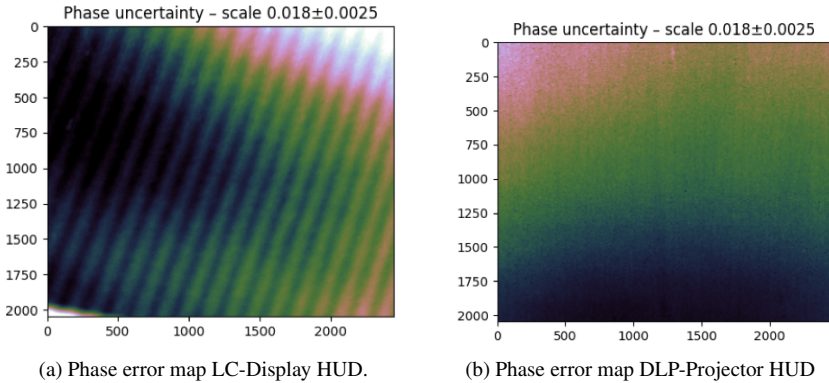
## 4.2 Parameters Influencing the Image Quality

As previously mentioned during the study, several noteworthy observations regarding the impact of varying parameters were recorded. Prior to the study, various displays and imaging systems were examined. These systems demonstrate differing radiation characteristics, resulting in varied intensity maps. It is imperative to be cognizant of photonic noise and its characteristics, as well as irradiance and its representation within the mathematical model, see equation 2.4. The study demonstrated that, under certain conditions, to achieve a satisfactory outcome is possible by selecting an appropriate optimisation strategy.



**Figure 4.1:** Differences in intensity distributions of various concepts

In Figure 4.1(a) the system with a low intensity gradient can be seen. In comparison, that right hand side shows the system with a high intensity gradient. It is observed that in both intensity maps, the intensity value of the highest magnitude is represented by the brightest colour, i.e. yellow. The scale of the intensities is presented in greyscales  $[I_n] = GS$ . The difference between these two computed mean value maps is evident in two respects: firstly, the amplitude of artefacts and secondly, the gradient between the minimum and maximum. The latter is observed to be three times higher from left to right. The discrepancy between the artefacts may be associated with the modulation map, as it exhibits analogous effects. The orientation of these artefacts is influenced by the geometry of the mirror and the orientation of the imaging source with respect to the mirror. In this study, it was determined that the quality of registration maps can be achieved with left- and right-optimised systems, albeit with different phase-shifting methods.



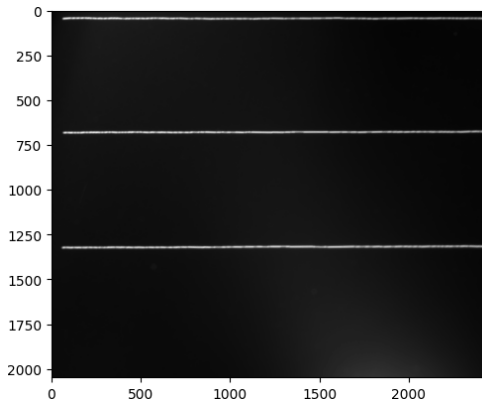
**Figure 4.2:** Differences in intensity distributions of various concepts

The subsequent finding pertains to the computed phase error maps, which are presented in Figures 4.2(a) and 4.2(b). As demonstrated in Figures 4.2(a) and 4.2(b), in conjunction with Figures 4.1(a) and 4.1(b), there is a clear distinction between the optimisation pathways leading to an optimal phase error map and a maximum attainable intensity map. In both figures, the phase error map is depicted in degrees  $[\Delta\varphi_n] = ^\circ$ , with the most significant value indicated as the brightest, i.e. white. This study reveals that, despite the substantial gradient disparity, the mean phase error remains constant, while the variance undergoes a change. Two factors have been identified as contributing to this outcome. Firstly, the left outcome is based on a gamma value  $\gamma_D = 2.3$  combined with a high number of phase shifting steps  $n_P = 20$ . Secondly, the right-hand optimisation path leads towards a low number of steps  $n_P = 4$  and a low gamma factor  $\gamma_D = 1.0$ . A plausibility check reveals that the gamma factor is responsible for the correction function of the grey value scaling. This suggests that the 20-step method, which exhibits heightened sensitivity to grey value disparities, may encounter a more constrained spectrum of grey levels. Conversely, the low gamma factor, which compensates for the less sensitive 4-step method, necessitates a broader range of grey levels. Consequently, the phase errors remain within a similar scope due to this compensatory effect. Furthermore, the adjusted figure in this section illus-

trates the relationship between phase error and the desired result for the adjusted image, as evidenced by Figures 4.2(a) and Figure 4.2(b). It is noteworthy that, even in the presence of artefacts or substructures within the phase error map (as depicted in Figure 4.1(b)), provided their variance remains below a specified threshold, they cannot be perceived in the adjusted image.

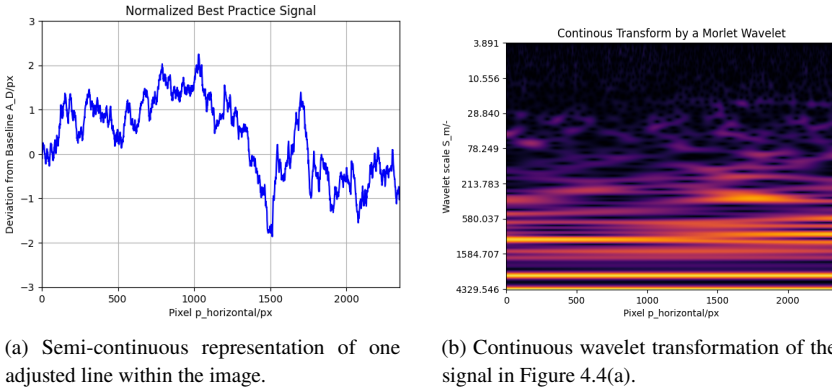
### 4.3 Quantified Best Practice

The image that has undergone the requisite adjustments is displayed, and is based on the phase map that is exhibited in Figure 4.2(a). The following image was assessed by a small number of various persons with different surroundings as horizontal with no defects. This preliminary assessment serves as a foundation for further research, particularly in relation to the technique itself and the data set obtained.



**Figure 4.3:** Best practice for an adjusted image observed via an aspheric non-transparent mirror.

As illustrated in the Figure 4.3, three horizontal lines are distributed across the field-of-view of the observer, specifically the luminance camera. As these lines are assessed, an analysis is conducted to describe them. Initially, the signal undergoes a transformation into a semi-continuous description.



**Figure 4.4:** Concept for a quantified analysis process.

This representation is derived by the centre-of-gravity for the gaussian distribution of every pixel illuminated along one line within the picture (see Figure 4.4(a)). Subsequently, the signal undergoes a continuous wavelet transformation to derive a proper description for the best practice, as illustrated in Figure 4.4(b). As demonstrated in Figure 4.4(a), the signal displayed is the mean of the three lines within the image. It has been normalised to its mean value, and the deviation of the signal towards this mean value is shown on the y-axis. The x-axis indicates the signal's position on the image  $p_{horizontal} = 1500 \cdot px$ . The presence of a singularity around the position can be discerned within this signal. Furthermore, the presence of an overall deviation or two deviations with opposite directions is a possibility. Finally, the signal is superimposed with a noise described by very high frequencies in respect to the signal length, a small amplitudes. The interpretation of the high spatial frequencies and small amplitudes in the signal is complex based on the given data. The underlying reasons for this phenomenon are attributable to the system's lower pixel density on the image side in comparison to the observer side, which is observed via an aspheric mirror. Additionally, the analysis method employed in this study utilises the centre-of-gravity function to evaluate the illuminated pixels on the observer side. Conversely, the wavelet transformation of the described signal



is presented. This analysis is performed to determine an appropriate method for decomposing the best practice and to describe the individual components. The figure presents a colour map of the analysed wavelet scales on the y-axis and the signal length on the x-axis. In a comprehensive wavelet analysis, the colour serves to denote the amplitude of the wavelet at a specific scale and point within the signal. In this instance, the primary objective is to determine the feasibility of signal decomposition to facilitate the extraction of the requisite information. A comparison of the signal with Figure 4.4(a) reveals the presence of four distinguishable components. The singularity, which has been previously discussed and located at  $p_{horizontal} = 1500 \cdot px$ , is the starting point for this analysis. The scale at which the singularity can be located as a spatial wavelength is interpreted, and it is determined that this could match the signal because the singularity might have a spatial wavelength of  $\lambda_{Singularity} = 200 - 300 \cdot px$ . Starting with this approach, the signal is shown to consist of two mid-frequencies, one low frequency and the singularity. In conclusion, the wavelength spectrum range of  $S_\lambda = ]213 \cdot px; 4329 \cdot px[$  the best practice is, and the superimposed deviation of the signal does not exceed  $A_D = \pm 2 \cdot px$ .

## 5 Conclusion and further work

It has been established that image adjustment is achievable through the utilisation of phase-shifting methodologies and the computational processing of registration data, culminating in the attainment of a perfected and evaluated line. This exemplar of a best practice demonstrates the relevant parameters and a potential outcome. Furthermore, the analysis of deviations in head-up display signals is possible through the employment of the continuous wavelet transformation method. However, it is essential to undertake a more profound analysis of this proposed method to facilitate the establishment of quantifiable parameters for head-up display signals. A key point pertains to the analysis of signal, specifically the information that can be derived by a wavelet transformation, and whether the discrete wavelet transform can yield equivalent information. Addressing these issues necessitates first establishing the nature of the scales, whether they are wavelengths or a more comprehensible form of frequency computation.

Subsequently, the correlation between the substructures within the phase error map and their occurrence within the signal can be analysed. Furthermore, the acceptance of the obtained horizontal lines should be investigated in a subject study by a representative subject group.

## References

- [1] Jan Burke et al. *Deflectometry for specular surfaces: an overview*. Version Number: 1. 2022. DOI: 10.48550/ARXIV.2204.11592. URL: <https://arxiv.org/abs/2204.11592> (visited on 12/31/2024).
- [2] *DIN EN ISO 9000:2015-11, Qualitätsmanagementsysteme - Grundlagen und Begriffe (ISO\_9000:2015); Deutsche und Englische Fassung EN-ISO\_9000:2015*. DOI: 10.31030/2325650. URL: <https://www.dinmedia.de/de/-/-/235671064> (visited on 12/22/2024).
- [3] Xiang Gao and Jonas Haeling. “Verfahren zur Kalibrierung eines Head-up-Displays eines Fahrzeugs”. de. DE102019004816A1. Jan. 2020. URL: [https://patents.google.com/patent/DE102019004816A1/de?q=\(verfahren+zur+kalibrierung+eines+Head-up+displays\)&inventor=gao%2c&oq=inventor:+gao%2c+verfahren+zur+kalibrierung+eines+Head-up+displays](https://patents.google.com/patent/DE102019004816A1/de?q=(verfahren+zur+kalibrierung+eines+Head-up+displays)&inventor=gao%2c&oq=inventor:+gao%2c+verfahren+zur+kalibrierung+eines+Head-up+displays) (visited on 01/02/2025).
- [4] Iksoon Lim et al. “Efficient Method for Head-Up Display Image Compensation by Using Pre-Warping”. In: Apr. 2019, pp. 2019–01–1008. DOI: 10.4271/2019-01-1008. URL: <https://www.sae.org/content/2019-01-1008/> (visited on 01/01/2025).
- [5] David Uhlig and Michael Heizmann. “A Probabilistic Approach for Spatio-Temporal Phase Unwrapping in Multi-Frequency Phase-Shift Coding”. In: *IEEE Access* 10 (2022), pp. 52377–52397. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3174121. URL: <https://ieeexplore.ieee.org/document/9771407/> (visited on 12/28/2024).

- [6] Daniel Wagner. “Wahrnehmung von Augmented Reality Head-up Displays”. de. In: (2023). Medium: PDF Publisher: Karlsruher Institut für Technologie (KIT). DOI: 10 . 5445 / IR / 1000154589. URL: <https://publikationen.bibliothek.kit.edu/1000154589> (visited on 12/28/2024).
- [7] Oliver Zink and Marcus Heim. “Verfahren zur Kalibrierung eines Head-up-Displays”. DE102012010120A1. Nov. 2013. URL: <https://patents.google.com/patent/DE102012010120A1/de> (visited on 01/01/2025).
- [8] Chao Zuo et al. “Phase shifting algorithms for fringe projection profilometry: A review”. en. In: *Optics and Lasers in Engineering* 109 (Oct. 2018), pp. 23–59. ISSN: 01438166. DOI: 10.1016/j.optlaseng.2018.04.019. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0143816618302203> (visited on 12/31/2024).



# Attention-Based Few-Shot Learning for Fine-Grained Vehicle Classification

*Stefan Wolf*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
stefan.wolf@kit.edu

## Abstract

While fine-grained vehicle classification has important applications in the security context, it is heavily limited by the availability of data. Particularly, the large number of vehicle models and the ongoing introduction of new models require regular and large dataset updates for a well-applicable vehicle classification system. In the field of few-shot learning, new visual classification approaches based on deep learning were proposed which claim to be more effective in terms of data usage. However, most few-shot approaches are evaluated in scenarios which only include a small number of classes. Thus, we evaluate attention-based few-shot approaches in a more difficult scenario which not only involves a significant number of classes with only a few images available but also including the classes of the base training for which abundant data is available. This new scenario better represents the challenges of few-shot learning for fine-grained classification in real-world scenarios where a classifier cannot afford to lose the capability of recognizing the base classes but also needs to be capable of being extended with new classes without large data collections. This scenario forces the approach to cope with the possibility of misclassifying a novel class as one of the many base classes rendering results more representative for real-world use cases. The results show that a modern transfer learning approach achieves good results even in this difficult scenario. Particularly, on a challenging dataset

involving a high variety in terms of camera perspectives, unsupervised attention can further increase the accuracy.

## 1 Introduction

Fine-grained vehicle classification has its applications particularly in the field of security. Since license plates are easily interchangeable, a vehicle model is a distinctive attribute for manhunts. However, for a well-applicable vehicle classification system, the system needs to support a broad range of vehicle models. Otherwise, the chance of the searched model being supported is low and the risk of false positives is high due to unsupported models in a real-world application being arbitrarily assigned to one of the known vehicles. Thus, a vehicle classification system should be easily extendable with new vehicle models. However, depending on the application, it is difficult to acquire a large amount of samples for every vehicle model. While for advertisement purposes, a lot of images are available and easy to acquire from the web, the applications are often different. For security applications, mainly surveillance images are relevant for which images are difficult to acquire in a data-protective manner. This leads to the conclusion that a vehicle classification system needs to cope with a limited amount of samples during training.

Thus, in this work, we focus on few-shot learning for fine-grained vehicle classification. In the context of visual classification, few-shot learning describes the task of learning a classification task of usually distinguishing about 5 classes with e.g., 1 or 5 training samples available per class. To give the classifier the chance of learning meaningful features, a base training is done with a large amount of samples not involving one of the classes of the few-shot learning. As mentioned before, a fine-grained vehicle classification system should support as many vehicle models as possible. Thus, we investigate an adjusted few-shot setting that involves not only distinguishing the novel classes of the few-shot stage but also distinguishing them from the base classes and the base classes from each other. Thus, it leads to a task of distinguishing well over 100 classes instead of only 5.

We evaluate different few-shot learning approaches like transfer learning [18], dynamic learning [6] or prototypical networks [16] combined with several attention approaches to increase the parameter efficiency. Particularly, we evaluate supervised key-point-based attention mechanisms, unsupervised attention and unsupervised bounding-box-based localization [20].

This work is structured as follows. Section 2 provides an overview of related works. Section 3 introduces the evaluated few-shot setup and compares it to the regular few-shot setup. Section 4 explains the evaluated approaches. Section 5 presents and discusses the quantitative results and Section 6 concludes this work.

## **2 Related work**

### **2.1 Few-shot learning**

A common pattern in few-shot learning is meta learning. Meta learning uses subsets of a large base dataset to simulate few-shot training tasks. This is commonly applied to metric learning which uses a distance between a query and a support embedding in order to assign an image to a class. The distance metric can either be hand-crafted [16, 2] or trained [17]. Another meta learning approach is optimization-based meta learning. These methods learn a weight update procedure in the meta training stage which they later use in the few-shot adaptation to update the weights based on a small set of support samples for new classes [4, 6]. Transfer learning describes the process of fine-tuning a pre-trained network towards a new task. It has been shown that transfer learning can also be employed for few-shot learning [14].

### **2.2 Fine-grained classification**

A wide range of approaches have been proposed specifically for fine-grained visual classification which target the specific difficulties like small inter-class variance and limited amount of data samples per class. To provide a guidance for the classification which parts of an image are important for classification, part-based approaches extract crops of important parts before classification [3,

15, 22]. Bilinear networks [5, 10, 13, 23] use two separate networks which are fused before classification with the idea that the networks learn localizing parts and extracting features separately. Fine-grained classification tasks commonly involve a hierarchical arrangement of classes, e.g. make, model and year for fine-grained vehicle classification. This hierarchy is exploited by hierarchical classification approaches that also learn a coarse-grained classifier which can also improve fine-grained classification due to a larger data availability of coarse-grained classes [8, 1]. Since fine-grained classification relies on recognizing small differences between classes, still images can be limiting due to only providing a single perspective and might be degraded by blur. Thus, performing fine-grained classification on videos instead of still images has been exploited to improve the recognition accuracy [24, 9].

### **2.3 Fine-grained few-shot learning**

Li et al. [12] propose the DN4 network which employs local features instead of image-level features to improve the few-shot classification performance, particularly for fine-grained scenarios. Tang et al. [18] integrate an attention mechanism guided by the pose of the object to be classified, called pose normalization, into classification networks to improve the accuracy of fine-grained few-shot classification.

## **3 Few-shot setup**

Few-shot learning in the context of visual classification describes the task of adapting a classifier to new classes not seen before with a very small amount of samples. This requires some meta knowledge which is gained in most of the approaches with a base training on a large dataset. Afterwards, a fine-tuning is applied for the adaption to the new classes. Common few-shot setups use a 5-way task which involves distinguishing 5 different classes based on a small number of samples, e.g., 1 or 5. However, in the context of fine-grained classification, this scenario is not sensible due to fine-grained classification tasks usually involving a high number of classes in the range of hundreds [11, 21] or thousands [21]. To



approach this issue, recent works refrain from using only a subset of the classes which were selected as novel classes in the data pre-processing but requires the classifier to distinguish all classes [20]. While this leads to a much higher number of classes to be distinguished and thus, a significantly more realistic task, this task definition is still off from realistic fine-grained classification where the user wants to be able to recognize as many classes as possible. Thus, we define a different few-shot scenario which involves recognizing both base and novel classes with a single model. Particularly, the classifier is tasked to be capable of distinguishing the union of the base and the novel classes after the few-shot fine-tuning. This is different from traditional few-shot settings and the adapted setting from Wertheimer and Hariharan [20] for which the classifier does not need to be able to recognize any of the base classes after fine-tuning. Note that contrary to settings involving catastrophic forgetting, all data of the base classes is available during fine-tuning. This means that for each of the base classes an abundant amount of training samples is available while for each novel class only a small amount of samples, 1, 5 or 10 in our experiments, is available. This leads to a significantly long-tailed distribution of the training set rendering training more difficult than for the traditional few-shot setup which involves a balanced fine-tuning set.

## 4 Methods

In this section, the evaluated methods are described. The methods can be split into the overall architecture, the feature extractor backbone and the employed attention mechanism. The backbone extracts features while the attention mechanism is integrated in the backbone to improve the semantic expressiveness of the features. The classification architecture decides the output class based on the features.

### 4.1 Architectures

The architecture describes how the actual classification is performed based on the extracted features. A transfer learning mechanism, a dynamic network and

and a prototypical network are evaluated. All architectures share the split of the training process into a base training which is performed on a dataset with a large number of classes and samples and a few-shot fine-tuning which includes a set of novel classes with only a low number of samples per class.

**Transfer networks.** The transfer learning mechanism is implemented according to Tang et al. [18]. It employs a linear classification layer followed by a softmax activation function. The optimization is performed based on a cross-entropy loss function as common for training deep neural networks. However, the backbone is only trained during the base training. During the few-shot fine-tuning, the linear classification head is reinitialized and the optimization is only performed for the head with the backbone weights being frozen. This reduces the possibility of overfitting on the few samples for the novel classes by lowering the number of optimized parameters.

**Dynamic networks.** Dynamic networks [6] are based on a meta-learning which simulates many few-shot tasks during training by choosing subsets of the base classes and run few-shot fine-tunings on them with a small number of samples during the base training stage. The few-shot fine-tuning is implemented by dynamic networks by passing the samples into the backbone and predicting classification weights with a few-shot classification weight generator based on the extracted features. The classification weight generator predicts per-class feature vectors based on the feature vectors of the images belonging to each class as a combination of the average of them and a trained attention mechanism being fed with the features. To prevent large differences in terms of magnitude between the weight vectors leading to an unbalanced classification decision, the weight vectors and the extracted feature vectors are  $l_2$ -normalized, i.e., a cosine similarity is applied for the classification instead of a dot-product.

**Prototypical networks.** Snell et al. [16] propose prototypical networks which use an embedding function with learnable parameters that maps each sample in an embedding space. Afterwards, a prototype is extracted for each class by averaging the embeddings of the corresponding support samples. A squared euclidean distance is used to decide the class of a query embedding based on

similarity to the prototype rendering it a metric learning approach which is trained in a meta-learning manner.

## 4.2 Backbones

For the experiments, we employ two backbones of different size with both being optimized for a low number of parameters to prevent overfitting in a few-shot scenario. Each backbone is followed by a global average pooling layer that reduces the feature map of the backbone’s last layer to a feature vector by averaging over all pixels per channel.

**Conv4.** The Conv4 backbone is representing a small backbone with only 4 convolutional layers. Every layer consists of the  $3 \times 3$ -convolution itself, a batch normalization layer, a ReLU activation function and a maximum pooling layer of size 2.

**ResNet18.** The ResNet18 [7] backbone is significantly larger and more optimized with residual connections than the Conv4. However, with regards to the overfitting-prone few-shot task, we choose the smallest ResNet variant.

## 4.3 Attention mechanisms

We adopt the idea of Tang et al. [18] to apply a pose normalization as an inductive bias reducing the intra-class variance which has to be captured by the model. Tang et al. [18] advocate for a supervised pose normalization strategy and against an end-to-end training for few-shot classification. However, we also evaluate unsupervised attention mechanisms in the expectation that a base training with a high number of images and classes will lead to an attention pattern with a similar effect. Due to the domain differences of working with cars instead of animals, we choose the name keypoints instead of pose. Besides the integrated keypoint prediction mechanism as proposed by Tang et al. [18], we evaluate an externally trained keypoint predictor with the predicted keypoints being integrated in a similar manner, an unsupervised attention mechanism that also integrated a heat map but has no pose loss and an unsupervised localization mechanism.

**Internal keypoint estimator.** The internal keypoint estimator as proposed by Tang et al. [18] is integrated into the classification network by employing a simple two-layer CNN on top of an intermediate layer of the backbone. The predicted keypoint heat maps are fused with the backbone’s feature map by applying a Hadamard product between each pixel of the heat map and the feature map per keypoint heat map. Each resulting attention-modulated feature map is then reduced to a feature vector by summing all pixel values per-channel and normalized by the spatial-wise sum of the keypoint heat map, i.e., a weighted global average pooling with the weights being the heatmap values. The resulting per-heat-map feature vectors are concatenated before being passed to the classifier. During training, an additional keypoint estimation loss is applied to the keypoint heat maps to guide the network to predict consistent keypoints, i.e., a pixel-wise log loss. Since the datasets used for evaluation lack any keypoint annotation, we employ a pre-trained vehicle keypoint prediction approach [19] to generate a pseudo ground truth.

**External keypoint estimator.** For the external keypoint estimator, we employ PAMTRI [19]. Based on the predicted keypoints, we construct a heat map for each keypoint and fuse them with the final feature map of the backbone similar to the attention mechanism based on the internal keypoint estimator. This loosens the limit in terms of model complexity of the internal keypoint estimator. Nonetheless, it might also be disadvantageous since the keypoint prediction is not trained combined with the classification loss which can lead to heat maps that are not well-suited for the classification.

**Unsupervised attention estimator.** The unsupervised attention estimator uses an architecture similar to the attention mechanism based on the internal keypoint estimator. However, it does not employ an additional keypoint estimation loss. Thus, it solely fulfills the purpose of a spatial attention mechanism.

**Unsupervised localization.** We evaluate the unsupervised localization based on bounding boxes as proposed by Wertheimer and Hariharan [20]. It employs two learned prototype vectors representing background and foreground. Note that these prototype vector should not be confused with the class prototype vectors

used for some architectures. Based on the final feature map, a foreground and a background mask is predicted based on the proximity of each pixels feature vector to the foreground and background prototype feature vector, respectively. Afterwards, both masks are applied independently to the feature map using a Hadamard product resulting in two attention-modulated feature maps which are reduced to two feature vectors by applying global average pooling. Both feature vectors are concatenated to construct the final feature vector for classification.

## 5 Experiments

### 5.1 Datasets

For the evaluation, two datasets are considered: Stanford Cars [11] and CompCars Surveillance (CompCars SV) [21]. Stanford Cars is a web-nature dataset containing 16,185 images from a total of 196 classes. For the experiments, 130, 17 and 49 classes were used for training, validation and testing, respectively. CompCars SV is a surveillance-nature dataset containing 44,481 front-view images from a total of 281 classes. Of these, 140, 71 and 70 are used for training, validation and testing, respectively.

### 5.2 Experimental setup

The experiments using a Conv4 backbone use images of size  $84 \times 84$  pixels while  $224 \times 224$  pixels are used for experiments involving the ResNet-18 backbone.

The internal keypoint estimator requires ground truth keypoints. Due to the lack of ground truth keypoint annotations for the datasets Stanford Cars and CompCars SV, we generate pseudo ground truth keypoints by using a keypoint estimation approach. We choose the PAMTRI [19] keypoint prediction approach using the model pre-trained by the original authors on a synthetic dataset. This pre-trained PAMTRI model is also used as the external keypoint estimator.

We use the class-wise macro-averaged F1 score as evaluation metric. While the images of all classes including the base classes is considered for the calculating

**Table 5.1:** Comparison of different backbones and few-shot architectures for 1, 5 and 10 shots on Stanford Cars. The metric is F1 score. ResNet18 clearly outperforms Conv4 by a large margin compared architecture for architecture and for all shots. Regarding the few-shot architectures, a simple transfer approach proves advantageous over the more complex dynamic or proto approaches.

Model	Stanford Cars			
	Shots	1	5	10
Transfer (Conv4)		$15.8 \pm 0.7$	$23.6 \pm 0.8$	$24.5 \pm 1.0$
Dynamic (Conv4)		$6.3 \pm 0.2$	$16.9 \pm 0.3$	$21.3 \pm 0.3$
Proto (Conv4)		$4.6 \pm 0.2$	$12.0 \pm 0.2$	$13.9 \pm 0.2$
Transfer (ResNet18)		<b><math>39.6 \pm 0.7</math></b>	<b><math>53.8 \pm 1.0</math></b>	<b><math>52.0 \pm 0.7</math></b>
Dynamic (ResNet18)		$19.4 \pm 0.5$	$37.9 \pm 0.5$	$42.8 \pm 0.4$
Proto (ResNet18)		$18.4 \pm 0.4$	$34.9 \pm 0.4$	$38.0 \pm 0.3$

the class-wise F1 scores, the reported average F1 scores are only averaged over the novel classes. This puts a stronger emphasis on the important novel classes while still taking confusions of base classes as novel into consideration for the evaluation.

### 5.3 Results

In this section, the results of the experiments are shown and discussed. We run each experiment with 1, 5 and 10 shots for the few-shot training. First, we evaluate the impact of the backbones and the few-shot architectures without employing an attention mechanism. The results for Stanford Cars are shown in Table 5.1 and the results for CompCars SV are shown in Table 5.2.

On both datasets, the results indicate clearly that the few-shot architecture can take advantage of the higher number of parameters of the ResNet18 backbone compared to the Conv4. Comparing the different few-shot architectures, the transfer network achieves higher scores for all evaluated settings compared to the dynamic or proto networks. Nonetheless, the gap between the transfer network and the other approaches is shrinking with more shots. Additionally, the benefit

**Table 5.2:** Comparison of different backbones and few-shot architectures for 1, 5 and 10 shots on CompCars SV. The metric is F1 score. ResNet18 clearly outperforms Conv4 by a large margin compared architecture for architecture and for all shots. Regarding the few-shot architectures, a simple transfer approach proves advantageous over the more complex dynamic or proto approaches.

Model Shots	CompCars Surveillance		
	1	5	10
Transfer (Conv4)	76.1 $\pm$ 0.6	81.3 $\pm$ 0.5	81.7 $\pm$ 0.5
Dynamic (Conv4)	42.4 $\pm$ 0.5	73.3 $\pm$ 0.3	79.1 $\pm$ 0.3
Proto (Conv4)	30.9 $\pm$ 0.4	60.3 $\pm$ 0.3	66.3 $\pm$ 0.3
Transfer (ResNet18)	<b>85.7</b> $\pm$ 0.6	<b>88.6</b> $\pm$ 0.4	<b>88.8</b> $\pm$ 0.4
Dynamic (ResNet18)	52.0 $\pm$ 0.5	77.2 $\pm$ 0.3	81.5 $\pm$ 0.2

from using 10 shots instead of 5 is almost negligible in most scenarios. Both findings indicate that even the ResNet18 is not prone to overfitting with the evaluated few-shot approaches and it might already be limited in its learnable parameters. A larger backbone like ResNet50 will probably lead to better results for 5 and 10 shots.

Next, we evaluate different attention mechanisms based on the best methods of the previous experiment, i.e., ResNet18 with a transfer network. The results are shown in Table 5.3 for Stanford Cars and in Table 5.4 for CompCars SV.

For Stanford Cars, the results show a clear advantage for the attention estimation giving a great increase in accuracy over the base approach for all number of shots. Unsupervised localization is consistently the second best option which also shows a significant advantage over the baseline. However, the attention mechanism supervised by keypoint estimation show a degradation of the accuracy for both an internal or external keypoint estimation. Only the internal keypoint estimation for 10 shots shows a slight advantage over the base network. These results show that the ResNet18 by itself is not able to extract features which are invariant to the perspective and an attention mechanism can greatly improve the feature extraction capabilities. However, the estimated keypoints are likely not accurate enough to give an advantage for the feature extraction. A

**Table 5.3:** Comparison of different attention mechanisms for 1, 5 and 10 shots on Stanford Cars with a ResNet18 backbone. The metric is F1 score. For all number of shots, the attention estimator outperforms the base model as well as the other attention mechanisms.

Model	Stanford Cars			
	Shots	1	5	10
Base		$39.6 \pm 0.7$	$53.8 \pm 1.0$	$52.0 \pm 0.7$
Unsupervised Localization		$41.4 \pm 1.2$	$59.0 \pm 0.9$	$63.7 \pm 0.7$
Attention Estimator		<b><math>49.0 \pm 0.8</math></b>	<b><math>62.4 \pm 0.7</math></b>	<b><math>66.9 \pm 1.0</math></b>
Internal Keypoints		$37.7 \pm 0.9$	$51.3 \pm 0.8$	$55.2 \pm 0.7$
External Keypoints		$37.6 \pm 0.8$	$45.9 \pm 0.6$	$47.7 \pm 1.0$

**Table 5.4:** Comparison of different attention mechanisms for 1, 5 and 10 shots on CompCars SV. The metric is F1 score. Only the unsupervised localization approach and only for 10 shots gives a significant advantage. While the impact of most attention mechanisms is small for 5 shots, all of them are degrading the accuracy for 1 shot. The reason for these results are likely the small variance in terms of view with all images being front view. Thus, the main advantage of the attention mechanisms, locating important features, is not effective.

Model	Shots	CompCars Surveillance		
		1	5	10
Base		<b><math>85.7 \pm 0.6</math></b>	$88.6 \pm 0.4$	$88.8 \pm 0.4$
Unsupervised Localization		$83.4 \pm 0.9$	<b><math>88.7 \pm 0.7</math></b>	<b><math>90.4 \pm 0.8</math></b>
Attention Estimator		$79.4 \pm 0.6$	$86.9 \pm 0.7$	$87.9 \pm 0.6$
Internal Keypoints		$73.3 \pm 1.1$	$83.0 \pm 0.8$	$84.9 \pm 0.9$
External Keypoints		$68.0 \pm 0.7$	$78.4 \pm 0.5$	$80.2 \pm 0.8$

better keypoint estimator which better fits the domain of the target data might improve the results.

On CompCars SV, the unsupervised attention estimator as well as the keypoint estimators are degrading the accuracy of the classification. Only the unsupervised localization for 10 shots is showing a significant advantage over the base network. This divergence in results compared to the Stanford Cars dataset is



likely due to the small variance in terms of camera perspective in the CompCars SV which renders the attention mechanisms unnecessary.

## 6 Conclusion

In this work, we introduced a new few-shot setting which better reflects the applications of fine-grained vehicle classification. Compared to the commonly assumed few-shot setting which only distinguishes novel classes, in our setting, the classifier is tasked to distinguish base classes as well as novel classes. This renders the task significantly more difficult by incorporating a decision between more than 100 classes compared to for example 5 classes as common in few-shot scenarios. Nonetheless, modern CNNs with an attention module and training in a transfer learning manner already achieve an F1 score of 49% in a complex scenario with a single shot. While the unsupervised attention module proved to be essential on the more complex Stanford Cars dataset, most attention mechanisms didn't show a gain for the view-wise simpler CompCars SV dataset. Only the unsupervised localization achieved a significant gain for the 10-shot setting. For future work, it is recommended to test larger backbone architectures since diminishing gains with more shots indicate a limitation in terms of model capacity.

## Acknowledgments

This research was performed in close collaboration between Stefan Wolf and Adrian Nelson.

## References

- [1] Marco Buzzelli and Luca Segantin. "Revisiting the CompCars Dataset for Hierarchical Car Classification: New Annotations, Experiments, and Results". In: *Sensors* 21 (Jan. 2021), p. 596. DOI: 10.3390/s21020596.

- [2] Yinbo Chen et al. “Meta-Baseline: Exploring Simple Meta-Learning for Few-Shot Learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 9062–9071.
- [3] Jie Fang et al. “Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.7 (2016), pp. 1782–1792.
- [4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135.
- [5] Yang Gao et al. “Compact bilinear pooling”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 317–326.
- [6] Spyros Gidaris and Nikos Komodakis. “Dynamic Few-Shot Visual Learning without Forgetting”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4367–4375.
- [7] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [8] Yuqi Huo et al. “Coarse-to-fine grained classification”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 1033–1036.
- [9] Jannik Koch, Stefan Wolf, and Jürgen Beyerer. “A Transformer-based Late-Fusion Mechanism for Fine-Grained Object Recognition in Videos”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 100–109.
- [10] Shu Kong and Charless Fowlkes. “Low-rank bilinear pooling for fine-grained classification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 365–374.
- [11] Jonathan Krause et al. “3D Object Representations for Fine-Grained Categorization”. In: *4th International IEEE Workshop on 3D Representation and Recognition (3DRR-13)*. Sydney, Australia, 2013.

- [12] Wenbin Li et al. “Revisiting Local Descriptor based Image-to-Class Measure for Few-shot Learning”. In: *CVPR*. 2019.
- [13] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. “Bilinear CNN models for fine-grained visual recognition”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1449–1457.
- [14] Akihiro Nakamura and Tatsuya Harada. “Revisiting fine-tuning for few-shot learning”. In: *arXiv preprint arXiv:1910.00216* (2019).
- [15] Marcel Simon and Erik Rodner. “Neural activation constellations: Unsupervised part model discovery with convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1143–1151.
- [16] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical Networks for Few-shot Learning”. In: *Advances in Neural Information Processing Systems*. 2017.
- [17] Flood Sung et al. “Learning to compare: Relation network for few-shot learning”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.
- [18] Luming Tang, Davis Wertheimer, and Bharath Hariharan. “Revisiting Pose-Normalization for Fine-Grained Few-Shot Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14352–14361.
- [19] Zheng Tang et al. “Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 211–220.
- [20] Davis Wertheimer and Bharath Hariharan. “Few-Shot Learning With Localization in Realistic Settings”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [21] Linjie Yang et al. “A large-scale car dataset for fine-grained categorization and verification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3973–3981.

- [22] Hantao Yao et al. “Coarse-to-fine description for fine-grained visual categorization”. In: *IEEE Transactions on Image Processing* 25.10 (2016), pp. 4858–4872.
- [23] Chaojian Yu et al. “Hierarchical bilinear pooling for fine-grained visual recognition”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 574–589.
- [24] Chen Zhu et al. “Fine-grained video categorization with redundancy reduction attention”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 136–152.

# Causality-Driven AI for Manufacturing Systems

*Shahenda Youssef*

Vision and Fusion Laboratory  
Institute for Anthropomatics  
Karlsruhe Institute of Technology (KIT), Germany  
shahenda.youssef@kit.edu

## Abstract

Causality-driven AI focuses on enabling interpretable, robust, and actionable decision-making, with a specific focus on manufacturing systems. Existing methods, including Structural Causal Models and Propensity Score Matching, have demonstrated significant applications in process optimization and fault detection. Despite significant advancements, these methods often face limitations with seamlessly incorporating domain knowledge, efficiently handling high-dimensional and heterogeneous data, and leveraging causality for scalable deep learning architectures. This work highlights these gaps and proposes a unified direction to enhance causality in deep learning by integrating causal inference methods that embed causal priors into neural networks, optimizing causal discovery algorithms, and improving both the performance and interpretability of deep learning models through causal attributions and loss function optimization.

## 1 Introduction

The integration of artificial intelligence (AI) into manufacturing systems has transformed traditional production processes, enabling higher efficiency, better quality control, and predictive capabilities. However, most existing AI models

rely heavily on data-driven approaches that capture correlations rather than causal relationships, assuming independent and identically distributed (i.i.d.) data. Real-world contexts often violate this assumption [18], leading to performance degradation under distribution shifts, and in dynamic environments, training on more data is impractical. This limitation can lead to models that lack interpretability, robustness under changing conditions, and actionable insights necessary for high-stakes industrial applications [13].

Causality-driven AI addresses these limitations by focusing on understanding the cause-effect relationships within manufacturing systems. Causal inference allows for the estimation of the effects of interventions and the identification of critical factors influencing system behavior, enabling actionable insights [16]. Causal discovery, on the other hand, involves uncovering the underlying causal structures from observational data, which is particularly valuable in complex manufacturing environments where domain knowledge may be incomplete or evolving [22].

Identifying and estimating causal effects requires addressing confounders variables that influence both the treatment (the intervention or action under study) and the outcome (the resulting effect of that intervention), potentially biasing the results. Instrumental variables (IVs) help isolate causal effects by influencing the treatment without directly affecting the outcome. Properly managing confounders and leveraging IVs ensures accurate and reliable causal inferences.

Current researches often address individual problems in isolation, such as causal discovery, inference, or integrating domain knowledge into machine learning models. While these efforts provide valuable insights, they fall short of offering a unified framework that combines these aspects cohesively. Our goal is to develop an integrated approach that seamlessly incorporates causality into deep neural networks (DNNs).

The remainder of this report is structured as follows: Section 2 and Section 3 overview causal discovery and inference methods, respectively. Section 4 reviews the state-of-the-art causality methods in manufacturing. Section 5 outlines the proposed framework and future research directions.

## 2 Causal Discovery

Causal discovery with a main focus on multivariate time series data focuses on identifying cause-and-effect relationships among multiple variables that evolve over time. In a multivariate time series, each time point contains observations for several variables, such as temperature, pressure, or humidity measured at regular intervals.

Let  $X(t) = [x_1(t), x_2(t), \dots, x_d(t)]^T$  denote a  $\mathcal{K}$ -dimensional multivariate time series, where  $x_i(t)$  represents the value of the  $i$ th variable at time  $t$ . The goal is to identify the causal relationships among the variables  $x_1, x_2, \dots, x_d$  using the observed data over time. The causal relationships can be represented as a directed acyclic graph (DAGs)  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the set of variables and  $\mathcal{E}$  is the set of directed edges indicating causal influences. For time series data, these relationships often extend to temporal dependencies, capturing both contemporaneous and lagged effects.

### Constraint-Based Approaches

Methods that rely on conditional independence (CI) tests to prune edges from a fully connected graph and then orient those edges based on separation properties like the Peter-Clark (PC) algorithm. For example, if  $x_i$  and  $x_j$  are conditionally independent given a set of variables  $S$ , then there is no direct causal link between them. Conditional independence is tested using metrics like partial correlation or mutual information.

PCMCI is a causal discovery algorithm for multivariate time series that tackles high dimensionality by conditioning on a set of variables that minimally includes the parents of the treatment  $X$  and the outcome  $Y$ , while avoiding the inclusion of irrelevant variables when removing links [17]. PCMCI applies local CI tests to prune edges. It introduces a dedicated test to assess whether two variables at time  $t$  are conditionally independent given a suitable conditioning set  $\mathbf{Z}_t$ , which may include lagged variables.

A common test in the Gaussian setting uses partial correlation. Suppose that we want to test whether  $X_{i,t}$  is independent of  $X_{j,t}$  given a set  $\mathbf{Z}_t$ , we compute the

partial correlation coefficient  $\rho_{X_{i,t}, X_{j,t} | \mathbf{Z}_t}$ . Under Gaussian assumptions, the Fisher- $z$  transform can be used:

$$z = \frac{1}{2} \ln\left(\frac{1+\rho}{1-\rho}\right) \quad \text{and} \quad z \sim \mathcal{N}\left(0, \frac{1}{|T-|\mathbf{Z}_t|-3|}\right), \quad (2.1)$$

where  $T$  is the number of the time steps, enabling a significance test to see if  $\rho \neq 0$ . If  $|z|$  is below a chosen threshold, PCMCI infers  $X_{i,t} \perp\!\!\!\perp X_{j,t} | \mathbf{Z}_t$  and removes the edge.

PCMCI usually proceeds in an iterative fashion to identify relevant conditioning sets for each potential causal link. By applying constraint-based pruning, PCMCI avoids enumerating all possible graphs, making it more tractable in high-dimensional scenarios.

## Score-Based Approaches

These methods define a scoring function to evaluate how well  $\mathcal{G}$  fits the data  $\mathbf{D}$ , then attempt to maximize (or minimize) this score—often using criteria such as the Bayesian Information Criterion (BIC)—to identify the causal graph that best explains the data [4].

Under certain assumptions (e.g., Gaussian noise), the log-likelihood can be expressed as a product of local likelihoods:

$$\log p(\mathbf{D} | \mathcal{G}) = \sum_{j=1}^n \sum_{t=1}^T \log p(X_{j,t} | \text{Pa}(X_{j,t})). \quad (2.2)$$

$$\text{BIC}(\mathcal{G}; \mathbf{D}) = \log p(\mathbf{D} | \mathcal{G}) - \frac{|\theta_{\mathcal{G}}|}{2} \log(T), \quad (2.3)$$

where  $|\theta_{\mathcal{G}}|$  is the number of free parameters in the graph  $\mathcal{G}$ .

## Nonlinear Causal Discovery

For nonlinear relationships, techniques like kernel-based methods or neural network models can be applied [21].



$$\mathbf{x}_t = F(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots, \mathbf{x}_{t-p}; \Theta) + \varepsilon_t, \quad (2.4)$$

where  $F$  could be an MLP, RNN, LSTM, or Transformer that ingests lagged data. To enforce sparsity and interoperability, we can incorporate a penalty on the network weights  $\Theta$ , such as:

$$\Omega(\Theta) = \lambda \sum_{i,j} \|\mathbf{W}_{ji}\|_1 \quad \text{or} \quad \lambda \sum_{i,j} \mathbf{1}\{|\mathbf{W}_{j,i}| > \delta\}, \quad (2.5)$$

where  $\mathbf{W}_{ji}$  might be the sub-block of weights that connects variable  $j$  at earlier time steps to variable  $i$ .

Some approaches also incorporate DAG constraints directly into the neural optimization. A known continuous DAG penalty is:

$$h(\mathbf{W}) = \text{trace}\left(\exp(\mathbf{W} \odot \mathbf{W})\right) - d, \quad (2.6)$$

where  $\mathbf{W}$  is a weight matrix,  $d$  is the dimension, and  $\odot$  denotes elementwise multiplication. Imposing  $h(\mathbf{W}) = 0$  ensures no directed cycles.

### 3 Causal Inference

Causal inference is the process of concluding a causal connection based on the conditions of an effect's occurrence. It involves establishing that a change in one variable (the cause) brings about a change in another variable (the effect) [14].

Structural Causal Models (SCMs) provide a formal framework to represent causal relationships using DAGs and structural equations. Causal relationships are modeled as:

$$X_i = f_i(\text{Pa}(X_i), U_i), \quad i = 1, \dots, p, \quad (3.1)$$

where  $X_i$  are the observed variables,  $\text{Pa}(X_i)$  are the parent variables, and  $U_i$  are the unobserved variables.

Causal inference in time series calls for robust methodologies that can handle temporal dependencies, confounding factors, and noise. These methods are crucial for uncovering causal relationships and must be carefully chosen to match the specific data characteristics and underlying assumptions.

## Causal Impact

Causal Impact is a Bayesian framework for estimating the effect of an intervention on a time series. It decomposes the observed series into components:

$$Y_t = \mu_t + \tau_t + \epsilon_t, \quad (3.2)$$

where  $\mu_t$  is the trend,  $\tau_t$  is the seasonal component, and  $\epsilon_t$  is the noise.

The counterfactual  $\hat{Y}_t$  (expected  $Y_t$  without intervention) is predicted using data before the intervention. The causal effect is computed as:

$$\text{Effect}_t = Y_t - \hat{Y}_t. \quad (3.3)$$

## Propensity Score Matching (PSM)

PSM adjusts for confounding by matching treated and control time series based on their propensity scores,  $e(X_i)$ , representing the probability of receiving treatment given observed covariates.

For each unit  $i$ , estimate the propensity score:

$$e(X_i) = P(D_i = 1|X_i), \quad (3.4)$$

where  $X_i$  are covariates, and  $D_i$  indicates treatment. Treated and control units with similar scores are matched, and outcomes are compared.

## Difference-in-Differences (DiD)

DiD estimates causal effects by comparing outcomes before and after an intervention between treated and control groups.

Let  $Y_{it}$  represent the outcome for unit  $i$  at time  $t$ :

$$Y_{it} = \alpha + \delta T_t + \gamma D_i + \beta(T_t \cdot D_i) + \epsilon_{it}, \quad (3.5)$$

where  $T_t$  is a time indicator (1 for post-intervention, 0 otherwise),  $D_i$  is a treatment group indicator (1 for treated, 0 for control),  $\beta$  is the treatment effect, and  $\epsilon_{it}$  is an error term. The causal effect is measured by  $\beta$ , estimated using ordinary least squares (OLS).

### Average Treatment Effect (ATE)

ATE quantifies the expected difference in outcomes if all units received the treatment versus no treatment. For binary treatment  $T \in \{0, 1\}$ , the ATE is:

$$ATE = \mathbb{E}[Y | \text{do}(T = 1)] - \mathbb{E}[Y | \text{do}(T = 0)], \quad (3.6)$$

where  $\mathbb{E}[Y | \text{do}(T = t)]$  represents the expectation of the outcome  $Y$  under the intervention  $\text{do}(T = t)$ , do-operator allow for the identification and estimation of causal effects from observational data under certain conditions.

The backdoor criterion ensures causal identification by controlling for confounders  $Z$  that block spurious paths between  $X$  and  $Y$ . The causal effect is computed as:

$$P(Y | \text{do}(T)) = \sum_Z P(Y|T, Z)P(Z). \quad (3.7)$$

The frontdoor criterion applies when a mediator  $M$  fully transmits the effect of  $T$  on  $Y$ , bypassing unobserved confounders. The causal effect is:

$$P(Y | \text{do}(T)) = \sum_M P(M|T) \sum_Z P(Y|M, Z)P(Z). \quad (3.8)$$

### Sensitivity Analysis

Sensitivity analysis is a critical tool to address unmeasured confounder variables and measurement errors in causal inference, by quantifying the impact of variations in input parameters on outcomes. Sensitivity analysis helps identify the most influential variables and their causal significance in complex manufacturing systems. By systematically varying input conditions and analyzing their effects on process outputs, it is possible to detect vulnerabilities, optimize critical parameters, and improve system reliability [6].

The sensitivity parameter,  $\delta_0$ , is defined as:

$$\delta_0 = \psi_0 - \psi_0^f, \quad (3.9)$$

where  $\psi_0$  is the observed data parameter and  $\psi_0^f$  is the full data parameter.  $\delta_0$  captures the deviation from the identifiability assumption, allowing us to understand how unmeasured confounding impacts causal inference.

Sensitivity analysis uses  $\delta_0$  to quantify the impact of unmeasured confounding. The sensitivity parameter adjusts for measurement error:

$$\delta_0 = \mathbb{E}(Y | A + \epsilon, W) - \mathbb{E}(Y | A^* + \epsilon, W), \quad (3.10)$$

where  $Y$  is the outcome variable,  $A$  is the treatment indicator variable,  $A^*$  is the surrogate exposure variable,  $\epsilon$  is an intervention shift, and  $W$  are covariates.

## 4 Causality in Manufacturing Systems

Causal models enable more reliable fault detection [11], root cause analysis [23, 15, 10], and process optimization [25, 12, 30] by identifying the true drivers of system behavior rather than relying on surface-level correlations [14]. Techniques such as SCMs [3] and Propensity Score Matching (PSM) [28] have been employed in manufacturing contexts to uncover these relationships, providing a foundation for more interpretable and generalizable AI systems [9, 19].

The paper by Wua et al. [24] introduces a quantitative causal analysis and optimization framework for controlling inclusion defects in steel products. The framework integrates PSM for causal analysis and AutoGluon-Tabular (AGT) for predictive modeling and optimization. PSM quantifies causal effects by simulating randomization, eliminating selection bias in observational data. After obtaining the two potential outcomes for the base sample (the defect sample) the causal effect of the treatment variable  $d$  on the outcome variable (inclusion defects)  $y$  is defined as the ATE:

$$TE(d \rightarrow y) = ATE|_{\text{base}} = (\mathbb{E}[Y | D = 1] - \mathbb{E}[Y | D = 0])|_{\text{base}} \quad (4.1)$$

The authors claim that this approach enhances industrial processes by enabling targeted interventions, real-time scalability, and interpretable insights, achieving an 8.85% defect reduction through causal analysis.

The work by Schwarz et al. [20] propose a data-driven causal framework to optimize rework decisions in manufacturing systems, focusing on yield improvement while managing rework costs. The authors address a multistage production

scenario where the rework decision depends on intermediate system states, with final quality assessments only available post-production. They apply causal machine learning techniques, particularly Double Machine Learning (DML), to estimate Conditional Treatment Effects and derive rework policies

$$\begin{aligned} \psi(W; \theta, \eta) := & (g(1, X) - g(0, X)) \\ & + \frac{A(Y - g(1, X))}{m(X)} \\ & - \frac{(1 - A)(Y - g(0, X))}{1 - m(X)}. \end{aligned} \quad (4.2)$$

where  $W$  is the observed data tuple  $(X, A, Y)$ ;  $X$  is the observed covariates,  $A$  is the binary treatment indicator,  $Y$  is the observed yield after production.  $g(A, X)$  is the outcome model, representing  $\mathbb{E}[Y \mid A, X]$ ,  $m(X)$  is the propensity score, representing  $\mathbb{P}(A = 1 \mid X)$ , and  $\theta$  is the target causal parameter.

This approach ensures that decisions are causally sound and robust against confounders. To evaluate the robustness of estimates against unobserved confounders, sensitivity analysis is used. This equation provides the upper bound for bias due to unobserved confounding:

$$|\tilde{\theta} - \theta|^2 \leq f(W, \rho, \zeta_y, \zeta_d), \quad (4.3)$$

where  $f$  is a known function and  $\rho, \zeta_y, \zeta_d$  are model-specific parameters capturing the confounding strength.

The framework is validated using real-world data from opto-electronic semiconductor manufacturing achieving a yield improvement of 2-3%. The study highlights the utility of causal reasoning in handling complex trade-offs in manufacturing, such as balancing cost and quality, and contributes a novel integration of causal machine learning into production planning and optimization.

The paper by Chattopadhyay et al. [3] introduces a novel methodology to compute causal attributions in neural networks by modeling them as SCMs. The central objective is to determine the ATE of an input neuron on an output neuron. For continuous input domains, the ATE is defined as:

$$\text{ATE}_{\text{do}(x_i=\alpha)}^y = \mathbb{E}[y \mid \text{do}(x_i = \alpha)] - \text{baseline}_{x_i}, \quad (4.4)$$

where  $\text{baseline}_{x_i}$  is the expected output when the input variable  $x_i$  takes its baseline value. The interventional expectation  $\mathbb{E}[y | \text{do}(x_i = \alpha)]$ , relies on:

$$\mathbb{E}[y | \text{do}(x_i = \alpha)] = \int y \cdot p(y | \text{do}(x_i = \alpha)) dy \quad (4.5)$$

This requires approximations via Taylor expansions and efficient estimation using causal regressors to handle high-dimensional data. The SCM-based approach uncovers direct causal relationships between inputs and outputs. By marginalizing over non-causal features, the method ensures attributions are invariant to spurious correlations or constant shifts in the input data.

The work by Zhang et al. [29] introduces a causal discovery and inference-based fault detection and diagnosis (FDD) method for Heating, Ventilation, and Air Conditioning (HVAC) systems. The proposed method enhances interpretability and reliability by uncovering causal relationships between faults and symptoms, a significant improvement over traditional data-driven models. Causality is established using do-calculus and quantified via the individual average causal effect (IACE), which measures the impact of interventions on system variables. The IACE is estimated using three criteria: probability-based (PIACE), expectation-based (EIACE), and mode-based (MIACE), expressed as:

$$\text{PIACE} = \frac{1}{2} \sum_{i=0}^{N_1} |P(S = s_i | \text{do}(F = 1)) - P(S = s_i | \text{do}(F = 0))| \quad (4.6)$$

$$\text{EIACE} = \frac{|\mathbb{E}(S(\text{do}(F = 1))) - \mathbb{E}(S(\text{do}(F = 0)))|}{S_{\max} - S_{\min}} \quad (4.7)$$

$$\text{MIACE} = \frac{|\text{argmax}_x P(S = x | \text{do}(F = 1))|}{S_{\max} - S_{\min}} \quad (4.8)$$

$$- \frac{|\text{argmax}_x P(S = x | \text{do}(F = 0))|}{S_{\max} - S_{\min}}. \quad (4.9)$$

Here,  $S$  denotes symptoms,  $F$  faults,  $N_1$  is the number of intervals, and  $S_{\max}$  and  $S_{\min}$  represent the symptom range.

The methodology constructs a Backward Structural Causal Model (BSCM), which reverses causal graphs to support fault detection. The FDD system achieves high diagnostic accuracy (99.58%) and significantly reduces the time for hyperparameter optimization and model training.

The paper by Hagedorn et al. [8] introduces a log data-driven causal reasoning framework to address unforeseen production downtimes in manufacturing processes. By leveraging Causal Graphical Models (CGMs), the study formalizes causal relationships between machine parameters, configurations, and error messages. Conditional independencies are determined using the global Markov property  $V_i \perp\!\!\!\perp V_j \mid S$  if  $V_i$  and  $V_j$  are d-separated by  $S$  in  $G$ .

Causal structure learning employs the PC algorithm, utilizing Conditional Independence (CI) tests to estimate skeleton graphs and orient edges with domain knowledge.

Causal inference applies the do-operator to quantify the impact of interventions:

$$P(V_j \mid \text{do}(V_i = v_i)) \quad (4.10)$$

providing actionable insights for mitigating downtime by altering machine configurations or operational strategies.

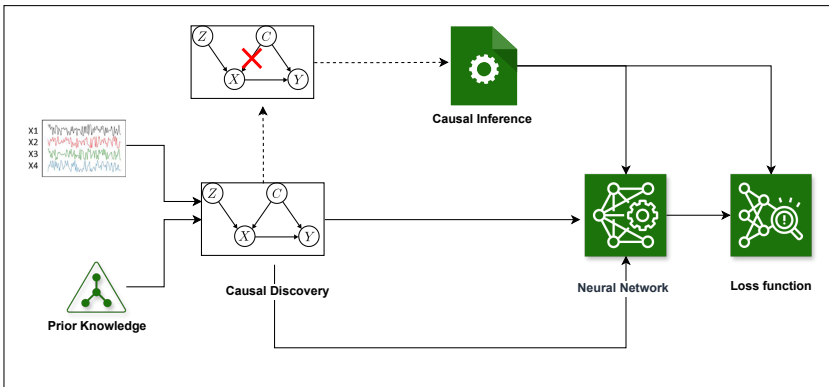
## 5 Discussion

The exploration of causality in machine learning has yielded promising advancements, particularly in addressing challenges such as interpretability and robustness. However, existing methods face practical limitations that hinder their broader adoption in deep learning contexts.

A critical observation from the reviewed literature is the need for methods that scale effectively with the complexity of deep neural networks. Traditional causality approaches, while theoretically sound, are computationally prohibitive for high-dimensional datasets. Future research should focus on optimizing these algorithms to operate within the constraints of real-world systems. Another pressing challenge lies in the integration of temporal data, which is vital for applications like predictive maintenance and healthcare. Current models often

fail to capture the dynamic nature of causal relationships in sequential datasets, limiting their applicability in domains where real-time insights are crucial. Real-world data is rarely stationary, and causal methods need to adapt to changing distributions while maintaining generalizability. Finally, integrating domain expertise with causal models can enhance interpretability and reliability, especially in specialized areas like healthcare and manufacturing.

The development of a framework that combines the strengths of causal inference and neural networks to enhance deep learning performance is shown in Figure 5.1. The framework explores multiple approaches to integrating causal prior knowledge into neural networks [27, 5], alongside the usual training data. This includes leveraging additional prior knowledge provided by an independent source, such as a causal graph, to guide the learning process and enhance model performance. Once a causal model is available, either through external human knowledge or causal discovery methods, causal inference enables both the estimation of intervention impacts and the identification of causal relationships from data.



**Figure 5.1:** The proposed framework integrates causal discovery and inference with neural networks.

Causal inference methods will be used to detect and account for confounding variables such as ATE, Inverse Probability Weighting (IPW), and IVs. The methods should integrate do-operations and causal interventions into deep learning



models to simulate the effects of changes in specific variables, particularly in complex systems with high uncertainty and dynamic conditions. Utilize causal sensitivity analysis to quantify the impact of each variable on the outcome. Sensitivity analysis will be performed through simulations in which variables are systematically manipulated via do-operations to observe changes in predictions [26, 7]. Incorporating causal prior knowledge requires modifications to the loss function, and formulating an appropriate term for the loss function can be complex. Introducing such a term leads to complex optimization problems [2, 1]. Sensitivity analysis is the cornerstone of robust causal inference, providing tools to evaluate the reliability of causal conclusions in the presence of unmeasured confounding. By integrating this method with causal discovery and inference techniques, the framework lays the foundation for scalable, interpretable, and reliable causal models across diverse domains.

## References

- [1] Luiz Chamon and Alejandro Ribeiro. “Probably approximately correct constrained learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16722–16735.
- [2] Luiz FO Chamon et al. “Constrained learning with non-convex losses”. In: *IEEE Transactions on Information Theory* 69.3 (2022), pp. 1739–1760.
- [3] Aditya Chattopadhyay et al. “Neural network attributions: A causal perspective”. In: *International Conference on Machine Learning*. PMLR, 2019, pp. 981–990.
- [4] David Maxwell Chickering. “Optimal structure identification with greedy search”. In: *Journal of machine learning research* 3.Nov (2002), pp. 507–554.
- [5] Tirtharaj Dash et al. “A review of some techniques for inclusion of domain-knowledge into deep neural networks”. In: *Scientific Reports* 12.1 (2022), p. 1040.

- [6] Iván Díaz and Mark J van der Laan. “Sensitivity analysis for causal inference under unmeasured confounding and measurement error problems”. In: *The international journal of biostatistics* 9.2 (2013), pp. 149–160.
- [7] Gabriel-Radu Frumusanu, Cezarina Afteni, and Alexandru Epureanu. “Data-Driven Causal Modeling of the Manufacturing System”. In: *Transactions of FAMENA* 45.1 (2021), pp. 43–62.
- [8] Christopher Hagedorn, Johannes Huegle, and Rainer Schlosser. “Understanding unforeseen production downtimes in manufacturing processes using log data-driven causal reasoning”. In: *Journal of Intelligent Manufacturing* 33.7 (2022), pp. 2027–2043.
- [9] Jean Kaddour et al. “Causal machine learning: A survey and open problems”. In: *arXiv preprint arXiv:2206.15475* (2022).
- [10] Marco Lippi et al. “Enabling causality learning in smart factories with hierarchical digital twins”. In: *Computers in Industry* 148 (2023), p. 103892.
- [11] Ruonan Liu et al. “Causal intervention graph neural network for fault diagnosis of complex industrial processes”. In: *Reliability Engineering & System Safety* 251 (2024), p. 110328.
- [12] Katerina Marazopoulou et al. “Causal discovery for manufacturing domains”. In: *arXiv preprint arXiv:1605.04056* (2016).
- [13] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [14] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.
- [15] Josephine Rehak, Shahenda Youssef, and Jürgen Beyerer. “Root cause analysis using anomaly detection and temporal informed causal graphs”. In: *ML4CPS—Machine Learning for Cyber-Physical Systems*. 2024.
- [16] Jakob Runge et al. “Causal inference for time series”. In: *Nature Reviews Earth & Environment* 4.7 (2023), pp. 487–505.
- [17] Jakob Runge et al. “Detecting and quantifying causal associations in large nonlinear time series datasets”. In: *Science advances* 5.11 (2019), eaau4996.

- [18] Bernhard Schölkopf. “Causality for machine learning”. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022, pp. 765–804.
- [19] Bernhard Schölkopf et al. “Toward causal representation learning”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.
- [20] Philipp Schwarz et al. “Management Decisions in Manufacturing using Causal Machine Learning–To Rework, or not to Rework?” In: *arXiv preprint arXiv:2406.11308* (2024).
- [21] Alex Tank et al. “Neural granger causality”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.8 (2021), pp. 4267–4279.
- [22] Matej Vuković and Stefan Thalmann. “Causal discovery in manufacturing: A structured literature review”. In: *Journal of Manufacturing and Materials Processing* 6.1 (2022), p. 10.
- [23] Sheng Wang et al. “Root cause diagnosis for process faults based on multisensor time-series causality discovery”. In: *Journal of Process Control* 122 (2023), pp. 27–40.
- [24] Yuchun Wu et al. “A quantitative causal analysis and optimization framework for inclusions of steel products”. In: *Advanced Engineering Informatics* 62 (2024), p. 102629.
- [25] Shu Yang and B Wayne Bequette. “Observational process data analytics using causal inference”. In: *AIChE Journal* 69.4 (2023), e17986.
- [26] Samuel Yousefi, Mustafa Jahangoshai Rezaee, and Armin Moradi. “Causal effect analysis of logistics processes risks in manufacturing industries using sequential multi-stage fuzzy cognitive map: a case study”. In: *International Journal of Computer Integrated Manufacturing* 33.10-11 (2020), pp. 1055–1075.
- [27] Shahenda Youssef. “Incorporating Causal Prior Knowledge into Deep Neural Networks”. In: *Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*. 2023, p. 93.
- [28] Shahenda Youssef, Frank Doehner, and Jürgen Beyerer. “Regression via causally informed Neural Networks”. In: *MLACPS–Machine Learning for Cyber-Physical Systems*. 2024.

- [29] Chaobo Zhang et al. “Causal discovery and inference-based fault detection and diagnosis method for heating, ventilation and air conditioning systems”. In: *Building and Environment* 212 (2022), p. 108760.
- [30] Xiaoge Zhang et al. “Enhancing the Performance of Neural Networks Through Causal Discovery and Integration of Domain Knowledge”. In: *arXiv preprint arXiv:2311.17303* (2023).





## Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

---

- Band 1** Jürgen Geisler  
**Leistung des Menschen am Bildschirmarbeitsplatz.**  
ISBN 3-86644-070-7
- Band 2** Elisabeth Peinsipp-Byma  
**Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse.** 2007  
ISBN 978-3-86644-149-1
- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)  
**Mensch-Maschine-Systeme.**  
ISBN 978-3-86644-457-7
- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)  
**Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-86644-469-0
- Band 5** Thomas Usländer  
**Service-oriented design of environmental information systems.**  
ISBN 978-3-86644-499-7
- Band 6** Giulio Milighetti  
**Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter.**  
ISBN 978-3-86644-568-0
- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)  
**Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-86644-609-0
- Band 8** Eduardo Monari  
**Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken.**  
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader  
**Multimodale Interaktion in Multi-Display-Umgebungen.**  
ISBN 3-86644-760-8
- Band 10** Christian Frese  
**Planung kooperativer Fahrmanöver für kognitive Automobile.**  
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)  
**Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen  
**Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES).**  
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)  
**Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts  
**Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip.**  
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert  
**Data-driven Methods for Fault Localization in Process Technology.** 2013  
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer  
**Probabilistische Szenenmodelle für die Luftbildauswertung.**  
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)  
**Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0212-8



- Band 18** Michael Teutsch  
**Moving Object Detection and Segmentation for Remote Aerial Video Surveillance.**  
ISBN 978-3-7315-0320-0
- Band 19** Marco Huber  
**Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications.**  
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)  
**Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov  
**Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen.**  
ISBN 978-3-7315-0426-9
- Band 22** Benjamin Kühn  
**Interessengetriebene audiovisuelle Szenenexploration.**  
ISBN 978-3-7315-0457-3
- Band 23** Yvonne Fischer  
**Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung.**  
ISBN 978-3-7315-0460-3
- Band 24** Jürgen Beyerer, Alexey Pak (Hrsg.)  
**Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0519-8
- Band 25** Pascal Birnstill  
**Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement.**  
ISBN 978-3-7315-0538-9
- Band 26** Philipp Woock  
**Umgebungskartenschätzung aus Sidescan-Sonar-daten für ein autonomes Unterwasserfahrzeug.**  
ISBN 978-3-7315-0541-9

- Band 27** Janko Petereit  
**Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments.**  
ISBN 978-3-7315-0580-8
- Band 28** Erik Ludwig Krempel  
**Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen.**  
ISBN 978-3-7315-0598-3
- Band 29** Jürgen Moßgraber  
**Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017**  
ISBN 978-3-7315-0638-6
- Band 30** Andrey Belkin  
**World Modeling for Intelligent Autonomous Systems.**  
ISBN 978-3-7315-0641-6
- Band 31** Chettapong Janya-Anurak  
**Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos.**  
ISBN 978-3-7315-0642-3
- Band 32** David Münch  
**Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information.**  
ISBN 978-3-7315-0644-7
- Band 33** Jürgen Beyerer, Alexey Pak (Eds.)  
**Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0678-2
- Band 34** Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)  
**Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0779-6
- Band 35** Michael Grinberg  
**Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking.**  
ISBN 978-3-7315-0781-9

- Band 36** Christian Herrmann  
**Video-to-Video Face Recognition for Low-Quality Surveillance Data.**  
ISBN 978-3-7315-0799-4
- Band 37** Chengchao Qu  
**Facial Texture Super-Resolution by Fitting 3D Face Models.**  
ISBN 978-3-7315-0828-1
- Band 38** Miriam Ruf  
**Geometrie und Topologie von Trajektorienoptimierung für vollautomatisches Fahren.**  
ISBN 978-3-7315-0832-8
- Band 39** Angelika Zube  
**Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung.**  
ISBN 978-3-7315-0855-7
- Band 40** Jürgen Beyerer and Miro Taphanel (Eds.)  
**Proceedings of the 2018 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-0936-3
- Band 41** Marco Thomas Gewohn  
**Ein methodischer Beitrag zur hybriden Regelung der Produktionsqualität in der Fahrzeugmontage.**  
ISBN 978-3-7315-0893-9
- Band 42** Tianyi Guan  
**Predictive energy-efficient motion trajectory optimization of electric vehicles.**  
ISBN 978-3-7315-0978-3
- Band 43** Jürgen Metzler  
**Robuste Detektion, Verfolgung und Wiedererkennung von Personen in Videodaten mit niedriger Auflösung.**  
ISBN 978-3-7315-0968-4
- Band 44** Sebastian Bullinger  
**Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion.**  
ISBN 978-3-7315-1012-3

- Band 45** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2019 Joint Workshop of  
Fraunhofer IOSB and Institute for Anthropomatics,  
Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1028-4
- Band 46** Stefan Becker  
**Dynamic Switching State Systems for Visual Tracking.**  
ISBN 978-3-7315-1038-3
- Band 47** Jennifer Sander  
**Ansätze zur lokalen Bayes'schen Fusion von  
Informationsbeiträgen heterogener Quellen.**  
ISBN 978-3-7315-1062-8
- Band 48** Philipp Christoph Sebastian Bier  
**Umsetzung des datenschutzrechtlichen Auskunftsanspruchs  
auf Grundlage von Usage-Control und Data-Provenance-  
Technologien.**  
ISBN 978-3-7315-1082-6
- Band 49** Thomas Emter  
**Integrierte Multi-Sensor-Fusion für die simultane  
Lokalisierung und Kartenerstellung für mobile  
Robotersysteme.**  
ISBN 978-3-7315-1074-1
- Band 50** Patrick Dunau  
**Tracking von Menschen und menschlichen Zuständen.**  
ISBN 978-3-7315-1086-4
- Band 51** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2020 Joint Workshop of  
Fraunhofer IOSB and Institute for Anthropomatics,  
Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1091-8
- Band 52** Lars Wilko Sommer  
**Deep Learning based Vehicle Detection in Aerial Imagery.**  
ISBN 978-3-7315-1113-7
- Band 53** Jan Hendrik Hammer  
**Interaktionstechniken für mobile Augmented-Reality-  
Anwendungen basierend auf Blick- und Handbewegungen.**  
ISBN 978-3-7315-1169-4

- Band 54** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2021 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1171-7
- Band 55** Ronny Hug  
**Probabilistic Parametric Curves for Sequence Modeling.**  
ISBN 978-3-7315-1198-4
- Band 56** Florian Patzer  
**Automatisierte, minimalinvasive Sicherheitsanalyse und Vorfalreaktion für industrielle Systeme.**  
ISBN 978-3-7315-1207-3
- Band 57** Achim Christian Kuwertz  
**Adaptive Umweltmodellierung für kognitive Systeme in offener Welt durch dynamische Konzepte und quantitative Modellbewertung.**  
ISBN 978-3-7315-1219-6
- Band 58** Julius Pfrommer  
**Distributed Planning for Self-Organizing Production Systems.**  
ISBN 978-3-7315-1253-0
- Band 59** Ankush Meshram  
**Self-learning Anomaly Detection in Industrial Production.**  
ISBN 978-3-7315-1257-8
- Band 60** Patrick Philipp  
**Über die Formalisierung und Analyse medizinischer Prozesse im Kontext von Expertenwissen und künstlicher Intelligenz.**  
ISBN 978-3-7315-1289-9
- Band 61** Mathias Anneken  
**Anomaliedetektion in räumlich-zeitlichen Datensätzen.**  
ISBN 978-3-7315-1300-1
- Band 62** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2022 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1304-9

- Band 63** Fabian Dürr  
**Multimodal Panoptic Segmentation of 3D Point Clouds.**  
ISBN 978-3-7315-1314-8
- Band 64** Jutta Hild  
**Nutzung von Blickbewegungen für die Mensch-Computer-Interaktion mit dynamischen Bildinhalten am Beispiel der Videobildauswertung.**  
ISBN 978-3-7315-1330-8
- Band 65** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2023 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1351-3
- Band 66** Tobias Michael Kalb  
**Principles of Catastrophic Forgetting for Continual Semantic Segmentation in Automated Driving.**  
ISBN 978-3-7315-1373-5
- Band 67** Arno Appenzeller  
**Datensouveränität für Betroffene über persönliche medizinische Daten durch technische Umsetzung einer datenschutzgerechten Forschungsplattform.**  
ISBN 978-3-7315-1377-3
- Band 68** Paul Georg Wagner  
**Trustworthy Distributed Usage Control Enforcement in Heterogeneous Trusted Computing Environments.**  
ISBN 978-3-7315-1390-2
- Band 69** Anne Borcharding  
**Use of Accessible Information to Improve Industrial Security Testing.**  
ISBN 978-3-7315-1400-8
- Band 70** Jürgen Beyerer, Tim Zander (Eds.)  
**Proceedings of the 2024 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.**  
ISBN 978-3-7315-1423-7



Lehrstuhl für Interaktive Echtzeitsysteme  
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik  
und Bildauswertung IOSB Karlsruhe

In 2024, the annual joint workshop of the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) and the Vision and Fusion Laboratory (IES) at the Karlsruhe Institute of Technology (KIT) was once again hosted in a Black Forest house near Triberg. From the 28th of July to the 3rd of August, doctoral students from both institutions presented extensive reports on their research and engaged in discussions on topics ranging from computer vision, industrial production, optimization, control theory, security, to large language models. The results and ideas presented are collected in this book as detailed technical reports, providing a comprehensive overview of the research programs of the IES Laboratory and the Fraunhofer IOSB.

ISSN 1863-6489  
ISBN 978-3-7315-1423-7

