Computing and Software for Big Science (2025) 9:6
https://doi.org/10.1007/s41781-025-00135-6

RESEARCH

End-to-End Multi-track Reconstruction Using Graph Neural Networks at Belle II

L. Reuter[1] · G. De Pietro[2] · S. Stefkova[3] · T. Ferber[4] · V. Bertacchi[5] · G. Casarosa[6] · L. Corona[7] · P. Ecker[8] · A. Glazov[9] · Y. Han[10] · M. Laurenza[11] · T. Lueck[12] · L. Massaccesi[13] · S. Mondal[14] · B. Scavino[15] · S. Spataro[16] · C. Wessel[17] · L. Zani[18]

Received: 28 November 2024 / Accepted: 16 March 2025

Abstract
We present the study of an end-to-end multi-track reconstruction algorithm for the central drift chamber of the Belle II experiment at the SuperKEKB collider using Graph Neural Networks for an unknown number of particles. The algorithm uses detector hits as inputs without pre-filtering to simultaneously predict the number of track candidates in an event and their kinematic properties. In a second step, we cluster detector hits for each track candidate to pass to a track fitting algorithm. Using a realistic full detector simulation including beam-induced backgrounds and detector noise taken from actual collision data, we find significant improvements in track finding efficiencies for tracks in a variety of different event topologies compared to the existing baseline algorithm used in Belle II. For events involving a hypothetical long-lived massive particle with a mass in the GeV-range, decaying uniformly along its flight direction into two charged particles, the GNN achieves a combined track finding and fitting charge efficiency of 85.4% per track, with a fake rate of 2.5%, averaged over the full detector acceptance. In comparison, the baseline algorithm achieves 52.2% efficiency and a fake rate of 4.1%. This is the first end-to-end multi-track machine learning algorithm for a drift chamber detector that has been utilized in a realistic particle physics environment.

Keywords Track finding · Tracking · Object condensation · Machine learning · Graph neural networks · Deep learning · End-to-end representation spaces

Introduction

Experimental particle physics experiments rely on the measurement of charged particles' kinematics, namely, their production point location and their momenta at the production point. These measurements are performed by tracking detectors that provide position measurements of energy depositions (or detector hits) left by charged particles ionizing material along their trajectories, commonly named tracks. In this paper, we describe a new track finding algorithm *CAT Finder* (*CDC AI Track*) using Graph Neural Networks (GNNs) in the Belle II central drift chamber (CDC) to reconstruct charged tracks in electron–positron collisions. The *CAT Finder* simultaneously detects an unknown number of objects, and infers their momenta and their point of origin. In a second step, we associate detector hits to each of those objects that are then used as starting point for a subsequent conventional track fitting algorithm. We find significant improvements in track finding efficiencies for displaced tracks that originate from a position separated from the interaction point by a macroscopic distance of a few centimetres up to a meter. Such particles appear in theories beyond the Standard Model from decays of long-lived neutral mediators like dark photons [1], or even displaced decay vertices that involve invisible particles in addition to a pair of charged particles [2, 3]. At the same time, the efficiency and fake rate for prompt tracks from the electron–positron interaction point (IP) is comparable to established methods in the central detector region but significantly better in the forward and backward detector regions.

The Belle II experiment is located at the high-intensity asymmetric e^+e^- collider SuperKEKB in Tsukuba, Japan. SuperKEKB collides primarily 4 GeV positron and 7 GeV electron beams at a center-of-mass energy of the $\Upsilon(4S)$ resonance at approximately 10.58 GeV to investigate rare B-meson decays and new physics phenomena. To achieve a

Extended author information available on the last page of the article

Published online: 03 May 2025

higher sensitivity to very rare processes at the Belle II experiment, SuperKEKB has the goal of increasing the instantaneous luminosity significantly compared to its predecessor, KEKB. However, this increase in luminosity also results in a significant increase in beam-induced background (called beam background in the following) that manifests in a high number of background detector hits, and a large number of charged and neutral background particles not originating from the IP [4]. On average an $\Upsilon(4S) \rightarrow B\bar{B}$ decay in Belle II will produce about 11 charged particles that typically feature momenta ranging from tens of MeV to a few GeV, while most direct searches for new physics feature lower charged track multiplicities.

This paper is organized as follows: Sect. "Related Work" gives an overview of related work on machine learning (ML) for track finding and end-to-end reconstruction. Section "The Belle II Central Drift Chamber Tracking Detector" describes the Belle II central drift chamber. The event simulation and details of the beam background simulation and data are reported in section "Data Set". The metrics used to quantify track finding and track fitting performance are defined in Sect. "Metrics". The existing and our new GNN-based track reconstruction algorithms are described in Sect. "Track Reconstruction Algorithms". The main performance studies are discussed in Sect. "Results". The results are summarized in Sect. "Conclusion and Outlook".

Related Work

While machine learning is widely used in high energy physics for event selection and analysis, the computationally intensive task of track reconstruction still largely relies on traditional algorithms. GNNs are recognized as a potential solution for handling irregular detector structures in high energy physics [5]. GNN architectures in particular have the ability to learn a latent space representation of the detector structure itself [6, 7], which is a key ingredient of the work presented in this paper and which has been applied to the Belle II calorimeter reconstruction [8]. These architectures have been proven highly effective in handling the complex and irregular spatial structures of particle detectors, enabling more accurate and efficient analysis of high energy physics data. Graph segmentation in our work relies on object condensation [9] which has been used, e.g., in end-to-end calorimeter reconstruction studies for the CMS HGCAL [10].

In the context of LHC track reconstruction in high pile-up events, the TrackML challenge [11, 12] has led to a significant increase of development activities in the area of ML-based track reconstruction [13–16]. The events from the TrackML challenge have a significantly higher number of tracks compared to Belle II, a higher fraction of sensor hits belonging to signal tracks, and simpler track kinematics from particles produced at the interaction point with high transverse momentum. GNN-based tracking pipelines aiming for rather detector-unspecific solutions have been developed by the Exa.TrkX project for the HL-LHC [15], and by the ETX4VELO project for LHCb [17]. Previous work has usually focused on simplified and idealized detector structures and simulations, and on tracks without significant displacement.

Comparisons of GNN track finding to conventional algorithms in realistic HL-LHC scenarios have been shown for the ATLAS inner tracking pixel detector [18].

GNNs for gaseous detectors have been studied for edge classification for the PANDA experiment [19] and BES III [20]. Utilizing deep learning techniques of semantic segmentation inspired by so-called U-Nets, hit classification in high background environments has been demonstrated for the drift chamber of the COMET experiment [21]. However, none of these works feature end-to-end ML-based solutions or conclusive studies of complex event typologies. Additional challenges arise from the differing input features between drift chambers as used in Belle II, and the silicon pixels and strips used in HL-LHC. HL-LHC detectors, such as silicon pixel detectors [16, 22] that provide 3D spatial information differ from drift chambers, which rely on indirect measurements of drift time and wire positions. This makes track reconstruction in drift chambers more complex due to the lack of direct spatial information. In addition to GNNs, a wide range of ML-algorithms are currently being investigating for usage in track reconstruction, including, e.g., large language models [23] or transformers [24].

Modern implementations of traditional track reconstruction algorithms are often enhanced with ML methods for specific tasks. For instance, the Belle II experiment incorporates gradient boosted decision trees [25] into its track reconstruction pipeline to improve beam background filtering and track-candidate search, and feed-forward neural networks for real-time reconstruction of the z position in the CDC [26].

For an up-to-date list of works in particle physics that utilize ML, we refer to the living review [27].

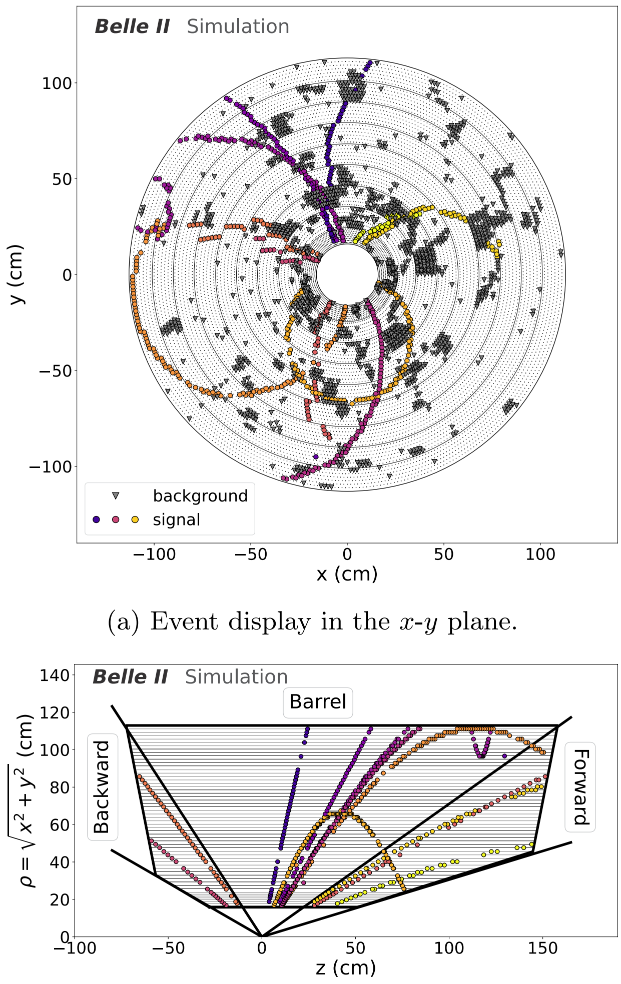
The Belle II Central Drift Chamber Tracking Detector

The Belle II detector is a charged particle spectrometer surrounded by particle-identification detectors, an electromagnetic calorimeter, and a K_L^0 and muon detector, arranged around the beam pipe in a cylindrical structure [28]. The positive z direction is pointing in the direction of the electron beam. The x axis is horizontal and points away from the accelerator center, while the y axis is vertical and points upwards. The longitudinal direction, the transverse plane with azimuthal angle ϕ, and the polar angle θ are defined with respect to the detector's solenoid axis.

The charged particle spectrometer consists of silicon-based pixel and silicon-strip detectors that are not used for this work, and a gas-filled CDC. The CDC is 2.3 m long, and has in total 14,336 sense wires and about 36,000 field wires forming drift cells with a size of about $1 \times 1\,\mathrm{cm}^2$ in the inner wire layers, to about $2 \times 2\,\mathrm{cm}^2$ elsewhere. The CDC covers the polar angle range $17° < \theta < 150°$ and the full azimuthal angle ϕ range. Particles with a polar angle between $17° < \theta < 35.4°$ leave the CDC early in the forward endcap, particles with $35.40° < \theta < 123.04°$ traverse the full detector, defined as barrel region, and particles $123.04° < \theta < 150°$ leave the CDC early in the backward endcap. The sense wires cover a radius between about 17 to 110 cm, and are arranged in 56 layers that are grouped in nine superlayers: the innermost superlayer consists of 8 layers with 160 sense wires each; the outer eight superlayers consist of 6 layers with 160 to 384 sense wires each. All superlayers alternate between wires aligned with the solenoid magnetic field, called axial layers A, and superlayers skewed by an angle between 66.8 and 74.1 mrad in the positive and −58 to −78.6 mrad in the negative direction, called stereo layers U and V. The resulting superlayer arrangement, numbered from inward to outward, is A1, U2, A3, V4, A5, U6, A7, V8, A9. The electrical field in each drift cell is approximately radial. The drift distance resolution of the CDC is about 120 µm.

A superconducting solenoid generates a magnetic field of about 1.5 T, which is directed along the central axis of the CDC support cylinder. Near the IP, there is a system of final focusing quadrupole and compensating solenoid magnets. The magnetic field is relatively uniform, with variations of approximately 1% throughout the entire tracking volume. More details can be found in Ref. [28–30].

The CDC provides spatial information in the plane perpendicular to the sense wire axis, which aligns with the Belle II $x - y$ plane for the axial layers and is tilted by their respective skew angles in the stereo layers. There is no information available for the z-coordinate on the wires, because the readout is only done on one side. This spatial information is encoded as the wire position and the signal's drift time (TDC), recorded when the energy deposition crosses the readout threshold, relative to an unknown global (common for all hits) time offset at the start of track finding. In addition, the digitized signal amplitude (ADC) and the time over the readout threshold (TOT) are also provided. The digitized signal amplitude is proportional to the energy deposition of a particle. A typical event display for the CDC is shown Fig. 1.



(a) Event display in the x-y plane.

(b) Event display in the z-ρ-plane.

Fig. 1 Typical event display in the x–y plane (**a**) and the z–ρ plane (**b**) for a simulated $\Upsilon(4S) \rightarrow B^+B^-$ event with *high data beam backgrounds*. In the x–y plane, filled colored circular markers represent signal hits, while filled gray triangular markers represent background hits. These markers correspond to the locations of the sense wires at the z position of the wire center, for wires with recorded ADC signals. In the z–ρ plane, where $\rho = \sqrt{x^2 + y^2}$, only the signal hits are shown. The three detector regions, forward endcap, barrel, and backward endcap, are also indicated in the z–ρ plane

Data Set

We use simulated Belle II events for the training and evaluation of the reconstruction algorithms. The full detector geometry and interactions of final state particles with detector material are simulated using `GEANT4` [31], which is combined with the simulation of a detector response to create digitized detector hits using the Belle II Analysis Software Framework `basf2` [32, 33].

There are three key signatures with qualitatively different behaviour relevant for tracking:

1. Low transverse momentum tracks forming circles in the CDC ($p_t \lesssim 0.4$ GeV) versus high momentum tracks moving straight through the CDC ($p_t > 0.4$ GeV);
2. Particles traversing all CDC layers versus those exiting through the endcaps, creating shorter tracks;
3. Decay vertices where the decay particles have a small opening angle with potentially overlapping tracks, versus those with a larger opening angle and well isolated tracks.

To effectively train our model on a comprehensive physics phase space, we utilize samples that do not follow conservation laws, but instead are drawn from a parameter space as defined in Table 1. To ensure a sufficient number of events with challenging signatures, we enriched our samples by generating events in the direction of the endcaps, including low momentum particles, and those with very small opening angles. In addition, we included a transition sample between displaced vertices and prompt tracks. This sample increased the tracking performance for displaced vertex samples and provided a faster model convergence. In general, the model performance improved for continuous transitions between different topologies in comparison to strictly independent topologies, which is the reason we also included a sample, where we combine all of the above signatures together in single events in the mix.

All events in categories 1–11 feature muons as primary charged particles, with their charges randomly chosen with equal probability. The *displaced* samples (categories 4–7) have a starting point v_x, v_y, v_z, that is displaced in 3D in the momentum direction of the respective charged particle. For the *displaced angled* sample (category 7) we generate a new momentum direction with a rotation angle α^{3D}_{gen} with respect to the vector connecting the origin and the starting point, along a randomly selected perpendicular direction. The *prompt* and *displaced* samples are generated independently in the forward (fwd), barrel (brl), and backward (bwd) detector regions as well as the full detector acceptance region based on the particle's polar angle at their production point. The *vertex* samples (categories 9 and 10) consist of two displaced angled particles with opposite charge, generated at the same starting point. This approach covers both large (category 9) and small (category 10) opening angles, effectively enriching the training sample with events with small opening angles, for which the reconstruction is more difficult.

Table 1 Event samples used for training and validation

Category	Name	θ_{gen} [°]	r^{3D}_{gen} [cm]	α^{3D}_{gen} [°]
1	Prompt fwd	17.0–35.4	0	0
2	Prompt brl	35.4–123.04	0	0
3	Prompt bwd	123.04–150.0	0	0
4	Displaced fwd	17.0–35.4	0–100	0
5	Displaced brl	35.4–123.04	0–100	0
6	Displaced bwd	123.04–150.0	0–100	0
7	Displaced angled	17.0–150.0	0–100	0–30
8	Prompt full	17.0–150.0	0–100	0
9	Vertex large	17.0–150.0	0–100	0–90
10	Vertex small	17.0–150.0	0–100	0–25
11	Mix 8+10	–	-	–

See text for details

The generated quantities $p_{\text{t,gen}}$ (prompt samples), p_{gen} (displaced and vertex samples), θ_{gen}, ϕ_{gen}, r^{3D}_{gen}, α^{3D}_{gen} are drawn randomly from independent uniform distributions for each charged particle. The displacement

$$r^{3D}_{\text{gen}} = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{1}$$

is calculated in 3D and not just in the plane transverse to the z-axis. Each event in category 1–8 contains 1–6 charged particles, this number is drawn from an independent uniform distribution. Each sample of category 1–8 contains 60,000 events with $0.05 < p_t < 6.0$ GeV (categories 1–3 and 8) and $0.05 < p < 6.0$ GeV (categories 4–7). To enrich the events in categories 1–3 and 8 with low momentum particles, we add a random number of prompt low momentum charged particles with $0.05 < p_t < 0.4$ GeV and all other quantities as above to each event. For the events in categories 4–7 we enrich with displaced low momentum charged particles with $0.05 < p < 0.4$ GeV. The number of low momentum charged particles is drawn from a Poisson distribution with mean $\lambda = 1$. On average the events contain 4.5 particles, resulting in 276,000 particles for each sample in categories 1–8.

The events in categories 9 and 10 contain two, four or six charged particles, where the number is drawn from an independent uniform distribution. Example event displays of the different event categories are shown in Appendix A. Each sample of categories 9 and 10 contains 120,000 events with 480,000 particles. This results in 240,000 events in each major category-group of prompt (categories 1–3 and category 8), displaced (categories 4–7), and vertex events (categories 9 and 10).

Category 11 contains a mix of tracks from category 8 and category 10. For this we generate a number of charged particles drawn from a Poisson distribution with $\lambda = 1.5$, and enrich the sample with low momentum particles (see above) drawn from a Poisson distribution with $\lambda = 1.5$. We finally add vertex events with small opening angles, where the number of decay vertices is drawn from a Poisson distribution with $\lambda = 1.5$. This sample contains 300,000 events. With this training setup, we observe a maximum of 15 charged particles per event in our training and evaluation categories.

For the evaluation in section "Results" we use three additional samples that are not used for the GNN training. We generate radiative muon pairs $e^+e^- \to \mu^+\mu^-(\gamma)$ using the KKMC event generator [34]. We generate dark Higgs $e^+e^- \to A'h(\to \mu^+\mu^-)$ events using MadGraph5@NLO [35] with an inelastic dark matter model [3] with on-shell two-body kinematics, with one dark Higgs $h \to \mu^+\mu^-$, and a fully invisible decay of a light dark photon A' with the dark Higgs masses $m_h = [0.5, 2.0, 4.0]$ GeV. The dark Higgs decay vertex position is drawn randomly from a uniform r^{3D}_{gen}-distribution to populate the parameter space of very displaced vertices. We generate neutral kaon $K^0_S \to \pi^+\pi^-$ events containing one K^0_S each. The K^0_S decay vertex is calculated from the nominal K^0_S lifetime [36] with a uniformly generated transverse momentum of $p_t(K^0_S) = [0.05 - 3]$ GeV. The average transverse decay distance is $v_\rho = 8.24$ cm.

As part of the simulation, we overlay randomly triggered events from data with a very low probability of containing actual collision events. The overlay events are taken from the last data-taking period of run I and correspond to high beam backgrounds (*high data beam backgrounds*) recorded at an instantaneous luminosity of about $\mathcal{L}_{\text{beam}} = 3.53 \times 10^{34}\ \text{cm}^{-2}\ \text{s}^{-1}$.

To speed-up the training using pre-trained GNN models (see section "Graph Neural Network Track Finding") and to evaluate the robustness of the GNN inference against varying beam background conditions (see section "Robustness to variable detector conditions"), we also use simulated beam background events approximating the collider conditions in 2021 to our signal particles [4, 37] (*low simulated beam backgrounds*). The simulated beam backgrounds correspond to an instantaneous luminosity of $\mathcal{L}_{\text{beam}} = 1.06 \times 10^{34}\ \text{cm}^{-2}\ \text{s}^{-1}$. In addition, we include *cross-talk* noise simulation to model the behavior of the CDC readout chips, where neighboring channels may be triggered by a large charge deposit in an adjacent channel. *Beam backgrounds* can either leave track signatures, or single wire hits due to low-energy photon conversions, with electron–positron pairs trapped in the magnetic field. *Cross-talk* hits on the other hand typically leave extended cluster-like hit patterns as visible in Fig. 1. The low simulated beam backgrounds contain on average 370 hits not belonging to signal particles, while the high data beam backgrounds contain on average 1230 such hits per event. We include inactive signal wires and signal wires with reduced hit-efficiency corresponding to the respective average detector conditions for the two beam background scenarios (see Appendix B).

Each CDC hit is matched to up to one simulated particle which is then used as training label in our supervised learning. If multiple simulated particles deposit energy in the same drift cell, we match to the simulated particle that leaves the first hit in time.

The total number of events in our training and validation sample is 1,120,000 before removal of about 2% of the events that did not contain any particle with enough matched signal hits in the CDC, or because they contained more than 15 particles. We use 80% of our combined sample for training, and 20% for validation of our models. The performance evaluation described in later sections of this paper is performed on statistically independent additional samples that were not used for training or validation. Each of these evaluation samples consist of 90,000 to 150,000 events, resulting in a total of 1 million evaluated events.

Metrics

For the evaluation of the track finding algorithms we first determine the *hit efficiency* and *hit purity* for each found track.

The *hit efficiency* ε_{hit} per track is defined as the number of CDC hits *matched* to a simulated particle and included in a found track, divided by the number of all CDC hits *matched* to the same simulated particle in the whole event:

$$\varepsilon_{\text{hit}} = \frac{n_{\text{hits}}(\text{matched and} \in \text{track})}{n_{\text{hits}}(\text{matched})} \tag{2}$$

A perfect *hit efficiency* is 1.0, indicating that all *matched* CDC hits are included in this track and no other found track contains hits *matched* to this simulated particle.

The *hit purity* $\mathfrak{p}_{\text{hit}}$ per track is defined as the number of CDC hits *matched* to a simulated particle and included in a found track, divided by the number of all CDC hits included in the found track:

$$\mathfrak{p}_{\text{hit}} = \frac{n_{\text{hits}}(\text{matched and} \in \text{track})}{n_{\text{hits}}(\in \text{track})} \tag{3}$$

A perfect *hit purity* is 1.0, indicating that all hits included in the found track are *matched* to the same particle.

We use the *hit efficiency* and *hit purity*, and a minimal number of hits to define if a found track is *related* to a simulated particle: We require $\varepsilon_{\text{hit}} > 0.05$, $\mathfrak{p}_{\text{hit}} > 0.66$, and $n_{\text{hits}}(\in \text{track}) \geq 7$. The hit efficiency criterion is chosen so low to account for tracks that curl inside the tracking volume and leave many hits behind. The hit purity criterion ensures proper matching by requiring that at least 66% of the hits be associated with a single unique particle, even when hits from one track are coming from multiple simulated particles. If more than one found track can be related to the same simulated particle, we choose the found track with the highest hit purity as *matched* to a simulated particle. In this case we call all other tracks *clone tracks*. If a track does not achieve the purity or efficiency requirements, it is defined as a *fake track*. If several tracks have the same hit purity, we choose the track with the highest hit efficiency as the correct match.

We define the *track efficiency* as the ratio of the number of matched tracks (trks) to the number of all simulated particles that are matched to at least one hit:

$$\varepsilon_{\text{trk}} = \frac{n_{\text{trks}}(\text{matched to part.})}{n_{\text{simulated}}(\geq 1 \text{matched hit})}. \tag{4}$$

We define the *track charge efficiency* as the ratio of the number of matched tracks reconstructed with the correct charge, to the number of all simulated particles that are matched to at least one hit:

$$\varepsilon_{\text{trk,ch}} = \frac{n_{\text{trks}}(\text{matched to part., corr. charge})}{n_{\text{simulated}}(\geq 1 \text{matched hit})}. \tag{5}$$

We define the *track finding purity* as the ratio of matched tracks to the number of all found tracks:

$$\mathfrak{p}_{\text{trk}} = \frac{n_{\text{trks}}(\text{matched to part.})}{n_{\text{trks}}}. \tag{6}$$

We define the *clone rate* as the ratio of number of *clone tracks* to the number of all tracks that are related to a particle,

$$\mathfrak{r}_{\text{clone}} = \frac{n_{\text{clone trks}}}{n_{\text{tracks}}(\text{related to part.})}, \tag{7}$$

and the *fake rate* as the ratio of *fake tracks* to the number of all found tracks

$$\mathfrak{r}_{\text{fake}} = \frac{n_{\text{fake trks}}}{n_{\text{trks}}}. \tag{8}$$

We define the *wrong charge rate* as the ratio of number of matched tracks with the wrong charge, to the number of all tracks that are matched to a particle,

$$\mathfrak{r}_{\text{wrong ch.}} = \frac{n_{\text{trks}}(\text{matched to part., wrong ch.})}{n_{\text{trks}}(\text{matched to part.})}. \tag{9}$$

Since tracks may be found, but then fail the track fitting step, we distinguish between *track finding efficiency*, *track charge finding efficiency*, *track finding clone rate*, *track finding fake rate*, and *wrong finding charge rate*, to indicate track objects after track finding, and *track fitting efficiency*, *track charge fitting efficiency*, *track fitting clone rate*, *track fitting fake rate*, and *wrong fitting charge rate*, to indicate tracks after track finding and track fitting. In the following, we will refer to these parameters as performance metrics. We evaluate the normalized residuals of track momentum components p_t and p_z, by comparing the reconstructed parameters with the simulated ones

$$\eta(p_{t,z}) = \frac{p_{t,z\,\text{rec}} - p_{t,z\,\text{simulated}}}{p_{t,z,\,\text{simulated}}} \tag{10}$$

for matched tracks.

These distributions are expected to peak at zero for an unbiased reconstruction.

We then define the resolution $r(p_{t,z})$ for each of these normalized residuals $\eta(p_{t,z})$ as the 68% coverage

$$r(p_{t,z}) = P_{68\%}\left(\left|\eta(p_{t,z}) - P_{50\%}(\eta(p_{t,z}))\right|\right), \tag{11}$$

where P_q is the qth quantile of the distribution of $p_{t,z}$, and $P_{50\%}$ is the median of $\eta(p_{t,z})$ [29]. For a normal distribution, $r(p_{t,z})$ is identical to the standard deviation.

Track Reconstruction Algorithms

Track reconstruction at Belle II is performed in two steps. In the first step track finding algorithms assign tracking detector hits into subsets of hits belonging to the same charged particle. The track finding algorithms also provide a first estimate of the track kinematics. In a second step a dedicated track fitting algorithm is using these starting values and the set of identified hits to perform a track fit. Since the *CAT Finder* performance is optimized using the *track charge finding efficiency* (see section "Metrics"), we first describe the track fitting procedure, then the baseline finder and finally the *CAT Finder*.

Track Fitting

The track finding algorithms need to provide three sets of information to the subsequent track fitting algorithm:

- an initial estimate of the particle kinematics and the particle charge;
- a set of ordered identified hits belonging to this track;
- an initial estimate of the covariance matrix of the track parameters.

The track fitting is performed using Kalman Filter algorithms implemented in GENFIT2 [38–41], that uses a Deterministic Annealing Filter (DAF) to downweight hits far away from the fitted trajectory. It is possible that a found track fails the track fitting if too many hits are rejected by the DAF. In the nominal Belle II reconstruction, three mass hypotheses (pions, kaons, and protons) are used during the fitting step, while the finding process is independent of the mass assumption. The mass hypothesis is used when calculating energy loss and material effects, with the kaon and proton hypotheses improving momentum resolution for kaons and protons. Since this work focuses on pions and muons, we have chosen to use only the pion hypothesis.

Baseline Track Finding

The baseline track-finding algorithm is described in detail in Ref. [29] and implemented in `basf2`. The baseline track-finding algorithm is based on the Legendre transformation [42] with a main focus on tracks that originate from the close proximity of the IP (called *Baseline Finder* in the following). Hits with ADC and TOT values compatible with background hits are removed with minimal loss of signal hits in a pre-processing step. The *Baseline Finder* is rather insensitive to missing hits in a track and starts using axial-layers only to find 2D tracks in the ρ–ϕ-plane. In a second step, hits in stereo layers are added that allow z determination. A cellular automaton [43] is used to find locally connected wire hits into track segments, followed by boosted decisions trees to add missing tracks-segments to tracks found by the *Baseline Finder*. The *Baseline Finder* uses a fit of a two-dimensional circle to all identified axial hits to obtain an initial estimate of the track curvature and hence its transverse momentum. Using this previous 2D fit result, the initial estimate of the track longitudinal momentum is obtained from a linear fit to the track skew line in the ρ–z-plane.

Graph Neural Network Track Finding

In the following section, we first describe the GNN architecture and the model inference of the *CAT Finder*. We then describe the post-processing steps to choose the final track candidates and extract the track parameter information, the hits assigned to each tracks, and the hit ordering.

Graph Neural Network Architecture

Due to the sparsity of the wire hits in the CDC, with an average hit occupancy of up to 15% per event for the high data beam backgrounds, the variable input size, and the non-uniform arrangement of the drift wires, we utilize a GNN architecture. The implementation of this GNN is done in `PyTorch Geometric` [44]. Each node in the graph corresponds to a wire hit. The input features for our model are the x- and y-positions at the z center of the corresponding sense wire, the layer, and superlayer, and layer within a superlayer information, the ADC count, and the TDC count information. Using cartesian coordinates as input features and prediction targets was crucial for the model's performance. Polar coordinates and angle-based predictions caused issues due to the discontinuity when the angle resets from 360° to 0, which the model struggled to learn effectively. In addition, using the track helix radius as a training target proved ineffective, as it becomes nearly infinite for high-momentum, close to straight tracks, and very small for low-momentum tracks, making it impossible to scale the model prediction target to a reasonable range and, therefore, impossible for the model to learn. All input features except the position are normalized by dividing by their maximum to a range between 0 and 1. The ADC count is clipped to a maximum of 600 before normalization to remove the influence of rare anomalous values. For signal hits the majority of ADC counts is between 25 and 300. The position coordinates range between −1.11 and 1.11. One of the targets for our model is the starting position in the same coordinate range. Therefore, we do not scale either the input or the target coordinates.

The objective of our algorithm is to provide the number of tracks, and the three-momentum, the starting point, the charge, and the hits associated with each track. Given the inherent challenge of not knowing the precise number of tracks in advance, we use an object condensation loss [9] for this task.

The architecture of our model is illustrated in Fig. 2 using $N = 4$ GravNet blocks [7]. The output of each GravNet block is used for both the subsequent block and directly for the final layer through concatenation. In a final step, four parallel linear layers are responsible for generating the model's outputs as described in detail below.

Each GravNet block starts with global average pooling [45], where the mean value for each feature of the graph is calculated. This averaged representation is then added to the original individual node features. This pooling technique enables the network to incorporate a collective understanding of the graph, complementing the information from individual nodes. This is followed by a sequence of two linear layers (LL), a batch normalization layer [46] and another linear layer. Each of the linear layers use an exponential linear unit (ELU) activation functions [47]. The GravNet layer is responsible for building the graph and for message passing between nodes. In its first step, the GravNet translates the input features into learned representation spaces encoding spatial information, called S, and learned features F_{LR}. Undirected edges are then built between each node j and its k nearest neighbours in the representation space S using an n-dimensional Euclidean distance $d = \sqrt{\sum_{i=1,\dots,n}(X_{i,j} - X_{i,k})^2}$, where X_j is the position of the jth node, and X_k is the position of the kth node. The learned features of the connected nodes are weighted dependent on their Euclidean distance and then aggregated by a summation, resulting in updated features for each node. These features are concatenated with the initial node features. Following the GravNet layer, the feature extraction process continues with batch normalization. The resulting output is then forwarded to the next GravNet block and directly to the final linear layer, being passed through an extra linear layer (LL2) on this path.

The five output layers in our model serve a dual purpose, addressing both object condensation and parameter

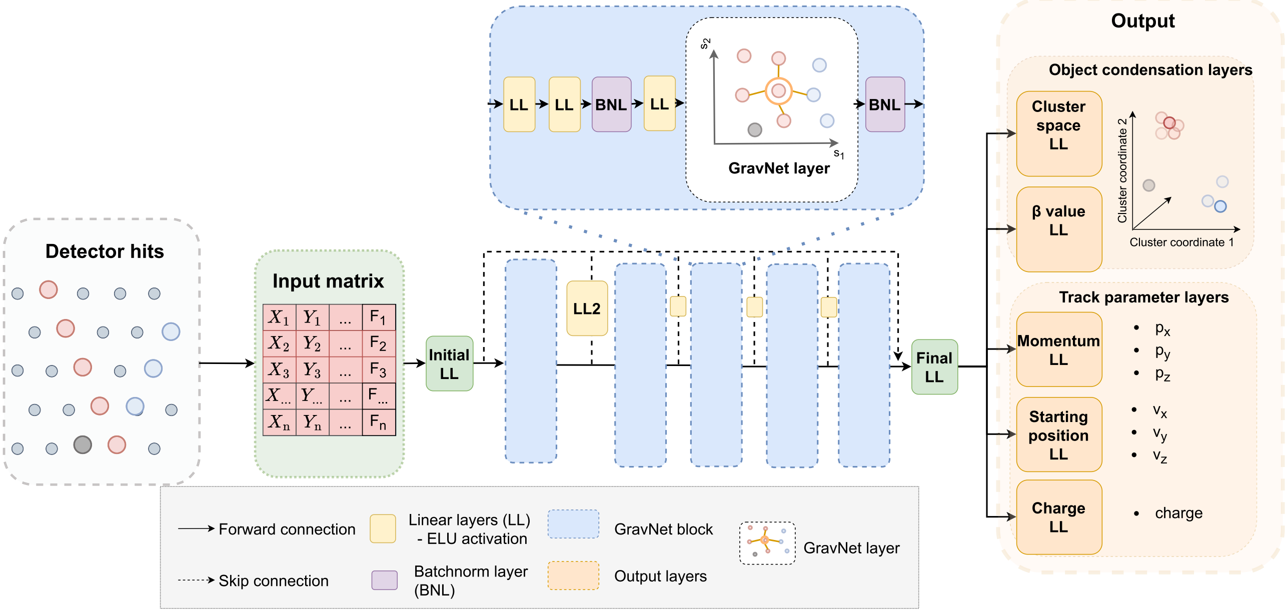


Fig. 2 An illustration of the GNN architecture

prediction tasks. Each node of the graph is assigned one object to identify: if a wire hit is caused by an energy deposition of a signal particle, this node is assigned a unique integer particle ID> 0. In contrast, if the hit is not created by a signal particle, an ID of 0 is assigned. We only use particles that have at least 7 matched hits in the event as signal particles. One output of the object condensation layers is a linear layer with a single node and a sigmoid activation function that generates a single output value β. This β value is used as a measure of a node being a condensation point. Another linear layer with the cluster space dimension of $CS = 3$ output nodes provides coordinates within a learned cluster space for each node. This is also where the model learns to attract the nodes from the same objects together to the node with the highest β value of the object and repels nodes that are from different objects. The primary objective is to have a single node with a high β value per signal particle.

For the track parameter prediction, we use a linear layer with three output nodes for predicting the three-momentum vector components for each node. A second linear layer with again three output nodes is used to predict the track starting position for each node. And finally, a linear layer with one output node and a sigmoid activation function is used for the charge prediction. We apply a threshold of 0.5 on the charge prediction, where predictions that exceed this threshold correspond to a positive charge of 1 and predictions below to a negative charge of -1. The target truth information for these predictions is taken from the simulated particle matched to the node. The loss for the parameter prediction is weighed with the β value, since we only require the actual condensation point to predict the particle parameters, condensing all information about the object on this one node. The total loss is then given by an unweighted sum of the different loss terms for the attraction, repulsion, β value which includes a component to enforce only one condensation point per object and a component to suppress background, and the model parameter predictions. The details of the loss function are described in [9].

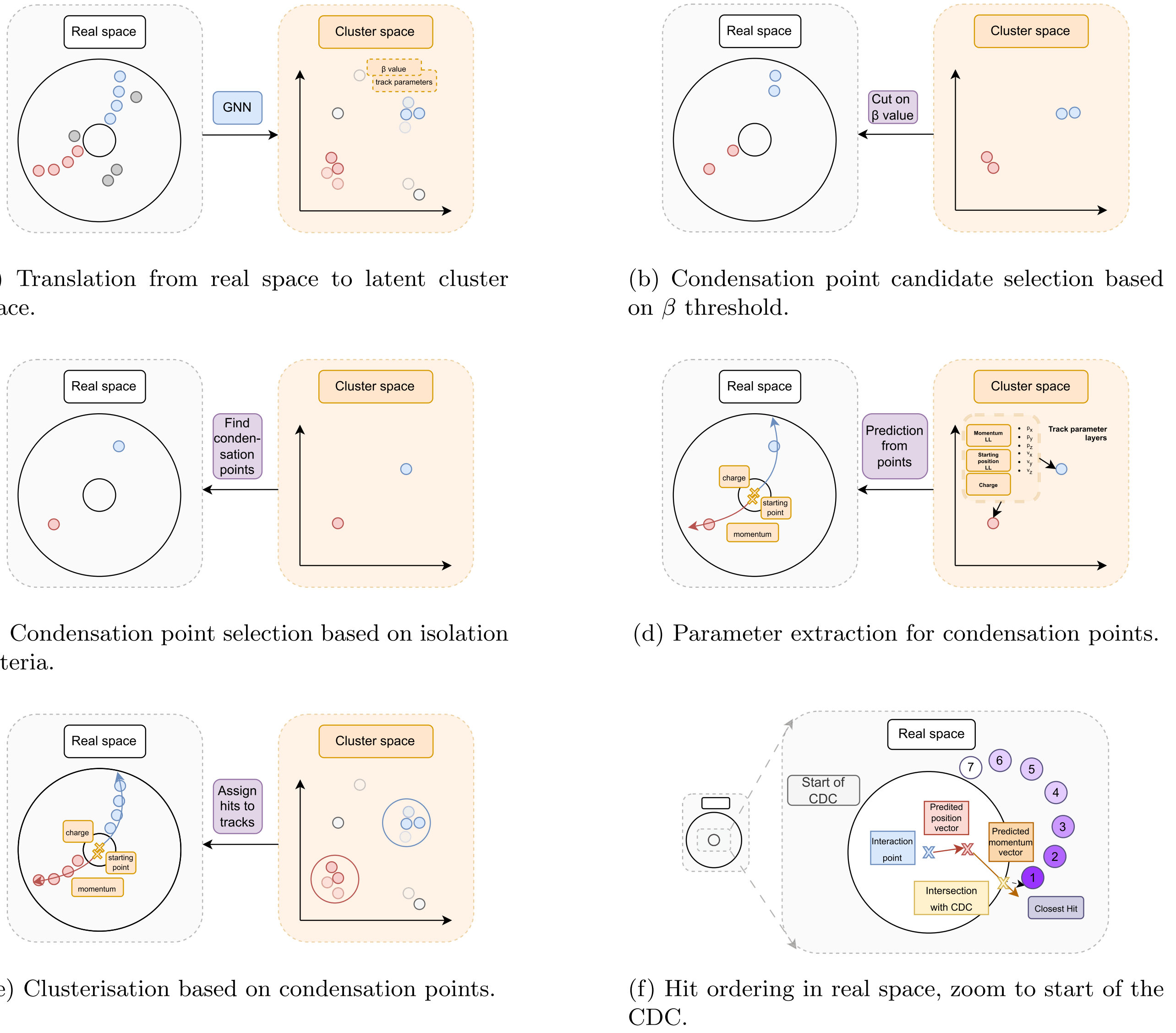
Graph Neural Network Post-processing

The procedure to retrieve the track information from the inference step of the trained model is shown schematically in Fig. 3. The following steps are performed in this order:

(a) Each event is inferred by the model so each node has a predicted position in the latent cluster space, a β value, and parameter predictions for all seven track parameters.

(b) We initiate the track finding process by introducing a threshold t_β, to the β values resulting in condensation point candidates, as illustrated in Fig. 3b.

(c) In the subsequent step we find isolated condensation points among the condensation point candidates: We compute distances r between the condensation point candidate in the event with the highest β value and all other condensation point candidates within the latent cluster space. We introduce distance threshold t_d, so that condensation point candidates located within a radius $r < t_d$ from the candidate with the highest β value are removed. This process iterates until only condensation points remain, each separated by distances



(a) Translation from real space to latent cluster space.

(b) Condensation point candidate selection based on β threshold.

(c) Condensation point selection based on isolation criteria.

(d) Parameter extraction for condensation points.

(e) Clusterisation based on condensation points.

(f) Hit ordering in real space, zoom to start of the CDC.

Fig. 3 Track finding using object condensation: **a** Latent space, **b** condensation point candidate selection based on β threshold, **c** condensation point selection based on isolation, **d** parameter extraction, **e** clustering, and **f** hit ordering in real space

exceeding the radius t_d. The result of these operations is a set of condensation points as shown in Fig. 3c, each corresponding to a found track.

(d) The parameters of each found track are given by the predicted momentum, position, and charge parameters of the respective condensation point (see Fig. 3d).

(e) To assign nodes to each condensation point we cluster nodes in the cluster space: we first calculate the distances between each condensation point and every other node within the cluster space. Any node with $r < t_h$ is assigned to the found condensation point, shown in Fig. 3e. We constrain the $t_h < t_d/2$, ensuring that each hit is used exclusively for at most one track. The found condensation point with its set of nodes is equivalent to a found track with assigned hits. To ensure that the found track can be fitted, we require at least seven hits assigned to the track, otherwise we reject the track and count it as not found.

(f) Finally, we order the assigned hits. The hits are ordered based on their positions in the x–y plane of the detector. The process begins by selecting the hit closest to the predicted starting point if the starting point is within the CDC, or closest to the intersection between the predicted particle direction and the inner CDC cylinder surface if the starting point is before the CDC. We then calculate the Euclidean distance between the starting point x and y to the x and y positions for all hits. Subsequently, the closest hit to the previous one is determined iteratively until all hits are ordered. We note that this procedure is not working for low p_t tracks that re-enter the CDC several times and we discuss these cases in section "Prompt Tracks".

An example event with corresponding learned latent space representation is shown in Fig. 4. Unlike the *Baseline Finder*, the *CAT Finder* does not provide an initial covariance matrix for the track fitting. All initial covariance matrix entries for the *CAT Finder* are set to 0.1. We observe that the track fitting time depends on the initial parameters of the covariance matrix. However, the impact on the track finding efficiency is negligible and we leave performance optimization of this to future work.

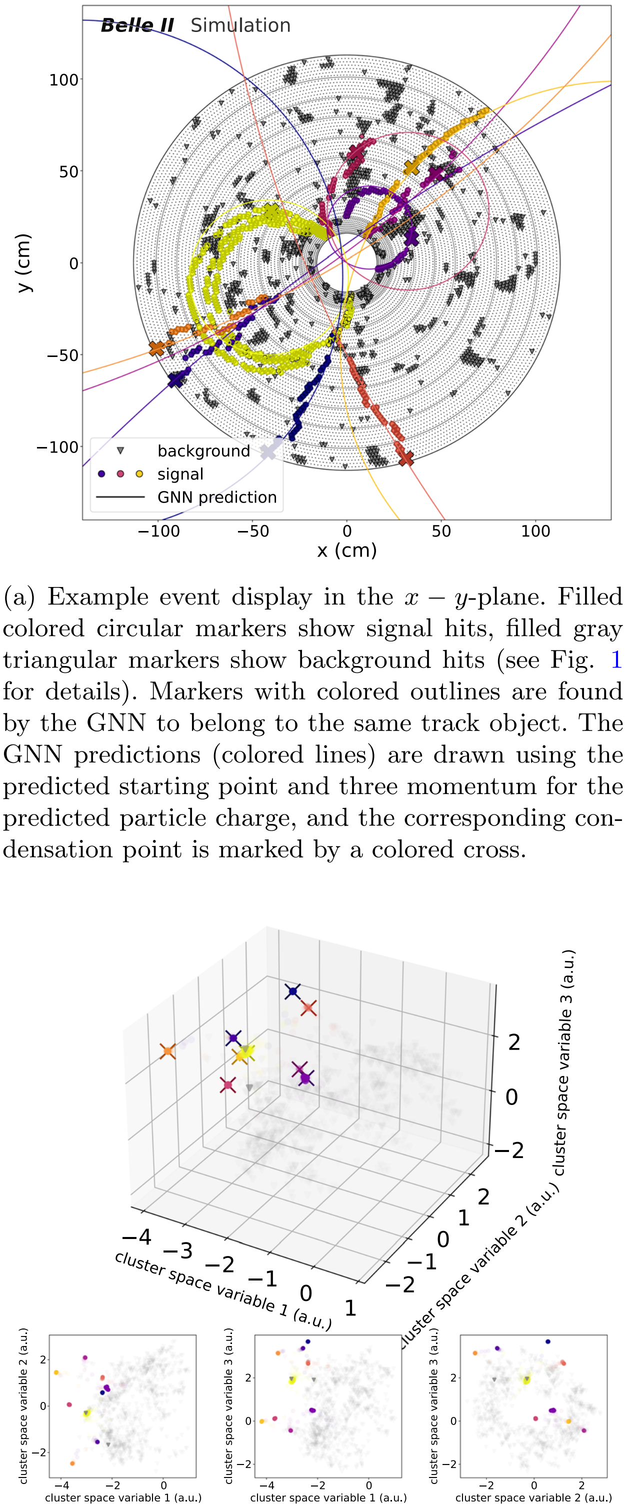
The hyperparameter optimization of the GNN, and the optimization of the node parameters β, t_d and t_h are described in section "Hyperparameter Optimization".

Hyperparameter Optimization

The *CAT Finder* requires to optimize both the model hyperparameters itself as usual, but also the track finding hyperparameters t_β, t_d, and t_h. We optimize the model parameters and the track finding parameters in two subsequent steps and focus on tracks coming from the interaction point. An optimal solution would be to co-optimize the full track finding and fitting for a wide range of physics processes in Belle II. We anticipate that this is an area for further exploration and future development.

The hyperparameter optimization of the model parameters is done using `Weights and Biases` [48] with respect to the model loss. We generated a new train data set with the same samples as for training (see Table 1), but with a reduced data set size of only 6% (about 62,000 events in total) with the low simulated beam backgrounds. The range of tested hyperparameters and the final values are summarized in Table 2. The optimal model has 797,812 trainable parameters. For the final training with the optimal hyperparameters, the learning rate is reduced by a factor of 2 once the learning stagnates for 30 epochs.

We use a two-phase training strategy to speed up model training. First, the model was trained on the simulated low beam background data set to learn track signatures, which is the most time-consuming part that takes about 500 epochs to converge. Then, we retrain the model on a *high data beam backgrounds* data set, with a factor 10 smaller learning rate focusing on the background suppression component of the loss, allowing the model to fine-tune the performance. Despite each epoch on the high data beam background taking over three times longer because of the much higher hit occupancy, our two-phase approach led to an overall significantly faster convergence in just 50 additional epochs.



(a) Example event display in the $x-y$-plane. Filled colored circular markers show signal hits, filled gray triangular markers show background hits (see Fig. 1 for details). Markers with colored outlines are found by the GNN to belong to the same track object. The GNN predictions (colored lines) are drawn using the predicted starting point and three momentum for the predicted particle charge, and the corresponding condensation point is marked by a colored cross.

(b) Cluster space representation (top) in 3D with condensation points marked by a cross, and (bottom) 2D projections. The colors are identical to those in Fig. 4a.

Fig. 4 **a** Event display and **b** Cluster space representation of one example event from category 11 (Table 1) for *high data beam backgrounds*

Table 2 Hyperparameters of the GNN Model with their examined range and the result after the optimization, ranked according to their relevance as given by [48]

Hyperparameter	Examined range	Result
Number of GravNet blocks N	2–7	4
Number of nearest neighbours in GravNet k	2–100	54
Momentum	0.1−0.8	0.77
GravNet spacial information space dimension S	3–6	4
Width of the linear layer LL	32–128	126
Dimension of the Object Condensation cluster coordinate space CS	2–5	3
Width of the linear layer LL2	16–64	16

They are trained on the event categories described in Table 1, using an independent data set with 6% of the full data sample size. We use 80% of the data for training and 20% for validation. The hyperparameters are chosen according to the minimal loss on the validation set

The three tracking hyperparameters t_β, t_d and t_h are optimized using samples from category 2 and $K_S^0 \to \pi^+\pi^-$ events with the K_S^0 momentum pointing in the CDC barrel region (see section "Data Set" for details). The track finding and fitting efficiency ε_{trk}, and the purity $\mathfrak{p}_{\text{trk}}$ are calculated for $t_\beta = [0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95]$, to achieve a ROC curve showing the trade-off between purity and efficiency. This is done for several combinations of condensation point distances $t_d = [0.1, 0.2, 0.3, 0.5, 0.7]$ and hit radii $t_h = [0.05, 0.1, 0.15, 0.25, 0.3]$. The results of this optimization are shown in Fig. 5 for the combined track finding and fitting charge efficiency. The working point $\mathfrak{w} = (t_\beta, t_d, t_h)$ is chosen so that first

$$\varepsilon_{\text{trk}}(\mathfrak{w}_i) + \mathfrak{p}_{\text{trk}}(\mathfrak{w}_i) \geq \varepsilon_{\text{trk}}(\text{baseline}) + \mathfrak{p}_{\text{trk}}(\text{baseline}), \tag{12}$$

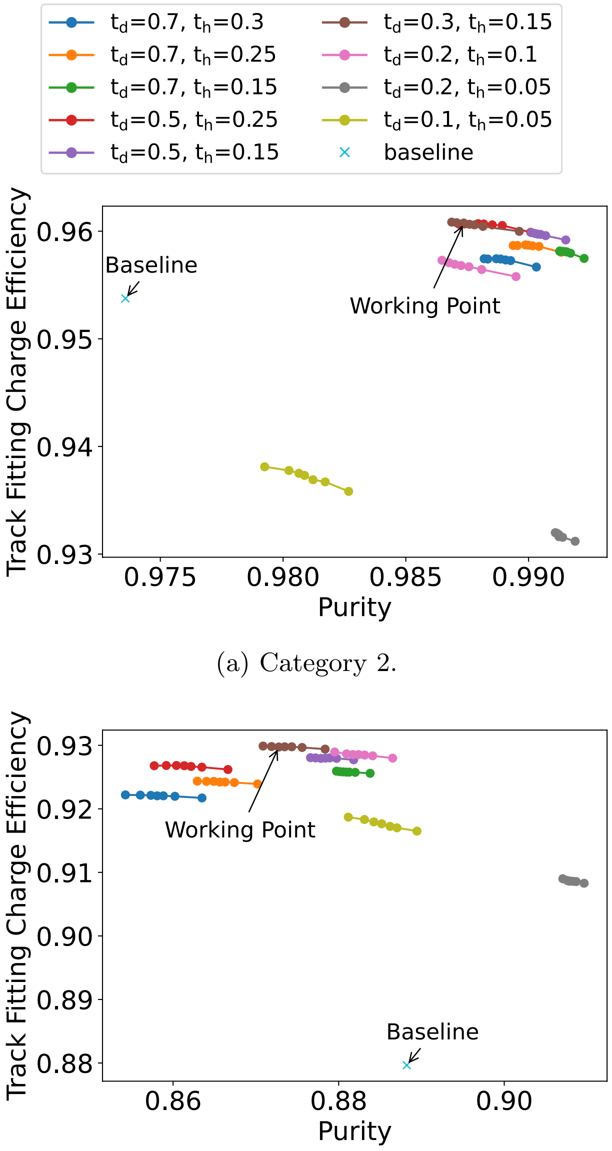
and then

$$\max_i \left(\varepsilon_{\text{trk}}(\mathfrak{w}_\text{i})_{\text{category 2}} + \varepsilon_{\text{trk}}(\mathfrak{w}_\text{i})_{K_S^0 \to \pi^+\pi^-} \right), \tag{13}$$

where $\varepsilon_{\text{trk}}(\mathfrak{w}_\text{i})_x$ is the track finding and fitting efficiency on the category 2 or the $K_S^0 \to \pi^+\pi^-$ sample. This results in the optimal values $t_\beta = 0.3$, $t_d = 0.3$ and $t_h = 0.15$. We note that a different choice of events to optimize these hyperparameters resulted in slightly different optimal values, but these optimal values were always rather close to the ones finally chosen.

Results

In this section we show a comparison of the *CAT Finder* and the *Baseline Finder*, with and without the GENFIT2 track fits, for prompt and displaced tracks. We first discuss the track finding and track fitting efficiency in section "Track Finding and Track Fitting Efficiency" and then compare the momentum resolution of the track finding and fitting algorithms in section "Track Momentum Resolution". We discuss the performance of the *CAT Finder* to infer the starting position of a track in section "Position Reconstruction". We show an analysis of the *CAT Finder* robustness against different beam background conditions in training and evaluation in section "Robustness to Variable Detector Conditions". Finally, we list various lessons we learned while training the model in section "Lessons Learned".



(a) Category 2.

(b) $K_S^0 \to \pi^+\pi^-$ (barrel).

Fig. 5 Combined track finding and fitting charge efficiency as function of purity for the *CAT Finder*, and the respective value for the *Baseline Finder* for **a** category 2 and **b** $K_S^0 \to \pi^+\pi^-$ for *high data beam backgrounds*. See text for details

Track Finding and Track Fitting Efficiency

The track finding efficiency [see Eq. (4)] depends on the fraction of matched signal hits, the fraction of beam background hits that are wrongly assigned to the track, and wrongly assigned signal hits from other particles. Particles that leave a small number of true signal hits are harder to reconstruct by the track finding algorithms. This affects particles with very small or very large polar angles in the forward or backward regions, or particle that are produced at a large distance from the IP. Particles that are close to other particles in real space often lead to a correlated efficiency loss, meaning that if one particle is lost, the other is lost, too. In our evaluation samples, this makes the light mass $h \rightarrow \mu^+\mu^-$ samples the most challenging event samples for track finding.

The baseline finder begins tracking in the $x-y$ plane, requiring a sufficient number of hits in the axial layers. This approach reduces efficiency for short tracks in the endcap regions. In the overall Belle II tracking chain, efficiency is restored through the use of tracks identified by the silicon vertex detector. However, for displaced vertex signatures, the CDC remains the sole tracking detector, with no additional support to recover efficiency.

Prompt Tracks

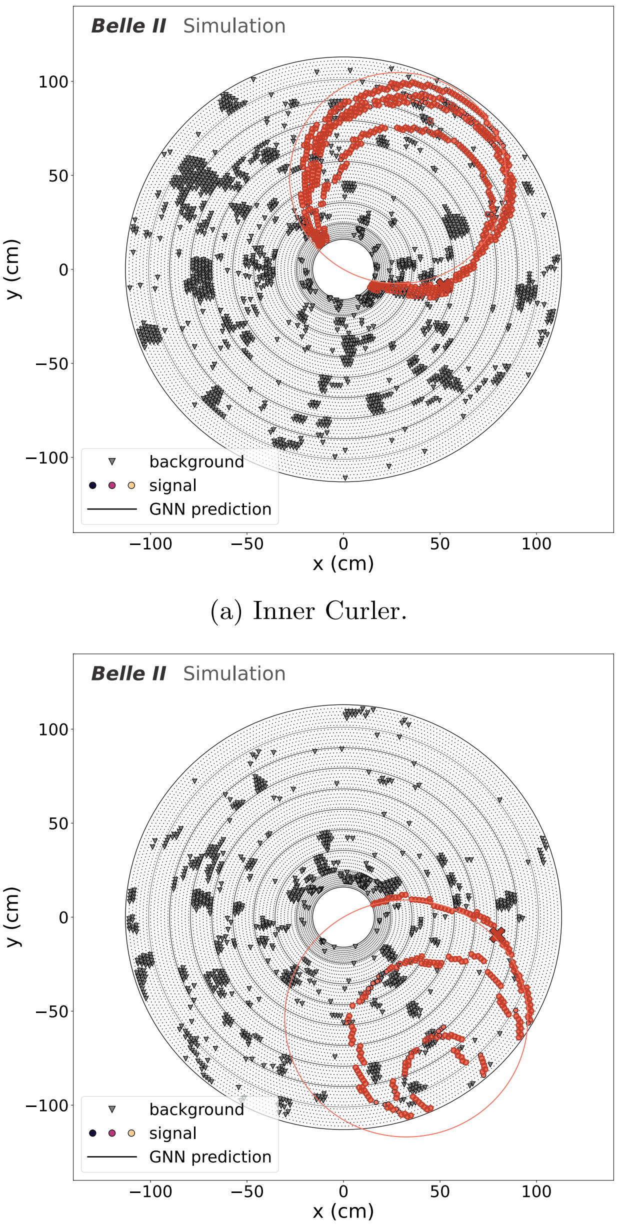
We evaluate the track finding efficiency for prompt tracks using the track categories 1–3 (see Table 1). The events have between 1 and 12 muons per event in the respective detector regions, with transverse momenta $0.05 < p_t < 6$ GeV.

Two categories of prompt tracks are particularly difficult in the Belle II tracking environment:

- Particles that re-enter the CDC several times without significant energy loss if their momenta is $p_t \lesssim 0.25$ GeV (so called *inner curler*), see Fig. 6a;
- Minimum-ionizing[1] charged particles with momenta around $p_t \approx 0.3$ GeV with polar angles pointing in the central part of the barrel that leave the CDC, travel through passive material or outer detectors, loose a significant amount of energy by ionization, and re-enter the CDC (so called *outer curler*), see Fig. 6b.

We split the discussion in this subsection into tracks that are non-curling, and tracks that curl.

To remove *inner curler* and *outer curler*, we exclude prompt particles with a distance between two consecutive hits larger than 20 cm, and all particles with more than 32 signal hits in the CDC's first superlayer A1. We stress that we only remove such tracks from the evaluation, but not the events that contain these curlers. Particularly, the samples in category 1–3 contain many events with a large number of low transverse momentum tracks that occupy the inner part of the CDC.



(a) Inner Curler.

(b) Outer Curler.

Fig. 6 Example event displays in the *x*–*y*-plane for **a** *inner curler* and **b** *outer curler* for *high data beam background*. Filled colored circular markers show signal hits, filled gray triangular markers show background hits (see Fig. 1 for details). Markers with colored outlines are found by the GNN to belong to the same track object. The GNN predictions (colored lines) are drawn using the predicted starting point and three momentum for the predicted particle charge, and the corresponding condensation point is marked by a colored cross

[1] Electrons, protons, and heavier mesons with such low transverse momentum usually loose too much energy in the calorimeter to produce a re-entering track.

Non-curling Tracks The track finding efficiencies, and the combined track finding and track fitting efficiency for the *Baseline Finder* in comparison with the *CAT Finder* are shown in Fig. 7. A similar comparison but for track charge efficiencies can be found in Appendix C.

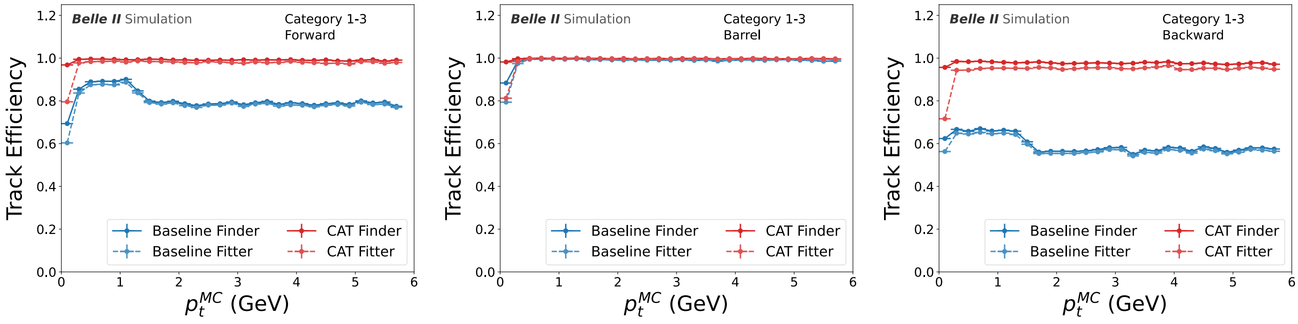
The performance metrics are summarized in Table 3. The track finding efficiency of the *CAT Finder* in the barrel is higher than the *Baseline Finder*, but also features larger fake and clone rates. In the endcaps, the *CAT Finder* has a significantly higher efficiency and charge efficiency and again higher fake and clone rates. Tracks that point towards the endcaps leave fewer hits, and those hits are in the inner CDC region that features a higher occupancy from background. The *CAT Finder* is able to efficiently reject beam background hits leading to much higher hit purity for tracks pointing towards the endcaps, and to a better track efficiency. The combined track finding and fitting efficiency shows the same trend of significantly better performance in the endcaps and comparable performance in the barrel, but the fake rate is significantly smaller than for the *Baseline Finder*, indicating that some fraction of the additionally found tracks by the *CAT Finder* cannot be fitted.

Table 3 The performance metrics for the prompt evaluation samples (category 1–3, *high data beam backgrounds*, see Table 1 and section “Data Set” for details) for non-curling tracks for *CAT Finder* and *Baseline Finder* in different detector regions

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$80.1^{+0.1}_{-0.1}$	$0.55^{+0.02}_{-0.02}$	0.01	$78.4^{+0.1}_{-0.1}$	$2.06^{+0.04}_{-0.04}$
CAT Finder	$98.94^{+0.03}_{-0.03}$	$1.62^{+0.03}_{-0.03}$	$0.21^{+0.01}_{-0.01}$	$98.89^{+0.03}_{-0.03}$	$0.06^{+0.01}_{-0.01}$
Baseline Fitter	$78.1^{+0.1}_{-0.1}$	$0.49^{+0.02}_{-0.02}$	0.01	$77.1^{+0.1}_{-0.1}$	$1.37^{+0.04}_{-0.04}$
CAT Fitter	$95.93^{+0.06}_{-0.05}$	$0.31^{+0.02}_{-0.02}$	$0.06^{+0.01}_{-0.01}$	$94.29^{+0.06}_{-0.06}$	$1.71^{+0.04}_{-0.04}$
Barrel					
Baseline Finder	$97.97^{+0.04}_{-0.04}$	$2.31^{+0.04}_{-0.04}$	$0.05^{+0.01}_{-0.01}$	$95.92^{+0.06}_{-0.06}$	$2.09^{+0.04}_{-0.04}$
CAT Finder	$99.61^{+0.02}_{-0.02}$	$3.34^{+0.05}_{-0.05}$	$0.59^{+0.02}_{-0.02}$	$99.16^{+0.03}_{-0.03}$	$0.46^{+0.02}_{-0.02}$
Baseline Fitter	$96.88^{+0.05}_{-0.05}$	$1.83^{+0.04}_{-0.04}$	0.03	$95.5^{+0.06}_{-0.06}$	$1.42^{+0.03}_{-0.03}$
CAT Fitter	$97.6^{+0.04}_{-0.04}$	$1.26^{+0.03}_{-0.03}$	$0.16^{+0.01}_{-0.01}$	$97.39^{+0.05}_{-0.04}$	$0.22^{+0.01}_{-0.01}$
Backward endcap					
Baseline Finder	$60.5^{+0.1}_{-0.1}$	$1.08^{+0.04}_{-0.04}$	$0.03^{+0.01}_{-0.01}$	$58.0^{+0.1}_{-0.1}$	$4.08^{+0.07}_{-0.07}$
CAT Finder	$97.64^{+0.04}_{-0.04}$	$1.2^{+0.03}_{-0.03}$	$0.14^{+0.01}_{-0.01}$	$97.42^{+0.04}_{-0.04}$	$0.22^{+0.01}_{-0.01}$
Baseline Fitter	$58.8^{+0.1}_{-0.1}$	$0.92^{+0.03}_{-0.04}$	$0.02_{-0.01}$	$56.8^{+0.1}_{-0.1}$	$3.28^{+0.06}_{-0.06}$
CAT Fitter	$92.43^{+0.07}_{-0.07}$	$0.69^{+0.02}_{-0.02}$	$0.03_{-0.01}$	$87.67^{+0.09}_{-0.09}$	$5.16^{+0.06}_{-0.06}$

Uncertainties below <0.01% are not shown in the table

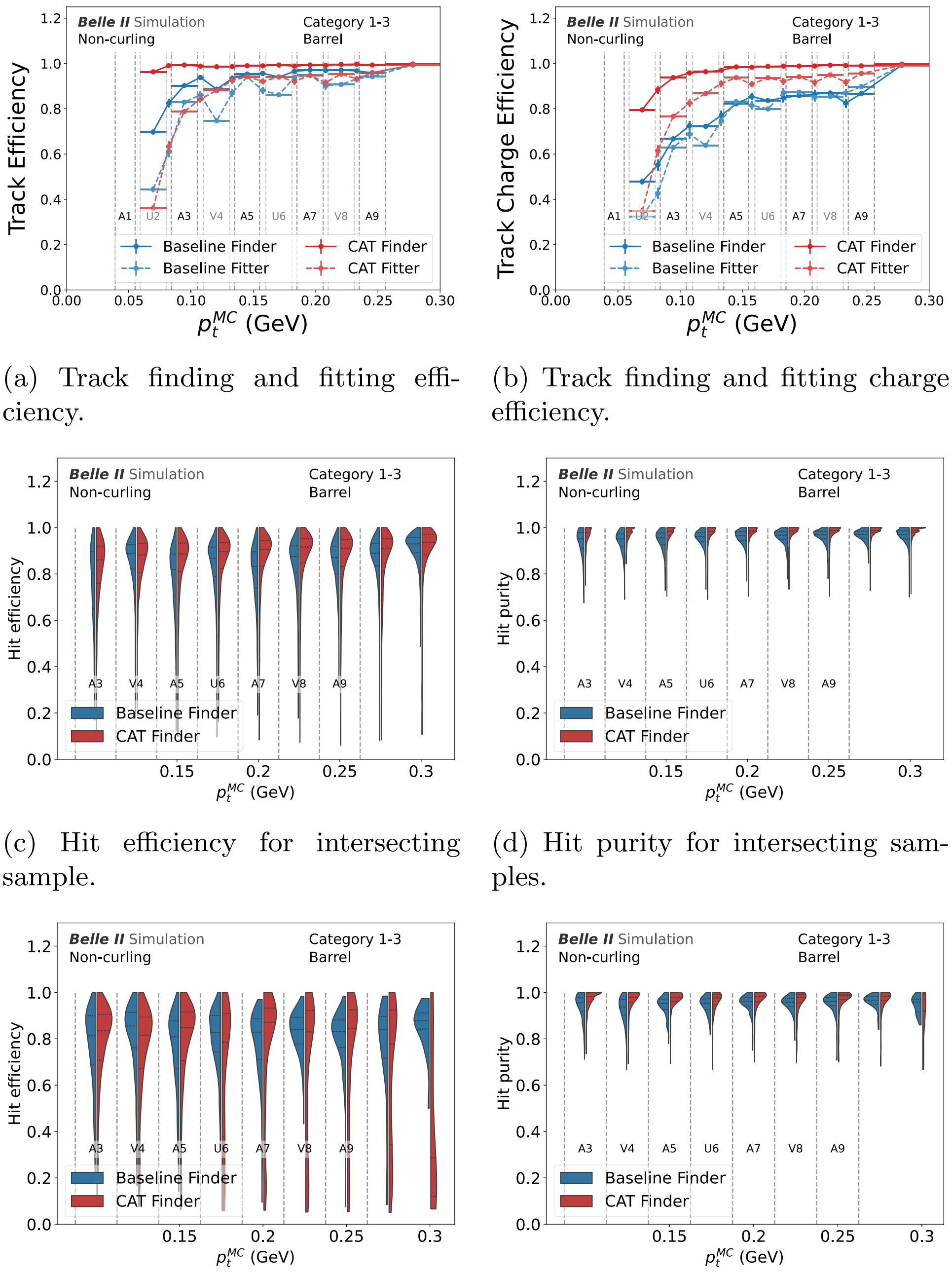
Despite the removal of curling tracks in Fig. 7 and Table 6, one observes a significant drop in efficiency for both *CAT Finder* and *Baseline Finder* for $p_t \lesssim 0.3$ GeV. A detailed inspection of this region is shown in Fig. 8. For the tracks with removed curlers, the *CAT Finder* shows a very high track finding efficiency down to transverse momenta of about 50 MeV that outperforms the *Baseline Finder* (see Fig. 8a). The hit efficiency and hit purity for the tracks that are found both by *CAT Finder* and *Baseline Finder* (intersecting sample) are much higher for the *CAT Finder* (see Fig. 8c, d). For tracks that are only found by the *CAT Finder* but not by the *Baseline Finder* or vice versa (additional sample), the *CAT Finder* has a



(a) Forward endcap. (b) Barrel. (c) Backward endcap.

Fig. 7 Track finding (markers connected by solid lines to guide the eye) and combined track finding and fitting efficiency (markers connected by dashed lines to guide the eye) for the prompt evaluation samples (category 1–3, *high data beam backgrounds*, see Table 1) with curler tracks removed, as function of simulated transverse momentum p_t^{MC} for the *Baseline Finder* (blue) and the *CAT Finder* (red) in the **a** forward endcap, **b** barrel, and **c** backward endcap. The vertical error bars that show the statistical uncertainty are smaller than the marker size. The horizontal error bars indicate the bin width. The uncertainties of the two track finding algorithms are correlated, since they use the same simulated events

Fig. 8 The top row shows the low momentum track finding (empty markers, connected by lines to guide the eye) and combined track finding and fitting charge efficiency (filled markers) (**a**) and the track finding and combined track finding and fitting charge efficiency (**b**) for the prompt evaluation samples (category 1–3, *high data beam backgrounds*, see Table 1) for non-curling tracks. The middle row shows hit efficiency and hit purity for tracks found by both *CAT Finder* and *Baseline Finder* (intersecting sample) (**c** and **d**) and the bottom row for the additional found tracks (**e** and **f**). The dashed horizontal dark (light) gray lines show the axial (stereo) superlayer boundaries how far the prompt track reaches with the given transverse momentum



(a) Track finding and fitting efficiency.

(b) Track finding and fitting charge efficiency.

(c) Hit efficiency for intersecting sample.

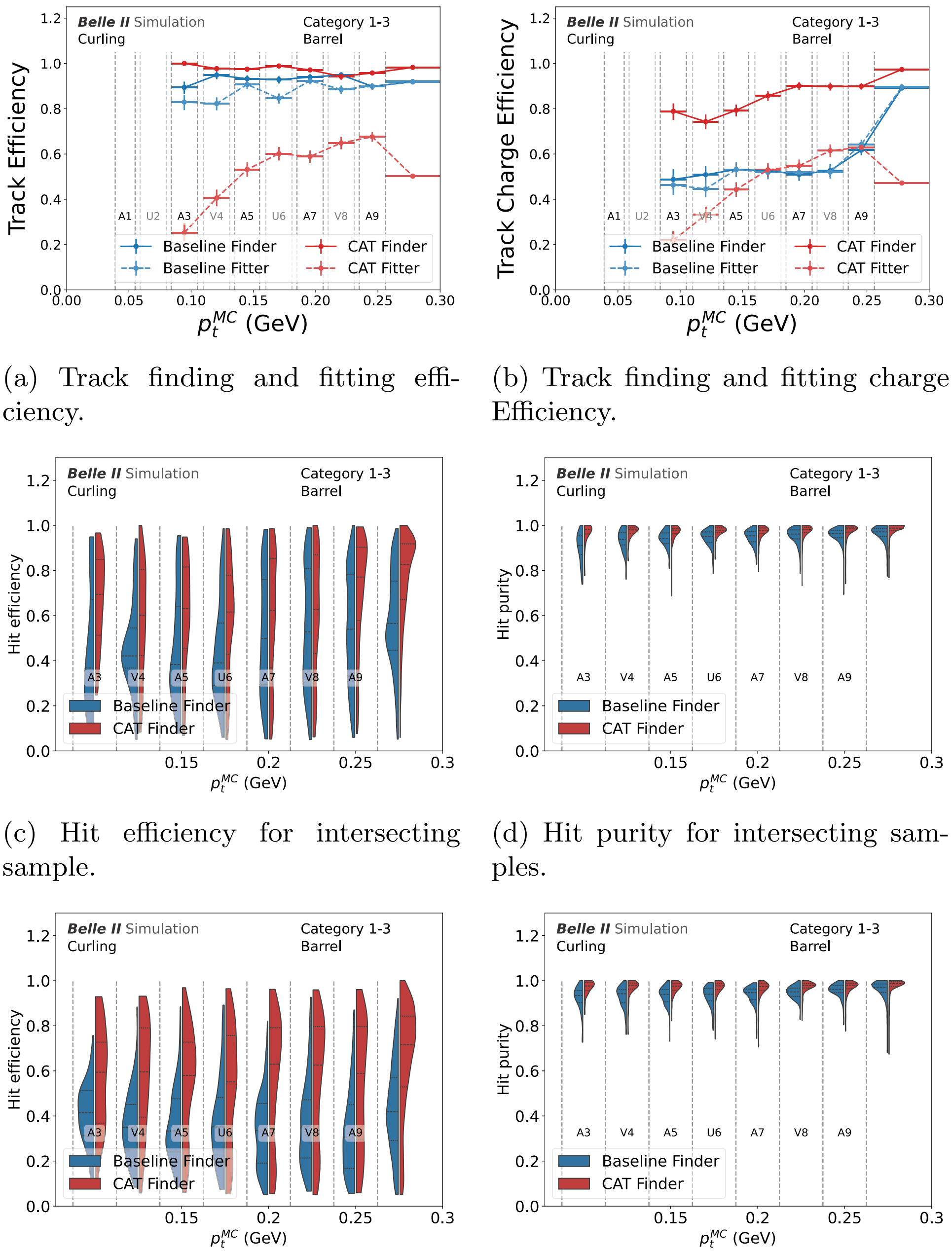
(d) Hit purity for intersecting samples.

(e) Hit efficiency for additional samples.

(f) Hit purity for additional samples.

higher hit efficiency and purity. We note that the number of additional tracks found only by the *Baseline Finder* is very small. *CAT Finder* has a higher track finding efficiency than the *Baseline Finder*, but the combined track finding and fitting efficiency is comparable. This indicates that the *CAT Finder* finds more complicated track topologies that cannot be fitted. Improving the fitting efficiency of these tracks will require additional tuning of the track-fitting algorithms, which is beyond the scope of this work. The track fitting charge efficiency for the *CAT Finder* is again significantly better than for the *Baseline Finder* since the charge and momentum direction prediction of the *CAT Finder* is better than that of the *Baseline Finder*, which in turn leads to a correct hit ordering and more successful fits for the *CAT Finder*.

Curling Tracks The same comparisons as in Fig. 8, but for curling tracks, are shown in Fig. 9. The track finding efficiency and especially the track finding charge efficiency of the *CAT Finder* for curling tracks is very high (see Fig. 9a,



(a) Track finding and fitting efficiency.

(b) Track finding and fitting charge Efficiency.

(c) Hit efficiency for intersecting sample.

(d) Hit purity for intersecting samples.

(e) Hit efficiency for additional samples.

(f) Hit purity for additional samples.

Fig. 9 Low momentum track finding (empty markers, connected by lines to guide the eye) and combined track finding and fitting charge efficiency (filled markers) (**a**) and the track finding and combined track finding and fitting charge efficiency (**b**) for curling tracks with *high data beam backgrounds*. The middle row shows hit efficiency and hit purity for tracks found by both *CAT Finder* and *Baseline Finder* (intersecting sample) (**c** and **d**) and the bottom row for the additional found tracks (**e** and **f**). The dashed horizontal dark (light) gray lines show the axial (stereo) superlayer boundaries how far the prompt track reaches with the given transverse momentum

b), showing that the *CAT Finder* is able to assign curler hits even from multiple curls to the same track object (see Fig. 9c, d).

The track fitting algorithms on the other hand can not handle these signatures and fail for many of these *CAT Finder* tracks (see Fig. 9a). The *Baseline Finder* on the other hand often only assigns the first curl to one object which produces tracks with very low hit efficiency, but that can be fitted successfully. When comparing the track fitting charge efficiency (see Fig. 9b), both *CAT Finder* and *Baseline Finder* show similar efficiency again for all momenta but for $p_t \approx 0.3$ GeV which contains most of the *outer curlers*, and the lowest momentum bin which contains very high number of inner curlers. Overall, the very high *track hit efficiency* and *track hit purity* for *CAT Finder* curling tracks that is even present for the additional samples, require significant adjustments of the track fitting algorithms to propagate the better track finding efficiency and the better track finding charge efficiency to the end of the tracking pipeline. Additional steps are needed to choose the outermost first curl for the best momentum estimation and give this information to the fitter while keeping the additional hits on the track to minimize clones, which is beyond the scope of this work.

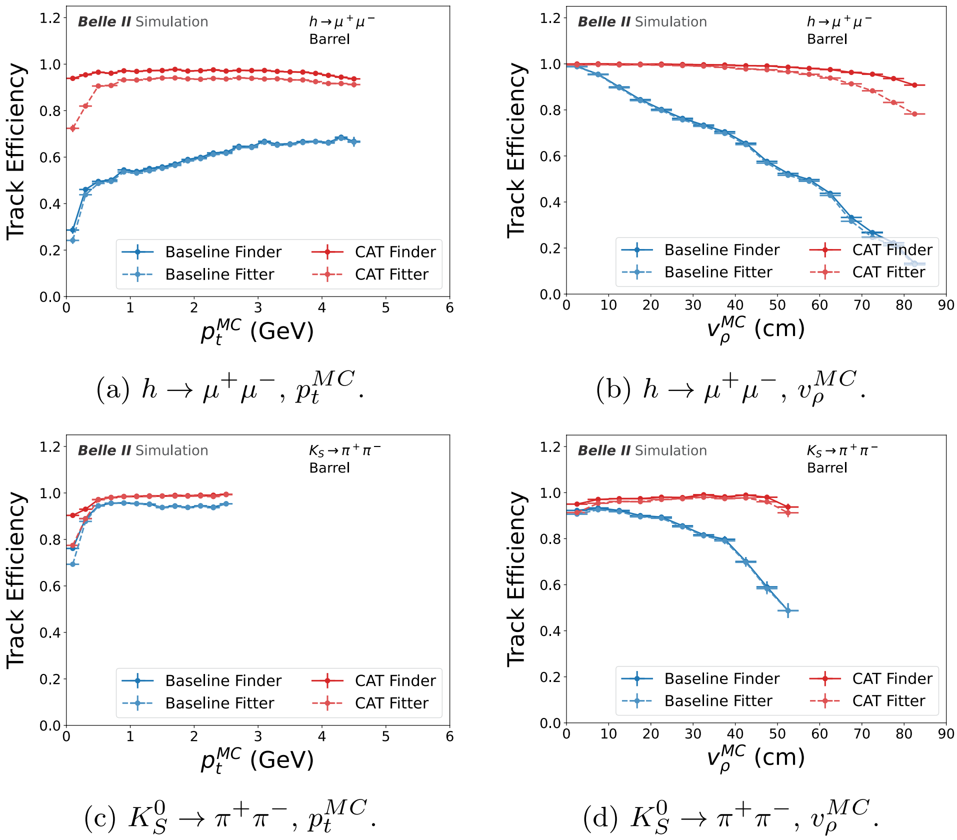
Prompt Tracks in $\mu^-\mu^+(\gamma)$

In addition to the training and evaluation samples described above, we compare the track finding algorithms on simulated $\mu^-\mu^+(\gamma)$ events. As one of the main calibration samples at Belle II, the target track fitting charge efficiency in the barrel is 100%. Compared to the event samples from category 1 to 3 described above, these events almost always feature two isolated, prompt, high momentum tracks. The *CAT Finder* shows a significantly higher track finding efficiency in both endcaps but at the same time a slighter higher fake rate than the *Baseline Finder*. After track fitting, the combined track finding and fitting efficiency of the *CAT Finder* is similar to the *Baseline Finder* but with a significantly lower fake rate. In the barrel, both algorithms achieve a combined track finding and fitting charge efficiency of 99.8% for the *CAT Finder* and 99.4% for the *Baseline Finder*, while the *CAT Finder* has the lower fake rate. In the two endcaps, the combined track finding and fitting charge efficiency for forward and backward endcaps is 95.1% for the *CAT Finder* compared to 73.2% for the *Baseline Finder*. Additional plots and numerical results for $\mu^-\mu^+(\gamma)$ events are shown in Appendix D.

Displaced Tracks

We evaluate the track finding efficiency for displaced tracks using events with dark Higgs decays $h \to \mu^+\mu^-$ with $m_h = [0.5, 2.0, 4.0]$ GeV where the dark Higgs decays uniformly along its flight direction into two charged particles. In addition, we consider events with single $K^0_S \to \pi^+\pi^-$ decays, where the distribution of the decay distance of the displaced vertex follows an exponential pattern. We do not split the sample into curling and non-curling tracks. The track finding efficiencies, and the combined track finding and track fitting efficiency for the *Baseline Finder* in comparison with the *CAT Finder* are shown in Fig. 10. The same information but for track charge efficiencies can be found in Appendix E. The performance metrics integrated over the full p_t range as shown in Fig. 10, are summarized in Tables 4 and 5.

The dark Higgs samples exhibit two types of decays: for large dark Higgs masses, decays into two muons occur with relatively large opening angles, resulting in distinct trajectories within the same superlayer. In contrast, for small dark Higgs masses, decays feature smaller opening angles, producing partially overlapping tracks. Since the lifetime



(a) $h \to \mu^+\mu^-$, p_t^{MC}.

(b) $h \to \mu^+\mu^-$, v_ρ^{MC}.

(c) $K^0_S \to \pi^+\pi^-$, p_t^{MC}.

(d) $K^0_S \to \pi^+\pi^-$, v_ρ^{MC}.

Fig. 10 Track finding (empty markers) and combined track finding and fitting efficiency (filled markers) for (top) displaced tracks in $h \to \mu^+\mu^-$ events and in (bottom) $K^0_S \to \pi^+\pi^-$ events with *high data beam backgrounds*, as function of (left) the true simulated transverse momentum p_t^{MC}, and (right) the true simulated displacement v_ρ^{MC} in the $x - y$ plane

Table 4 The performance metrics per track for $h \rightarrow \mu^+\mu^-$ (m_h = [0.5,2.0,4.0] GeV) samples with *high data beam backgrounds* decaying uniformly along its flight direction into two charged particles (see section "Data Set" for details) for different track finding algorithms in different detector regions

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$36.2^{+0.4}_{-0.4}$	$15.1^{+0.8}_{-0.8}$	$0.3^{+0.1}_{-0.1}$	$33.8^{+0.3}_{-0.4}$	$6.5^{+0.3}_{-0.3}$
CAT Finder	$88.1^{+0.2}_{-0.2}$	$15.8^{+0.4}_{-0.4}$	$0.64^{+0.09}_{-0.10}$	$83.9^{+0.3}_{-0.3}$	$4.8^{+0.2}_{-0.2}$
Baseline Fitter	$35.5^{+0.4}_{-0.4}$	$17.5^{+0.7}_{-0.7}$	$0.22^{+0.1}_{-0.13}$	$34.3^{+0.4}_{-0.4}$	$3.4^{+0.2}_{-0.2}$
CAT Fitter	$79.7^{+0.3}_{-0.3}$	$7.4^{+0.3}_{-0.3}$	$0.1^{+0.03}_{-0.04}$	$75.4^{+0.3}_{-0.3}$	$5.4^{+0.2}_{-0.2}$
Barrel					
Baseline Finder	$59.5^{+0.1}_{-0.1}$	$4.94^{+0.07}_{-0.07}$	$0.53^{+0.02}_{-0.03}$	$56.4^{+0.1}_{-0.1}$	$5.13^{+0.08}_{-0.08}$
CAT Finder	$96.89^{+0.05}_{-0.05}$	$5.12^{+0.06}_{-0.06}$	$1.56^{+0.03}_{-0.03}$	$94.94^{+0.06}_{-0.06}$	$2.01^{+0.04}_{-0.04}$
Baseline Fitter	$58.8^{+0.1}_{-0.1}$	$3.57^{+0.06}_{-0.06}$	$0.33^{+0.02}_{-0.02}$	$57.4^{+0.1}_{-0.1}$	$2.36^{+0.05}_{-0.05}$
CAT Fitter	$92.75^{+0.07}_{-0.07}$	$2.12^{+0.04}_{-0.04}$	$0.54^{+0.02}_{-0.02}$	$89.21^{+0.08}_{-0.08}$	$3.81^{+0.05}_{-0.05}$
Backward endcap					
Baseline Finder	$17.2^{+0.4}_{-0.4}$	$4.4^{+0.2}_{-0.2}$	$0.32^{+0.06}_{-0.07}$	$15.1^{+0.3}_{-0.3}$	$12.1^{+0.7}_{-0.8}$
CAT Finder	$71.0^{+0.4}_{-0.4}$	$14.8^{+0.3}_{-0.3}$	$0.74^{+0.07}_{-0.07}$	$64.6^{+0.5}_{-0.5}$	$9.1^{+0.3}_{-0.3}$
Baseline Fitter	$16.5^{+0.4}_{-0.4}$	$4.8^{+0.3}_{-0.3}$	$0.23^{+0.05}_{-0.07}$	$15.6^{+0.3}_{-0.3}$	$5.8^{+0.5}_{-0.6}$
CAT Fitter	$58.0^{+0.5}_{-0.5}$	$3.1^{+0.1}_{-0.1}$	$0.08^{+0.02}_{-0.03}$	$53.3^{+0.5}_{-0.5}$	$8.2^{+0.3}_{-0.4}$

is uniformly distributed in the respective dark Higgs direction, the sample contains many very displaced tracks. The *CAT Finder* demonstrates significantly higher track-finding efficiencies both before and after track fitting in all detector regions, for all transverse momenta, and for all displacements. It also exhibits the lowest fake and clone rates, achieving a combined track-finding and fitting charge efficiency of 85.4% per track, with a fake rate of 2.5%, averaged over the full detector acceptance. In comparison, the *Baseline Finder* achieves 52.2% efficiency and a fake rate of 4.1%.

We finally evaluate the performance on a signal sample close to the expected Belle II sensitivity for such a BSM scenario [3] with a dark Higgs mass m_h =1.5 GeV and a mixing angle $\sin(\theta) = 10^{-4}$, which corresponds to a lifetime of $c\tau$ =21.5 cm in the dark Higgs restframe. The *CAT Finder* efficiency to reconstruct both tracks in the event is 87.2% compared to the baseline algorithm with only 44.9%, with a smaller fake rate of 2.5% for the *CAT Finder* and 3.3% for the *Baseline Finder*; restricted to tracks that are in the barrel region, the efficiency is 90.0% for the *CAT Finder* and 52.2% for the *Baseline Finder*. The fake rates are 2.1% for the *CAT Finder* and 3.0% for the *Baseline Finder*.

Table 5 The performance metrics per displaced pion track in $K^0_S \rightarrow \pi^+\pi^-$ samples with *high data beam backgrounds* with a uniformly generated transverse momentum of $p_t(K^0_S) = [0.05 - 3]$ GeV

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$63.2^{+0.2}_{-0.2}$	$3.5^{+0.1}_{-0.1}$	$0.13^{+0.02}_{-0.03}$	$62.5^{+0.2}_{-0.2}$	$1.2^{+0.06}_{-0.06}$
CAT Finder	$93.2^{+0.1}_{-0.1}$	$6.8^{+0.1}_{-0.1}$	$0.26^{+0.02}_{-0.02}$	$92.7^{+0.1}_{-0.1}$	$0.45^{+0.03}_{-0.03}$
Baseline Fitter	$61.9^{+0.2}_{-0.2}$	$4.1^{+0.1}_{-0.1}$	$0.1^{+0.02}_{-0.02}$	$61.2^{+0.2}_{-0.2}$	$1.23^{+0.06}_{-0.06}$
CAT Fitter	$88.6^{+0.1}_{-0.1}$	$3.01^{+0.08}_{-0.08}$	$0.08^{+0.01}_{-0.01}$	$86.6^{+0.2}_{-0.2}$	$2.23^{+0.07}_{-0.07}$
Barrel					
Baseline Finder	$91.25^{+0.09}_{-0.09}$	$6.89^{+0.08}_{-0.08}$	$0.83^{+0.03}_{-0.03}$	$88.5^{+0.1}_{-0.1}$	$3.0^{+0.06}_{-0.06}$
CAT Finder	$96.15^{+0.06}_{-0.06}$	$11.52^{+0.09}_{-0.09}$	$1.99^{+0.04}_{-0.04}$	$95.56^{+0.07}_{-0.07}$	$0.61^{+0.03}_{-0.03}$
Baseline Fitter	$90.05^{+0.1}_{-0.1}$	$5.39^{+0.07}_{-0.07}$	$0.59^{+0.02}_{-0.03}$	$88.1^{+0.1}_{-0.1}$	$2.16^{+0.05}_{-0.05}$
CAT Fitter	$93.43^{+0.08}_{-0.08}$	$5.13^{+0.07}_{-0.07}$	$0.54^{+0.02}_{-0.02}$	$92.99^{+0.08}_{-0.08}$	$0.46^{+0.02}_{-0.02}$
Backward endcap					
Baseline Finder	$44.0^{+0.2}_{-0.2}$	$2.51^{+0.08}_{-0.08}$	$0.1^{+0.02}_{-0.02}$	$42.7^{+0.2}_{-0.2}$	$3.0^{+0.1}_{-0.1}$
CAT Finder	$90.1^{+0.1}_{-0.1}$	$9.6^{+0.1}_{-0.1}$	$0.42^{+0.03}_{-0.03}$	$89.4^{+0.1}_{-0.1}$	$0.74^{+0.04}_{-0.04}$
Baseline Fitter	$42.6^{+0.2}_{-0.2}$	$2.24^{+0.08}_{-0.08}$	$0.07^{+0.01}_{-0.02}$	$41.2^{+0.2}_{-0.2}$	$3.2^{+0.1}_{-0.1}$
CAT Fitter	$83.2^{+0.2}_{-0.2}$	$2.35^{+0.07}_{-0.07}$	$0.12^{+0.01}_{-0.02}$	$79.3^{+0.2}_{-0.2}$	$4.7^{+0.1}_{-0.1}$

The average transverse decay distance is $v_\rho = 8.24$ cm (see section "Data Set" for details) for different track finding algorithms in different detector regions

In contrast to the $h \rightarrow \mu^+\mu^-$ decays, the $K^0_S \rightarrow \pi^+\pi^-$ decays occur on average at smaller displacement from the IP and they have a smaller total momentum. Even though the $h \rightarrow \mu^+\mu^-$ and the K^0_S samples probe rather different displacement kinematics, the general trend of the *CAT Finder* for $K^0_S \rightarrow \pi^+\pi^-$ and dark Higgs is comparable: the *CAT Finder* has a significantly higher track finding and fitting efficiency, and a comparable or even lower fake and clone rate than the *Baseline Finder*.

Track Momentum Resolution

The *CAT Finder* provides estimates of the track three-momentum for each condensation point. We use these estimators as starting values for subsequent track fitting algorithms, but they can be used as end-to-end result of a complete single-step GNN-based track reconstruction algorithm. The resolutions of the fitting step are based on the results of the full GENFIT2 algorithm (see section "Track Fitting") for both the *CAT Finder* and the *Baseline Finder*. A comparison of track helix parameter resolutions can be found in Appendix G.

Prompt Tracks

We evaluate the track momentum resolution for matched prompt tracks using the track categories 1–3 (see Table 1). For prompt tracks we evaluate the resolution on the non-curling tracks (see section “Track finding and Track Fitting Efficiency”) for tracks found by both the *CAT Finder* and the *Baseline Finder*. The transverse momentum resolution $\eta(p_t)$ and the longitudinal momentum resolution $\eta(p_z)$ for the *CAT Finder* and the *Baseline Finder* are shown in Fig. 11.

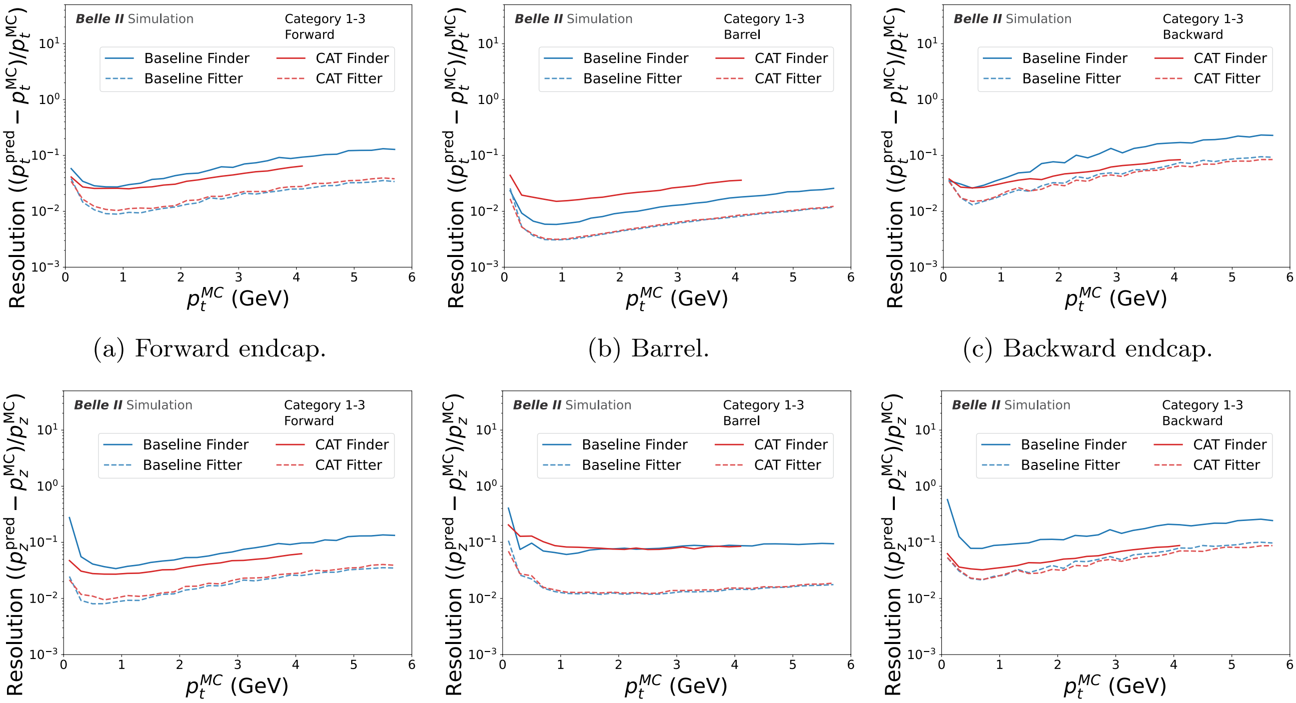
Since our training samples do not include tracks with transverse momenta above 6 GeV, the model does not predict transverse momentum values above around 6.5 GeV which leads to a biased distribution of the momentum distribution (see Appendix F for details). For this reason we do not report the CAT Finder resolution for $p_t > 4$ GeV.

The $\eta(p_t)$ resolution before track fitting for the *CAT Finder* is comparable for the different detector regions, and amounts to a few percent. The *Baseline Finder* performs better than the *CAT Finder* in the barrel and reaches a relative resolution of better than 1%. It performs significantly worse than the *CAT Finder* in both endcaps, due to the much lower hit efficiency and hit purity. The $\eta(p_z)$ resolution before track fitting on the other hand is comparable for *CAT Finder* and the *Baseline Finder* in the barrel and plateaus around 1%. As for the transverse momentum resolution, the *CAT Finder* performs significantly better in the endcaps. After track fitting with GENFIT2, the transverse and the longitudinal momentum resolutions are very similar for the two algorithms in all detector regions. We attribute this similarity to the comparable hit efficiency of the two algorithms for not too small transverse momenta.

Displaced Tracks

We evaluate the track momentum resolution for matched displaced tracks using events with dark Higgs decays and with single $K_S^0 \to \pi^+\pi^-$ decays as described in section “Data Set”. We do not remove curling tracks, and we evaluate the resolution separately for the intersecting and the additional samples in the barrel only.

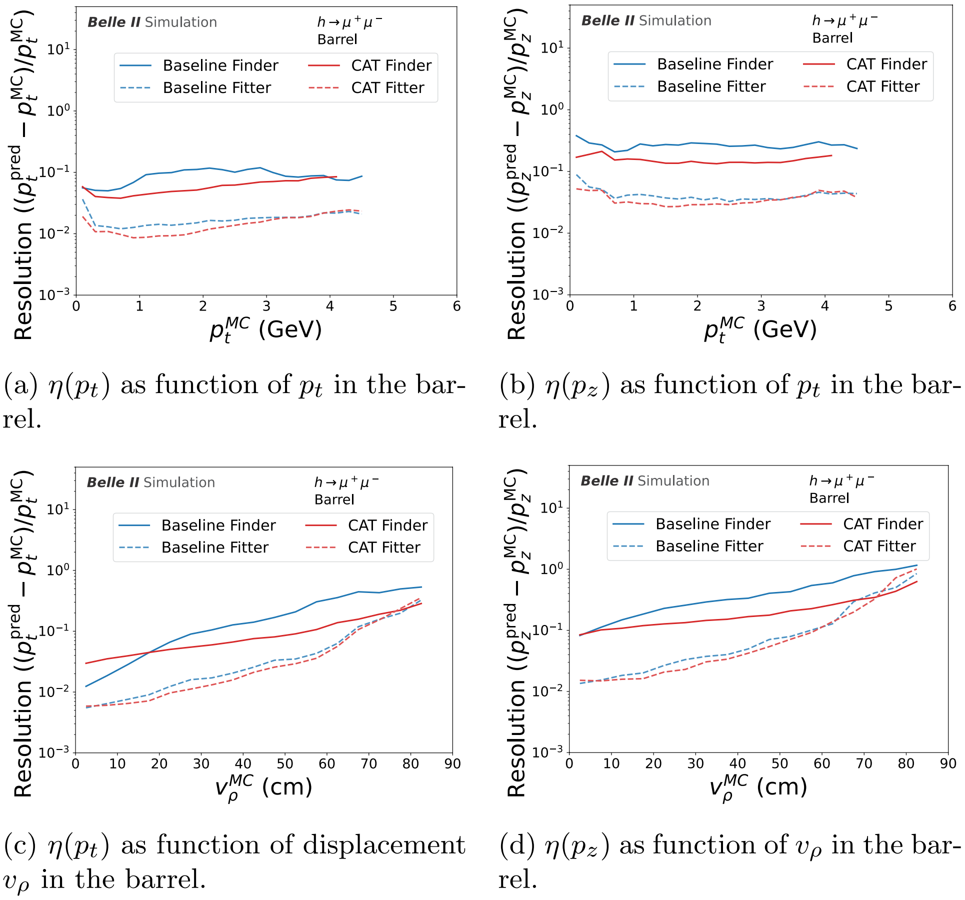
The transverse momentum resolution $\eta(p_t)$ and the longitudinal momentum resolution $\eta(p_z)$ for the *CAT Finder* and the *Baseline Finder* are shown in Fig. 12. The corresponding information for the additional *CAT Finder* sample for $h \to \mu^+\mu^-$ decays and for the intersecting and the additional *CAT Finder* sample for $K_S^0 \to \pi^+\pi^-$ decays are shown in Appendices H and I. For the displaced tracks, the additional



(a) Forward endcap. (b) Barrel. (c) Backward endcap.

(d) Forward endcap. (e) Barrel. (f) Backward endcap.

Fig. 11 Relative (top) transverse and (bottom) longitudinal momentum resolution as function of simulated transverse momentum p_t^{MC} for the intersecting prompt evaluation sample (category 1–3, *high data beam backgrounds*, see Table 1) in the (left) forward endcap, (center) barrel, and (right) backward endcap for tracks found by both (red) *CAT Finder* and (blue) *Baseline Finder*. For the *CAT Finder* the resolution is shown only for p_t <4 GeV, see Appendix F for details



(a) $\eta(p_t)$ as function of p_t in the barrel.

(b) $\eta(p_z)$ as function of p_t in the barrel.

(c) $\eta(p_t)$ as function of displacement v_ρ in the barrel.

(d) $\eta(p_z)$ as function of v_ρ in the barrel.

Fig. 12 Relative resolution of (first column) transverse and (second column) longitudinal momentum as function of simulated transverse momentum p_t^{MC} (top row) and simulated displacement v_ρ^{MC} (bottom row) for displaced tracks from $h \to \mu^+\mu^-$ decays with *high data beam backgrounds* in the barrel for tracks found by both (red) *CAT Finder* and (blue) *Baseline Finder* for the intersecting sample. For the *CAT Finder* the resolution is shown only for p_t <4 GeV, see Appendix F for details

Baseline Finder sample is too small to provide meaningful information.

The $\eta(p_t)$ and $\eta(p_z)$ resolutions for dark Higgs tracks before track fitting is better for the *CAT Finder* for all transverse momenta and displacement regions, but for very small displacements, where the *Baseline Finder* shows a better resolution.

The $K_S^0 \to \pi^+\pi^-$ sample features on average a much smaller displacement than the $h \to \mu^+\mu^-$ sample which is visible in the better transverse and longitudinal momentum resolutions for both the *CAT Finder* and the *Baseline Finder*.

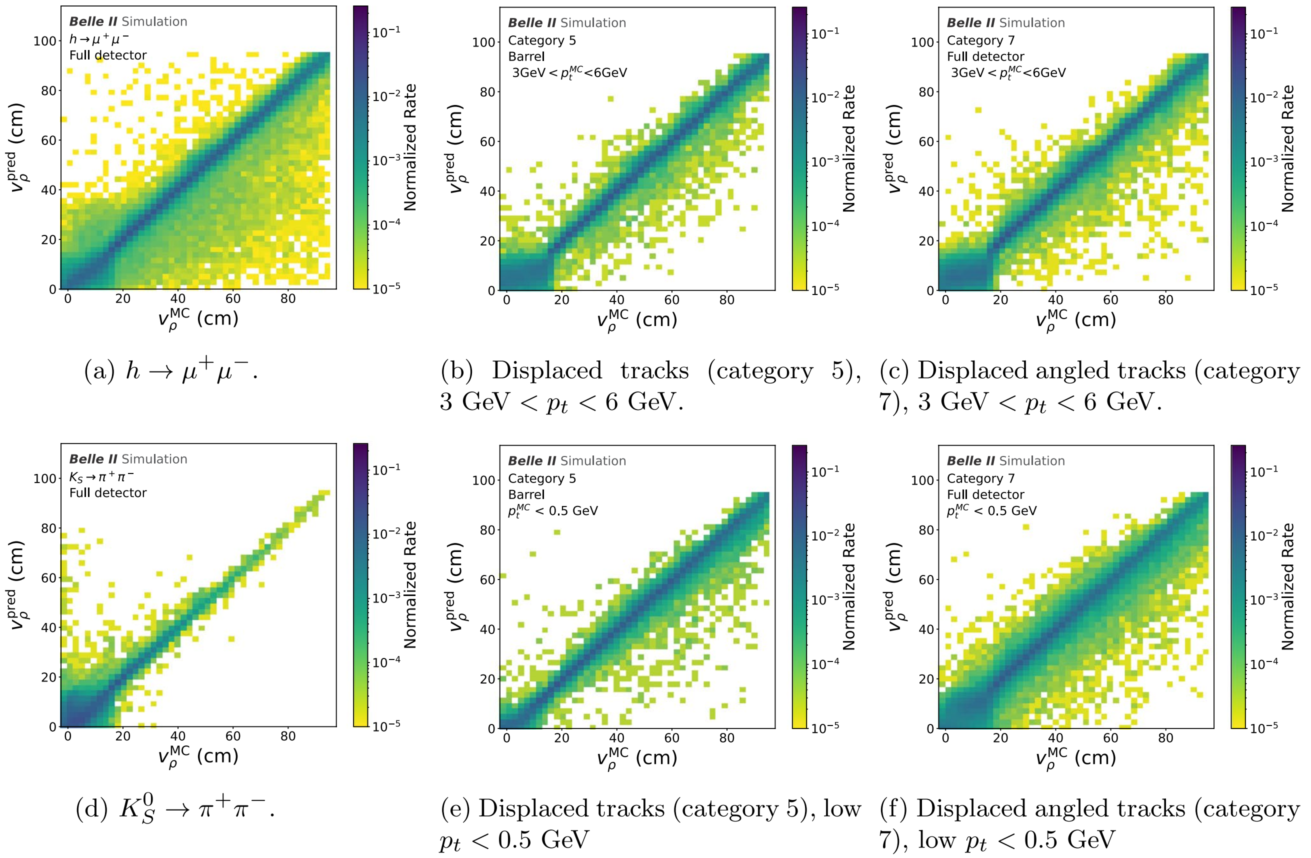
After track fitting with GENFIT2, the transverse and the longitudinal momentum resolutions are again similar for the two algorithms.

Position Reconstruction

In this chapter, we evaluate only the position prediction of the *CAT Finder*, which is directly derived from the GNN output. Position information is not available from either the *Baseline Finder* or post-fitting for any approach, as the tracks are described as a helix without a defined starting point.

The position prediction of the *CAT Finder* for truth-matched displaced tracks are shown in Fig. 13 for different displaced samples.

For the sample with a displaced vertex ($h \to \mu^+\mu^-$ and $K_S^0 \to \pi^+\pi^-$), the *CAT Finder* is able to provide an unbiased prediction with reasonable resolution even in the inner part of the detector with no close-by CDC hits (see Fig. 13a and b). This indicates that the *CAT Finder* utilizes the information of nearby detector hits from the second track to actually infer a vertex position and not just the track starting position. For individual tracks, we observe a rather complex and non-trivial behaviour of the GNN for particles with low transverse momentum: For tracks originating within the CDC volume, the GNN learns a helical representation of the parameters, with the starting point located anywhere along the trajectory and the momentum vector tangential to the helix. Since the training samples are enhanced with prompt events and are biased towards tracks originating from the interaction point, with the negative momentum vector directed towards it, the network can infer the starting point by selecting it such that the negative momentum vector points back to the interaction point (see Fig. 14). As a result, the *CAT Finder* is able to infer the track starting point even



(a) $h \to \mu^+\mu^-$.

(b) Displaced tracks (category 5), 3 GeV $< p_t <$ 6 GeV.

(c) Displaced angled tracks (category 7), 3 GeV $< p_t <$ 6 GeV.

(d) $K_S^0 \to \pi^+\pi^-$.

(e) Displaced tracks (category 5), low $p_t <$ 0.5 GeV

(f) Displaced angled tracks (category 7), low $p_t <$ 0.5 GeV

Fig. 13 Two-dimensional histograms showing the correlation between the reconstructed position of the *CAT Finder* model output v_ρ^{pred} in the $x-y$ plane and the simulated position v_ρ^{MC} for **a** $h \to \mu^+\mu^-$, **b** displaced tracks with high transverse momentum, **c** displaced angled tracks with high transverse momentum, **d** $K_S^0 \to \pi^+\pi^-$, **e** displaced tracks with low transverse momentum, and **f** displaced angled tracks with low transverse momentum, each with *high data beam backgrounds*

for $v_\rho^{MC} \lesssim 20$ cm for tracks with low transverse momentum (see Fig. 13e). This is no longer the case for particles with higher p_t, because the trajectory increasingly approaches a straight line, with a constant momentum-direction vector along the particle trajectory almost everywhere from 0 to 16 cm, where the CDC starts (see Fig. 13b).

As a cross check, we tested the *CAT Finder* performance on displaced tracks with non-pointing momentum vectors[2] and observe no predictive power for $v_\rho^{MC} \lesssim 20$ cm as expected (see Fig. 13c and f).

Robustness to Variable Detector Conditions

The detector conditions are changing during Belle II data taking due to changes in accelerator settings and during beam injections resulting in variations of the beam background conditions. In addition, the hit position resolutions and wire efficiencies are not constant over long periods of data taking. The reconstruction algorithms must be robust against moderate changes. As described in section "Graph Neural Network Track Finding", we use a model pre-trained on lower background and finalize training for higher background.

While retraining models and re-optimizing hyperparameters may be required for optimal performance, we test the robustness of the *CAT Finder* by comparing models trained and tested on the same beam background and wire efficiency maps, against models trained and tested on two different beam background environments and wire efficiency maps.

The performance metrics integrated over the full momenta range are summarized in Table 6 for a model trained and evaluated on low simulated beam background. The same information for the model trained and evaluated on

[2] The samples are similar to the $h \to \mu^+\mu^-$ or $K_S^0 \to \pi^+\pi^-$ samples but with only one track instead of a decay into two particles.

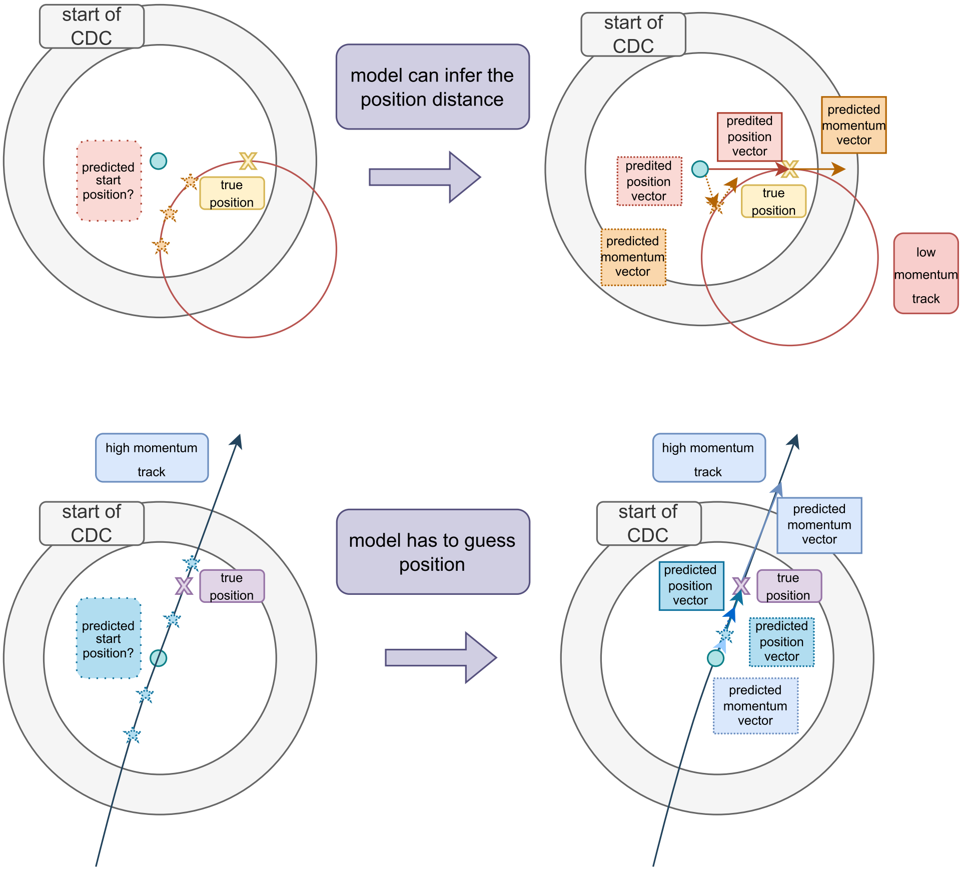

start of CDC
predicted start position?
true position
model can infer the position distance
start of CDC
predited position vector
predicted momentum vector
predicted position vector
true position
predicted momentum vector
low momentum track
high momentum track
start of CDC
true position
predicted start position?
model has to guess position
high momentum track
start of CDC
predicted momentum vector
predicted position vector
true position
predicted position vector
predicted momentum vector

Fig. 14 Illustration how the GNN learns to predict the starting position for particles with low transverse momentum even without the presence of another nearby track from a common vertex

high data beam background is given in Table 3. For a model trained on low simulated beam background but evaluated on high data beam background, the information is given in Table 7.

The track performance when moving from low simulated beam background to high data beam background (i.e. comparing Tables 3 and 6) decreases significantly for the *Baseline Finder* in the endcaps. A performance reduction is also visible for the *CAT Finder*, but to a much smaller extend.

When evaluating the model pretrained on low simulated beam background on high data beam background, the performance is slightly worse compared to the model trained on high data beam background (i.e. comparing Tables 3 and 7). The track fitting efficiency in the barrel is reduced by about 1 percent point, in the endcaps by about 3–4 percent points. The fake rate, however, is slightly lower for the pre-trained model, indicating that hyperparameter optimization may be enough to recover the efficiency loss. We note that even for the non-optimal model, the *CAT Finder* outperforms the *Baseline Finder* in all detector regions. This demonstrates a robust generalisation with respect to different levels of beam background without the need to retrain the GNNs frequently.

Lessons Learned

During the training and evaluation of the GNN, we faced various challenges that impacted the model performance and interpretability of the results.

- The model exhibits overfitting to physical correlations when displaced vertex samples with physical constraints are included in the training data set. Early trainings included the $K_S^0 \rightarrow \pi^+\pi^-$ and $h \rightarrow \mu^+\mu^-$ evaluation samples (see section "Data Set") also in the training data sets. The model identified that only $h \rightarrow \mu^+\mu^-$ events exhibited displaced vertices with significant displacements and captured the absolute momentum scale of the two decay particles, which together sum to half the collision energy at Belle II. This affected only the GNN's parameter inference but not the fit results of GENFIT2 which mostly rely on the correct momentum

Table 6 The performance metrics for the prompt evaluation samples (category 1–3, see Table 1 and section "Data Set" for details) for non-curling tracks for *CAT Finder* and *Baseline Finder* in different detector regions for a model trained and evaluated on *low simulated beam-background*

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$87.05^{+0.09}_{-0.09}$	$0.83^{+0.03}_{-0.03}$	0.01	$84.97^{+0.1}_{-0.1}$	$2.39^{+0.05}_{-0.05}$
CAT Finder	$99.26^{+0.02}_{-0.02}$	$1.02^{+0.03}_{-0.03}$	$0.15^{+0.01}_{-0.01}$	$99.22^{+0.02}_{-0.02}$	$0.03_{-0.01}$
Baseline Fitter	$85.18^{+0.1}_{-0.1}$	$0.78^{+0.03}_{-0.03}$	0.01	$84.3^{+0.1}_{-0.1}$	$0.99^{+0.03}_{-0.03}$
CAT Fitter	$97.12^{+0.05}_{-0.05}$	$0.32^{+0.02}_{-0.02}$	$0.04^{+0.01}_{-0.01}$	$96.42^{+0.05}_{-0.05}$	$0.72^{+0.02}_{-0.02}$
Barrel					
Baseline Finder	$98.71^{+0.03}_{-0.03}$	$2.06^{+0.04}_{-0.04}$	0.03	$96.73^{+0.05}_{-0.05}$	$2.0^{+0.04}_{-0.04}$
CAT Finder	$99.72^{+0.01}_{-0.01}$	$2.15^{+0.04}_{-0.04}$	$0.42^{+0.02}_{-0.02}$	$99.4^{+0.02}_{-0.02}$	$0.33^{+0.02}_{-0.02}$
Baseline Fitter	$97.68^{+0.04}_{-0.04}$	$1.75^{+0.04}_{-0.04}$	0.01	$96.27^{+0.05}_{-0.05}$	$1.44^{+0.03}_{-0.03}$
CAT Fitter	$98.13^{+0.04}_{-0.04}$	$0.97^{+0.03}_{-0.03}$	$0.13^{+0.01}_{-0.01}$	$97.97^{+0.04}_{-0.04}$	$0.17^{+0.01}_{-0.01}$
Backward endcap					
Baseline Finder	$69.5^{+0.1}_{-0.1}$	$0.72^{+0.03}_{-0.03}$	0.02	$66.2^{+0.1}_{-0.1}$	$4.66^{+0.07}_{-0.07}$
CAT Finder	$98.54^{+0.03}_{-0.03}$	$0.75^{+0.02}_{-0.02}$	$0.11^{+0.01}_{-0.01}$	$98.43^{+0.03}_{-0.03}$	$0.11^{+0.01}_{-0.01}$
Baseline Fitter	$67.8^{+0.1}_{-0.1}$	$0.63^{+0.03}_{-0.03}$	0.02	$65.8^{+0.1}_{-0.1}$	$2.98^{+0.06}_{-0.06}$
CAT Fitter	$95.12^{+0.06}_{-0.06}$	$0.3^{+0.02}_{-0.02}$	$0.03_{-0.01}$	$91.66^{+0.08}_{-0.08}$	$3.63^{+0.05}_{-0.05}$

Uncertainties below <0.01% are not shown in the table

direction but not on the absolute value for the initialisation of the fit. Removing physical samples from the training resolved that issue with negligible loss of performance.

- When the model is trained solely on prompt tracks and displaced vertices, the training process is generally unstable and fails to achieve adequate performance. The inclusion of displaced and displaced-angle tracks as intermediate samples, bridging the two event topologies, helps mitigate this issue.
- The definition of the related and matched tracks (see section "Metrics") if secondary particles are produced during the full simulation has a significant impact on the training stability and performance of the model. A large number of secondary particles, primarily low-momentum electrons, produce highly localized, cluster-like energy depositions, which the model learned to interpret as track-like signatures. This in turn significantly increased the fake and clone rates and complicated optimization metrics. Removing secondary particles results in more stable training. For both the *Baseline Finder* and the *CAT Finder*, including the secondary particles in the evaluation reduces the track finding efficiency.
- Earlier versions of our trainings used separate training samples with events that either had only low transverse momentum particles or only high transverse momentum particles. Evaluation on these samples showed good performance, but evaluation on samples that contained events with both low and high transverse momentum particles showed very low efficiency for low momentum particles. We attribute this behaviour to the fact that the training sample with high transverse momentum signal tracks contained low transverse momentum beam background tracks, but not vice versa. Enriching all samples with a rather large number of signal tracks with low transverse momentum mitigated this problem.
- The network surprised us with the ability to predict the track starting point even if no detector hits or additional tracks forming a vertex were nearby. We described the underlying mechanism in section "Position Reconstruction" and conclude that this is an interesting feature with no real practical relevance, since physical samples will not contain such tracks that violate momentum conservation.
- We observed up to a 10% variation in validation loss due to differences in repeated training and model initialization, leading to a 0.5% variation in track finding and fitting charge efficiency across specific samples and detector regions. Since the loss was not the primary target of our optimization, we accepted this variation. Potential mitigation strategies, such as improved model initialization, repeated trainings, or larger batch sizes, can be explored in future work.

Table 7 Performance metrics for the evaluation samples for different track finding algorithms in different detector regions evaluated on *high data beam background*, but trained on *low simulated beam background* for non-curling tracks

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$80.1^{+0.1}_{-0.1}$	$0.55^{+0.02}_{-0.02}$	0.01	$78.4^{+0.1}_{-0.1}$	$2.06^{+0.04}_{-0.04}$
CAT Finder	$97.73^{+0.04}_{-0.04}$	$1.46^{+0.03}_{-0.03}$	$0.59^{+0.02}_{-0.02}$	$97.18^{+0.05}_{-0.05}$	$0.56^{+0.02}_{-0.02}$
Baseline Fitter	$78.1^{+0.1}_{-0.1}$	$0.49^{+0.02}_{-0.02}$	0.01	$77.1^{+0.1}_{-0.1}$	$1.37^{+0.04}_{-0.04}$
CAT Fitter	$93.89^{+0.07}_{-0.07}$	$0.22^{+0.01}_{-0.01}$	$0.19^{+0.01}_{-0.01}$	$91.93^{+0.08}_{-0.08}$	$2.09^{+0.04}_{-0.04}$
Barrel					
Baseline Finder	$97.97^{+0.04}_{-0.04}$	$2.31^{+0.04}_{-0.04}$	$0.05^{+0.01}_{-0.01}$	$95.92^{+0.06}_{-0.06}$	$2.09^{+0.04}_{-0.04}$
CAT Finder	$99.5^{+0.02}_{-0.02}$	$2.25^{+0.04}_{-0.04}$	$1.94^{+0.04}_{-0.04}$	$98.69^{+0.03}_{-0.03}$	$0.82^{+0.03}_{-0.03}$
Baseline Fitter	$96.88^{+0.05}_{-0.05}$	$1.83^{+0.04}_{-0.04}$	0.03	$95.5^{+0.06}_{-0.06}$	$1.42^{+0.03}_{-0.03}$
CAT Fitter	$96.89^{+0.05}_{-0.05}$	$0.76^{+0.02}_{-0.02}$	$0.45^{+0.02}_{-0.02}$	$96.46^{+0.05}_{-0.05}$	$0.44^{+0.02}_{-0.02}$
Backward endcap					
Baseline Finder	$60.5^{+0.1}_{-0.1}$	$1.08^{+0.04}_{-0.04}$	$0.03^{+0.01}_{-0.01}$	$58.0^{+0.1}_{-0.1}$	$4.08^{+0.07}_{-0.07}$
CAT Finder	$95.02^{+0.06}_{-0.06}$	$1.03^{+0.03}_{-0.03}$	$0.4^{+0.02}_{-0.02}$	$94.28^{+0.06}_{-0.06}$	$0.78^{+0.02}_{-0.03}$
Baseline Fitter	$58.8^{+0.1}_{-0.1}$	$0.92^{+0.03}_{-0.04}$	$0.02_{-0.01}$	$56.8^{+0.1}_{-0.1}$	$3.28^{+0.06}_{-0.06}$
CAT Fitter	$87.55^{+0.09}_{-0.09}$	$0.44^{+0.02}_{-0.02}$	$0.16^{+0.01}_{-0.01}$	$82.8^{+0.1}_{-0.1}$	$5.4^{+0.07}_{-0.07}$

Uncertainties below <0.01% are not shown in the table

- The hyper-parameter optimization of the latent space parameters and in particular of the working points t_D and t_h (see section "Graph Neural Network Track Finding") is very challenging if one wants to balance optimal performance not only for prompt tracks, but also for different displaced signatures. On the other hand, it is rather straight forward to optimize the GNN for specific signatures if physics requirements can be restricted to certain decay topologies or momentum ranges.

Conclusion and Outlook

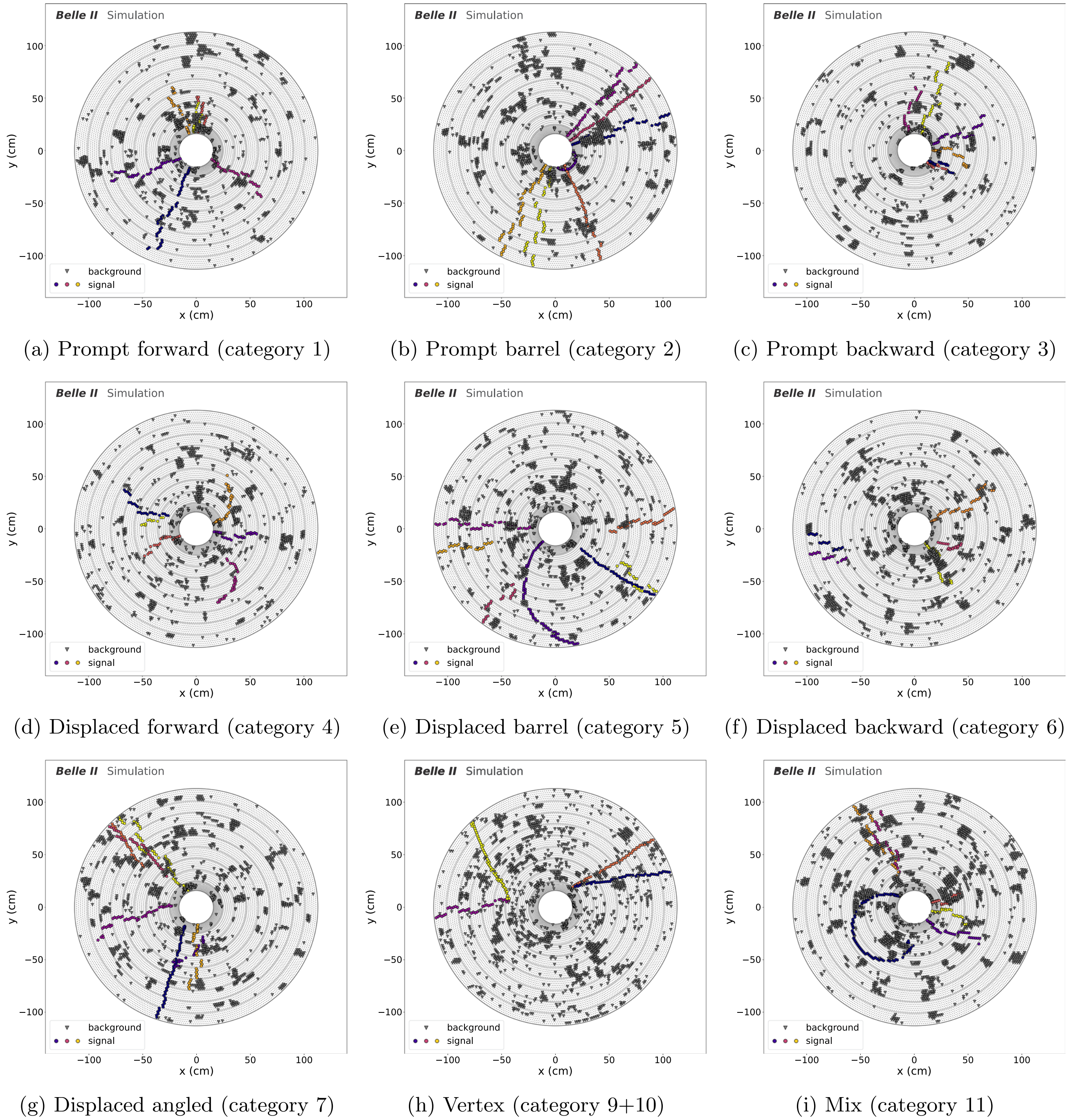
We have presented the implementation and a detailed study of the *CAT Finder* (*CDC AI Tracking*), an end-to-end multi-track reconstruction algorithm utilizing graph neural networks (GNNs) for the Belle II central drift chamber. The *CAT Finder* uses detector hits as inputs and simultaneously predicts the number of track candidates in an event, their associated hits, and their kinematic properties. We have used a full detector simulation and included beam backgrounds from actual collision data, and compared the GNN-based algorithm to the baseline track finding algorithm currently used in Belle II for a wide range of event signatures, including tracks from decays that are macroscopically displaced from the interaction point. We find significant improvements in track finding efficiencies for displaced tracks in the barrel, and for both prompt and displaced tracks in the detector endcaps. The combined track-finding and fitting efficiency, as well as the low fake track rates for prompt tracks, are both comparable to the existing Belle II track-finding method in the barrel region. To fully capitalize on the enhanced track-finding efficiency of the *CAT Finder*, future adjustments to the track fitting procedure will be necessary.

This work represents a significant conceptual step towards enabling end-to-end real-time GNN-based tracking on FPGAs [49]. For real-time trigger applications, the clusterisation based on condensation points, the hit ordering in real space, and the track fitting step can be omitted if the inferred kinematics from the GNN provide sufficient resolution for trigger decisions. However, future work is needed on input quantization, pre-processing to remove beam background, as well as network size reduction.

To our knowledge, the *CAT Finder* is the first end-to-end machine learning tracking algorithm that has been utilized in a realistic particle physics environment, and the first completely GNN-based track finding in a drift chamber detector.

Appendix A: Data Set Event Displays

Typical event displays showing examples of the different training samples as described in section "Data Set" and Table 1 are shown in Fig. 15.



(a) Prompt forward (category 1)

(b) Prompt barrel (category 2)

(c) Prompt backward (category 3)

(d) Displaced forward (category 4)

(e) Displaced barrel (category 5)

(f) Displaced backward (category 6)

(g) Displaced angled (category 7)

(h) Vertex (category 9+10)

(i) Mix (category 11)

Fig. 15 Typical event displays showing examples of the different training samples for *high data beam backgrounds*. Filled colored circular markers show signal hits, filled gray triangular markers show background hits. The markers correspond to the locations of the sense wires at the z position of the center of the wire for the wires with recorded ADC signals

Appendix B: CDC Wire Inefficiencies

Displays of the wire efficiencies and dead wires used for the low simulated beam background conditions are shown in Fig. 16. Displays of the wire efficiencies and dead wires used for the high data beam background conditions are shown in Fig. 17.

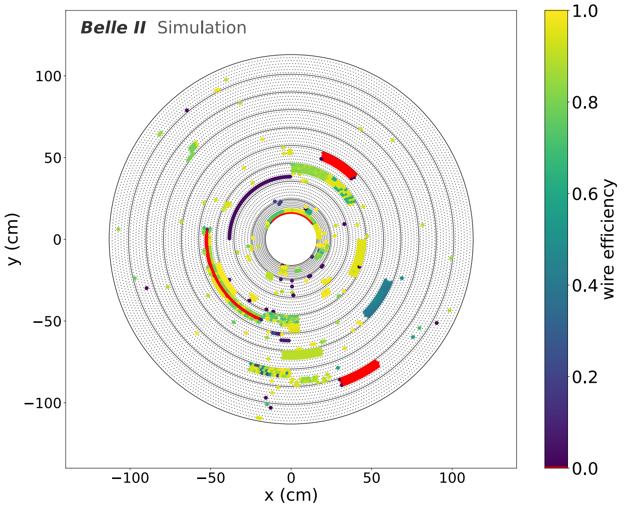


Fig. 16 Overview of the simulated wire efficiency for the default simulated samples. Coloured wires have an efficiency below 1. Red wires have an efficiency of 0

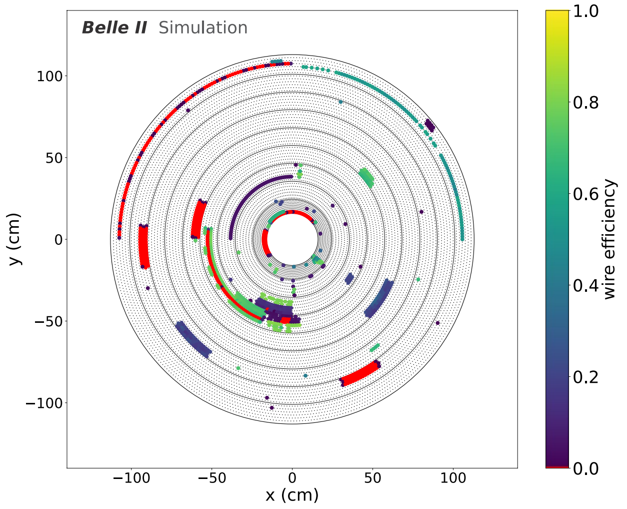
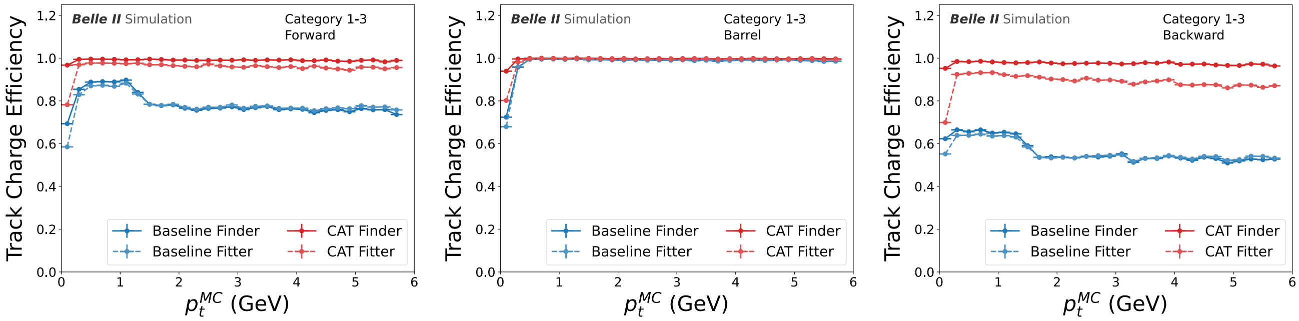


Fig. 17 Overview of the simulated wire efficiency for the simulated samples with beam background and detector conditions taken from data. Coloured wires have an efficiency below 1. Red wires have an efficiency of 0

Appendix C: Track Charge Efficiency for Category 1–3

The track finding charge efficiencies, and the combined track finding and track fitting charge efficiencies for the *Baseline Finder* in comparison with the *CAT Finder* are shown in Fig. 18 for non-curling tracks from category 1 to 3.



(a) Forward endcap.

(b) Barrel.

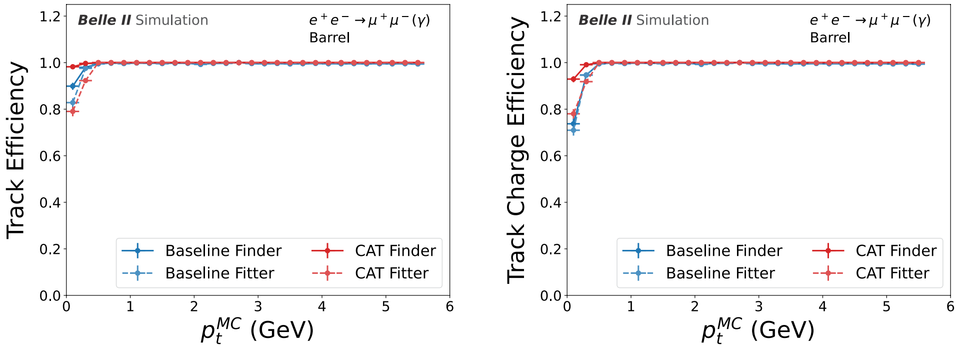
(c) Backward endcap.

Fig. 18 Track finding (empty markers, connected by solid lines to guide the eye) and combined track finding and fitting charge efficiency (filled markers, connected by dashed lines to guide the eye) for the prompt evaluation samples (category 1–3, see Table 1, *high data beam backgrounds*) with curler tracks removed, as function of simulated transverse momentum p_t^{MC} for the *Baseline Finder* (blue) and the *CAT Finder* (red) in the **a** forward endcap, **b** barrel, and **c** backward endcap. The vertical error bars that show the statistical uncertainty are smaller than the marker size. The horizontal error bars indicate the bin width. The uncertainties of the different track finding algorithms are correlated, since they use the same simulated events

Appendix D: Efficiencies, Fake Rates, and Clone Rates for $\mu^-\mu^+(\gamma)$

The track finding charge efficiencies, and the combined track finding and track fitting charge efficiency for the *Baseline Finder* in comparison with the *CAT Finder* are shown in Fig. 19.

Track finding and fitting efficiency ε_{trk}, fake rate $\mathfrak{r}_{\text{fake}}$, clone rate $\mathfrak{r}_{\text{clone}}$, track charge efficiency $\varepsilon_{\text{trk,ch}}$ and wrong charge rate $\mathfrak{r}_{\text{wrong ch.}}$ integrated over the full p_t for $\mu^-\mu^+(\gamma)$ events are shown in Table 8.



(a) Track finding and fitting efficiency.

(b) Track finding and fitting charge efficiency.

Fig. 19 Track finding (empty markers, connected by lines to guide the eye) and combined track finding and fitting charge efficiency (filled markers) for $\mu^-\mu^+(\gamma)$ evaluation sample in the barrel with *high data beam backgrounds*. See Fig. 7 caption for details

Table 8 The performance metrics for the $\mu^-\mu^+(\gamma)$ evaluation samples for different track finding algorithms in different detector regions for low beam background

(in %)	ε_{trk}	$\mathfrak{r}_{\text{fake}}$	$\mathfrak{r}_{\text{clone}}$	$\varepsilon_{\text{trk,ch}}$	$\mathfrak{r}_{\text{wrong ch.}}$
Forward endcap					
Baseline Finder	$84.0^{+0.2}_{-0.2}$	$1.31^{+0.07}_{-0.08}$	$0.01^{+0.01}_{-0.01}$	$82.1^{+0.2}_{-0.2}$	$2.25^{+0.1}_{-0.1}$
CAT Finder	$99.91^{+0.02}_{-0.02}$	$8.1^{+0.2}_{-0.2}$	$0.01^{+0.01}_{-0.01}$	$99.89^{+0.02}_{-0.02}$	$0.02^{+0.01}_{-0.01}$
Baseline Fitter	$83.5^{+0.2}_{-0.2}$	$0.98^{+0.06}_{-0.07}$	$0.01^{+0.01}_{-0.01}$	$82.3^{+0.2}_{-0.2}$	$1.4^{+0.08}_{-0.08}$
CAT Fitter	$99.34^{+0.05}_{-0.05}$	$1.21^{+0.07}_{-0.07}$	$0.0_{-0.01}$	$97.21^{+0.1}_{-0.1}$	$2.14^{+0.09}_{-0.09}$
Barrel					
Baseline Finder	$99.51^{+0.02}_{-0.02}$	$2.91^{+0.05}_{-0.05}$	$0.05^{+0.01}_{-0.01}$	$99.4^{+0.02}_{-0.02}$	$0.11^{+0.01}_{-0.01}$
CAT Finder	99.99	$5.11^{+0.07}_{-0.07}$	$0.04^{+0.01}_{-0.01}$	$99.96^{+0.01}_{-0.01}$	$0.02_{-0.01}$
Baseline Fitter	$99.48^{+0.02}_{-0.02}$	$2.11^{+0.04}_{-0.04}$	$0.05^{+0.01}_{-0.01}$	$99.4^{+0.02}_{-0.02}$	$0.08^{+0.01}_{-0.01}$
CAT Fitter	$99.85^{+0.01}_{-0.01}$	$1.78^{+0.04}_{-0.04}$	0.01	$99.84^{+0.01}_{-0.01}$	0.02
Backward endcap					
Baseline Finder	$66.4^{+0.3}_{-0.3}$	$4.1^{+0.2}_{-0.2}$	$0.02^{+0.01}_{-0.01}$	$62.8^{+0.3}_{-0.3}$	$5.4^{+0.2}_{-0.2}$
CAT Finder	$99.73^{+0.03}_{-0.03}$	$7.4^{+0.2}_{-0.2}$	0.0	$99.68^{+0.04}_{-0.03}$	$0.05^{+0.01}_{-0.02}$
Baseline Fitter	$65.7^{+0.3}_{-0.3}$	$3.6^{+0.1}_{-0.1}$	$0.02^{+0.01}_{-0.01}$	$63.3^{+0.3}_{-0.3}$	$3.6^{+0.1}_{-0.1}$
CAT Fitter	$97.88^{+0.09}_{-0.09}$	$4.0^{+0.1}_{-0.1}$	0.0	$92.9^{+0.2}_{-0.2}$	$5.1^{+0.1}_{-0.1}$

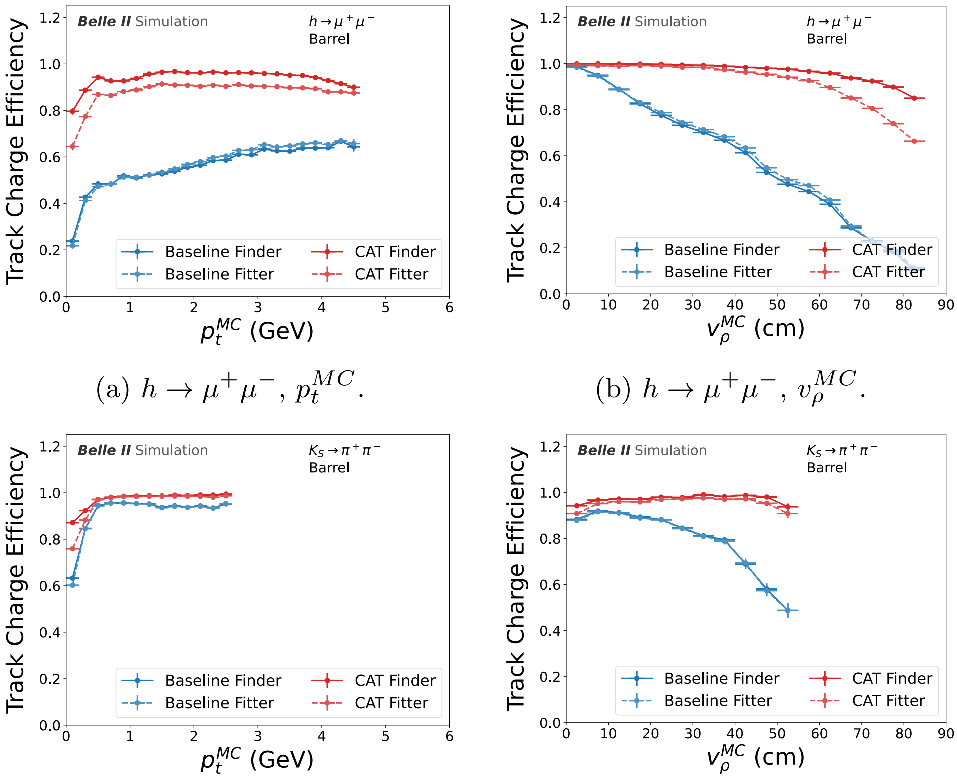
Uncertainties below <0.01% are not shown in the table

Appendix E: Track Charge Efficiency for $h \rightarrow \mu^+\mu^-$ and $K_S^0 \rightarrow \pi^+\pi^-$

The track finding efficiencies, and the combined track finding and track fitting efficiency for the *Baseline Finder* in comparison with the *CAT Finder* for $h \rightarrow \mu^+\mu^-$ and $K_S^0 \rightarrow \pi^+\pi^-$ events are shown in Fig. 20.

Appendix F: High Transverse Momentum Track Resolution

Figure 21 shows the relative transverse momentum resolution for tracks found and fitted by both the *CAT Finder* and the *Baseline Finder* in Fig. 21a–f) and the relative longitudinal momentum resolution in the transverse momentum bin of 4GeV < p_t <6 GeV. This can be observed in Fig. 21c, where there is no significant tail on the right side. We note that even if the actual momentum was significantly higher than 6 GeV, the initial prediction from the *CAT Finder* would be sufficient as starting value for the subsequent track fitting algorithm. While the central part of the resolution distribution for the CAT Finder is broader compared to the other cases, the distribution has significantly smaller tails.



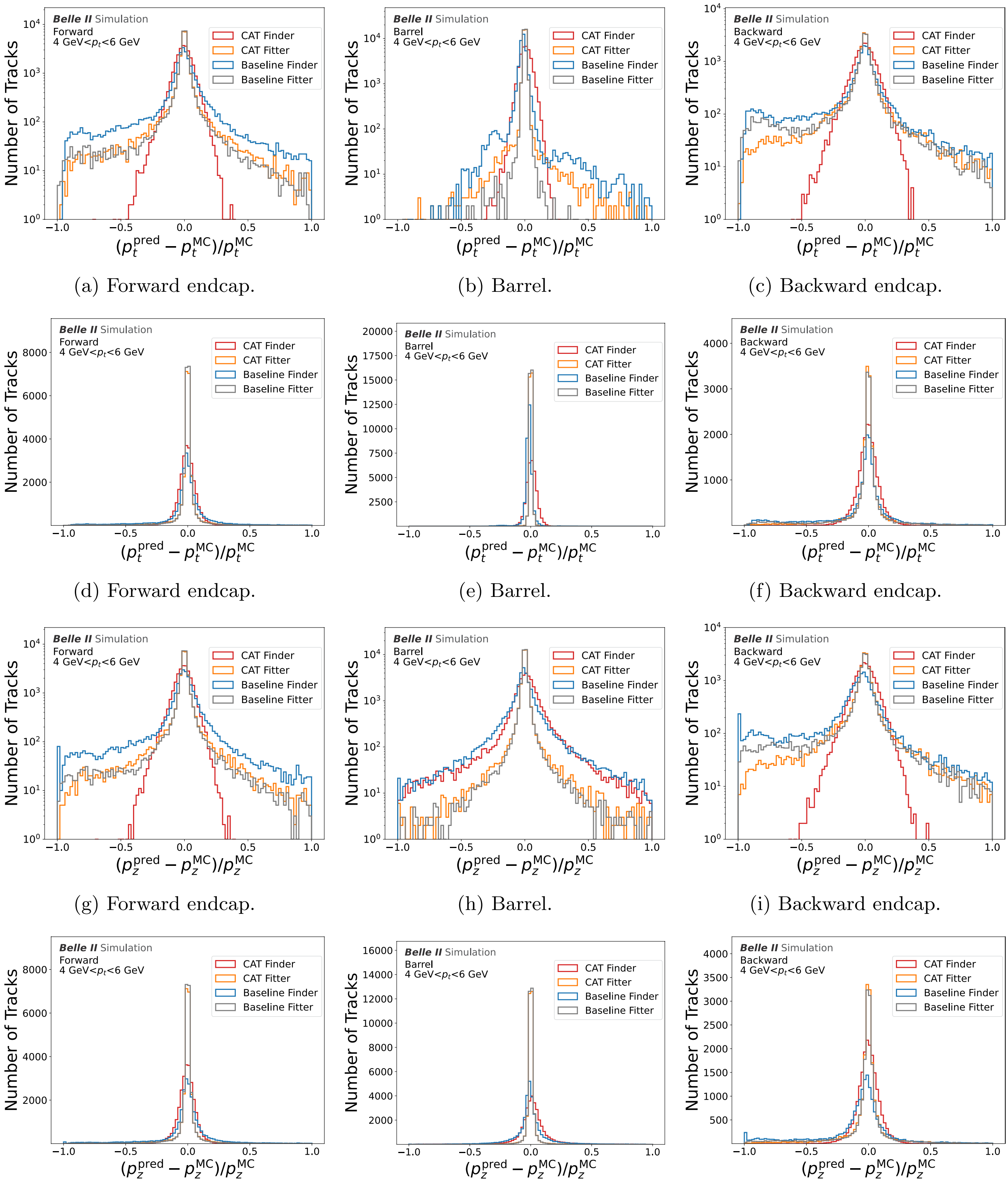
(a) $h \rightarrow \mu^+\mu^-$, p_t^{MC}.

(b) $h \rightarrow \mu^+\mu^-$, v_ρ^{MC}.

(c) $K_S^0 \rightarrow \pi^+\pi^-$, p_t^{MC}.

(d) $K_S^0 \rightarrow \pi^+\pi^-$, v_ρ^{MC}.

Fig. 20 Track finding (empty markers) and combined track finding and fitting charge efficiency (filled markers) for (top) displaced tracks in $h \rightarrow \mu^+\mu^-$ events and in (bottom) $K_S^0 \rightarrow \pi^+\pi^-$ events with *high data beam backgrounds*, as function of (left) the true simulated transverse momentum p_t^{MC}, and (right) the true simulated displacement v_ρ^{MC}



(a) Forward endcap.

(b) Barrel.

(c) Backward endcap.

(d) Forward endcap.

(e) Barrel.

(f) Backward endcap.

(g) Forward endcap.

(h) Barrel.

(i) Backward endcap.

(j) Forward endcap.

(k) Barrel.

(l) Backward endcap.

Fig. 21 Relative **a–f** transverse and **g–l** longitudinal momentum resolution as function of simulated transverse momentum p_t^{MC} for the intersecting prompt evaluation sample (category 1–3, see Table 1) in the (left) forward endcap, (center) barrel, and (right) backward endcap for tracks found (red) and fitted (orange) by both the *CAT Finder* and (blue and grey) the *Baseline Finder* for the high transverse momentum bin of 4 GeV< p_t < 6 GeV

Appendix G: Track Helix Parametrization Resolution

The track parametrization follows a helix model, computed at the point of closest approach (POCA) to the collision point. We define the distance between the POCA and the collision point on the transverse (longitudinal) plane as d_0 (d_z), and the angle defined by the transverse (longitudinal) momentum at the POCA as ϕ (θ). We evaluate the absolute residuals for these track features ϕ, θ, d_0, and d_z

$$\eta(\phi, \theta, d_0, d_z) = (\phi, \theta, d_0, d_z)_{\text{rec}} - (\phi, \theta, d_0, d_z)_{\text{simulated}} \quad (14)$$

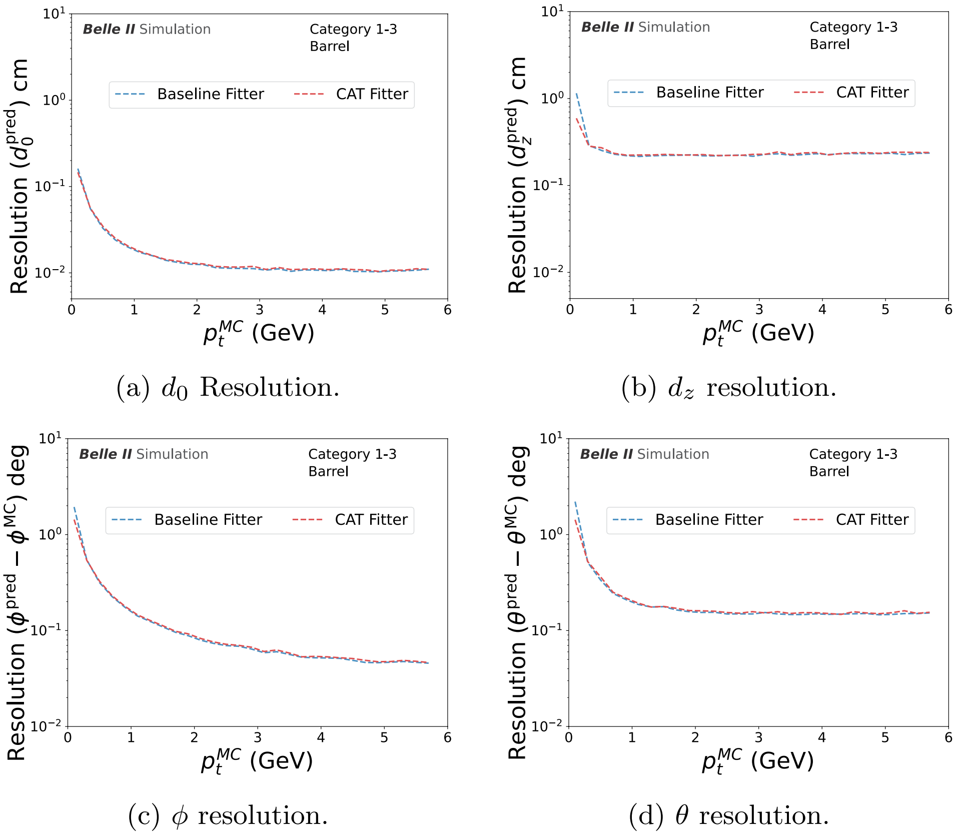
for matched tracks.

We then define the resolution $r(\phi, \theta, d_0, d_z)$ for the absolute residuals $\eta(\phi, \theta, d_0, d_z)$ as the 68% coverage

$$r(p_{\phi,\theta,d_0,d_z}) = P_{68\%}\left(\left|\eta - P_{50\%}(\eta)\right|\right), \quad (15)$$

where P_q is the q-th quantile of the distribution of $p_{t,z}$, and $P_{50\%}$ is the median of $\eta(p_{t,z})$ [29].

The fitting parameter resolutions as function of simulated transverse momentum p_t^{MC} are shown in Fig. 22.



(a) d_0 Resolution.

(b) d_z resolution.

(c) ϕ resolution.

(d) θ resolution.

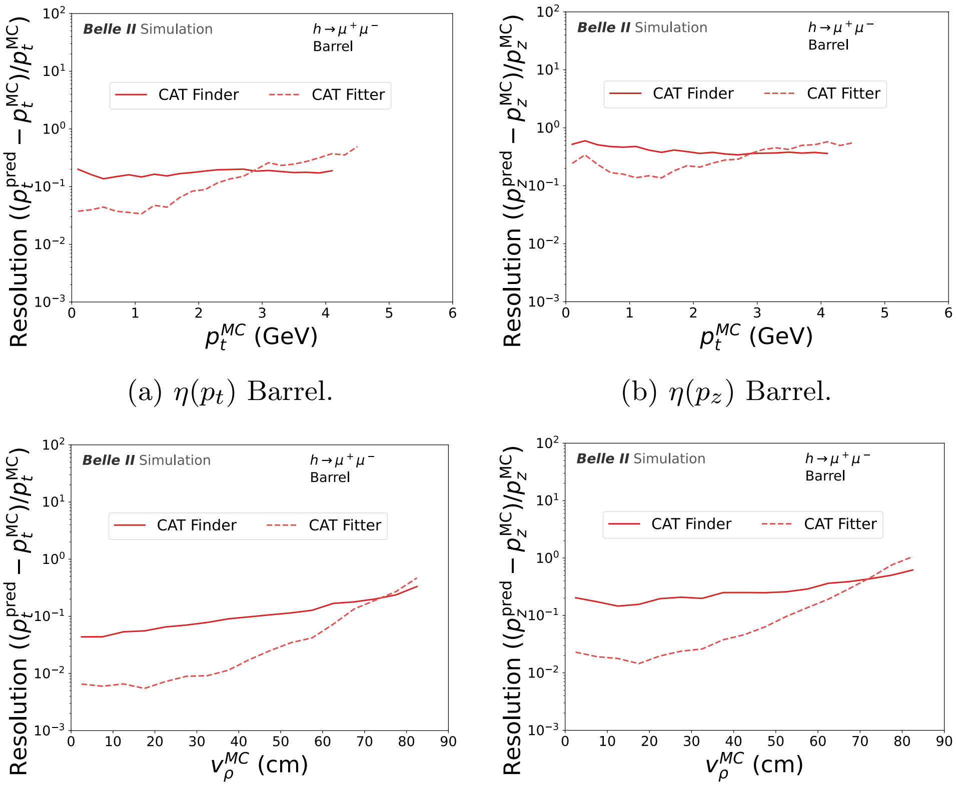
Fig. 22 Fitting parameter resolution as function of simulated transverse momentum p_t^{MC} for the intersecting prompt evaluation sample (category 1–3, see Table 1) barrel

Appendix H: Track Momentum Resolution for Additional *CAT Finder* Samples in $h \rightarrow \mu^+\mu^-$ Events

The relative momentum resolutions for displaced tracks from $h \rightarrow \mu^+\mu^-$ decays in the barrel for tracks only found by *CAT Finder* are shown in Fig. 23.

Appendix I: Track Momentum Resolution in $K_S^0 \rightarrow \pi^+\pi^-$ Events

The relative momentum resolutions for displaced tracks from $K_S^0 \rightarrow \pi^+\pi^-$ decays in the barrel are shown in Fig. 24.



(a) $\eta(p_t)$ Barrel.

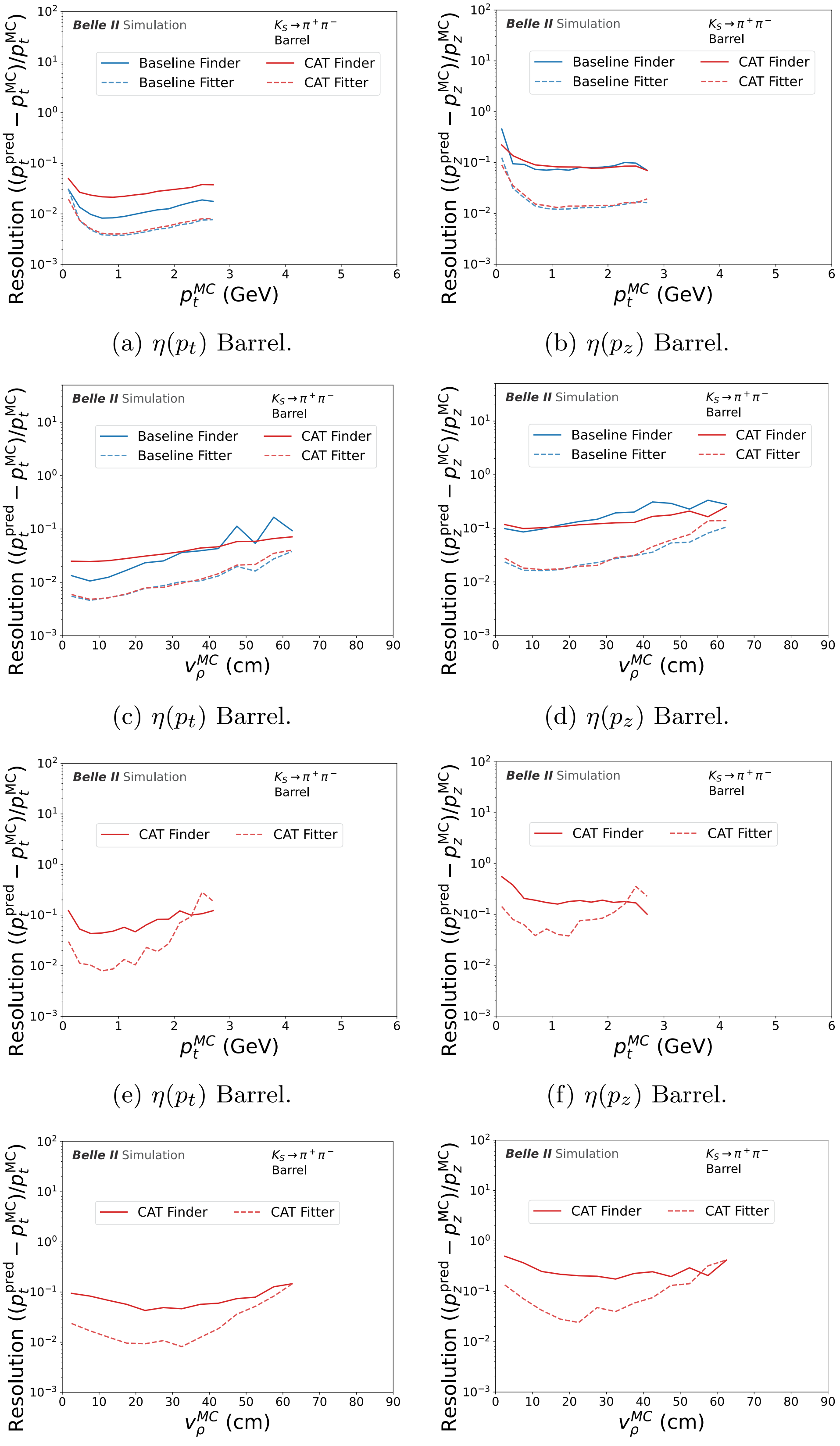
(b) $\eta(p_z)$ Barrel.

(c) $\eta(p_t)$ Barrel.

(d) $\eta(p_z)$ Barrel.

Fig. 23 Relative resolution of (first column) transverse and (second column) longitudinal momentum as function of (top row) simulated transverse momentum p_t^{MC} and (bottom row) simulated displacement v_ρ^{MC} for displaced tracks from $h \rightarrow \mu^+\mu^-$ decays in the barrel for tracks only found by *CAT Finder*

Fig. 24 Relative resolution of (first column) transverse and (second column) longitudinal momentum as function of (top row) simulated transverse momentum p_t^{MC} and (bottom row) simulated displacement v_ρ^{MC} for displaced tracks from $K_S^0 \to \pi^+\pi^-$ decays. Top **a**–**d** row shows the resolution for tracks found by both *CAT Finder* (red) and *Baseline Finder* (blue), and bottom **e**–**h** row for tracks only found by *CAT Finder*



(a) $\eta(p_t)$ Barrel.

(b) $\eta(p_z)$ Barrel.

(c) $\eta(p_t)$ Barrel.

(d) $\eta(p_z)$ Barrel.

(e) $\eta(p_t)$ Barrel.

(f) $\eta(p_z)$ Barrel.

(g) $\eta(p_t)$ Barrel.

(h) $\eta(p_z)$ Barrel.

Acknowledgements It is a great pleasure to thank (in alphabetical order) Isabel Haide, Jan Kieseler, and Yannis Kluegl for discussions. The training of the models was performed on the TOpAS GPU cluster at the Scientific Computing Center (SCC) at KIT.

Author Contributions L. R. developed the GNN implementation. L. R., G. D. P. , S. S. and T.F. wrote the manuscript and contributed to the analysis of the results. All authors contributed to the tracking algorithm development, implementation in the Belle II software framework and the operation of the tracking detectors. All authors provided critical feedback to the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This work is funded by BMBF ErUM-Pro 05H24VK1.

Data Availability The data sets generated and analysed during the current study are property of the Belle II collaboration and not publicly available. The Belle II software is available at [32, 33]. The instructions and code to replicate the studies in this paper are available at [50].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ferber T, Garcia-Cely C, Schmidt-Hoberg K (2022) Belle II sensitivity to long-lived dark photons. Phys Lett B 833:137373
2. Duerr M et al (2020) Invisible and displaced dark matter signatures at Belle II. J High Energy Phys 2:39
3. Duerr M et al (2021) Long-lived dark Higgs and inelastic dark matter at Belle II. J High Energy Phys 4:146
4. Natochii A et al (2022) Beam background expectations for Belle II at SuperKEKB. https://arxiv.org/abs/2203.05731
5. Shlomi J, Battaglia P, Vlimant J-R (2020) Graph neural networks in particle physics. Mach Learn Sci Technol 2(2):021001
6. Wang Y et al (2018) Dynamic graph CNN for learning on point clouds. https://arxiv.org/abs/1801.07829
7. Qasim SR et al (2019) Learning representations of irregular particle-detector geometry with distance-weighted graph networks. Eur Phys J C 79(7):608
8. Wemmer F et al (2023) Photon reconstruction in the Belle II calorimeter using graph neural networks. Comput Softw Big Sci 7:13
9. Kieseler J (2020) Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data. Eur Phys J C 80(9):886
10. Qasim SR et al (2022) End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks. Eur Phys J C 82(8):753
11. Amrouche S et al (2019) The tracking machine learning challenge: accuracy phase. In: The NeurIPS '18 competition. pp 231–264
12. Amrouche S et al (2023) The tracking machine learning challenge: throughput phase. Comput Softw Big Sci 7(1):1
13. Choma N et al (2020) Track seeding and labelling with embedded-space graph neural networks. https://arxiv.org/abs/2007.00149
14. Caillou S et al (2024) Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC. EPJ Web Conf 295:09028
15. Ju X et al (2021) Performance of a geometric deep learning pipeline for HL-LHC particle tracking. Eur Phys J C 81(10):876
16. Lieret K et al (2023) High pileup particle tracking with object condensation. https://arxiv.org/abs/2312.03823
17. Correia A et al (2024) Graph neural network-based track finding in the LHCb vertex detector. https://arxiv.org/abs/2407.12119
18. Caillou S et al (2024) Physics performance of the ATLAS GNN4ITk track reconstruction chain. EPJ Web Conf 295:03030
19. Akram A, Ju X (2022) Track reconstruction using geometric deep learning in the straw tube tracker (STT) at the PANDA experiment. https://arxiv.org/abs/2208.12178
20. Jia X et al (2024) BESIII track reconstruction algorithm based on machine learning. EPJ Web Conf 295:09006
21. Kaneko F et al (2024) Extracting signal electron trajectories in the COMET Phase-I cylindrical drift chamber using deep learning. https://arxiv.org/abs/2408.04795
22. Lieret K, DeZoort G (2024) An object condensation pipeline for charged particle tracking at the high luminosity LHC. EPJ Web Conf 295:09004
23. Huang A et al (2024) A language model for particle tracking. https://arxiv.org/abs/2402.10239
24. Caron S et al (2024) TrackFormers: in search of transformer-based particle tracking for the high-luminosity LHC era. https://arxiv.org/abs/2407.07179
25. Keck T (2017) FastBDT: a speed-optimized multivariate classification algorithm for the Belle II experiment. Comput Softw Big Sci 1(1):2
26. Bähr S et al (2024) The neural network first-level hardware track trigger of the Belle II experiment. https://arxiv.org/abs/2402.14962
27. HEP ML Community (2025) A living review of machine learning for particle physics. https://iml-wg.github.io/HEPML-LivingReview/
28. Abe T et al (2010) Belle II technical design report. https://arxiv.org/abs/1011.0352
29. Bertacchi V et al (2021) Track finding at Belle II. Comput Phys Commun 259:107610
30. Kou E et al (2019) The Belle II physics book. Prog Theor Exp Phys 2019:123C01 (**Erratum: Prog. Theor. Exp. Phys. 2020, 029201 (2020)**)
31. Agostinelli S et al (2003) GEANT4—a simulation toolkit. Nucl Instrum Methods A 506:250–303
32. Kuhr T et al (2019) The Belle II core software. Comput Softw Big Sci 3(1):1–12
33. Belle II Collaboration (2022) Belle II analysis software framework (basf2) (release-06-00-09). https://doi.org/10.5281/zenodo.6949513
34. Jadach S, Ward BFL, Wąs Z (2000) The precision Monte Carlo event generator KK for two-fermion final states in e^+e^- collisions. Comput Phys Commun 130:260
35. Alwall J et al (2014) The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. J High Energy Phys 07:079
36. Workman RL et al (2022) Review of particle physics. Prog Theor Exp Phys 2022:083C01
37. Liptak ZJ et al (2022) Measurements of beam backgrounds in SuperKEKB Phase 2. Nucl Instrum Methods Phys Res A 1040:167168
38. Hoppner C et al (2010) A novel generic framework for track fitting in complex detector systems. Nucl Instrum Methods Phys Res A 620:518–525

39. Rauch J, Schlüter T (2015) GENFIT—a generic track-fitting toolkit. J Phys Conf Ser 608(1):012042
40. Bilka T et al (2019) Implementation of GENFIT2 as an experiment independent track-fitting framework. https://arxiv.org/abs/1902.04405
41. Jojosito et al (2023) GenFit/GenFit. https://doi.org/10.5281/zenodo.10301439
42. Alexopoulos T et al (2008) Implementation of the Legendre transform for track segment reconstruction in drift tube chambers. Nucl Instrum Methods Phys Res A 592:456–462
43. Glazov A et al (1993) Filtering tracks in discrete detectors using a cellular automaton. Nucl Instrum Methods Phys Res A 329:262–268
44. Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch geometric. https://arxiv.org/abs/1903.02428
45. Lin M, Chen Q, Yan S (2014) Network in network. https://arxiv.org/abs/1312.4400
46. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. https://arxiv.org/abs/1502.03167
47. Clevert D-A, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (ELUs). https://arxiv.org/abs/1511.07289
48. Biewald L (2020) Experiment tracking with weights and biases, 2020. Software available from https://wandb.ai/site/
49. Neu M et al (2024) Real-time graph building on FPGAs for machine learning trigger applications in particle physics. Comput Softw Big Sci 8(1):8
50. Reuter L et al (2025) Code for the paper “End-to-End Multi-Track Reconstruction using Graph Neural Networks at Belle II”. https://doi.org/10.5281/zenodo.15167005

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

L. Reuter[1] · G. De Pietro[2] · S. Stefkova[3] · T. Ferber[4] · V. Bertacchi[5] · G. Casarosa[6] · L. Corona[7] · P. Ecker[8] · A. Glazov[9] · Y. Han[10] · M. Laurenza[11] · T. Lueck[12] · L. Massaccesi[13] · S. Mondal[14] · B. Scavino[15] · S. Spataro[16] · C. Wessel[17] · L. Zani[18]

✉ L. Reuter
lea.reuter@kit.edu

G. De Pietro
giacomo.pietro@kit.edu

S. Stefkova
slavomira.stefkova@uni-bonn.de

T. Ferber
torben.ferber@kit.edu

V. Bertacchi
valerio.bertacchi@alumni.sns.it

G. Casarosa
giulia.casarosa@pi.infn.it

L. Corona
luigi.corona@pi.infn.it

P. Ecker
patrick.ecker@kit.edu

A. Glazov
alexander.glazov@belle2.org

Y. Han
yubo.han@desy.de

M. Laurenza
martina.laurenza@physics.uu.se

T. Lueck
thomas.lueck@lmu.de

L. Massaccesi
ludovico.massaccesi@pi.infn.it

S. Mondal
suryamondal@gmail.com

B. Scavino
bianca.scavino@physics.uu.se

S. Spataro
spataro@to.infn.it

C. Wessel
christian.wessel@desy.de

L. Zani
laura.zani@roma3.infn.it

[1] http://orcid.org/0000-0002-5930-6237
[2] http://orcid.org/0000-0001-8442-107X
[3] http://orcid.org/0000-0003-2628-530X
[4] http://orcid.org/0000-0002-6849-0427
[5] http://orcid.org/0000-0001-9971-1176
[6] http://orcid.org/0000-0003-4137-938X
[7] http://orcid.org/0000-0002-2577-9909
[8] http://orcid.org/0000-0002-6817-6868
[9] http://orcid.org/0000-0002-8553-7338
[10] http://orcid.org/0000-0001-6775-5932
[11] http://orcid.org/0000-0002-7400-6013
[12] http://orcid.org/0000-0003-3915-2506
[13] http://orcid.org/0000-0003-1762-4699
[14] http://orcid.org/0000-0002-3054-8400
[15] http://orcid.org/0000-0003-1771-9161
[16] http://orcid.org/0000-0001-9601-405X
[17] http://orcid.org/0000-0003-0959-4784
[18] http://orcid.org/0000-0003-4957-805X