ETP-KA/2026-10

Karlsruhe Institute of Technology

Graph Neural Network based Hit Filtering for the Belle II Central Drift Chamber

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

angenommene

Dissertation

von

M.Sc. Greta Sophie Heine

geb. in Worms

Tag der mündlichen Prüfung: 22.05.2026
Referent: Prof. Dr. Torben Ferber
Korreferent: Prof. Dr. Markus Klute

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.
Karlsruhe, 15.04.2026

. . . . . .
(M.Sc. Greta Sophie Heine)

Abstract

The Belle II experiment operates at high instantaneous luminosity, where an increasing level of beam-induced background poses significant challenges for both the offline and online track-reconstruction algorithms. This work presents the development, implementation, and performance evaluation of a hit-filtering algorithm based on graph neural networks for the central drift chamber of Belle II. The hit filter is designed for application in offline track reconstruction as well as in the Level-1 trigger tracking system, with a targeted deployment on FPGA devices. Applied to offline track reconstruction, the proposed algorithm improves track efficiency in Monte Carlo studies by up to 6.1 %, reduces the track fake rate by up to 8.0 %, and improves track resolution by up to 7.7 % for key physics channels relative to the default filter, while maintaining comparable execution time. An adapted version of the algorithm applied to online triggering improves track efficiency by up to 18 % evaluated on $\mu^+\mu^-(\gamma)$ events from late 2025, while satisfying constraints on computing resources, trigger rates, and hardware utilization. In Field-Programmable Gate Array (FPGA) implementation studies evaluated for a graph size of 820 nodes and 3 593 edges, the design utilizes 60.80 % of look-up tables, 27.23 % of flip-flops, and no DSPs at a 210.92 ns end-to-end latency and 128.008 MHz system frequency. The design thus satisfies the resource and sub-microsecond latency requirements for integration into the Belle II L1 trigger. Furthermore, this work establishes a neural network compression workflow for hardware software co-designed hit filtering in real-time trigger systems, including 4 bit weight quantization, pruning, and the BOP metric as a quantitative approximation of the model size, which is applicable to the development of future machine-learning-based trigger designs.

Disclaimer

Algorithm development and data analysis in high-energy physics, including the work in this thesis, are a collaborative effort. The SuperKEKB accelerator, which provides the beams used by the Belle II experiment, and the Belle II detector are designed, built, operated, and maintained by the SuperKEKB accelerator group and the Belle II collaboration, respectively. The Belle II collaboration also produces the centrally provided simulated and recorded datasets and maintains the computing infrastructure for their processing. The collaboration also develops and maintains the software environment essential for Belle II data analyzes. I have been a Belle II collaboration member since 2024 and performed all studies presented in this thesis. Below, I detail prior work by other researchers on which this thesis is based and analysis components conducted by collaborators:

- The procedure for mapping detector hits to hit graphs, in particular, the geometrical constraints used in graph construction, was developed by Philipp Dorwarth [1].
- The background overlays were provided by a tool developed by Jonas Eppelt [2].
- The composition of the training dataset is based on the work of Lea Reuter [3].
- The hardware implementation of the algorithm, described in Section 9.6, was carried out by Marc Neu and Fabio Mayer.

The algorithm developed in this thesis has been published in [4, 5], together with part of the results in Chapters 6, 7 and 9. I implemented my algorithm into the Belle II Analysis Software Framework together with Giacomo De Pietro.

This thesis uses Artificial Intelligence (AI) tools for grammatical and stylistic improvements and for code development. Writefull[1] was used for spell and grammar checks and to paraphrase selected sentences to improve academic clarity and precision. GitHub Copilot[3], Perplexity[2], and ChatGPT[4] were used to support the development of C++ and Python code, mainly for restructuring, documentation, and optimization tasks that do not constitute the core scientific work of this thesis. All suggestions were reviewed, tested, and approved by me to ensure robust and reliable results.

[1] Writefull: An AI writing assistant for academic and technical texts. See https://www.writefull.com/ (Access Date: 2026-04-13).

[2] GitHub Copilot: An AI pair programmer powered by large language models that is used to assist with writing code. See https://github.com/features/copilot/ (Access Date: 2026-04-13).

[3] Perplexity: An AI-powered virtual assistant that combines large language models with web and literature search. See https://www.perplexity.ai/ (Access Date: 2026-04-13).

[4] ChatGPT: A virtual AI assistant based on large language models. See https://openai.com/chatgpt/ (Access Date: 2026-04-13).

Acknowledgements

Without the people by my side, it would not have been possible for me to see this project through and complete this work in the end.
I would like to express my deepest gratitude to Torben Ferber and Markus Klute for supervising and supporting my work. Torben initially took over the formal supervision of my master's thesis, even though I was based at another institute, but quickly became the main mentor for my master's project. I would not have been able to achieve more than a small fraction of the results without being adopted into his group and benefiting from the support I received there. During my PhD, I could always turn to him for advice, and his detailed and critical feedback on my work, presentations, and publications significantly improved the quality of my work and helped me grow as a person. I am particularly grateful that, in his group, I experienced for the first time what it means to be treated as an equal as a woman in physics. From the very beginning, my expertise was taken seriously and my statements were not questioned more than those of my male peers, which made me realize how often I had previously been treated differently in professional and private contexts.
I would like to thank Markus for funding the first months of my PhD when I worked on the DELight project, which bridged the time until my scholarship funded by KSETA started. I greatly enjoyed our coffee-break discussions and outreach activities and I will fondly remember the characteristic "Markus smirk" when he entered our office because he needed something.
My sincere thanks go to Günter Quast, who has accompanied me since my second semester. During my first appointment committee, which, funnily, was the committee that selected Torben's predecessor, he kindly looked after freshman-year me, who was rather intimidated by all the professors in the room. I still remember how he paid for my pizza at dinner with one of the candidates when I realized too late that I did not have cash with me. Later, he supervised my bachelor's thesis, which I greatly enjoyed. He has always made me feel that he fully respects me and my work and that he is proud of my achievements. I am very grateful for the endless discussions we had, especially about committee work at KIT, from which I learned a great deal, and for his continuous support in all the committees we shared.
I also wish to thank Margarete Milada Mühlleitner, whom I first met in the same appointment committee as Günter. I still remember how she complimented my clothing style back then, which immediately gave me a good feeling about her. Years later, I very much enjoyed working with her on the organization of the German Conference for Women in Physics in Karlsruhe, including our late-night meetings that revealed that we are both workaholics. I am grateful that she agreed to act as my formal mentor during my PhD.
In particular, I would like to thank Giacomo De Pietro for his tremendous support during the last years of my PhD. As the Belle II software coordinator, he explained everything

I needed to know about the Belle II software and helped me understand it on a much deeper level than I could have reached on my own. I also thank him for the Pokémon plush toys he brought back from business trips to Japan, which connected in a very cute and thoughtful way to the Pokémon analogies I used to describe parts of my PhD work.
My heartfelt thanks go to Isabel Haide, whom I have known since the first week of my physics studies, when she was one of my mentors during the introduction week. In the same semester, we started Taekwondo together. She was my office neighbor for almost the entire duration of my master's thesis and PhD, and I could always come to her with all kinds of questions. I am very grateful for her kindness and continuous support.
I gratefully acknowledge Marc Neu, who carried out the extensive implementation of the algorithm on FPGA. Working with him was a great pleasure. He is always impressively well organized and extremely smart, and I deeply appreciate his contribution to this work.
Furthermore, I would like to thank all my other colleagues for their collaboration, professional support, and the wonderful times we spent together, including joint holidays and proof-reading of this thesis: Lea Reuter, Alexander Heidelbach, Jonas Eppelt, Slavomira Stefkova, Patrick Eckert, Lennard Damer, Marc-Philipp Thomas, Raynette van Tonder, Priyanka Cheema, Matthias Schnepf, and Pablo Goldenzweig. I also thank my colleagues from the CMS group, namely Robin Hofsaess, Jan Hauke Voss, Cedric Verstege, Lars Sowa, Nils Faltermann, Jost von den Driesch, and Christian Winter, for the wonderful time together.
I would like to thank Taichiro Koga, Christian Kiesling, Simon Hiesl and Kai Unger for providing me with all the information I needed about the Belle II Level-1 trigger system and for patiently answering the many questions I had about it.
My warmest thanks go to all my friends, and in particular to Rebecca, who has been one of my best friends since our first semester. Despite the hundreds of kilometers between us, our friendship is as strong as ever and I am proud to have her in my life. I also want to thank Juli and Jessi, with whom I shared countless lunch and coffee breaks that provided essential emotional support during some of my most stressful times of the PhD. Finally, I would like to thank Simon, who not only read my entire master's thesis and provided incredibly helpful comments, but also carefully proof-read parts of this PhD thesis.
I owe many thanks to the Fachschaft Physik, where I spent many wonderful years. The work and community there allowed me to grow in many ways and I have made countless memories that I will cherish for the rest of my life.
I am deeply grateful to my family and, in particular, to my father, who sparked my interest in natural sciences and physics. They have always supported me throughout my studies and this PhD and have made it possible for me to study in the first place.
Finally, I would like to thank my wonderful boyfriend Alex, who was always there for me and who took over most of the cooking during the stressful final weeks of writing this thesis, making it much easier for me to focus.
I gratefully acknowledge the financial support of KSETA, which funded this PhD project and provided an excellent research and training environment

Contents

1. Introduction

Particle physics experiments, such as the Belle II experiment [6] at the SuperKEKB electron-positron collider, provide precision tests of the Standard Model (SM) and search for physics phenomena that cannot be explained within the SM framework. These experiments rely on the precise reconstruction of charged particle trajectories (tracking) to accurately determine particle momenta, identify particles, and reconstruct the kinematics of the underlying physics processes. To achieve high sensitivity to rare processes, Belle II aims to record an integrated luminosity of $\int \mathcal{L}\,\mathrm{d}t = 50\ \mathrm{ab}^{-1}$. As SuperKEKB approaches its target instantaneous luminosity of $6 \cdot 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, beam-induced backgrounds increase, resulting in higher detector occupancy and accelerated detector aging. These increasing background rates impose stringent constraints on the Belle II event reconstruction for analyzes (offline) and trigger algorithms (online), as they directly degrade track reconstruction efficiency, purity, and resolution [7–9]. Consequently, the anticipated high-background operating conditions necessitate the implementation of efficient hit-filtering prior to track reconstruction in both offline and online event reconstruction, *i.e.* the reduction of hits not associated with charged particle trajectories.

This thesis presents the development and implementation of a graph neural network (GNN)-based hit filtering algorithm for the main tracking detector of Belle II, the central drift chamber (CDC). The proposed GNN-based hit filter takes advantage of both the per-hit features and the spatial-temporal correlations in the hit topologies to distinguish hits originating from charged particle trajectories from those arising from background processes or electronic noise. The algorithm is designed for use in offline and high-level trigger (HLT) track reconstruction that are processed on CPU, as well as in the Level-1 trigger (L1 trigger) tracking system operating on FPGAs. Integrated into the Belle II Analysis Software Framework (basf2) framework for offline reconstruction, the GNN-based filter functions as a drop-in replacement for existing hit filtering algorithms. For the L1 trigger application, the algorithm is further developed under the strict L1 trigger constraints including sub-microsecond latency, and a total L1 trigger rate of less than 30 kHz, targeting deployment on FPGA hardware for future integration within the Belle II L1 trigger.

The structure of this thesis is organized as follows. It begins with an overview of related work on machine learning (ML)-based tracking and trigger algorithms in Chapter 2. Subsequently, SuperKEKB and Belle II are introduced, with particular emphasis on the CDC, the trigger system, beam-induced backgrounds, and planned upgrades, in order to establish the experimental context of this work. A more detailed description of the offline and online track reconstruction algorithms is provided in Chapter 4, thereby defining the environment in which the new algorithm operates. In Chapter 5, the fundamental con-

cepts of the GNN-based hit-filtering approach are presented, including graph construction, network architecture, and training methodology. Chapter 6 and Chapter 7 specify the datasets and performance metrics employed for training and evaluation.
In Chapter 8, the optimization of the offline application is discussed and its performance is compared to established baseline methods, while Chapter 9 describes the development of a compressed, FPGA-compatible online implementation, its integration into the simulated L1 trigger pipeline, and studies of its prospective hardware realization, followed by a short outlook in Chapter 10. Finally, Chapter 11 summarizes and concludes the main results.

2. Related Work

Machine learning (ML) has become a key tool in high-energy physics analyzes, improving event reconstruction, classification, and anomaly detection [10–16]. For charged-particle tracking in particular, ML-based methods are widely used [3, 17–21] where a key challenge is to identify and separate an a priori unknown number of signal hits and tracks from a large background. This task is analogous to object detection in computer vision, where the goal is to localize and classify multiple objects in an image. Such conventional object detection tasks are mainly based on convolutional neural networks (CNNs) or transformer-based architectures tailored to grid-structured Euclidean data such as images or sequences [22, 23].

The Belle II CDC, by contrast, has a non-uniform cylindrical geometry with irregular wire positions and alternating axial and stereo super-layers. This non-Euclidean layout, combined with intrinsically sparse hits (only a subset of wires fire in a given event), makes standard CNN-based methods ill suited for CDC hit filtering and track reconstruction. A more natural description is graph-based, where nodes represent CDC hits and edges encode spatial and temporal relations.

GNNs for such graph-structured data have been successfully applied in many areas, including materials science, chemistry, medicine, and especially high-energy physics (HEP) [24–26]. In high-energy physics they have been extensively studied for tracking and related reconstruction tasks [3, 15, 17–21]. Through message passing along graph edges, GNNs learn complex structural and relational patterns in event topologies. This makes them also well suited for background suppression, as they jointly exploit intrinsic hit features and spatial-temporal correlations in local neighborhoods. In addition, recent work on the deployment of GNNs for track reconstruction on FPGAs [27–29] and ultra-fast architectures for jet tagging [30] has demonstrated the feasibility of real-time GNN inference in high-rate collider environments.

While these works primarily focus on track-level reconstruction or jet tagging leveraging higher-level objects, the approach discussed in this work is distinguished by its focus on hit-level filtering directly on CDC sense wire data. This enables real-time detector-level background suppression, achieving sub-microsecond latencies prior to track reconstruction in the Belle II L1 trigger.

In addition, the L1 trigger implementation of the GNN-based hit filter is designed to operate in a substantially different regime from that considered in previous studies: it is designed to process graphs comprising up to $O(10^3)$ nodes per CDC sector at a sustained input rate of 32 million events per second, while maintaining an end-to-end latency below one microsecond. To my knowledge, this specific combination of high throughput, low latency, and large input graph size has not been demonstrated in previously published work.

3. SuperKEKB and the Belle II Experiment

Belle II is an e^+e^- experiment, located at the high-luminosity SuperKEKB accelerator, designed for precision studies of heavy flavor physics and dark sector searches [31, 32]. In this chapter, I provide an overview of the Belle II experiment and the SuperKEKB accelerator. Particular emphasis is placed on the central drift chamber (CDC) and the Level-1 trigger (L1 trigger) system, with a detailed discussion of the beam-induced background and the resulting detector degradation. This serves to establish the experimental context relevant for the subsequent, more detailed description of the Belle II tracking and trigger subsystems.

3.1. The Belle II physics program

The primary objective of Belle II is the detailed study of B-meson decays produced at the $\Upsilon(4S)$ resonance that almost exclusively decays to a pair of $B\bar{B}$ mesons $\mathcal{B}(\Upsilon(4S) \to B\bar{B}) > 96\,\%$ [33]. These decays provide stringent tests of the Cabibbo–Kobayashi–Maskawa (CKM) mechanism, enabling sensitive probes of new physics through rare decay modes and precise CP violation measurements.

Operating at an e^+e^- collider, Belle II benefits from clean collision events characterized by a well-defined initial state and negligible pileup. In contrast, proton colliders are characterized by partonic interactions with unknown initial states and by the presence of multi-parton scattering processes, both of which significantly complicate event reconstruction. The precisely known collision center-of-mass energy, together with the fact that the $\Upsilon(4S)$ decays almost exclusively to a pair of $B\bar{B}$, allows full reconstruction of final states, including those with missing energy from neutrinos or other invisible particles. In particular, $b \to s$ transitions such as $b \to s\nu\bar{\nu}$, measured, for example, via $B^+ \to K^+\nu\bar{\nu}$ [34], constitute a key flavour changing neutral current (FCNC) process as a probe for potential new physics contributions. Since these processes are predicted to be strongly suppressed within the Standard Model (SM), any significant observation of these processes can be interpreted as a signal beyond the SM.

Another probe of new physics is the search for charged Higgs bosons in flavor transitions to τ leptons, including $B \to \tau\nu$ and $B \to D^{(*)}\tau\nu$. Furthermore, τ leptons are investigated in the context of CP violation as well as lepton flavour violation (LFV) searches in rare decay channels such as $\tau \to \mu\gamma$, and further studied for high-precision measurements of the electric dipole moment and the anomalous magnetic moment (g-2) of the τ lepton.

Many extensions of the SM predict weakly coupled light particles that could constitute a dark sector, such as dark photons, or axion-like particless (ALPs). At Belle II these processes are studied in low-multiplicity final states, signatures with missing energy, and

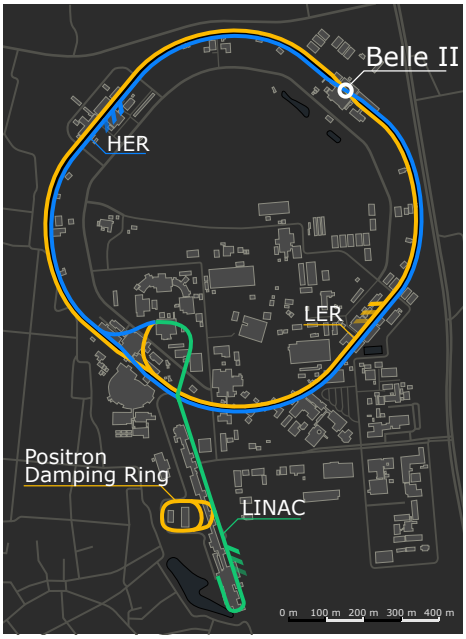


Figure 3.1.: Geographic layout of the SuperKEKB accelerator complex at KEK in Tsukuba, Japan. The high-energy electron ring (HER, indicated in blue) and low-energy positron ring (LER, indicated in orange) are indicated, together with the linear accelerator (LINAC, indicated in green) and the positron damping ring. The location of the Belle II experiment detector at the interaction point is marked. The image was designed by T. Blesgen.

displaced decays [35]. Other searches focus on exotic hadrons that cannot be explained within the conventional quark model [36].
In addition to operating at the $\Upsilon(4S)$ resonance, Belle II records large samples of continuum events $e^+e^- \to \ell^+\ell^-$ ($\ell = e, \mu, \tau$), such as $e^+e^- \to \mu^+\mu^-(\gamma)$, as well as light-quark pair production $e^+e^- \to q\bar{q}$. These processes serve as both essential calibration channels and as precision probes of electroweak and quantum electrodynamics (QED) [32]. Moreover, Belle II performs off-resonance scans below and above $\Upsilon(4S)$, *e.g.* to study the continuum processes and to search for new exotic hadron states.

3.2. The SuperKEKB accelerator

SuperKEKB [31], shown in Figure 3.1, is an asymmetric-energy e^+e^- collider located at KEK in Tsukuba, Japan. Electrons and positrons are first accelerated to their target energy by an injector linear accelerator (LINAC). For the reduction of emittance, *i.e.* , to decrease the phase-space area occupied by the beam in the transverse position-momentum plane, the positrons are additionally transported through a damping ring (DR). Subsequently, the accelerated electrons and positrons are injected into two independent storage rings: the 7 GeV high energy ring (HER) for electrons and the 4 GeV low energy ring (LER) for positrons that intersect at a single interaction region, where the Belle II detector is installed.
The accelerator is operated predominantly at a center-of-mass energy of about 10.58 GeV, corresponding to the $\Upsilon(4S)$ resonance, which is just above the B-meson pair-production threshold. The asymmetric beam energies boost the center-of-mass frame along the beam axis. This results in a measurable spatial separation between the decay vertices of the two B-mesons, enabling time-dependent CP violation measurements.
SuperKEKB is designed to reach a peak instantaneous luminosity of the order $\mathcal{L} =$

$6 \cdot 10^{35}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [7], approximately 30-40 times higher than that of its predecessor KEKB, which achieved $\mathcal{L} = 2.1 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ [37] that constitutes the collider luminosity world record at that time. SuperKEKB has since exceeded this record and has achieved an instantaneous luminosity of about $\mathcal{L} = 5.244 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ (March 2026) [38] at the time of this thesis, more than twice the KEKB record.
The long-term objective of the SuperKEKB and Belle II programs is to accumulate an integrated luminosity of order $\int \mathcal{L}\mathrm{d}t = 50\,\mathrm{ab}^{-1}$. For comparison, the Belle experiment at KEKB collected about $1\,\mathrm{ab}^{-1}$ [39]. This substantial increase in the amount of data will enable significantly improved sensitivity to rare decays, more precise determinations of CKM matrix elements, and more stringent tests of the Standard Model in the flavor sector.

The luminosity [7] can be expressed as

$$L = \frac{N_+ N_- n_b f_0}{2\pi \phi_x \Sigma_z \Sigma_y^*} \propto \frac{I_\pm \xi_{y\pm}}{\beta_y^*}. \tag{3.1}$$

Here, ± indicates positrons (+) or electrons (-), N is the number of particles per bunch, n_b the number of bunches, f_0 the revolution frequency, ϕ_x the half crossing angle, and Σ_z and Σ_y^* are the longitudinal and vertical beam sizes at the interaction point (IP), respectively. The quantities $I_\pm = e^\pm N_\pm n_b f_0$ are the beam currents in the LER and HER, $\xi_{y\pm}$ are the dimensionless vertical beam-beam parameters, and β_y^* is the vertical beta function at the IP.
Achieving the desired luminosity requires increasing the beam currents $I_\pm$, enhancing the vertical beam-beam parameters $\xi_{y\pm}$, and reducing the vertical beta functions β_y^*. Since increasing beam currents by more than a factor of two is highly challenging in terms of technical feasibility and operational costs, SuperKEKB adopts the so-called nano-beam collision scheme [40]. In the nano-beam scheme, beam bunches with small beam sizes collide at a relatively large horizontal crossing angle, leading to a reduction of β_y^* by roughly a factor of 20 compared to the value of KEKB [7] without the necessity to increase beam currents. Luminosity can be further enhanced by increasing the beam-beam parameter $\xi_{y\pm}$. This also strengthens beam-beam interactions and may cause instabilities that must be carefully controlled during operation. To approach the design luminosity, ongoing and planned upgrades and optimization measures are under investigation, including modifications of the interaction region, improvements to the injection complex, and continued refinement of collimation systems and vacuum conditions [7]. However, increasing the luminosity comes at the price of increased beam-induced backgrounds in the Belle II detector, which will be discussed in section 3.6.

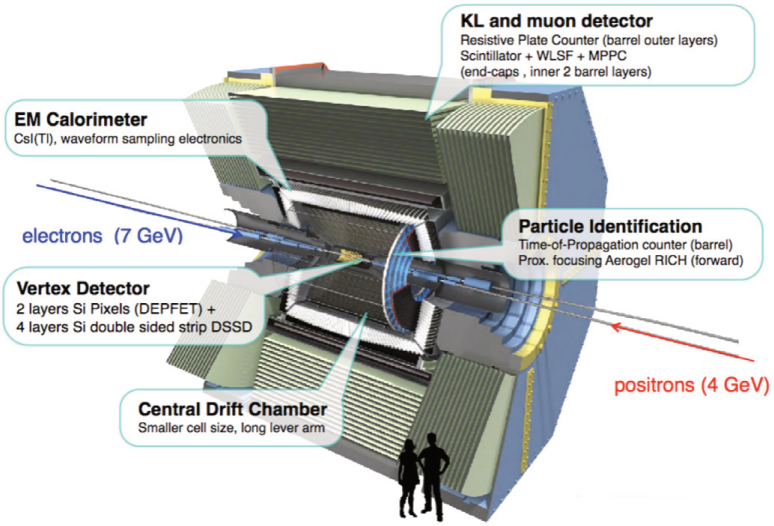


Figure 3.2.: Overview of the Belle II detector including its sub-detectors. Taken from [41].

3.3. The Belle II detector

The Belle II detector [6] is a general-purpose, nearly hermetic 4π spectrometer installed at the IP of SuperKEKB, optimized for time-dependent CP violation in B-meson decays and precision measurements. In order to achieve these goals, Belle II [42] is designed with

- an impact-parameter resolution of 10 to 15 µm, resulting in 20 to 30 µm vertex resolution;
- a relative charged-particle transverse momentum resolution of approximately 0.3 % at p_T=1 GeV/c;
- an observed hadron identification efficiency of 90 % at 10 % contamination with uncertainties of the order of 1 %;
- energy-dependent resolutions of electrons and photons in the 1.6 to 4 % range;
- an observed lepton-identification performance of 0.5 % pion contamination at 90 % electron efficiency, and 7 % kaon contamination at 90 % muon efficiency with typical uncertainties at the 1 to 2 % level.
- a Level-1 trigger system [8] that delivers nearly 100 % efficiency for $B\bar{B}$ events at $<$ 30 kHz rate with $<$ 4.4 µs latency.

Belle II has a cylindrical geometry concentric with the beam pipe, as shown in Figure 3.2, divided into three regions: backward, barrel, and forward. Due to the asymmetric beam energies of the electron and positron beams, the IP is not located at the geometric center of the detector, but instead is displaced toward the backward region, thus enlarging the forward region relative to the backward region. The coordinate system is defined with its

origin at the IP: the z-axis follows the incoming electron beam, the y-axis points vertically upward, and the x-axis completes a right-handed system. The detector acceptance is asymmetric as well, covering polar angles θ from 12.4 to 155.1°, reflecting the boost from the asymmetric-energy collisions, where the polar angle θ is defined between $[0, \pi]$, with $\theta = 0$ parallel to the positive z-axis. The azimuthal angle ϕ is defined between $[-\pi, \pi]$, with $\phi = 0$ along the positive x-axis at $y = 0$.
The Belle II detector consists of several concentric subsystems, surrounded by a 1.5 T superconducting solenoid and iron yoke, that provide tracking, vertexing, calorimetry, and particle identification:

- Closest to the IP is a two-layer **silicon-pixel detector (PXD)** [43], which provides highly granular spatial measurements of the charged particle tracks near the beam pipe. With about eight million pixels, it provides the high spatial resolution required to reconstruct decay vertices of short-lived particles such as B-mesons.

- The **silicon-strip vertex detector (SVD)** [44] surrounds the PXD, and consists of four double-sided layers of silicon strip detectors. The SVD extends the tracking volume and improves both vertex and momentum resolution, especially for low-momentum tracks.

- Outside of the silicon detectors, the **CDC** [45] provides tracking and $\mathrm{d}E/\mathrm{d}x$ energy-loss measurements for charged particles. A detailed description of the CDC is given in section 3.4. Information from the pixel, silicon, and drift detectors is combined to reconstruct particle trajectories and determine their momenta..

- Beyond the tracking system, two dedicated particle-identification systems cover complementary angular regions: the **time-of-propagation counter (TOP)** [46] in the barrel and the **aerogel-based ring-imaging Cherenkov counter (ARICH)** [47] in the forward end-cap. Both rely on Cherenkov light detection and ring-imaging techniques to distinguish hadron species, in particular kaons and pions, over a wide momentum range.

- Surrounding these subsystems is the **electromagnetic calorimeter (ECL)** [48], composed of thallium-doped CsI crystals arranged in a barrel and two end-caps. It measures the energies and times of photons and electrons, contributing to both neutral-particle reconstruction and particle identification through shower-shape analyses.

- The outermost system is the **K-long and muon detector (KLM)** [49], embedded in the iron yoke that returns the magnetic flux of the solenoid. The KLM employs resistive-plate chambers and plastic scintillator modules to identify muons and detect neutral K_L^0-mesons and neutrons escaping the inner detectors.

Together, these subsystems enable high-precision track reconstruction, vertex determination, particle identification, and energy measurement across a wide kinematic phase space.

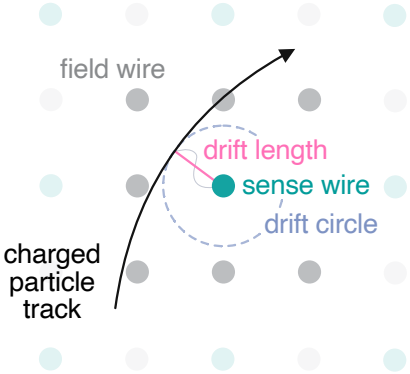


Figure 3.3.: Schematic view of a drift cell where a charged particle (black) passes a drift cell bound by the field wires (grey), including the drift path of the electrons to the sense wire (green) and the resulting drift length (pink) and drift circle (blue).

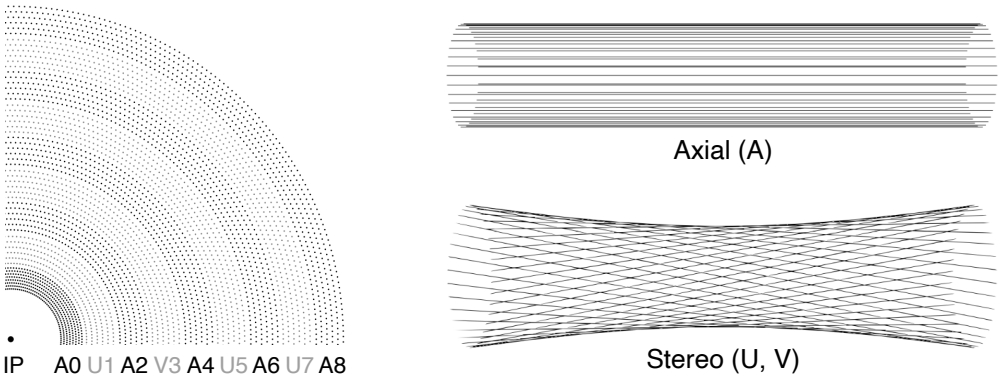
3.4. The central drift chamber (CDC)

The CDC [6, 50] is the main tracking detector of Belle II and the main sub-detector studied in this work. It provides the trajectories of charged particles and momentum information, as well as particle identification through energy loss measurement $\mathrm{d}E/\mathrm{d}x$. The latter is particularly important for low-momentum tracks that do not reach the outer particle-identification systems. Furthermore, it is one of the trigger sub-detector systems providing trigger signals for charged particles.

The 233 cm long cylindrical wire chamber spans radii from 16 to 113 cm and is organized into three distinct regions: the forward region ($17° < \theta < 35.4°$), the barrel region ($35.4° < \theta < 123°$) and the end-cap region ($123° < \theta < 150°$) which cover the full azimuthal range $0 < \phi < 2\pi$. It is filled with a 50:50 helium-ethane gas mixture, resulting in an average electron drift velocity of 3.3 cm/µs.

The drift chamber comprises 14 336 gold-plated tungsten sense wires arranged in nine concentric rings, so-called super-layers. The sense wires and additional 42 240 aluminum field wires form drift cells of approximately $10 \times 8\,\mathrm{mm}^2$ and 18×18 mm for the inner-most and all other super-layers, respectively, as illustrated in Figure 3.3. In the inner-most super-layer, a reduced cell size is implemented to mitigate channel occupancy arising from high beam-induced background.

Charged particles traverse the CDC along approximately helical trajectories in the solenoidal magnetic field and ionize the gas along their paths, producing electrons and ions in each drift cell. In the electric field, the electrons drift toward the positively charged sense wire. Close to the sense wire, the field strength increases and initiates an avalanche multiplication process. Secondary ionization amplifies the initial charge and generates a signal that is read out on the end-plates of the wire by front-end electronics (FEEs) mounted on the backward side of the CDC. The registered signal on the wire is digitized and converted into a drift length via calibrated time-to-space relations with a spatial resolution of approximately 120 µm in the transverse plane. Since the azimuthal position of the particle within the drift cell is a priori unknown, the measurement is expressed as a drift radius around the sense wire.



(a) Super-layer configuration (b) Axial and stereo super-layers

Figure 3.4.: **(a)** The 14 336 sense wires of the CDC are arranged concentrically around the interaction point (IP) in 56 layers, which are grouped into 9 super-layers. These wires are oriented either axially (denoted A) or in a stereo configuration (denoted U and V). **(b)** The axial wires are aligned parallel to the beam axis, whereas the stereo wires are inclined with respect to the beam axis. The stereo angle alternates between odd- and even-numbered stereo super-layers. The figures are adapted from [45].

The sense wires, depicted in Figure 3.4, are arranged in 56 layers grouped into 9 super-layers and are oriented either axial (denoted as A) or stereo (denoted as U and V). Axial super-layers comprise wires that are aligned parallel to the beam axis in the z-direction and therefore enable measurements in the transverse $(r - \phi)$ plane.

As the sense wires are read out only at a single end-plate, no direct z-position information can be obtained from hits in the axial layers. Therefore, the wires in the stereo super-layers are inclined by small stereo angles, as specified in Table 3.1. This inclination induces a z-dependent displacement of the hit positions in the transverse view, providing an additional spatial degree of freedom.

Table 3.1.: Configuration of the CDC sensor wires, taken from [6].

Super-layer	N_{layer}	$N_{\text{sensor cells}}$	r (cm)	Stereo angle (mrad)
Axial A0	8	160	16.80 to 23.80	0.0
Stereo U1	6	160	25.70 to 34.80	45.4 to 45.8
Axial A2	6	192	36.52 to 45.57	0.0
Stereo V3	6	224	47.69 to 56.69	−55.3 to −64.3
Axial A4	6	256	58.41 to 67.49	0.0
Stereo U5	6	288	69.53 to 78.53	63.1 to 70.0
Axial A6	6	320	80.25 to 89.25	0.0
Stereo V7	6	352	91.37 to 100.37	−68.5 to −74.0
Axial A8	6	384	102.09 to 111.14	0.0

By combining hits from axial and stereo layers, the reconstruction algorithms perform a helical fit to the resulting set of three-dimensional space points and thus determine the full three-dimensional trajectory and momentum of the charged particle, which will be described in detail in section 4.1.

3.5. The Belle II online system

The Belle II online system includes the real-time data acquisition and trigger infrastructure responsible for the readout of detector signals, the execution of low-latency event selection, and the management of data flow during experimental operation.

At SuperKEKB the bunch-crossing rate is approximately 250 MHz, whereas the collision rate and the resulting rate of physics processes is much lower. For example, the production rate of the $\Upsilon(4S)$ resonance at the design luminosity is only about 0.67 kHz [32]. The most common background interaction is Bhabha scattering, $e^+e^- \rightarrow e^+e^-(\gamma)$, in which the scattered electron or positron falls within the acceptance of the ECL, occurring at a rate of roughly 44.6 kHz. The remaining interactions are predominantly caused by beam-induced background processes, described in section 3.6. Consequently, a substantial fraction of the recorded events must be rejected.

The Belle II online system is therefore composed of multiple stages: The data acquisition (DAQ) acquires data from the sub-detectors and, upon receipt of the L1 trigger [8, 51] signal, forwards the corresponding event information to the high-level trigger (HLT). It is limited to an event recording rate of less than 30 kHz due to limited bandwidth. Therefore, the L1 trigger system reduces the event rate online (*i.e.* in real-time before events are stored) to a level that can be processed by the DAQ system, while preserving a high efficiency for the physics processes of interest [32]. The HLT [52] executed on a computing farm then performs additional event selection and data reduction to decrease the overall data volume to a level compatible with the available storage capacity on disk. In the following, the components of the Belle II online system will be discussed in detail.

3.5.1. Data acquisition system

The DAQ system is designed to acquire the continuous data stream from all sub-detectors. For example, in the CDC, raw detector signals are initially processed and digitized locally on the 299 FEE boards mounted directly on the backward side of the detector, each processing 48 sense wire channels [53]. Each of these channels is processed by an amplifier-shaper-discriminator ASIC [54], which provides an analog signal for charge measurements.

The analog signal waveform as illustrated in Figure 3.5 is digitized by an analog-to-digital converter (ADC) and a time-to-digital Converter (TDC) that provide a ADC count, a TDC time stamp and a time-over-threshold (TOT) measurement. When a sampled ADC value exceeds the channel-specific pedestal value, the corresponding TDC time stamp is recorded and the 25 consecutive sampling points (taken every 32 ns) accumulate to form an ADC count. In addition, the TOT value, which is the number of sample points above

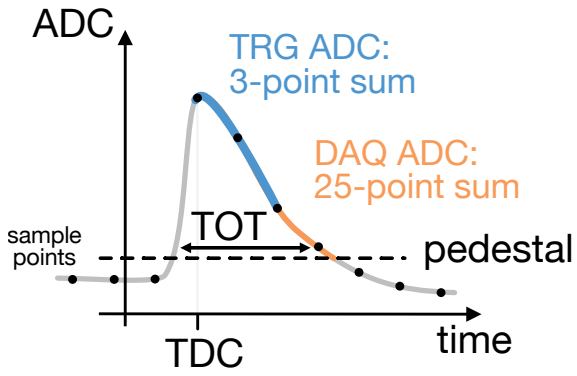


Figure 3.5.: Sense-wire pulse shape of a CDC wire hit. The first sample exceeding the pedestal triggers the TDC signal. The next three consecutive samples are sent to the L1 trigger (TRG), while up to 25 samples are summed above pedestal and sent to the DAQ system. For most hits, far fewer than 25 samples contribute to this DAQ ADC sum. The complementary TOT signal records how many samples have amplitudes above the pedestal.

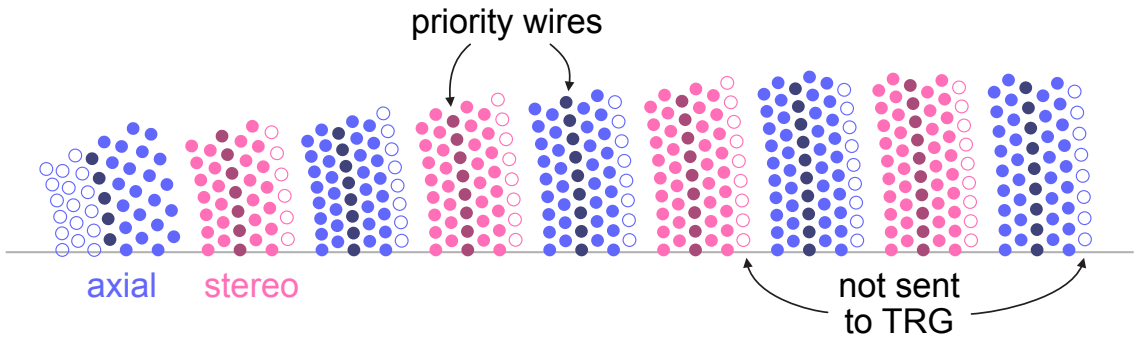


Figure 3.6.: CDC wires are arranged in alternating axial (purple) and stereo (pink) super-layers. Hits on the first three layers and the last layer in every super-layer except for the first super-layer are not sent to the L1 TRG system (white). In the CDC TRG system, one out of five layers is considered a priority layer (darker color).

the pedestal value, provides complementary timing and charge information. The measured TDC value can be decomposed as

$$\alpha \cdot \mathrm{TDC} = T_0 - T_{\mathrm{tof}} - T_{\mathrm{drift}} - T_{\mathrm{prop}} \tag{3.2}$$

with a resolution of α =1 ns. T_0 is the collision time, T_{tof} the time-of-flight from the interaction point to the drift cell, T_{drift} the electron drift time to the sense wire, and T_{prop} the signal propagation time along the wire to the FEEs. The dominant contribution is the drift time of electrons within the drift cell T_{drift}, which can reach values greater than 600 ns, while the other terms are typically of the order $\mathcal{O}(1\,\mathrm{ns})$.
The TDC value of each wire is mainly used to determine the drift radius of the corresponding hit, and the complete set of TDC values in an event allows calculation of the global event time T_0. In an alternative scenario where $T_{\mathrm{drift}} \ll T_{\mathrm{prop}}$, the TDC measurement could in principle provide z-dependent information, since the signal is read out from only one wire end. Alternatively, equipping both wire ends with FEE readout would allow direct reconstruction of the longitudinal hit position, but this is not possible with the current detector setup.

Table 3.2.: Maximum resulting L1 trigger latency constraint imposed by the different sub-detector systems due to the buffer of the detector front-ends. The overall latency is limited by the lowest of these budgets. The values are taken from [55].

Sub-detector	max. latency budget (μs)
PXD	$\approx 10\,\mu$s
SVD	$5\,\mu$s
CDC	$15\,\mu$s
TOP	$9\,\mu$s
ARICH	up to $1015\,\mu$s
ECL	$100\,\mu$s
KLM	$5.2\,\mu$s

In parallel to the primary DAQ readout chain, a reduced-precision copy of the raw detector signal on independent FEE boards is routed to the L1 trigger system: a coarser time resolution of 2 ns is used, the ADC value is calculated as the sum of 3 sampling points and no TOT information is provided.
In addition, due to bandwidth limitations only a subset of the available CDC layers is transmitted to the downstream L1 trigger reconstruction algorithms, as illustrated in Figure 3.6: in the inner-most super-layer, the first three layers are excluded, and in each of the remaining super-layers the outermost layer is omitted.

3.5.2. The Level-1 trigger system

The L1 trigger receives reduced-granularity information from several sub-detector-specific trigger systems and provides a global accept or reject decision for each event.
The overall system must comply with the following design constraints:

- A maximum L1 trigger rate of 30 kHz, consistent with the DAQ throughput capacity of 3 GB/s.
- A total trigger latency <4.4 µs, including data transmission and processing across all L1 trigger modules.
- Event timing precision of less than 10 ns and event separation of 500 ns.
- Near 100 % efficiency for hadronic events, including $e^+e^- \to B\bar{B}$ and $e^+e^- \to q\bar{q}$ with $q = u, d, s, c$.

The latency constraint is determined by the buffer size of the FEEs, which buffer data for roughly one beam bunch revolution before a L1 trigger decision must be issued [51]. Table 3.2 presents an approximate overview of the respective maximum acceptable latency constraints applied by each of the various sub-detectors. Currently, the overall L1 trigger latency is constrained by the SVD with a latency budget of 5 µs.

Four main sub-trigger systems provide input to the global trigger logic of the L1 trigger system as displayed in Figure 3.7:

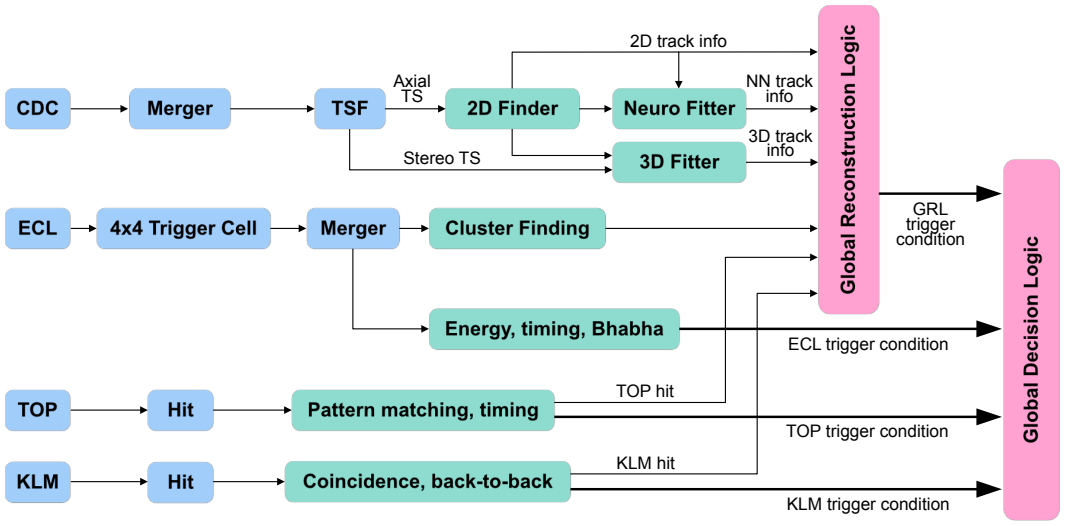


Figure 3.7.: Schematic diagram of the Belle II Level-1 trigger system, illustrating its key components and data flow. Adapted from [8].

- The **CDC** trigger [8, 56] is one of the primary sub-triggers reconstructing two- and three-dimensional charged tracks from CDC hit patterns. It provides track multiplicities, approximate angle, z and momentum information over the entire Belle II acceptance. A detailed description of this sub-trigger is given in section 4.2.
- The **ECL** trigger [57] operates on trigger cells that aggregate signals from multiple crystals to 4x4 trigger cells. It provides measurements of the total deposited energy, individual cluster energies, and cluster positions. Furthermore, it performs Bhabha scattering identification by exploiting the characteristic back-to-back cluster topology. Bhabha identification is crucial in the L1 trigger as Bhabha scattering constitutes the highest background source and needs to be filtered at an early stage.
- The **TOP** trigger [58] provides event timing and hit-topology information derived from Cherenkov photon propagation times in the TOP detector. This is achieved by correlating the observed photon arrival time with precomputed reference patterns obtained from simulation.
- The **KLM** trigger [51] provides coincidence and back-to-back condition information.

The outputs of the sub-trigger systems, such as charged tracks from the CDC, ECL clusters, and locations of the TOP and KLM hits, are collected by the global reconstruction logic (GRL) that combines the information from the sub-trigger systems and provides global trigger observables to the final global decision logic (GDL). The GDL subsequently derives the final trigger decision by evaluating Boolean combinations of the various trigger signals supplied directly by the sub-detector systems as well as by the GRL.

The L1 trigger system is fully implemented on Field-Programmable Gate Array (FPGA)-based boards. It processes events in a dead-time-free pipeline synchronized with a 127.216 MHz system clock distributed by the trigger and timing distribution system. All L1 trigger modules, except for the front-end and merger devices, are executed on UT3 or UT4 boards that feature a Xilinx Virtex-6 or a Xilinx Ultrascale XCVU080/160 chip, respectively [8]. The system operates with a common revolution signal of 10 µs, which is used to synchronize counters and data pipelines throughout the trigger chain.

3.5.3. The high-level trigger system

The HLT system [52, 59] consists of a computing cluster comprising up to 15 computing nodes that house around 5 000 processing cores. It performs a full event reconstruction applying the Belle II Analysis Software Framework (basf2) to the raw data acquired from the sub-detector FEEs in real time. It should be noted that the PXD data are not sent to the HLT. Instead, the HLT processes data from the other sub-detectors and, for selected events, transmits the corresponding region-of-interest information to the PXD readout. The PXD data are then combined with the rest of the Belle II event data downstream of the HLT.
The objective of the HLT is to achieve a data reduction by a factor between 1/3 and 1/5, such that the output rate remains below 1 GB/s. Based on this online reconstruction, events are selectively filtered before being written to persistent storage for subsequent offline processing and analysis.
In addition to filter-based selection, the HLT performs an initial coarse classification (skimming) that produces the so-called HLT skims. All events that satisfy at least one HLT condition are subjected to a fast analysis and assigned to one or more broad physics categories, such as Bhabha, hadronic, tau, muon-pair, and others. A corresponding flag is attached to each event, which can later be accessed directly from the raw data without requiring a full event reconstruction.

3.5.4. Field programmable gate arrays

The L1 trigger system relies primarily on FPGA technology due to strict throughput and latency requirements. The main advantage of FPGAs over general-purpose processors such as central processing units (CPUs) and graphics processing units (GPUs) is high parallelization with pipelining: Independent operations can run in parallel, allowing trillions of operations per second with relatively low power consumption. Comparable throughput can only be achieved by ASICs that are custom-made for their specific tasks. In contrast to ASICs, FPGAs support low-cost flexible prototyping with much shorter development cycles than ASICs. Furthermore, a key advantage of FPGAs is their post-manufacturing reconfigurability, which is especially useful in machine learning (ML) applications, where the neural network architecture and parameters can be dynamically updated during operation.

A conventional FPGA architecture, illustrated in Figure 3.8, consists of thousands of configurable logic blocks (CLBs). These CLBs typically contain a programmable look-up table

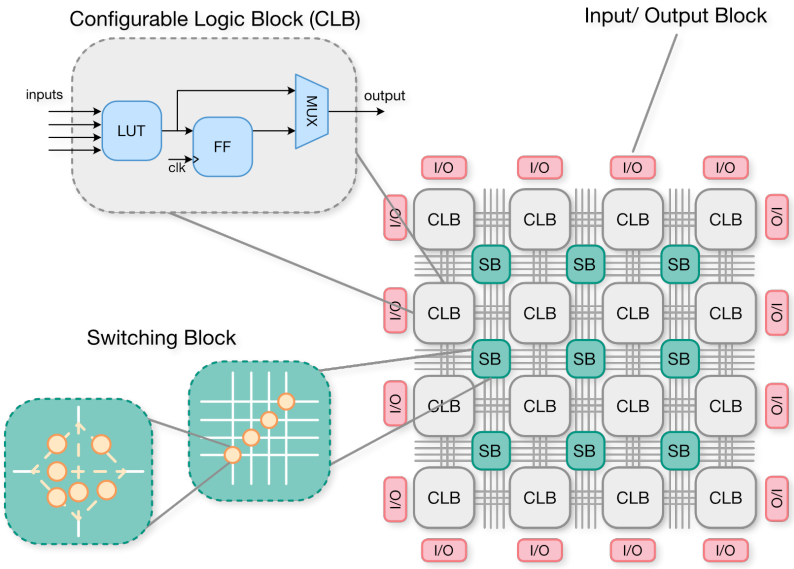


Figure 3.8.: Simplified illustration of an FPGA consisting of input/output (I/O) Blocks, switching block (SB) and configurable logic block (CLB) consisting of a look-up table (LUT), a flip-flop (FF), and a multiplexer (MUX).

(LUT) that performs logical or arithmetic operations via table look-up, a flip-flop (FF) that forwards the signal at the subsequent positive clock (clk) edge, and a multiplexer (MUX) that enables either the forwarding or the bypass of the corresponding signal. The CLBs are interconnected through programmable switching blocks (SBs), while input/output (I/O) blocks provide the interface to external signals.
Since implementing multipliers exclusively with CLBs is highly resource-intensive, modern FPGAs incorporate dedicated digital signal processors (DSPs), which are specialized for low-latency arithmetic operations and offer a good trade-off between performance and resource utilization. Together with FFs, LUTs and DSPs, on-chip block random access memorys (BRAMs) constitute the four main components of an FPGA and are used to store larger data sets directly on the device.

Today, the term FPGA is sometimes used interchangeably with system-on-chips (SoCs), which integrate multiple embedded subsystems, including CPUs, bus interconnects, RAM, ROM, and various peripherals, on a single device. Within the Belle II L1 trigger infrastructure, the UT3 and UT4 boards are deployed, both of which are based on the Xilinx Zynq Ultrascale+ SoC.

3.6. Beam-induced backgrounds at Belle II

The Belle II detector is subject to significant backgrounds [6, 7, 32], where any process that is not part of the decay of interest is considered background. In the following, the general categories of background sources at Belle II are described, followed by a detailed discussion of the beam-induced background contributions.

3.6.1. Types of background

The cross section for $B\bar{B}$ hadronic events is approximately 1.1 nb, whereas the dominant process in terms of rate is Bhabha scattering ($e^+e^- \to e^+e^-$), which, after applying analysis selection criteria, has an effective cross section of about 125 nb. Backgrounds such as Bhabha scattering are typically categorized as background *events* or *processes* and are largely suppressed by the trigger system or by the application of basic selection cuts in physics analyzes.
A second class of background arises from physics processes that closely imitate the signal under study. For example, in analyses of the decay $B \to K^{(*)}\ell^+\ell^-$, a major background source is the decay $B \to J/\psi K^{(*)}$, which has a substantially larger branching fraction and a very similar final-state topology. This type of background is referred to as *physics background*. Additional physics backgrounds arise from other decay processes that can mimic the signal, particularly when combined with particle mis-identification or mis-reconstruction of the event kinematics or topology.
A third class of background originates from non-resonant $e^+e^- \to q\bar{q}$ hadronic events, in which light quarks hadronize, thereby emulating the characteristics of signal events. This background is denoted as *continuum background*.
The fourth background category, the main subject of the remainder of this section, are *beam-induced* backgrounds (or short beam backgrounds). Beam backgrounds originate from beam-beam and beam-environment interactions, in which stray particles deviate from the nominal beam trajectory and subsequently collide with accelerator components or residual gas molecules, thereby producing secondary particle cascades [60, 61]. These backgrounds affect detector occupancies, pile-up, radiation dose, and noise levels and therefore directly impact the performance and lifetime of the sub-detectors, as well as the irreducible analysis background and the stability of trigger and DAQ systems. The sub-detectors that are most vulnerable to beam-induced backgrounds are the TOP particle identification system and the CDC. As the instantaneous luminosity at SuperKEKB ramps up toward its target value by increasing the beam currents and reducing the beam sizes with the nano-beam scheme, the background levels grow correspondingly.

3.6.2. Beam background sources

Beam-induced backgrounds arise from several distinct processes. That is, six main sources listed in Table 3.3 are relevant for Belle II [32]:

Radiative Bhabha scattering corresponds to the process $e^+e^- \to e^+e^-\gamma$, in which an energetic photon is radiated predominantly along the beam direction. These photons can interact with the iron of the accelerator magnets downstream of the interaction region and initiate photo-nuclear reactions that produce neutrons. These then scatter back into the Belle II detector and contribute particularly to the background in the outer KLM system. Since the radiative Bhabha rate is proportional to the instantaneous luminosity $\mathcal{L}_{\text{inst}}$, it constitutes one of the major luminosity-related background sources at Belle II. For the mitigation of this background source, additional neutron shielding has been installed.

Table 3.3.: Main beam background components at SuperKEKB/Belle II, their qualitative rates [32], and dominant parameter dependencies. The reported rates are based on beam background simulations provided by the accelerator group.

Type	Rate (MHz)	Dependencies
radiative Bhabha	2739	$\propto \mathcal{L}_{\text{inst}}$
two-photon QED	206	$\propto \mathcal{L}_{\text{inst}}$
beam-gas interactions	157	$\propto I_{\text{beam}} \cdot P_{\text{vac}}$
Touschek scattering	114	$\propto I_{\text{beam}}^2 / \sqrt{\beta_y^*}$
synchrotron radiation	< 1	$\propto B^2 \cdot E_{\text{beam}}^2$
injection background	-	$\propto I_{\text{beam}}$

Another problem is that the scattered electrons and positrons in radiative Bhabha events lose energy and experience "over-bending" in the downstream magnets, which causes them to hit the beam pipe and produce electromagnetic showers. This effect is reduced by using separate magnets for the incoming and outgoing beams so that only particles with very large energy losses are intercepted in the magnet region.

Two-photon processes of the type $e^+e^- \rightarrow e^+e^-e^+e^-$ and related QED interactions generate low-momentum electron-positron pairs. These low-momentum particles can spiral along the solenoidal magnetic field lines and leave multiple hits in the inner-most detector layers, thereby increasing occupancies and fake-hit probabilities in the vertex detector. Similar to radiative Bhabha scattering, the rate of these processes scales with luminosity $\mathcal{L}_{\text{inst}}$ and thus becomes increasingly relevant as SuperKEKB moves toward higher luminosities.

Beam-gas interactions constitute a major single-beam background source and are caused by elastic and inelastic scattering of beam particles on residual gas molecules in the vacuum chamber. Therefore, this interaction is proportional to the beam current I_{beam} and the residual gas pressure P_{vac}. These interactions can deflect beam particles or produce secondary particles, leading to additional losses and shower particles that may reach the detector. Beam-gas backgrounds are mitigated using movable collimators and shielding elements. Further mitigation includes improved vacuum design and continuous vacuum scrubbing of the beam pipes, reducing P_{vac} and thus the rate of beam-gas interactions.

Touschek scattering is an intra-bunch Coulomb-scattering process in which momentum is transferred between particles within the same bunch, producing off-momentum particles that can be lost on the beam pipe. The resulting particle losses generate secondary radiation and shower particles that can enter the detector volume and create additional hits. This, in particular, affects the inner-most tracking detectors and the calorimeter. Since the Touschek rate scales inversely with the transverse beam size β_y^*, this contribution increases significantly at SuperKEKB compared to KEKB, which is a direct consequence of the nano-beam scheme [32]. As for beam-gas

backgrounds, the mitigation is based on movable horizontal and vertical collimators and metal shields that intercept off-momentum particles before they reach the IP.

Synchrotron radiation is emitted when the electron and positron beams are bent in the magnetic fields of the accelerator. A fraction of the emitted photons can strike the inner beam pipe in the vicinity of the interaction region and scatter into the Belle II detector, potentially depositing energy in the inner-most layers. To suppress synchrotron radiation backgrounds, the inner surface of the beam pipe is coated with gold and equipped with ridge structures that prevent photon propagation toward the IP.

Injection background occurs when new bunches are injected into the storage rings multiple times per hour and not all particles are immediately captured in stable orbits. This effect is *e.g.* , attributable to imperfections in the injection kicker, which cause transient bursts of beam losses and associated detector activity. To suppress this background contribution, a trigger veto synchronized with the injection period is implemented, at the cost of increased DAQ dead-time.

Detailed simulations together with dedicated machine studies make it possible to decompose the various background contributions by operating with and without collisions and by tuning beam parameters that influence backgrounds, thereby exploiting their differing impacts on the individual background sources. These comprehensive simulations enable projections to future machine configurations, forming the foundation for future upgrade studies [60, 61]. A description of the L1 trigger rate calculation and extrapolation methodology can be found in the appendix A.2.2.

3.6.3. Background conditions across runs

Beam-induced backgrounds in the CDC depend on the instantaneous luminosity, beam conditions, and gas properties of the chamber. In order to monitor the background level, the background-sensitive observable number of extra CDC hits $\langle n \rangle_{\text{extraCDChits}}$, which counts hits that are not associated with any reconstructed CDC track and are therefore attributed to beam-induced background rather than physics tracks, is shown in Figure 3.9. Despite intra-run variations due to *e.g.* beam tuning, a clear trend is visible: higher instantaneous luminosity, that is implicitly increased over experiments and runs, generally results in a larger number of beam background hits.

Towards the end of Run I (experiments 22 to 26), the background increased by orders of magnitude (from 127 to 1437 hits per event) within a period of approximately seven months due to an increase in instantaneous luminosity. The same trend can be seen for Run II data. However, experiment 35, run 740 (October 2024) shows a significantly higher average number of beam background hits than experiment 26, run 1430 (1605 vs. 1179), even though the peak instantaneous luminosity is lower ($2.11 \cdot 10^{34}$ vs. $3.64 \cdot 10^{34}$ $\text{cm}^{-2}\text{s}^{-1}$) and the beam conditions are comparable. This indicates additional contributions originat-

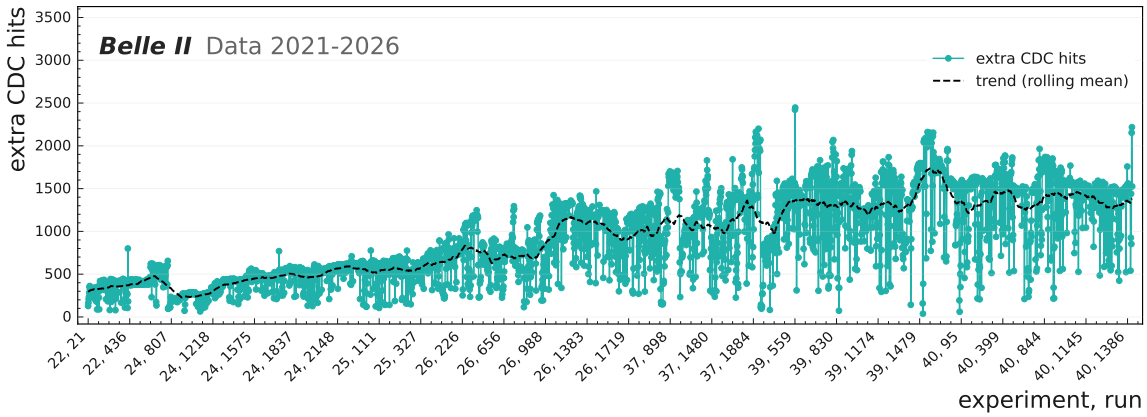


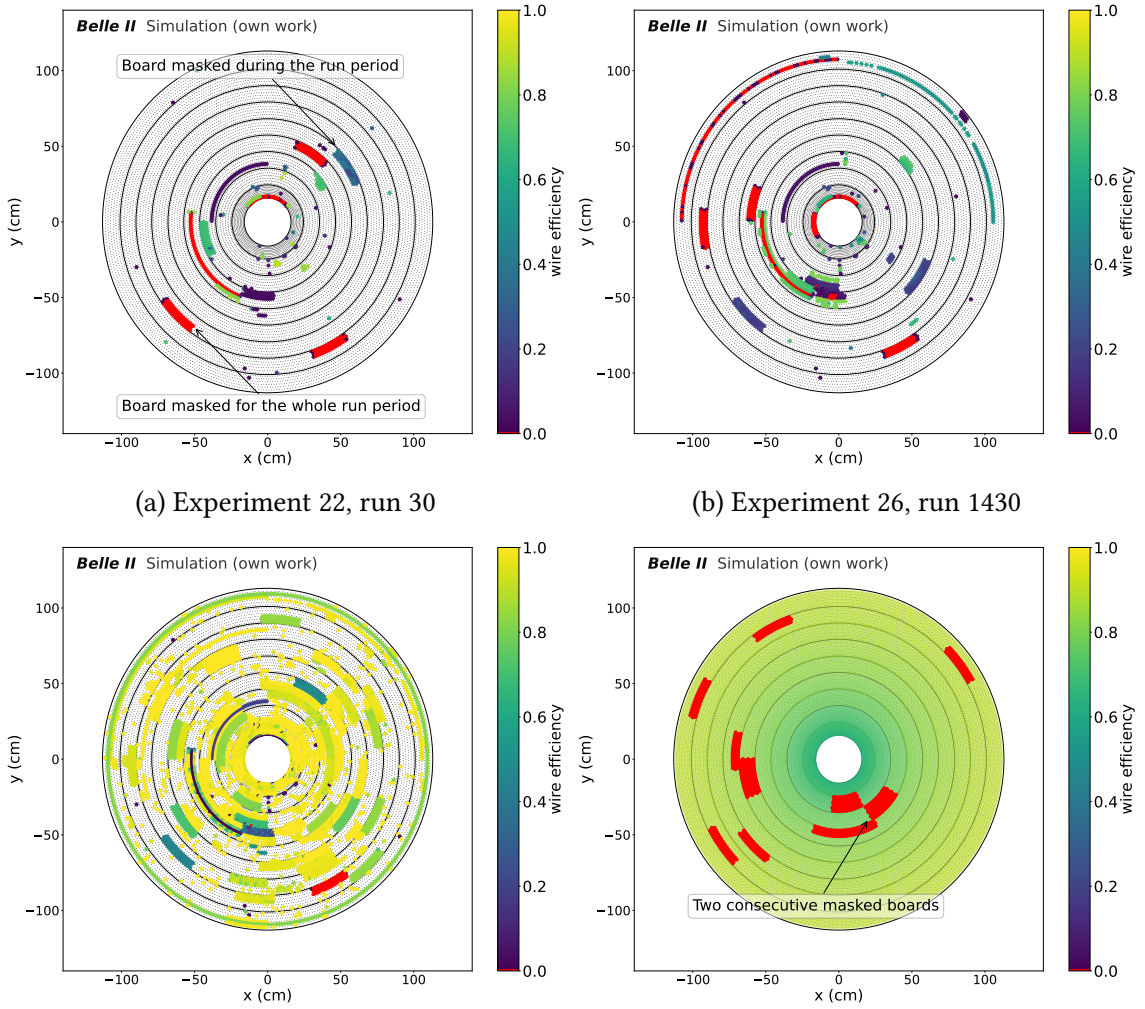
Figure 3.9.: Increasing number of extra CDC hits $\langle n \rangle_{\text{extraCDChits}}$ as a measure of increasing background levels shown for experiments 21 to 40 (November 2021 to April 2026). Values provided by [62].

ing from the CDC itself, in particular variations in gas conditions and possible detector aging phenomena such as the Malter effect described in section 3.7.
The highest background conditions observed to date occurred in experiment 39, where the average number of extra CDC hits reached $\langle n \rangle_{\text{extraCDChits}} = 2450$ in run 561 at the beginning of February 2026, corresponding to about 17 % of all CDC wires being hit on average per event. At the target luminosity, the expected background level is $\langle n \rangle_{\text{extraCDChits}} = 2800$, which implies an occupancy of roughly 19 % of CDC wires per event. For comparison, a typical $e^+e^- \to \mu^+\mu^-(\gamma)$ event produces $\mathcal{O}(100)$ CDC hits, whereas an average $e^+e^- \to B\bar{B}$ event induces $\mathcal{O}(700)$ hits from charged decay products in addition to the superimposed background contribution.

The progressively increasing wire occupancy presents substantial challenges for both the current trigger system and the offline reconstruction algorithms. On the one hand, dense background levels directly degrade tracking efficiency and spatial-momentum resolution while increasing the rate of spurious (fake) and duplicate (clone) track candidates. On the other hand, higher occupancy results in larger event sizes and consequently increases the computational cost of the reconstruction, which exhibits approximately quadratic scaling with respect to the number of CDC hits. Furthermore, increased wire occupancies accelerate detector aging, as discussed in the subsequent section.

3.7. CDC degradation

Beam-induced backgrounds do not only directly influence the track reconstruction performance, but also the operational lifetime of the CDC. In the following, two principal consequences for the CDC, illustrated in Figure 3.10, are examined in detail: (i) reductions in wire detection efficiency and (ii) masking of readout boards affected by (temporal) radiation-induced damage.



(a) Experiment 22, run 30

(b) Experiment 26, run 1430

(c) Run I average

(d) Approximated conditions for Run II.

Figure 3.10.: Wire efficiency maps of the CDC used in simulations for different experiments and runs are shown in Fig. 3.10a and Fig. 3.10b, along with the average map over the full Run I period and approximated conditions for the Run II period. Colored wires have reduced efficiency (<1); red wires are completely off. Large colored regions indicate disabled boards: if red, they were off for the entire period; otherwise, they were disabled only during the time when a specific issue occurred. Figures are taken from [63].

3.7.1. Wire efficiency loss

The CDC wire efficiency is mainly effected by the Malter effect [64, 65], which is the direct cause of higher wire occupancies that lead to an increase in the total chamber current and result in an accelerated rate of material deposition and charge accumulation on the wires. These effects alter the local electric field configuration and reduce the amplification gain. This makes the wires less sensitive to ionization signals, resulting in fewer registered hits along particle tracks. This is especially true in the inner-most CDC layers, where the particle flux is highest due to their proximity to the IP. In severe cases this

can lead to micro-discharges or electrical instabilities that introduce additional noise and further compromise track reconstruction. Due to this gain reduction, the reconstruction algorithms have fewer reliable points to identify and reconstruct tracks, which increases the uncertainties on track parameters and, in particular, decreases the momentum resolution. In addition, particle identification, which is based on $\mathrm{d}E/\mathrm{d}x$, is affected due to the decreased collected charge and the reduced number of hits.

3.7.2. Masked boards

Another major challenge for the track reconstruction algorithms arises from CDC readout boards that are masked during data taking. The FEE boards are located within the detector volume and are therefore continuously exposed to high radiation levels. Radiation gradually degrades the electronics and can cause failures that corrupt data, produce un-physical outputs, or, in severe cases, destabilize the DAQ. High input rates from beam-induced background further stress the readout by filling buffers faster than they are emptied, driving the system into a busy state and temporarily suspending data collection. For example, low-kinetic-energy background neutrons increase the rate of single-event upsets (SEUs) in the CDC FEEs. Such soft errors like SEUs do not permanently damage the hardware, but disrupt operation until reset and can therefore disrupt the Belle II DAQ system and consequently degrade the overall data-taking efficiency.

Masking instructs the DAQ system to ignore hits from a given board, either proactively if a malfunction is already known or dynamically when a DAQ problem is detected. For recurring problems, this is particularly advantageous: while an occasional firmware reset is acceptable, frequent interruptions would cause significant dead time, whereas masking provides a temporary mitigation at the cost of a modest loss of information. The main drawback is the reduction in detector acceptance and coverage. Within each CDC superlayer, the boards are arranged in ϕ, and masking a single board, or two boards aligned in ϕ, creates localized gaps without recorded data. The track reconstruction must then extrapolate across these gaps, increasing uncertainties on track parameters and, consequently, on physics observables. This is especially problematic when these gaps occurs in an axial layer, since track finding relies exclusively on axial layers.

3.8. Detector upgrades

Achieving the design luminosity of SuperKEKB requires comprehensive modifications to the interaction region, which in turn requires an extended shutdown period, referenced as long shutdown 2 (LS2). To preserve physics performance and trigger efficiency in the presence of enhanced beam-induced backgrounds accompanying the luminosity increase, a new inner detector system and several associated detector and trigger upgrades are planned, making full use of the shutdown period for their implementation. This section outlines principle planned upgrades, with particular emphasis on the CDC and the L1 trigger.

3.8.1. CDC upgrades

The CDC is subject to several upgrade considerations, motivated by the expected increase in beam-induced background levels, as its present performance is already being adversely affected by the current beam background conditions, as discussed in section 3.7. The inner-most CDC layers are most exposed to beam backgrounds and corresponding increasing occupancies and detector aging. One of the proposed mitigation strategies consists of reducing the gas amplification gain of the inner-most sense wires, or, in a more radical approach, removing or partially deactivating the most severely affected inner-most wire layers.

3.8.1.1. CDC trigger hardware upgrades

To cope with the increased beam-induced backgrounds expected at the target luminosity, and to efficiently trigger low-multiplicity non-B physics processes (*e.g.* τ, di-lepton final states, and dark/new particle searches), a series of hardware and firmware upgrades is planned for the L1 trigger system [7]. First, the fraction of UT4 boards increases in the near term. In addition, a UT5, based on Xilinx Versal FPGAs that provide extensive DSP resources, is under development (2024-2032). The UT5 is designed to enable more sophisticated trigger algorithms, including those based on machine learning.
Around 2026, an upgrade of the CDC FEEs boards is planned, employing optical transceivers with increased radiation tolerance to ensure the reliable operation of the CDC readout system until the end of the Belle II experiment. Furthermore, the FEE upgrade program encompasses targeted improvements in mitigation of cross-talk effects and an increase in data transfer throughput [7]. The optical link bandwidth will be increased from 3 to 10 GB/s, allowing transmission of complete TDC and ADC information for all sense wires, rather than for only a subset. In addition, the precision of the ADC count will be improved by employing a summation of 25-samples, and the TDC resolution will be improved to 1 ns consistent with data transmitted to the DAQ system. The changes are summarized in Table 3.4.

Table 3.4.: Information send from CDC front end board to the L1 TRG, taken from [7] and from private communication with T. Koga [66].

sent to the CDC TRG	present	new
existence of wire hit	45 of 56 layers	all layers
hit timing (TDC)	9 of 56 layers 2 ns precision	all layers 1 ns precision
charge (ADC)	9 of 56 layers 3-point sum 1 bit	all layers 25-point sum 4 bit

To accommodate the higher input bandwidth, the CDC merger boards will be migrated to UT4 boards. The firmware will be comprehensively redesigned to take advantage of

enhanced TDC / ADC data, with the aim of improving the tracking performance and suppressing background contributions.

3.8.1.2. CDC trigger firmware upgrades

The current CDC track trigger utilizes a two-dimensional Hough transformation applied to hits from the five axial super-layers to reconstruct the transverse momentum p_T and the azimuthal angle ϕ of track candidates. An extension of this method to a fully three-dimensional Hough transformation is under development, in which hits from the stereo super-layers are incorporated directly into the Hough parameter space [67]. A more detailed description is provided in subsection 4.2.2.
The current neural z-trigger, which processes the input candidates provided by the Hough finder, utilizes neural network-based algorithms for track fitting. To accommodate the higher trigger rates anticipated at increased instantaneous luminosities, an upgraded deep neural network first-level hardware track trigger [9], based on a deep neural network (DNN) architecture, has been developed. In addition, an expanded set of input observables for this DNN is under investigation. The implementation of the DNN-based neural network trigger is planned to be integrated on the same board as the proposed 3D Hough finder in order to minimize data transmission latency. This integrated system will then replace the two-dimensional Hough-based track finder currently used, the additional event time finder (ETF), as well as the existing three-dimensional track fitting algorithm.
Moreover, a dedicated displaced-vertex trigger [68] is being studied to enhance sensitivity to event topologies in which two oppositely charged particles originate from a common decay vertex that is significantly displaced from the IP.

3.8.2. New inner sub-detectors

Due to a modified focusing system at the IP aimed at increasing the luminosity, an upgrade of the inner part of Belle II is required. The primary option under investigation is a new inner tracking system, denoted vertex detector (VTX) [69], which is intended to replace both the PXD and the SVD. This replacement is associated with an increased buffer capacity, which in turn will increase the total L1 trigger latency requirement from 5 µs to approximately 10 µs.
In a configuration in which the inner layers of the CDC are removed, the implementation of an additional detector subsystem between the VTX and the CDC is under consideration. This inner tracking and timing detector (ITT) [70] might comprise a silicon strip tracker, analogous to the current SVD, as well as an additional fast timing layer (FTL) designed to improve the identification of low-momentum particles via time-of-flight measurements.
Currently, the PXD and SVD subsystems do not deliver trigger signals. With the planned VTX upgrade, a dedicated vertex trigger could be implemented to complement the single track trigger (STT) in suppression of beam-induced backgrounds originating outside the nominal interaction region.

4. Tracking at Belle II

In Belle II, two conceptually independent track reconstruction approaches are used. The first is the offline track reconstruction, which is primarily used for physics analyzes but is also executed within the HLT for early event filtering. The second is the tracking algorithm implemented at the L1 trigger level on FPGA-based hardware. The latter operates under stringent latency constraints and, as a consequence, provides reduced reconstruction precision compared to the full offline tracking. In the following, both track reconstruction algorithms are described in detail.

4.1. Offline track reconstruction

The offline track reconstruction (tracking) in Belle II [45] is a multi-stage process that combines measurements from all three tracking detectors, PXD, SVD, CDC, into a single set of charged particle trajectories. The reconstructed tracks are described by five helix parameters at the point of closest approach (POCA) to the nominal IP: the transverse impact parameter d_0, the azimuthal angle ϕ_0 of the transverse momentum at the POCA, the signed curvature ω, the longitudinal impact parameter z_0, and the tangent of the dip angle $\tan\lambda$.
In this section, I summarize the baseline tracking algorithm implemented in basf2 [71, 72]. The full offline track reconstruction is divided into several dedicated algorithms tailored to the different sub-detectors:

1. **CDC hit preparation**: The reconstruction starts with a pre-processing of the CDC hits for tracking and applying a basic hit pre-selection.

2. **CDC track finding**: After hit preparation, two independent CDC track finding algorithms are run sequentially to identify potential track candidates and attach associated hits. The first is a global pattern recognition based on the Legendre transformation [73], which is followed by a local cellular automaton [74] that builds short tracks without assuming an IP origin. The resulting track candidates are then merged.

3. **CDC-SVD combination**: The CDC tracks seed a combinatorial Kalman filter (CKF) that extrapolates the tracks into the SVD volume. It attaches compatible silicon strip clusters using a multi-variate analysis (MVA) classifier to select the most probable continuation of each track candidate.

4. **SVD-only track finding**: Tracks that leave too few hits in the CDC, typically with transverse momentum below 100 MeV/c, are reconstructed in the SVD by a second cellular automaton.

5. **Combined fit**: The resulting SVD tracks are extrapolated to the remaining CDC hits and refitted, complementing the CDC-seeded tracks.

6. **PXD attachment**: PXD hits are attached to the CDC-SVD tracks in a final CKF step, significantly improving the d_0 and z_0 resolutions by the high-resolution PXD space points.

7. **Track fitting**: In the final `GENFIT2` [75] track fitting stage, the track parameters are refined while accounting for energy loss and material effects. The resulting helix parameters and covariance matrices are stored for analysis.

In the context of this work, I focus on the CDC-only modules 1, 2, and 7, as this choice enables a direct comparison between the graph neural network (GNN)-based filter developed in this work and the existing CDC hit filters. The CDC-only modules will be explained in more detail in the following.

4.1.1. CDC hit preparation

In the first tracking step, the raw CDC signals are converted to *wire hits* and initial background suppression is applied by the `WireHitPreparer` module [72]. Early hit filtering in the CDC is crucial for constraining the combinatorial complexity of track reconstruction, which scales approximately $\propto N_{\text{hits}}^2$ with the number of recorded hits. Conceptually, this stage comprises the following steps:

1. Application of basic quality and timing cuts to remove clearly un-physical or out-of-time signals.

2. Bad wire masking based on the recorded CDC conditions of the used data run.

3. Hit filtering using one of the two given filters:
   a) The **legacy cut-based hit filter** uses fixed cuts, namely ADC≥15, TOT≥2, and ADC/TOT ≥3 for the inner-most and ADC≥18, TOT≥2, and ADC/TOT ≥3 for the other super-layers.
   b) The **MVA-based hit filter** employs FastBDT [76] gradient-boosted decision trees for per-hit classification.

4. Cross-talk filtering that requires the number of hits per readout ASIC to be within set limits and uses the deviation of each TDC from the median TDC to identify and discard cross-talk hits.

The GNN filter introduced in this work is employed at this tracking stage and included into the CDC `WireHitPreparer` module as an alternative to the pre-existing filtering algorithms. The corresponding filtered hits are marked as background by dedicated background flags, ensuring that they are excluded from the subsequent tracking stages.

4.1.2. CDC track finding

The baseline CDC reconstruction employs two complementary pattern-recognition algorithms, described in detail in [45]: a global Legendre-transform-based method and a local cellular-automaton-based track finding.

Global CDC track finding is initiated using information from the sense wires in the axial layers and by searching for high-density regions in the Legendre space. Track candidates are reconstructed in an iterative procedure that keeps track candidates satisfying predefined quality criteria, such as a minimum number of associated hits, and removes these hits from further iterations. A fast fit refines parameters, merges clones or loop fragments, and resolves hit ambiguities.
Subsequently, the hits from the stereo layers are associated with the transverse 2D trajectories and are used to determine the longitudinal track parameters by identifying dense regions in a second Legendre space. Stereo hits that are compatible with more than one track candidate are excluded from all corresponding candidates.

The local CDC finder applies a weighted cellular automaton on graphs built from the hits without requiring tracks to point to the IP. It first builds segments from triplets of neighboring hits within a super-layer (vertices), weighted by χ^2 of a least-squares fit, and connects neighboring triplets that share two hits to edges. In a second stage, axial and stereo segments from adjacent super-layers are combined into longer track candidates using circle and line fits (Riemann-circle methods). Finally, the cellular automaton selects high-weight paths as track candidates.

The global and local finder outputs are then merged using a FastBDT with a priority on the global finder outputs. The resulting found tracks comprise an initial estimate of the track parameters, including the point of origin, the momentum, and the electric charge, as well as an ordered sequence of associated hits. However, at present the cellular-automaton is not fully used, reducing the efficiency for displaced vertices.

4.1.3. CDC track fitting

Track fitting is performed using a CKF-based algorithm implemented in the `GENFIT2` [77] framework. Starting from an initial seed provided by the track finding stage, the fitter propagates the track state through the detector and updates it sequentially with each associated hit. Since this procedure relies on a consistent, ordered sequence of measurements, it exhibits a high sensitivity to incorrectly assigned hits originating from other tracks or from a beam-induced background. Such wrongly assigned hits can significantly compromise the fit quality or even cause the fit to fail. To address this issue, a deterministic annealing filter (DAF) is applied prior to the CKF application in order to down-weight or exclude hits that are incompatible with the track hypothesis.
The propagation explicitly accounts for material effects, including ionization energy loss and multiple Coulomb scattering, as well as for the non-uniform solenoidal magnetic field within the detector volume. For a consistent treatment of the material interactions,

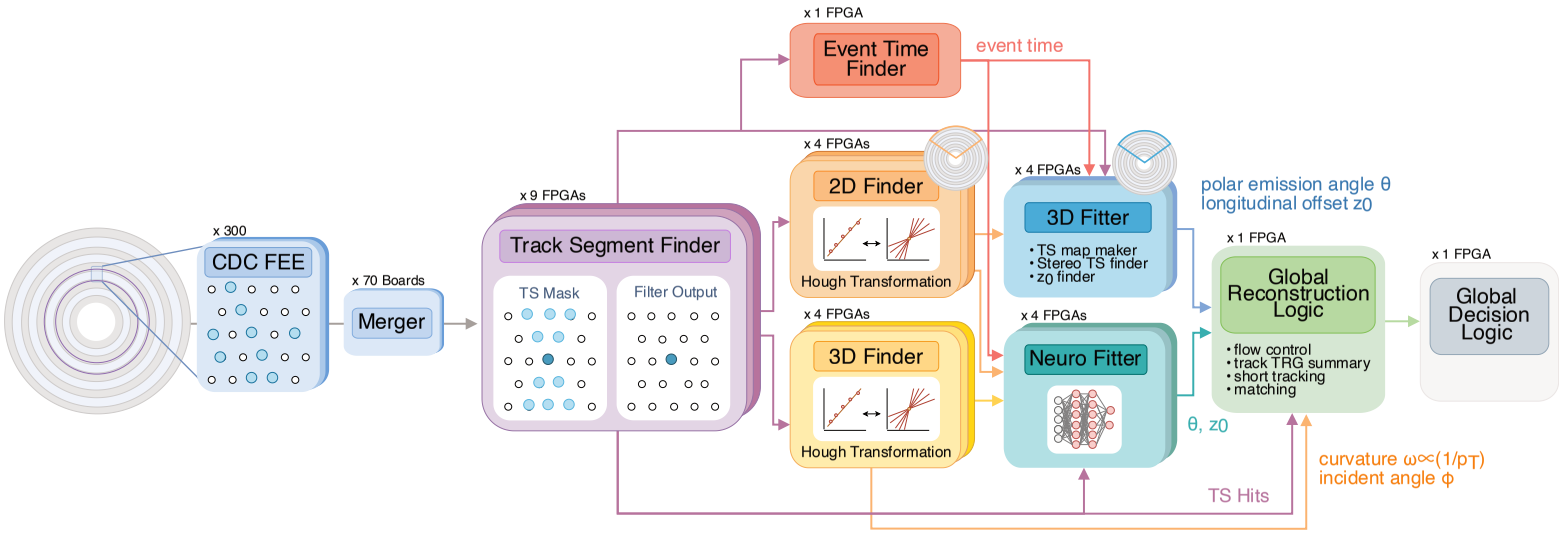
separate fits are performed under different particle-mass hypotheses (pion, kaon, and proton), ensuring that ionization losses and multiple-scattering contributions are evaluated according to the assumed particle species. In this thesis, the pion mass hypothesis is adopted as the default configuration for all performance studies.

4.2. The track trigger system

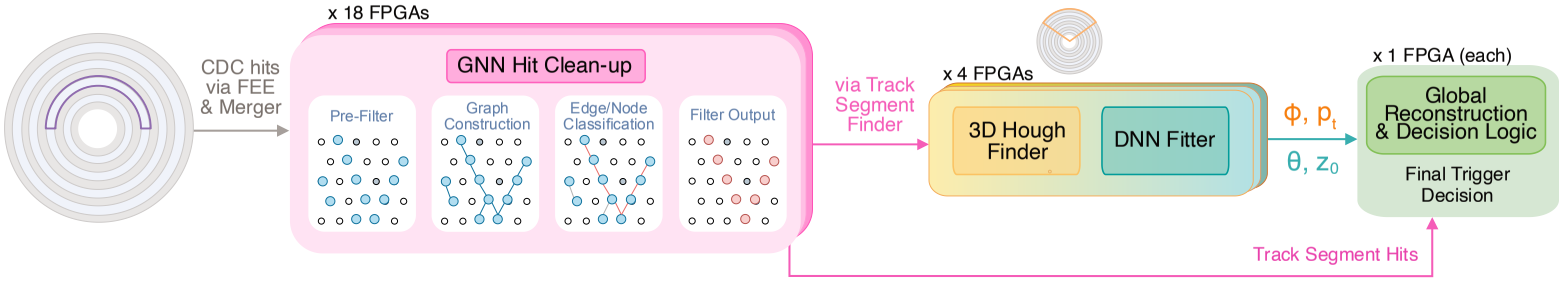
The CDC trigger system constitutes one of the core subsystems of the L1 trigger system, tasked with the real-time reconstruction of charged-particle trajectories. The CDC trigger is implemented on FPGA-based universal trigger boards and operates at a clock frequency of 31.804 MHz. The CDC trigger chain, schematically depicted in Figure 4.1 for the current setup as well as a proposed future setup, is structured as a sequence of successive processing stages. An overview of the complete processing chain is presented in the following sections.

4.2.1. Track segment finding

Following the CDC TRG FEEs and the merger boards, the TSF constitutes the first processing stage within the CDC trigger system. Its primary objective is the reduction and compression of the raw hit information. This is achieved by associating time- and position-coincident wire hits into local TSs within each super-layer, employing predefined geometrical patterns in conjunction with LUT-based logic. The TSs constitute the fundamental input objects for all subsequent track reconstruction stages and thus provide the interface between the raw CDC signals and higher-level online reconstruction [78].
A track segment is defined as a compact predefined pattern of neighboring sense wires that span five adjacent layers around a central priority wire, with distinct pattern definitions employed for the innermost super-layer and for the outer super-layers, as shown in Figure 4.2a. Before event-wise TS construction, the TSF applies basic data-quality checks, including the masking of known dead channels and the rejection of characteristic cross-talk clusters characterized by dense time-coincident hit clusters on the same front-end chip. Subsequently, for every predefined segment pattern, the TSF continuously tests whether the instantaneous pattern of wire signals in an event is compatible with a valid track segment.
The corresponding decision logic is realized as a finite-state machine, encoded in predefined data-driven LUTs. The current configuration requires at least four hits in consecutive TS-layers for a TS to be formed. For each accepted segment, the algorithm assigns a discrete identifier that classifies the corresponding hit-pattern topology as indicative of a left- or right-side passage of a potential track, or as an undefined type, as illustrated in Figure 4.2b. In addition, timing quantities are computed, such as the time of the priority hit and the earliest hit time within the segment.
In high-occupancy conditions, a single physical track or beam-induced background can produce overlapping hit patterns in neighboring segments, resulting in redundant information. To mitigate this effect, the TSF applies neighbor suppression: segments whose priority hit matches the central priority position are retained, while active segments

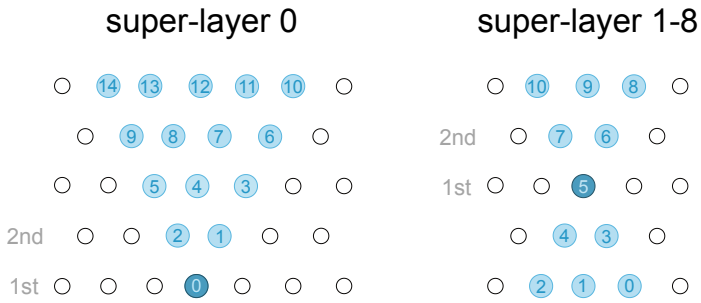


(a) Current CDC trigger pipeline

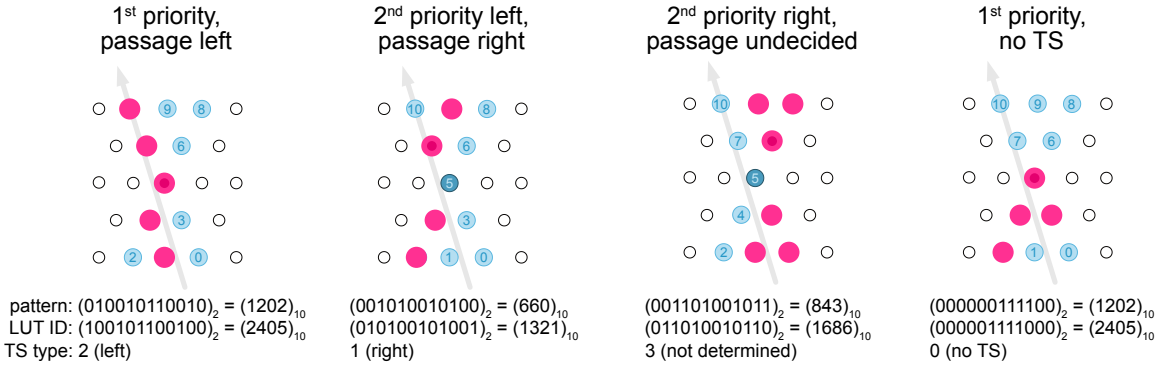


(b) Proposed future CDC trigger pipeline

Figure 4.1.: The CDC L1 trigger consists of multiple modules. In the current setup (a), hit information is sent directly from the FEEs to the track segment finder (TSF) via merger boards. The track segment finder (TSF) output goes to all subsequent modules. Tracks are found by the 2D Hough finder, and soon also the 3D finder (already implemented in simulation and planned for near-future detector integration). The resulting tracks are sent to the track fitters, and all track information is then processed by the global reconstruction logic (GRL) and global decision logic (GDL) for the final trigger decision. In a future setup (b), the merger board output first passes through the GNN hit filter before reaching the TSF. The 2D Hough finder and the event time finder (ETF) will be replaced by the 3D Hough finder, planned for integration on the same board as an updated neural network fitter (DNN fitter).



(a) track segment shapes



(b) track segment types

Figure 4.2.: (a) The track segments (blue) form a triangle in super-layer 0 and an hourglass shape in super-layers 1-8. (b) Based on the hit wire patterns in each track segment (TS), different TS types are defined. Each pattern receives an identifier derived from the binary code of hit wires (hit = 1, otherwise 0). For the LUT identifier, this binary code is shifted by one bit, and 1 is added if, among all priority wires, only the second-priority left wire is hit. The TS type, which can be 0 (no TS), 1 (right passage), 2 (left passage), or 3 (undetermined), is then stored in the LUT at the entry indexed by the decimal value of the LUT code.

whose priority position is displaced from the TS center and have an active neighbor in the same layer are suppressed.

4.2.2. Hough track finding

The CDC L1 trigger reconstructs charged-particle trajectories in the transverse r-ϕ plane by application of a circular two-dimensional Hough transformation [8, 67] applied to the TSs introduced in subsection 4.2.1. Following a conformal mapping of the TSs into the Hough coordinate system, each axial TS is represented as a straight line in the Hough parameter space spanned by the azimuthal angle ϕ and the curvature $\omega \propto 1/p_T$. In this representation, physical tracks correspond to intersections of multiple such lines.

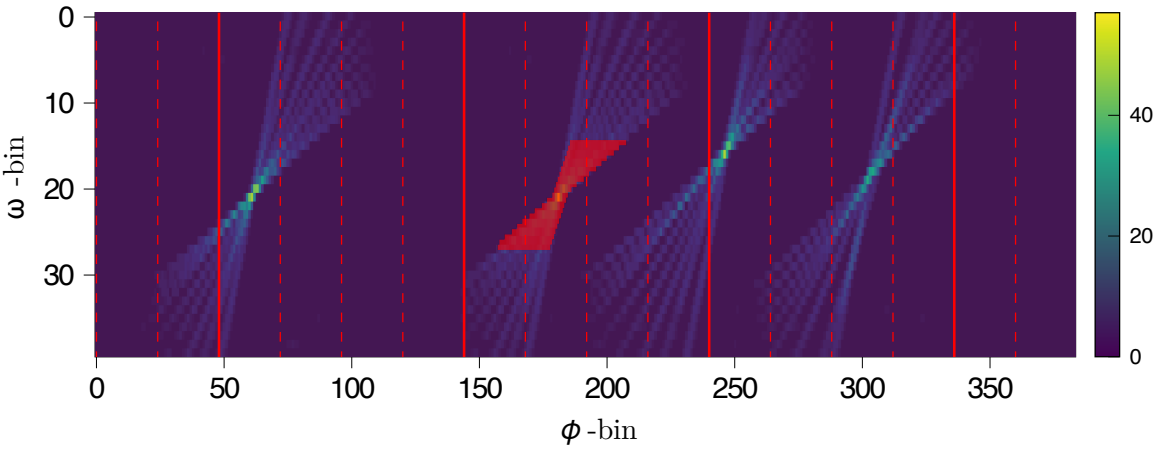


Figure 4.3.: Example of an event in the Hough plane (ϕ-ω). Each hit maps to a line, and tracks correspond to local maxima in this plane. Around each identified maximum, a predefined hourglass-shaped region (red) is cut out. The red lines mark the CDC regions used to parallelize the algorithm. Figure adapted from work by Simon Hiesl.

The **2D Hough Finder** is based on the Hough algorithm, illustrated in Figure 4.3, that discretizes the parameter space into a finite lattice of Hough cells, and identifies track candidates as local maxima that satisfy the requirement of at least four contributing axial super-layers out of a possible five. This selection provides a substantial suppression of electronic noise and beam-induced background, at the cost of a reduction in track finding efficiency, as trajectories traversing the detector end-caps, as well as generally short tracks, are intrinsically not reconstructible by design. The resulting two-dimensional track candidates (*2D tracks*), parameterized by ϕ and curvature ω, and their associated TSs, are subsequently passed on to downstream track fitting algorithms.

The upgraded **3D Hough Finder** [67] planned to be integrated into the Belle II CDC L1 trigger in the near future extends the existing 2D Hough track finder to allow the reconstruction of fully three-dimensional trajectories. The 3D Hough finder combines axial and stereo TSs to track candidates by performing an analogous multi-dimensional clustering in the Hough parameter space. For the clustering step, it employs a fixed three-dimensional window centered on the global maximum, with configurable criteria on the minimum cell weight of the peak, the total cluster weight, and the cut-out shape of neighboring cells, where the cut-out size is independently configurable in all three dimensions (ϕ, ω, θ). Analogously to the 2D Hough finder, the 3D Hough finder imposes the requirement that TSs be present in at least four out of the five axial super-layers and, in addition, in at least three out of the four stereo super-layers for a track to be reconstructed. The 3D Hough finder is executed on four processing boards in parallel, with each board responsible for one azimuthal quadrant of the entire CDC plane. Duplicate tracks are rejected downstream by the GRL on the basis of their similarity in curvature and transverse momentum (p_T). Due to latency constraints, each board is limited to reconstructing at most four tracks.

4.2.3. Event time finding

The event time t_0 is determined by one of two distinct CDC ETF implementations. The **legacy ETF** estimates t_0 directly from the raw arrival times of TS hits, without invoking any track reconstruction. For each TS, the ETF processes only the earliest priority cell times. To suppress the influence of noise-dominated super-layers, the algorithm restricts the input to at most ten hits per super-layer and clock cycle. The event time is then defined as the smallest time value at which at least three hits are registered. This approach is conceptually simple and computationally efficient. However, it is vulnerable to beam-induced background hits that can arrive early and might therefore introduce a systematic bias in the t_0 estimate.

Therefore, the CDC trigger utilizes an alternative **Hough ETF** based on the two-dimensional Hough transformation to improve the robustness of the determination of the event time in the presence of a beam-induced background. The primary distinction from the 2D Hough algorithm employed in the track finder is that the TSs are temporally ordered according to their arrival times and are injected into the Hough plane in discrete increments of one data-clock period (32 ns). The search is terminated as soon as the first valid Hough peak is identified. Since t_0 is inferred from the earliest hit cluster that is compatible with a physical track signature, rather than from the global earliest hit, this method is substantially less vulnerable to contamination from early background hits than the legacy ETF.

4.2.4. Track fitting

The track candidate outputs of the 2D Hough finder carry only transverse-plane parameters from the axial super-layer TSs (ϕ, ω) and provide no information about the longitudinal direction. Two subsequent fitting stages, separated in the basf2 TRG simulation but unified in a single firmware block in the detector, refine these track candidates and extend them to full three-dimensional helical tracks.

The **2D track fitter** improves the transverse track parameters by performing a weighted least-squares circle fit to the axial TSs associated with each Hough candidate. In addition to the priority wire positions, it incorporates drift-time information: the raw hit time of each priority cell is corrected for t_0 and converted to a drift length via a calibrated x-t relation. The fit minimizes the residuals between the drift-corrected hit positions and the fitted circle and provides a charge-sign estimation.

The **3D track fitter** extends each refined 2D track into three dimensions by incorporating hits from the four stereo super-layers. Since the stereo wires are tilted with respect to the beam axis, a charged particle crossing a stereo layer at a known radius and azimuthal angle produces a hit whose longitudinal coordinate can be inferred from the wire geometry and the measured drift length. The fitter first narrows the set of candidate stereo hits to those geometrically compatible with the extrapolated 2D helix, then resolves the remaining left-right and layer ambiguities by performing a Hough vote and a final linear

least-squares fit in the longitudinal parameter space spanned by the z-vertex position z_0 and the polar-angle cotangent $\cot\theta$. Together with the transverse parameters from the 2D track fitter, this produces a complete four-parameter helical track description $(\phi_0, \omega, z_0, \cot\theta)$ of the *3D track* candidate.

In parallel to the conventional track fitters, which suffer from combinatorial ambiguities under high-background conditions and cannot guaranty a fixed execution time, a **neuro-z trigger** [56] provides an independent estimate of the longitudinal track parameters. This neural network-based track trigger takes as input the same Hough-based track candidates and selects a suitable trained model from a bank of multi-layer perceptrons (MLPs), each specialized for a specific region of the track phase space. The network output directly provides the z-vertex position z_0 and the polar angle θ. The output of the neuro trigger are the so-called *neuro tracks*.
Currently, the neural trigger and the analytic 3D fitter deliver two independent estimates of the longitudinal track parameters. In future upgrade scenarios, only the neural trigger is foreseen to be employed and will be integrated on the same board as the 3D Hough-based track finder, thereby reducing the data transmission latency between the two processing stages.

4.2.5. Short track finding

The track finding and fitting algorithm described above is configured to reconstruct only "full tracks," defined as charged particle trajectories that traverse at least seven out of the nine CDC super-layers (four axial and three stereo) in the current configuration. Consequently, charged particles emitted at shallow polar angles toward the endcap regions, as well as low-momentum particles whose trajectories curl back within the CDC before reaching the outer super-layers, remain invisible to this reconstruction procedure. This limitation leads to a reduced trigger acceptance for a non-negligible fraction of physically relevant tracks.
To restore sensitivity to such short trajectories, a dedicated **short track finding** algorithm has been implemented directly in the GRL firmware. This short track finder exploits TS hit information from the five innermost super-layers that are not already associated with reconstructed full tracks. It discretizes the full 360° azimuthal range into 64 uniform bins used in a pattern-recognition approach to identify *short tracks* on the basis of predefined hit patterns.

4.2.6. Global trigger system

The two-stage global trigger system of Belle II, comprising the GRL and the GDL, consolidates the outputs of the four sub-trigger systems into a final L1 trigger accept-or-reject decision for each bunch crossing.

The GRL receives detailed trigger observables from all four sub-trigger systems:

- track parameters ϕ_0, ω, z_0, and θ for each of the CDC quadrants;

- cluster positions and energies from the ECL trigger;
- hit patterns from the KLM trigger; and
- photon arrival time patterns from the TOP trigger.

Based on these inputs, the GRL performs three categories of processing. Firstly, it summarizes the CDC tracking information: it counts full tracks within a sliding 500 ns timing window, suppresses duplicate tracks by requiring $\Delta\omega < 8$ and $\Delta\phi < 8$, and derives geometric relations between tracks, such as opening angles or back to-back topologies. Secondly, it identifies short tracks as described in subsection 4.2.5. Thirdly, it performs matching between full CDC tracks and hits in the outer sub-detectors by extrapolating each two-dimensional track to the corresponding detector radius and requiring azimuthal coincidences. All resulting trigger conditions are transmitted to the GDL as a compact set of binary bits.

The GDL receives these input bits from the GRL, together with direct inputs from the sub-trigger systems, and combines them using Boolean operations (NOT, OR, AND) according to a so-called trigger menu. This trigger menu is a predefined set of output bits, each corresponding to a specific physics or calibration trigger condition. It encompasses a broad spectrum of processes, ranging from high-multiplicity hadronic events selected by multi-track and opening-angle requirements to low-multiplicity dark-sector signatures exploiting the small-opening-angle short-track trigger. Furthermore, it also includes calibration samples selected by Bhabha and $\mu^+\mu^-$ back-to-back conditions.
Whenever any output bit of the trigger menu is satisfied, the GDL issues the L1 trigger accept signal, which initiates the readout of all sub-detectors by the DAQ system. The overall L1 trigger efficiency for hadronic $B\bar{B}$ and $q\bar{q}$ events exceeds 98 % when using CDC-based conditions alone, and reaches approximately 99 % when ECL-based conditions are included.

4.2.7. Trigger bits and decision logic

In the following, a concise overview of the most relevant stand-alone trigger bits of the CDC is provided. The input bits are transmitted from the GRL to the GDL, and the output bits of the GDL are subsequently used to generate a trigger accept-or-reject decision according to the predefined trigger menu.

4.2.7.1. Trigger input bits

The CDC stand-alone input bits provided by the GRL are derived from the multiplicity of 2D, 3D, neuro- and short track candidates. The naming convention encodes the type of tracks involved: **f** (2D track), **y** (neuro track), **s** (short track) and other features such as an opening angle $> 90\,^\circ$ (**o**) or the back-to-back property (**b**). The most important ones are:

- $\mathbf{t2}_f$: there are $f + 1$ 2D track candidates (with $f \in [0, 1, 2, 3^1]$);
- $\mathbf{t3}_z$: there are $z + 1$ 3D track candidates (with $z \in [1, 2, 3^1]$);

Table 4.1.: Output trigger bits, prescale factors, and logical conditions are specified, where index ranges denote inclusive logical OR operations over the corresponding input signals. For each trigger bit, an additional requirement is imposed: neither the Bhabha veto nor the injection veto may be activated.

trigger bit	prescale	condition
bfyo[1]	50	$t2_{1-3}$ & ty_{0-3} & $cdc_{open,90}$
f	20000	$t2_{0-3}$
ffy	1	$t2_{2-3}$ & ty_{0-3}
fy30	1	$t2_{1-3}$ & ty_{0-3} & f2f30
fyb	1	$t2_{1-3}$ & ty_{0-3} & $b2b_5$
fyo	1	$t2_{1-3}$ & ty_{0-3} & $cdc_{open,90}$
s	4000	ts_{0-3}
ssb	10	ts_{1-3} & s2s5
stt	1	typ
stt6	1000	typ6
syb	1	ts_{0-3} & ty_{0-3} & s2f5
syo	1	ts_{0-3} & ty_{0-3} & s2fo
y	500	ty_{0-3}

- $\mathbf{ty}_y$: there are $y+1$ neuro track candidates (with $y \in [1, 2, 3^1]$);
- $\mathbf{ts}_s$: there are $s+1$ short track candidates (with $s \in [0, 1, 2, 3^1]$);
- $\mathbf{typ}_p$: at least one track that meets the STT conditions[2] with momentum $> p/10$ (where $p \in [4, 5, 6, 7]$, *i.e.* the default bit typ = typ_7 requires $p > 0.7$ GeV/c);

and their respective angular dependencies:

- $\mathbf{cdc}_{\text{open,90}}$: there are two tracks with an opening angle >90 °;
- $\mathbf{f2f}_{30}$: there are two tracks with an opening angle >30 °;
- $\mathbf{b2b}_X$: there are two back-to-back tracks ($\Delta\phi \approx 180°$), within a symmetric tolerance window of $\pm 10 \cdot X°$ (with $X \in [3, 5, 7, 9]$, *i.e.* for $X = 3$, the tolerance is ±30°);
- $\mathbf{s2s}_Y$: there are two back-to-back short tracks ($\Delta\phi \approx 180°$), within a symmetric tolerance window of $\pm 10 \cdot Y°$ (with $Y \in [3, 5, o]$, where "o"=9);
- $\mathbf{s2f}_Z$: there is a pair of back-to-back tracks with one 2D track and one short track ($\Delta\phi \approx 180°$), within a symmetric window of $\pm 10 \cdot Z°$ (with $Z \in [3, 5, o]$, where "o"=9).

4.2.7.2. Trigger output bits

The output trigger bits cover different combinations of input bits under the condition that no injection and no Bhabha veto by the ECL sub-trigger was issued. The output

[1] In the case of $i = 3$, more than 3 tracks are found.

[2] The STT requires a single neural track with an estimated longitudinal impact parameter $|z|$<15 cm and a momentum above a given threshold (typically p>0.7 GeV/c)

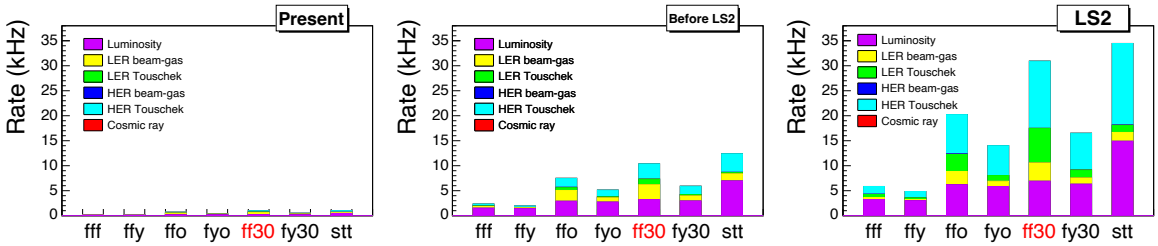


Figure 4.4.: Extrapolation of the L1 trigger rate for CDC stand-alone trigger-bits. An increase in trigger rates, induced by rising beam-induced background levels, is observed. The dominant background contribution arises from luminosity-dependent processes. LER and HER denote the low- and high-energy ring background components, respectively. This figure is adapted from [79].

trigger bits that contribute to the L1 trigger decision are shown in Table 4.1. The naming convention encodes the type of tracks involved: **f** (2D track), **y** (neuro track), **s** (short track) and other features such as an opening angle $> 90\,^{\circ}$ (**o**) or the back-to-back property (**b**).
To prevent the total L1 trigger rate from exceeding the 30 kHz limit, individual trigger bits can be prescaled, *i.e.* only a fraction of $1/n$ of the events satisfying this trigger condition are effectively read out.
Therefore, there are two different stages for the trigger bits: **final trigger decision logic (FTDL)** (before prescaling) and **pre-scaled and masked (PSNM)** (after prescaling).

The STT is the most important minimum-bias CDC trigger bit, since it is the only non-prescaled single track trigger bit. It requires at least one good neuro track that satisfies $|z| < 15\,\text{cm}$ and $p > 0.7$ GeV/c. The momentum cut suppresses QED backgrounds like $e^+e^- \longrightarrow e^+e^-e^+e^-$ peaking at low momenta or Touschek-scattered spallation protons [56]. The STT trigger bit is critical for low-multiplicity physics channels including τ-pair production, certain dark matter signatures, or processes like $e^+e^- \longrightarrow \pi^+\pi^- n\gamma$ ($n \geq 0$). The cross sections for the latter processes are important for better understanding the muon $g-2$ anomaly [56].

4.2.8. Trigger rate extrapolation

The projected CDC stand-alone trigger rates for the non-prescaled trigger bits are presented in Figure 4.4 for three distinct operating scenarios: the "present" configuration, corresponding to experiment 30, run 2211 in early 2024 ($\mathcal{L}_{\text{inst}} = 0.18 \cdot 10^{35}\,\text{cm}^{-2}\text{s}^{-1}$), the anticipated conditions "before LS2" ($\mathcal{L}_{\text{inst}} = 2.8{\cdot}10^{35}\,\text{cm}^{-2}\text{s}^{-1}$), and the post-"LS2" scenario ($\mathcal{L}_{\text{inst}} = 6 \cdot 10^{35}\,\text{cm}^{-2}\text{s}^{-1}$). Each trigger rate is decomposed into its constituent background contributions: The luminosity-correlated component (e.g., Bhabha and two-photon processes), beam-gas scattering in the LER and HER, Touschek scattering in the LER and HER, and the approximately constant cosmic-ray pedestal. In the appendix A.2.2, I de-

[1]The "b" here denotes "without Bhabha veto" taken into account.

scribe the methodology used to quantify individual contributions to the total trigger rate and the procedure used to extrapolate these rates to future operating conditions at SuperKEKB, following the approach detailed in [79].

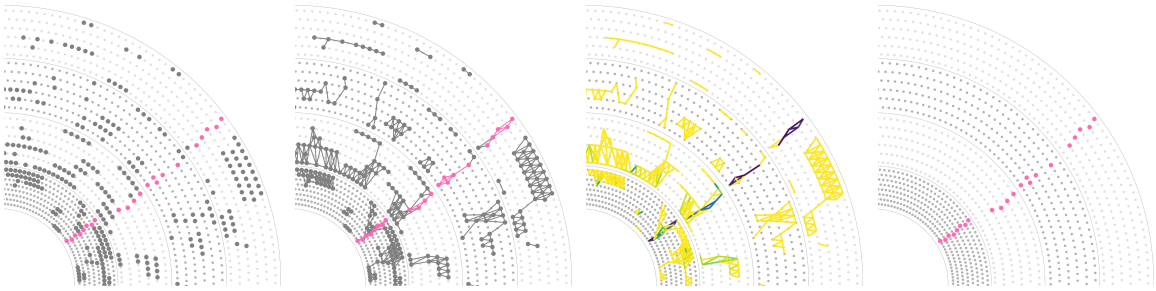
Under the present operating conditions, all CDC trigger bit rates remain well below the 30 kHz threshold. The STT bit constitutes the largest contribution to the overall CDC trigger rate. For example, in the post-LS2 extrapolation, the STT rate alone is expected to reach approximately 35 kHz, and the prescaled `ff30` trigger bit is also projected to exceed the 30 kHz limit.

These projections indicate that operating the STT in its current configuration, without further modifications, is not feasible at LS2 luminosities. A trigger rate approaching or exceeding 30 kHz from a single trigger bit would saturate the available DAQ bandwidth, leaving no margin for additional physics or calibration triggers. Mitigating this issue necessitates either a substantial upgrade of the trigger algorithm, a tightening of the z-vertex and momentum selection criteria, or the introduction of a prescaling of the STT bit, the latter two inevitably resulting in a reduction in physics acceptance.

5. GNN-based hit filtering: the algorithm

Increasing beam-induced background levels and detector aging, as detailed in subsection 3.6.2- section 3.7, pose substantial challenges for the Belle II reconstruction algorithms. In particular, the tracking reconstruction is strongly affected by the increasing background occupancy [45, 60, 61]. To suppress spurious background hits, two alternative filtering strategies are currently applied in the offline track reconstruction: a legacy, cut-based hit filter and a more recent, higher-performing MVA-based filter. Although the latter significantly outperforms the cut-based method, its performance is expected to reach its limits at the even higher background conditions anticipated in future operation.
Consequently, this work investigates a third alternative filtering strategy based on GNNs. GNNs are being extensively studied for tracking applications in essentially all high-energy physics (HEP) experiments [3, 15, 17–21]. In such approaches, the neural network operates on a graph representation of the detector hits, in which hits are represented as graph nodes and edges encode spatial and temporal correlations between hits. This graph-based representation can be exploited not only for the track reconstruction itself but also in a preceding hit-filtering step.
Performing the filtering prior to tracking offers a key advantage, as tracking is computationally expensive and its complexity scales approximately with the square of the number of hits. Eliminating a substantial fraction of hits prior to the tracking stage can therefore, in principle, significantly reduce the computational load and may also improve the overall reconstruction performance, measured, for instance, in terms of the number of correctly reconstructed tracks.
A principal advantage of the GNN approach over the MVA method is that, in the case of the GNN, the neighborhood of each hit is explicitly incorporated into the per-hit filtering decision. Consequently, local patterns can be exploited, which is expected to enhance the classification performance.

The following sections present the fundamental principles of the proposed GNN-based algorithm, which is composed of multiple processing stages, as illustrated in Figure 5.1. In the first step, the relevant CDC hit features are extracted for each event. Subsequently, in a graph construction step, the individual hits are mapped onto a graph representation in which edges encode the spatial and temporal correlations between hits. A GNN is then applied to this graph to infer, for each hit and its associated edges, the probability that the corresponding objects are compatible with a signal-like or background-like origin. In the final stage, the GNN output is used to remove hits that are classified as background-like.
In the following, I provide a description and a default configuration of the GNN filter algorithm, that serves as a baseline for optimization and dedicated design studies discussed in section 8.1.



(a) before filtering (b) graph encoding (c) GNN classification (d) after filtering

Figure 5.1.: Overview of the GNN-based hit filtering algorithm: (a) CDC hits (signal hits in pink, background hits in grey), are encoded as graphs (b) in which edges connect geometrically compatible hits. The GNN carries out classification on edge- or node-level (c) to detect signal-like patterns, with darker colors indicating signal-like and lighter colors indicating background-like predictions. Finally, the classification scores are used for the hit filtering step (d) [4].

5.1. Hit information used for the graphs

The hit-filtering algorithm requires a detailed characterization of each CDC hit in terms of geometric and pulse-shape observables, where the latter encode the deposited energy and timing information of the hit, as described in detail in section 3.4. For each event, the CDC hit data are recorded with the attributes listed in Table 5.1, together with their respective variable ranges.

- identifiers of the hit sense wire (super-layer ID, continuous layer ID, local layer ID, cell ID),
- the transverse coordinates of the wire in the x-y plane (x, y),
- the transverse coordinates of the wire in the r-ϕ plane (r, ϕ),
- pulse-shape-related quantities (TDC, ADC and TOT).

For training purposes, additional truth-level information from the associated truth particle (originating either from Monte Carlo (MC) simulation, baseline track reconstruction, or absent in the case of background hits) is appended, including:

- a unique identifier of the associated particle (object ID),
- a signal-versus-background classification label (binary label).

These truth-level quantities are not used by the algorithm during inference, but they are essential for the training and validation phases.

The transverse wire coordinates, x and y, of the stereo wires depend on the chosen longitudinal coordinate z. Consequently, I define two reference configurations: either the

Table 5.1.: Available hit information. Ranges that are not directly related to geometric quantities represent approximate estimates, provided solely to give an impression of the underlying feature distributions.

Parameter	Range
super-layer ID	0 to 8
layer ID	0 to 56
local layer ID	0 to 5 (7 in first super-layer)
wire cell ID	0 to 384 (different for each super-layer)
x	-111.14 to 111.14 cm
y	-111.14 to 111.14 cm
r	16.78 to 111.14 cm
ADC	0 to ∞ (realistic: up to $\approx$ 35 000)
TDC	4 200 to 5 250
TOT	0 to ∞ (realistic: up to $\approx$ 200)
truth particle ID	0 to ∞ (realistic: up to $\approx$ 50 000)
signal label	0 or 1

transverse wire positions are evaluated at the nominal interaction point, $z = 0$ or at the midpoint of the wires. In the initial implementation, I adopt the latter option, *i.e.* the midpoint of the wire, as the reference working point.

Furthermore, two distinct categories of CDC hit inputs are available. The first option is the `CDCHits` collection, which contains the complete set of recorded hits, including those associated with wires that are subsequently identified and flagged as background-like. The second option is the `CDCWireHit` collection, in which the raw observables have already undergone a preliminary filtering procedure. In this stage, measurements affected by basic cross-talk rejection criteria are removed, entire malfunctioning boards are masked, and, optionally, default hit-selection filters, such as the MVA filter, are applied.
The impact of this pre-filtering in `CDCWireHit` on model training can be either beneficial or detrimental, and it is not known a priori which choice is optimal. To retain the full informational content and delegate the selection of relevant features to the network, `CDCHit` objects are therefore used for model training in the initial design configuration. Identically to the cut-based or MVA filter, the output of the GNN hit filter are `CDCWireHit` objects. The bad-wire masking is performed on these GNN-based hit outputs, while the cross-talk filtering stage remains disabled.

5.2. Graph encoding of hit data

The GNN employed in this work operates on a graph-based representation of the CDC hits, in which the nodes correspond to individual hits and the edges encode the respective relationships between them. The graph construction strategy, in particular the specifica-

Table 5.2.: Initial graph-building configuration parameters.

Parameter	Setting (default)
ADC cuts	$(10, \infty)$
TDC cuts	$(4240, 4980)$
minimal number of CDC hits of track	4
node features	x, y, ADC, TDC
edge features	Δr, $\Delta\phi$, ΔTDC
same layer distances	$[-1, 1]$
next layer distances	$[0]$
next-to-next layer distances	$[-1, 0, 1]$
inter-super-layer edges	`False`
edge direction	uni-directional

tion of the valid connection topology between hits, is based on the methodology described in [1]. The default configuration parameters used for graph construction are summarized in Table 5.2 and will be discussed in detail in the following sections.

5.2.1. Input data preparation

The CDC hits of a given event are preprocessed for subsequent graph construction in the following way:

- **Cut application**: Loose pre-selection criteria based on hit-level observables such as *ADC* and *TDC* (see Table 5.2) are imposed to suppress evidently spurious background hits. These criteria enable the model to concentrate on the most informative and challenging features to classify and as a side effect reduce both the size of the constructed graph and the resulting data volume for all subsequent processing stages.

- **Hit shuffling**: The set of hits is randomly permuted to eliminate potential ordering biases, since hits are typically ordered either by MC generation sequence or by readout location in data.

- **Short track handling**: Hits associated with short tracks, *i.e.* tracks for which the number of corresponding CDC hits falls below a specified threshold (*e.g.* fewer than five hits), are re-labeled as background hits during the network training phase. Such short tracks that do not even traverse a full super-layer, are intrinsically difficult to classify and are unlikely to be reconstructed by subsequent tracking algorithms. Consequently, the neural network is expected to prioritize the more relevant, longer tracks.

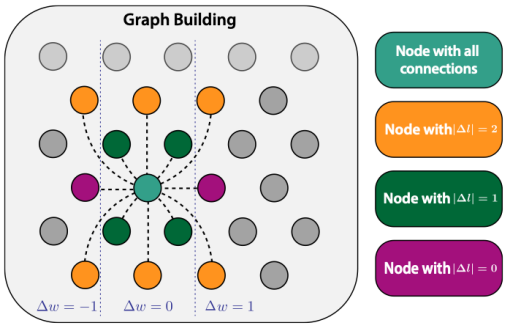


Figure 5.2.: Graph building schematic based on [1]. Figure from Philipp Dorwarth.

5.2.2. Node representation

The graph is defined as a set of nodes and edges, where each CDC hit is represented as a node within this graph. In the initial design, the node feature set comprises the spatial coordinates (x, y), as well as *ADC* and *TDC* values. For training and validation purposes, an additional MC-based signal label is assigned to each node.

5.2.3. Edge construction

The graph edges encode relationships between hits by capturing both spatial and temporal correlations defining edges in the following way:

- **Wire pair generation**: To limit redundant edges, a fully connected topology is avoided and geometric constraints are imposed to define the graph structure following previous studies [1]. The hit distances are defined by the azimuthal distance parallel to the layers Δw, and the radial distance Δl as shown in Figure 5.2. Here, the distance between a hit and its two closest neighbors in the same or next-to-next layer is defined as $\Delta w = \pm 1$, and the distance to the two closest wires in the next layer is defined as $\Delta w = 0$. The graph edges are kept if one of the following conditions is fulfilled: $\Delta w = 1$ for edges within the same super-layer ($\Delta l = 0$), $\Delta w = 0$ for edges between adjacent layers ($\Delta l = 1$), and $\Delta w = 0$ or 1 for edges between next-to-next layers ($\Delta l = 2$). The two immediate neighbors in the same layer are defined by $\Delta w = \pm 1$, and the two closest wires in the alternately shifted adjacent layer by $\Delta w = 0$.

  In principle, the wire distance Δw can be directly inferred from the unique wire identifiers assigned within each layer. For hits occurring in the same layer, this distance is exactly given by the difference of the corresponding wire IDs, provided that the periodic boundary condition between the maximum and minimum wire ID is properly taken into account. For distances involving neighboring layers, an additional shift by ± 1 must be included. However, this procedure is not applicable for graphs that also incorporate inter-super-layer edges, since each super-layer comprises a different number of wires per layer. As a consequence, the difference between two wire IDs (even after correcting for an azimuthal offset) becomes distorted and loses

Table 5.3.: Initial graph pre-processing configuration parameters.

Parameter	Setting (default)
node attribute scales	x: 0.089, y: 0.089, ADC: 0.014, TDC: 0.012
edge attributes scales	Δr: 2.5, $\Delta\phi$: 250, ΔTDC: 0.013
clipping values	-
shifting values	TDC: 4200

a clear geometric interpretation. For this reason, I employ a super-layer-dependent azimuthal distance $\Delta\phi$ as the criterion for selecting valid wire pairs.

- **Edge attribute calculation**: All wire pairs passing the selection are used to build graph edges, each stored as an edge index pair. For each edge, the relational features such as Δr, $\Delta\phi$, and ΔTDC are computed as the edge attribute vector.
- **Edge direction**: Edges can, in principle, be uni-directional, bi-directional, or undirected. The initial implementation uses uni-directional edges.
- **Ground-truth labelling**: For training and validation, each edge is labelled *signal* if both hits come from the same truth object (a MC particle or reconstructed track); otherwise, it is labelled *background*.

The resulting graph objects comprise node feature tensors, edge attribute tensors, edge index representations, ground-truth labels for both edges and nodes, as well as supplementary metadata, such as the event identifier.

5.3. Pre-processing of graphs

Before the GNN can be applied to the graph representation of the CDC hits for inference or training, the graphs must be pre-processed. In machine learning applications, it is generally preferable that all features lie within a comparable order of magnitude. By normalizing feature ranges and removing outliers in a numerically consistent and robust way, both model convergence and predictive performance can be improved.
Consequently, the node and edge features of the input graph are transformed by applying clipping, shifting, and scaling, according to the predefined parameters summarized in Table 5.3. The pre-processing steps comprise:

- **Feature clipping**: Node features are restricted to lie within predefined lower and upper bounds. In the initial configuration, feature clipping is not applied.
- **Feature shifting**: Node features are translated by a predefined offset such that their effective origin is close to zero. This is especially relevant for the TDC feature, whose values begin only at approximately 4 200.
- **Feature scaling**: All features are rescaled such that their magnitudes are normalized to a comparable order of magnitude.

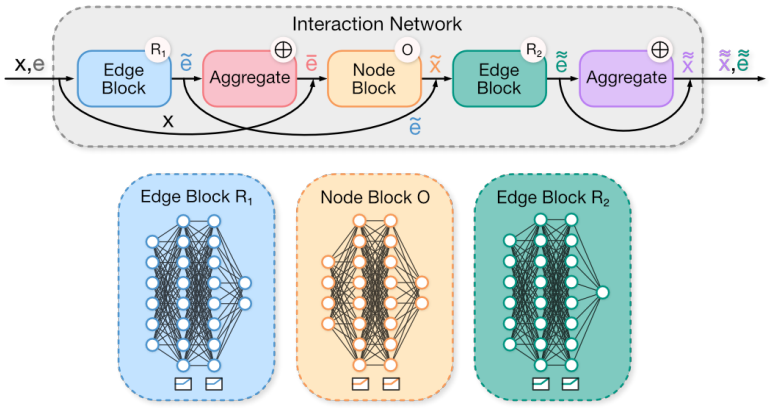


Table 5.4.: Number of trainable parameters

MLP	layer	$n_{weights}$	n_{biases}
R1	0	112	8
R1	1	64	8
R1	2	32	4
O	0	72	8
O	1	64	8
O	2	40	5
R2	0	112	8
R2	1	64	8
R2	2	8	1

Figure 5.3.: Graphical representation of the interaction network architecture [80] with an overview of the number of trainable parameters (Table 5.4). Parameters are given per sub-MLP and parameter type (weight or bias) for a network configuration with two hidden layers with eight hidden neurons per MLP each. The total number of trainable parameters is 626.

Table 5.5.: Initial model configuration parameters.

Parameter	Setting (default)
model architecture	Interaction Network
hidden size	8
hidden depth	2
aggregation 1	sum
aggregation 2	max

5.4. The interaction network

The employed neural network architecture is a modified version of the interaction network (IN) architecture [80], which is a GNN-based model designed to reason about interactions among entities in complex systems. This architecture has demonstrated strong suitability for track reconstruction tasks [15, 17–21]. Moreover, it can be implemented with a relatively small number of trainable parameters, which is particularly advantageous for deployment at the TRG level [27, 29, 81–83], where stringent constraints on network size apply.

The network comprises three consecutive MLP blocks: an edge block R_1 (that updates edge features), a node block O (that updates node features), and a final edge block R_2 (that performs edge-level classification).
For each edge, the first edge block R_1 processes the edge features jointly with the corresponding incoming and outgoing node features $(x_{\text{in}}, x_{\text{out}}, e)$, and produces updated edge representations $\tilde{e}$. These updated edge features are subsequently aggregated at the node

level to yield $\bar{e}$ via a scatter-based aggregation operation, which in the default configuration is implemented as a summation over the responses of all incident edges for each node. The node block O then takes $(\bar{e}, x)$ as input and outputs updated node features $\tilde{x}$. In combination with $\tilde{e}$, these updated node features constitute the input $(\tilde{x}_{\text{in}}, \tilde{x}_{\text{out}}, \tilde{e})$ to the final edge block R_2. The block R_2 predicts one-dimensional edge scores $\tilde{\tilde{e}}$, which are again aggregated per node via a second scatter operation to obtain scalar node-level scores for each hit. A sigmoid activation function is then applied to these scores, yielding edge- and node-wise probabilities $\tilde{\tilde{e}}$ and $\tilde{\tilde{x}} \in [0, 1]$, which are interpreted as the posterior likelihoods that a given edge or node is signal-like (values close to 1) or background-like (values close to 0).

In the initial configuration detailed in Table 5.5, each MLP block consists of two hidden layers with `ReLU` activation [84], each with 8 hidden network nodes, resulting in a total of 626 trainable parameters, as illustrated in Figure 5.3. This highly compact design renders the model computationally efficient: training convergence is achieved with merely $O(1000)$ training events (each consisting of hundreds of CDC hits), and the training process typically requires about 15 minutes on a local CPU. By contrast, more complex architectures with $O(10^4\text{–}10^7)$ parameters generally demand on the order of millions of training samples and training times of several weeks.
This low demand in computational cost and resource consumption is particularly advantageous during algorithm development and optimization, as it facilitates extensive hyper-parameter studies to determine an optimal configuration, as discussed in chapter 8. Moreover, the small number of parameters enables rapid re-training under varying background conditions.
The iterative message-passing procedure, in which each node updates its representation based on its own attributes and those of its neighbors, enables the network to learn and encode local structural patterns present in the data. Therefore, despite its simplicity, the model reliably discriminates background from signal hits across a wide range of investigated background levels, ranging from the low background environment of experiment 22 to the higher anticipated levels of experiment 0, as discussed in section 8.2.

5.5. Hit filtering step

The filtering step constitutes the central component of the hit filtering algorithm. This step utilizes the output scores of the pre-trained GNN model to apply a threshold-based selection. The workflow comprises the following stages:

- **Model inference**: The trained IN is applied to the graph-encoded representation of the event to compute node and edge scores. These scores as illustrated in Figure 5.1c quantify the probability that a given node or edge is signal-like or background-like, respectively.

- **Node & edge selection**: Nodes and edges with a score below a predefined threshold (cut value) are removed from the graph (as shown in Figure 5.1d for an exemplary

Table 5.6.: Initial training hyper-parameter configuration.

Parameter	Setting (default)
validation split	0.125
batch size	1
shuffle	`True`
learning rate	0.01
step size	4
gamma	0.7
loss	binary cross entropy
training target	edges
number of epochs	50
patience	5

cut value of 0.4). To ensure a compact and consistent representation, the surviving nodes are remapped to a new, contiguous index space.

- **Hit filtering**: For each event, all hits in the `CDCWireHit` collection are cross-checked against the selected-hit list. For hits not included in this selected hit list, the background flags are set. In the subsequent tracking stages, these hits are ignored by the reconstruction algorithms. In addition, the GNN output corresponding to each hit is made available as a self-relational weight for every CDC hit in the `CDCHits` object.

5.6. Network training

The IN is trained based on the hyper-parameter configuration specified in Table 5.6. For training, the dataset is divided into a training and a validation subset (defined by the *validation split*), with the latter employed to monitor training performance and to mitigate over-fitting. Training is conducted on batches consisting of single events, each comprising hundreds of hits, and the events are shuffled at the beginning of each epoch to avoid biases arising from any intrinsic file ordering. Treating each event as an individual batch obviates the need for padding or complex batching strategies, while still yielding a large effective sample size due to the high number of nodes and edges per graph.
During each training epoch, a binary cross-entropy loss is computed to account for the binary nature of the classification task. This loss is minimized using the Adam optimizer [85] in conjunction with a step-wise learning rate scheduler. An initially relatively high learning rate of 0.01 provides fast descent toward a low-loss region of the parameter space, while the subsequent step-wise reductions, applied every four epochs by a multiplicative factor of $\gamma = 0.7$, result in smaller parameter updates. This scheduling strategy facilitates fine-grained optimization of the model parameters and helps to stabilize oscillatory behavior in the neighborhood of local minima.
Training is performed for a maximum of 50 epochs with an early-stopping criterion: if the validation loss does not improve for five consecutive epochs, the optimization is termi-

nated. This strategy prevents unnecessary computation after effective convergence and further constrains over-fitting.

6. Datasets

A reliable evaluation of an ML-based algorithm requires training and evaluation datasets that fully cover a wide range of detector operating conditions and physics topologies expected during deployment. For the GNN-based hit-filtering algorithm developed in this work, I accordingly construct a dedicated dataset collection, inspired by previous related studies [3], comprising MC simulations with data-driven beam-background overlays as well as real collision data.

6.1. Monte Carlo simulation

I simulate MC samples using the basf2 framework (`release-10`) where signal particles are first generated with dedicated event generators: single-particle tracks are produced with a particle gun; heavy-flavor hadron decays (predominantly those of B and D mesons) are simulated using `EvtGen`; and inclusive decay final states as well as the continuum production of light quark-antiquark pairs are modeled with PYTHIA8. Subsequently, the transport of the generated particles through the detector volume is simulated with `GEANT4`, after which the detector responses are digitized.

6.1.1. Background overlays

To reproduce realistic detector occupancies, all generated events are combined with beam-induced background overlays [32]. Beam-related backgrounds, as described in detail in section 3.6, can either be incorporated through run-independent simulation or through run-dependent randomly triggered data, and are added at the level of digitized hits:

Simulated beam backgrounds are generated using the SAD accelerator simulation [86]. In this run-independent simulation, beam-particle losses arising from processes such as radiative Bhabha scattering, Touschek scattering, and beam-gas interactions are tracked throughout the accelerator lattice. Whenever a simulated particle deviates from the design orbit and collides with material in the Belle II interaction region, its phase-space coordinates are recorded and subsequently provided as input to the detector simulation. This procedure yields background samples that are merged with physics events by superimposing one or more background events per bunch crossing and performing a joint digitization of signal and background. In this way, pile-up effects are inherently included and correlations among sub-detectors are preserved.

Table 6.1.: Common configuration of the particle-gun samples used for training, specifying the number and type of generated tracks, the azimuthal angle ϕ range, and the transverse momentum p_T; all ranges are drawn from independent uniform distributions. In addition, the samples are enriched with extra Poisson-sampled low-momentum tracks.

Setting	Value
particles	μ^-, μ^+
azimuthal angle ϕ	0 to 360°
number of tracks	1 to 6
transverse momentum p_T	0.05 to 6 GeV/c
number of low-momentum tracks	Poisson (1)
low-momentum $p_{T,low}$	0.05 to 0.4 GeV/c

Data-driven background overlays use run-dependent background events obtained from random triggers. Since random triggered data contain only beam-induced backgrounds and detector noise above the readout thresholds, pile-up contributions from sub-threshold activity can be modeled only approximately and cannot be fully reconstructed.

For my studies, I employ exclusively data-driven background overlays using an internal software tool [2].

6.1.2. Particle gun samples

To suppress biases by specific decay topologies and increase the sensitivity to potential beyond-the-SM signatures, including dark photons, axion-like particles, and magnetic monopoles, I use technical particle-gun samples for the GNN training.
All particle-gun-based samples are generated with a common set of characteristics, as summarized in Table 6.1. The particle-gun events contain up to six μ^- or μ^+ tracks per event, with the track multiplicity drawn from a uniform distribution, and with the transverse momentum p_T of each track uniformly distributed in the range 0.05 to 6 GeV/c.
Muons interact minimally with the detector material and neither shower nor decay within the CDC. This makes them particularly suitable for isolating the intrinsic tracking performance, unaffected by hadronic interactions or electromagnetic energy losses.
In addition, each event is enriched with a random number of low-p_T muon tracks ($p_T \in$ 0.05 to 0.4 GeV/c), drawn from a Poisson distribution with an expected value (mean) of 1. This low-p_T enrichment reflects the topologies of generic $B\bar{B}$ events that comprise a large number of charged-particle tracks with momenta substantially below 1 GeV/c, including low-p_T ("curling") trajectories ($p_T < 0.3$ GeV/c) that are locally nearly orthogonal to the radial direction and can produce up to several hundred detector hits. As discussed in section 8.2, including such low-momentum curlers in the training dataset substantially improves inference performance on $B\bar{B}$ samples. Their presence prevents the network from learning that only radially outward-propagating track segments are signal-like and

from systematically classifying all other topologies as background.

I define several categories characterized by distinct ranges in polar angle θ, transverse displacement from the interaction point, longitudinal z-shift, and pointing angle α_{point}, as listed in Table 6.2. These categories are grouped as follows:

1-3) Prompt muons: originating at the IP, generated for the forward, barrel, and backward regions of the detector, respectively.

4-6) Displaced muons: that share the same angular coverage as Categories 1-3 but contain tracks with production vertices displaced from the IP. These displaced topologies improve the model's ability to identify hits from non-prompt tracks.

7) Displaced & non-pointing muons: tracks that do not intersect the interaction point but are rotated by an angle α_{point}.

8) z-shifted muons: prompt barrel muons with an additional longitudinal z-shift, introducing controlled variations along the beam axis to test the model's robustness to such shifts.

9) Displaced muon vertices: non-pointing vertices artificially formed by pairs of particle-gun tracks with a common, displaced vertex assigned a non-zero pointing angle.

6.1.3. KKMC and EvtGen-based physics samples

In addition to the particle-gun samples, I use physics-motivated samples to train and evaluate the algorithm on realistic physics topologies.

10) Di-muon events: I use `KKMC`-generated $\mu^+\mu^-(\gamma)$ samples to train and evaluate the model in clean, low-multiplicity conditions. The samples include non-radiative and radiative topologies. In the non-radiative case, the two muons are back-to-back in the center-of-mass frame, and non-curling, with momenta around 5 GeV/c. Radiative $\mu^+\mu^-(\gamma)$ events provide a wider range of track momenta and opening angles, as photon emission prevents the muons from being back-to-back. Together, these samples provide a clean reference for assessing hit- and track-level reconstruction performance.

11-12) Generic B-events: To evaluate tracking in high-multiplicity environments, I use `EvtGen`-generated B-meson pair samples (separated into $B^0\bar{B}^0$ and B^+B^-). These events typically contain ten or more charged CDC tracks, spanning a wide range of particle species (electrons, muons, pions, kaons, protons, and their respective antiparticles) and momenta from a few tens of MeV/c up to several GeV/c.
The $B\bar{B}$ samples serve two purposes. First, they provide physics-motivated evaluation samples to test track reconstruction in the environment most relevant for the

Table 6.2.: Overview of technical and physical samples used in this work: **(top)** Description of the particle-gun training samples used in this work for the different topology categories in addition to the common parameters listed in Table 6.1. **(middle)** Description of physics MC simulated samples at $\sqrt{s}$ =10.58 GeV [32]. **(bottom)** Data-driven samples selected by different HLT skims, including special debug runs with waveform readout, which are used to validate reconstruction and trigger performance under realistic running conditions.

Particle gun samples

category	type	θ	displacement	z-shift	α_{point}
1	prompt forward	17° to 35.4°	0 cm (IP)	0 cm	0°
2	prompt barrel	35.4° to 123.0°	0 cm (IP)	0 cm	0°
3	prompt backward	123.0° to 150.0°	0 cm (IP)	0 cm	0°
4	displaced forward	17° to 35.4°	0 to 100 cm	0 cm	0°
5	displaced barrel	35.4° to 123.0°	0 to 100 cm	0 cm	0°
6	displaced backward	123.0° to 150.0°	0 to 100 cm	0 cm	0°
7	angled displaced	17.0° to 150.0°	0 to 100 cm	0 cm	0 to 30°
8	prompt z-shift	35.4° to 123.0°	0 cm (IP)	−10 to 10 cm	0°
9	displaced vertex	17.0° to 150.0°	0 to 100 cm	0 cm	0 to 10°

MC generator samples

category	type	process	generator
10	di-muon	$\mu^+\mu^-$ (γ)	`KKMC`
11	mixed B-events	$B^0\bar{B}^0$	`EvtGen`
12	charged B-events	B^+B^-	`EvtGen`
13	pions (from K_S^0)	$B \to XK_S^0(\to \pi^+\pi^-)$	`EvtGen`
14	pions/protons (from Λ)	$B \to X\Lambda(\to \pi^- p)$	`EvtGen`

Data samples

category	type	HLT skim	selection criteria
15	di-muon	mumu tight	see section 6.2
16	background only	delayed Bhabha	50 µs after Bhabha trigger
17	waveform	debug	special waveform runs

Belle II program. Second, they are included in the training dataset for realistic track multiplicities and particle-type mixtures. This enables the network to learn patterns characteristic for hadronic events, such as overlapping tracks, secondary vertices, and non-trivial kinematic correlations without bias toward specific decay channels due to the wide variety of accessible final states.

13-14) K_S^0 and Λ decays: To probe displaced track signatures, I use long-lived decays $B \to X K_S^0 (\to \pi^+\pi^-)$ and $B \to X\Lambda (\to \pi^- p)$, which have decay vertices displaced inside the CDC. Final-state pion and proton tracks are selected by requiring that their MC mother particles are identified as either K_S^0 or Λ.

6.2. Data-driven samples

Beyond simulated datasets, I use recorded Belle II data for training and validation under realistic detector and trigger conditions. These samples are selected online by the HLT applying the default basf2 event reconstruction.

15) HLT-selected di-muon events (`mumuskim`): Like the simulated $\mu^+\mu^-(\gamma)$ samples, these offer a clean, low-multiplicity reference of prompt, high-momentum tracks. The HLT selection[1] requires two oppositely charged, nearly back-to-back tracks in the center-of-mass frame, each with momentum above 0.4 GeV/c and an associated ECL cluster below 0.5 GeV. The total reconstructed ECL energy, including photon clusters, must be less than 2 GeV to efficiently suppress Bhabha and hadronic backgrounds. This skim is primarily optimized for the selection of events containing two muon tracks. However, it can also accommodate the inclusion of low-energetic radiative photons.

16) Background only (delayed Bhabha) events: In addition to the run-dependent background overlays used in the MC simulation, I also use the raw data from which these overlays are produced. This ensures that the dataset behaves like other unprocessed samples, without modifications from MC event simulation. The random HLT event selection is based on the delayed Bhabha trigger bit, issued 50 µs after the primary Bhabha trigger, corresponding to five revolutions of the accelerator ring after issuing the Bhabha trigger bit at the IP.

17) Waveform events: During standard physics data-taking and in MC simulations, only processed observables of the hit waveforms, namely the integrated 25-sample ADC sum, the leading-edge TDC value, and the TOT, are recorded, while digitized waveforms are discarded to limit data volume [87]. Dedicated diagnostic runs at nominal and reduced gas gain (*e.g.* , experiment 30 runs 1729-1730[2], experiment 33 runs 699-700 and 702[1], and experiment 37 runs 1303-1306 and 1310) use full waveform

[1]HLT selection definition: `https://gitlab.desy.de/belle2/software/basf2/-/blob/main/hlt/softwaretrigger/calculations/src/SkimSampleCalculator.cc#L606-607`

[2]Trigger operation database entry: `https://elog.belle2.org/elog/TRG+operation/936`

Table 6.3.: Overview of used data and simulated run conditions with information taken from [62].

Experiment	Run	Date	$n_{\mu,\text{events}}$	Peak inst. luminosity (in $10^{34}\,\text{cm}^{-2}\text{s}^{-1}$)	$\langle n \rangle_{\text{extraCDChits}}$
Data					
22	26	30.11.2021	313237	2.40	285
26	1894	21.06.2022	52679	3.77	1219
35	2894	27.12.2024	18953	3.84	1751
37	1893	21.12.2025	6325	3.05	1951
Simulated					
0	0	future	∞	60.0	2800

readout under representative beam conditions, with systematically varied CDC high voltage and front-end thresholds. These runs enable detailed studies of gain variations, cross-talk, and hit information at the waveform level, which are then used to validate and optimize the CDC response.

For this work, raw waveform data are required for L1 trigger studies since the TRG processes only the sum of 3 of the 25 sampled ADC points. The 3-point sum is neither modeled nor stored in standard runs, so any algorithm intended to operate on it must be designed and evaluated using this reduced 3-point information. The 25-point sum is planned to be made available to the L1 trigger in the future, but for initial detector tests it is essential to first understand the currently used 3-point sum in detail. For my studies, I employ waveform data acquired during experiment 37, run 1304[2], recorded on 4th of December 2025.

6.3. Experiment conditions

I employ several representative background and luminosity configurations for training and validation, utilizing either the Belle II data directly or simulated samples with background overlays, as detailed in subsection 6.1.1. The experimental conditions correspond to individual Belle II runs between 2021 and 2025 or to simulated future conditions, as summarized in Table 6.3. To ensure that a given run condition provides a large enough sample size for both training and validation, the number of muon events, $n_{\mu,\text{events}}$, is used to monitor the effective size of each run-specific dataset, noting that the total recorded events far exceed those containing muons. Beam-induced occupancy is characterized by the mean number of extra CDC hits per event $\langle n \rangle_{\text{extraCDChits}}$.

As an early-run configuration with moderate background, I use experiment 22, run 26[1]

[1]Trigger operation database entry: `https://elog.belle2.org/elog/TRG+operation/1006`

[2]Data path on KEKCC: `/group/belle2/TMP/Data/Raw/e0037/r01304/sub00/debug.0037.01304.HLT01.m01.f00000.root`

(November 2021). It serves as a low-occupancy reference sample for training and validation with a peak instantaneous luminosity of $2.4{\times}10^{34}$ cm^{-2}s^{-1} and about $\langle n\rangle_{\text{extraCDChits}}$=300. Experiment 26, run 1894[2] (June 2022) corresponds to the late phase of Run I, with a peak luminosity of 3.77×10^{34} cm^{-2}s^{-1} and roughly $\langle n\rangle_{\text{extraCDChits}}$=1 200. I use it as the main medium-background benchmark.

To probe background levels representative of the current operational conditions, I use data from experiments 35 and 37. Experiment 35, run 2894[3] (December 2024), in the early phase of Run II, reached a peak instantaneous luminosity of 3.84×10^{34} cm^{-2}s^{-1} with about $\langle n\rangle_{\text{extraCDChits}}$=1 750. Experiment 37, run 1893[4] (December 2025), corresponds to a higher background level, with about $\langle n\rangle_{\text{extraCDChits}}$=2 200 at a peak luminosity of 3.05×10^{34} cm^{-2}s^{-1}. The higher number of extra CDC hits in this latter run despite the lower instantaneous luminosity is most plausibly explained by detector aging, as discussed in section 3.7.

A projected configuration for a hypothetical future operating scenario (experiment 0, run 0[5]) is derived from simulated beam-background overlay samples tuned to a peak instantaneous luminosity of 6.0×10^{35} cm^{-2}s^{-1}, yielding about 2 800 extra CDC hits per event. It is used to extrapolate tracking and hit-filtering performance to the design luminosity.

[1] Background overlay paths on KEKCC: `/group/belle2/dataprod/BGOverlay/BGOrd/rel8/BGOExp22rel8/release-08-00-04/e0022/4S/r00026/beambg/sub00/`

[2] `/group/belle2/dataprod/BGOverlay/BGOrd/rel8/BGOExp26rel8/release-08-00-04/e0026/4S/r01894/beambg/sub00/`

[3] `/group/belle2/dataprod/BGOverlay/BGOrd/rel8/BGOExp35rel8/release-08-02-05/e0035/4S/r02894/beambg/sub00/`

[4] `/group/belle2/dataprod/Calibration/Bucket43/BGO/`

[5] `/group/belle2/dataprod/BGOverlay/nominal_phase3/prerelease-08-00-00a/overlay/BGx1`

7. Metrics

In this chapter, I introduce and define the hit-, track-, and trigger-bit-level metrics used throughout this work. These metrics are used to quantify the performance of the GNN-based filter algorithm in suppressing background hits and to assess how this filtering propagates to higher-level reconstruction entities, such as tracks and trigger decisions.

7.1. Definition of signal particle tracks

First, I identify signal particle trajectories as the reference set for all hit-level and track-level performance metrics. For MC-simulated events, I consider the MC particle trajectories as the *true tracks*. For collision data, by contrast, no generator-level truth information is available. To construct an effective proxy for the ground truth in the performance evaluation, I apply the complete basf2 offline reconstruction chain to the recorded events and treat the resulting reconstructed tracks as an operational approximation of the *true tracks* in the context of TRG studies. Although the reconstruction efficiency of the offline algorithm is not ideal, it can still be regarded as a ground truth reference, since the trigger-level reconstruction is not required to outperform the more detailed offline reconstruction.

In addition, I impose a set of selection criteria on these true particle tracks:

- The particle must be either primary-like or a daughter of a selected long-lived mother (K_S^0 or Λ). For true tracks derived from offline reconstruction, "primary-like" is defined as tracks originating in the vicinity of the interaction point, characterized by moderate impact parameters: $|d_0| < 15\,\text{cm}$ and $|z_0| < 20\,\text{cm}$.

- The true track must produce at least a minimum number of CDC hits (default: 7), ensuring that only reconstructible tracks with sufficient CDC information enter the evaluation.

- Optionally, the track must satisfy lower bounds on the transverse momentum p_T and the total momentum p.

Particles that fulfill all of these requirements are classified as *eligible particles*; all others are regarded as *bad* and are excluded from the signal definition in both hit-level and track-level metrics.

7.2. Hit metrics

For each CDC hit, its association with eligible and bad tracks, as defined in section 7.1, is determined. If the hit is associated with at least one eligible signal particle, it is classified as a *signal hit*. If it is associated exclusively with bad tracks, it is classified as a *bad hit*. Hits without any association to a true track are classified as *fake hits*.

For a given filtering method (cut-based, MVA, or GNN), each hit is either retained or rejected through the assignment of a background flag. Combining the assigned background flag with the signal classification of the hits, I define and compute the following hit-level performance metrics:

- **Hit efficiency**

$$\epsilon_{\text{hit}} = \frac{k_{\text{true, hit}}}{n_{\text{true, hit}}}, \quad \epsilon_{\text{hit, per track}} = \sum_{\text{tracks i}} \frac{k_{\text{true, hit, } i}}{n_{\text{true, hit, } i}},$$

  denotes the fraction of true signal hits $n_{\text{true, hit}}$ that are retained after the hit filtering, $k_{\text{true, hit}}$. The quantity can be evaluated globally over all hits, or computed on a per-track basis and subsequently averaged. It characterizes the capability of a given filter to preserve hits associated with genuine signal tracks.

- **Hit purity**

$$P_{\text{hit}} = \frac{k_{\text{true, hit}}}{k_{\text{hit}}}, \quad P_{\text{hit, per track}} = \sum_{\text{tracks i}} \frac{k_{\text{true, hit, } i}}{k_{\text{hit, } i}},$$

  represents the fraction of selected (i.e. retained) hits k_{hit} that correspond to true signal hits. As for the efficiency, this quantity can be evaluated over all hits or averaged on a per-track basis; in the latter case, only background hits that are actually assigned to tracks are taken into account. Hit purity quantifies the level of noise from background hits in the selected sample after filtering.

- **Hit fake rate**

$$F_{\text{fake, hit}} = \frac{k_{\text{fake, hit}}}{k_{\text{hit}}},$$

  is defined as the fraction of selected hits that are classified as fake hits $k_{\text{fake, hit}}$. It measures how often the filter incorrectly keeps hits that are not associated with any eligible signal particle tracks and thus directly complements the purity metric.

- **Hit bad rate**

$$F_{\text{bad, hit}} = \frac{k_{\text{bad, hit}}}{k_{\text{hit}}},$$

  is defined as the fraction of selected hits that are identified as bad hits $k_{\text{bad, hit}}$. In contrast to pure fake hits, these hits are associated with particles that fail the signal track selection, so this rate is particularly relevant for understanding noise from mis-selected or out-of-acceptance tracks.

- **Hit background rejection**

$$\mathrm{rej}_{\mathrm{bkg,\,hit}} = \frac{n_{\mathrm{fake,\,hit}} - k_{\mathrm{fake,\,hit}}}{n_{\mathrm{fake,\,hit}}},$$

  quantifies the fraction of background hits $n_{\mathrm{fake,\,hit}}$ that are successfully removed by the filter. A high background rejection indicates that the filter is effective at suppressing spurious hits, thereby reducing combinatorics in the subsequent tracking stages. The background rejection associated with a discrete hit efficiency, *e.g.* at 90 % is denoted as $\mathrm{rej}_{\mathrm{bkg,\,hit,\,90\%eff}}$.

- **Extra CDC hits**

$$\langle n \rangle_{\mathrm{extraCDChits}},$$

  denotes the number of hits that are not associated with any reconstructed track and are therefore predominantly background-like. In other words, this is the number of hits remaining unassigned after tracking. This metric serves as an initial indicator of the effectiveness of the filter in suppressing background noise.

Statistical uncertainties for ratio-like metrics are obtained by treating the numerator and denominator as binomially distributed counts. The central values are computed as k/n, as defined above, and the associated asymmetric uncertainties are derived from the Wilson score confidence interval [88] and reported in the form $\mathrm{value}^{\Delta\mathrm{up}}_{\Delta\mathrm{down}}$, where Δup and Δdown denote the upper and lower confidence bounds, respectively.

In addition to the scalar metrics defined at the nominal working point of each filter, *i.e.* , after applying the GNN classification cut, I calculate receiver operating characteristic (ROC) curves based on the continuous GNN output score:

- The true positive rate (TPR) and false positive rate (FPR) as functions of the score threshold are computed using `scikit-learn` [89].
- The area under the curve (AUC) is then evaluated as

$$\mathrm{AUC} = \int \mathrm{TPR}(\mathrm{FPR})\, d\mathrm{FPR},$$

  providing a threshold-independent scalar metric of the hit-level classification performance.

Uncertainties on the ROC curves and AUC values are estimated by repeating the network training multiple times and computing the corresponding ensemble of ROC curves. All ROC curves are interpolated onto a common grid of thresholds. At each grid point, I compute the mean TPR and FPR, and use the standard deviation across models as an estimate of the model-to-model spread.
The AUC is primarily employed to quantify the intrinsic discrimination power of the GNN model during development and optimization, enabling a comparative assessment of different GNN architectures and configurations. The scalar metrics, by contrast, characterize the performance at the fixed working point adopted in the main evaluation studies.

7.3. Track segment metrics

Analogously to the hit-level metrics introduced in section 7.2, I categorize the reconstructed TSs used in the TRG pipeline according to their relationship to the corresponding truth signal tracks into three distinct classes: *signal*, *bad*, and *fake* TSs.
However, in contrast to the CDC hits, TSs are not raw detector objects but already the output of a reconstruction step. In the standard TS building, segments are formed only if they meet predefined criteria on the minimum number of hits per TS, as detailed in subsection 4.2.1. In other words, TSs comprising hits in fewer than four distinct detector layers are not constructed, even in cases where the associated hits originate from signal tracks. By construction, any efficiency evaluated solely on the already existing TSs would therefore trivially amount to 100 %. To obtain a well-defined reference "truth" sample for evaluating TS performance metrics, I therefore employ an alternative TS builder configuration that requires only a single CDC hit per TS. In this way, every potential TS that could, in principle, be formed from the underlying hits is explicitly constructed.

As for the hit-level metrics I define the following metrics

- **TS efficiency**

  $$\epsilon_{\text{TS}} = \frac{k_{\text{true, TS}}}{n_{\text{true, TS}}},$$

  quantifies the fraction of hypothetical true TSs, $n_{\text{true, TS}}$, that are correctly reconstructed by the TSF, $k_{\text{true, TS}}$.

- **TS purity**

  $$P_{\text{TS}} = \frac{k_{\text{true, TS}}}{k_{\text{TS}}},$$

  represents the fraction of the built TSs k_{TS}, that correspond to true signal TSs.

- **TS fake rate**

  $$F_{\text{fake, TS}} = \frac{k_{\text{fake, TS}}}{k_{\text{TS}}},$$

  denotes the fraction of all built TSs that are classified as fake TSs, $k_{\text{fake, TS}}$.

- **TS bad rate**

  $$F_{\text{bad, TS}} = \frac{k_{\text{bad, TS}}}{k_{\text{TS}}},$$

  is defined as the fraction of all built TSs that are identified as bad TSs, $k_{\text{bad, TS}}$.

- **TS background rejection**

  $$\text{rej}_{\text{bkg, TS}} = \frac{n_{\text{fake, TS}} - k_{\text{fake, TS}}}{n_{\text{fake, TS}}},$$

  characterizes the fraction of hypothetical background TSs, $n_{\text{fake, TS}}$, that are successfully rejected by the TSF.

7.4. Track metrics

In direct analogy to the hit-level metrics, I define track-level performance metrics with respect to the signal particle tracks introduced in section 7.1:

- **Track finding efficiency**

  $$\epsilon_{\text{found}} = \frac{k_{\text{true, found}}}{n_{\text{true, track}}},$$

  denotes the fraction of true signal tracks, $n_{\text{true, track}}$, that are successfully identified by the track finding algorithm, $k_{\text{true, found}}$.

- **Track fitting charge efficiency**

  $$\epsilon_{\text{fitted}} = \frac{k_{\text{true, fitted}}}{n_{\text{true, fitted}}},$$

  (or short track fitting efficiency in this work) quantifies the fraction of true signal tracks, $n_{\text{true, fitted}}$, for which a reconstructed track with correctly assigned charge, $k_{\text{true, fitted}}$, is obtained.

- **Track purity**

  $$P_{\text{fitted}} = \frac{k_{\text{true, fitted}}}{k_{\text{fitted}}},$$

  gives the fraction of all reconstructed tracks, k_{fitted}, that are correctly matched to true signal tracks.

- **Track fake rate**

  $$F_{\text{fake, fitted}} = \frac{k_{\text{fake, fitted}}}{k_{\text{fitted}}},$$

  is the fraction of reconstructed tracks that cannot be associated with any signal track, $k_{\text{fake, fitted}}$. This rate quantifies the contribution from purely spurious tracks originating from background or mis-reconstruction.

- **Track bad rate**

  $$F_{\text{bad, fitted}} = \frac{k_{\text{bad, fitted}}}{k_{\text{fitted}}},$$

  is defined as the fraction of reconstructed tracks that are associated with "bad" signal tracks, $k_{\text{bad, fitted}}$, e.g. out-of-acceptance tracks or tracks otherwise not considered part of the target signal.

- **Track clone rate**

  $$F_{\text{clone, fitted}} = \frac{k_{\text{clone, fitted}}}{k_{\text{fitted}}},$$

  measures the fraction of reconstructed tracks that are identified as clones, $k_{\text{clone, fitted}}$, i.e. cases in which more than one reconstructed track is associated with the same true signal track.

- **Track fβ score**

$$f_{\beta,\text{fitted}} = \frac{(1+\beta^2)\, k_{\text{true, fitted}}}{(1+\beta^2)\, k_{\text{true, fitted}} + \beta^2 \left(n_{\text{true, fitted}} - k_{\text{true, fitted}}\right) + k_{\text{fake, fitted}}},$$

  provides a combined measure of the reconstructed track efficiency and purity. The parameter $\beta > 1$ *e.g.* f_2 assigns a larger weight to efficiency, whereas $\beta < 1$ emphasizes purity. This scalar metric is particularly useful for comparing the overall performance of different reconstruction or filtering configurations.

For the matching between reconstructed and true tracks in the offline tracking performance evaluation, I employ the official basf2 definitions of "isSignal" and "isClone". In the context of the TRG application, I instead adopt a criterion based on a minimum hit purity and hit efficiency of 40 % for the match between reconstructed and true tracks. In other words, a reconstructed track is considered to be associated with a given true track only if at least 40 % of the hits on the true track are recovered by the reconstructed track, and at least 40 % of the hits on the reconstructed track originate from that true track. The default track matching configuration in the TRG simulation employs a threshold of 10 % for both quantities; however, this looser requirement results in a large number of spurious associations, particularly under high-background conditions.

In addition to the classification-oriented performance metrics, I also evaluate the **resolution** of key track parameters after track reconstruction.
For each reconstructed track that can be unambiguously associated with a corresponding true signal track, I define the residual for a generic track parameter x as

$$\Delta x = \begin{cases} \dfrac{x - x^{\text{MC}}}{x^{\text{MC}}} & \text{for } x \in \{p_T, p_z\}, \\ x - x^{\text{MC}} & \text{for } x = z_0, \end{cases}$$

i.e. using relative residuals for momentum-like parameters and absolute residuals for the longitudinal impact parameter.
From the distribution of Δx, I determine the median value

$$m_x = \text{median}(\Delta x)$$

and quantify the resolution using the symmetric 68 % interval around the median, defined as

$$r_{68,x} = \text{P}_{68}\left(|\Delta x - m_x|\right),$$

such that 68 % of the residuals lie within $\pm r_{68,x}$ of the median. To mitigate an efficiency-induced bias in the resolution determination, namely, the situation in which one filter retains *e.g.* more low-p_T tracks than another, potentially leading to an apparent degradation of the measured resolution, I evaluate the resolution exclusively on tracks that are reconstructed both by the configuration under study and by the baseline configuration (the MVA filter in the case of offline track reconstruction, and the default 3D Hough algorithm in the TRG case).

7.4.1. Trigger bit metrics

To assess the performance of the trigger algorithm at the level of individual trigger bits, I introduce a set of event-level metrics that quantify how effectively a given trigger bit selects signal events while suppressing background contributions. In contrast to the hit-, TS- and track-based metrics, which are defined for individual detector objects, the trigger bit metrics are evaluated on an event-by-event basis.

For each trigger bit, events are categorized according to their track topology and the corresponding trigger-bit requirements: An event is classified as a *good event* if it contains the required number of signal particles and, for trigger bits with angular constraints, if the opening angles between the relevant tracks satisfy the criteria defined in subsection 4.2.7. Conversely, an event is classified as a *bad event* if it is associated exclusively with particles that fail the signal selection. All remaining events are categorized as *fake events*.

On the basis of these event classes, I define the following trigger-bit performance metrics:

- **Trigger bit efficiency**

$$\epsilon_{\text{trgbit}} = \frac{k_{\text{true, trgbit}}}{n_{\text{true, trgbit}}},$$

  which quantifies the fraction of good signal events, $n_{\text{true, trgbit}}$, that are correctly accepted by the trigger bit, $k_{\text{true, trgbit}}$.

- **Trigger bit fake rate**

$$f_{\text{fake, trgbit}} = \frac{k_{\text{fake, trgbit}}}{k_{\text{trgbit}}},$$

  denotes the fraction of all triggered events that are not associated with any truth particles satisfying either the signal or bad event criteria, $k_{\text{fake, trgbit}}$. These events are purely background-induced triggers and represent the dominant source of unwanted trigger activity.

- **Trigger bit bad rate**

$$f_{\text{bad, trgbit}} = \frac{k_{\text{bad, trgbit}}}{k_{\text{trgbit}}},$$

  is defined as the fraction of all triggered events, k_{trgbit}, that are identified as bad events, $k_{\text{bad, trgbit}}$, *i.e.* events associated with out-of-acceptance or otherwise non-eligible particles.

- **Trigger bit rate**

$$R_{\text{trgbit}} = \frac{k_{\text{trgbit}}}{n} \cdot \begin{cases} \dfrac{1}{\Delta t} & \text{for background-only samples,} \\ \sigma \cdot \mathcal{L}_{\text{inst}} & \text{for physics processes,} \end{cases},$$

which is estimated by scaling the fraction of events accepted by the trigger bit, k_{trgbit}/n, by the expected abundance of the corresponding process. For pure background samples, the rate is determined by the coincidence window $\Delta t = 100\,\text{ns}$. For physics processes, the event abundance is determined by the production cross-section σ, scaled by the instantaneous luminosity $\mathcal{L}_{\text{inst}}$ of the data-taking conditions.

In addition to the metrics for individual trigger bits, I also report the inclusive CDC trigger performance $\bigvee$ CDC, obtained from the logical OR of all CDC trigger bits and the total L1 trigger signal issued by the combination of all sub-detectors after prescaling and masking (PSNM). The trigger-bit metrics are evaluated separately for the FTDL and PSNM outputs of each trigger bit.

7.4.2. Number of bit operations

In order to analyze and compare the computational complexity of a given neural network configuration, I estimate the number of bit operations (BOPs) per network layer, following [90]. An MLP is defined by the number of its weights N_{w}, its biases N_{b}, its inputs N_{in} and the corresponding bit widths b_{w}, b_{b}, and b_{in}. The number of bit operations per network layer is then given by

$$\text{BOPs}_{\text{layer}} = N_{\text{w}} b_{\text{w}} b_{\text{in}} + N_{\text{b}} b_{\text{b}} + N_{\text{w}} b_{\text{acc}}, \tag{7.1}$$

accounting for an accumulator width of $b_{\text{acc}} = b_{\text{in}} + b_{\text{w}} + \log_2 N_{\text{in}}$. The BOPs for an MLP is the sum over its network layers $\text{BOPs}_{\text{MLP}} = \sum_{\text{layer}} \text{BOPs}_{\text{layer}}$.
For an interaction network model comprising the three MLPs R_1, R_2 and O operating on a graph with N_{nodes} nodes and N_{edges} edges, the total number of multiply-accumulate (MAC) operations becomes

$$\text{BOPs} = N_{\text{edges}} \cdot (\text{BOPs}_{R_1} + \text{BOPs}_{R_2}) + N_{\text{nodes}} \cdot (\text{BOPs}_{O} + \text{BOPs}_{\text{aggr}}). \tag{7.2}$$

The BOP metric calculations in this work assume a graph dimension with $N_{\text{nodes}} = 820$ and $N_{\text{edges}} = 3593$. This value is based on the assumption of a CDC segmentation into 20 sectors, with the largest sector corresponding to the quoted values under the additional assumption that only the L1 trigger layers are considered, *i.e.* , only five out of all layers in each super-layer.

8. Offline GNN-based hit filtering

In this chapter, I optimize and evaluate the GNN-based hit filtering algorithm in the context of offline track reconstruction on a common benchmark dataset comprising two distinct physics samples: $e^+e^- \to \mu^+\mu^-(\gamma)$ and $e^+e^- \to B^0\bar{B}^0$ events.
The $\mu^+\mu^-(\gamma)$ sample (category 10 in Table 6.2) provides a clean topology with few high-quality tracks and low combinatorial complexity. In contrast, the $B^0\bar{B}^0$ sample (category 11 in Table 6.2) probes a complementary multi-track environment with higher hit occupancy and diverse decay topologies. Both samples are derived from Belle II MC simulation with an intermediate background level corresponding to experiment 26, run 1894, as given in Table 6.3.
I perform the analysis using the basf2 framework employing release-10 track reconstruction configured to use exclusively CDC-based information, without incorporating measurements from any additional tracking sub-detectors. The impact of design choices on the performance of the downstream track fitting is quantified using the track fitting f_2 score defined in section 7.4. Only tracks generated with at least seven hits in the CDC were considered for evaluation.
In the following, I compare the filtering performance among three approaches:

1. the legacy cut-based filter used before `release-09`,
2. the MVA-based filter included in the current default reconstruction, and
3. the GNN-based filter developed in this work.

All three filtering algorithms are implemented within the `WireHitPreparer` module (see subsection 4.1.1) and are applied prior to the track finding stage, marking hits as background for the subsequent tracking algorithms.

8.1. Algorithm design optimization

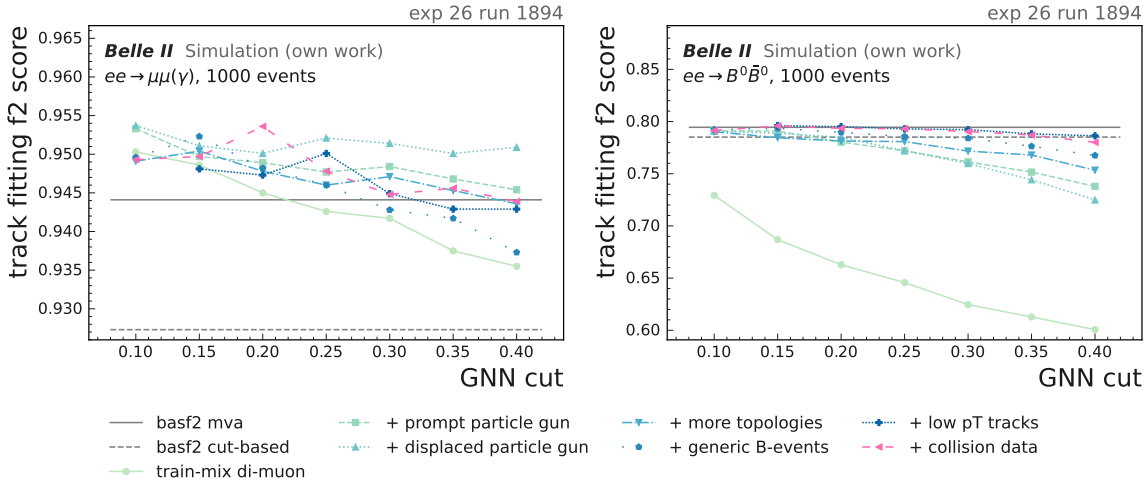
This section details the optimization of the GNN-based hit-filtering algorithm, building upon the baseline design introduced in chapter 5. During the optimization several key components are refined: the composition of the training dataset, the selection and preprocessing of input features, the graph-construction methodology, the training strategy, the neural network architecture, and the selection of the operational working point. The final configuration for each parameter optimization step that is used in the subsequent steps is indicated in pink.
To consistently compare design choices, all configurations are tested on the common benchmark, whereas design choices are based primarily on the $B^0\bar{B}^0$ sample, while the $\mu^+\mu^-(\gamma)$ sample is used in parallel for monitoring.

For each configuration, the f_2 score is evaluated on the same set of 1 000 events per sample type, comprising over a million hits and $\mathcal{O}(10^3 - 10^4)$ tracks. This choice represents a compromise between statistical precision and the computational cost of running the full tracking and fitting chain during development. The results discussed below therefore show small but non-negligible fluctuations of 0.2 to 0.5 %pt, driven by the limited sample size of 1 000 events per trial and stochastic variations in model training. Error bars are omitted to increase visual clarity. A quantitative discussion of the statistical fluctuations and error bars is given in the appendix A.1.1. To mitigate the impact of unstable or failed trainings, I train three models per configuration and use the best-performing one for downstream tracking and evaluation.
Using the same benchmark samples for all configurations induces correlations, but allows a controlled comparison of how algorithmic changes affect track-level performance in both the low-multiplicity $\mu^+\mu^-(\gamma)$ and the more complex $B^0\overline{B}^0$ case.

8.1.1. Sample composition for training

A central design choice for the GNN-based hit filtering concerns the composition of the training dataset. In the following, the composition of the training sample is incrementally extended from a minimal initial configuration containing only $\mu^+\mu^-(\gamma)$ samples (category 10 in Table 6.2), to a composition that more closely reflects the full range of Belle II event topologies in $\Upsilon(4S)$.
To mitigate potential biases arising from variations in the number of events used to construct each sample composition, a fixed number of 1 400 training events is used for each configuration. Since each event consists of an average number of $> 1\,200$ hits in the CDC corresponding to roughly 2 400 edges per event and therefore ~ 3.4 million edges in total, this provides a sufficiently large sample size to incorporate different event signature types and achieve convergence in training. Increasing the number of train samples is further discussed in subsection 8.1.10.
In Figure 8.1, the f_2 scores as a function of the GNN-cut threshold are presented for the different compositions of the training sample using the configuration of the baseline algorithm as introduced in chapter 5 and are compared to the performance of the two legacy hit filters in basf2 (MVA and the cut-based legacy filter). The GNN-cut is defined as the threshold applied to the GNN classification score associated with each hit. For example, for a cut value of 0.2, all hits with a classification score below 0.2 are rejected, while all hits with a score greater than or equal to this threshold are retained.

The initial configuration, trained exclusively on $\mu^+\mu^-(\gamma)$ events, exhibits better performance in terms of the f_2 score evaluated on $\mu^+\mu^-(\gamma)$ events relative to both basf2 baseline configurations for GNN selection thresholds below 0.2. Nevertheless, when this configuration is applied to $B^0\overline{B}^0$ events, which encompass a substantially wider range of track topologies, it becomes evident that the network fails to capture all relevant topological features. This deficiency leads to markedly degraded performance in comparison to the baseline configurations.
As an initial extension, I incorporate *prompt particle-gun* samples (categories 1–3 in Table 6.2). These additional samples enlarge the kinematic phase-space coverage in both



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.1.: Track fitting f_2 score as a function of the GNN-cut threshold for different training sample compositions. The initial composition (train-mix di-muon) exclusively contains simulated $\mu^+\mu^-(\gamma)$ samples used for the network training. The sample composition is successively extended by adding simulated prompt particle gun tracks, displaced particle gun tracks, more particle gun topologies (displaced tracks, displaced z_0 tracks and artificial vertices), generic B events ($B^0\bar{B}^0$ and B^+B^-), low p_T track enrichment, and Belle II collision data (HLT-selected $\mu^+\mu^-(\gamma)$). All sample categories are described in detail in Table 6.2. The different configurations are evaluated on the same 1 000 MC simulated $\mu^+\mu^-(\gamma)$ events (a) and $B^0\bar{B}^0$ events (b) for experiment 26, run 1894 conditions, and compared to the legacy basf2 MVA- and cut-based hit filters.

momentum and polar angle, thereby enabling the GNN to learn a more generalizable mapping between hit patterns and track-like structures, rather than being restricted to the specific $\mu^+\mu^-(\gamma)$ topology. This modification alone significantly improves the performance on $B^0\bar{B}^0$ events, yielding results that surpass the cut-based selection for cut values <0.2, although they remain below those of the MVA filter. Furthermore, the performance on $\mu^+\mu^-(\gamma)$ events is improved as well, with the f_2 score now exceeding the MVA baseline for cut values up to 0.4.

In a subsequent step, I further augment the training mixture with *displaced particle-gun* samples (categories 4–6), in which tracks emulate particles produced by long-lived states or secondary decays occurring at displaced vertices relative to the nominal interaction point. Although robustness with respect to displaced vertices is particularly important in the context of track finding tasks such as those considered in [3], adding this sample category further improves the performance on both benchmark channels.

To explore even more complex and challenging configurations, I introduce an additional *more topologies* category (categories 7–9). These additional particle-gun samples comprise displaced and non-pointing trajectories, artificially generated vertices, and prompt tracks shifted along the z-axis, thereby populating an extended region of phase space. This

extension is specifically designed to probe tracks that originate from vertices that are substantially displaced along the beam axis, and pathological configurations in which tracks traverse the CDC under atypical incident angles, for example, due to cosmic muons or beam-induced background particles. Achieving high-resolution reconstruction of such background-induced tracks is essential, as it enables their efficient suppression through dedicated selection criteria in subsequent stages of the analysis. This addition induces a slight reduction in performance on $\mu^+\mu^-(\gamma)$ samples, while producing an improvement in the f_2 score evaluated on $B^0\bar{B}^0$ events.
In a subsequent step, I incorporate generic $B\bar{B}$ *samples* (categories 11-12) into the training sample mixture. These samples introduce physics-driven multi-track final states, higher hit occupancies, and a mixture of prompt and displaced tracks from B decays, thereby making the sample mixture more representative of the conditions encountered in physics data collection at the $\Upsilon(4S)$ resonance. Once again, performance on the $B^0\bar{B}^0$ benchmark is improved, while it is slightly reduced on the $\mu^+\mu^-(\gamma)$ sample.
Finally, a *low enrichment p_T is* applied to all particle-gun samples to compensate for the otherwise sparse population of low-momentum tracks, which are particularly challenging for track reconstruction in the presence of strong magnetic bending and multiple scattering. This extension further improves the f_2 score for $B^0\bar{B}^0$ events, achieving a performance level comparable to that of the MVA filter, while inducing a marginally negative impact on $\mu^+\mu^-(\gamma)$.
The final stage of the composition extension incorporates $e^+e^- \to \mu^+\mu^-(\gamma)$ selected by the Belle II HLT from collision data (category 14), embedding realistic operational conditions into the training sample. From a performance perspective, including this sample does not further improve the track fitting f_2 score on the MC simulated benchmark samples. However, in principle it is beneficial for the GNN to be trained on the detector conditions expected in operation, even if the nominal simulated performance does not immediately improve, as this reduces the risk of simulation-to-data mis-modeling when applying the filter to collision data. Therefore, I keep this sample category in the training sample composition for all subsequent analyzes.
In the following, all models are trained on the final sample composition, which comprises sample categories 1-12 and 15 as specified in Table 6.2.

8.1.2. Input feature selection

In the baseline design, a minimal set of input features is utilized. Specifically, the node features consist of the hit coordinates x and y together with the measured ADC and TDC values, while the edge features comprise Δr, $\Delta\phi$ and ΔTDC.
Motivated by the hypothesis that more detailed information on the position of the hit, the context of the detector, and the shape of the pulse might increase the discrimination between signal and background hits, I extend the feature set to include the complete set of available hit-related variables for both nodes and edges (see Table 5.1). In particular, the cylindrical coordinates (r, ϕ), the time-over-threshold (TOT), the ADC/TOT ratio, and the discrete indices encoding the *super-layer*, (continuous or local) *layer*, and *wire* number provide additional geometric context and pulse-shape information.

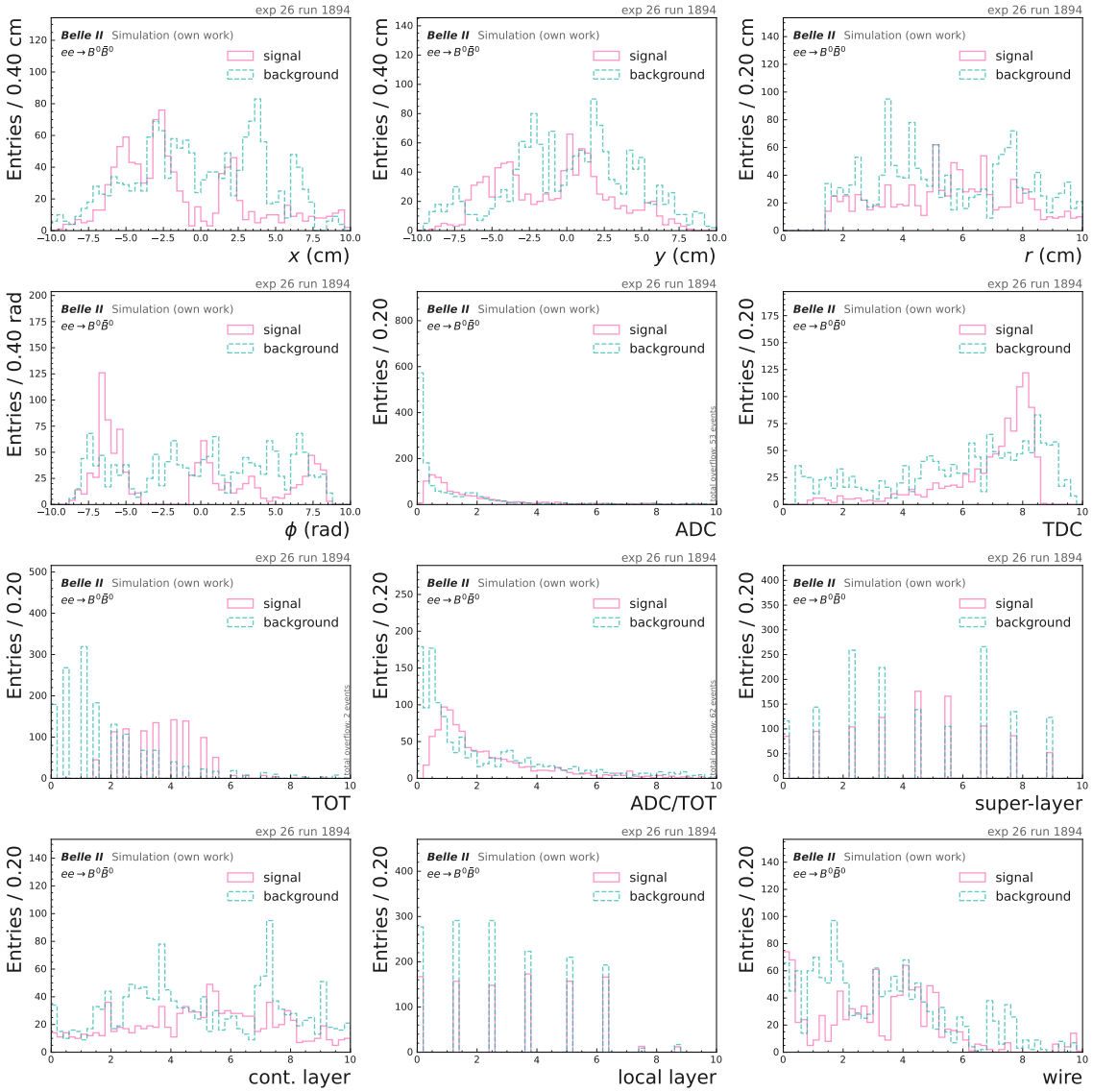


Figure 8.2.: Distributions of all available node input features of MC simulated $B^0\bar{B}^0$ samples (experiment 26, run 1894) for signal and background hits after normalization to the range [-10,10], without additional selection cuts.

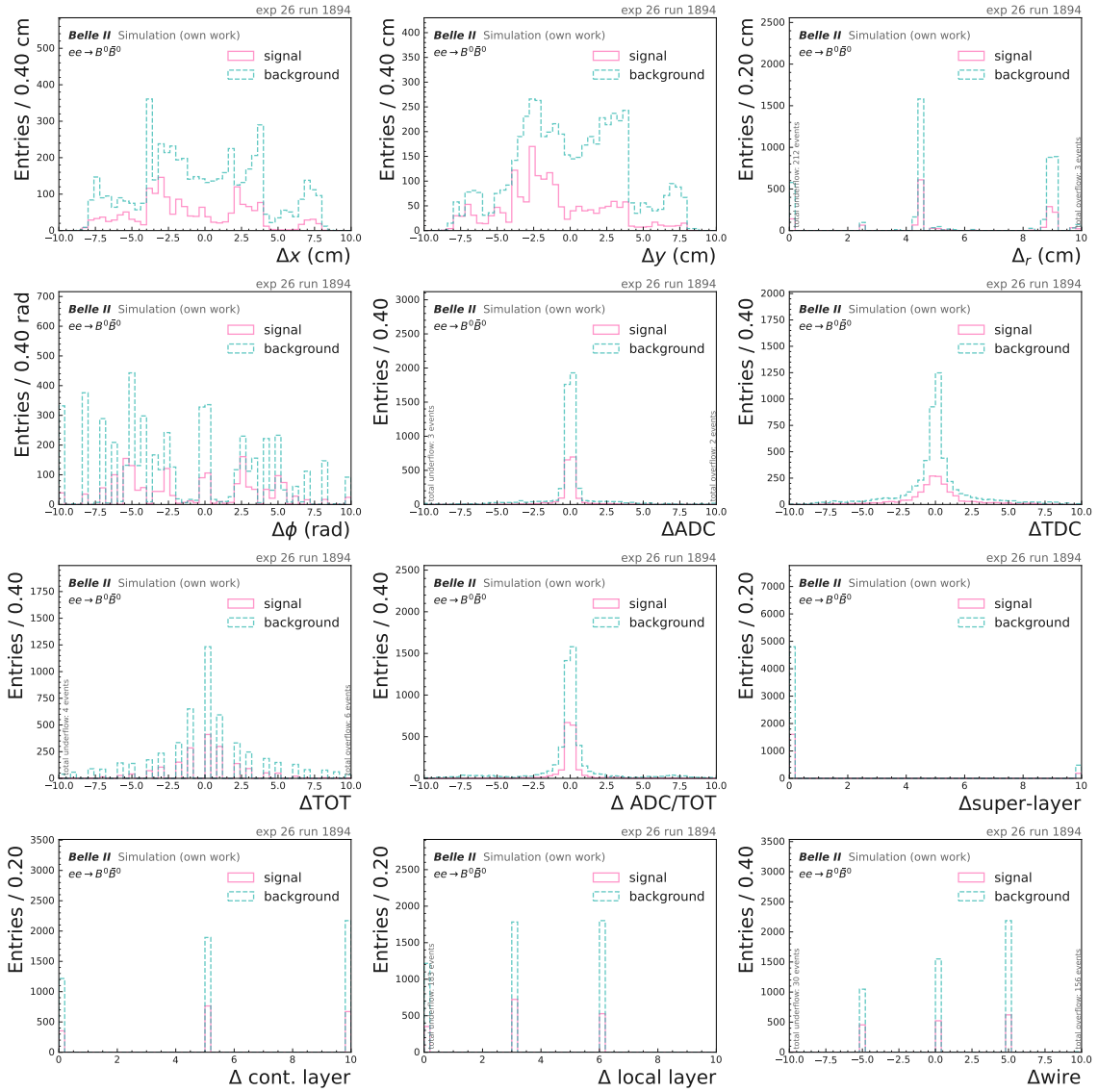
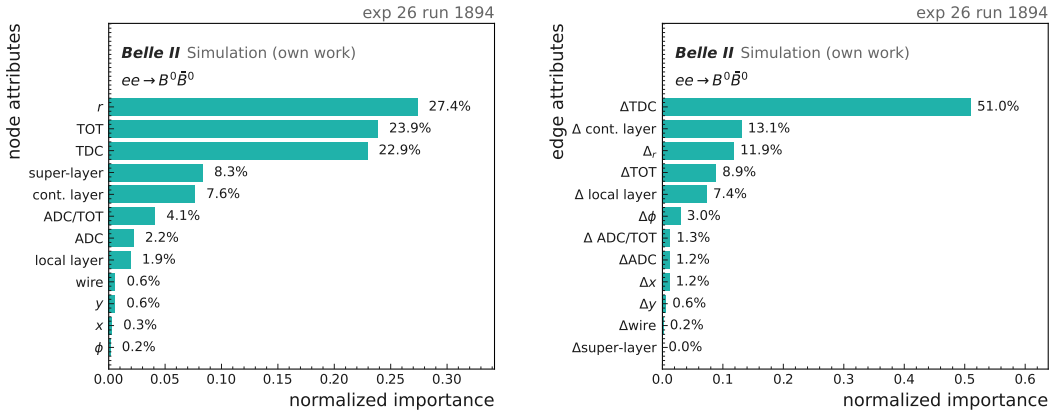


Figure 8.3.: Distributions of all available edge input features of MC simulated $B^0\bar{B}^0$ samples (experiment 26, run 1894) for signal and background hits after normalization to the range [-10,10], without additional selection cuts.



(a) node feature importance

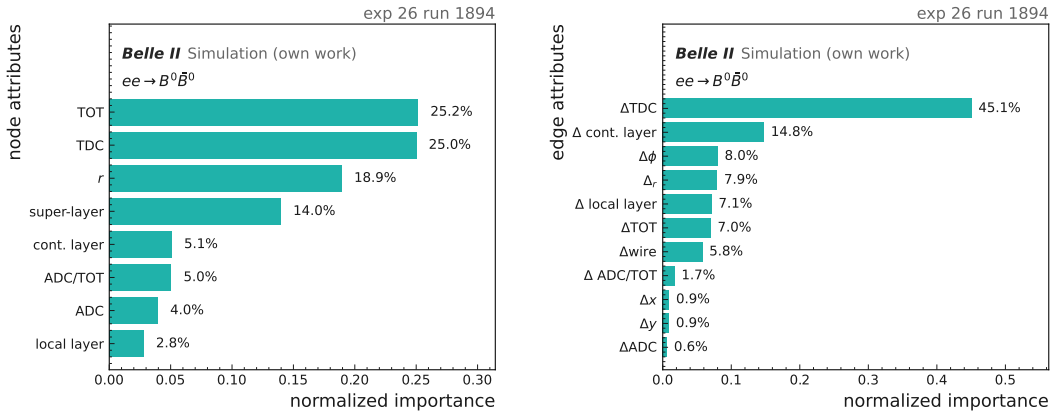
(b) edge feature importance

Figure 8.4.: Normalized node (a) and edge (b) feature importance obtained from a model trained with the full input feature set. During inference the specific features are masked to determine their individual importance. The displayed feature importance is evaluated on the $B^0\bar{B}^0$ benchmark sample for 1 000 events with a background condition corresponding to experiment 26, run 1894.

The underlying distributions of these inputs for signal and background hits from $B^0\bar{B}^0$ events are presented in Figure 8.2 and Figure 8.3 for node and edge features, respectively. The features are displayed after normalization to the common interval $[-10, 10]$, without the application of any additional selection criteria. These distributions demonstrate that, for the majority of features, the signal and background populations exhibit a pronounced separation in their respective distributions. This behavior motivates their selection as candidate input variables to the GNN classifier, which aims to learn the underlying multi-dimensional feature distribution in order to discriminate signal from background hits.

To avoid an unnecessary increase in model complexity and to isolate informative variables, I perform an iterative feature-selection study, based on feature importance inferred from a model trained on the complete feature set. The feature importance as illustrated in Figure 8.4 is evaluated by successively masking individual features at inference time in order to quantify the model's dependence on each input.

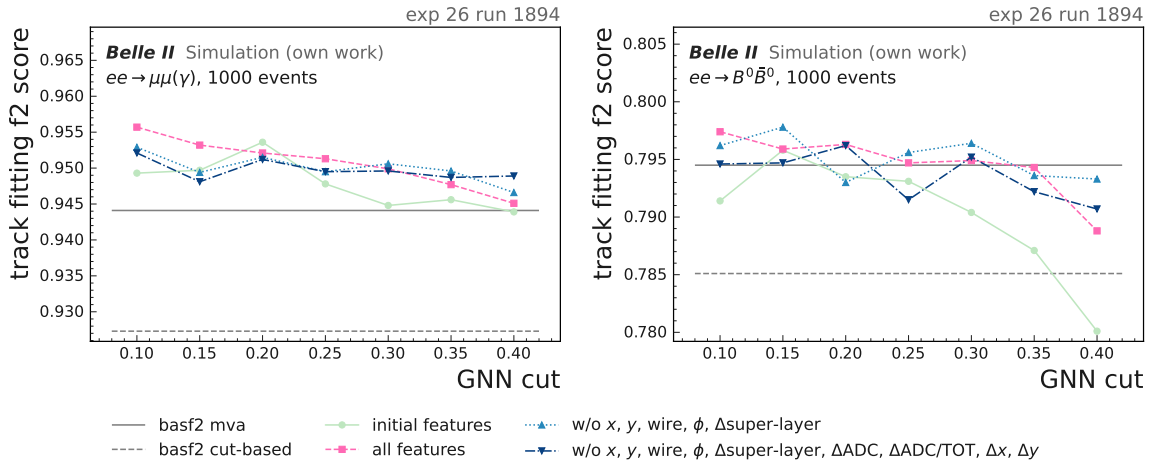
The feature importance scores suggest that at the node level the most influential variable is the time difference ΔTDC between two connected hits, followed by the hit time TDC, the time-over-threshold (TOT), and the radial distance from the IP r. However, these findings must be interpreted with caution, as the importance might exhibit substantial fluctuations across different trained model instances and inference samples. A quantitative assessment of this fluctuation was not conducted within the scope of the present study. Nevertheless, this feature importance information provides a qualitative indication of the relative relevance of the features, particularly with respect to identifying the least informative ones.



(a) node feature importance

(b) edge feature importance

Figure 8.5.: Normalized feature importance obtained from models trained with the input feature set after removing all features with importance below 1 %.



(a) evaluated on $\mu^+\mu^-(\gamma)$

(b) evaluated on $B^0\bar{B}^0$

Figure 8.6.: The track fitting f_2 score is only marginally affected by the removal of a subset of hit features during both training and inference. The features are iteratively eliminated in two pruning stages, guided by their respective feature-importance scores. For all subsequent analyses, the configuration including the full set of features is employed as the baseline.

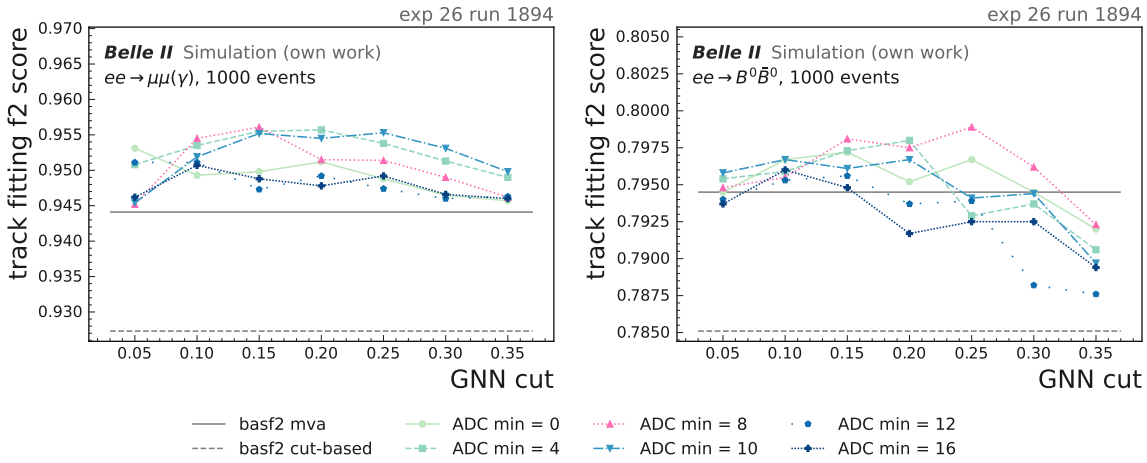
In an initial feature-pruning step, I remove all features with a normalized importance below 1% from both the node and edge representations, and retrain the model subsequently. The resulting importance distributions, shown in Figure 8.5, are then used to perform a second, more stringent feature-pruning step, in which all remaining features with an importance below 2% are discarded.
For each of these reduced feature sets, the track fitting f_2 score as a function of the GNN-cut threshold is evaluated on the common benchmark channels as shown in Figure 8.6. Although feature-pruned models exhibit broadly similar behavior, the configuration that includes all available features achieves the best overall track-level performance in both test samples. Simultaneously, the associated increase in model complexity remains moderate: the number of trainable parameters increases from 626 in the minimal feature configuration to 1 233 when employing the full feature set, without inducing any significant change in inference time in the studied setup. Therefore, I adopt the full-feature configuration as the default choice for all subsequent optimization steps. The edge feature Δsuper-layer exhibits a value of 0 in the current configuration, as no edges between distinct super-layers are utilized at this stage. However, this feature is retained to accommodate extensions of the model in which such inter-super-layer edges will be incorporated as discussed in subsection 8.1.6.

8.1.3. Pre-selection cuts

The pre-selection cuts on ADC, TDC, and TOT (see subsection 5.2.1) applied prior to the GNN inference directly affect the available hits used in training and inference, consequently, the track finding efficiency. To quantify and optimize these selections, I first assess their direct impact on the hit efficiency and hit background rejection, as defined in section 7.2, in the absence of any further hit filtering. The corresponding results are summarized in Table 8.1. The highlighted rows denote the working points selected based on the subsequent cut-optimization studies.
The resulting track fitting f_2 scores as a function of the GNN score threshold are presented in Figure 8.7-Figure 8.12 for the lower and upper ADC, TDC, and TOT cuts, respectively. Each scan is performed sequentially, using the cut configuration optimized in the preceding step as its baseline, such that later optimization choices implicitly incorporate all earlier ones.
The lower ADC threshold governs the rejection of very small pulses, which are frequently attributable to electronic noise or background hits. As shown in Table 8.1, increasing the cut value $\mathrm{ADC}_{\mathrm{min}}$ from 0 to 16 progressively improves the hit background rejection from approximately 54 % to 57 %, at the cost of a modest reduction in hit efficiency by 1.34 %pt. The corresponding f_2-score scan in Figure 8.7 exhibits only minor variations over this range for both sample types, with a slight deterioration for the most restrictive thresholds $\mathrm{ADC}_{\mathrm{min}} > 10$. The relatively loose working point $\mathrm{ADC}_{\mathrm{min}} = 8$ is adopted, as it maintains the hit efficiency at 98.75 % while still providing 54.59 % background suppression on hit-level.
The upper ADC threshold might reject very large pulses that can arise from channel saturation, cross-talk, or signal pile-up. In Table 8.1, a scan of $\mathrm{ADC}_{\mathrm{max}}$ values between 400 and 2 000 demonstrates that tightening this selection improves hit background rejection up to

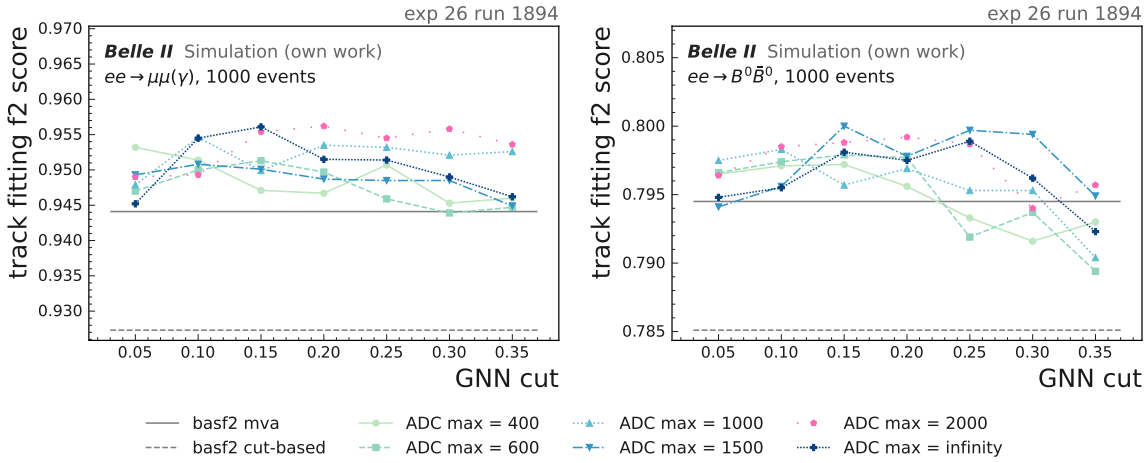
Table 8.1.: Impact of varying the lower and upper preselection thresholds for ADC, TDC, and TOT on hit efficiency and background rejection. Initial and final values for each type are shown, with each threshold type using the final configuration of the previous one. Metrics are evaluated on the $B^0\bar{B}^0$ sample for experiment 26, run 1894. The final configuration yields a hit efficiency of 98.56 % and a background rejection of 56.66 %, *i.e.* a much higher efficiency than the MVA filter but with substantially lower background rejection. At this stage, the goal is to retain as many true hits as possible and leave most background suppression to the subsequent GNN-based filter.

Cut	hit efficiency	hit background rejection
basf2 cut-based	$95.45^{+0.09}_{-0.09}$	$59.55^{+0.13}_{-0.13}$
basf2 mva	$93.89^{+0.10}_{-0.11}$	$76.26^{+0.11}_{-0.11}$
ADC min = 0	$98.78^{+0.05}_{-0.05}$	$54.32^{+0.13}_{-0.13}$
ADC min = 4	$98.78^{+0.05}_{-0.05}$	$54.32^{+0.13}_{-0.13}$
ADC min = 8 (final)	$98.75^{+0.05}_{-0.05}$	$54.59^{+0.13}_{-0.13}$
ADC min = 10 (initial)	$98.67^{+0.05}_{-0.05}$	$55.07^{+0.13}_{-0.13}$
ADC min = 12	$98.51^{+0.05}_{-0.05}$	$55.67^{+0.13}_{-0.13}$
ADC min = 16	$97.47^{+0.07}_{-0.07}$	$56.88^{+0.13}_{-0.13}$
ADC max = 400	$96.95^{+0.07}_{-0.08}$	$63.95^{+0.13}_{-0.13}$
ADC max = 600	$97.54^{+0.07}_{-0.07}$	$62.25^{+0.13}_{-0.13}$
ADC max = 1000	$98.12^{+0.06}_{-0.06}$	$59.85^{+0.13}_{-0.13}$
ADC max = 1500	$98.43^{+0.05}_{-0.06}$	$57.80^{+0.13}_{-0.13}$
ADC max = 2000 (final)	$98.56^{+0.05}_{-0.05}$	$56.66^{+0.13}_{-0.13}$
ADC max = ∞ (initial)	$98.75^{+0.05}_{-0.05}$	$54.59^{+0.13}_{-0.13}$
TDC min = 0 (final)	$98.58^{+0.05}_{-0.05}$	$56.62^{+0.13}_{-0.13}$
TDC min = 4200	$98.58^{+0.05}_{-0.05}$	$56.62^{+0.13}_{-0.13}$
TDC min = 4240	$98.56^{+0.05}_{-0.05}$	$56.66^{+0.13}_{-0.13}$
TDC min = 4280 (initial)	$97.95^{+0.06}_{-0.06}$	$57.07^{+0.13}_{-0.13}$
TDC min = 4320	$96.78^{+0.08}_{-0.08}$	$58.12^{+0.13}_{-0.13}$
TDC min = 4450	$92.36^{+0.12}_{-0.12}$	$62.54^{+0.13}_{-0.13}$
TDC max = 4950	$98.53^{+0.05}_{-0.05}$	$58.77^{+0.13}_{-0.13}$
TDC max = 4980 (final, initial)	$98.58^{+0.05}_{-0.05}$	$56.62^{+0.13}_{-0.13}$
TDC max = 5010	$98.63^{+0.05}_{-0.05}$	$54.99^{+0.13}_{-0.13}$
TDC max = 5040	$98.63^{+0.05}_{-0.05}$	$54.85^{+0.13}_{-0.13}$
TDC max = 5070	$98.63^{+0.05}_{-0.05}$	$54.85^{+0.13}_{-0.13}$
TDC max = ∞	$98.63^{+0.05}_{-0.05}$	$54.85^{+0.13}_{-0.13}$
TOT min = 0	$98.64^{+0.05}_{-0.05}$	$38.77^{+0.13}_{-0.13}$
TOT min = 1	$98.64^{+0.05}_{-0.05}$	$38.77^{+0.13}_{-0.13}$
TOT min = 2	$98.62^{+0.05}_{-0.05}$	$45.15^{+0.13}_{-0.13}$
TOT min = 3 (final, initial)	$98.56^{+0.05}_{-0.05}$	$56.66^{+0.13}_{-0.13}$
TOT max = 15	$98.22^{+0.06}_{-0.06}$	$57.53^{+0.13}_{-0.13}$
TOT max = 20	$98.52^{+0.05}_{-0.05}$	$56.77^{+0.13}_{-0.13}$
TOT max = 25	$98.56^{+0.05}_{-0.05}$	$56.67^{+0.13}_{-0.13}$
TOT max = ∞ (final, initial)	$98.56^{+0.05}_{-0.05}$	$56.66^{+0.13}_{-0.13}$



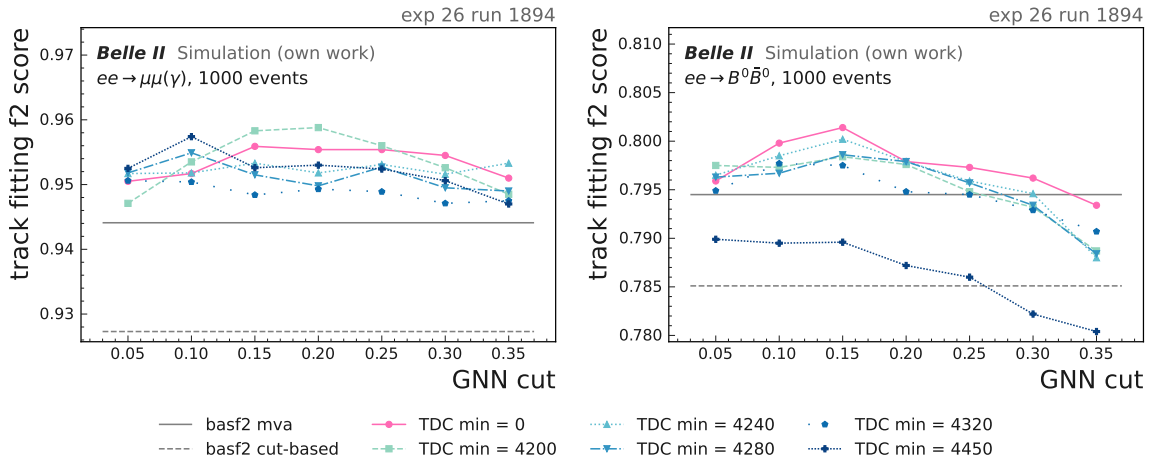
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.7.: Track fitting f_2 score for different $\mathrm{ADC}_{\mathrm{min}}$ cuts. The pre-selection cut $\mathrm{ADC}_{\mathrm{min}} = 8$ is chosen for subsequent studies.



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.8.: Track fitting f_2 score for different $\mathrm{ADC}_{\mathrm{max}}$ cuts. The pre-selection cut $\mathrm{ADC}_{\mathrm{max}} = 2000$ is chosen for subsequent studies.



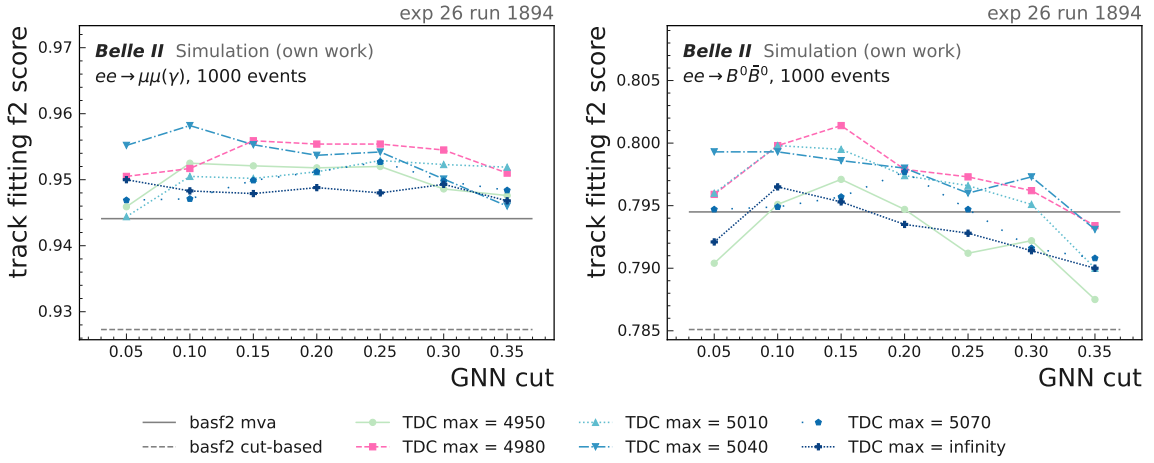
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.9.: Track fitting f_2 score for different $\mathrm{TDC}_{\mathrm{min}}$ cuts. Increasing the threshold value leads to a decrease in the f_2 score, in particular for the evaluation on the $B^0\bar{B}^0$ sample.

approximately 64 %, albeit at the cost of a significant reduction in hit efficiency down to 96.95 % for the most restrictive thresholds $\mathrm{ADC}_{\mathrm{max}} < 1\,000$. The corresponding performance metrics are presented in Figure 8.8, where very stringent cuts ($\mathrm{ADC}_{\mathrm{max}} \lesssim 1000$) induce a modest decrease in f_2, while intermediate to loose thresholds yield nearly indistinguishable behavior. Consequently, I adopt $\mathrm{ADC}_{\mathrm{max}} = 2000$ as a conservative working point that suppresses only the most extreme outliers while preserving potentially informative signal hits, particularly in high-occupancy conditions.

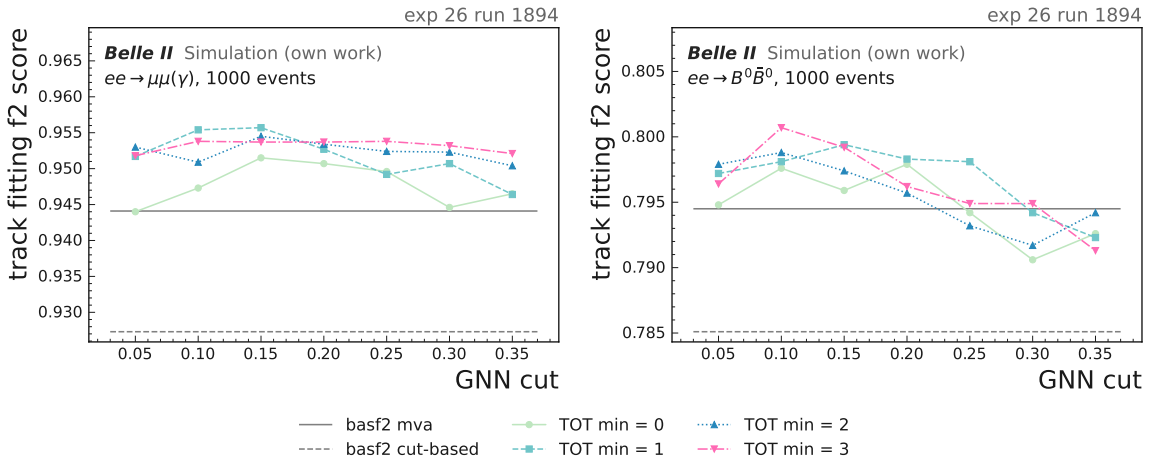
The lower TDC threshold is primarily intended to reject very late hits that lie outside the physically plausible drift-time window. Table 8.1 shows that increasing $\mathrm{TDC}_{\mathrm{min}}$ from 0 to approximately 4240 has no effect on either efficiency or background rejection. Larger cuts improve hit-level background rejection by up to several percentage points while in particular the most stringent setting at 4 450 produces a substantial efficiency loss to 93.36 %. The f_2 curves in Figure 8.9 indicate that a moderate tightening of $\mathrm{TDC}_{\mathrm{min}}$ is either neutral with respect to, or slightly harmful for, track-level performance, and that the most restrictive value $\mathrm{TDC}_{\mathrm{min}} = 4\,450$ significantly degrades f_2, particularly in the $B^0\bar{B}^0$ sample down to $f_2 = 0.78$. On this basis, $\mathrm{TDC}_{\mathrm{min}} = 0$ is adopted, *i.e.* no explicit lower cut is applied, and the GNN is instead relied upon to suppress late-time background hits by exploiting their full feature context.

The upper TDC bound eliminates hits that occur unphysically early with respect to the expected drift-time window, which are most likely from electronics noise or background hits. As summarized in Table 8.1, tightening $\mathrm{TDC}_{\mathrm{max}}$ from ∞ to approximately 4980 or 4950 improves hit background rejection from about 55% to nearly 59%, while maintaining the hit efficiency above 98.5%. The corresponding f_2 scores shown in Figure 8.10 exhibit a optimum at $\mathrm{TDC}_{\mathrm{max}} = 4980$, beyond which further tightening does not provide additional benefit and eventually degrades performance. On this basis, $\mathrm{TDC}_{\mathrm{max}} = 4980$ is chosen



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.10.: Track fitting f_2 score for different $\mathrm{TDC}_{\mathrm{max}}$ cuts. The pre-selection cut $\mathrm{TDC}_{\mathrm{max}} = 4980$ is chosen for subsequent studies.

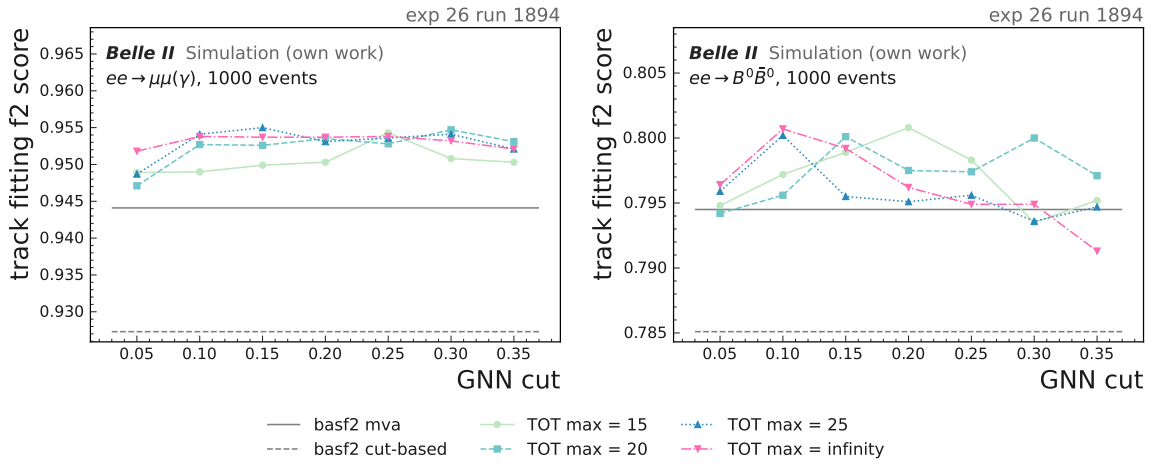


(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.11.: Track fitting f_2 score for different $\mathrm{TOT}_{\mathrm{min}}$ cuts. The pre-selection cut $\mathrm{TOT}_{\mathrm{min}} = 3$ is chosen for subsequent studies.

as the working point, representing an effective compromise between suppressing unphysical early hits and preserving signal hits at small drift times.

The $\mathrm{TOT}_{\mathrm{min}}$ selection complements the $\mathrm{ADC}_{\mathrm{min}}$ requirement by suppressing hits with very short time-over-threshold, which are typically associated with small deposited charges or spurious electronic pulses. Increasing $\mathrm{TOT}_{\mathrm{min}}$ from 0 to 3 substantially improves hit-level background rejection from approximately 39% to 57%, while inducing only a modest loss in hit efficiency. On this basis, I scan $\mathrm{TOT}_{\mathrm{min}}$ in the range 0-3, in conjunction with the previously optimized ADC and TDC selections, and quantify the



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.12.: Track fitting f_2 score for different $\mathrm{TOT}_{\mathrm{max}}$ cuts. The pre-selection cut $\mathrm{TOT}_{\mathrm{max}} = \infty$ is chosen for subsequent studies.

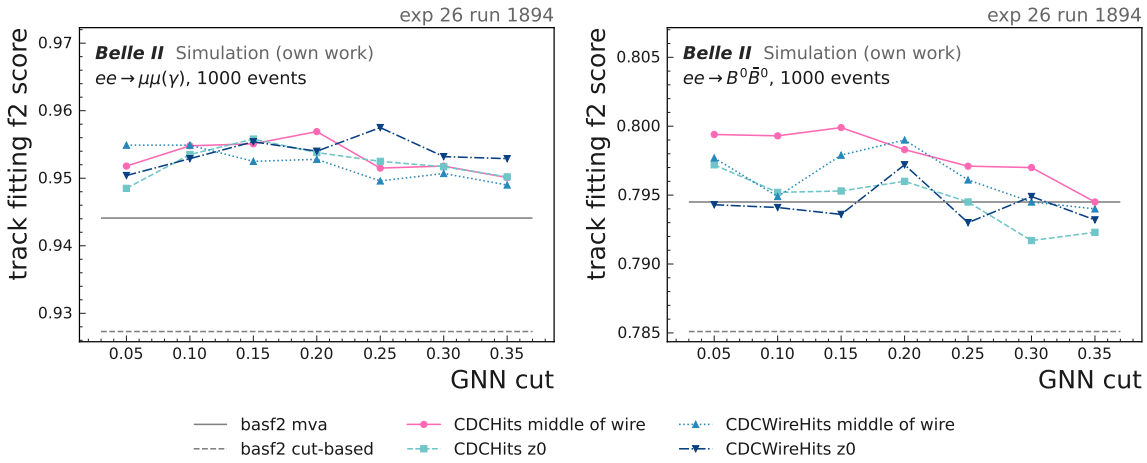
Table 8.2.: Final pre-filter cut configuration for ADC, TDC, and TOT.

Parameter	Pre-selection range
ADC	$[8, 2000]$
TDC	$[0, 4980]$
TOT	$[3, \infty)$

impact on the f_2 score, as shown in Figure 8.11. The results indicate that $\mathrm{TOT}_{\mathrm{min}} = 3$ provides the best or near-best track-level performance in both $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ samples, demonstrating that the rejection of very short-TOT hits remains beneficial even in the presence of the GNN classifier. Consequently, I adopt $\mathrm{TOT}_{\mathrm{min}} = 3$ as the default lower threshold.

Finally, the upper TOT selection is studied by removing hits exhibiting exceptionally large time-over-threshold values, which might be associated with saturated or otherwise anomalous pulse shapes. As illustrated in Table 8.1, both hit-level efficiency and background rejection show only a weak dependence on $\mathrm{TOT}_{\mathrm{max}}$ for values in the range 15 to ∞. In agreement with this observation, the f_2 curves shown in Figure 8.12 vary only mildly over the scanned interval and do not indicate a clear optimum at particularly restrictive cut values. In the absence of substantial performance improvement and to minimize the loss of potentially rare signal hits in this regime, I therefore adopt the inclusive choice $\mathrm{TOT}_{\mathrm{max}} = \infty$, *i.e.* without explicit upper bound on TOT.

In summary, the final pre-selection configuration employed in subsequent analyzes is specified in Table 8.2. This configuration, which employs relatively loose but physically motivated selection criteria, is analogous to those used in the cut-based filter (defined in



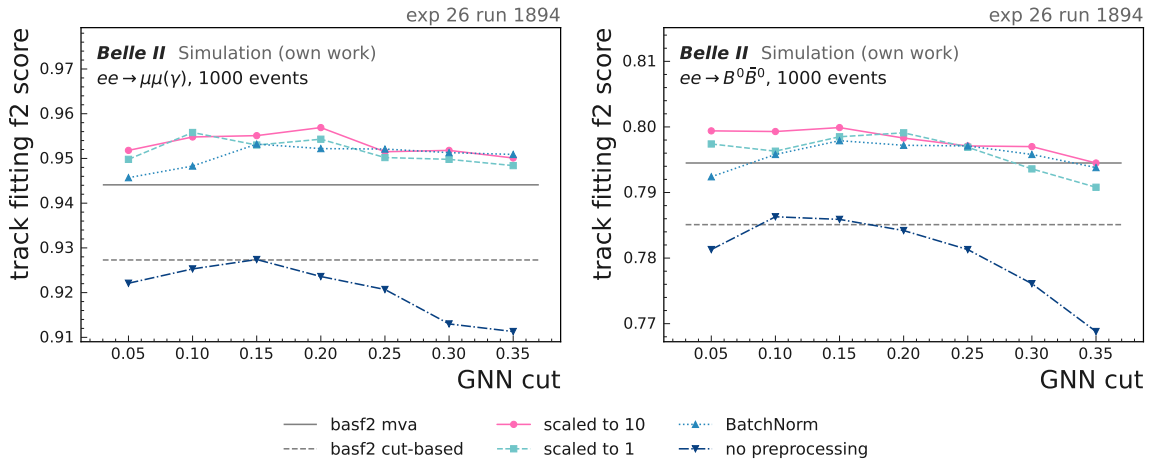
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.13.: Track fitting f_2 score evaluated for the two hit modes `CDCHits` and `CDCWireHits` used for training and inference. In addition two configurations with either middle-of-wire or $z = 0$ coordinates are compared. The best performance is obtained using the `CDCHits` information with extracting the hit positions at the middle of the wire.

subsection 4.1.1) but overall less restrictive. It achieves an effective compromise between hit efficiency at 98.6 % and hit background suppression at 56.7 % before the GNN is applied.

8.1.4. Hit mode

In addition, I investigate alternative configurations of the underlying hit representation used as input to GNN. As introduced in section 5.1, two distinct hit collections are considered: `CDCHits`, which comprise all reconstructed hits in the CDC, and `CDCWireHits`, which are the output objects of the `WireHitPreparer` module (see subsection 4.1.1). The `CDCWireHits` impose simple cross-talk filters and an explicit masking on known malfunctioning boards in the specified run conditions and thus contain a reduced subset of hits.
In addition, for each of the two hit collections, the transverse hit coordinates x and y can either be evaluated at the center of the sense wire or projected onto the plane at $z = 0$, resulting in four distinct hit configurations in total.
To quantify the impact of these choices on tracking performance, I train and evaluate the GNN for all four hit configurations. The corresponding track fitting f_2 scores as a function of the GNN-cut threshold are presented in Figure 8.13. In the $B^0\bar{B}^0$ case, the two $z = 0$ configurations consistently achieve slightly lower f_2 scores than their `CDCHits` counterparts. Furthermore, the `CDCWireHits` configuration exhibits overall poorer performance compared to the `CDCHits` configuration, indicating that the information loss induced by the cross-talk filter and masking of bad boards is not fully recovered by the GNN. This finding suggests that, given the current training strategy and network architecture, retaining the complete set of available hits and allowing the network to learn cross-talk affected patterns is more effective than removing corresponding hits a priori.



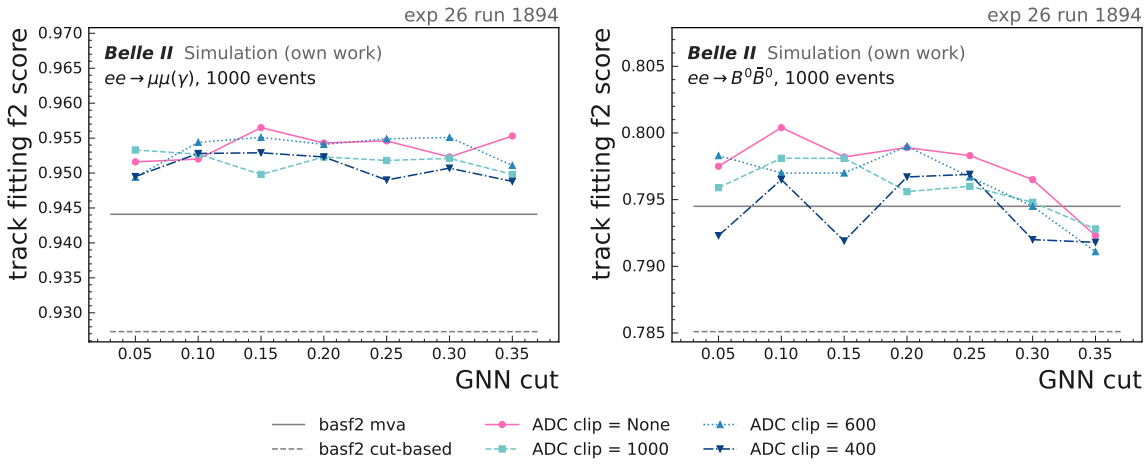
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.14.: Track fitting f_2 score for different input pre-processing schemes (baseline scaling to $[-10, 10]$, scaling to $[-1, 1]$, `BatchNorm` layer, and no pre-processing). The differences among the various schemes are marginal but remain substantial when compared to the configuration without pre-processing. The baseline scheme is retained for subsequent analyses.

8.1.5. Pre-processing

An additional design consideration concerns the pre-processing of the input features prior to the GNN inference, as discussed in section 5.3. In the baseline configuration, all node and edge features are linearly rescaled to the common interval $[-10, 10]$, as illustrated in Figure 8.2 and Figure 8.3. To assess the robustness of the model with respect to this choice, I compare three alternative schemes: rescaling to the narrower range $[-1, 1]$, replacing the fixed rescaling with a learnable batch-normalization layer [91], and the raw, un-normalized inputs without any pre-processing for comparison. The resulting track fitting f_2 scores as a function of the GNN-cut threshold are shown in Figure 8.14, evaluated on the standard benchmark channels. The curves corresponding to the baseline, $[-1, 1]$ rescaling, and batch normalization are nearly indistinguishable within statistical uncertainties, indicating that the model performance is largely insensitive to the precise rescaling convention, provided that the inputs are approximately normalized. In contrast, the configuration without any pre-processing exhibits a significantly degraded f_2 score in both samples, demonstrating that some form of feature normalization is essential for stable training and sufficient performance. Based on these observations, I retain the original baseline rescaling to $[-10, 10]$ for all subsequent studies.

In a separate pre-processing study, I investigate whether explicitly clipping large ADC values can stabilize the training or improve the final tracking performance. For this purpose, I introduce an upper clip in the ADC feature on 400, 600, 1 000, 2 000, and 4 000 counts, and compare these configurations with the default setup without clipping. The resulting track fitting f_2 scores as a function of the GNN-cut threshold are shown in



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

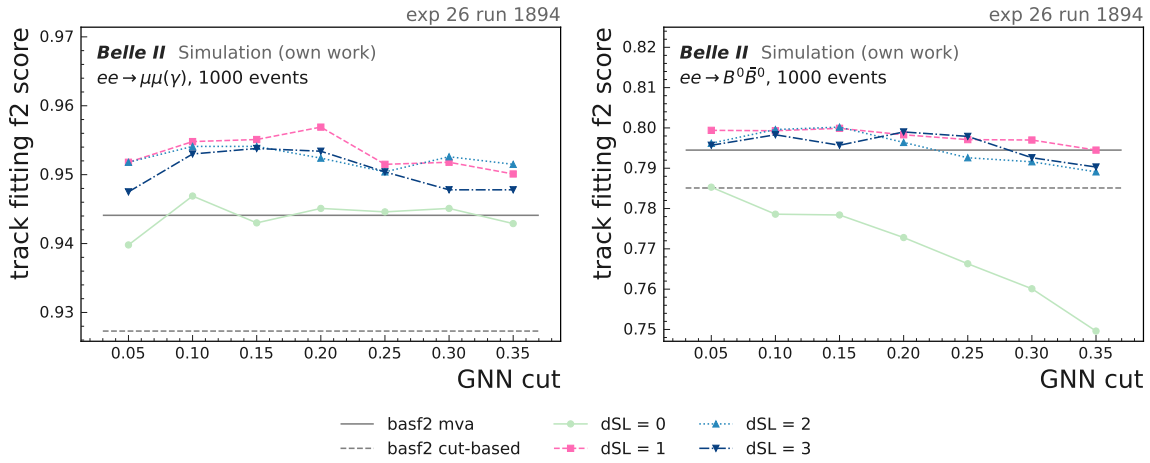
Figure 8.15.: Track fitting f_2 score as a function of the GNN-cut threshold for different ADC clipping values (no clipping and upper clips at 400, 600, 1 000, 2 000, and 4 000). For subsequent analyses no ADC clipping is applied.

Figure 8.15, evaluated on the standard $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ test samples at experiment 26, run 1894. Within the statistical uncertainties associated with the correlated evaluation, none of the considered clipping configurations exhibits a statistically significant improvement relative to the un-clipped baseline. I therefore conclude that explicit ADC clipping is not beneficial for the present setup and do not apply it in the final pre-processing scheme.

8.1.6. Graph dimensions

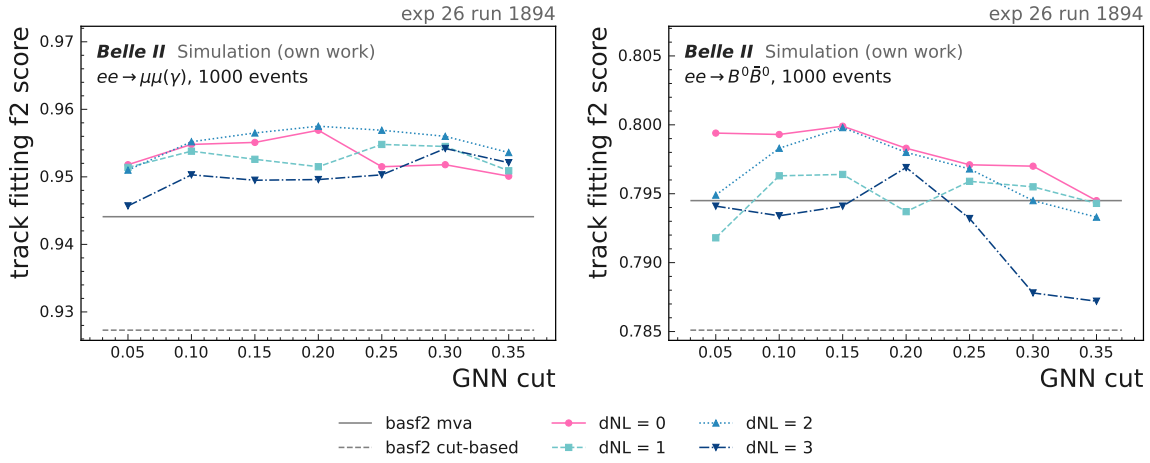
For graph-construction studies, I adopt as a starting point the baseline configuration introduced in section 5.2, in which each hit is connected to at most one-wire distance neighbors in the same layer ($d_{\mathrm{SL}} = 1$), half-wire distance neighbors in the subsequent layer ($d_{\mathrm{NL}} = 0$), and one-wire distance neighbors in the next-to-next layer ($d_{\mathrm{NNL}} = 1$). This connectivity pattern approximates the local track topology in the CDC while keeping the average number of nodes, and consequently the number of edges, low. To assess whether increased local connectivity can increase the performance of the GNN, I vary the maximum number of neighbors allowed in the same layer, the next layer, and the next-to-next layer in separate parameter scans. The corresponding track fitting f_2 scores as a function of the GNN-score threshold are presented in Figure 8.16-8.18. In all three scans, none of the extended configurations leads to a significant improvement over the baseline choice $(d_{\mathrm{SL}}, d_{\mathrm{NL}}, d_{\mathrm{NNL}}) = (1, 0, 1)$, while increasing the number of edges and thus the computational cost of message passing.

In a second step, I investigate the impact of edge directionality. In the default, unidirectional configuration, edges follow the radial and azimuthal direction of increasing wire number, such that each pair of neighboring hits is connected by a single directed edge. In the bi-directional configuration, each such pair is represented by two anti-parallel



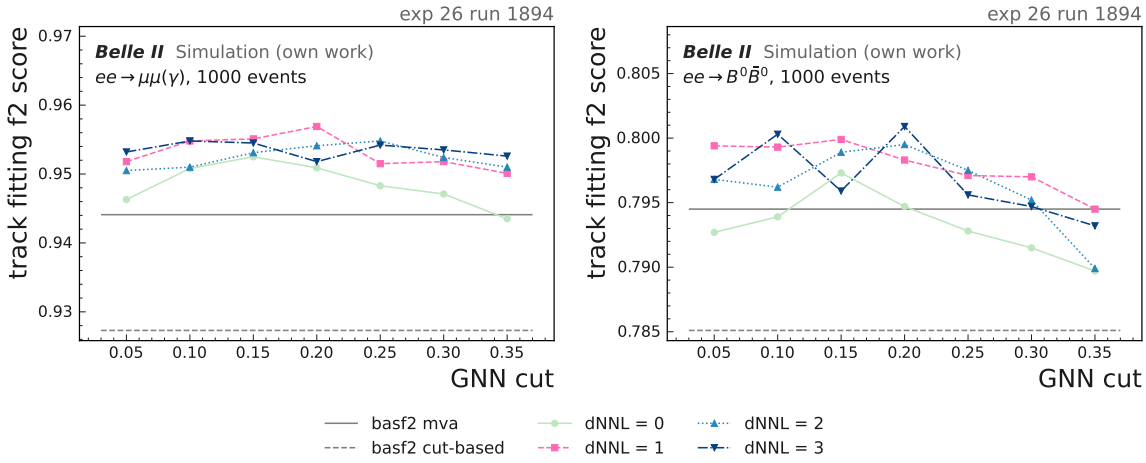
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.16.: Track fitting f_2 score as a function of the GNN-cut threshold for different numbers of allowed neighbors in the same CDC layer ($d_{\mathrm{SL}} = 0, 1, 2, 3$).



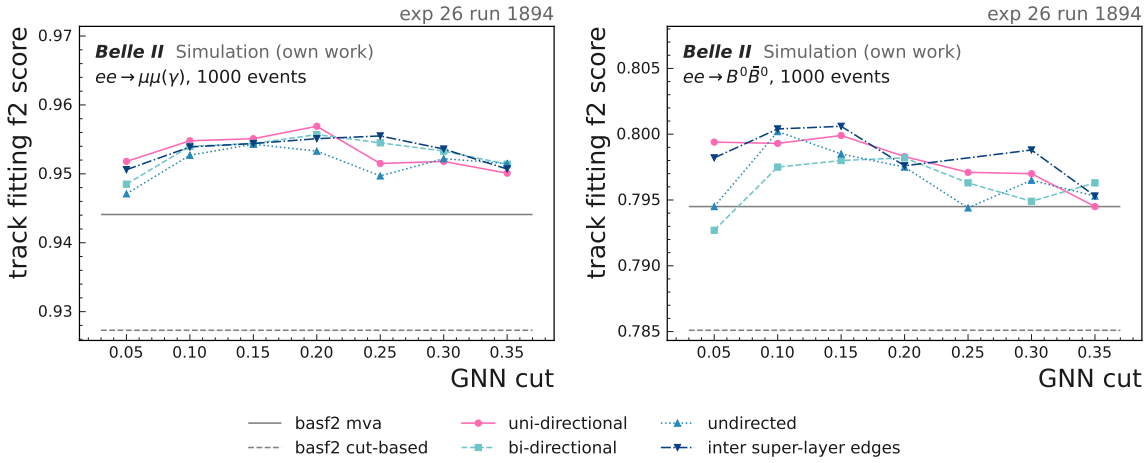
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.17.: Track fitting f_2 score as a function of the GNN-cut threshold for different numbers of allowed neighbors in the next CDC layer ($d_{\mathrm{NL}} = 0, 1, 2, 3$).



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.18.: Track fitting f_2 score as a function of the GNN-cut threshold for different numbers of allowed neighbors in the next-to-next CDC layer ($d_{\mathrm{NNL}} = 0, 1, 2, 3$).



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.19.: Track fitting f_2 score as a function of the GNN-cut threshold for different graph-direction configurations (uni-directional, bi-directional, undirected) and for graphs with additional inter-superlayer edges.

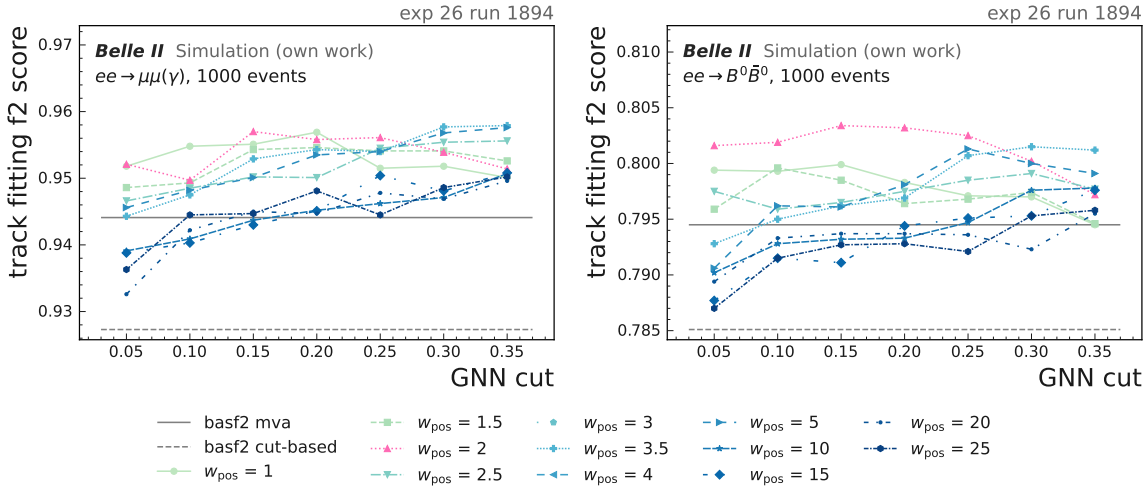
directed edges, thereby enabling information to propagate explicitly in both directions along the track. In the undirected configuration, edges are treated as directionless in the graph representation, *i.e.* a single edge encodes a symmetric relationship between the two hits.
The comparison of these options in Figure 8.19 demonstrates that bi-directional and undirected graphs do not improve the track fitting f_2 score relative to the uni-directional baseline, while they effectively double the number of edges that must be processed by the GNN. Given the absence of performance gain and the increased computational cost, I adopt the uni-directional edge definition as the default.
Finally, I extend the graph by inter-superlayer edges that explicitly connect hits located near the boundaries between neighboring superlayers. In the test configuration using experiment 26, run 1894 conditions considered here, these additional edges do not produce a statistically significant improvement in f_2, and the corresponding curve in Figure 8.19 remains compatible with the baseline within uncertainties. However, in dedicated studies performed at lower occupancies (for example, experiment 22, run 26), such inter-superlayer connections played a critical role in ensuring that hits located in the vicinity of superlayer boundaries have a sufficient number of neighboring hits and can therefore be correctly associated with particle tracks. For this reason, I retain the inter-superlayer edges in the final graph-building configuration, even though their benefit is only marginal in the intermediate-background benchmark used in this study.

8.1.7. Loss functions

In the baseline configuration, as well as in all parameter studies conducted thus far, the binary cross-entropy (BCE) loss function [92] has been employed, which is defined as

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)\right], \tag{8.1}$$

where $y_i \in \{0,1\}$ denotes the ground-truth class label, and $\hat{p}_i \in (0,1)$ represents the predicted probability assigned to sample i, for a total of N samples. This loss function operates under the assumption that the classification classes are approximately balanced in terms of their relative frequencies.
However, the hit-classification task exhibits a class imbalance, with a post-pre-selection signal-to-background ratio of approximately 1:4 in the samples for experiment 26, run 1894, and even higher imbalances in the case of simulated future background conditions. This motivates the adoption of loss functions that explicitly account for such an imbalance.
In addition to the baseline BCE loss, I therefore examine several variants of loss functions that are commonly used in imbalanced segmentation and object-detection tasks. Specifically, I consider an extension of the BCE loss by class-weighted logits, the focal loss [93], a class-balanced focal loss incorporating effective-number-of-samples weighting [94], as well as the Tversky and Dice losses [95, 96]. For each loss type, I perform a scan over a restricted hyper-parameter space centered around values recommended in the respective literature and identified by preliminary optimization studies. The resulting track fitting



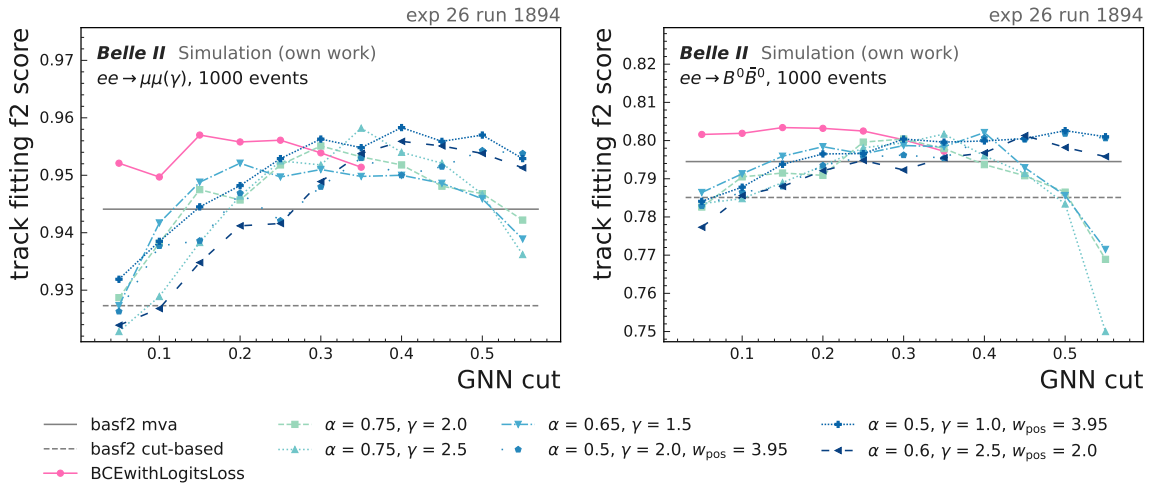
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.20.: Track fitting f_2 score as a function of the GNN-cut threshold for the `BCEWithLogits` loss using different positive-class weights w_{pos}. The best performance is obtained with $w_{\text{pos}} = 2$.

f_2 scores are then evaluated and compared on the standard $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ test samples for experiment 26, run 1894.
I begin with the weighted `BCEWithLogits` loss, introducing a scalar positive-class weight w_{pos} applied to the BCE loss

$$\mathcal{L}_{\text{BCELogits}} = -\frac{1}{N}\sum_{i=1}^{N}\left[w_{\text{pos}} y_i \log\sigma(z_i) + (1-y_i)\log(1-\sigma(z_i))\right], \tag{8.2}$$

where $z_i \in \mathbb{R}$ is the raw logit output of the network and $\sigma(z) = (1+e^{-z})^{-1}$ is the sigmoid function. Compared to applying a stand-alone sigmoid layer followed by a separate BCE loss, this combined formulation exhibits improved numerical stability, as it mitigates numerical overflow and underflow in the presence of logits with large positive or negative magnitudes. The scan is performed over a range of w_{pos} values, where values substantially larger than three would approach or exceed the inverse of the class imbalance ratio. The corresponding f_2 curves in Figure 8.20 indicate that a moderate up-weighting of the signal class yields improved performance relative to the unweighted baseline, whereas very large weights cause a deterioration in performance. The best overall performance is observed for $w_{\text{pos}} = 2$, which is adopted as a baseline configuration for subsequent comparisons. Although this value is somewhat smaller than the naive expectation based on the 1:4 class imbalance, it likely reflects the influence of other elements of the training configuration (such as the GNN architecture), which already tend to favor high signal efficiency.
As an alternative, the *focal loss* [93] introduces a focus parameter γ that attenuates the contribution of well-classified examples (*e.g.* obvious outliers), as well as a coefficient α



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.21.: Track fitting f_2 score as a function of the GNN-cut threshold for a Dice loss with different smoothness parameters. The best configuration with $\alpha = 0.5$, $\gamma = 1.0$ and w_{pos} reaches similar f_2 score values as the baseline applying `BCEWithLogits` loss.

that functions analogously to a class-weighting factor.

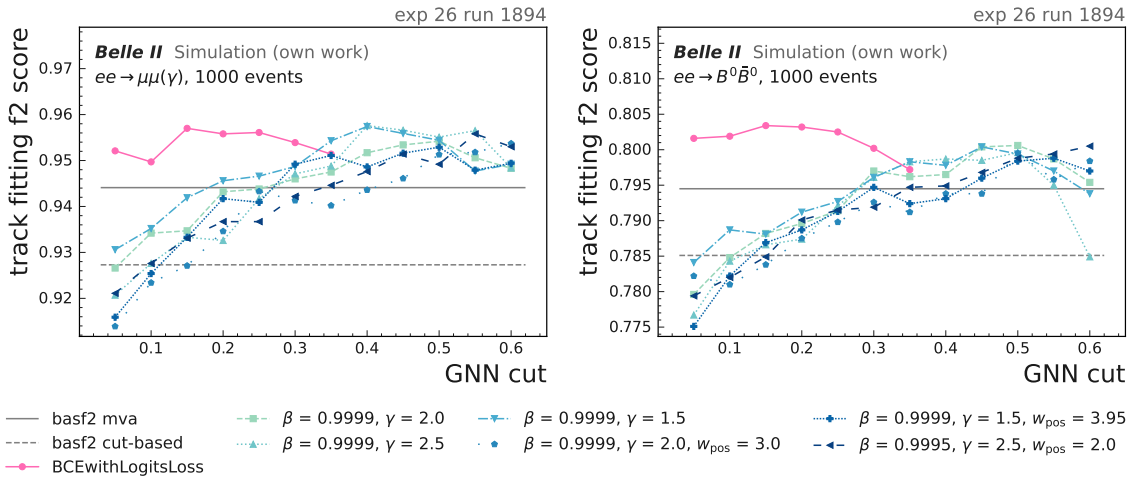
$$\mathcal{L}_{\text{FL}} = -\frac{1}{N}\sum_{i=1}^{N} \alpha y_i w_{\text{pos}} (1-\hat{p}_i)^{\gamma} \log \hat{p}_i + (1-\alpha)(1-y_i)\hat{p}_i^{\gamma} \log(1-\hat{p}_i), \tag{8.3}$$

with a class-weighting factor $\alpha \in [0,1]$ and focusing parameter $\gamma \geq 0$ that down-weights well-classified samples. For $\gamma = 0$ and $\alpha = 0.5$ the focal loss reduces to the BCE loss. Six hyperparameter configurations are evaluated: $(\alpha, \gamma) = (0.75, 2.0)$, $(0.75, 2.5)$, and $(0.65, 1.5)$, along with three additional variants that incorporate an explicit positive-class weight $w_{\text{pos}} \in \{3.95, 3.95, 2.0\}$ in combination with $(\alpha, \gamma) = (0.5, 2.0)$, $(0.5, 1.0)$, and $(0.6, 2.5)$, respectively. As illustrated in Figure 8.21, several of these configurations achieve f_2 scores that are comparable to those obtained with the best `BCEWithLogits` configuration. However, none of them yields a consistent or substantial improvement across both test samples and the investigated range of GNN-cut thresholds.

A class-imbalance-aware generalization of the focal loss is the *class-balanced focal loss*, which is defined as

$$\mathcal{L}_{\text{CB}} = -\frac{1-\beta}{1-\beta^{n_y}} \frac{1}{N} \sum_{i=1}^{N} \left[w_{\text{pos}} (1-\hat{p}_i)^{\gamma} \log \hat{p}_i + (1-y_i)\, \hat{p}_i^{\gamma} \log (1-\hat{p}_i) \right], \tag{8.4}$$

where n_y denotes the number of samples belonging to the ground-truth class y, and $\beta \in [0,1)$ replaces the parameter α of the original focal loss by explicitly encoding the effective number of samples per class.
For the class-balanced focal loss, I adopt the effective-number formulation with weights $\beta \in \{0.9999, 0.9995\}$, $\gamma \in \{1.5, 2.0, 2.5\}$ and positive-class weights $w_{\text{pos}} \in \{2.0, 3.0, 3.95\}$,



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.22.: Track fitting f_2 score as a function of the GNN-cut threshold for class-balanced focal loss with different (β, γ) and positive-class weight configurations. The optimum appears to be shifted toward higher values relative to the baseline configuration employing the `BCEWithLogits` loss. It is possible that the f_2 score could further improve for larger GNN cut thresholds. Nevertheless, within the explored region of the parameter space, the baseline configuration yields the best performance.

resulting in six distinct configurations (see Figure 8.22). Multiple configurations achieve performance comparable to `BCEWithLogits`. However, no single setting consistently exceeds the simpler weighted `BCEWithLogits` baseline.

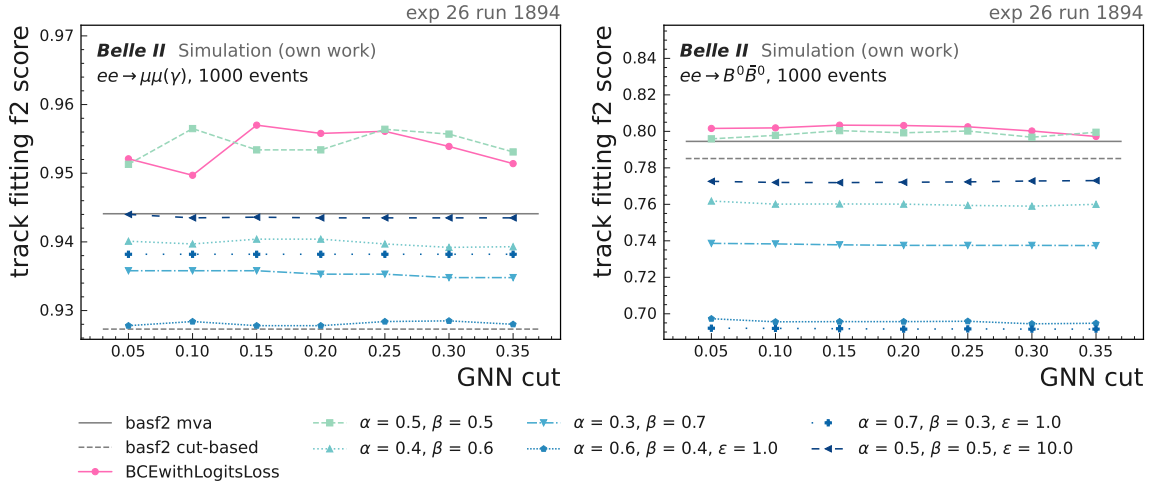
The *Dice loss* is derived from the Dice coefficient, a statistical metric that quantifies the similarity between two sets of data and is formally defined as

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{p}_i + \epsilon}{\sum_{i=1}^{N} (y_i \hat{p}_i) + \epsilon}, \tag{8.5}$$

where $\epsilon > 0$ is a smoothing constant that ensures numerical stability and prevents division by zero in the absence of positive samples. I vary the smoothness parameter over the interval $\epsilon \in [10^{-4}, 10.0]$.
To assign a higher weight to false negatives (FNs) relative to false positives (FPs) in the presence of pronounced class imbalance, the *Tversky loss* generalizes the Dice loss as follows:

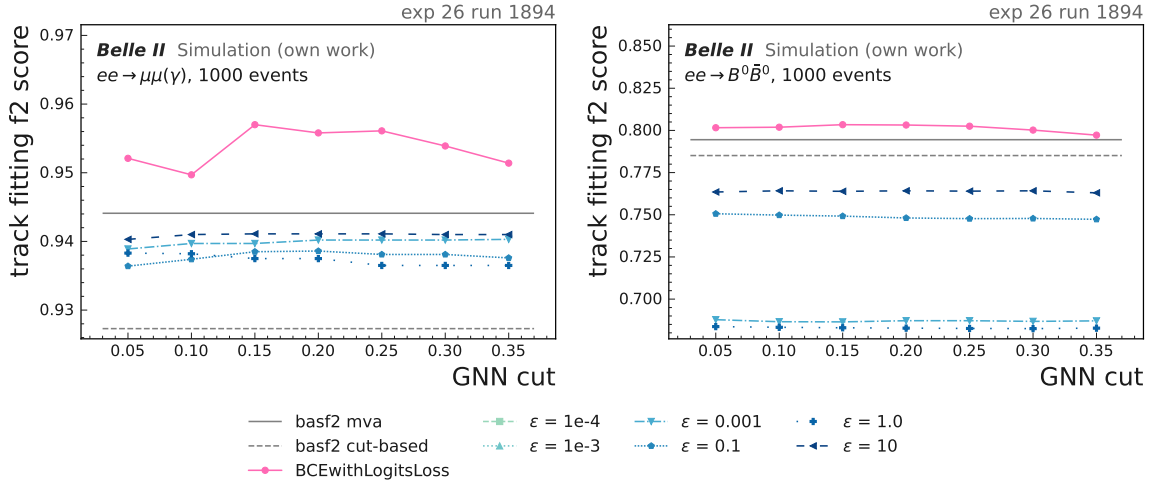
$$\mathcal{L}_{\text{Tversky}} = 1 - \frac{\sum_{i=1}^{N} y_i \hat{p}_i + \epsilon}{\sum_{i=1}^{N} \left(y_i \hat{p}_i + \alpha(1 - y_i)\hat{p}_i + \beta y_i (1 - \hat{p}_i)\right) + \epsilon}, \tag{8.6}$$

where the three terms in the denominator correspond to the counts of true positive (TP), FP, and FN, respectively. The weighting parameters $\alpha \geq 0$ and $\beta \geq 0$, constrained by $\alpha + \beta = 1$, regulate the relative contribution of false positives and false negatives to the



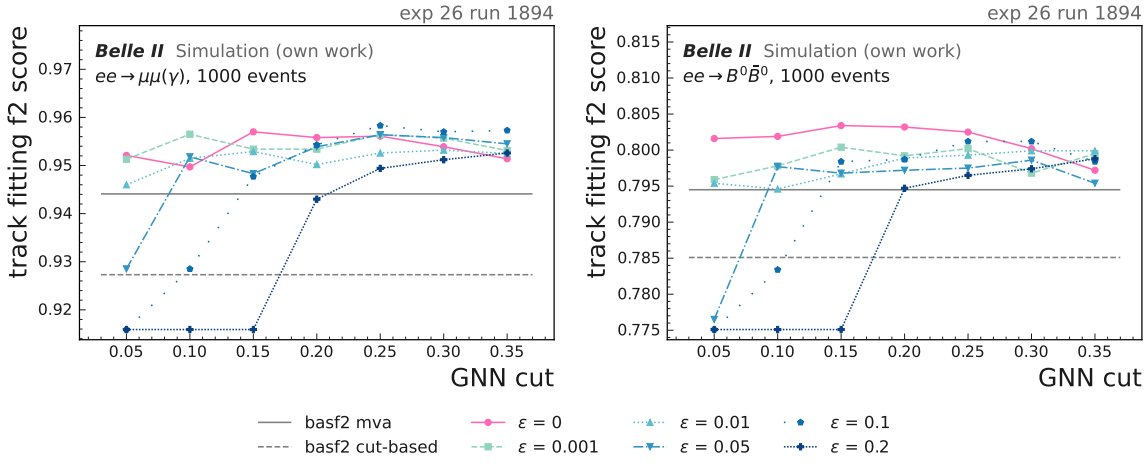
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.23.: Track fitting f_2 score as a function of the GNN-cut threshold for Tversky loss with different (α, β) and smoothness parameter settings. The configuration with $\alpha = \beta = 0.5$ achieves the best performance, which is nearly equivalent to that of the baseline model trained using the `BCEWithLogits` loss function.



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.24.: Track fitting f_2 score as a function of the GNN-cut threshold for focal loss with different (α, γ) and positive-class weight configurations. None of the configurations is able to reach competitive f_2 score values compared to the baseline using the `BCEWithLogits` loss.



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

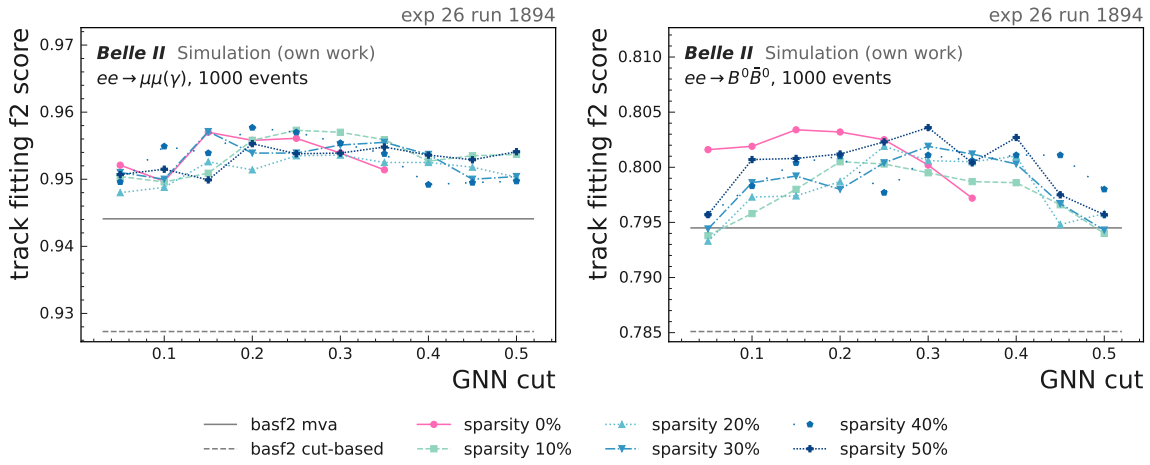
Figure 8.25.: Track fitting f_2 score as a function of the GNN-cut threshold for different label-smoothing parameters ($\epsilon = 0.001, 0.01, 0.05, 0.1, 0.2$). In particular evaluated on $B^0\bar{B}^0$, none of the configurations is able to reach competitive f_2 score values in the tested parameter space compared to the baseline using the `BCEWithLogits` loss.

loss function. For $\alpha = \beta = 0.5$, the Tversky loss is equivalent to the Dice loss. I evaluate the Tversky loss for different combinations of parameters $(\alpha, \beta, \epsilon)$.
The corresponding results are presented in Figure 8.23 and Figure 8.24. In general, the resulting f_2 scores are significantly lower than those achieved with the best-tuned `BCEWithLogits` configuration. Most parameter settings yield substantially inferior performance, and the outcomes exhibit a pronounced sensitivity to both the smoothness and the asymmetry parameters.

Overall, these studies indicate that several of the specialized loss functions can attain performance similar to that of an optimized `BCEWithLogits` loss, but none provides a clear improvement. By contrast, the weighted `BCEWithLogits` loss with $w_{\text{pos}} = 2$ produces consistently robust results across both the $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ benchmarks, as well as over the relevant range of GNN-score thresholds. Consequently, I adopt `BCEWithLogits` with a positive-class weight $w_{\text{pos}} = 2$ as the default training loss for subsequent studies.

8.1.8. Additional regularization studies

To assess whether the generalization capability of the GNN can be further increased, I examine three additional regularization strategies: loss smoothing, weight pruning, and dropout. For each method, the effect is quantified by comparing the track fitting f_2 score on the standard $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ test samples at experiment 26, run 1894, against a baseline configuration in which the corresponding modification is not applied.
In theory, loss smoothing is employed to mitigate excessive model confidence by slightly relaxing the strictness of the binary target labels. Specifically, I replace the hard binary



(a) evaluated on $\mu^+\mu^-(\gamma)$

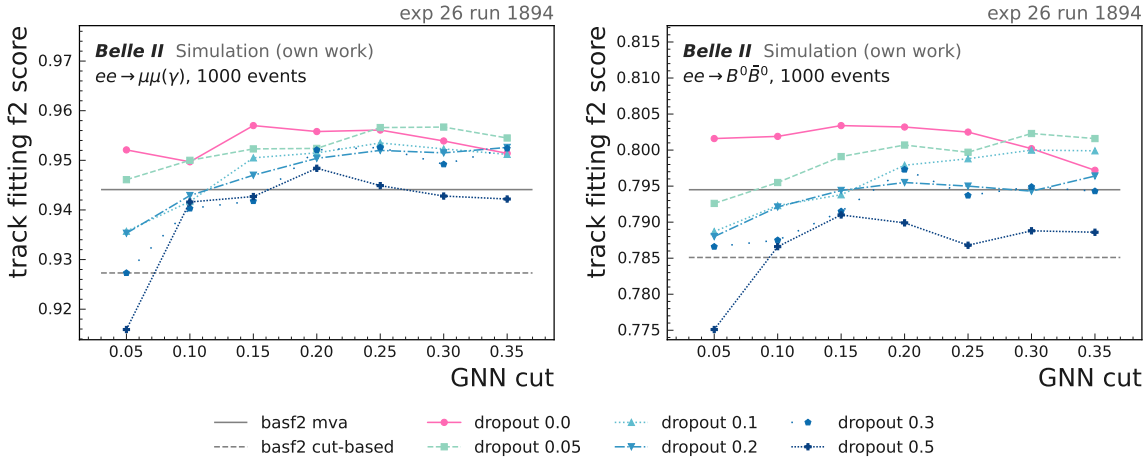
(b) evaluated on $B^0\bar{B}^0$

Figure 8.26.: Track fitting f_2 score as a function of the GNN-cut threshold for different post-training weight sparsity levels (10%, 20%, 30%, 40%, 50%), and compared to the un-pruned baseline (sparsity = 0%). In the $B^0\bar{B}^0$ configuration, the f_2 metric remains consistently lower than that of the baseline across all evaluated settings. A marginal improvement over the baseline is observed only at a sparsity level of 50 %.

labels by $y' = (1 - \epsilon)\, y + 0.5\, \epsilon$ with smoothing parameters $\epsilon \in \{0.001, 0.01, 0.05, 0.1, 0.2\}$. The corresponding f_2 scores as a function of the GNN-cut threshold are presented in Figure 8.25. Across the entire range of tested values, none of the curves exhibit an improvement over the baseline, and the performance degradation becomes more pronounced for larger values of ϵ. This observation suggests that, for the current configuration, label smoothing does not yield a measurable performance gain.

Unstructured pruning is evaluated by progressively imposing sparsity on network weights during training, targeting final sparsity levels of 10%, 20%, 30%, 40% and 50%. For each pruned model, the tracking performance is re-assessed without any additional fine-tuning. As shown in Figure 8.26, moderate pruning levels lead to a degradation of the f_2 score relative to the dense baseline, while a more aggressive sparsity of 50% produces a slight improvement in performance around a GNN cut value of 0.3 in the $B^0\bar{B}^0$ sample. This effect is likely attributable to statistical fluctuations. Given that the un-pruned model is already highly compact in terms of parameter count, the marginal gains in memory usage and computational efficiency do not compensate for the potential risk of performance degradation, and pruning is therefore not applied in the final configuration.

Finally, I investigate dropout as a standard stochastic regularization method. A dropout layer is inserted between each linear layer and its subsequent ReLU activation, and the dropout probabilities are scanned over the range $p_{\text{drop}} \in \{0.0, 0.05, 0.1, 0.2, 0.3, 0.5\}$, where $p_{\text{drop}} = 0$ defines the baseline configuration. The corresponding f_2 curves shown in Figure 8.27 indicate that even relatively small dropout rates produce a noticeable performance degradation. Given the relatively small size of the model and the absence of



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

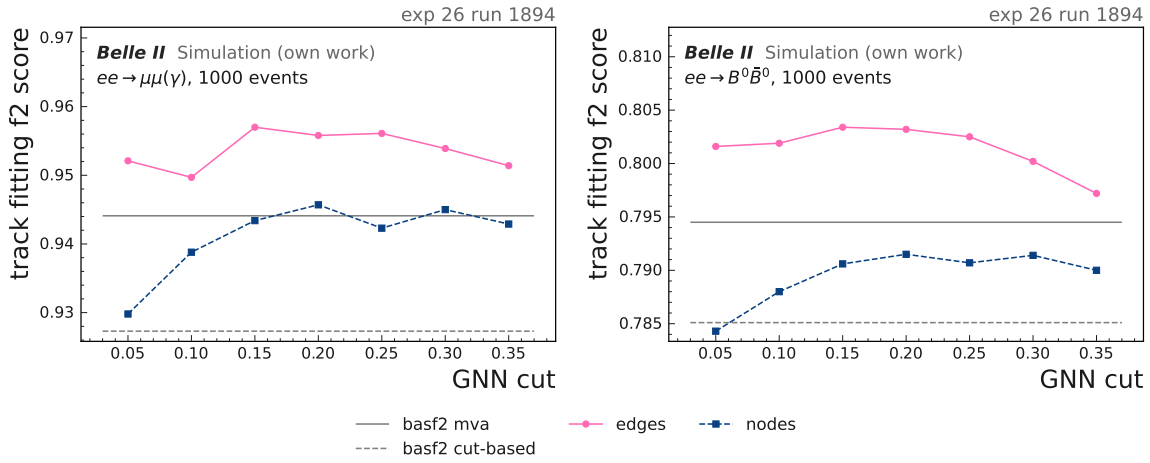
Figure 8.27.: Track fitting f_2 score as a function of the GNN-cut threshold for different dropout probabilities ($p_{\text{drop}} = 0.0, 0.05, 0.1, 0.2, 0.3, 0.5$) applied between the graph-convolution blocks. The baseline without dropout is not outperformed in the investigated parameter space.

clear evidence of over-fitting, this behavior appears to be reasonable.

In conclusion, none of the regularization strategies examined, including loss smoothing, post-training pruning, or additional dropout, results in an improvement over the baseline configuration. For the final model, these modifications are therefore omitted and, instead, I rely on the implicit regularization induced by the compact network architecture (*i.e.* small number of trainable parameters), the specific choice of loss function, and the heterogeneity of the training dataset.

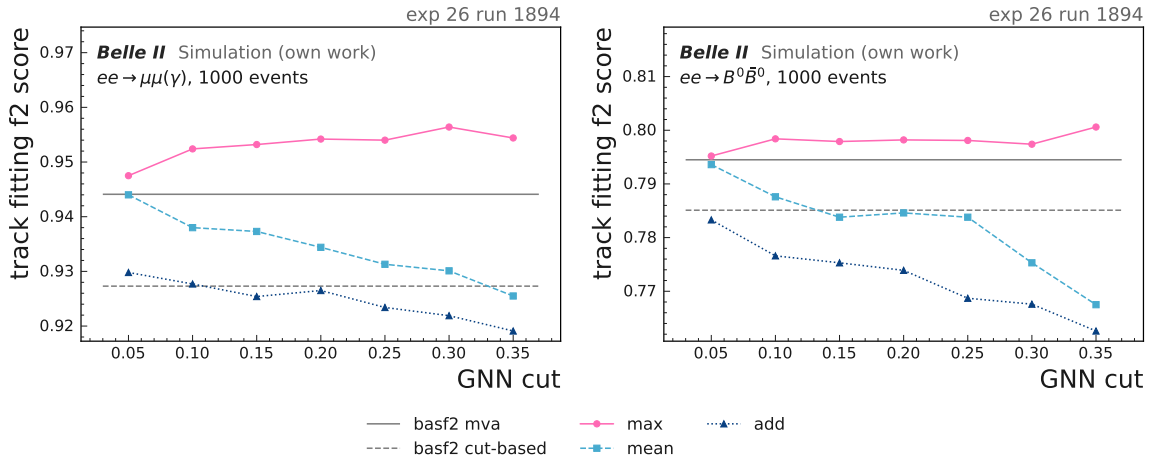
8.1.9. Training targets and aggregation strategy

A natural question is whether the GNN should be trained directly on node-level targets, since the final output of the model is a per-node score. In the baseline configuration, the training target is defined on the edges, with labels indicating whether a given edge connects two hits associated with the same reconstructed track, and the node-level scores are subsequently obtained by aggregating the predictions of the edges incident to each node. To evaluate the alternative strategy, I construct a node-target variant in which each hit is explicitly labeled as signal or background, and the loss is computed on the node output. The comparison in Figure 8.28 shows that this node-target training fails to achieve the edge-target baseline performance in both $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ samples. These results indicate that edge-based learning might provide richer information to the network, and therefore I adopt edge-level targets for all subsequent studies. An alternative explanation may be that the current configuration has been specifically optimized for edge-level targets, and that re-optimization of the algorithm could instead favor node-level targets.



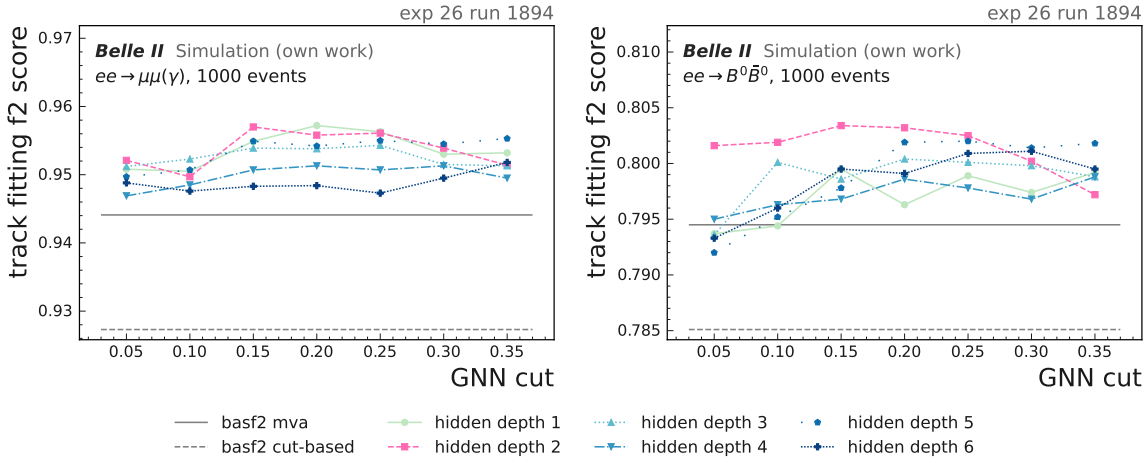
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.28.: Track fitting f_2 score as a function of the GNN-cut threshold for models trained with edge-level targets (baseline) and node-level targets. A model trained on edge-level targets clearly outperforms a model trained on nodes directly.



(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.29.: Track fitting f_2 score as a function of the GNN-cut threshold for different output aggregation functions (max, mean, add) used to obtain node scores from edge predictions. A model applying `max` output aggregation performs significantly better than using `mean` or `add` aggregation.



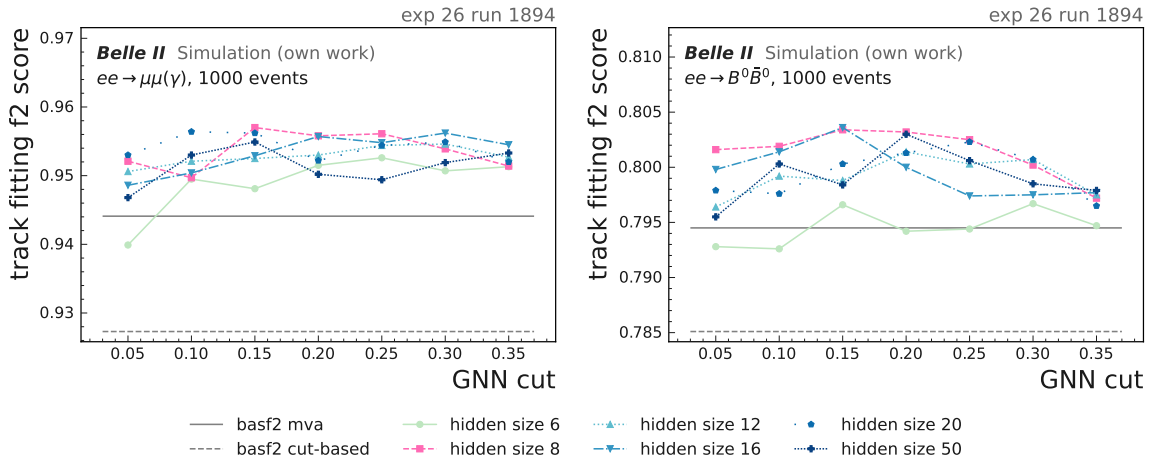
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.30.: Track fitting f_2 score as a function of the GNN-cut threshold for different hidden depths (number of linear layers from 1 to 6). The baseline model configured with a hidden layer depth of two demonstrates the best overall performance.

Another design choice for the GNN architecture is the selection of the aggregation scheme. In the intermediate message-passing layers, the baseline model applies an additive (`add`) aggregation of incoming messages, whereas at the final stage the edge-level scores incident on a node are reduced via a `max` operation to obtain the node-level prediction. I investigate alternative aggregation mechanisms by substituting the intermediate *add* aggregation with *max* or *mean*, and by evaluating *max*, *mean*, and *add* as output aggregation operators. Under the available training configuration, the intermediate *max* and *mean* variants do not exhibit convergence, suggesting that, under these conditions, they may fail to yield gradients of sufficient stability. For the output aggregation, the comparison in Figure 8.29 shows that *max* consistently outperforms both *mean* and *add* aggregation in terms of the track fitting f_2 score across the entire range of GNN-cut thresholds. Based on these findings, I retain the original configuration: training with edge-level targets, using additive aggregation for intermediate message passing and `max` aggregation for the final output aggregation.

8.1.10. Model dimension and larger training dataset

To evaluate which model size is required for the hit-filtering task, I scan both the depth and the width of the GNN while keeping all other settings fixed.
Initially, I vary the number of hidden linear layers (hereafter referred to as the "hidden depth") from one to six, with a default configuration consisting of two hidden layers. For each depth, the model is trained from scratch and evaluated on the standard $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ test samples at experiment 26, run 1894. The resulting track fitting f_2 scores as a function of the GNN-cut threshold are shown in Figure 8.30. Across the scanned range, the curves lie very close to each other, with no general improvement beyond the base-



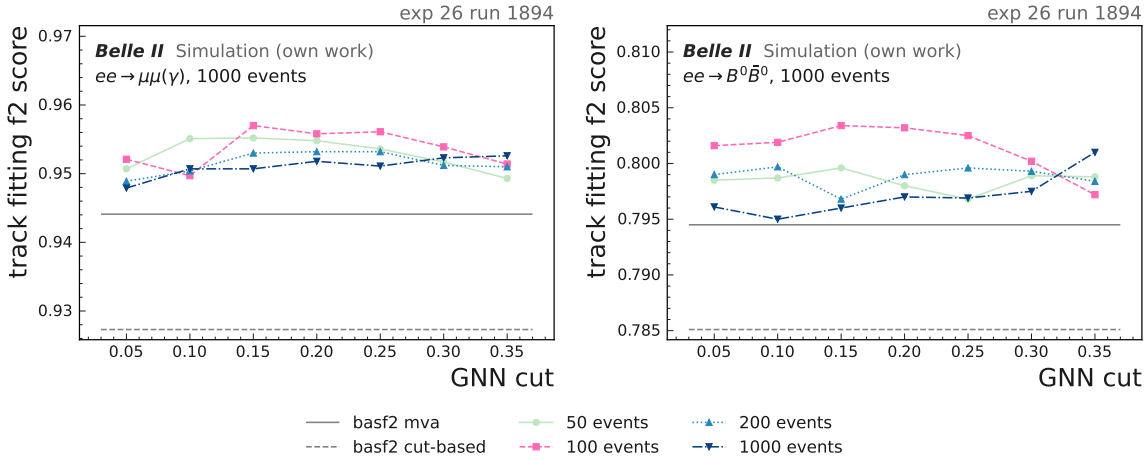
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.31.: Track fitting f_2 score as a function of the GNN-cut threshold for different hidden sizes (number of channels per hidden layer: 6, 8, 12, 16, 20, 50). The baseline model, configured with a hidden layer comprising eight hidden nodes, achieves the best overall performance.

line and a slight tendency towards decreased performance for the deepest configurations, which also entail higher computational costs during training and increased memory requirements. This indicates that the model is already well described by a small number of hidden layers and that additional layers mainly increase complexity without adding useful discriminating power.

In a second experiment, I vary the dimensionality of the hidden feature representation (hereafter referred to as the "hidden size") while keeping the network depth fixed, using a default hidden size of 8. The number of model nodes ranges from 6 to 50. As shown in Figure 8.31, neither very small nor very large hidden sizes outperform the baseline choice. Within uncertainties, all widths tested yield similar f_2 scores on both benchmarks, with a slight degradation at the extremes. Since larger models incur higher memory usage and computational cost, I retain the compact baseline width as the default. Overall, these scans confirm that a relatively small GNN is sufficient for this application and that increasing capacity does not translate into better hit filtering performance. It should be noted, however, that models with a larger number of parameters may require more training samples to achieve convergence. This hypothesis has not been empirically tested or confirmed in this study.

To quantify the influence of the training dataset size on the final model performance, I train otherwise identical models on progressively larger subsets of the full training sample. Starting from a very small training mixture with 50 events per sample category, I increase the number of training events to 200 and finally to 1 000, while keeping the validation and test datasets fixed to the standard $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ benchmarks at experiment 26, run 1894. The resulting track fitting f_2 scores as a function of the GNN-cut threshold are shown in Figure 8.32. For $B^0\bar{B}^0$, the curves corresponding to 50, 200 and



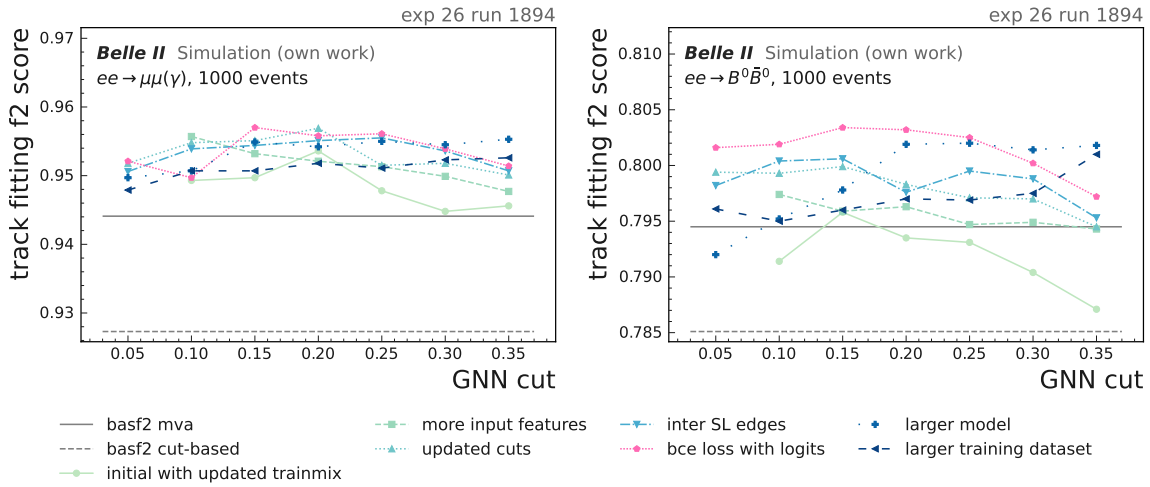
(a) evaluated on $\mu^+\mu^-(\gamma)$ (b) evaluated on $B^0\bar{B}^0$

Figure 8.32.: Track fitting f_2 score as a function of the GNN-cut threshold for different training sample sizes (50, 200, and 1000 events per category). Increasing the number of events for each of the sample categories does not improve the performance.

1 000 training events per category perform noticeably worse than the baseline obtained with 100 events per category. This behavior suggests that a sample of a few hundred events already provides sufficient statistics for the compact GNN to learn the relevant hit-topology patterns and that further increasing the training set size to 1 000 events does not produce any additional improvement. Given the substantial computational cost associated with larger training samples, I therefore consider training samples with 100 events per category to represent an appropriate compromise between performance and training time.

8.1.11. Summary of the design optimization

The studies in this chapter investigate the impact of a wide range of design choices on the performance of the GNN-based hit filter. Starting from a minimal initial configuration, I varied the training sample composition, the input feature set, the pre-selection cuts, the graph-building strategy, the loss function, regularization techniques, training targets and aggregation, as well as the model size and the size of training samples. For each configuration, the performance in terms of the track fitting f_2 score was evaluated on a common test bench of 1 000 $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events for experiment 26, run 1894 conditions.
The main optimization steps are summarized in Figure 8.33, where each optimization step leads to improved performance, in particular in the $B^0\bar{B}^0$ sample case, and ultimately to a substantial improvement over the legacy basf2 MVA and cut-based filters in benchmark channels. Although neither increasing the model size nor the size of the training dataset improves performance, I retain these configurations in this overview for the sake of completeness. In this context, the optimal operating point of the GNN appears to correspond to a cut value of 0.2.



(a) evaluated on $\mu^+\mu^-(\gamma)$

(b) evaluated on $B^0\bar{B}^0$

Figure 8.33.: Track fitting f_2 score for successive optimization steps of the GNN-based hit filter. The optimization includes adding input features, refined pre-filtering of ADC, TDC, and TOT, incorporating edges across super-layer boundaries, and introduction of a weighting factor in the BCE loss to reduce signal-background class imbalance. Increasing model capacity or training set size does not yield measurable performance gains.

The optimization steps and their impact on both track-level metrics (track fitting f_2 score, efficiency, fake rate) and hit-level metrics (hit efficiency and background rejection) are summarized in Table 8.3 for $B^0\bar{B}^0$ and $\mu^+\mu^-(\gamma)$ samples, respectively.

The reported values are obtained using a common GNN cut threshold of 0.2 and the track fitting f_2 score is adopted as the primary scalar metric to compare the filters. For $B^0\bar{B}^0$ events, the track fitting f_2 score, the track fitting efficiency and the hit background rejection rate exhibit a pronounced improvement, while the track fake rate is overall reduced. An analogous trend is observed for $\mu^+\mu^-(\gamma)$ events, where the already high baseline track efficiency increases further. The observed increase in track fake rate and the corresponding degradation in hit background rejection for $\mu^+\mu^-(\gamma)$ events can be attributed to the increased generalization capacity of the model towards B-events, achieved at the expense of a slight loss in performance on purely $\mu^+\mu^-(\gamma)$ samples compared to a model trained exclusively on $\mu^+\mu^-(\gamma)$ data.

Nevertheless, despite being higher than in the initial model configuration, the track fitting fake rate remains significantly lower compared to the basf2 legacy filter applications. Performance gains compared to the initial baseline configuration, particularly in the sample $B^0\bar{B}^0$, are primarily driven by the use of an enriched and more realistic training data set, the incorporation of additional geometrical and detector-information features, optimized ADC/TDC/TOT pre-selection criteria, an edge-based training target with `max` aggregation applied to the output aggregation, and a carefully tuned `BCEWithLogits` loss function.

In contrast, several modifications that are often beneficial in other deep-learning applications do not provide additional advantages in this context. Neither application of reg-

ularization techniques (dropout, loss smoothing, pruning) nor alternative loss functions (focal, class-balanced focal, Tversky, Dice) nor denser graphs or alternative aggregation schemes lead to a robust improvement over the chosen baseline. Likewise, increasing the model depth or width beyond a compact architecture, or enlarging the training dataset well beyond a few hundred events per sample category, do not improve the track- or hit-level metrics and can even slightly degrade them. This indicates that for the given problem and operating point, the final configuration represents a good balance between complexity and performance, and that further gains are more likely to come from the refined integration into the global tracking chain and application to different operating conditions than from additional modifications of the core GNN design.

Table 8.3.: Overview of track- and hit-level performance metrics evaluated on $\mu^+\mu^-(\gamma)$ (top) and $B^0\bar{B}^0$ (bottom) events (exp 26, run 1894) for the successive optimization steps of the GNN-based hit filter evaluated at a GNN-cut threshold of 0.2. The table lists the track fitting f_2 score, track fitting efficiency, track fake rate, hit efficiency, and hit background rejection, and compares each configuration to the legacy basf2 MVA and cut-based hit filters. The final configuration after optimization is "bce loss with logits". All uncertainties are statistically correlated due to evaluation on the same samples.

configuration	track f2 (%)	track eff (%)	track fake (%)	hit eff (%)	hit rej (%)
$\mu^+\mu^-(\gamma)$					
basf2 mva	$94.41^{+0.25}_{-0.26}$	$93.79^{+0.57}_{-0.63}$	$1.80^{+0.37}_{-0.31}$	$96.27^{+0.21}_{-0.22}$	$88.04^{+0.09}_{-0.09}$
basf2 cuts from DB	$92.73^{+0.29}_{-0.30}$	$91.86^{+0.66}_{-0.71}$	$2.61^{+0.44}_{-0.38}$	$98.45^{+0.13}_{-0.15}$	$69.67^{+0.13}_{-0.13}$
initial	$94.50^{+0.25}_{-0.26}$	$93.48^{+0.59}_{-0.64}$	$0.33^{+0.18}_{-0.12}$	$96.40^{+0.20}_{-0.22}$	$98.34^{+0.04}_{-0.04}$
updated train-mix	$95.36^{+0.23}_{-0.24}$	$94.78^{+0.53}_{-0.58}$	$1.15^{+0.30}_{-0.24}$	$98.16^{+0.15}_{-0.16}$	$94.45^{+0.06}_{-0.07}$
more input features	$95.21^{+0.23}_{-0.24}$	$94.60^{+0.54}_{-0.59}$	$1.16^{+0.30}_{-0.24}$	$98.59^{+0.13}_{-0.14}$	$96.41^{+0.05}_{-0.05}$
updated cuts	$95.69^{+0.22}_{-0.23}$	$95.09^{+0.51}_{-0.49}$	$0.90^{+0.27}_{-0.21}$	$98.40^{+0.14}_{-0.15}$	$96.21^{+0.05}_{-0.05}$
inter SL edges	$95.51^{+0.23}_{-0.24}$	$94.97^{+0.52}_{-0.57}$	$1.15^{+0.30}_{-0.24}$	$98.54^{+0.13}_{-0.14}$	$96.34^{+0.05}_{-0.05}$
bce loss with logits	$\mathbf{95.58}^{+0.22}_{-0.24}$	$\mathbf{95.03}^{+0.51}_{-0.57}$	$\mathbf{1.28}^{+0.32}_{-0.25}$	$\mathbf{98.91}^{+0.11}_{-0.12}$	$\mathbf{95.26}^{+0.06}_{-0.06}$
larger model	$95.42^{+0.23}_{-0.24}$	$94.97^{+0.52}_{-0.57}$	$1.65^{+0.35}_{-0.29}$	$98.92^{+0.11}_{-0.12}$	$94.54^{+0.06}_{-0.06}$
larger training dataset	$95.18^{+0.23}_{-0.25}$	$94.53^{+0.54}_{-0.59}$	$1.03^{+0.29}_{-0.23}$	$98.82^{+0.12}_{-0.13}$	$96.29^{+0.05}_{-0.05}$

configuration	track f2 (%)	track eff (%)	track fake (%)	hit eff (%)	hit rej (%)
$B^0\bar{B}^0$					
basf2 mva	$79.45^{+0.18}_{-0.19}$	$77.57^{+0.42}_{-0.42}$	$1.84^{+0.15}_{-0.14}$	$93.89^{+0.10}_{-0.11}$	$76.26^{+0.11}_{-0.11}$
basf2 cuts from DB	$78.51^{+0.19}_{-0.19}$	$76.37^{+0.43}_{-0.43}$	$2.03^{+0.16}_{-0.15}$	$95.45^{+0.09}_{-0.09}$	$59.55^{+0.13}_{-0.13}$
initial	$66.28^{+0.22}_{-0.22}$	$61.95^{+0.49}_{-0.49}$	$1.06^{+0.13}_{-0.12}$	$70.11^{+0.20}_{-0.20}$	$90.62^{+0.08}_{-0.08}$
updated train-mix	$79.35^{+0.18}_{-0.19}$	$77.53^{+0.42}_{-0.42}$	$1.89^{+0.15}_{-0.14}$	$95.80^{+0.09}_{-0.09}$	$80.47^{+0.10}_{-0.10}$
more input features	$79.63^{+0.18}_{-0.18}$	$77.91^{+0.42}_{-0.42}$	$1.74^{+0.15}_{-0.13}$	$96.51^{+0.08}_{-0.08}$	$81.29^{+0.10}_{-0.10}$
updated cuts	$79.83^{+0.18}_{-0.18}$	$78.12^{+0.41}_{-0.41}$	$1.63^{+0.14}_{-0.13}$	$96.50^{+0.08}_{-0.08}$	$81.17^{+0.10}_{-0.10}$
inter SL edges	$79.76^{+0.18}_{-0.18}$	$78.12^{+0.41}_{-0.42}$	$1.85^{+0.15}_{-0.14}$	$97.18^{+0.07}_{-0.07}$	$80.65^{+0.10}_{-0.10}$
bce loss with logits	$\mathbf{80.32}^{+0.18}_{-0.18}$	$\mathbf{78.80}^{+0.41}_{-0.42}$	$\mathbf{1.86}^{+0.15}_{-0.14}$	$\mathbf{97.95}^{+0.06}_{-0.06}$	$\mathbf{79.22}^{+0.11}_{-0.11}$
larger model	$80.19^{+0.18}_{-0.18}$	$78.56^{+0.41}_{-0.42}$	$1.92^{+0.15}_{-0.14}$	$97.98^{+0.06}_{-0.06}$	$78.76^{+0.11}_{-0.11}$
larger training dataset	$79.70^{+0.18}_{-0.18}$	$78.09^{+0.41}_{-0.42}$	$1.84^{+0.15}_{-0.14}$	$97.98^{+0.06}_{-0.06}$	$80.62^{+0.10}_{-0.10}$

8.2. Baseline Filtering Comparisons

I assess the performance of the proposed GNN-based hit-filtering algorithm using the same evaluation datasets and track reconstruction configuration as employed during the design-optimization phase, but applied to an enlarged evaluation sample to improve statistical precision. The evaluation dataset consists of 50 000 simulated events each for $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ processes. For the GNN-based filter, I employ the final configuration of the model obtained after the optimization described in section 8.1, using a classification threshold of 0.2. For each background scenario, an independent GNN model is trained on samples with the same background level as that used for the corresponding evaluation. The primary objective of this study is to perform a systematic comparison between the currently used filter based on MVA and the GNN-based filter. For completeness and as a reference, the legacy cut-based filter is also included in the analysis.

8.2.1. Hit filtering performance

In Table 8.4, I present the hit-level performance of the GNN-based filter compared to the cut-based and MVA-based baseline filtering approaches. The corresponding hit-level performance metrics are defined in section 7.2, where a distinction is made between the hit metrics evaluated per track and those evaluated over all hits in an event.

For the low-background case (exp. 22), the GNN filter increases the track-wise hit efficiency $\epsilon_{\text{hit, per track}}$ by 1.41 %pt in the $\mu^+\mu^-(\gamma)$ case and 2.44 %pt in the $B^0\bar{B}^0$ case relative to the MVA filter. Under high-background conditions (exp. 0), the corresponding improvement reaches 5.17 %pt (for $\mu^+\mu^-(\gamma)$) and 5.48 %pt (for $B^0\bar{B}^0$). Simultaneously, the hit purity per track $P_{\text{hit, per track}}$ increases by up to 3.47 %pt (for $\mu^+\mu^-(\gamma)$) and 4.27 %pt (for $B^0\bar{B}^0$).

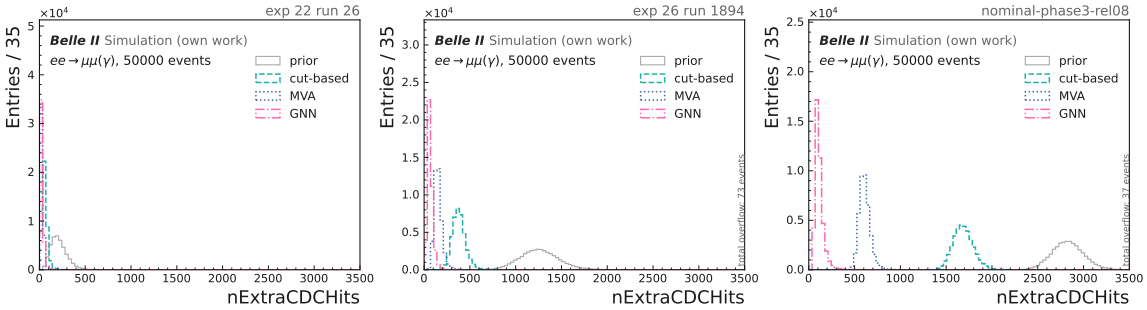
The improvement of the hit efficiency averaged over all hits ϵ_{hit} is comparable in magnitude to the efficiency gain measured per track. In contrast, the hit background rejection rej_{hit} is substantially improved by the GNN filter, with gains of up to 57.44 %pt (for $\mu^+\mu^-(\gamma)$) and 51.85 %pt (for $B^0\bar{B}^0$) relative to cut-based selection and 16.90 %pt (for $\mu^+\mu^-(\gamma)$) and 13.12 %pt (for $B^0\bar{B}^0$) relative to the MVA filter in the high-background configuration.

In absolute values, *e.g.* in the high-background scenario (exp. 0) for $\mu^+\mu^-(\gamma)$ events, the GNN filter increases the hit background rejection from approximately 39 % by the cut-based method and 79 % by the MVA filter to nearly 96 %, while simultaneously increasing the hit efficiency per track from about 87 % (cut-based) and 90 % (MVA) to almost 95 %. In the corresponding $B^0\bar{B}^0$ sample, the GNN filter also improves the hit background rejection from 36 % (cut-based) and 74 % (MVA) to nearly 88 %, again at a distinctly higher hit efficiency per track.

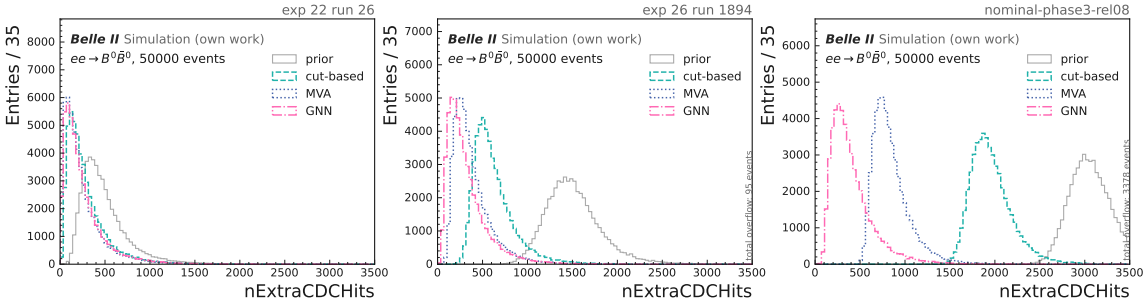
The impact of this increased hit-level filtering performance achieved with the GNN filter on the subsequent track reconstruction is discussed in detail in subsection 8.2.3.

Table 8.4.: Hit-level performance metrics of the GNN filter compared to the default basf2 filtering methods for different background levels. The metrics include hit efficiency and hit purity per track, and overall hit metrics hit efficiency, hit background rejection, and the average number of extra CDC hits $\langle n_{\text{extraCDChits}} \rangle$, with statistical uncertainties indicated. The differences between the GNN filter and other filters are highlighted in green and red next to the given metrics.

Hit Metrics	**per Track Eff.** (%)	**per Track Pur.** (%)	**Efficiency** (%)	**Bkg. Rej.** (%)	$\langle n \rangle_{\text{extraCDChits}}$
$\mu^+\mu^-$ (γ)					
Exp. 22 Run 26					
cut-based	$98.63^{+0.00}_{-0.01}$ +0.30	$99.37^{+0.00}_{-0.01}$ −0.11	$99.19^{+0.09}_{-0.11}$ +0.43	$71.39^{+0.35}_{-0.35}$ +19.35	58^{+31}_{-31} −36
mva	$97.52^{+0.01}_{-0.01}$ +1.41	$99.67^{+0.00}_{-0.01}$ +0.41	$98.18^{+0.14}_{-0.16}$ +1.44	$87.89^{+0.25}_{-0.26}$ +2.85	58^{+20}_{-20} −3
best GNN model	$98.93^{+0.00}_{-0.01}$	$99.26^{+0.00}_{-0.01}$	$99.62^{+0.06}_{-0.08}$	$90.74^{+0.22}_{-0.23}$	22^{+21}_{-21}
Exp. 26 Run 1894					
cut-based	$95.26^{+0.01}_{-0.02}$ +1.84	$96.81^{+0.01}_{-0.02}$ −0.01	$98.45^{+0.13}_{-0.15}$ +0.46	$69.67^{+0.13}_{-0.13}$ +25.24	383^{+89}_{-89} −316
mva	$94.40^{+0.01}_{-0.02}$ +2.70	$98.39^{+0.01}_{-0.01}$ +1.59	$96.27^{+0.21}_{-0.22}$ +2.64	$88.04^{+0.09}_{-0.09}$ +6.87	154^{+48}_{-48} −87
best GNN model	$97.10^{+0.01}_{-0.01}$	$96.80^{+0.01}_{-0.02}$	$98.91^{+0.11}_{-0.12}$	$94.91^{+0.06}_{-0.06}$	66^{+32}_{-32}
Exp. 0 Run 0					
cut-based	$87.22^{+0.02}_{-0.02}$ +7.75	$92.05^{+0.01}_{-0.02}$ +0.74	$98.69^{+0.12}_{-0.13}$ −0.97	$38.42^{+0.09}_{-0.09}$ +57.44	1687^{+114}_{-114} −1572
mva	$89.80^{+0.01}_{-0.02}$ +5.17	$96.26^{+0.01}_{-0.02}$ +3.47	$94.73^{+0.25}_{-0.26}$ +3.99	$78.96^{+0.07}_{-0.08}$ +16.90	610^{+63}_{-63} −496
best GNN model	$94.97^{+0.01}_{-0.02}$	$92.79^{+0.01}_{-0.02}$	$97.72^{+0.16}_{-0.17}$	$95.86^{+0.04}_{-0.04}$	115^{+53}_{-53}
$B^0\bar{B}^0$					
Exp. 22 Run 26					
cut-based	$89.28^{+0.00}_{-0.01}$ +2.06	$97.54^{+0.00}_{-0.01}$ −0.13	$96.81^{+0.07}_{-0.07}$ +2.43	$34.87^{+0.24}_{-0.23}$ +4.98	289^{+230}_{-230} −34
mva	$88.90^{+0.01}_{-0.01}$ +2.44	$97.88^{+0.00}_{-0.00}$ +0.21	$96.28^{+0.08}_{-0.08}$ +2.96	$44.26^{+0.25}_{-0.25}$ −4.41	243^{+220}_{-220} +13
best GNN model	$91.34^{+0.00}_{-0.01}$	$97.41^{+0.00}_{-0.01}$	$99.24^{+0.03}_{-0.04}$	$39.85^{+0.24}_{-0.24}$	256^{+237}_{-237}
Exp. 26 Run 1894					
cut-based	$85.18^{+0.01}_{-0.01}$ +3.83	$94.98^{+0.00}_{-0.01}$ +0.22	$95.45^{+0.09}_{-0.09}$ +2.30	$59.55^{+0.13}_{-0.13}$ +19.52	622^{+231}_{-231} −300
mva	$85.11^{+0.01}_{-0.01}$ +3.90	$96.54^{+0.00}_{-0.01}$ +1.78	$93.89^{+0.10}_{-0.11}$ +3.86	$76.26^{+0.11}_{-0.11}$ +2.81	380^{+213}_{-213} −57
best GNN model	$89.01^{+0.01}_{-0.01}$	$95.20^{+0.00}_{-0.01}$	$97.75^{+0.06}_{-0.07}$	$79.07^{+0.11}_{-0.11}$	323^{+221}_{-221}
Exp. 0 Run 0					
cut-based	$78.83^{+0.01}_{-0.01}$ +7.87	$90.40^{+0.01}_{-0.01}$ +1.44	$96.36^{+0.08}_{-0.09}$ −1.73	$35.76^{+0.09}_{-0.09}$ +51.85	1961^{+236}_{-236} −1553
mva	$81.22^{+0.01}_{-0.01}$ +5.48	$94.67^{+0.00}_{-0.01}$ +4.27	$92.40^{+0.12}_{-0.12}$ +2.23	$74.49^{+0.08}_{-0.08}$ +13.12	844^{+208}_{-208} −436
best GNN model	$86.70^{+0.01}_{-0.01}$	$91.84^{+0.01}_{-0.01}$	$94.63^{+0.10}_{-0.10}$	$87.61^{+0.06}_{-0.06}$	408^{+240}_{-240}



(a) $\mu^+\mu^-$ (γ), exp 22, run 26 (b) $\mu^+\mu^-$ (γ), exp 26, run 1894 (c) $\mu^+\mu^-$ (γ), exp 0, run 0



(d) $B^0\bar{B}^0$, exp 22, run 26 (e) $B^0\bar{B}^0$, exp 26, run 1894 (f) $B^0\bar{B}^0$, exp 0, run 0

Figure 8.34.: Distributions of number of extra CDC hits before (green) and after filtering with the respective filtering methods cut-based (magenta), mva (blue) and GNN (light green) for (a) - (c) 50 000 $\mu^+\mu^-$ (γ) events and (d) - (f) 50 000 $B^0\bar{B}^0$ events for three different background levels.

8.2.2. Number of extra CDC hits

In the following, I examine how the three filtering strategies influence the number of extra CDC hits that are not associated with reconstructed tracks, $\langle n \rangle_{\text{extraCDChits}}$, as defined in subsection 8.2.1. This observable is commonly used as a monitoring quantity for the beam-induced background level and thus provides a complementary measure of the effectiveness of the hit filters compared to the hit-level metrics discussed in section 7.2. The mean values of $\langle n \rangle_{\text{extraCDChits}}$ are summarized in Table 8.4, and the corresponding distributions are shown in Figure 8.34.

Even under low-background conditions (exp. 22), all three filters significantly reduce $\langle n \rangle_{\text{extraCDChits}}$ compared to the unfiltered case, that has 215 hits for $\mu^+\mu^-$ (γ) and 490 hits for $B^0\bar{B}^0$. For the $\mu^+\mu^-(\gamma)$ sample, the GNN-based filter achieves a reduction by an order of magnitude to 22 hits, in contrast to 58 hits obtained with each of the two default filtering algorithms. For the $B^0\bar{B}^0$ sample, the performance of the GNN-based filter lies between that of the existing filters.

At higher background levels, the GNN-based filter yields a substantially stronger reduction of $\langle n \rangle_{\text{extraCDChits}}$ than the baseline filters in both benchmark channels. In the $\mu^+\mu^-(\gamma)$ sample, the GNN reduces $\langle n \rangle_{\text{extraCDChits}}$ from 2 828 to 115, compared to reductions to 1 687

and 610 achieved by the cut-based and MVA algorithms, respectively. In the $B^0\bar{B}^0$ sample, the GNN-based filter reduces $\langle n \rangle_{\text{extraCDChits}}$ from 3 086 to 408, whereas the baseline filters reduce it to 1 961 (cut-based) and 844 (MVA).
In the high-background scenario, the GNN-based approach therefore improves the filtering performance by a factor of 2-4 relative to the MVA filter and by a factor of 4-15 relative to the cut-based filter, depending on the evaluation sample.

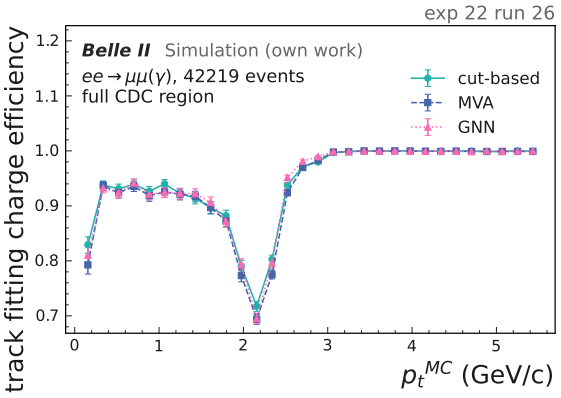
8.2.3. Effect on tracking performance

In Table 8.5, I summarize the impact of the different filtering strategies on the quality of the reconstructed tracks for $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events in the three considered background scenarios. The GNN filter consistently yields the highest track fitting charge efficiency, with the relative improvement becoming particularly pronounced at high background levels, where it recovers a substantial fraction of tracks that are lost by both the cut-based and the MVA filters: ϵ_{fitted} improves by 20.04 %pt in the $\mu^+\mu^-(\gamma)$ channel and 12.75 %pt in the $B^0\bar{B}^0$ channel compared to the cut-based filter and by 5.40 %pt (for $\mu^+\mu^-(\gamma)$) and 3.02 %pt (for $B^0\bar{B}^0$) compared to the MVA filter.
At the same time, it maintains a competitive fake rate: for $\mu^+\mu^-(\gamma)$ events in the high-background scenario, the fake rate is reduced by 0.94 %pt with respect to the cut-based filter and by 0.51 %pt relative to the MVA filter. For $B^0\bar{B}^0$ events, a moderate increase in fake rate of up to 0.19 %pt is observed compared to the MVA filter, while it decreases by up to 0.77 %pt with respect to the cut-based filter. Especially in the $B^0\bar{B}^0$ case, an increase in the clone rate of up to 0.27 %pt is observed across all background scenarios for the GNN filter. One possible explanation is that the stronger background rejection at higher hit efficiency can lead to ambiguously assigned hits in the vicinity of true tracks, which may in turn lead to the reconstruction of more than one compatible candidate for the same underlying trajectory. Nevertheless, the clone rate remains below the percent level and therefore constitutes a small effect compared to the gains in track fitting charge efficiency. A straightforward strategy to reduce both the fake rate and the clone rate would be to apply a more stringent GNN classification threshold, at the expense of a reduced charge efficiency for the fitted tracks.
As an additional consequence of the increased hit efficiency that was evaluated on the same events, a slight improvement in the transverse momentum resolution and in the z_0 resolution is observed relative to the baseline filters.

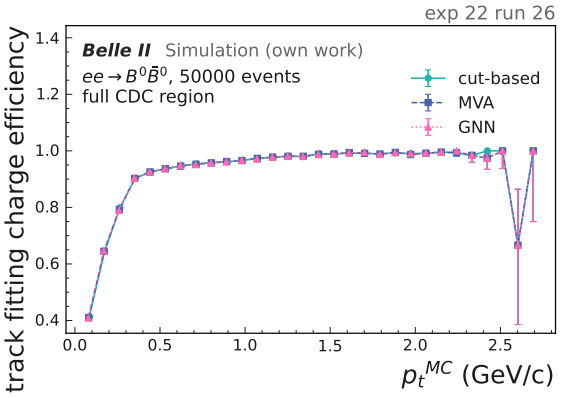
A more comprehensive insight into the charge efficiency performances as a function of the transverse momentum p_T^{MC} and the dip angle λ^{MC} is given in Figure 8.35 and Figure 8.36. Across all scenarios, the charge reconstruction efficiency exhibits a general increase with increasing p_T^{MC}. This behavior is expected, as particle trajectories with higher transverse momentum are less curved in the magnetic field and are therefore easier to reconstruct than low-p_T tracks. When comparing the GNN filter (magenta) with the baseline filters (green and blue) in the high background scenario (exp. 0), in both benchmark channels the performance towards low p_T values is improved whereas the difference for higher p_T values is marginal in both benchmark channels. In particular, in the $\mu^+\mu^-(\gamma)$ case the steep performance drop between 2 and 3 GeV/c is partially recovered

Table 8.5.: Track fit performance metrics of different filtering methods evaluated on 50 000 $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events for different background conditions. The metrics include track fitting charge efficiency, track fake rate, track clone rate, transverse momentum resolution p_T (r_{68}), and resolution of the z-coordinate of the point-of-closest-approach z_0 (r_{68}), with statistical uncertainties indicated. The differences between the GNN filter and the default basf2 filters are highlighted in green and red next to the given metrics.

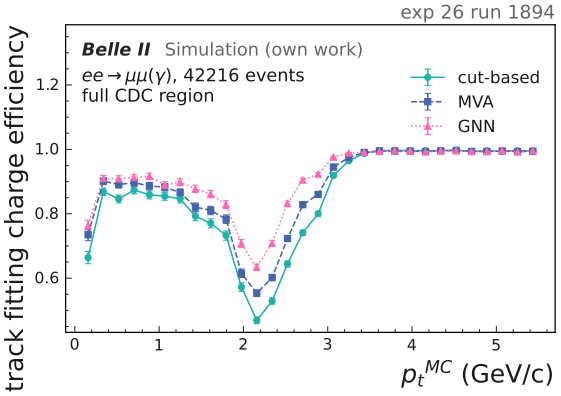
Track Metrics	**Efficiency** (%)	**Fake Rate** (%)	**Clone Rate** (%)	p_T **Res.** (%)	z_0 **Res.** (cm)
$\mu^+\mu^-(\gamma)$					
Exp. 22 Run 26					
cut-based	$96.18^{+0.07}_{-0.07}$ −0.01	$0.40^{+0.02}_{-0.02}$ −0.12	$0.03^{+0.01}_{-0.01}$ +0.01	$0.76^{+0.01}_{-0.00}$ +0.01	$0.21^{+0.00}_{-0.00}$
mva	$95.85^{+0.07}_{-0.07}$ +0.32	$0.27^{+0.02}_{-0.02}$ +0.01	$0.03^{+0.01}_{-0.01}$ +0.01	$0.78^{+0.00}_{-0.00}$ −0.01	$0.21^{+0.00}_{-0.00}$
best GNN model	$96.17^{+0.07}_{-0.07}$	$0.28^{+0.02}_{-0.02}$	$0.04^{+0.01}_{-0.01}$	$0.77^{+0.01}_{-0.00}$	$0.21^{+0.00}_{-0.00}$
Exp. 26 Run 1894					
cut-based	$90.08^{+0.11}_{-0.11}$ +3.86	$2.00^{+0.05}_{-0.05}$ −0.97	$0.02^{+0.01}_{-0.00}$ +0.01	$0.95^{+0.01}_{-0.01}$ +0.04	$0.29^{+0.00}_{-0.00}$
mva	$91.85^{+0.10}_{-0.10}$ +2.09	$1.37^{+0.04}_{-0.04}$ −0.34	$0.02^{+0.01}_{-0.00}$ +0.01	$1.01^{+0.01}_{-0.01}$ −0.02	$0.30^{+0.00}_{-0.00}$ −0.01
best GNN model	$93.94^{+0.08}_{-0.09}$	$1.03^{+0.04}_{-0.04}$	$0.03^{+0.01}_{-0.01}$	$0.99^{+0.01}_{-0.01}$	$0.29^{+0.00}_{-0.00}$
Exp. 0 Run 0					
cut-based	$73.40^{+0.16}_{-0.16}$ +20.04	$6.81^{+0.10}_{-0.10}$ −0.94	$0.02^{+0.01}_{-0.01}$ +0.01	$0.72^{+0.00}_{-0.00}$ +0.08	$0.29^{+0.00}_{-0.00}$ −0.05
mva	$88.04^{+0.11}_{-0.12}$ +5.40	$6.38^{+0.09}_{-0.09}$ −0.51	$0.02^{+0.01}_{-0.00}$ +0.01	$0.82^{+0.00}_{-0.00}$ −0.02	$0.26^{+0.00}_{-0.00}$ −0.02
best GNN model	$93.44^{+0.09}_{-0.09}$	$5.87^{+0.08}_{-0.08}$	$0.03^{+0.01}_{-0.01}$	$0.80^{+0.00}_{-0.00}$	$0.24^{+0.00}_{-0.00}$
$B^0\bar{B}^0$					
Exp. 22 Run 26					
cut-based	$79.46^{+0.06}_{-0.06}$ +0.37	$1.59^{+0.02}_{-0.02}$ +0.03	$1.02^{+0.02}_{-0.01}$ +0.24	$0.95^{+0.00}_{-0.00}$ −0.01	$0.37^{+0.00}_{-0.00}$
mva	$79.62^{+0.06}_{-0.06}$ +0.21	$1.41^{+0.02}_{-0.02}$ +0.21	$1.08^{+0.02}_{-0.02}$ +0.18	$1.08^{+0.00}_{-0.00}$ −0.13	$0.41^{+0.00}_{-0.00}$ −0.04
best GNN model	$79.83^{+0.06}_{-0.06}$	$1.62^{+0.02}_{-0.02}$	$1.26^{+0.02}_{-0.02}$	$0.95^{+0.00}_{-0.00}$	$0.37^{+0.00}_{-0.00}$
Exp. 26 Run 1894					
cut-based	$74.24^{+0.06}_{-0.06}$ +2.26	$2.07^{+0.02}_{-0.02}$ −0.22	$0.61^{+0.01}_{-0.01}$ +0.27	$0.87^{+0.00}_{-0.00}$ +0.02	$0.47^{+0.00}_{-0.00}$ −0.03
mva	$75.81^{+0.06}_{-0.06}$ +0.69	$1.68^{+0.02}_{-0.02}$ +0.17	$0.64^{+0.01}_{-0.01}$ +0.24	$1.03^{+0.00}_{-0.00}$ −0.14	$0.52^{+0.00}_{-0.00}$ −0.08
best GNN model	$76.50^{+0.06}_{-0.06}$	$1.85^{+0.02}_{-0.02}$	$0.88^{+0.01}_{-0.01}$	$0.89^{+0.00}_{-0.00}$	$0.44^{+0.00}_{-0.00}$
Exp. 0 Run 0					
cut-based	$61.69^{+0.07}_{-0.07}$ +12.75	$3.74^{+0.03}_{-0.03}$ −0.77	$0.51^{+0.01}_{-0.01}$ +0.17	$0.79^{+0.00}_{-0.00}$	$0.51^{+0.00}_{-0.00}$ −0.14
mva	$71.42^{+0.06}_{-0.06}$ +3.02	$2.78^{+0.03}_{-0.03}$ +0.19	$0.52^{+0.01}_{-0.01}$ +0.16	$0.99^{+0.00}_{-0.00}$ −0.20	$0.49^{+0.00}_{-0.00}$ −0.12
best GNN model	$74.44^{+0.06}_{-0.06}$	$2.97^{+0.03}_{-0.03}$	$0.68^{+0.01}_{-0.01}$	$0.79^{+0.00}_{-0.00}$	$0.37^{+0.00}_{-0.00}$



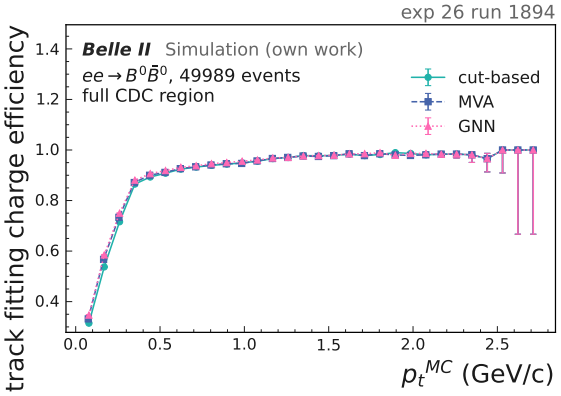
(a) $\mu^+\mu^-$ (γ), exp 22, run 26



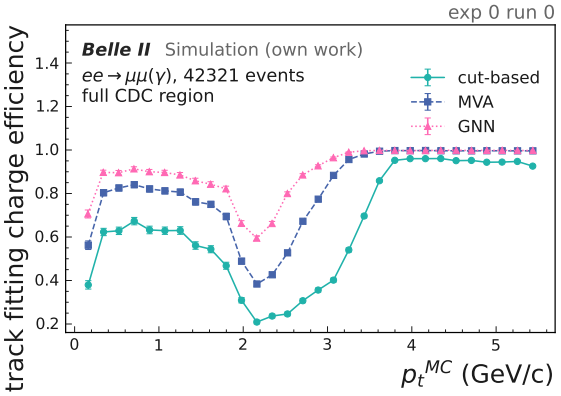
(b) $B^0\bar{B}^0$, exp 22, run 26



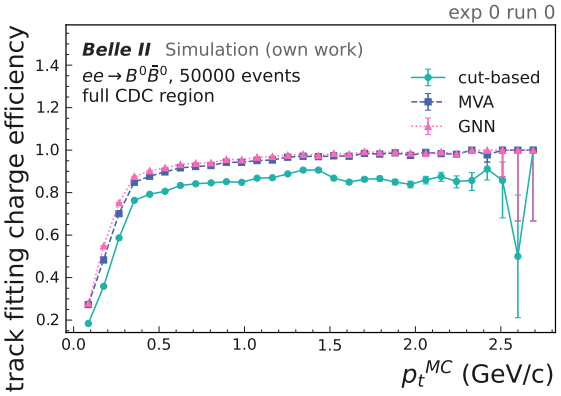
(c) $\mu^+\mu^-$ (γ), exp 26, run 1894



(d) $B^0\bar{B}^0$, exp 26, run 1894



(e) $\mu^+\mu^-$ (γ), exp 0, run 0

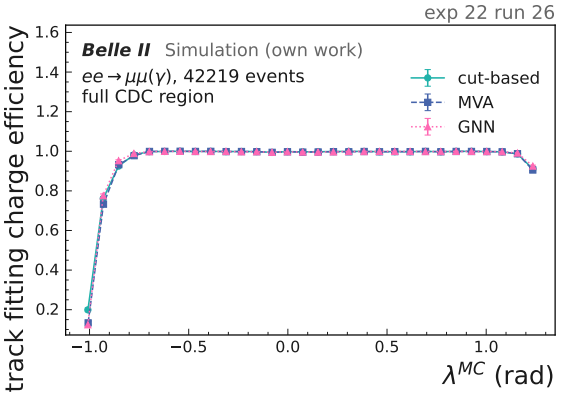


(f) $B^0\bar{B}^0$, exp 0, run 0

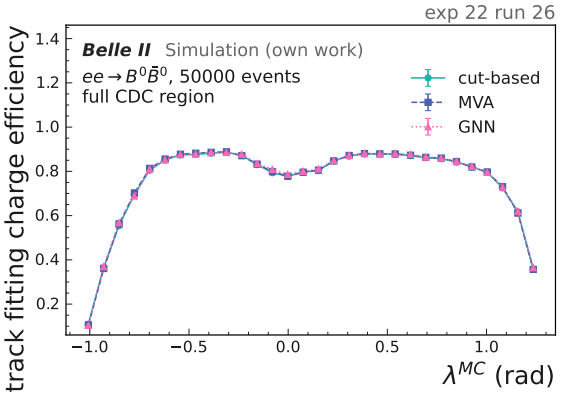
Figure 8.35.: Track fitting charge efficiency over the transverse momentum p_T^{MC} for the full detector range comparing cut-based filtering (green), MVA filtering (blue) with GNN filtering (magenta) for 50 000 simulated $B^0\bar{B}^0$ events and 50 000 simulated $\mu^+\mu^-(\gamma)$ events for three different background levels. All three filters are applied to an identical set of events, thereby inducing statistical correlations among their outputs. The efficiencies are obtained for all tracks that leave at least seven hits in the CDC. The full CDC acceptance region is used.

by the GNN filter. This decrease is attributable to the fact that tracks originating from $\mu^+\mu^-(\gamma)$ events in this p_T range occur predominantly in the end-cap regions, where their classification is intrinsically more challenging. The improved performance in the low-p_T region indicates that the GNN filter provides a more robust track reconstruction across the full p_T range and effectively restores tracking performance in the low-p_T region.

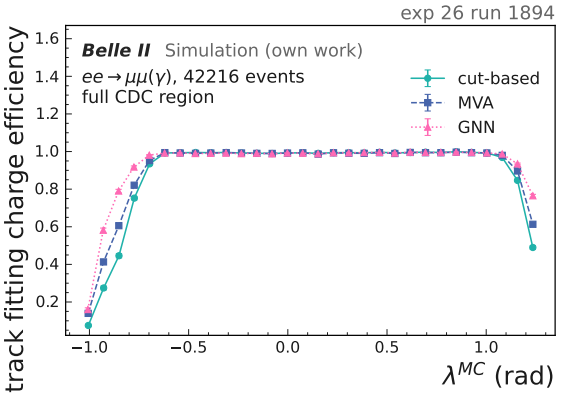
The dependence of the charge efficiency on the detector region, parametrized by the polar-angle-dependent dip angle $\lambda(\theta) = 0.5\pi - \theta$, is illustrated in Figure 8.36. The most significant performance gains achieved by the GNN-based filter are observed in the end-cap regions at small and large values of λ^{MC}, where the reduced number of detector layers traversed and the increased background levels make the tracking particularly challenging. The reduction in $B^0\bar{B}^0$ events in the vicinity of $\lambda = 0$ is predominantly attributable to the fact that the corresponding tracks are most likely low-p_T trajectories exhibiting significant curvature. An additional contributing factor is that, in this particular detector region close to the IP, the charge assignment is more prone to mis-classification. Consequently, the dip is less pronounced and becomes nearly invisible when considering only the track fitting or track finding efficiency, rather than the track fitting charge efficiency as presented here. In particular, in the high-background scenario (exp. 0) for $\mu^+\mu^-(\gamma)$ events, the GNN filter recovers a substantial fraction of the charge efficiency in the outermost λ^{MC} bins. For instance, in the backward end-cap region the charge efficiency increases from approximately 21 % (cut-based) and 55 % (MVA) to 75 % with the GNN filter for the $\mu^+\mu^-(\gamma)$ sample, and from about 22 % (cut-based) and 44 % (MVA) to 52 % with the GNN filter corresponding to an absolute improvement of up to 20 %pt and 8 %pt compared to the MVA filter and even larger improvements compared to the cut-based filter.
A comprehensive overview of the charge efficiencies separated for each of the individual detector regions is provided in Table A.1, while the region-resolved dependencies on p_T^{MC} and λ^{MC} are displayed in the appendix (A.2-A.5).



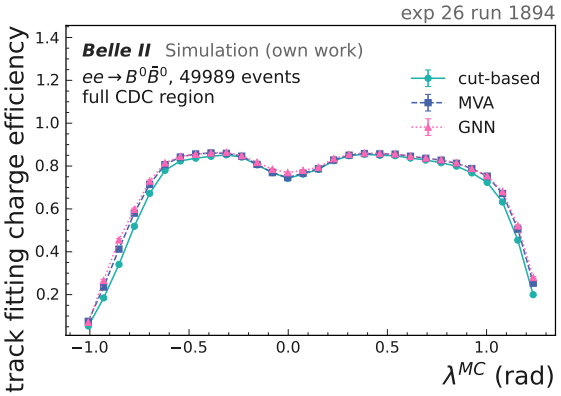
(a) $\mu^+\mu^-$ (γ), exp 22, run 26



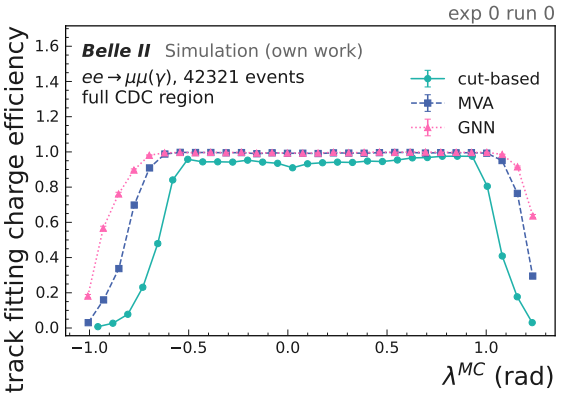
(b) $B^0\bar{B}^0$, exp 22, run 26



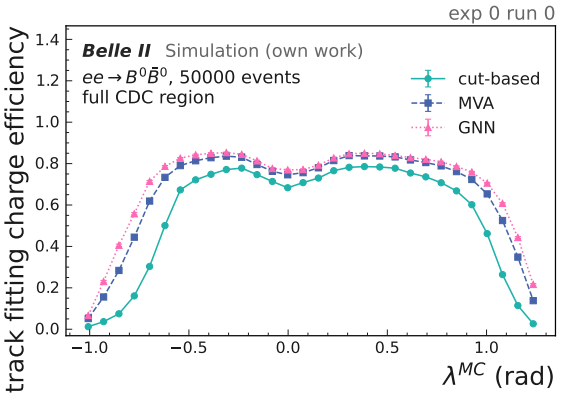
(c) $\mu^+\mu^-$ (γ), exp 26, run 1894



(d) $B^0\bar{B}^0$, exp 26, run 1894

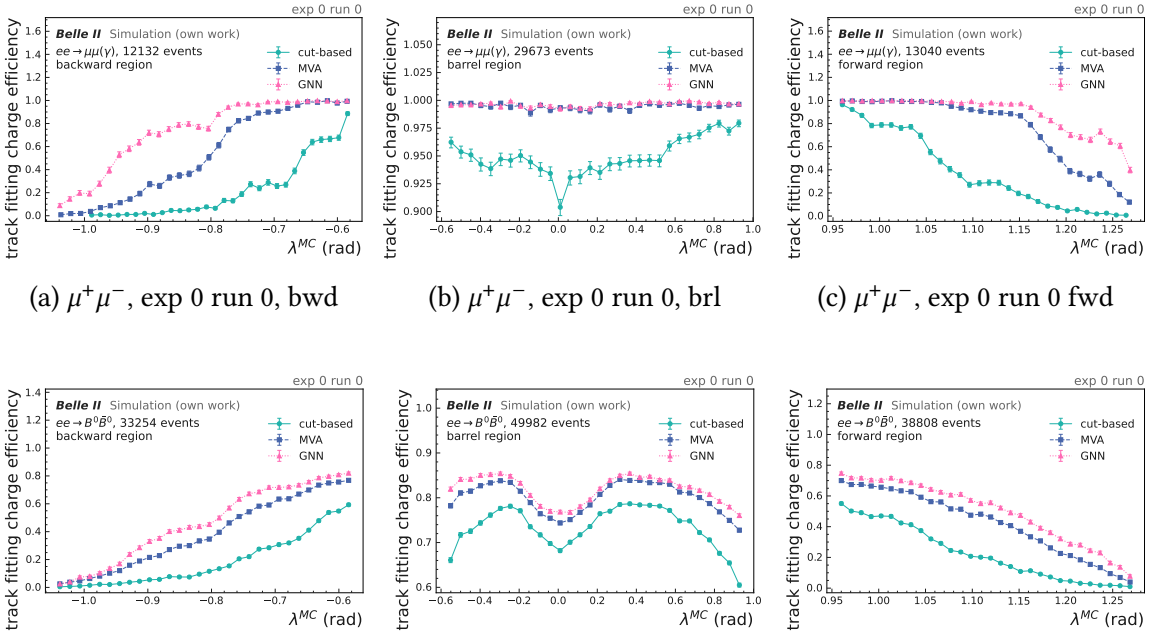


(e) $\mu^+\mu^-$ (γ), exp 0, run 0



(f) $B^0\bar{B}^0$, exp 0, run 0

Figure 8.36.: Track fitting charge efficiency over the dip angle λ^{MC} for the full detector range comparing cut-based filtering (green), MVA filtering (blue) with GNN filtering (magenta) for 50 000 $B^0\bar{B}^0$ events and 50 000 $\mu^+\mu^-$ (γ) events for three different background levels.



(a) $\mu^+\mu^-$, exp 0 run 0, bwd (b) $\mu^+\mu^-$, exp 0 run 0, brl (c) $\mu^+\mu^-$, exp 0 run 0 fwd

(d) $B^0\bar{B}^0$, exp 0 run 0, bwd (e) $B^0\bar{B}^0$, exp 0 run 0, brl (f) $B^0\bar{B}^0$, exp 0 run 0 fwd

Figure 8.37.: Track fitting charge efficiency over the dip angle λ^{MC} for three different detector regions, backward (bwd), barrel (brl) and forward (fwd), comparing cut-based filtering (green) and MVA filtering (blue) with GNN filtering (magenta) for 50 000 $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events for the high-background scenario (exp. 0). All three filters are applied to an identical set of events, thereby inducing statistical correlations among their outputs. The efficiencies are obtained for all tracks that leave at least seven hits in the CDC. The scaling of the y-axis is chosen to be non-uniform across the different detector regions in order to increase the visibility of the displayed distributions.

Table 8.6.: Track fitting charge efficiency for pions and protons originating from K_S^0 ($\rightarrow \pi^+\pi^-$) and Λ ($\rightarrow p\ \pi^-$) in 50 000 $B^0\bar{B}^0$ events, with the final-state particle type resulting from these decays highlighted in bold font.

	K_S^0 ($\rightarrow \boldsymbol{\pi^+\pi^-}$) (in %)	Λ ($\rightarrow \mathbf{p} + \pi^-$) (in %)	Λ ($\rightarrow p + \boldsymbol{\pi^-}$) (in %)
Exp. 22 Run 26			
cut-based	$76.67^{+0.19}_{-0.19}$ +0.90	$89.17^{+0.65}_{-0.69}$ +0.18	$46.35^{+1.09}_{-1.08}$ +2.70
mva	$77.23^{+0.19}_{-0.19}$ +0.34	$86.36^{+0.72}_{-0.75}$ +2.99	$48.24^{+1.09}_{-1.09}$ +0.91
best GNN model	$77.57^{+0.19}_{-0.19}$	$89.35^{+0.64}_{-0.68}$	$49.05^{+1.09}_{-1.09}$
Exp. 26 Run 1894			
cut-based	$69.09^{+0.21}_{-0.21}$ +4.08	$82.88^{+0.80}_{-0.83}$ +1.08	$37.51^{+1.07}_{-1.06}$ +5.17
mva	$71.62^{+0.20}_{-0.20}$ +1.55	$82.97^{+0.80}_{-0.83}$ +0.99	$40.99^{+1.09}_{-1.08}$ +1.69
best GNN model	$73.17^{+0.20}_{-0.20}$	$83.96^{+0.78}_{-0.81}$	$42.68^{+1.09}_{-1.08}$
Exp. 0 Run 0			
cut-based	$52.19^{+0.23}_{-0.23}$ +18.18	$64.56^{+1.04}_{-1.05}$ +11.43	$23.59^{+0.95}_{-0.92}$ +11.45
mva	$65.08^{+0.21}_{-0.22}$ +5.29	$75.37^{+0.93}_{-0.95}$ +0.62	$31.51^{+1.03}_{-1.01}$ +3.53
best GNN model	$70.37^{+0.21}_{-0.21}$	$75.99^{+0.92}_{-0.95}$	$35.04^{+1.06}_{-1.04}$

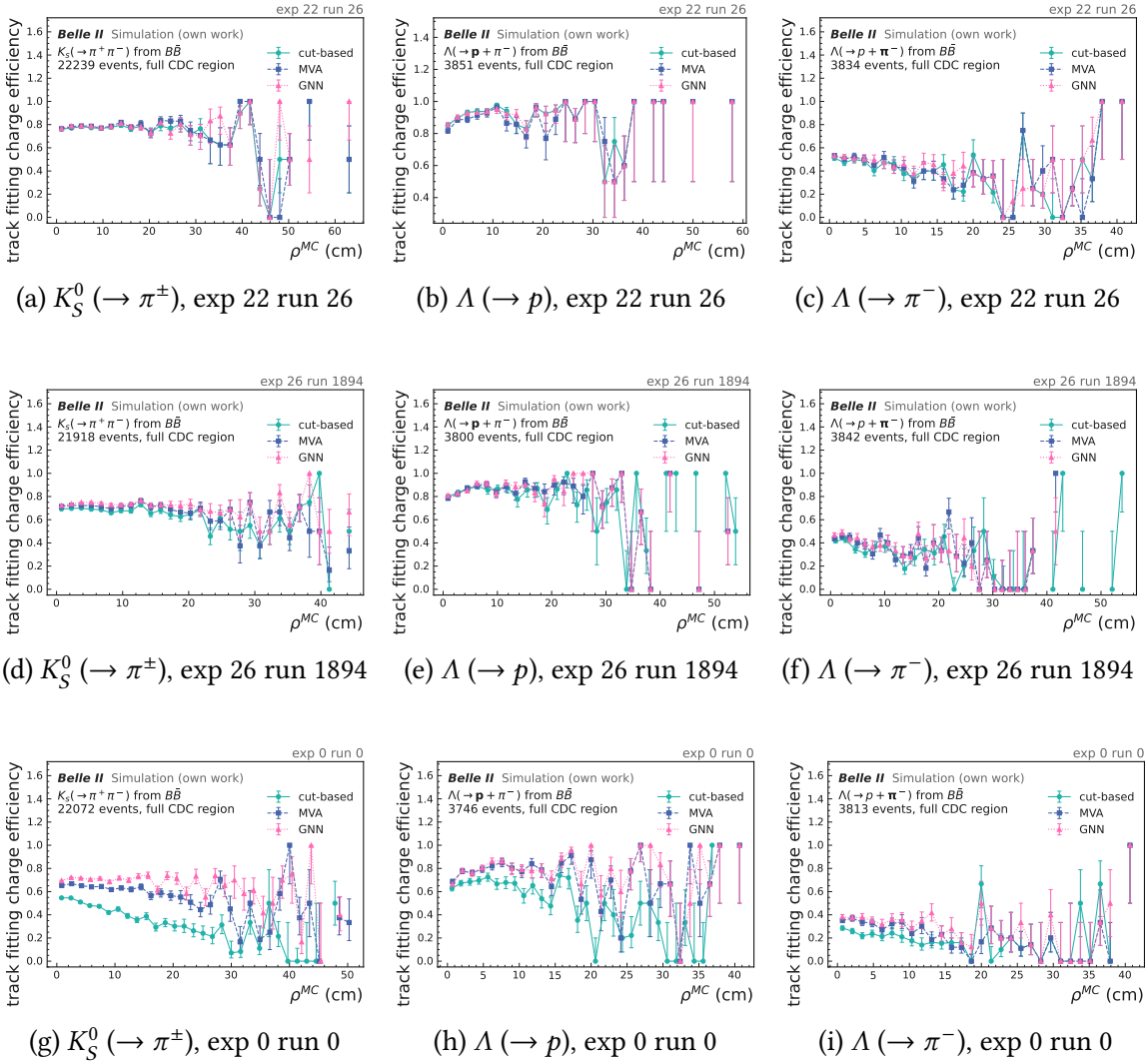
8.2.4. Charge efficiencies for displaced K_S^0 and Λ decays

The charge efficiencies for pions and protons originating from the decays $B \rightarrow XK_S^0 \rightarrow \pi^+\pi^-$ and $B \rightarrow X\Lambda \rightarrow p\pi^-$ (corresponding to categories 13 and 14 in subsection 6.1.3) are summarized in Table 8.6. In addition, their dependence on the displacement of the decay vertex, quantified by the variable ρ^{MC}, is presented in Figure 8.38. The efficiencies are determined with no explicit constraint on the decay-vertex position.

In all scenarios, the GNN filter outperforms the default filters. The highest improvement can be observed in the high-background scenario (exp. 0) for pions originating from K_S^0 with an improvement of 18.18 %pt over the cut-based filter and 5.29 %pt over the MVA filter, while for $\Lambda \rightarrow p\pi^-$ the maximum gain reaches about 11.4 %pt (cut-based) and 3.5 %pt (mva).

The overall efficiency for pions from Λ decays is consistently lower than that for the other final-state particles. This behavior is primarily driven by differences in the p_T spectra of the tracks. The mean p_T of pions from Λ decays is (110 ± 50) MeV/c, compared to (410 ± 260) MeV/c for protons from Λ decays and (330 ± 220) MeV/c for pions from K_S^0 decays, which directly translates into the observed efficiency differences. Charge efficiencies as functions of p_T^{MC} and the dip angle λ^{MC} are provided in the appendix (A.6-A.7).

The displaced decay analysis for $K_S^0 \rightarrow \pi^+\pi^-$ and $\Lambda \rightarrow p\pi^-$ demonstrates that the GNN-based filter not only improves track reconstruction for prompt trajectories as discussed in subsection 8.2.3, but, in particular, also maintains and even improves the reconstruction efficiency for the decay products of long-lived particles with displaced decay vertices. Consequently, the GNN-based hit filtering does not introduce a bias against displaced



(a) K_S^0 ($\to \pi^\pm$), exp 22 run 26 (b) Λ ($\to p$), exp 22 run 26 (c) Λ ($\to \pi^-$), exp 22 run 26

(d) K_S^0 ($\to \pi^\pm$), exp 26 run 1894 (e) Λ ($\to p$), exp 26 run 1894 (f) Λ ($\to \pi^-$), exp 26 run 1894

(g) K_S^0 ($\to \pi^\pm$), exp 0 run 0 (h) Λ ($\to p$), exp 0 run 0 (i) Λ ($\to \pi^-$), exp 0 run 0

Figure 8.38.: Track fitting charge efficiency as a function of the displacement of the K_S^0 and Λ decay vertices ρ^{MC} for pion and proton tracks from K_S^0 and Λ decays from 50 000 $B^0\bar{B}^0$ events comparing cut-based filtering (green) and mva filtering (blue) with GNN filtering (magenta) for for three different background levels. The following selections are applied: $n_{\text{CDCHitsperTrack}} \geq 7$. The scaling of the y-axis is chosen to be non-uniform across the different detector regions in order to increase the visibility of the displayed distributions.

topologies and remains robust for tracks originating at large distances from the interaction point.

8.2.5. Processing time analysis

An important aspect in the practical implementation of the GNN filter within basf2 is its computational overhead. Regardless of the potentially improved filtering quality, an algorithm that is excessively slow cannot be used in Belle II analyzes processing millions of events, nor in the HLT, where the default basf2 reconstruction chain is executed for event triggering. No strict upper limit is imposed on the processing time. Nevertheless, as a practical design guideline, the latency of the hit-filtering stage should remain subdominant relative to the track finding and track fitting steps.
The processing time of the three filtering strategies is evaluated on 50 000 events each for the $\mu^+\mu^-(\gamma)$ and the $B^0\bar{B}^0$ sample, as shown in Table 8.7, by inserting the GNN filter into the `CDCWireHitPreparer` module in basf2 at the same position in the track reconstruction chain where the default filters are applied.
The reported values and uncertainties are obtained from the basf2 statistics module. The large uncertainties primarily reflect statistical fluctuations arising from event-to-event variability and from variations in the computational load of the machine on which the evaluation was executed. The basf2 statistics module estimates processing times from the distribution of per-call processing durations, which in this study exhibit a very broad range due to strong dependencies on event topology and background conditions. A small number of computationally very expensive events dominates the variance, leading to uncertainties that are large compared to the corresponding mean values. Despite these large uncertainties, consistent qualitative trends are visible, which I consider sufficient for the purposes of this study.
The GNN filter is implemented in C++, with the trained GNN model imported through `ONNX` and executed using `ONNX Runtime`. The results demonstrate that, despite the current GNN implementation in C++ not being optimized for processing time, the overall tracking time remains comparable to or only moderately larger than that achieved with the default filters, while still delivering the increased physics performance discussed in the preceding sections.
For $\mu^+\mu^-(\gamma)$ events, the GNN filter significantly reduces the track finding time at medium and high background occupancies relative to the baseline approaches, *e.g.* for experiment 0 from 210 ms (cut-based) and 42 ms (MVA) to 20 ms per call. An analogous behavior is observed for $B^0\bar{B}^0$ events, where the track finding stage benefits from the more aggressive background suppression provided by the GNN-based filter. This improvement is particularly pronounced under high-background conditions, yielding a reduction in processing time of 271 ms and 33 ms relative to the cut-based and MVA filters, respectively.
In contrast, the track fitting time tends to increase, especially when compared to the MVA filter. The main reason for the longer track fitting time are the higher track finding and hit efficiencies, *i.e.* more tracks with more related hits need to be fitted. For example, in the high background scenario (exp. 0) for $B^0\bar{B}^0$ events, the average processing time per individual track is 22.3 ms for the cut-based filter, 19.3 ms for the MVA-based filter,

Table 8.7.: Average processing times per event of the filtering, track finding, and track fitting steps, and the total tracking time, for different background levels and filtering methods. Times are given in ms per module call with uncertainties obtained from the basf2 statistics module. In the majority of cases, the track finding time is reduced when employing the GNN-based filter in comparison to the baseline filters. However, the track fitting time is increased, which can be attributed to the higher track finding efficiency achieved by the GNN filter. The differences between the GNN filter and other filters are highlighted in green and red next to the given metrics.

Timing Metrics	$t_{\text{Filtering}}$ (ms/call)	t_{Finding} (ms/call)	t_{Fitting} (ms/call)	t_{Total} (ms/call)
$\mu^+\mu^-$ (γ)				
Exp. 22 Run 26				
cut-based	0.7 ± 0.3 +5.7	6.1 ± 3.2 −1.9	51.4 ± 21.0 −7.4	81.8 ± 78.6 −3.6
mva	1.0 ± 0.4 +5.4	5.0 ± 2.3 −0.8	50.3 ± 19.5 −6.3	78.2 ± 66.8 0.0
best GNN model	6.4 ± 0.6	4.2 ± 1.9	44.0 ± 16.0	78.2 ± 67.9
Exp. 26 Run 1894				
cut-based	1.9 ± 0.3 +7.4	25.2 ± 12.0 −16.2	45.5 ± 26.8 −2.4	96.4 ± 70.6 −9.3
mva	3.1 ± 0.6 +6.2	11.8 ± 5.3 −2.8	39.6 ± 15.9 +3.5	77.8 ± 69.0 +9.3
best GNN model	9.3 ± 1.2	9.0 ± 2.9	43.1 ± 18.5	87.1 ± 68.2
Exp. 0 Run 0				
cut-based	3.7 ± 0.4 +13.2	209.8 ± 44.0 −190.3	115.0 ± 77.3 −68.3	346.2 ± 101.9 −237.2
mva	6.2 ± 0.5 +10.7	41.6 ± 9.9 −22.1	39.3 ± 18.2 +7.4	107.3 ± 60.1 +1.7
best GNN model	16.9 ± 1.4	19.5 ± 5.7	46.7 ± 24.7	109.0 ± 61.9
$B^0\bar{B}^0$				
Exp. 22 Run 26				
cut-based	1.9 ± 0.6 +9.3	45.1 ± 35.0 +0.8	186.7 ± 55.7 −2.7	609.3 ± 433.3 −44.9
mva	2.7 ± 0.7 +8.5	40.5 ± 31.2 +5.4	183.4 ± 55.0 +0.6	596.5 ± 435.2 +67.8
best GNN model	11.2 ± 2.2	45.9 ± 37.1	204.0 ± 60.8	664.3 ± 460.1
Exp. 26 Run 1894				
cut-based	3.0 ± 0.6 +10.4	75.1 ± 41.2 −25.5	166.8 ± 50.0 +13.2	519.2 ± 404.7 +43.6
mva	4.7 ± 0.8 +8.7	49.5 ± 31.0 +17.3	162.1 ± 48.6 +17.9	485.4 ± 362.4 +77.4
best GNN model	13.4 ± 2.2	49.6 ± 34.9	180.0 ± 53.2	562.8 ± 785.3
Exp. 0 Run 0				
cut-based	4.8 ± 0.7 +17.8	337.4 ± 96.7 −270.6	160.8 ± 59.5 +16.8	753.1 ± 388.1 −173.4
mva	8.2 ± 1.6 +14.4	99.7 ± 43.6 −32.9	156.8 ± 51.9 +20.8	532.0 ± 387.0 +47.7
best GNN model	22.6 ± 3.0	66.8 ± 39.3	177.6 ± 53.9	579.7 ± 375.0

Table 8.8.: Track fit performance metrics evaluated for exp. 37, run 1893 background conditions comparing a GNN trained on exp. 37 with a GNN trained on exp. 26 and the baseline filters (cut-based and MVA). The track metrics are evaluated on 50 000 $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events for exp. 37, run 1893 background conditions. The metrics include track fitting charge efficiency, track fake rate, track clone rate, transverse momentum resolution p_T (r_{68}), and resolution of the z-coordinate of the point-of-closest-approach z_0 (r_{68}), with statistical uncertainties indicated.

Track Metrics	**Efficiency** (%)	**Fake Rate** (%)	**Clone Rate** (%)	p_T **Res.** (%)	z_0 **Res.** (cm)
$\mu^+\mu^-(\gamma)$					
Exp. 37 Run 1893					
cut-based	$89.00^{+0.11}_{-0.11}$	$2.53^{+0.06}_{-0.06}$	$0.03^{+0.01}_{-0.01}$	$0.86^{+0.01}_{-0.00}$	$0.28^{+0.00}_{-0.00}$
mva	$91.67^{+0.10}_{-0.10}$	$1.78^{+0.05}_{-0.05}$	$0.04^{+0.01}_{-0.01}$	$0.92^{+0.01}_{-0.00}$	$0.28^{+0.00}_{-0.00}$
GNN trained on exp. 37	$93.47^{+0.09}_{-0.09}$	$1.33^{+0.04}_{-0.04}$	$0.05^{+0.01}_{-0.01}$	$0.90^{+0.01}_{-0.01}$	$0.27^{+0.00}_{-0.00}$
GNN trained on exp. 26	$93.39^{+0.09}_{-0.09}$	$1.13^{+0.04}_{-0.04}$	$0.03^{+0.01}_{-0.01}$	$0.90^{+0.01}_{-0.01}$	$0.27^{+0.00}_{-0.00}$
$B^0\bar{B}^0$					
Exp. 37 Run 1893					
cut-based	$73.62^{+0.06}_{-0.06}$	$2.40^{+0.02}_{-0.02}$	$0.86^{+0.01}_{-0.01}$	$0.89^{+0.00}_{-0.00}$	$0.46^{+0.00}_{-0.00}$
mva	$75.73^{+0.06}_{-0.06}$	$1.88^{+0.02}_{-0.02}$	$0.83^{+0.01}_{-0.01}$	$1.06^{+0.00}_{-0.00}$	$0.50^{+0.00}_{-0.00}$
GNN trained on exp. 37	$76.63^{+0.06}_{-0.06}$	$2.04^{+0.02}_{-0.02}$	$0.91^{+0.01}_{-0.01}$	$0.91^{+0.01}_{-0.01}$	$0.43^{+0.00}_{-0.00}$
GNN trained on exp. 26	$76.57^{+0.06}_{-0.06}$	$2.00^{+0.02}_{-0.02}$	$0.92^{+0.01}_{-0.01}$	$0.90^{+0.00}_{-0.00}$	$0.43^{+0.00}_{-0.00}$

and 20.6 ms for the GNN-based filter. The remaining processing time difference for the track fitting between the MVA and the GNN filters can be attributed to the higher hit efficiency and the correspondingly larger number of associated CDC hits per track that must be processed, as reported in Table 8.4. Overall, the total tracking time with the GNN filter included remains within a factor of order unity of the default configuration, although the current C++/`ONNX Runtime` implementation has not yet been optimized for computational performance. The dominant contribution to the increased processing time arises from the track fitting stage, which is a direct consequence of the improved track finding efficiency and is therefore not straightforward to reduce without compromising physics performance.

8.3. Background dependence

In the current approach, a distinct expert model must be trained and, when necessary, further optimized for each specific background scenario. In an ideal setting, a single model would exhibit sufficient generalizability to perform robustly across all background conditions. One potential strategy to achieve this is to train a common model on data drawn from multiple background configurations and subsequently evaluate its performance on

several distinct background conditions, benchmarking it against that of individually optimized expert models.
However due to time constraints, this study tests how well a model trained on one background condition generalizes to a higher background-level condition. Specifically, a model is trained on data from experiment 26, run 1893, with $\langle n \rangle_{\text{extraCDChits}} = 1219$ and then applied to events from experiment 37, run 1893, which has an approximately 60 % higher background, $\langle n \rangle_{\text{extraCDChits}} = 1951$. Table 8.8 summarizes the performance on 50 000 $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ events under these conditions and includes, for comparison, the baseline filter performance.
The GNN trained on exp. 37 achieves the best overall performance. However, the model trained on exp. 26 is statistically almost indistinguishable across all metrics.
In the $\mu^+\mu^-(\gamma)$ channel, the exp. 26 GNN improves on the MVA filter by 1.72 %pt and on the cut-based filter by 4.39 %pt, trailing the exp. 37 GNN by only 0.08 %pt. It recovers about 98.1 %pt of the charge-efficiency of the exp. 37 model, despite being trained on data with much less background.
For $B^0\bar{B}^0$ events, the pattern is similar, but the gains are smaller: the cut-based filter reaches 73.62 %, the MVA 75.73 %, the exp. 37 GNN 76.63 %, and the exp. 26 GNN 76.57 %. Here, the exp. 26 GNN improves on the MVA by 0.84 %pt and on the cut-based filter by 2.95 %pt, trailing the exp. 37 GNN by only 0.06 %pt. Thus, the generalization gap in track fitting efficiency is negligible relative to the improvement over the baselines.
In $\mu^+\mu^-(\gamma)$ events, the MVA fake rate is 1.78 %, while the exp. 37 and 26 GNN models achieve lower values of 1.33 % and 1.13 %. For $B^0\bar{B}^0$ samples, the MVA achieves the lowest fake rate at 1.88 %, with the GNN filters at 2.04 % and 2.00 %, which differ only slightly from each other.
The clone rate remains low for all methods. In $\mu^+\mu^-(\gamma)$ events, all approaches lie in the range 0.03 to 0.05 %, and in $B^0\bar{B}^0$ events they cluster around 0.83 to 0.92 %, with the GNN filters only marginally exceeding the baseline methods. In addition, kinematic resolutions are broadly stable, with the GNN filters matching or improving upon the baselines.

Despite the substantially more challenging background conditions in exp. 37, the model trained on exp. 26 exhibits performance comparable to that of the model trained directly on exp. 37. The track fitting efficiency differs only by 0.08 %pt in $\mu^+\mu^-(\gamma)$ samples and 0.06 %pt in $B^0\bar{B}^0$ events, values that are significantly smaller than the improvement compared to the application of the MVA filter. This behavior indicates that the model predominantly learns a background-invariant representation of the track structure, rather than overfitting to the specific noise characteristics present in the training data. Nevertheless, training on samples that span a broader range of background levels may further increase its generalization capability.

8.4. High-level trigger application

The results of this study suggest that the GNN-based hit filter should, in principle, be compatible with an application in the Belle II HLT as introduced in subsection 3.5.3. The processing time measurements discussed in subsection 8.2.5 show that despite the complex

filtering algorithm, the total tracking time with the GNN filter remains of the same order as for the existing configurations with a relative increase of 10 to 15 % per call compared to the currently used MVA filter. Simultaneously, the track finding benefits significantly from the strong background reduction in CDC hits, particularly under high-background conditions. From a physics perspective, the demonstrated robustness of the GNN filter at high background, including displaced topologies such as K_S^0 and Λ decays, indicates that an HLT deployment would not only help to stabilize tracking performance as luminosity increases, but could also improve the trigger's sensitivity to displaced signatures. This is feasible only under the condition that the available processing time is adequate, a requirement that remains subject to further studies.

8.5. Summary

I have developed and evaluated a GNN-based hit filter for the Belle II CDC track reconstruction, and compared its performance with the existing cut-based and MVA filters under various background conditions and for different classes of physics events. I initially implemented the algorithm in `Python` for design, training, and optimization, and subsequently integrated it into the standard basf2 track reconstruction chain using an `ONNX` representation of the trained model executed via `ONNX Runtime` in C++. Across $\mu^+\mu^-(\gamma)$, $B^0\bar{B}^0$, and dedicated displaced-decay samples, the GNN filter consistently improves both hit-level and track-level performance, with the largest gains observed in the highest-background (exp 0) scenario and in the detector end-cap regions.

From the perspective of computational performance, the corresponding study demonstrates that the GNN filter, even in its current non-optimized C++ implementation, maintains the overall tracking runtime at a level similar to that of the existing configuration, while substantially reducing the time required for track finding due to the strong suppression of background hits. The moderate increase in track fitting time relative to the MVA filter is consistent with a higher track finding efficiency and a higher hit efficiency and the consequently larger number of associated hits per track that must be fitted.

At the hit level, the GNN filter provides a substantial suppression of background hits and a pronounced reduction in the number of extra CDC hits $\langle n \rangle_{\text{extraCDChits}}$, while preserving high signal hit efficiency. This results in significantly cleaner input to the tracking algorithms compared to both the baseline cut-based and MVA approaches. For $\mu^+\mu^-(\gamma)$ events evaluated for the high background scenario (exp. 0), the average number of extra hits $\langle n \rangle_{\text{extraCDChits}}$ is reduced from approximately 2 800 before filtering to approximately 1 700 with the cut-based filter, 600 with the MVA filter, and only 100 with the GNN filter. This corresponds to an improvement of more than an order of magnitude relative to the cut-based filter and more than a factor of four relative to the MVA filter. Concurrently, the per-track hit efficiency increases from about 79 % (cut-based) and 81 % (MVA) to nearly 87 % with the GNN-based filter.

The increased hit filtering directly translates into improved track-level performance. The GNN-based approach achieves a higher charge efficiency in track fitting for all the configurations studied and for both both $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ samples, with a relative gain increasing as the background level increases. Under the most challenging experiment 0 conditions, the efficiency for $B^0\bar{B}^0$ events increases from approximately 62 % for the cut-based filter and 71 % for the MVA filter to 74 % with the GNN filter. For $\mu^+\mu^-(\gamma)$ events, the efficiency improves from about 73 % (cut-based) and 88 % (MVA) to 93 %. At the same time, fake rates remain at a comparable level, and transverse momentum and z_0 resolutions exhibit a modest improvement, attributable to higher hit efficiency.
In particular, for low-p_T tracks and tracks in the end-cap regions, the GNN recovers the efficiency losses from the baseline filters, with the largest gains observed in the backward end-cap. For $\mu^+\mu^-(\gamma)$ events in this region, the efficiency increases from about 21 % (cut-based) and 55 % (MVA) to 75 %, and for $B^0\bar{B}^0$ events from approximately 22 % (cut-based) and 43 % (MVA) to 52 %. These gains correspond to relative improvements of up to 252 % (cut-based) and 36.4 % (MVA) in the $\mu^+\mu^-(\gamma)$ sample and up to 134 % (cut-based) and 18.4 % (MVA) in the $B^0\bar{B}^0$ sample.

9. Online GNN-Based Hit Filtering

The GNN-based hit-filtering algorithm is intended to be integrated as an early preprocessing stage within the CDC L1 trigger chain (see Figure 4.1). In this chapter, I adapt the hit filtering algorithm optimized for the offline tracking application from chapter 8 to the specific constraints of the Belle II L1 trigger system and evaluate the impact on downstream trigger modules.
The hit filtering is planned to be positioned between the extraction of CDC hit information by the FEE boards and the TSF, which compresses the hit information into TSs. Subsequent track reconstruction stages operate exclusively on the TS information and therefore process the cleaned hits identically to the unfiltered default hits.

I compare the track- and trigger-bit-level performance of three configurations:

1. `2DHough`: the current two-dimensional Hough track finder,

2. `3DHough`: the prospective three-dimensional Hough track finder, and

3. `adjusted 3DHough+GNN`: the prospective 3D Hough track finder preceded by GNN-based hit filtering and employing updated TS conditions, as described in section 9.3.

Since the hit-cleanup application is intended for future deployment in the online trigger chain, I evaluate the CDC trigger based on the `3DHough` track finder configuration that is planned for use in the detector in the near future, both with and without the hit-cleanup enabled. The `2DHough` configuration is added for comparison purposes. The analysis is performed within the basf2 framework using the release-10 trigger simulation.

For the evaluation of the proposed GNN-based hit-filtering algorithm, I use two distinct categories of Belle II collision data samples. On the one hand, the track trigger must maintain a high trigger efficiency, and thus a high track finding efficiency, for physics signal events. In particular, low-multiplicity events are of interest, since the primary objective is to identify at least one high-quality track per event rather than to fully reconstruct complex multi-track final states. To quantify the track reconstruction efficiency, I therefore use the low-multiplicity HLT-selected $\mu^+\mu^-$ sample (`mumuskim`) from exp. 35, run 2894 as defined in Table 6.2 (category 15).
On the other hand, the L1 trigger must exhibit a very low fake rate in the presence of background processes, which occur at significantly higher rates than signal events. To quantify the fake rate on background events, I use a background-only sample from delayed Bhabha events (category 16 in Table 6.2) monitoring the TRG rate of the most relevant track-related trigger bit, STT. The total allowed trigger rate is 30 kHz. However,

since multiple trigger bits contribute to the final trigger decision, as described in subsection 4.2.7, the rate associated with any single trigger bit should not exceed $O(8\,\text{kHz})$.
As a proxy for the ground truth, I use offline reconstructed tracks and consider only generated tracks with at least seven hits in the CDC, reconstructed momentum $p > 0.7$ GeV/c, transverse momentum $p_T > 0.2$ GeV/c, and impact parameter $z_0 < 15\,\text{cm}$. These track selection criteria are chosen to match the STT requirements.

9.1. Real-time constraints and design considerations

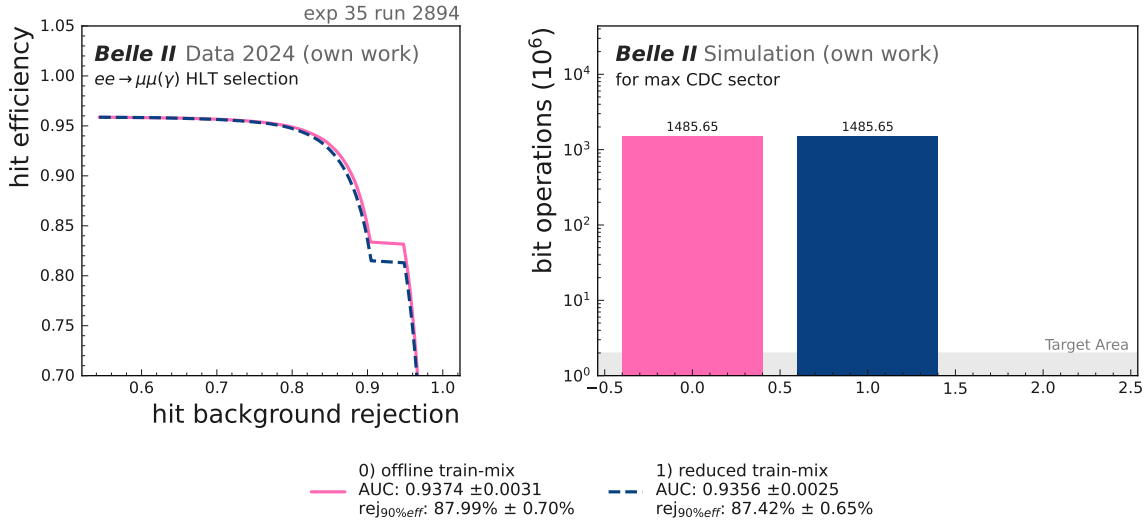
For implementation within the L1 trigger system, the hit filtering algorithm must satisfy several constraints. It must process up to 14 336 sense wires in the CDC at the clock frequency $f_{\text{CDC}} = 31.804\,\text{MHz}$, with an approximate latency budget of $O(500\,\text{ns})$. The targeted UT4 platform imposes strict limits on algorithmic complexity and, in particular, on the number of MAC operations, which directly constrain the maximal size and complexity of the GNN model.
To meet the latency requirement, a precomputed graph definition is planned, assuming that all wires are hit and fully connected via all geometrically allowed edges. During inference on the FPGA, for each event and hit, the graph attributes are updated with hit-specific ADC and ΔTDC information, while all remaining geometric information is predefined. This static graph construction strategy provides lower and deterministic latency compared to dynamic graph building on-the-fly, at the cost of increased FPGA resource usage.
Due to the large number of channels and stringent throughput and latency requirements, the algorithm is planned to be distributed across 20 parallel boards. The corresponding CDC sectors are partitioned by super-layer and azimuthal angle, following [97]: the six innermost super-layers each contain two sectors, and the two outermost super-layers each contain three sectors. The largest of these sectors, in super-layer 6, has 978 sense wires and 4 545 edges including an overlap between neighboring sectors of two wires in the azimuthal direction.
Restricting the sectors to the CDC layers currently used in the L1 trigger processing (*i.e.* five layers per super-layer), the largest sector size is reduced to 800 nodes and 3 505 edges. With the system frequency fixed at $f_{\text{GNN}} = 127.216\,\text{MHz}$ (= four times f_{CDC}) aligned with the L1 trigger clock, this configuration yields a reuse factor of $R = 4$, *i.e.* the processing is executed four times within a single clock cycle of the CDC clock. Under these conditions, up to 204 nodes and 893 edges need to be processed per clock cycle.
The inputs to the hit-filtering algorithm also differ from those in offline tracking. As discussed in subsection 3.5.1, the L1 trigger system receives inputs with reduced resolution compared to the DAQ, so fewer, lower-precision inputs are available. For example, currently, ADC values have only 1 bit resolution and *i.e.* only indicate whether a threshold is exceeded. In the future, and as assumed in this work, the ADC is expected to provide 4 bit resolution. Since the mapping (binning) to 4 bit is not yet fixed and is a free design parameter of this study, I also investigate possible binning schemes for this input quantization.

9.2. Network design and compression for the Level-1 trigger

In this section, the final offline tracking model described in chapter 8 is progressively adapted to the specific operational constraints of the Belle II L1 trigger system. The offline design prioritizes maximization of track fitting performance with a focus on track fitting efficiency under comparatively loose computational constraints. In contrast, the trigger-oriented implementation prioritizes a low track fitting fake rate above track fitting efficiency. It must be performed on FPGA hardware and is therefore subject to stringent limitations on latency, arithmetic resources, and input features as described in section 9.1. To reconcile these conflicting requirements, I develop a compression workflow that starts from the offline reference model and progressively introduces a series of modifications: adjustments to the available hit information and input feature set, an updated graph-building scheme, low-precision quantization and pruning, as well as structural compression of the network architecture.
The impact of each modification step is quantified using the `mumuskim` sample from exp. 35, run 2894 as defined in (Table 6.2 category 15). The principal performance metric throughout this chapter is the hit-level ROC curve and its associated AUC score, evaluated on 1 000 events (corresponding to about $2.4 \cdot 10^6$ individual hits, with multiple hits per wire). For each model configuration, ROC curves are computed as the average over five independent evaluation trials. Within each trial, the best-performing model among three independently trained instances is selected, thereby mitigating the impact of training-induced stochastic variability and instabilities. The definitions of hit efficiency and purity underlying these ROC curves, as well as the BOP metric used as a proxy for the utilization of hardware resources, are provided in chapter 7. Based on preliminary FPGA implementation studies, the projected target size is estimated to be within a range of 1 to 2.5 million BOPs. This large uncertainty interval is specified to account for the inherent uncertainty in the BOP estimation. The reported BOP values correspond to the largest CDC sector, under the assumption of a CDC segmentation into 20 sectors and the additional constraint that only the L1 trigger layers, *i.e.* , the innermost five layers, are taken into account.

9.2.1. Train sample composition

As an initial trigger-specific adaptation, I compare the default training-sample composition from the offline tracking studies subsection 8.1.1 with a modified composition optimized for the L1 trigger use case. The offline training mixture contains particle-gun tracks with a minimum transverse momentum of 50 MeV, additional low-p_T enrichment for particle gun samples, and an average track multiplicity of six tracks per event. For the trigger application, where the primary objective is the efficient identification of at least one well-reconstructed track, I construct a reduced training sample in which the particle-gun phase space is constrained to $p_T > 250$ MeV, corresponding to the current acceptance of the Belle II trigger. In addition, the low-p_T enrichment is removed, and the mean number of signal tracks per event is reduced to about four.
As illustrated in Figure 9.1, both mixtures yield comparable performance, with AUC values of 0.9374 for the offline training mixture and 0.9356 for the reduced training mixture. The background rejection at the operating point corresponding to a hit efficiency



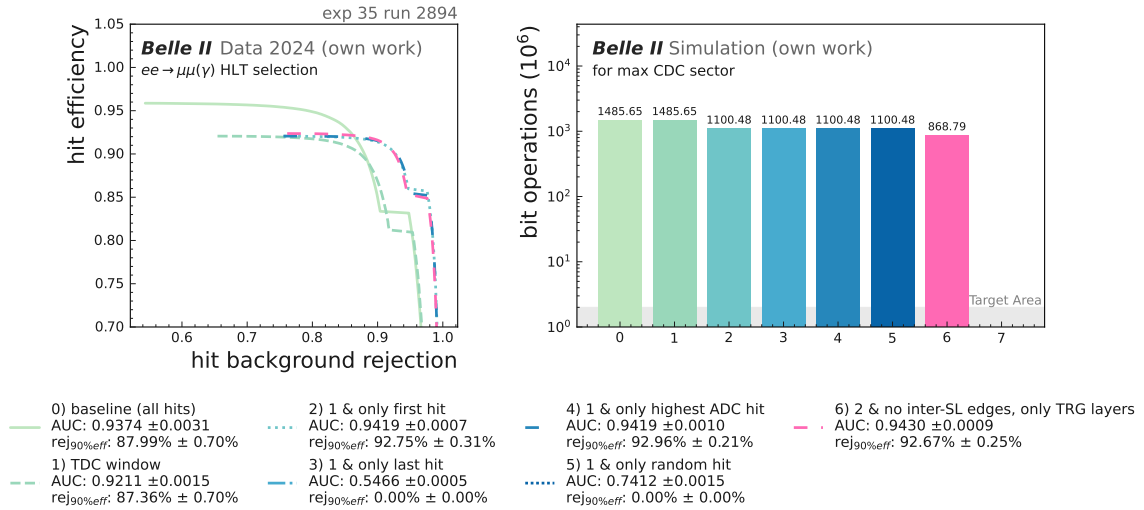
(a) ROC curve

(b) number of bit operations

Figure 9.1.: Evaluation of hit-classification performance and model complexity for the "offline train-mix" and "reduced train-mix". Hit-level performance is measured via the ROC curve and its derived quantities AUC and $\text{rej}_{90\%\text{eff}}$ (a) computed on the HLT-selected `mumuskim` sample (experiment 35, run 2894). Model complexity is measured via BOPs for the largest CDC sector (978 nodes, 4 545 edges) (b). The "offline train-mix", containing higher track multiplicities and lower-p_T tracks, yields a slightly higher AUC= 0.9374 ± 0.0031 and $\text{rej}_{90\%\text{eff}} = (87.99 \pm 0.70)$ % than the "reduced train-mix" with AUC= 0.9356 ± 0.002 and $\text{rej}_{90\%\text{eff}} = (87.42 \pm 0.65)$ %. The BOPs remain identical, as expected, but exceed the target size by orders of magnitude.

of 90 % attains values of $\text{rej}_{90\%\text{eff}} = 87.99$ % and 87.42 %, respectively. In this and all subsequent configuration evaluations, the AUC and the $\text{rej}_{90\%\text{eff}}$ metrics exhibit a strong correlation. Because the AUC is less sensitive to the sharp edges in the ROC curve, the following discussion will primarily focus on the AUC metric rather than on $\text{rej}_{90\%\text{eff}}$. Within uncertainties, no statistically significant improvement is observed when employing the trigger-specific reduction. The reduced mixture exhibits marginally inferior performance, suggesting that the broader coverage in p_T and track multiplicity in the offline training mixture remain beneficial for generalization to $\mu^+\mu^-(\gamma)$ data. Therefore, I retain the full offline train sample composition, indicated in pink in the plot. Throughout this section, all selected configurations will be highlighted in pink. The sharp edge at a hit efficiency of 80 % originates from the discrete character of both the ROC-curve estimate and the classifier output scores, in conjunction with the finite size of the test sample.

The BOPs per CDC sector are identical between the two configurations, since the training mixture affects only the classification performance while leaving the hardware complexity of the model unchanged. The BOPs exceed the target threshold by several orders of magnitude. Consequently, substantial model compression is required to meet the hardware constraints.



(a) ROC curve

(b) number of bit operations

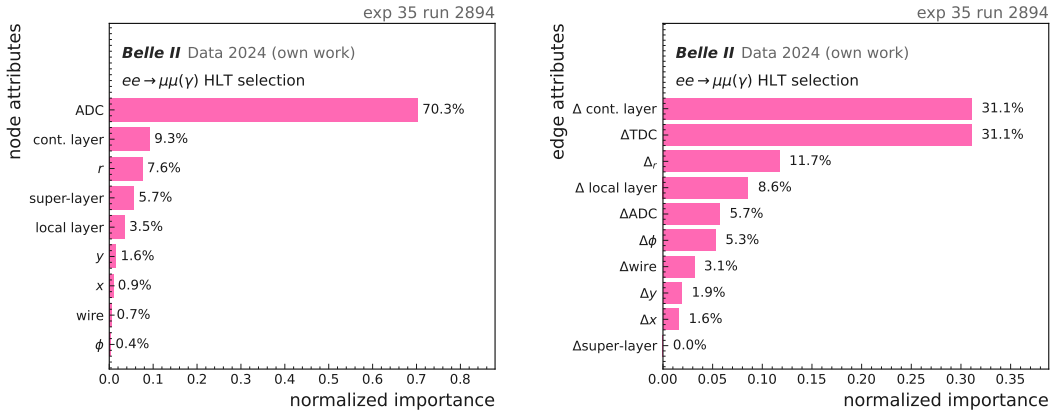
Figure 9.2.: Impact of different hit-selection modes: The baseline configuration (0) uses all hits in an event. Restricting the TDC range to 500 ns (1) reduces the AUC from 0.9374 to 0.9211 while leaving the BOPs unchanged, as expected. The single-hit configurations (2-5) build upon the reduced TDC window and lower the model complexity from 1 486 to 1 100 MBOPs. Among these, keeping only the first hit (2) and only the hit with the highest ADC (4) obtain the best performance, with AUC=0.9419. Finally, starting from the "only first hit" setup, removing inter-layer edges across super-layer boundaries and keeping only hits on TRG layers (6) further reduces the size to 869 MBOPs.

9.2.2. Trigger-compatible timing window and hit selection

In the next step, I adapt the hit selection to the hits available at the L1 trigger level. Based on the basf2 simulation of the TSF described in subsection 4.2.1, all hits are constrained to a common TDC window between 4 450 and 4 950, instead of the range applied in offline tracking from 0 to 4 980 discussed in subsection 8.1.3. In the real L1 trigger system, this time window is individually configured for each CDC superlayer. However, for the present study, a single global window is applied, consistent with the TSF configuration used in the trigger simulation.

In addition to this timing requirement, at most one hit per wire is allowed to serve as a node in the graph. This choice is motivated by the requirement of deterministic behavior in the FPGA implementation with a fixed, pre-calculated graph definition. On average, approximately 25 % of the hit wires register two hits within the same event. Selecting a single representative hit per wire therefore yields a substantial reduction in the overall graph size.

I evaluate several strategies for selecting a single hit per wire and event: retaining all hits as a reference configuration, restricting to only the first or only the last hit within the TDC window, selecting the hit with the maximum ADC value, and, for comparison, choosing a random hit. The corresponding hit-level ROC curves and AUC values obtained for the



(a) node-feature importance

(b) edge-feature importance

Figure 9.3.: Normalized node (a) and edge (b) feature importance obtained from a model trained with the full input feature set. During inference the specific features are masked to determine their individual importance. The displayed feature importance is evaluated on the `mumuskim` benchmark sample for 1 000 events with a background condition corresponding to exp. 35, run 2894.

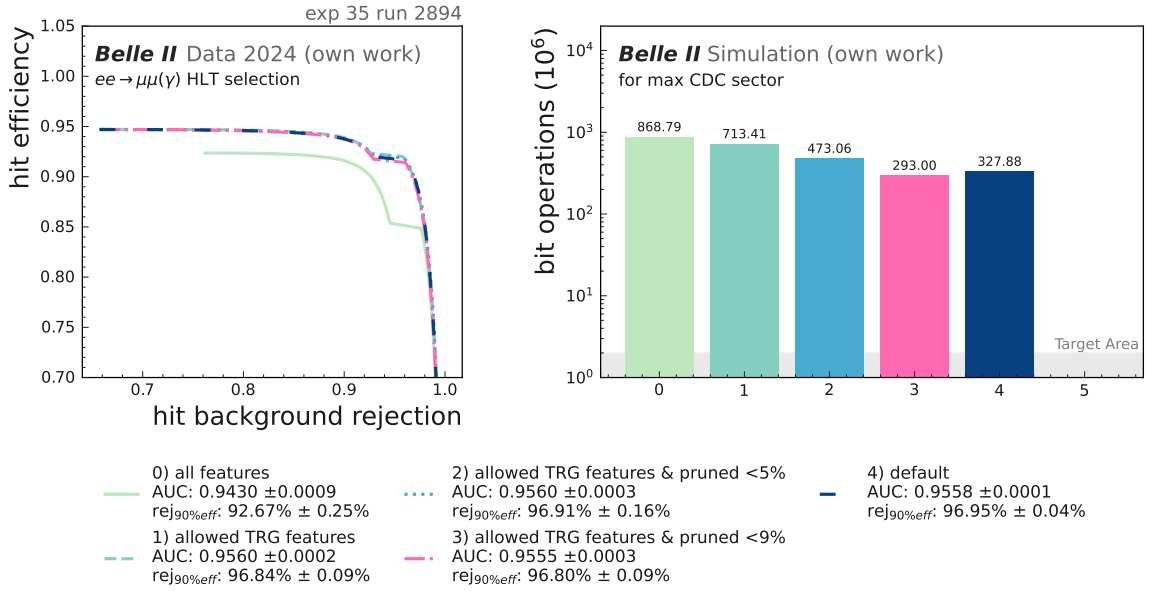
`mumuskim` sample are shown in Figure 9.2. The application of the TDC window alone reduces the AUC from 0.9374 to 0.9211. Among single-hit strategies, the “last-hit” selection exhibits markedly inferior performance, with an AUC of 0.5466, *i.e.* significantly worse than even the random-selection baseline. In contrast, choosing either the first hit (AUC = 0.9419) or the highest-ADC hit (0.9419) yields a small improvement relative to the "all-hits" reference. Since the first hit mode is straightforward to implement deterministically in hardware, and its performance is effectively indistinguishable from that of the highest-ADC mode, I adopt “first hit in the trigger TDC window” as the default hit-selection strategy for the design. The corresponding decrease in model size amounts to a reduction from 1 486 to 1 100 MBOPs.

Finally, I restrict the input to hits originating from the trigger layers defined in section 4.2 and disable inter-super-layer edges. The latter configuration reflects the intended parallelization strategy of the algorithm across independent CDC sectors, in which no connections are foreseen across the superlayer boundaries. In contrast, constraining the input to trigger layers is not a fundamental limitation of the planned hardware system, where all layers are expected to transmit information to the trigger. However, restricting the layers to trigger layers is required to ensure consistency with the current trigger simulation that assumes TSs that contain only hits from those layers. In addition, this decision further reduces the size of the graphs and therefore BOPs.

9.2.2.1. Trigger-available hit features and feature pruning

The offline model uses several hit-level features that are not available at the TRG-level, in particular the TOT value, the ratio ADC/TOT, and the calibrated hit TDC value. The TOT blue is not transmitted to the L1 trigger and thus the derived ADC/TOT quantity is not available as well. Moreover, the absolute TDC value per hit is not meaningful at this early stage of the pipeline, since there is no event time t_0 calculated yet and only a relative timing between hits is available. For this reason, only the TDC differences between neighboring hits, encoded as edge features ΔTDC, carry useful information for the trigger application, while the absolute TDC information is discarded.
To determine a minimal feature set that remains compatible with trigger constraints, I begin with the complete set of trigger-available hit features and assess their relevance using a masking-based explanation approach inspired by the `Pytorch Geometric` [98] explainer method. In this method, each node or edge feature is individually masked at inference time, and the resulting variation in the model output is used as a proxy for its importance. The corresponding node and edge-level importance scores that guide the iterative feature-pruning steps are shown in Figure 9.3.
In an initial feature-pruning step, I remove all features with an estimated importance below 5 %. This affects the node features x, y, ϕ, and the wire index, as well as the edge features Δwire, Δx, Δy, and Δsuper-layer.
In a subsequent step, I recompute the feature importance on this reduced feature set, and all remaining features with an importance below 9 % are eliminated. Concretely, this affects the node features *layer*, *super-layer*, and radius r, as well as the edge features Δlayer and Δr. The decision to discard r instead of the *continuous layer* index is motivated by the assumption that both variables encode closely related geometric information, while r is a floating-point quantity, the *cont. layer* is already represented as a discrete integer in the range 1-56, which is better for hardware implementation.
The final pruned configuration retains only two node features, ADC and *cont. layer*, and three edge features, ΔTDC, Δcont.layer, and Δϕ. The corresponding feature importance scores after application of the first feature-pruning step are provided in the appendic (A.8).

The resulting performance associated with each of the feature sets discussed is presented in Figure 9.2.
Interestingly, the trigger-compatible and pruned feature sets achieve marginally higher AUC scores on the `mumuskim` sample than the original offline feature configuration. A plausible explanation is that, in this scenario, the removal of less robust or poorly modeled quantities enhances generalization across different running conditions (training on MC, exp. 26, evaluation on data, exp. 35). This, in turn, mitigates the model's sensitivity to subtle mismatches between simulation and data present in those additional features.
The first feature-pruning step does not induce a reduction in the hit-level ROC performance: the AUC score remains at 0.9560, while the nominal BOP per maximum CDC sector decreases from approximately 713 to 473 MBOPs.
Despite the substantial reduction in the input dimensionality due to the second feature-pruning step, the AUC remains at 0.9555, while the nominal BOP decreases further to approximately 293 MBOPs. For comparison, the "default" feature set employed in previ-



(a) ROC curve

(b) number of bit operations

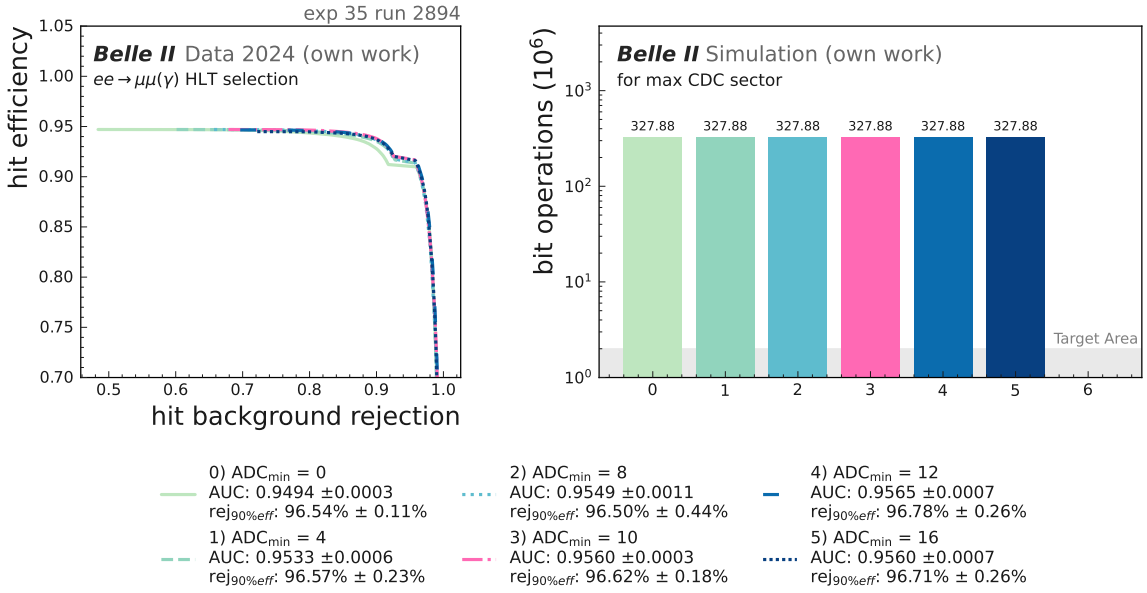
Figure 9.4.: Starting from all offline-available hit features, restricting to trigger-available features and then pruning those with importance below 5 % and 9 % preserves the AUC at about 0.9560 while reducing the BOPs from 869 to 293 MBOPs. The "default" feature set used in prior studies [4, 5] achieves the same AUC but at a higher complexity of about 327 MBOPs.

ous studies [4, 5], comprising x, y and ADC at the node-level, as well as Δr, $\Delta\phi$ and ΔTDC at edges, achieves the same AUC but requires approximately 328 MBOPs. Consequently, I adopt the final pruned feature set, as it maintains the hit-level classification performance while substantially reducing the computational complexity relative to both the full offline feature set and the previously used default configuration.

9.2.3. Pre-selection on ADC

Analogously to the offline tracking studies, I investigate whether pre-selections on the hit ADC values can improve the overall performance. These selection cuts are applied before training and inference and remove hits with very small or excessively large ADC values before the graph is constructed. Since the BOP metric, by design, assumes a fixed graph size, the nominal BOP per CDC sector remains unchanged across all ADC-cut configurations considered in this study.

In a first step, I scan the lower ADC threshold from $ADC_{\text{min}} = 0$ to $ADC_{\text{min}} = 16$, as shown in Figure 9.5. The hit-level ROC curves exhibit a small dependence on this threshold: the AUC increases slightly from 0.949 without an ADC cut to 0.956 around $ADC_{\text{min}} = 10$, and remains stable for thresholds up to $ADC_{\text{min}} = 16$. Based on this scan, I select a working point of $ADC_{\text{min}} = 10$.



(a) ROC curve

(b) number of bit operations

Figure 9.5.: Increasing the lower ADC threshold, the AUC improves from 0.9494 at $\mathrm{ADC}_{\mathrm{min}} = 0$ to 0.9560 around $\mathrm{ADC}_{\mathrm{min}} = 10$, while the nominal BOPs remain unchanged, as expected.

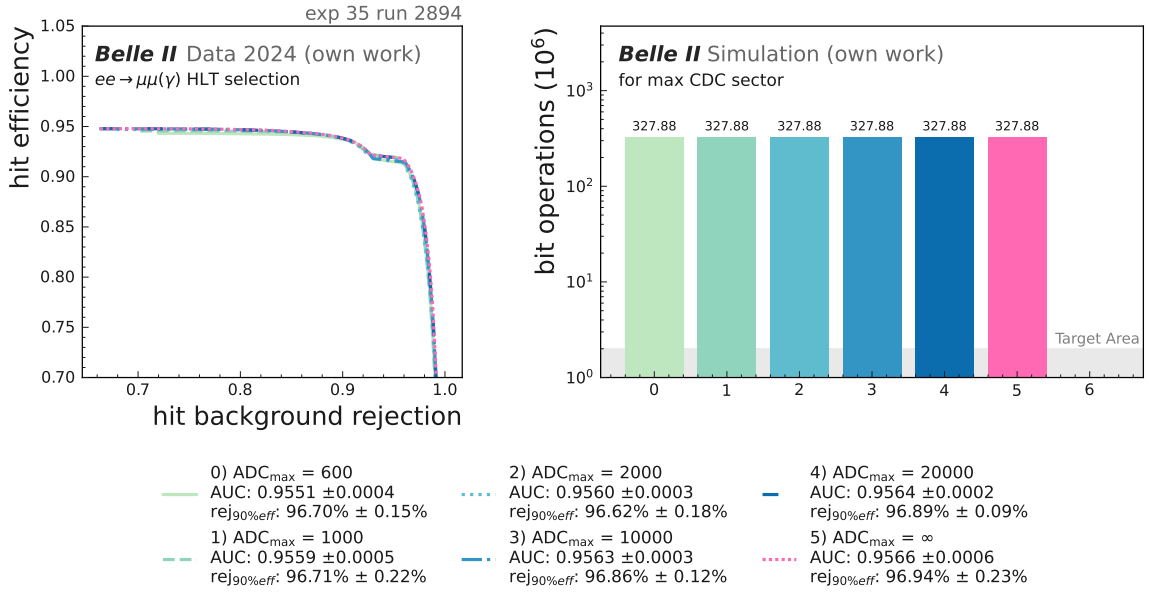
In a second step, I vary the upper ADC cut from $ADC_{\mathrm{max}} = 600$ to $ADC_{\mathrm{max}} = 20\,000$ and also consider the configuration without any upper cut, see Figure 9.6. The AUC exhibits a monotonic increase as ADC_{max} increases, attaining values in the range from 0.9551 to 0.966. Therefore, I do not impose an explicit upper ADC threshold in the final configuration.

9.2.4. Network size reduction

To further reduce the computational complexity of the GNN, I perform a scan of the hidden size (number of neurons per layer) and hidden depth (number of layers per MLP) of the network. Starting from the baseline configuration with hidden size $h = 8$ and hidden depth $d = 2$, I progressively decrease the hidden size down to $h = 3$ and the depth to $d = 1$. The corresponding hit-level ROC curves and BOP estimates for the combinations of hidden size and depth (h, d) are presented in Figure 9.7.

In this scan, the AUC exhibits only gradual degradation as the network capacity is reduced. The baseline model with $(h, d) = (8, 2)$ achieves an AUC of 0.9566, while configurations with reduced size down to $(5, 1)$ still achieve AUC values of 0.9551. Even for the most compact architectures with $h = 3$, the AUC remains above 0.9483.

In contrast, the BOP cost decreases substantially with reduced hidden size and depth, since fewer neurons and layers directly translate into fewer multiply-accumulate operations. The baseline $(h, d) = (8, 2)$ model requires approximately 328 MBOPs, while the configuration with $(h, d) = (5, 1)$ achieves a comparable AUC of 0.9551 at only about



(a) ROC curve (b) number of bit operations

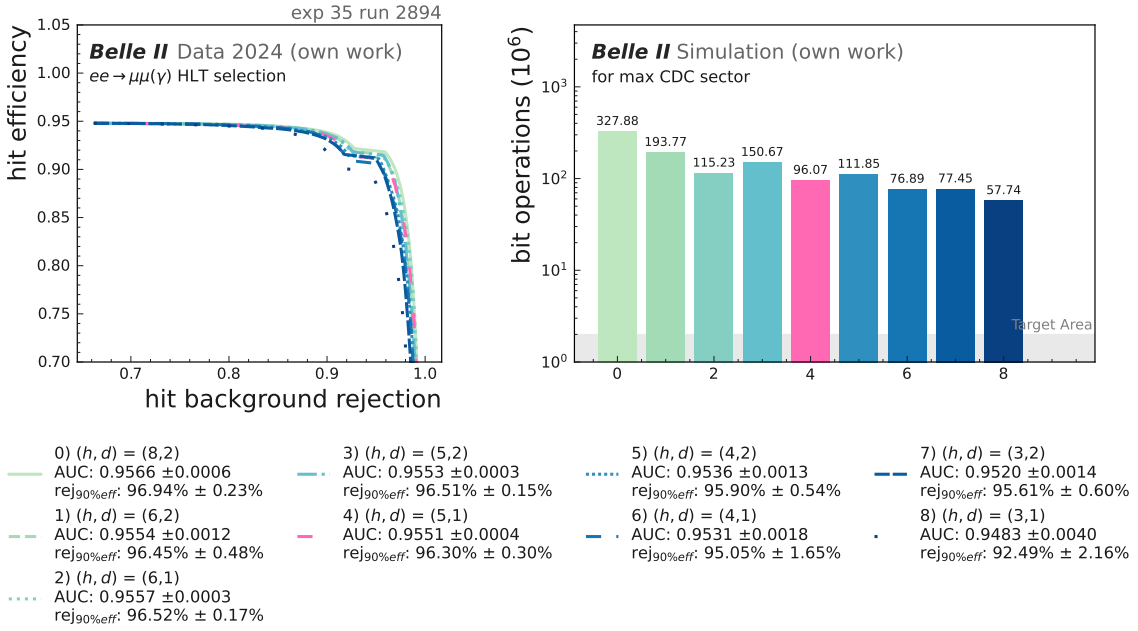
Figure 9.6.: Within uncertainties, the AUC increases monotonically between 0.9551 and 0.9566 over the full range from $\mathrm{ADC}_{\max} = 600$ to $\mathrm{ADC}_{\max} = \infty$, while the nominal BOPs are unaffected, motivating the choice to omit an upper ADC cut in the final configuration.

96 MBOPs, corresponding to a reduction in computational cost by more than a factor of three. Therefore, I adopt $(h, d) = (5, 1)$ as the default configuration, as it provides a good compromise between hit-classification performance and model size.

9.2.5. Network quantization

For further network compression, I employ quantization-aware training (QAT) using `Brevitas` [99], a `PyTorch`-based library for neural-network quantization that supports both post-training quantization and QAT. Since `Brevitas` is built on `PyTorch` [92], it integrates natively into the existing `PyTorch`-based workflow and eliminates the need to maintain a parallel TensorFlow/Keras tool-chain, as would be necessary for alternatives such as `qKeras` [100]. Although `qKeras` remains publicly available, its most recent tagged release dates to 2021, and it appears to only be minimally maintained, with several issues related to recent `TensorFlow` versions still unresolved. By contrast, `Brevitas` is under active development, with releases and documentation updates continuing through mid-2025.

`Brevitas` supplies quantized variants of commonly used neural-network layers that can serve as drop-in replacements for standard `PyTorch` layers. Conceptually, `Brevitas` emulates reduced numerical precision during training by inserting quantization operators into the forward pass, thereby approximating fixed-point arithmetic while retaining compatibility with `PyTorch`'s differentiation. Weights, activations, and biases are represented as low-precision fixed-point integers with configurable bit width, scaling factor, and



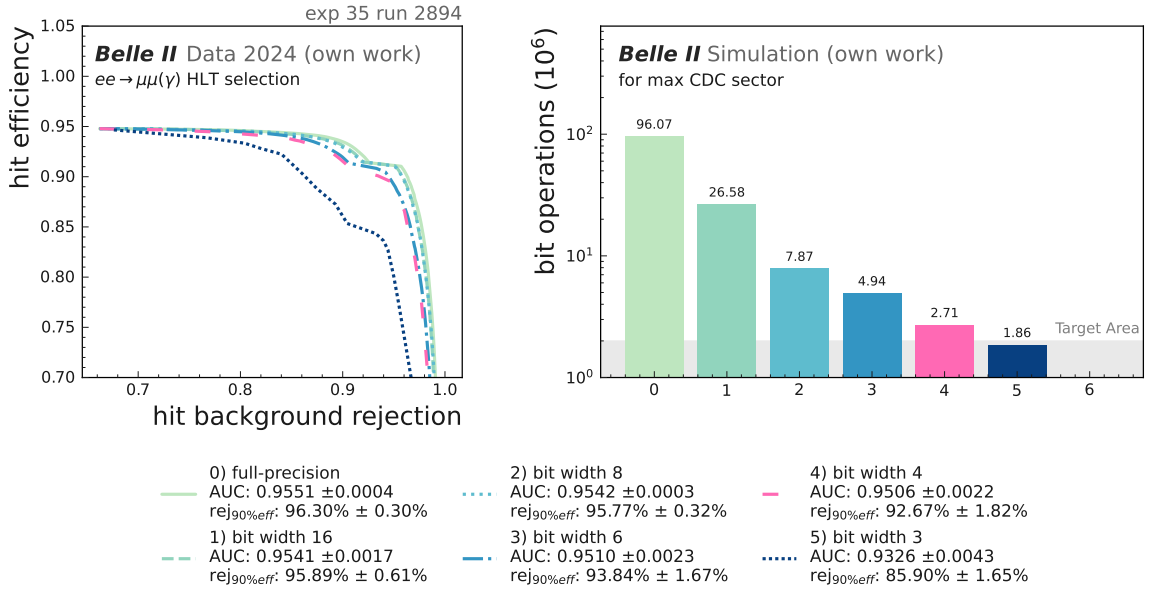
(a) ROC curve

(b) number of bit operations

Figure 9.7.: The AUC decreases from 0.9566 to 0.9483 as the network size is reduced using different combinations of hidden size h and hidden depth d. Concurrently, the total BOP cost decreases by more than a factor of five, and specifically by more than a factor of three when comparing the baseline configuration $(h, d) = (8, 2)$ to the selected configuration $(h, d) = (5, 1)$.

zero-point The associated scale parameters are represented as differentiable quantities that are optimized during training. For activations, `Brevitas` applies an initial calibration phase in which a mean-squared error loss between full-precision and quantized activations is used during the first 300 training steps (*i.e.* the first 300 events in the first batch). Bias quantization is derived from the product of the input and weight scales, ensuring that the bias term shares the same effective dynamic range as the accumulated products. The output of quantized layers is returned in the de-quantized form as floating-point tensors constrained to a finite set of discrete values determined by the chosen bit width, which enables subsequent components of the `PyTorch` computation graph to operate on standard tensor types while still being subject to quantization effects. During training, quantizers are active in the forward pass with differentiable scaling parameters, whereas the backward pass employs gradient estimators for the quantization steps and propagates gradients through the de-quantized representations. For integration in `C++` as well as export format for hardware implementation, the final network architecture is exported to `onnx` format. `Brevitas` provides custom `onnx` export methods for quantized layers, including sub-8 bit quantization by scaling to 8 bit and clipping the integer ranges to lower precisions.

To integrate `Brevitas` into the definition of the interaction network, the linear and activation layers within the MLPs employed for edge and node updates are replaced by their



(a) ROC curve (b) number of bit operations

Figure 9.8.: The AUC score decreases moderately from 0.9551 (full-precision) to 0.9542 for a bit width of $b_w = 8$, and to 0.9506 for $b_w = 4$, while the BOP cost drops by more than an order of magnitude from full precision to 4 bit, motivating $b_w = 4$ as the default configuration.

quantized counterparts. At the boundaries of each MLP, quantized identity operations are inserted to enforce quantized inputs and outputs. These boundary quantizers not only guaranty that all internal computations operate on a fixed-point representation, but also define the input scaling factors that in turn determine the bias scaling in the first linear layers of each MLP. The general message-passing scheme remains unchanged.

An additional design decision concerns the rounding mode used for the floating-point to fixed-point conversion. Although the default rounding mode provided by `Brevitas` is *round-to-nearest*, this work adopts a deterministic *floor-rounding* strategy, applied to weights, activations and biases, in order to facilitate hardware implementation. For export to `onnx` [101], I implemented a custom quantization operator, as floor rounding is not natively supported.

I investigate the impact of network quantization by performing a scan over the weight bit width in the range 16 to 4 bit. All other numerical precisions are defined as functions of the weight bit width b_w as follows:

$$b_{in} = b_w + 1 \,, \tag{9.1}$$

$$b_{act} = b_w \,, \tag{9.2}$$

$$b_{acc} = b_w + b_{act} + \lceil \log_2 D_{in} \rceil + 1 \,, \tag{9.3}$$

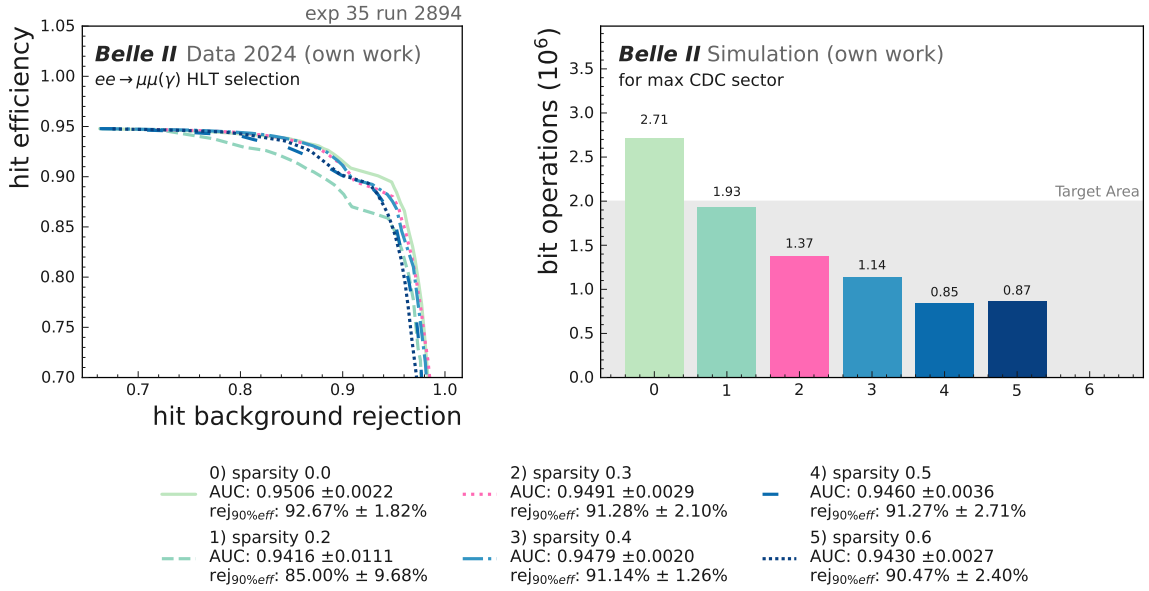
$$b_{bias} = b_{acc} \,, \tag{9.4}$$

$$b_{out} = b_w + 2 \,. \tag{9.5}$$

Here, b_{in}, b_{act}, b_{acc}, b_{bias}, and b_{out} denote the bit widths for the inputs, activations, accumulators, biases, and outputs, respectively, and D_{in} denotes the input fan-in dimensionality, defined as twice the number of node features plus the number of edge features. This configuration ensures that input activations are quantized with a slightly higher precision than weights, while each ReLU activation function progressively reduces the precision to $b_{act} = b_w$ for each subsequent layer. The bias bit width is chosen to be large enough to represent the accumulation sum over all products with an additional safety margin. In the present work, all network parameters are uniformly quantized using a single global bit width. In principle, instead, individual bit widths could be assigned to each model parameter and treat the bit width itself as a differentiable variable during training, thereby enabling the model to learn the required precision for each parameter. A systematic investigation of this approach is left for future work.
In practice, occasional overflows are observed in specific weight and activation representations, indicating that the allocated bit widths are insufficient to accommodate all numerical values encountered in practice. However, these occasional overflows do not appear to affect overall network performance, nor do they induce instabilities during training. The application of pruning, as elaborated in subsection 9.2.6, effectively mitigates such overflow phenomena.
Figure 9.8 shows that the AUC remains basically unchanged when precision is reduced from 16 bit to 8 bit and exhibits only a minor decline when precision is further reduced to 4 bit. A more substantial degradation is observed only at 3 bit, where the AUC decreases to approximately 0.933, indicating that this degree of quantization is too aggressive for the considered architecture. Concurrently, the estimated BOPs decrease significantly as the bit width is reduced: lowering the precision to 8 bit already reduces the arithmetic cost by more than an order of magnitude, and an additional reduction to 4 bit further reduces the cost to 2.71 MBOPs. Consequently, the 4 bit configuration offers a substantial reduction in computational complexity while incurring only a relative loss in classification performance of 0.5 % compared to the full-precision model.

9.2.6. Unstructured weight pruning

To further reduce the computational cost of the quantized IN, I apply global unstructured weight pruning [102] to fully connected layers of the network. During training of the randomly initialized dense network, parameter learning and pruning are performed jointly by simultaneously updating both the weights and biases, as well as associated weight masks. The binary weight masks are implemented using the `PyTorch` pruning utilities, such that the logical architecture of the network remains unchanged, while the number of effective MACs is reduced. The masks enforce sparsity by setting individual weights to zero in accordance with a prescribed global sparsity level, while imposing no structural constraints on the spatial arrangement or positions of the remaining non-zero parameters. Pruning is performed progressively during training until a specified target sparsity is reached.
The impact of network pruning for different levels of target sparsity is illustrated in Figure 9.9. For each target sparsity, an individual model is trained. Both the AUC and the number of BOPs decrease monotonically with increasing sparsity except for a sparsity



(a) ROC curve

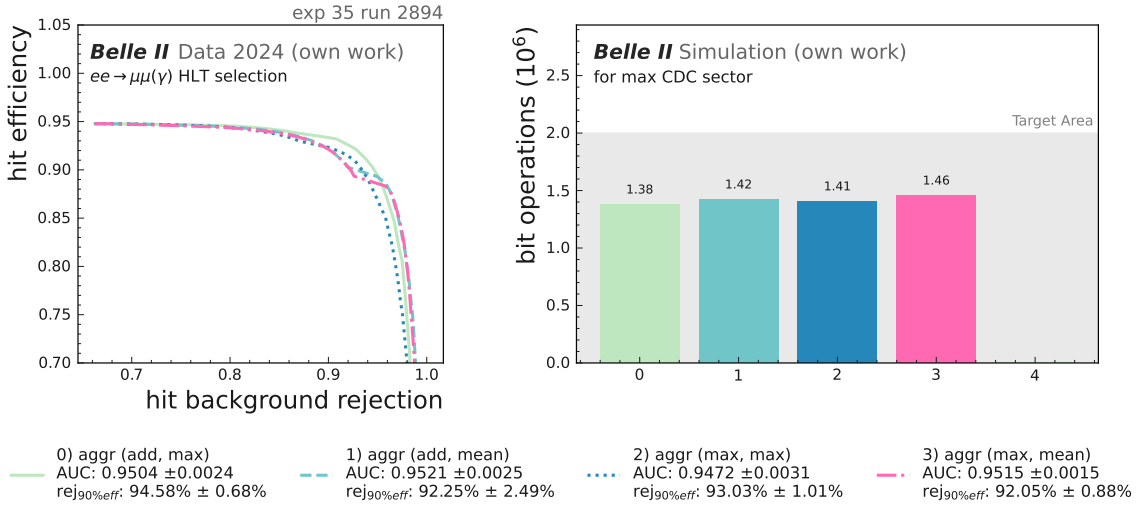
(b) number of bit operations

Figure 9.9.: The AUC score remains close to the un-pruned reference (sparsity 0.0) for pruned models with weight sparsities of order 0.3-0.4, while the BOP decreases approximately in proportion to the sparsity, motivating the choice of a final sparsity of 0.3 as a compromise between performance and computational cost.

level of 0.2 where the AUC is observed to be lower than for more strongly pruned configurations. This is likely to be attributable to statistical fluctuations and convergence to a suboptimal local minimum during training, rather than to a inherent degradation specific to this sparsity configuration.
Based on this study, I select the final global sparsity of 0.3 as the default operating point. At this sparsity level, the performance of hit discrimination at 0.9491 is comparable to that of the un-pruned network with 0.9506, while the number of BOPs is reduced from 2.71 to 1.33 MBOPs. Consequently, the model satisfies the target size constraints.
As an additional beneficial effect, the bit overflows that occurred in the un-pruned configuration, as discussed in subsection 9.2.5, are mostly eliminated by the pruning process. The reason for this is that pruning reduces the magnitude of partial sums in the linear network layers and, therefore, avoids accumulation overflows in computing dot products as discussed in [103].

9.2.7. Aggregation

The IN architecture employed in this work comprises two distinct aggregation stages: an intermediate aggregation of edge-level messages to node representations between the R_1 and O network blocks, and a final aggregation of edge-level scores into a node-level output score. From a hardware implementation standpoint, the intermediate aggregation is subject to stringent resource constraints, as summing a large number of fixed-point



(a) ROC curve

(b) number of bit operations

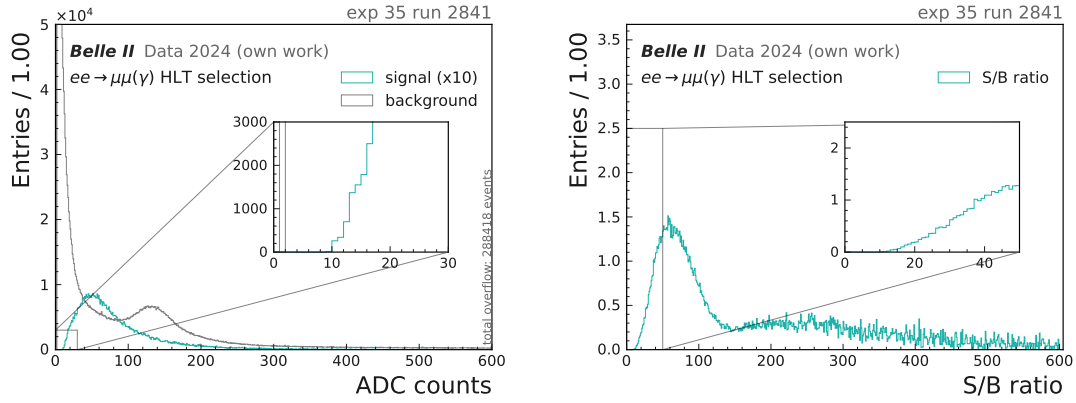
Figure 9.10.: Evaluation of the impact of different combinations of intermediate and output aggregation (`add,max`), (`add,mean`), (`max,max`), (`max,mean`). All combinations achieve similar AUC, with the mean-based output aggregation performing best in general, while the BOP cost is unaffected by the aggregation choice besides statistical fluctuations due to the unstructured pruning.

messages would necessitate wide accumulators, thereby complicating the design and increasing resource utilization. To mitigate this, the intermediate aggregation is replaced by a *max* reduction, which selects the largest incoming message per node without any accumulation, and thus eliminates the risk of numerical blow-up. In the case where a node is not incident to any edge, its accumulation value is defined as 0 to ensure deterministic behavior.

For the final aggregation, the situation differs. At this stage, the number of incident edges per node has already been reduced by preceding network components, and the contribution of this aggregation to the overall BOPs is subdominant relative to that of the dense linear layers. Consequently, more complex aggregation schemes, such as *mean*, could be implemented on the hardware.

To quantify the impact of these design choices, all combinations of aggregation strategies are compared in Figure 9.10. Employing *mean* aggregation for the final output consistently yields improved performance compared to *max*, and, when combined with *max* aggregation at the intermediate stage, it slightly outperforms the original configuration based on *add* and *max*.

The selection of the aggregation method does not affect the resulting BOPs, except for statistical fluctuations induced by weight pruning. Consequently, the choice between the available aggregation schemes is primarily determined by considerations of AUC score performance and simplicity of hardware implementation. In the following, I adopt the *max* configuration for intermediate aggregation and the *mean* configuration for output aggregation.



(a) ADC count distributions

(b) ADC signal to background ratio

Figure 9.11.: Distribution of signal and background hit ADC values (a) for CDC hits in the HLT-selected `mumuskim` data sample (exp. 35, run 2 841). Signal hits are scaled by a factor 10 for better visibility. The corresponding signal-to-background ratio, S/B (b), exhibits a maximum around a value of approximately 50 and displays a long tail to higher values in its distribution.

9.2.8. Pre-quantization of inputs

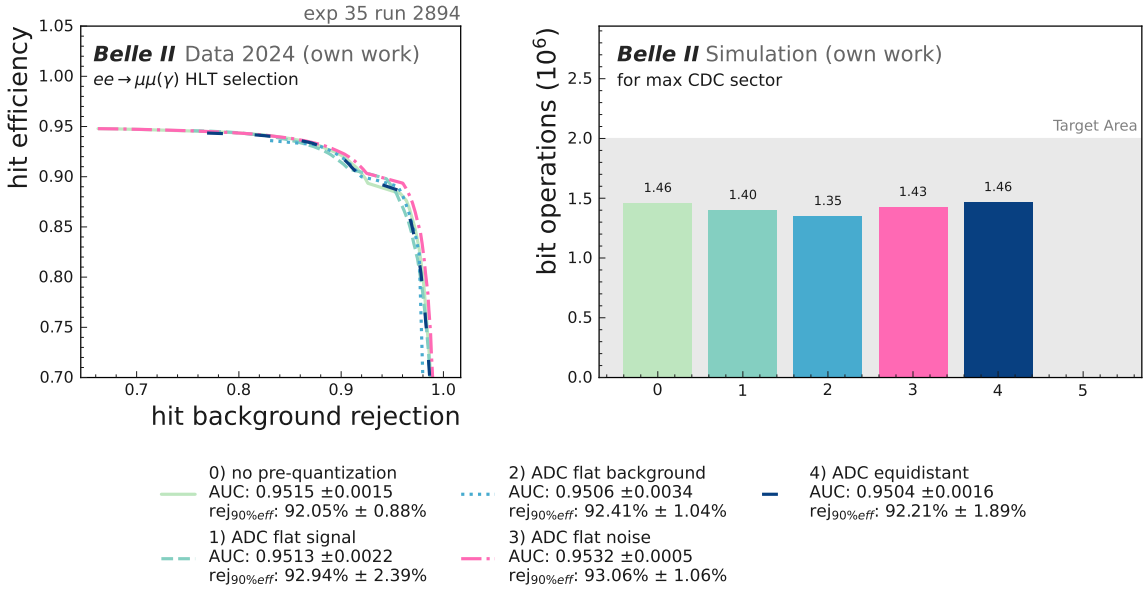
For the TRG application, only a reduced precision of the ADC inputs is available. Currently, a precision of 1 bit is available, and future settings will allow for the 4 bit precision of the ADC input with freely configurable binning. In Figure 9.11 it can be seen that the ADC counts span a wide range of values with markedly different distributions for signal and background hits. Therefore, I investigate different binning schemes for the 4 bit precision that explicitly exploit the observed shapes of the signal and background spectra: *flat signal*, *flat background*, *flat noise* (defined as the square root of the background distribution) and an *equidistant* binning between the 0 and 600 ADC counts. Applying pre-selection cuts on ADC at a value of 10, the bin limits according to 4 bit are

$$\begin{aligned}
\text{flat signal:}\quad & [10, 26, 34, 41, 47, 52, 58, 65, 71, 80, 90, 103, 121, 145, 194, \infty) \\
\text{flat background:}\quad & [10, 12, 15, 20, 27, 37, 53, 71, 94, 114, 131, 146, 168, 206, 312, \infty) \\
\text{flat noise:}\quad & [10, 19, 32, 51, 73, 96, 118, 138, 159, 186, 222, 272, 338, 417, 505, \infty) \\
\text{equidistant:}\quad & [10, 49, 88, 127, 167, 206, 245, 285, 324, 363, 403, 442, 481, 521, 560, \infty)
\end{aligned}$$

Based on these bins, each original value x is first mapped to an integer representation by assigning each value a discrete bin index k

$$k(x) = \min\left\{j \mid x < b_j\right\} \quad \text{(unsigned)} \tag{9.6}$$

$$k(x) = \min\left\{j \mid x \leq b_j\right\} \quad \text{(signed)} \tag{9.7}$$



(a) ROC curve

(b) number of bit operations

Figure 9.12.: Evaluation of different ADC binning schemes applying 4-bit precision. The tested configurations comprise no "pre-quantization", “flat signal”, “flat background”, “flat noise”, and "equidistant" binning, all of which achieve similar AUC scores. The highest AUC score is obtained with the "flat noise" binning scheme at 0.9532 exceeding the case without pre-quantization of the ADC values at 0.9515.

where an unsigned representation is used for the ADC value. In the next step, the raw bin index is shifted by a constant offset in the signed case

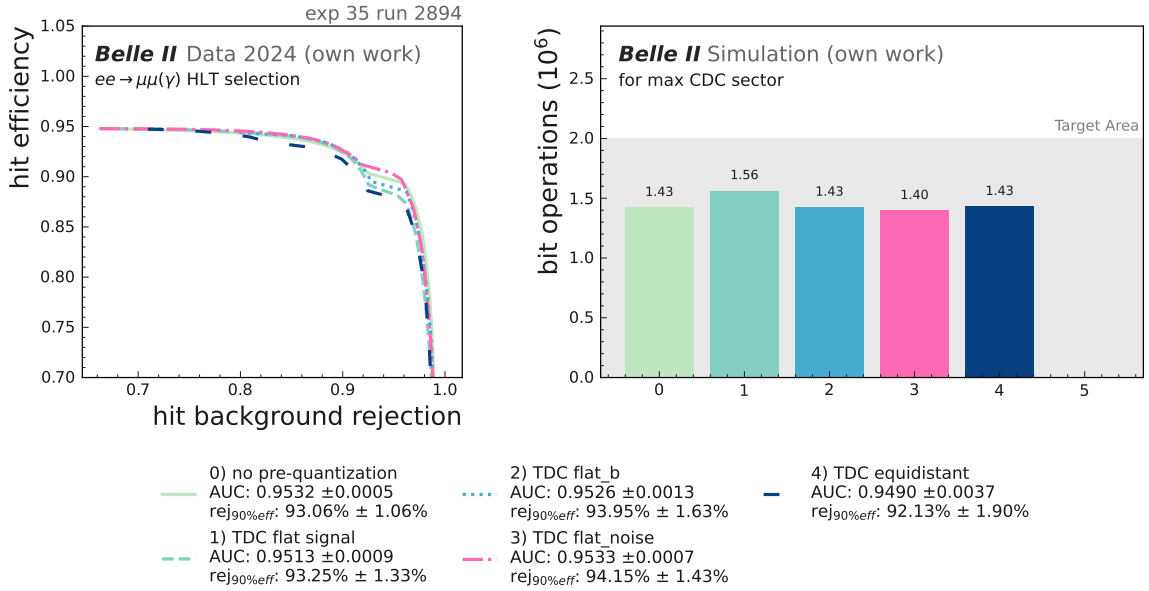
$$k \longrightarrow k' = k(x) \qquad \text{(unsigned)} \tag{9.8}$$

$$k \longrightarrow k' = k(x) - 2^{\text{bit width}-1} \qquad \text{(signed)} \tag{9.9}$$

Afterward, the values are clipped to the maximal available interval. Finally, the integer representation is mapped back to a normalized floating-point value in $[-1, 1]$ for signed and $[0, 1]$ in the unsigned case by dividing the maximum absolute integer magnitude

$$\tilde{x} = \frac{k'(x)}{\max(|k'_{min}|, |k'_{max}|)}. \tag{9.10}$$

The effect of the different ADC binning schemes is shown in Figure 9.12. The different modes exhibit nearly identical hit-level classification performance, with AUC values in the range 0.9504 to 0.9532. This indicates that restricting the ADC resolution to 4 bits induces only a small degradation in classification quality and that the precise choice of the binning scheme constitutes a higher-order effect within the statistical precision of the present study.
Among the configurations investigated, the *flat noise* binning, derived from the square root of the background distribution, yields the highest nominal AUC, marginally sur-



(a) ROC curve (b) number of bit operations

Figure 9.13.: Effect of different ΔTDC binning schemes, after applying pre-quantization to all graph input features. The reference configuration uses the “ADC flat noise” binning, while the five tested ΔTDC modes are "no pre-quantization" on TDC values, “flat signal”, “flat background”, “flat noise”, and “equidistant”. The best performance is obtained with "flat noise" TDC binning scheme at AUC=0.9533.

passing both the un-quantized reference and the alternative binning strategies. By construction, this binning depends on the detailed shape of the background distribution in the calibration sample and is therefore expected to be less robust against changes in the background conditions than either a signal-driven or a purely equidistant binning. Nevertheless, the practical impact of this background dependence is anticipated to be small and can be mitigated by a future retuning of the bin edges, if required. For the purposes of the current analysis, the *flat noise* scheme is therefore adopted as the default configuration, as it offers a modest performance gain while retaining the flexibility to update the binning should the background environment in the CDC evolve significantly in subsequent data collection periods.

In a next step, I test the effect of pre-quantization of the other input features in addition to ADC on the performance. For the continuous layer (`clayer`) ID, I apply a simple equidistant binning in the range $[0, 55]$ and for $\Delta\phi$ that already exhibits discrete values by graph construction. Also, the `clayer` distance Δclayer can only take the values 0,1 or 2 by construction and is therefore quantized by design. For the time-related edge feature ΔTDC, again I test the four different binning schemes discussed in the ADC case with

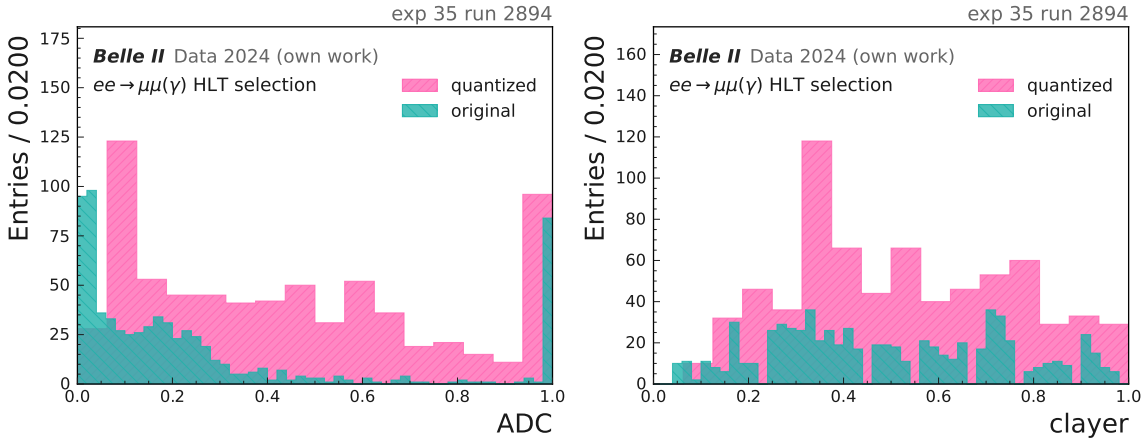


Figure 9.14.: Comparison of the original normalized floating-point node features and their quantized counterparts for ADC and continuous layer index (*clayer*) in the HLT-selected `mumuskim` data sample (exp. 35, run 2894). The ADC values are quantized using the "flat noise" binning scheme, which accounts for the observed deviation of the resulting distribution from the original one.

corresponding bin edges

$$\begin{aligned}
\text{flat signal:}\quad & (-\infty, -138, -88, -62, -42, -28, -18, -8, 8, 18, 28, 42, 62, 88, 138, 342, \infty)\\
\text{flat background:}\quad & (-\infty, -70, -32, -18, -2, 0, 0, 0, 2, 18, 32, 70, 342, \infty)\\
\text{flat noise:}\quad & (-\infty, -210, -144, -96, -60, -36, -20, -4, 4, 20, 36, 60, 96, 144, 210, 342, \infty)\\
\text{equidistant:}\quad & (-\infty, -300, -256, -214, -170, -128, -84, -42, 42, 84, 128, 170, 214, 256,\\
& 300, 342, \infty).
\end{aligned}$$

The resulting AUC values shown in Figure 9.13 remain rather similar across all configurations, indicating that the additional pre-quantization of the timing edges has only a minor impact on the overall classifier performance.
An additional beneficial effect of performing feature quantization prior to network inference is the complete suppression of bit overflow events within the network, beyond the overflow prevention already achieved through pruning.
An overview of the comparisons between the original normalized floating-point node and edge inputs and their quantized counterparts is presented in Figure 9.14 and Figure 9.15, respectively. These comparisons indicate that the global distributions are largely preserved, and in both cases, the full dynamic range of the features, spanning from -1 to 1, is effectively utilized.

9.2.9. Compression summary

The cumulative impact of the individual compression steps on classifier performance and computational cost is summarized in Figure 9.16. Starting from the full-precision reference model, which achieves an AUC of 0.9374 at a computational cost of 1 100 MBOPs, successive optimizations, namely restriction to a single hit per wire and event, selection

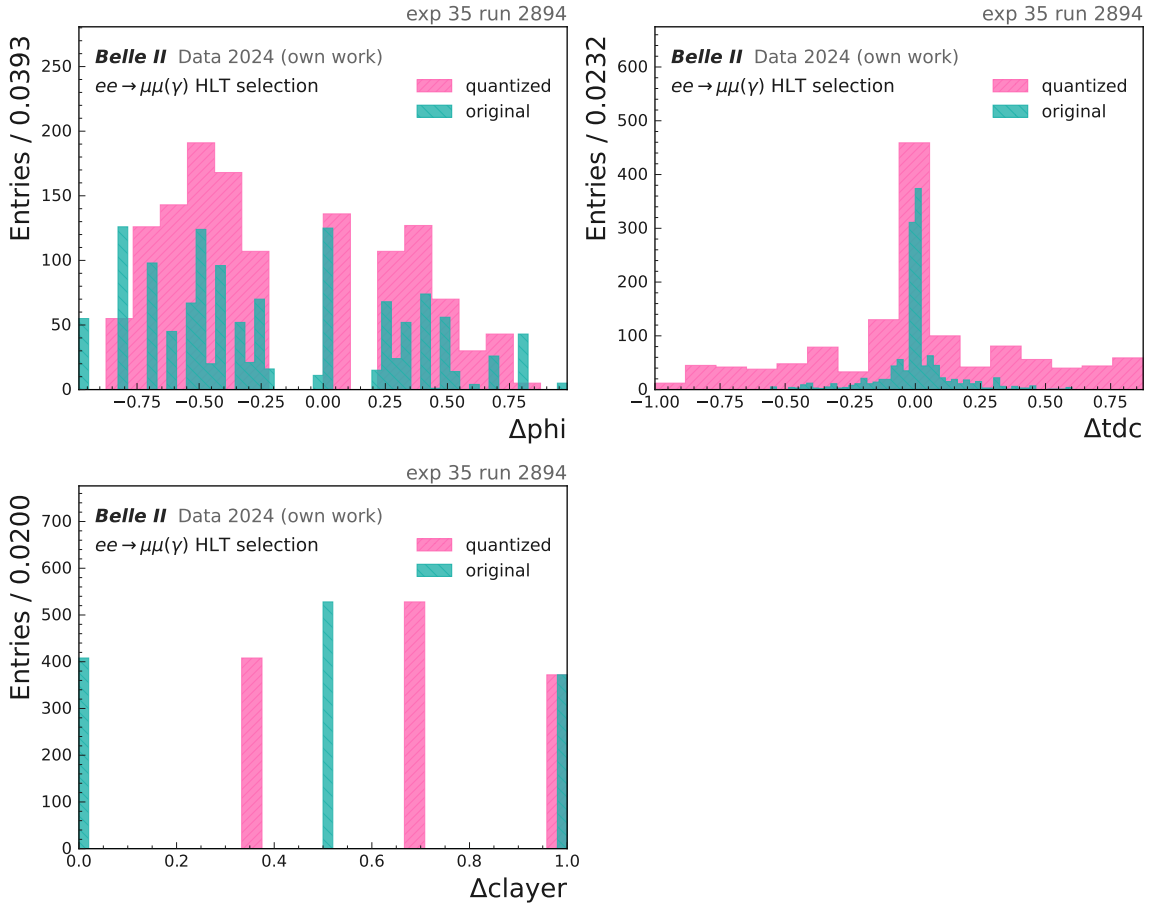
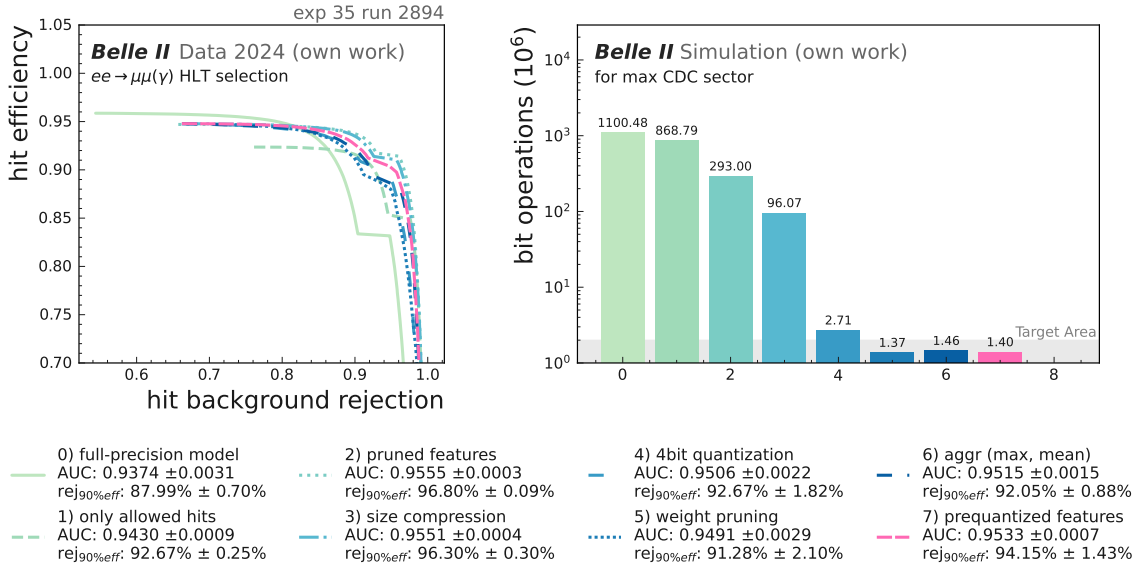


Figure 9.15.: Comparison of the original normalized floating-point edge features and their quantized counterparts for $\Delta\phi$, ΔTDC, and Δclayer in the HLT-selected `mumuskim` data sample (exp. 35, run 2894). The quantized representations reproduce the main structures of the angular, timing, and layer-difference spectra, with only minor discretization artifacts due to the finite binning.

of hits on TRG layers only, input feature pruning, and model size compression, collectively increase the AUC to 0.9551, while simultaneously reducing the BOP count by more than an order of magnitude.

Subsequent introduction of 4-bit quantization for the network weights results in a slight reduction of the AUC to 0.9506, but further decreases the number of BOPs to 2.71 MBOPs, thereby demonstrating that low-precision arithmetic is viable without incurring a substantial degradation in physics performance. The additional weight pruning marginally reduces the AUC to 0.9491, yet again improving the computational efficiency by eliminating redundant parameters. Adopting the *max, mean* aggregation scheme recovers the AUC to 0.9515 with an essentially unchanged BOP cost. Finally, pre-quantization of the input features yields the overall best configuration, achieving an AUC of 0.9533 at a computational cost of 1.36 MBOPs within the targeted resource budget. Overall, the final design realizes a compact, low-precision model whose hit-level accuracy even surpasses



(a) ROC curve

(b) number of bit operations

Figure 9.16.: Summary of the impact of successive compression and optimization steps on the hit-level ROC curve (left) and the BOP per maximum CDC sector (right) for the GNN-based hit classifier, evaluated on the HLT-selected `mumuskim` data sample (exp. 35, run 2894). Configurations 0-7 correspond to the full-precision model, restriction to only allowed hits, feature pruning, size compression, 4-bit quantization, weight pruning, "max, mean" aggregation, and pre-quantized input features, respectively. The final configuration with pre-quantized features attains an AUC of 0.953 at a reduced BOP budget by orders of magnitude compared to the full-precision baseline.

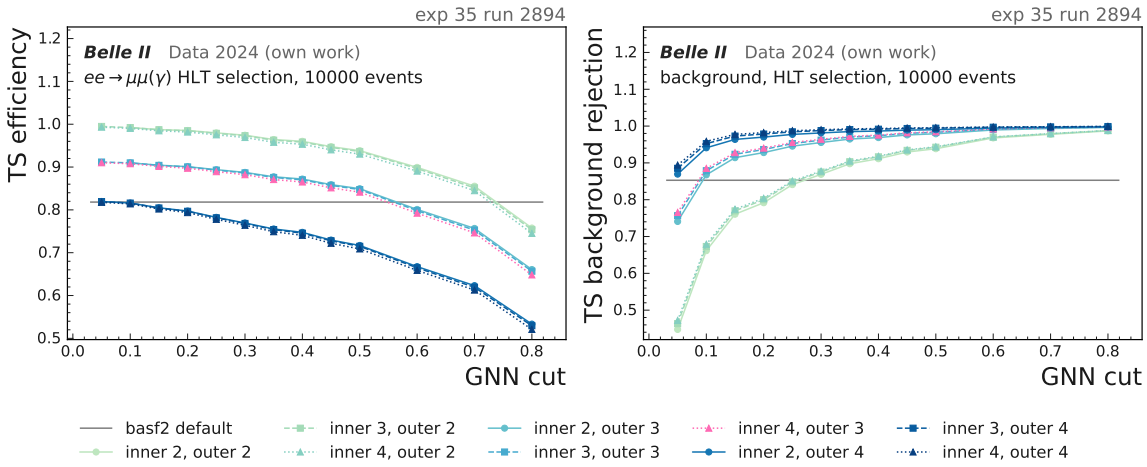
that of the larger full-precision baseline, while achieving a model size that is three orders of magnitude smaller than that of the initial model.

9.3. Integration into the Level-1 trigger pipeline

For the integration of the GNN hit filtering into the CDC trigger pipeline (described in section 4.2), it is necessary to modify several parameters of the downstream trigger modules.
The primary reason is that the current configuration is optimized for unfiltered hits. Consequently, when the input inherits a lower wire occupancy due to a higher hit background rejection, initially fewer TSs are formed. This in turn leads to a substantial reduction in track finding efficiency, as subsequent algorithms require at least four axial TSs in subsequent layers to reconstruct a track.
Preliminary investigations suggest that this effect persists even under idealized conditions. In particular, assuming a hypothetical and perfectly efficient hit-filtering algorithm that removes 100 % of background hits while retaining 100 % of signal hits, a significant fraction of tracks can no longer be reconstructed.
The principal tunable parameters are those of the TSF and track finding modules, which will be discussed in the following sections. The impact of the design decisions is assessed using 10 000 `mumuskim` and `background-only` events, respectively.

9.3.1. Number of hits per track segment

The first modification examined concerns the minimum number of hits required in subsequent layers for a TS to form. At present, this requirement is set to four hits for TSs in both the innermost super-layer and all other super-layers. As detailed in subsection 4.2.1, this condition is implemented via LUTs. For this study, I generated dedicated LUTs for an arbitrary number of hit layers ranging from one to five, as well as for the total number of hits in a TS, without imposing any restriction on the specific number of layers in which hits occur. However, the latter configuration did not yield any improvement relative to the default configuration and is therefore ignored in the following discussions.
The results obtained for different LUT configurations of the innermost and outer super-layers, evaluated at the TS level, are presented in Figure 9.17. The corresponding impact on the track and trigger rate levels in terms of trigger track efficiency and STT trigger rate as defined in Table 9.2-7.4.1 are shown in Figure 9.18.
These results demonstrate that the requirement on the number of hit layers in the innermost super-layer ("inner") has a negligible impact on the overall performance, particularly on the signal and trigger efficiencies. In contrast, the requirement on the number of hit layers in the remaining super-layers ("outer") exhibits a substantial effect on both the efficiency and the level of background rejection, as well as on the trigger rate for background-dominated events. Moreover, although the default configuration ("inner 4, outer 4") achieves the strongest TS-level background suppression and, consequently, an even lower background trigger rate than the baseline configuration, it consistently underperforms relative to the baseline in terms of efficiencies, down to the lowest GNN score thresholds.
Conversely, the configurations employing two hit layers in the outer super-layers ("outer 2", light green) achieve track reconstruction efficiencies exceeding 80 % (in contrast to the baseline value of 59 %), but display a pronounced dependence of the STT trigger rate on



(a) track segment efficiency evaluated on the `mumuskim` sample

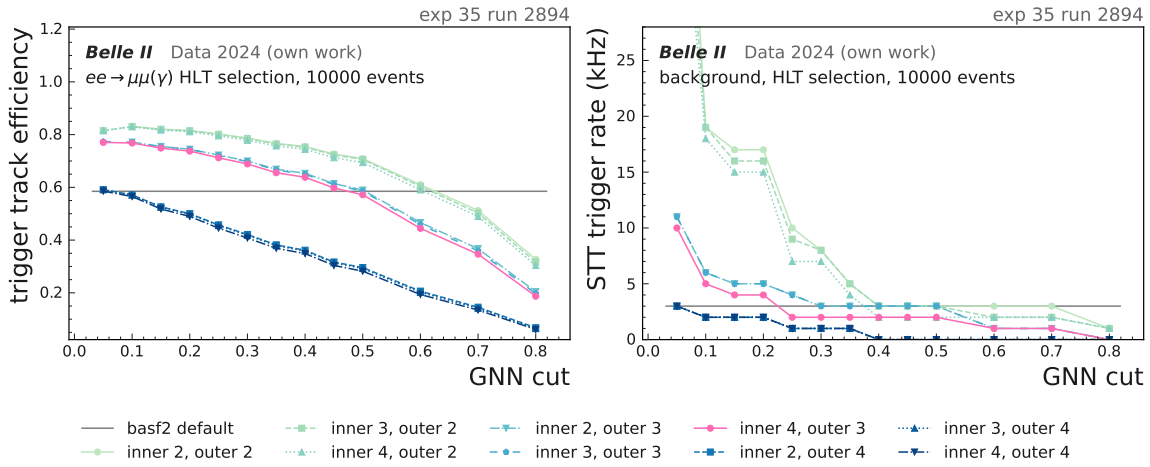
(b) track segment background rejection evaluated on the `background-only` sample

Figure 9.17.: TS efficiency evaluated on $\mu^+\mu^-(\gamma)$ HLT-selected data (`mumuskim`) (a) and background rejection evaluated on background only data (b) as a function of the GNN threshold cut position. The curves correspond to different minimum hit layer requirements in the inner and outer super-layers during TS construction. The configuration that imposes a requirement of four hits in the innermost super-layer and three hits in the outer super-layers ("inner 4, outer 3") achieves an intermediate level of both efficiency and background rejection. The basf2 baseline without hit filtering prior to TS building is shown for reference (grey).

the applied GNN selection threshold, rendering them comparatively unstable for use in the trigger system.
As a compromise between maximizing track reconstruction efficiency and maintaining a low and stable trigger rate, I identify a configuration with three hit layers per outer TS, while retaining four hit layers in the inner-most super-layer ("inner 4, outer 3") as the best option.

9.3.2. Number of track segments

In a higher abstract-level step, I adjust the number of track segments that are needed for a track to be found. The current configuration requires TSs in four subsequent axial super-layers and three subsequent stereo super-layers for a track to be found, where each TS requires four layers hit ("TS 4, axial 4"). This requirement is a very effective background filter, in particular in lower background scenarios. However, in order to meet this requirement, by design only tracks that traverse four out of five axial super-layers are reconstructible, *i.e.* low momentum tracks or tracks in the end-caps that leave the CDC earlier are not reconstructible.
The combined impact of the minimum required number of aligned axial super-layer TSs and the minimum number of hit layers per outer super-layer TS is presented in Fig-



(a) trigger track efficiency evaluated on `mumuskim` sample

(b) STT trigger rate evaluated on `background-only` sample

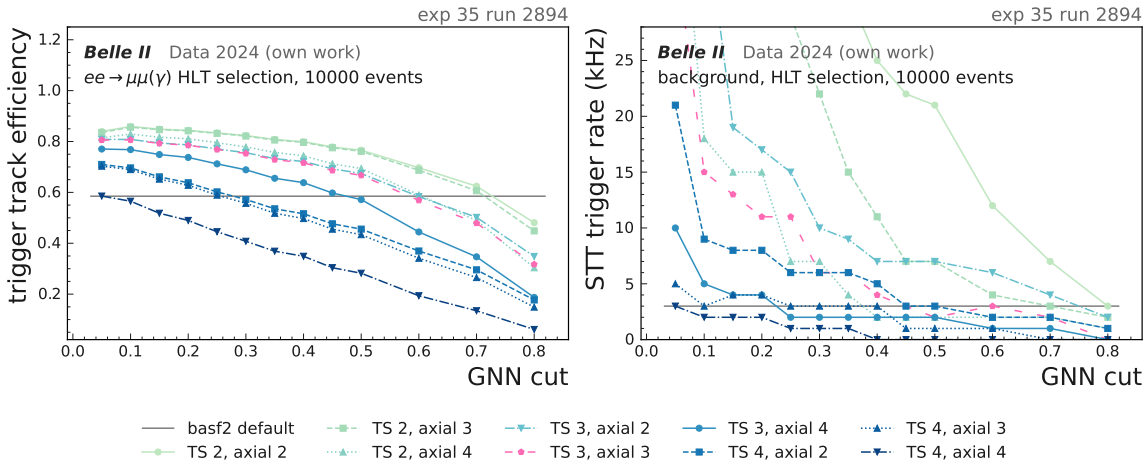
Figure 9.18.: Trigger track efficiency (a) and STT rate (b) for different minimum hit layer requirements in the inner and outer super-layers during TS construction. The track efficiency closely follows the TS efficiency curve. The number of "outer" hits associated with each TS exerts a substantial influence on the overall trigger rate. In particular, the "outer 2" configurations yield very high trigger rates, exceeding the basf2 reference by up to approximately an order of magnitude at low GNN threshold values.

ure 9.19. The minimum number of hit layers required per TS exhibits a pronounced influence on both the tracking efficiency and the STT trigger rate, as already discussed in the previous section. In contrast, the dependence of both metrics on the axial super-layer TSs is comparatively weaker, although generally more pronounced for the STT trigger rate. Restricting the discussion to the previously identified configuration of three hits per TS, I chose an operating point characterized by the requirement of three azimuthally aligned axial TSs ("TS 3, axial 3"). This selection is adopted as a compromise between maintaining high signal efficiency and achieving a sufficiently low trigger rate from background processes.

All remaining trigger chain hyper-parameters including the number of required TSs in stereo super-layers and Hough cluster shape variables exhibit only a marginal effect on the performance and are therefore kept at their default values. The corresponding parameter-dependent plots are provided in the appendix (A.2.4).

9.4. Baseline filtering comparisons

For the evaluation of the impact of GNN-based hit filtering on the baseline trigger track reconstruction, I employ the same sample categories and reconstruction settings as in the design optimization phase. The evaluation dataset consists of 50 000 events for the `mumuskim` sample and additional 50 000 events for the `background-only` sample for three



(a) evaluated on `mumuskim` sample (b) evaluated on `background-only` sample

Figure 9.19.: Evaluation of the impact of different minimum hit layer requirements in the outer super-layers during TS construction as well as the required number of aligned axial TSs. The configuration "TS 3, axial 3", which requires three hits for a TS to be built and three aligned axial track segments for a track to be reconstructed, achieves trigger rates comparable to the basf2 baseline for GNN cut values above 0.4, while simultaneously providing higher track reconstruction efficiency for cut values up to 0.6.

different background scenarios. In the future scenario exp. 0, instead of collision data, MC simulation is used.

For the GNN, I use the final configuration of the model obtained after optimization detailed in section 9.2. For each background scenario, an independent GNN model is trained on simulated samples generated with a background level that matches that of the corresponding evaluation dataset. As a working point I use configurations such that the STT trigger rate is roughly $\mathcal{O}(8\,\text{kHz})$.

For experiments 35 and 37, I train models using a BCE loss with a weight of 5 and employ a GNN classification threshold at 0.1, whereas for exp. 0, I account for the substantially higher background by increasing the loss weight applied in model training for this background to 8, the GNN threshold to 0.3 and the number of layers hit per TS to 4.

9.4.1. Hit filtering and track segment performance

In Table 9.1, I present the impact of the GNN-based filter configuration at the hit-level in comparison with the unfiltered reference. Since the default trigger simulation does not apply an explicit hit-level filtering beyond the TSF, I report hit-level metrics only for the filtered configuration. The hit and TS efficiencies quoted are evaluated in the `mumuskim` samples, while the background rejections are derived from samples that contain only background events.

Table 9.1.: Hit-level performance metrics of the GNN filter compared to the default basf2 filtering methods for different background levels. The metrics include hit efficiency, hit background rejection, TS efficiency and TS background rejection, with statistical uncertainties indicated.

Hit/TS Metrics evaluated on	**Hit Eff.** (%) $\mu^+\mu^-$ (γ)	**Hit Bkg. Rej.** (%) background-only	**TS Eff.** (%) $\mu^+\mu^-$ (γ)	**TS Bkg. Rej.** (%) background only
Exp. 35 run 2894				
basf2 TSF	–	–	$81.82^{+0.33}_{-0.33}$ +8.96	$85.28^{+0.07}_{-0.07}$ +3.22
TSF + GNN	$94.49^{+0.09}_{-0.09}$	$82.19^{+0.03}_{-0.03}$	$90.78^{+0.25}_{-0.25}$	$88.50^{+0.06}_{-0.06}$
Exp. 37 Run 1893				
basf2 TSF	–	–	$75.04^{+0.38}_{-0.38}$ +9.34	$84.36^{+0.07}_{-0.07}$ +6.86
TSF + GNN	$94.64^{+0.10}_{-0.10}$	$84.20^{+0.02}_{-0.02}$	$84.38^{+0.32}_{-0.32}$	$91.22^{+0.05}_{-0.05}$
Exp. 0 Run 0				
basf2 TSF	–	–	$90.59^{+0.25}_{-0.25}$ −15.59	$90.13^{+0.04}_{-0.04}$ +9.20
TSF + GNN	$91.46^{+0.12}_{-0.12}$	$82.68^{+0.02}_{-0.02}$	$76.11^{+0.38}_{-0.38}$	$99.33^{+0.01}_{-0.01}$

For both data-taking runs, the GNN filter outperforms the basf2 TSF at the TS level. In exp. 35, the TS efficiency increases by almost 9 %pt with respect to the baseline, accompanied by an improvement in the background rejection of 3 %pt. A comparable behavior is observed for exp. 37, where the TS efficiency gain reaches 9.3 %pt and the background rejection improves by almost 7 %pt. The larger relative gain in TS efficiency for exp. 37 can be attributed to the poorer performance of the unfiltered TSF in this run compared to exp. 35, leading to a lower absolute TS efficiency in the baseline configuration. The enhanced background rejection in exp. 37 is a consequence of the generally higher background environment, such that, for identical configuration settings, a larger fraction of background hits is removed.
For the chosen working point used in exp. 0, the results deviate significantly from those obtained in the data-taking runs. Here, the default TSF achieves a high TS efficiency of 90.6 %, which can be understood from the elevated background level that facilitates satisfying the four-layer hit requirement in the TS construction. However, this also leads to a substantial increase in fake tracks and trigger rates, which will be examined in the following section. At the selected working point, the GNN filter increases the background rejection to 99.3 %, at the cost of a reduced efficiency of 76.1 %.

9.4.2. Trigger track performance

In the following, I compare the track and trigger-bit-level performance of the three track finding configurations introduced at the beginning of this chapter (`2DHough`, `3DHough`, `adjusted 3DHough+GNN`).
In all experimental scenarios, the `2DHough` substantially exceeds the STT rate limits (see Table 9.4), despite the fact that this data was originally recorded with the `2DHough` finder

Table 9.2.: Track-level performance metrics of the adjusted 3DHough + GNN filter compared to the basf2 2DHough and 3DHough tracking. Metrics shown are fitted trigger tracks efficiency, fake rate, clone rate, z_0 resolution and p_T resolution.

Track Metrics evaluated on	**Efficiency (%)** $\mu^+\mu^-$ (γ)	**Fake Rate (%)** $\mu^+\mu^-$ (γ)	**Clone Rate (%)** $\mu^+\mu^-$ (γ)	**z_0 Res. (cm)** $\mu^+\mu^-$ (γ)	**p_T Res. (%)** $\mu^+\mu^-$ (γ)
Exp. 35 run 2894					
basf2 2DHough	$68.19^{+0.28}_{-0.28}$ +8.56	$5.78^{+0.17}_{-0.16}$ −3.36	$0.55^{+0.06}_{-0.05}$ −0.55	$1.70^{+0.02}_{-0.01}$ +0.03	$17.09^{+0.23}_{-0.27}$ −0.45
basf2 3DHough	$58.44^{+0.30}_{-0.30}$ +18.31	$2.12^{+0.12}_{-0.11}$ +0.3	$0.01^{+0.01}_{-0.01}$ −0.01	$1.83^{+0.02}_{-0.02}$ −0.10	$16.57^{+0.28}_{-0.23}$ +0.07
adj. 3DHough + GNN	$76.75^{+0.25}_{-0.26}$	$2.42^{+0.11}_{-0.11}$	$0.00^{+0.01}_{-0.00}$	$1.73^{+0.01}_{-0.01}$	$16.64^{+0.18}_{-0.20}$
Exp. 37 Run 1893					
basf2 2DHough	$55.18^{+0.17}_{-0.17}$ +12.68	$9.24^{+0.13}_{-0.13}$ −6.13	$0.66^{+0.04}_{-0.04}$ −0.65	$1.73^{+0.01}_{-0.01}$ +0.02	$18.02^{+0.15}_{-0.17}$ −0.95
basf2 3DHough	$50.26^{+0.17}_{-0.17}$ +17.60	$3.11^{+0.08}_{-0.08}$ +0.15	$0.01^{+0.01}_{-0.00}$	$1.90^{+0.01}_{-0.01}$ −0.15	$16.74^{+0.17}_{-0.13}$ +0.33
adj. 3DHough + GNN	$67.86^{+0.16}_{-0.16}$	$3.26^{+0.07}_{-0.07}$	$0.01^{+0.01}_{-0.00}$	$1.75^{+0.01}_{-0.01}$	$17.07^{+0.13}_{-0.07}$
Exp. 0 Run 0					
basf2 2DHough	$15.00^{+0.14}_{-0.14}$ −15.24	$72.27^{+0.19}_{-0.19}$ +71.99	$27.22^{+0.36}_{-0.36}$ +27.22	$3.57^{+0.03}_{-0.05}$ +2.97	$24.06^{+1.22}_{-0.83}$ −6.18
basf2 3DHough	$65.19^{+0.18}_{-0.18}$ −34.95	$0.58^{+0.04}_{-0.03}$ −0.31	$0.04^{+0.01}_{-0.01}$ −0.04	$6.69^{+0.04}_{-0.03}$ −0.15	$17.04^{+0.16}_{-0.15}$ +0.84
basf2 3DHough ×1/4	$16.30^{+0.36}_{-0.36}$ +13.96	$0.15^{+0.08}_{-0.06}$ +0.12	$0.01^{+0.02}_{-0.02}$ −0.01	$6.69^{+0.08}_{-0.06}$ −0.15	$17.04^{+0.32}_{-0.30}$ +0.84
adj. 3DHough + GNN	$30.24^{+0.18}_{-0.18}$	$0.27^{+0.04}_{-0.03}$	$0.00^{+0.00}_{-0.00}$	$6.54^{+0.05}_{-0.04}$	$17.88^{+0.19}_{-0.15}$

deployed as the active hardware trigger. A trigger rate of 45 kHz or higher would not have been operationally acceptable, indicating that the simulated 2DHough configuration does not faithfully reproduce the behavior of the hardware trigger used during data collection.

I identify two explanatory hypotheses: First, the 2DHough finder implemented in the simulation may not precisely correspond to the version that was running on the trigger hardware during data taking. Second, the finder configuration parameters may have been incorrectly retrieved from the conditions database.

A comprehensive debugging and validation of the 2DHough simulation was not feasible within the available time. The 2DHough results should therefore be treated with caution and are presented primarily for completeness. The main focus of this study is the 3DHough-based configuration, and all conclusions in the subsequent sections are drawn from the comparison between the adjusted 3DHough+GNN configuration and the 3DHough baseline.

The results for exp. 0 are obtained using MC-simulated background events. Since these backgrounds may not be modeled with full accuracy, the corresponding results should also be interpreted with care. Accordingly, the primary emphasis is placed on the data-driven results from experiments 35 and 37, while the outcome of exp. 0 is included mainly for completeness.

The performance at the track-level of different track reconstruction configurations is shown in Table 9.2. For both data-taking runs, the GNN configuration achieves a substantially higher track reconstruction efficiency than either baseline. In exp. 35, efficiency reaches 76.75 %, representing a gain of 18.31 %pt over the 3DHough baseline. Similar improvements are observed in exp. 37, where the efficiency of 67.86 % exceeds the 3DHough baseline by 17.60 %pt. These gains are directly attributable to the higher TS efficiencies discussed in subsection 9.4.1.

The fake rate of the GNN configuration remains comparable to the 3DHough baseline with a small increase of 0.15 %pt. The clone rate is effectively zero for both experiments, matching or improving on both baselines. Furthermore, the z_0 resolution is improved compared to the 3DHough case, while the relative p_T resolution is decreased.

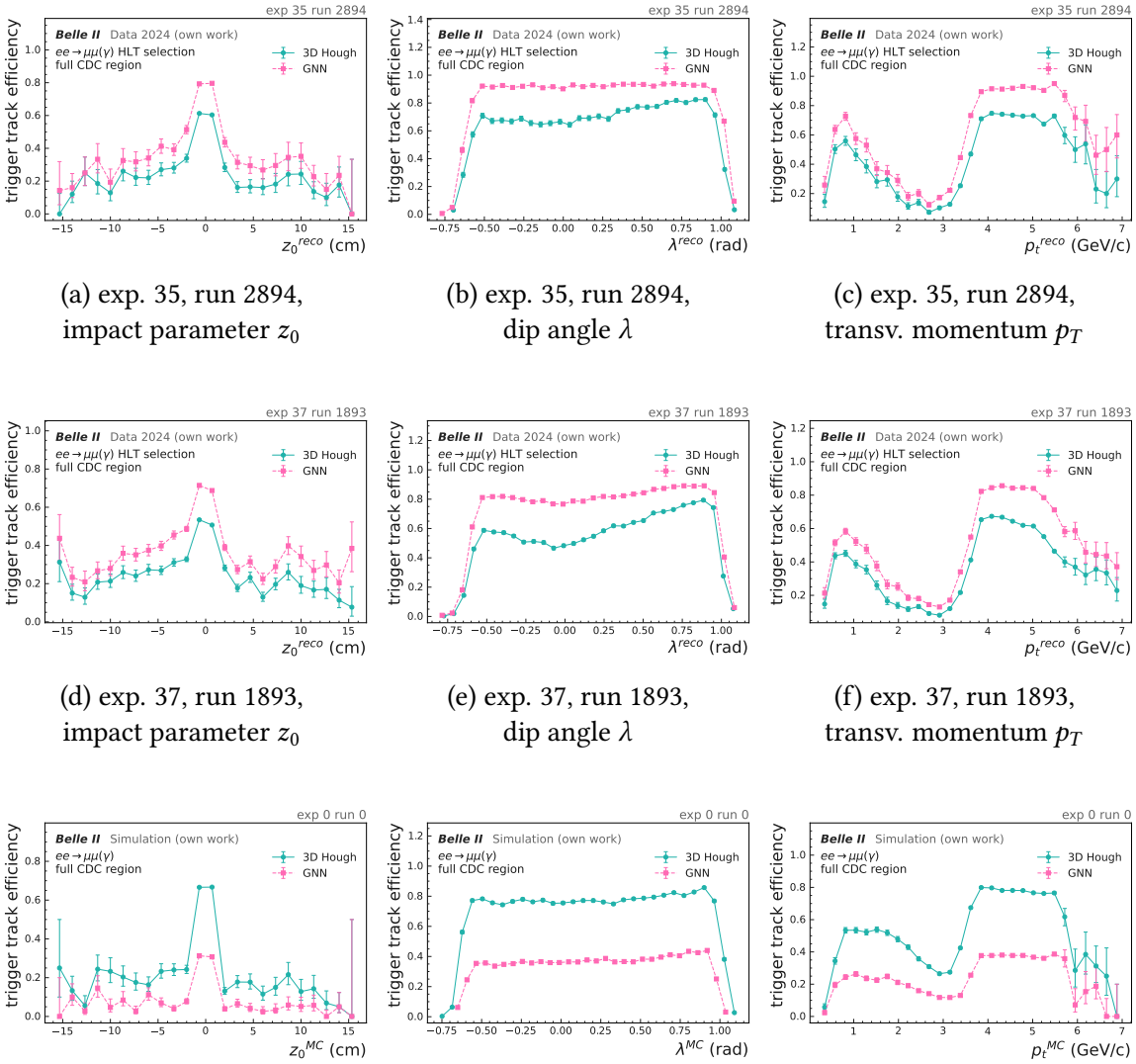
The simulated future scenario (exp. 0) again exhibits a qualitatively different behavior, consistent with the effect discussed for the TS level. The basf2 2DHough configuration is severely affected by the low background suppression, with a track efficiency of only 15 % and a fake rate exceeding 72 %, rendering it effectively non-functional in this regime. One explanation for the low track efficiency might be the requirement of 40 % hit purity of a found track to be counted as a found signal track as defined in section 7.4. The 3DHough baseline performs substantially better with an efficiency of 65 % and a fake rate below 0.6 %. However, the associated STT rate as discussed in subsection 9.4.3 exceeds the limits by roughly a factor of four. Therefore, I additionally consider a configuration of the 3DHough algorithm that assumes an input pre-scaling of trigger tracks by a factor of four (3DHough x1/4). Under this condition, the efficiency decreases to 16.30 %. The 3DHough+GNN configuration, achieves an efficiency of only 30.24 %, which lies significantly below the 3DHough baseline but almost twice the efficiency obtained by the hypothetical pre-scaled 3DHough x1/4. This efficiency drop is the direct consequence of the necessary high GNN cut, which aggressively suppresses background at the hit level, and in doing so also removes a non-negligible fraction of signal TSs. The fake rate of 0.27 % is therefore only half of the 3DHough case.

The trigger track fitting efficiencies are presented in Figure 9.20 as functions of the impact parameter z_0, the dip angle λ, and the transverse momentum p_T of the truth signal track. For experiments 35 and 37, the truth information is derived from the offline reconstruction of tracks, whereas for exp. 0 it is obtained directly from the generated MC particles.

In agreement with Table 9.2, the overall trigger track efficiency increases when incorporating the GNN filter for the two data-taking periods, experiments 35 and 37, while it decreases for experiment 0 due to the chosen working point that restricts the trigger rate to remain within the predefined limits. The 3DHough configuration shown is not tuned with respect to the trigger rate. Consequently, for exp. 0, the corresponding curve serves purely as a reference and would not be compatible with the present trigger configuration.

For experiments 35 and 37, the improvement in trigger track efficiency introduced by the GNN is approximately uniform over the full z_0 and p_T ranges, with a slightly more pronounced gain at higher p_T values. Performance distributions as a function of p_T are analogous to those observed in the offline tracking scenario discussed in subsection 8.2.3. A more detailed examination of the shapes of these curves is provided therein. The improvement in efficiency as a function of λ is similarly consistent across the entire angular range. Moreover, the pronounced drop in track efficiency observed without the GNN filter near $\lambda \approx 0$, which is in particular visible for the exp. 37 case, and for tracks in the backward region, is largely mitigated by the GNN filter.

For the observables z_0, λ, and p_T, as well as for the number of hits associated with the true signal track, the efficiencies in the different detector regions are provided in the appendix (A.18-A.21).



(a) exp. 35, run 2894, impact parameter z_0

(b) exp. 35, run 2894, dip angle λ

(c) exp. 35, run 2894, transv. momentum p_T

(d) exp. 37, run 1893, impact parameter z_0

(e) exp. 37, run 1893, dip angle λ

(f) exp. 37, run 1893, transv. momentum p_T

(g) exp. 0, run 0, impact parameter z_0

(h) exp. 0, run 0, dip angle λ

(i) exp. 0, run 0, transv. momentum p_T

Figure 9.20.: Trigger track fitting efficiency as functions of the impact parameter z_0, the dip angle λ and the transverse momentum p_T for three different experiments with increasing background levels comparing the basf2 `3DHough` finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for exp. 35 ((a)-(c)) and 37 ((d)- (f)) and generated $\mu^+\mu^-$ pairs for exp. 0 ((g)-(i)).

Table 9.3.: Performance of the STT trigger line, the inclusive CDC trigger signal ⋁CDC obtained by combining all CDC trigger bits after pre-scaling, and the total L1 trigger signal incorporating all sub-detector contributions. The corresponding trigger rates are presented for the adjusted `3DHough` + GNN configuration and are compared to the basf2 `2DHough` and `3DHough` tracking algorithms, evaluated on 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events. Contrary to expectations, the L1 trigger value does not reach 100 %. This is most likely because it is derived from simulations, which can slightly differ from the corresponding hardware implementation.

stt TRG bit	Efficiency (%)	Fake Rate (%)	Rate (kHz)	⋁CDC Eff. (%)	L1 Eff. (%)
Exp. 35 run 2894					
basf2 2DHough	$77.09^{+0.34}_{-0.35}$ +5.41	$0.39^{+0.06}_{-0.05}$ −0.02	$0.03^{+0.00}_{-0.00}$ +0.01	$78.04^{+0.34}_{-0.34}$ +4.88	$96.12^{+0.15}_{-0.16}$ +1.91
basf2 3DHough	$71.51^{+0.37}_{-0.37}$ +10.99	$0.21^{+0.05}_{-0.04}$ +0.16	$0.03^{+0.00}_{-0.00}$ +0.01	$71.44^{+0.37}_{-0.37}$ +11.48	$95.27^{+0.17}_{-0.18}$ +2.76
adj. 3DHough + GNN	$82.50^{+0.31}_{-0.32}$	$0.37^{+0.06}_{-0.05}$	$0.04^{+0.00}_{-0.00}$	$82.92^{+0.31}_{-0.31}$	$98.03^{+0.11}_{-0.12}$
Exp. 37 Run 1893					
basf2 2DHough	$70.21^{+0.21}_{-0.22}$ +5.88	$0.81^{+0.05}_{-0.05}$ −0.27	$0.02^{+0.00}_{-0.00}$ +0.01	$70.76^{+0.21}_{-0.21}$ +5.51	$94.68^{+0.10}_{-0.11}$ +1.52
basf2 3DHough	$64.50^{+0.22}_{-0.23}$ +11.59	$0.40^{+0.04}_{-0.04}$ +0.14	$0.02^{+0.00}_{-0.00}$ +0.01	$64.51^{+0.22}_{-0.22}$ +7.76	$94.05^{+0.11}_{-0.11}$ +2.15
adj. 3DHough + GNN	$76.09^{+0.20}_{-0.20}$	$0.54^{+0.04}_{-0.04}$	$0.03^{+0.00}_{-0.00}$	$76.27^{+0.20}_{-0.20}$	$96.20^{+0.09}_{-0.09}$
Exp. 0 Run 0					
basf2 2DHough	$63.49^{+0.60}_{-0.61}$ −18.73	$5.71^{+0.36}_{-0.34}$ −5.70	$0.45^{+0.00}_{-0.00}$ −0.17	$97.83^{+0.17}_{-0.19}$ −53.98	$99.97^{+0.02}_{-0.03}$ −32.49
basf2 3DHough	$76.15^{+0.22}_{-0.22}$ −31.39	$0.04^{+0.01}_{-0.01}$ −0.03	$0.48^{+0.00}_{-0.00}$ −0.20	$76.34^{+0.21}_{-0.22}$ −32.49	$99.74^{+0.02}_{-0.03}$ −0.43
basf2 3DHough ×1/4	$19.04^{+0.44}_{-0.44}$ +25.72	$0.01^{+0.02}_{-0.01}$	$0.12^{+0.00}_{-0.00}$ +0.16		
adj. 3DHough + GNN	$44.76^{+0.26}_{-0.26}$	$0.01^{+0.01}_{-0.01}$	$0.28^{+0.00}_{-0.00}$	$43.85^{+0.25}_{-0.25}$	$99.31^{+0.04}_{-0.04}$

9.4.3. Trigger rates

Table 9.3 presents the performance of the STT bit, evaluated on the `mumuskim` samples, compared to the inclusive CDC and L1 trigger efficiencies. At the track level, the efficiency for the two data-taking runs is enhanced by more than 10 %pt when employing the GNN, relative to the `3DHough`-based configuration. In contrast, for exp. 0 the efficiency is reduced with respect to the unmodified basf2 configuration. However, once the `3DHough` configuration is pre-scaled to achieve a realistic STT background trigger rate, the efficiency gain for the `3DHough+GNN` exceeds 25 %pt.

Due to the higher track fake rate in the GNN configuration also the STT fake rate increases relative to the `3DHough` configuration.

The trigger rate for this physics process remains consistently low across the data-taking runs and all configurations, at approximately 0.03 kHz, and increases up to 0.28 kHz for the GNN configuration in exp. 0.

The inclusive CDC trigger efficiency ⋁ CDC, obtained by combining all CDC trigger bits, is improved by application of the GNN filter by up to 11.5 %pt in exp. 35 and 7.8 %pt in exp. 37 with respect to the `3DHough` configuration. When also other sub-detector trigger signals from the ECL, KLM, and TOP are included, the total L1 trigger efficiency after pre-scaling increases by $\mathcal{O}(20\,\%\text{pt})$ compared to CDC trigger lines only and reaches efficiencies exceeding 90 %. The relative differences between configurations are reduced since trigger bits from other sub-detectors partially compensate for missing tracks. Nev-

Table 9.4.: Estimated PSNM trigger rates for single non-zero CDC trigger lines, their combined OR value and the L1 signal composed of all sub-detectors. For different runs the adjusted `3DHough` + GNN configuration is compared to the basf2 `2DHough` and `3DHough` tracking evaluated on background only events.

Trigger bits Pre-scale factor	**ffy (kHz)** × 1	**fy30 (kHz)** × 1	**fyb (kHz)** × 1	**fyo (kHz)** × 1	**stt (kHz)** × 1	**syb (kHz)** × 1	**syo (kHz)** × 1
Exp. 35 run 2894							
basf2 2DHough	$3.00^{+0.88}_{-0.68}$	$6.20^{+1.22}_{-1.02}$	$2.20^{+0.77}_{-0.57}$	$3.80^{+0.98}_{-0.78}$	$\mathbf{41.40^{+2.97}_{-2.77}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
basf2 3DHough	$0.00^{+0.20}_{-0.00}$	$0.60^{+0.46}_{-0.26}$	$0.40^{+0.40}_{-0.20}$	$0.40^{+0.40}_{-0.20}$	$\mathbf{3.20^{+0.91}_{-0.71}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
adj. 3DHough + GNN	$0.00^{+0.20}_{-0.00}$	$3.00^{+0.88}_{-0.68}$	$1.60^{+0.67}_{-0.47}$	$2.60^{+0.83}_{-0.63}$	$\mathbf{7.80^{+1.35}_{-1.15}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
Exp. 37 Run 1893							
basf2 2DHough	$12.40^{+1.68}_{-1.48}$	$19.60^{+2.08}_{-1.88}$	$3.80^{+0.98}_{-0.78}$	$8.40^{+1.40}_{-1.20}$	$\mathbf{99.20^{+4.53}_{-4.34}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
basf2 3DHough	$0.20^{+0.32}_{-0.12}$	$1.40^{+0.64}_{-0.44}$	$1.20^{+0.60}_{-0.40}$	$1.40^{+0.64}_{-0.44}$	$\mathbf{6.00^{+1.20}_{-1.00}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
adj. 3DHough + GNN	$0.40^{+0.40}_{-0.20}$	$2.40^{+0.80}_{-0.60}$	$2.00^{+0.74}_{-0.54}$	$2.40^{+0.80}_{-0.60}$	$\mathbf{6.60^{+1.25}_{-1.05}}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$
Exp. 0 Run 0							
basf2 2DHough	$5637.80^{+22.17}_{-22.19}$	$5941.00^{+21.94}_{-21.98}$	$4902.80^{+22.36}_{-22.35}$	$5462.80^{+22.26}_{-22.27}$	$\mathbf{4080.20^{+22.00}_{-21.96}}$	$5.20^{+1.12}_{-0.92}$	$13.80^{+1.76}_{-1.56}$
basf2 3DHough	$1.60^{+0.67}_{-0.47}$	$19.60^{+2.08}_{-1.88}$	$10.00^{+1.52}_{-1.32}$	$18.40^{+2.02}_{-1.82}$	$\mathbf{33.20^{+2.67}_{-2.48}}$	$9.00^{+1.44}_{-1.24}$	$19.80^{+2.09}_{-1.89}$
adj. 3DHough + GNN	$0.00^{+0.20}_{-0.00}$	$2.20^{+0.77}_{-0.57}$	$1.80^{+0.71}_{-0.51}$	$2.20^{+0.77}_{-0.57}$	$\mathbf{8.40^{+1.40}_{-1.20}}$	$1.40^{+0.64}_{-0.44}$	$1.80^{+0.71}_{-0.51}$

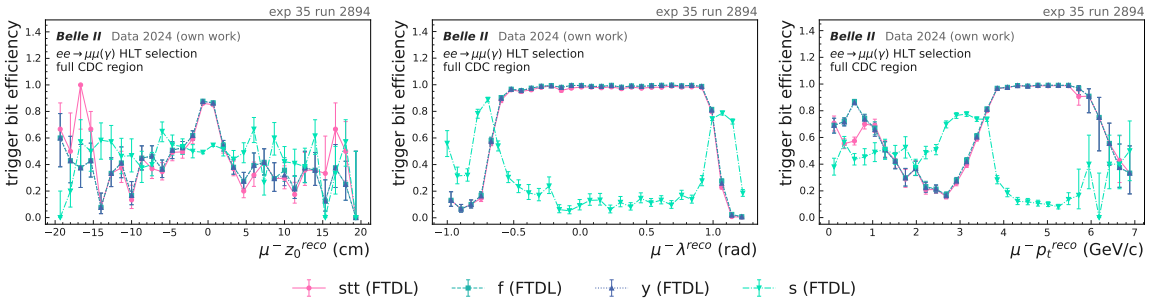
Trigger bits Pre-scale factor	**f (kHz)** × 1/20000	**s (kHz)** × 1/4000	**ssb (kHz)** × 1/10	**stt6 (kHz)** × 1/1000	**y (kHz)** × 1/5000	**$\bigvee$CDC (kHz)**	**L1 (kHz)**
Exp. 35 run 2894							
basf2 2DHough	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.56^{+0.56}_{-0.28}$	$51.07^{+3.91}_{-3.63}$	$694.00^{+11.45}_{-11.28}$
basf2 3DHough	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.28^{+0.45}_{-0.17}$	$4.47^{+1.26}_{-0.99}$	$8.65^{1.70}_{1.42}$
adj. 3DHough + GNN	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$0.00^{+0.20}_{-0.00}$	$0.28^{+0.45}_{-0.17}$	$8.93^{+1.72}_{-1.44}$	$14.51^{+2.15}_{-1.88}$
Exp. 37 Run 1893							
basf2 2DHough	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$92.21^{+4.37}_{-4.18}$	$117.21^{+4.91}_{-4.72}$
basf2 3DHough	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$6.60^{+1.25}_{-1.05}$	$13.20^{+1.73}_{-1.53}$
adj. 3DHough + GNN	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$7.60^{+1.34}_{-1.14}$	$16.00^{+1.89}_{-1.69}$
Exp. 0 Run 0							
basf2 2DHough	$0.40^{+0.40}_{-0.20}$	$0.20^{+0.32}_{-0.12}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$1.00^{+0.56}_{-0.36}$	$135.00^{+5.26}_{-5.06}$	$694.00^{+11.45}_{-11.28}$
basf2 3DHough	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$60.40^{+3.57}_{-3.37}$	$355.00^{+8.37}_{-8.18}$
adj. 3DHough + GNN	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.00^{+0.20}_{-0.00}$	$0.20^{+0.32}_{-0.12}$	$11.40^{+1.61}_{-1.41}$	$66.60^{+3.74}_{3.54}$

ertheless, the GNN configuration still outperforms the `3DHough` configuration by 2.8 %pt in exp. 35 and 2.2 %pt in exp. 37.
In principle, the efficiency of the L1 trigger is expected to approach 100 %, since the events investigated were selected by this specific L1 trigger in conjunction with the HLT. The observed deviation is most likely attributable to the fact that the efficiency estimate is obtained from simulations, which can exhibit small discrepancies relative to the corresponding hardware implementation.

In Table 9.4 I summarize the estimated PSNM trigger rates after the application of prescaling for `background-only` events. The trigger bits are decomposed into individual trigger bits for the three experimental conditions considered.
For the two data-driven runs, both the `3DHough` configuration as well as the `3DHough+GNN` configuration, yield total trigger rates that remain well within the acceptable operational limits. In all configurations, the dominant contribution arises from the STT trigger bit

(highlighted in bold), while the remaining trigger bits are at the level of $O(2\,\text{kHz})$ or effectively zero, in particular for the pre-scaled trigger bits.
In the simulated exp. 0 scenario, the unmodified `3DHough` baseline already exceeds the overall trigger-rate limit in the STT bit and reaches 60.40 kHz when aggregating all CDC trigger bits. In contrast, the GNN-based configuration keeps the STT rate below 9 kHz and produces a total CDC trigger rate of 11.4 kHz, although at the expense of a comparatively low single-track efficiency of 30 %, as discussed in the previous section.
However, once all other sub-detectors are included, the total L1 trigger rate increases to 66.6 kHz for the GNN configuration and 355 kHz for the `3DHough` configuration, respectively. This indicates that the trigger algorithms of other sub-detectors must also be re-optimized to accommodate the anticipated future background conditions. It should be emphasized that these rates are obtained exclusively from background events, with no physics events included. Consequently, no single physics event could be recorded under these conditions, as the full trigger bandwidth is saturated by background. The unscaled FTDL trigger rates for the trigger lines affected by a pre-scale factor are provided in the appendix (A.3).

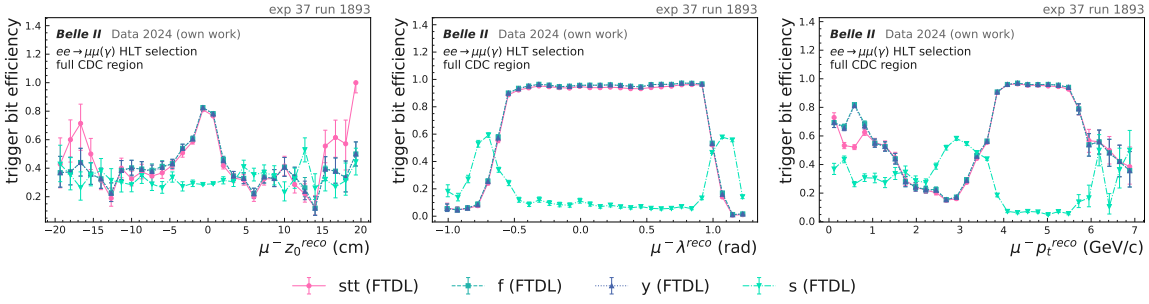
In Figure 9.21, the dependence of the single-track trigger bits (STT, `f`, `y`, `s`) on z_0, λ, and p_T of the signal μ^- is shown. The distributions for the track reconstruction-based trigger bits STT, `f` and `y` are consistent with the track efficiencies presented in Figure 9.20, which is expected, particularly for the `y` bit, whose condition is the existence of at least one fitted track and should therefore coincide with the corresponding track efficiency. Only in the exp. 0 scenario, a slight decrease is observable when going from `f` to `y` to STT, indicating that some tracks are lost between the track finding and track fitting stages, and additionally through the quality requirements on z_0 and p_T imposed by STT.
The short-track bit `s` covers complementary regions of phase space, which is especially evident in the λ scan. There, the distribution for `s` peaks in the end-cap regions where the efficiencies of the other trigger bits decrease. Furthermore, in the p_T scan, the efficiency dip around 2 to 3 GeV/c is recovered for the `s` bit, which is consistent with the expectation that tracks in this p_T range preferentially populate the end-cap rather than the barrel region.
The pre-scaled trigger efficiencies for the given trigger bits, together with the inclusive CDC trigger rate (including all other CDC trigger bits) and the total L1 trigger rate, are provided in the appendix (A.22). As in the background-only case (see Table 9.4), the pre-scaling reduces the `f`, `y`, and `s`-bit rates, and thus their effective efficiencies, to zero in the end-caps. The current pre-scaling factors are tuned to the existing detector and trigger. For example, the `s` bit is heavily pre-scaled since it effectively counts hits and is highly sensitive to background occupancy. Since the GNN filter strongly suppresses background hits, it may allow these pre-scaling factors to be relaxed.
For example, as shown in the appendix (A.3), the `s`-bit rate drops to 3.4 kHz for the exp. 0 GNN working point. With the background hit rate reduced by the GNN, the `s`-bit pre-scaling could be relaxed or removed, further increasing the overall trigger efficiency in the end-caps.
The trigger bit efficiencies for each of the detector regions separately are provided in the appendix (A.23-A.25) .



(a) exp. 35, run 2894, impact parameter z_0

(b) exp. 35, run 2894, dip angle λ

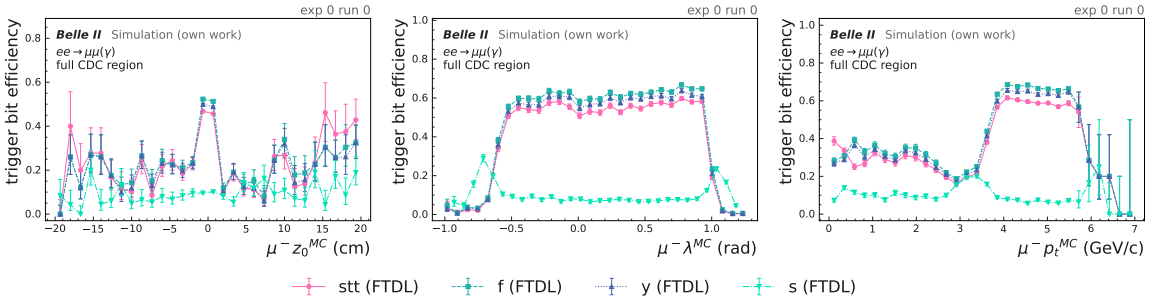
(c) exp. 35, run 2894, transv. momentum p_T



(d) exp. 37, run 1893, impact parameter z_0

(e) exp. 37, run 1893, dip angle λ

(f) exp. 37, run 1893, transv. momentum p_T



(g) exp. 0, run 0, impact parameter z_0

(h) exp. 0, run 0, dip angle λ

(i) exp. 0, run 0, transv. momentum p_T

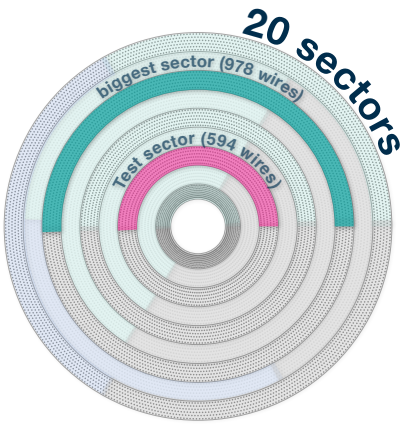
Figure 9.21.: FTDL trigger bit efficiencies for single-track trigger lines (STT, `f`, `y`, `s`) before pre-scaling over impact parameter z_0, the dip angle λ and transverse momentum p_T for three different experiments with increasing background levels for the GNN configuration evaluated on the µm of 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for exp. 35 and 37 and generated $\mu^+\mu^-$ pairs for exp. 0. The corresponding PSNM trigger bits are provided in Figure A.22.

9.5. Implementation in the detector

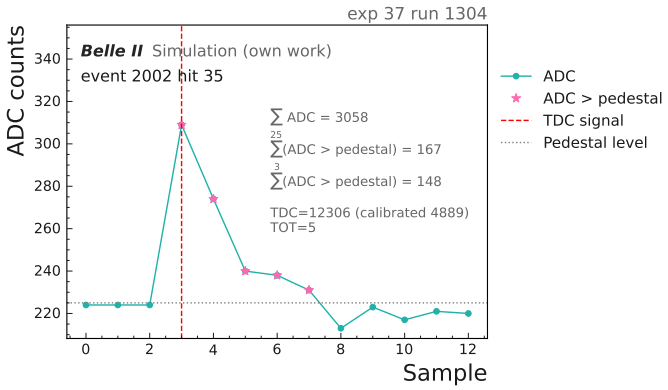
For implementation within the detector, the GNN hit-filtering algorithm must first be validated under realistic operating conditions. This can be achieved by deploying the algorithm on an FPGA for one of the 20 CDC sectors and integrating it into the detector in a parasitic mode, *i.e.* acquiring data without participating in the L1 trigger decision logic.

9.5.1. Test sector implementation

For the implementation of the algorithm on a representative test sector selected from the 20 CDC sectors, several adjustments are required with respect to the configuration employed in the trigger simulation study. The first and most straightforward modification concerns the adjustment of the sigmoid-based model output. In the simulation setup, the output of the GNN is passed through a sigmoid function, and only subsequently a threshold is applied to these transformed outputs for the filtering step. However, on FPGA, the implementation of a sigmoid activation based on the exponential function is computationally costly. Consequently, the raw output of the GNN is employed directly, with the decision threshold adjusted appropriately. In preliminary studies, no difference in classification performance was observed between the two thresholding strategies.
Since, at present, only ADC counts with a resolution of 1 bit are available, the algorithm has to be adapted accordingly by a retraining using a binary ADC representation. In simulation, the ADC threshold is set to eight, while in firmware it is set to six due to the difference in the calculation of the ADC count sum in the L1 trigger and the simulation of it (due to the 3-point sum vs. the 25-point sum). For a future scenario in which ADC binning with 4 bit resolution is available, but the 25-point sum is not, the algorithm must instead be based on the 3-point sum of ADC values (see subsection 3.5.1). A discussion of this 3-point sum adjustment is given in subsection 9.5.2.
Furthermore, the algorithm must be constrained to operate exclusively on the designated test sector rather than relying on global configuration settings. Consequently, the model must be re-trained solely on hits originating from this specific sector. No sector-dependent features are explicitly provided as model inputs; *i.e.* neither the ADC count, the continuous layer index, the azimuthal separation $\Delta\phi$, nor the time difference ΔTDC include any explicit super-layer-dependent encoding. Nevertheless, these quantities, as well as the overall wire occupancy, differ implicitly between super-layers. Therefore, retraining the model on sector-specific data is expected to yield improved performance.
For the test sector, two spare 4-lane `GTH` ports of the TSF number 2 can be allocated. Each of these ports operates at 12 Gbit/s with a clock of 32 MHz and transmits $382 \times 4 = 1\,536$ bit per clock cycle. A single merger board provides 640 bit. Therefore, two merger boards supply 1 480 bit to the new module, corresponding to approximately 33 % wires in the given super-layer. Each FEE board covers 16×3 wires (*e.g.* in super-layer 2 this corresponds to 12 wires in the ϕ direction and 2 wires in the r direction). Each merger board aggregates data from 4 FEEs boards (6 in the ϕ and 1 in the r direction). Consequently, the merger boards `MGR2-0` and `MGR2-1` read out the CDC FEEs boards `CDCFE` 48-51 and `CDCFE` 52-55, respectively.



(a) Partitioning of the CDC into 20 sectors. The test sector, comprising 594 wires, and the largest sector, comprising 978 wires, are indicated. The exact azimuthal orientation of the sectors is likely not represented with full accuracy.



(b) Example waveform from experiment 37, run 1304. The ADC count of 167 is obtained from all sampling points after subtracting the channel-specific pedestal. Using only the first three samples above the pedestal yields an ADC sum of 148. The TDC value is taken at the first sample above the pedestal threshold, and the number of samples above the pedestal (pink stars) defines the TOT, here 5.

The test sector illustrated in Figure 9.22a, which is defined based on the merger boards, comprises a total of 594 wires, corresponding to 495 graph nodes and 2 163 graph edges, considering only wires located in the trigger layers.

9.5.2. ADC 3-point sum

In anticipation of scenarios where the 25-point sum may not be available, or during intermediate detector tests in which 4 bit information is already provided, but only the 3-point sum is implemented, the 3-point sum is also investigated. It should be noted that the use of additional sampled points has direct implications for system latency: each additional sample point increases the latency by 32 ns. Consequently, employing the full 25-point sum for the L1 trigger results in an additional latency of 704 ns.

One limitation of the recorded datasets is that the full waveforms are not recorded. Instead, only the pedestal-subtracted ADC sums are stored, and the underlying waveforms are not simulated either. However, as described in section 6.2 dedicated calibration runs exist in which the complete waveforms are recorded by effectively eliminating pedestal suppression.
One of these dedicated data-taking runs is exp. 37, run 1304, which was conducted under comparatively low background conditions, characterized by an instantaneous luminosity of $\mathcal{L}_{\text{inst}} = 3.5 \cdot 10^{33}\,\text{cm}^{-2}\text{s}^{-1}$. Under these conditions, the events contain on average approximately 900 hits, which is substantially fewer than in, for example, exp. 26, run 1894, where already the additional CDC hits alone amount to $\langle n \rangle_{\text{extraCDChits}} = 1\,219$.

A representative waveform for exp. 37, run 1304 is shown in Figure 9.22b. Here, the raw sum of ADC counts is equal to 3 058. After subtracting the channel-specific pedestal value from each sampling point, the total ADC count is reduced to 167. This value corresponds to the effective 25-point sum, even though the actual number of sampled points is lower than 25. Here, the term "25-point sum" denotes the maximum sum possible if up to 25 sampling points exceeded the pedestal threshold and were therefore included in the integration.
The TOT value is defined as the number of sampling points whose amplitudes exceed the pedestal level (indicated by the pink stars), and the TDC value is assigned to the first sample that surpasses this threshold. The raw TDC value is initially 12 306, and is shifted to the event time t_0=4 889 after calibration. The 3-point sum is obtained by summing three consecutive ADC counts, starting from the sample for which the TDC signal is generated. in the illustrated example this yields a value of 148, which is close to that of the full 25-point sum.
Overall, the 25-point sum and the 3-point sum exhibit comparable magnitudes since, in most cases, not all 25 sampling points contribute significantly to the total count. Instead, only a small subset of samples exceeds the pedestal and thus dominates the aggregate signal.
The 3-point sum can, in general, be interpreted as a smoothed or "blurred" representation of the 25-point sum, *i.e.* corresponds to a wider distribution that is shifted to lower values. Consequently, a subsequent step could consist of establishing a quantitative relationship between the 3-point sum and the 25-point sum. Finally, this would guide an adjustment of the ADC binning scheme as discussed in subsection 9.2.8, followed by a retraining of the model using both an artificially generated 3-point sum out of the 25-point sum and the original 3-point sum for validation purposes. Owing to the 4 bit binning of the ADC values, the resulting impact is expected to remain modest, provided that an appropriately optimized binning strategy is employed.

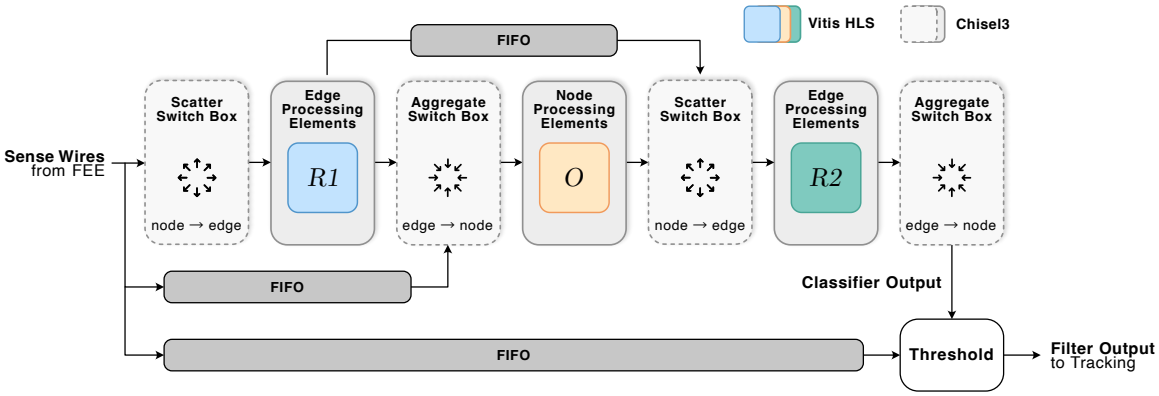


Figure 9.23.: Block diagram of the hardware-accelerated interaction network architecture [4], in which Vitis HLS-synthesized network modules are instantiated as dedicated processing element (PE). Static graphs, constructed from FEE-provided sense-wire data, are propagated and iteratively updated through a sequence of scatter and aggregate switch boxes interposed between MLP processing elements (PEs), which are implemented using Chisel and compiled into a register-transfer level (RTL) design. The final classifier outputs, following the application of configurable thresholds, are forwarded to downstream tracking modules. PEs are depicted in gray and interconnected via first-in-first-out buffers (FIFOs) or simple shift-register structures. All data interfaces conform to the AXI4-Stream [104] specification and are fully decoupled using a ready-valid handshake protocol.

9.6. Online GNN implementation on hardware

For deployment within the detector on FPGA, the algorithm must first be mapped onto FPGA architecture for each of the 20 proposed CDCs sectors, which will be detailed in this section. The work presented here was carried out in close cooperation with members of the Institut für Technik der Informationsverarbeitung (ITIV). Consequently, the implementation is described as a joint effort as it was not conducted exclusively by me.

9.6.1. Hardware implementation methodology

The interaction network introduced in section 5.4 is implemented as a data-flow accelerator on FPGA, as illustrated in Figure 9.23, using a deployment methodology that semi-automatically transforms the network into an register-transfer level (RTL) design. In this methodology, each layer of the neural network is mapped onto a dedicated processing element (PE), with three distinct types of PEs: (1) scatter switch boxes, (2) aggregate switch boxes, and (3) neural network MLP PEs. The switch boxes embed the graph data structure into the data-flow accelerator following the approach described in [97], and are realised as hardware generators written in the Chisel hardware construction language [105]. The network MLP PEs $R1$, $R2$, and O encapsulate the network weights and biases and are implemented using AMD Vitis HLS [106] together with architecture templates from the low-latency high-level synthesis (HLS) library [107]. Analogously to hls4ml [108], a reuse

Table 9.5.: Post-synthesis resource utilization for the out-of-context implementation on the AMD Alveo V80 evaluation board is reported for all 20 CDC sectors. The first six super-layers (SLs) each contain two sectors, while the outer-most SLs each comprise three sectors. By applying a reuse factor of four, the effective number of instantiated nodes and edges is reduced by a factor of four. The resulting utilization scaling linearly with *MBOPs*, in terms of FFs and LUTs, is reported both in absolute values and as percentages relative to the total available device resources. The values are provided by Marc Neu (ITIV).

						programmable logic			
						FF		LUT	
sectors	SL	N_{nodes}	N_{edges}	MBOPs	Reuse	abs.	%	abs.	%
0, 1	0	400	1745	0.71	4	666 711	12.95	755 118	29.33
2, 3	1	400	1745	0.71	4	681 023	13.23	766 867	29.79
4, 5	2	480	2097	0.86	4	821 501	15.96	915 826	35.58
6, 7	3	560	2449	1.00	4	980 551	19.05	1 082 400	42.05
8, 9	4	640	2801	1.14	4	1 091 548	21.10	1 231 879	47.85
10, 11	5	720	3153	1.29	4	1 248 055	24.24	1 392 989	54.11
12, 13	6	800	3505	1.43	4	1 401 679	27.23	1 565 184	60.80
14, 15, 16	7	640	2801	1.14	4	1 048 195	20.36	1 124 416	43.64
17, 18, 19	8	640	2801	1.14	4	1 155 272	22.44	1 261 476	49.00

factor $R \in \{2^i : i \in \mathbb{N}^+\}$ is defined, which specifies the degree of spatial parallelism of the data-flow accelerator. All data interfaces conform to the AXI4-Stream [104] protocol and are decoupled via a ready-valid handshake mechanism.

9.6.2. Resource utilization

To assess the feasibility of the proposed implementation, following the methodology of [97] and [4], the data-flow accelerator is synthesized out-of-context for an Alveo V80 evaluation board using AMD Vivado 2024.2 [109]. During synthesis, the target frequency is set to $f_{GNN} = 128.008\,\text{MHz}$, marginally above the L1 trigger system clock and corresponding to a clock period of 3.906 ns. For design validation, cycle-accurate verification is conducted using CoCoTb 1.9.2 [110] in combination with ModelSim 2023.4 [111]. In addition, functional tests are carried out to confirm the algorithmic correctness of the module.

The synthesized resource utilization for each of the proposed CDC sectors is summarized in Table 9.5. For each sector, the corresponding numbers of nodes and edges are defined by one half or one third of the number of wires on the TRG layers, respectively. The current study was conducted without accounting for an overlap between sectors. The anticipated effect of adding such an overlap of *e.g.* to azimuthal layers of wires per side with an additional 20 nodes and 88 edges is below 5 %.

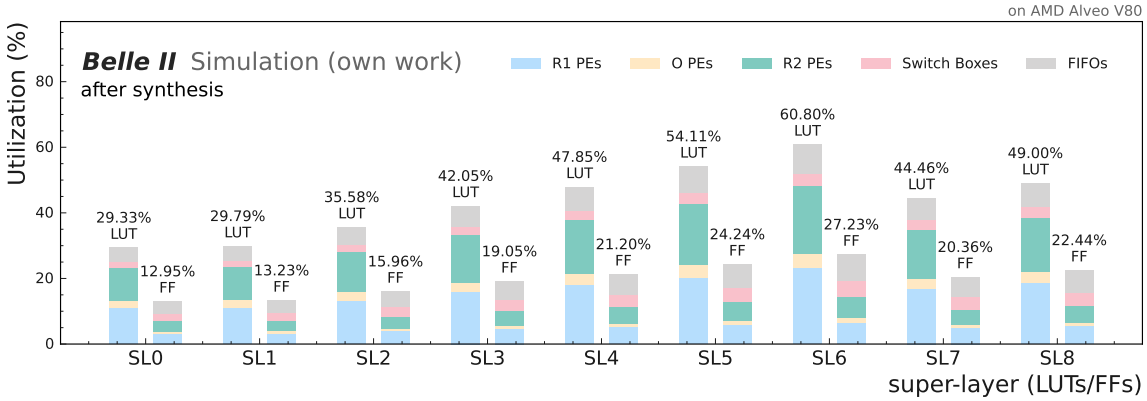


Figure 9.24.: FPGA resource utilization per GNN logic block for the different super-layers, showing modest LUT/FF usage. The results are reported from Vivado 2024.2 after synthesis in out-of-context mode.

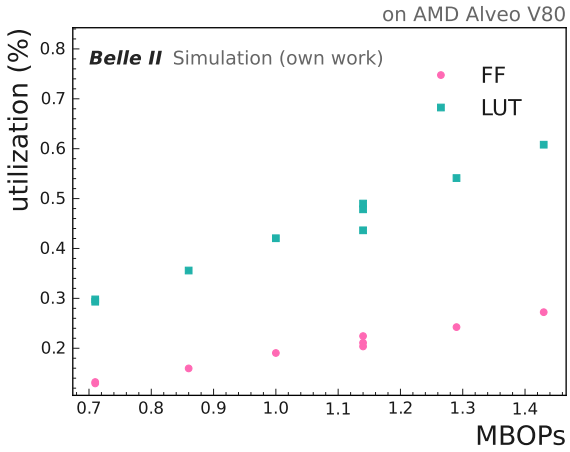


Figure 9.25.: FPGA resource utilization for FFs and LUTs scale linearly with the number of bit operations (MBOPs).

The highest utilization is observed for LUTs, reaching 60.80 % for the largest input graph in super-layer 6, compared to 29.33 % for the smallest graph. For FFs, the maximum and minimum utilization are 27.23 % and 12.95 %, respectively. No DSPs are used in the current design. Overall, the results indicate that the resource utilization scales approximately linearly with the number of edges in the input graph.
The resource utilization of each super-layer (SL), decomposed into the core accelerator components, including the edge block ($R1$, $R2$) and the node block (O) PEs, the switch boxes, and the FIFOs (buffers and queues), is presented in Figure 9.24. The dominant contribution arises from the relational network blocks $R1$ and $R2$, which together account, for example, in the largest sector (SL6), for 44.01 % of the total LUT usage and 12.85 % of the total FF usage. All figures refer to post-synthesis results and do not include possible changes induced by the placement and routing steps. Consequently, minor deviations are expected in the final implementation.

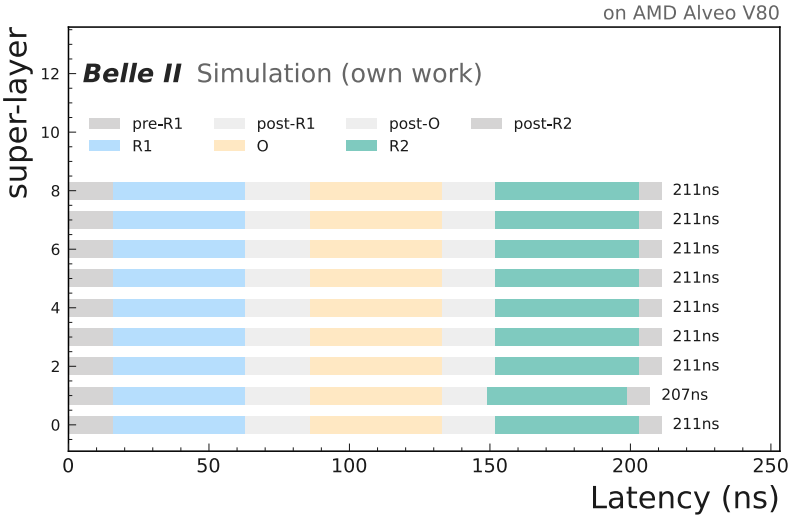


Figure 9.26.: FPGA latency per GNN MLP for different super-layers. The total pipeline latency amounts to 207 to 211 ns.

Figure 9.25 depicts the relationship between the BOP-based model size proxy employed throughout this thesis and the actual resource utilization of the FPGA implementation in terms of FFs and LUTs. The results indicate a clear linear correlation between the resource utilization and the BOPs metric. The deviation observed around 1.15 MBOPs is most likely attributable to inaccurate assumptions regarding the numbers of nodes and edges used in the implementation study, in comparison to the corresponding numbers reported in this thesis. These correlations strongly support the hypothesis that the BOPs metric can be used as a proxy for hardware utilization during a software-hardware co-design.

9.6.3. Latency and throughput analysis

The design satisfies all timing constraints and achieves a sustained throughput of 32 million events per second. The corresponding processing latencies are summarized in Figure 9.26 reporting the latency for each of the super-layer-specific implementations. For all super-layer sectors except super-layer 1, the total end-to-end latency is 210.92 ns, which corresponds to 27 clock cycles at f_{GNN}. The pipeline latency per main network logic block (R_1, R_2, and O) lies in the range 46.87 to 54.68 ns. Consequently, approximately 70.3 % of the total latency is attributable to the network blocks R_1, R_2, and O, while the remainder arises from data transmission, as well as pre- and post-processing of the data flow. Since the latency is significantly below the required $\mathcal{O}(500\,\text{ns})$, the implementation of the data-flow accelerator on an FPGA is deemed feasible on the AMD Alveo V80 platform. With a reuse factor of $R = 4$, the design reaches the required operating frequency of 31.804 MHz, thereby meeting the performance specification.

However, given that the design must ultimately operate on the UT4 board within the detector, further dedicated investigations are necessary. Initial studies indicate that, on the UT4, the limiting occupancy of LUTs reaches a maximum value of 81.06 % for the test

sector (SL2) at a pipeline latency from end-to-end of 417 ns. Although this utilization level is high, it is still regarded as acceptable for the intended test application.

9.7. Summary

In this chapter, I adapted, the offline GNN-based hit-filtering algorithm for deployment at the TRG level. For the CDC L1 trigger, the GNN-based hit filtering is introduced as an early pre-processing stage, positioned between the FEE readout and the TSF, such that all downstream tracking algorithms operate exclusively on cleaned hits. The offline-optimized model from chapter 8 is transformed to comply with the stringent real-time and hardware constraints of an FPGA-based implementation, including a latency budget of $\mathcal{O}(500\,\text{ns})$, support for processing of 14 336 wires at 31.8 MHz, and parallelization across 20 CDC sectors, while relying on reduced-resolution trigger inputs such as low-precision ADC measurements.
A compression workflow is presented that starts from the offline reference model and gradually introduces trigger-specific adaptations in hit feature selection, model architecture, arithmetic precision, and weight pruning. Throughout this workflow, the performance is evaluated using hit-level ROC curves and AUC values on the HLT-selected `mumuskim` data set, while BOPs are employed as a proxy for hardware resource usage. Overall, the final design reduces the computational cost from 1 100 to 1.40 MBOPs while simultaneously improving the hit-classification performance, as quantified by the AUC score, from 0.9374 to 0.9533. This corresponds to an enhancement in background-hit rejection at a working point of 90 % signal hit efficiency from 87.99 to 94.15 % relative to the full-precision offline model.

At the individual-hit level, the compressed GNN-based filter achieves a substantial suppression of background hits and background TSs, while maintaining a high efficiency for signal hits and TSs. For `mumuskim` events from a recent data-taking period (exp. 37, run 1893), the hit-level background rejection after application of the GNN-based hit filter of 94.64 % results in a TS efficiency of 84.38 %, in contrast to only 75.04 % obtained with the default basf2 TSF. In addition, for events without signal hits, the hit-level background rejection of 84.20 % yields a TS-level background rejection of 91.22 %, surpassing the default basf2 performance of 84.36 %.
The improved TS filtering performance directly results in improved trigger-level track reconstruction performance. For experiment 37, the track fitting efficiency evaluated on `mumuskim` samples is 50.26 % when employing the three-dimensional Hough-transform-based track finder (`3DHough`) followed by the neural network track fitter, and increases to 67.86 % when additionally incorporating the GNN-based hit filter. Concurrently, the track fitting fake and clone rates remain at a comparable level, while the z_0 resolution exhibits a modest improvement, which can be attributed to the increased hit reconstruction efficiency.
As a consequence, the STT trigger rate for `mumuskim` events is improved as well, with the efficiency rising from 64.50 % when using the `3DHough` algorithm alone to 76.09 % when the GNN filter is included. The inclusive CDC trigger rate, accounting for all track-based trigger bits, improves correspondingly from 64.51 % (`3DHough` only) to 76.27 % (with GNN). For `background-only` events, where a minimal trigger rate is desired, the dominant STT contribution amounts to 6.00 kHz in the `3DHough`-only configuration and increases moderately to 6.60 kHz upon integration of the GNN filter. Considering all L1 trigger

bits, the total L1 trigger rate is 13.20 kHz for the `3DHough`-only setup and 16.00 kHz when including the GNN, which remains below the overall trigger-rate limit of 30 kHz.

Additional FPGA implementation studies evaluated for the AMD Alveo V80 platform indicate that the design corresponding to the largest CDC sector, consisting of 800 graph nodes and 3505 edges, utilizes 60.80 % of the available LUTs and 27.23 % of the FFs, requires no DSPs, and achieves a total pipeline latency of 210.92 ns at a system clock frequency of 128.008 MHz. Consequently, the implementation satisfies both the resource utilization constraints and the sub-microsecond latency requirements. In addition, the FPGA implementation study demonstrates that the BOP metric exhibits a linear correlation with resource utilization. Consequently, this metric can be employed as a practical and quantitative instrument in software–hardware co-design workflows.

10. Outlook

Although the GNN-based hit filter developed in this work exhibits improved performance for both offline and L1 trigger applications, several ideas for further improvement remain.
For the offline application, a separate model currently has to be trained for each background scenario. While it has been demonstrated that the model exhibits reasonably good generalization properties, it would be beneficial to investigate a single model trained on a suitably diverse mixture of background conditions, which could then be deployed across a wide range of background levels.
Alternative GNN architectures, such as graph convolutional networks (GCNs) [112], may be good alternative model architectures in the offline context. In this regime, larger and more complex network architectures are acceptable, as inference latency is considerably less critical than for online trigger applications.
Up to this point, track reconstruction has been evaluated exclusively using CDC information in combination with the GNN hit filter. A systematic study is required to assess the performance once SVD and PXD information are incorporated into the final track reconstruction chain.
For the L1 trigger application, the algorithm must first be validated on dedicated evaluation hardware, rather than exclusively in simulation. In a subsequent step, it could be integrated parasitically into the detector for a test sector, where it would process data in real-time without contributing to the actual L1 trigger decision, analogous to the methodology employed in previous studies [55].
In parallel, improved model compression techniques are under investigation, including pruning strategies based on singular value decomposition [113] and knowledge distillation [114].
At present, only global weight quantization is applied for model quantization. In principle, `Brevitas` also provides differentiable quantization, which would allow individual optimization of the quantization parameters for each weight, potentially improving performance. This would require the definition of an appropriate loss function that simultaneously optimizes classification accuracy and the number of bit operations.
A more radical strategy would be to substitute the Brevitas-based quantization scheme with either high granularity quantization (HGQ) [115], which combines network quantization and pruning, or NeuraLUT [116], which offers substantially more compact model representations and therefore appears particularly promising for deployment on FPGAs.

11. Conclusion

In this thesis, I present the development, implementation and evaluation of a hit-filtering algorithm based on graph neural networks (GNNs) for the Belle II central drift chamber (CDC), targeting both offline track reconstruction and Level-1 (L1) trigger application.

For the offline application, I designed, optimized, and integrated the GNN-based hit filter into the standard Belle II Analysis Software Framework (basf2) reconstruction chain using an `onnx` representation of the model allowing for drop-in replacement for the existing cut-based and multi-variate analysis (MVA) filters. Across simulated $\mu^+\mu^-(\gamma)$, $B^0\bar{B}^0$ and displaced-decay samples, the GNN-based filter consistently improves hit- and track-level performance compared to the existing filters.
The largest gains are observed for the highest-background conditions. For example, at the hit level, the GNN reduces the number of extra CDC hits (used as a measure for the background) by 96.4 %, compared to 39.3 % obtained by the cut-based filter and 78.6 % by the MVA filter. Simultaneously, the per-track hit efficiency increases from 79 % (cut-based) and 81 % (MVA) to nearly 87 % by the GNN filter.
The improved hit filtering results in an increased track fitting efficiency, with relative gains ranging from 0.3 % in the low-background scenario evaluated on the $B^0\bar{B}^0$ sample, up to 5.7 % and 27.3 % in the high-background scenario evaluated on the $\mu^+\mu^-(\gamma)$ sample compared to the MVA and cut-based filter, respectively. Specifically, for $B^0\bar{B}^0$ events, the track fitting efficiency increases from 62 % (cut-based) and 71 % (MVA) to 74 % (GNN) and for $\mu^+\mu^-(\gamma)$ events from 73 % (cut-based) and 88 % (MVA) to 93 % (GNN). The fake and clone rates remain comparable, while the resolutions of p_T and z_0 exhibit a modest decrease of up to 24.5 % in the $B^0\bar{B}^0$, high-background configuration relative to the MVA baseline. Applied to displaced $K^0_S \to \pi^+\pi^-$ events in the high background scenario, the GNN improves the track fitting efficiency from 52 % (cut-based) and 65 % (MVA) to 70 % (GNN). The largest gains are observed in the backward end-cap. In this region, the $\mu^+\mu^-(\gamma)$ efficiency increases from about 21 % (cut-based) and 55 % (MVA) to 75 % (GNN), and the $B^0\bar{B}^0$ efficiency from about 22 % (cut-based) and 43 % (MVA) to 52 % (GNN). These correspond to relative improvements of up to 252 % (cut-based) and 36.4 % (MVA) in the $\mu^+\mu^-(\gamma)$ sample, and up to 134 % (cut-based) and 18.4 % (MVA) in the $B^0\bar{B}^0$ sample.
In terms of computational performance, the GNN filter preserves the overall tracking runtime, reduces track finding time by down to a factor of 0.66, and increases track fitting time by at most a factor of 1.13, consistent with its higher track finding and per-track hit efficiency.
The improved track reconstruction efficiency effectively increases physics yield and sensitivity to rare decays. For example, looking at the 2024 running period with 128 effective data taking days, the 4.2 % relative gain in track fitting efficiency for $B^0\bar{B}^0$ events corresponds hypothetically to four days of data taking for the same effective integrated

luminosity. A 8.1 % gain in the displaced pion reconstruction efficiency from $K_S^0 \to \pi^+\pi^-$ corresponds similarly to approximately a week of effective Belle II running time for analyzes dominated by these decay modes.

To put this into perspective, in 2024, SuperKEKB consumed approximately 300 GW h of electricity, corresponding to 0.14 Mt of CO_2 emissions [117], comparable to the annual emissions of a small Japanese town of 13 000-15 000 inhabitants. Under these conditions, hypothetical shorter running times for the same integrated luminosity would effectively save about 16 GW h of electricity and 7.5 kt of CO_2, equivalent to more than 6 000 round trip flights between Frankfurt and Tokyo, assuming 1 200 kg of CO_2 per passenger per round trip.

In addition to its offline track reconstruction application, I further adapted the offline-optimized GNN-based hit filter for deployment in the CDC L1 trigger as an early preprocessing stage between the front-end electronic (FEE) readout and the track segment finder (TSF). In this configuration, all subsequent trigger-level tracking algorithms operate exclusively on filtered hits and track segments (TSs).

A principal conceptual advantage of this hit-filtering approach is that the algorithm operates on local patterns rather than on complete or large parts of event displays, as required for track reconstruction algorithms, whose processing time typically scales with the square of the number of hits. In contrast, the proposed algorithm can be directly parallelized across multiple super-layer-specific boards, thereby enabling efficient processing of the large hit multiplicities arising from increasing beam-induced backgrounds. Following this hit-filtering stage, the detector occupancy is substantially reduced, which in turn permits the application of more sophisticated track-finding algorithms. In principle, the edge-level output generated by the hit-filtering algorithm could also be utilized for the construction of tracklets, serving as a preliminary track-finding stage.

I developed a compression workflow to meet the constraints of the L1 trigger and Field-Programmable Gate Array (FPGA) implementation within a total latency budget of $\mathcal{O}(500\,\text{ns})$, processing 14 336 wires at 31.8 MHz in 20 CDC sectors, and reduced input information compared to the offline application. The workflow includes modifying hit feature inputs, adapting the model architecture, applying low-bit quantization to 4 bit precision, and weight pruning to 30 % sparsity. The final compressed model reduces the computational cost from 1 100 to 1.4 million bit operations (MBOPs) while improving the area under the curve (AUC) score for hit filtering from 0.937 to 0.953, corresponding to increased background-hit rejection at 90 % hit efficiency from 88 to 94 %.

Evaluated on high-level trigger (HLT)-selected $\mu^+\mu^-(\gamma)$ data from late 2025, the CDC trigger with integration of the compressed GNN filter achieves 94.6 % hit-level background rejection at 84.4 % TS efficiency (vs. 75.0 % for the default TSF) in trigger simulation. For `background-only` events, it improves TS-level background rejection from 84.4 to 91.2 %.

These gains result in improvements in the reconstruction of tracks at the trigger-level. For late 2025 $\mu^+\mu^-(\gamma)$ samples, the track fitting efficiency increases from 50.3 % with the default three-dimensional Hough finder plus neural-network-based fitter (`3DHough`) to 67.9 % with an additional GNN filter, with similar fake and clone rates and slightly improved z_0 resolution.

Consequently, the single track trigger (STT) and inclusive CDC trigger efficiencies for tracks in $\mu^+\mu^-(\gamma)$ events increase from 64.5 % without to 76 % with the GNN filter, while the total L1 trigger rate for `background-only` events increases from 13.2 to 16.0 kHz, therefore staying below the 30 kHz design limit.
FPGA implementation studies for the AMD Alveo V80 demonstrate that the design for the largest CDC sector with 820 nodes and 3 593 edges utilizes 60.80 % of look-up tables (LUTs), 27.23 % of flip-flops (FFs), and no digital signal processors (DSPs). Simultaneously, the implementation achieves 210.92 ns end-to-end latency at a 128.008 MHz system frequency. The design thus satisfies the resource and sub-microsecond latency requirements for integration into the Belle II L1 trigger.
In addition, the FPGA implementation study indicates that the number of bit operation (BOP) metric used throughout this work as an approximation of the model size exhibits a linear relationship with hardware resource utilization. Accordingly, this metric can be employed as a practical and quantitative tool within software-hardware co-design workflows.

In conclusion, the GNN-based hit filtering improves tracking efficiency, purity, and resolutions for key physics channels in both offline reconstruction and online triggering, while satisfying constraints on computing resources, trigger rates, and FPGA hardware utilization. In addition, this work establishes a compression workflows for the software-hardware co-design of GNNs in real-time trigger systems, including feature selection and architecture optimization, ultra-low quantization, and weight pruning, which can be applied to the development of future machine-learning-based trigger designs.

A. Appendix

A.1. Offline hit filtering

A.1.1. GNN optimization evaluation fluctuations

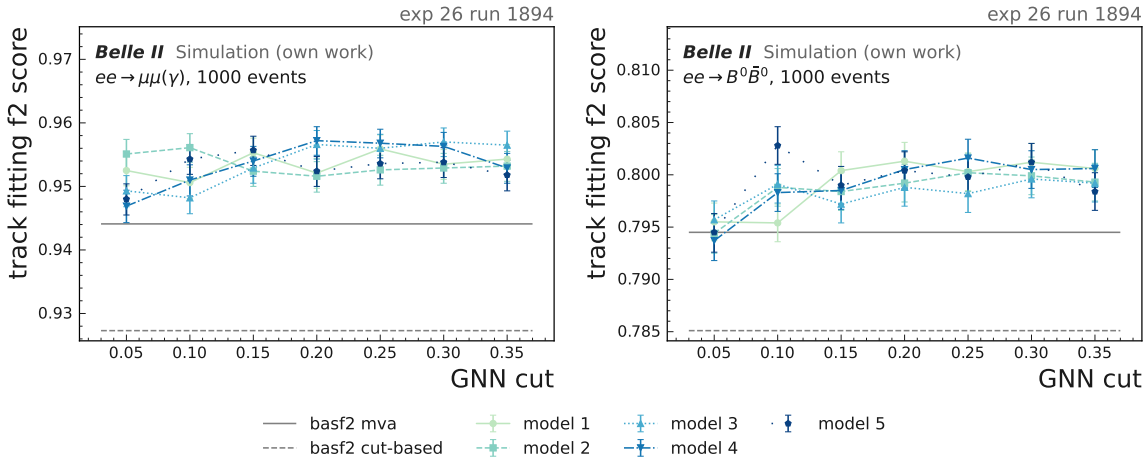


Figure A.1.: Track fitting f_2 score as a function of the GNN cut for five independent trainings of the same configuration, illustrating model training fluctuations of 0.2 to 0.5 %pt due to limited statistics and training stochastics. The hit filtering is based on the final, optimized configuration and evaluated on the common benchmark comprising 1 000 events each for $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ samples for a background corresponding to experiment 26, run 1894.

In the optimization study presented in section 8.1, a small sample size of 1 000 events per configuration are employed. The rationale for employing a relatively small sample size of 1 000 events is to prioritize the broad exploration of the extensive GNN hyper-parameter space over achieving high-precision estimates for a limited subset of parameters. Consequently, considerable fluctuations in the f_2 score are observed between different training runs, as illustrated in Figure A.1. The figure reports the f_2 score for five independent trainings performed under identical configurations.
For the $B\bar{B}$ samples, which constitute the primary basis for decision-making, the observed fluctuations between trainings of approximately 0.2 %pt are significantly smaller than for the $\mu^+\mu^-(\gamma)$ samples with variations of approximately 0.5 %pt. The error bars indicate that the statistical fluctuations arising from the limited sample size are of the same order of magnitude as the variations observed between different model instances. The influence of

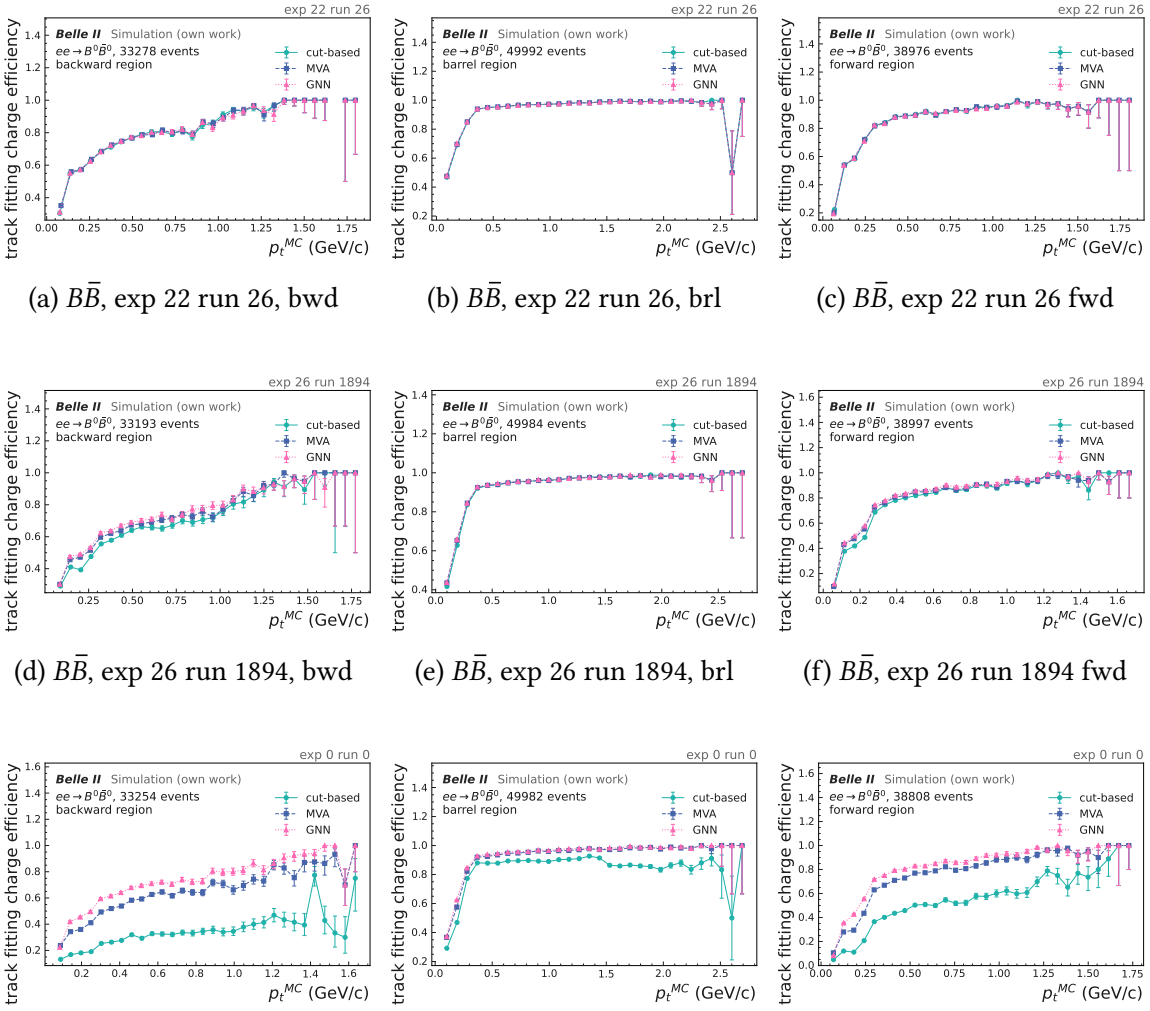
the prior could, in principle, be mitigated by employing a larger test sample. The run-to-run variations are an expected consequence of stochastic effects in the training procedure: relatively small changes in the random initialization of the GNN can induce measurable shifts in the performance curves as a function of the GNN score threshold.
Although this sample size is not optimal for high-precision performance measurements, it represents a practical compromise between statistical robustness and the computational cost associated with evaluating a large number of network configurations. Since these studies are used exclusively to guide architectural and hyper-parameter choices, and not for the final performance assessment discussed in section 8.2, I consider the residual run-to-run fluctuations acceptable.

A.1.2. Charge efficiency for different detector regions

Table A.1.: Track fitting charge efficiency per detector region for the cut-based, MVA, and GNN filtering approaches for all three background configurations and both $\mu^+\mu^-(\gamma)$ and $B^0\bar{B}^0$ samples. The differences with respect to the best GNN model are indicated in green (improvement) and red (degradation).

Charge Eff.	Forward (%)		Barrel (%)		Backward (%)		Full (%)	
$\mu^+\mu^-$ (γ)								
Exp. 22 Run 26								
cut-based	$97.51^{+0.13}_{-0.14}$	+0.40	$99.86^{+0.02}_{-0.02}$	−0.04	$82.81^{+0.34}_{-0.34}$	−0.33	$96.76^{+0.06}_{-0.06}$	
mva	$97.37^{+0.13}_{-0.14}$	+0.54	$99.81^{+0.02}_{-0.02}$	+0.01	$81.14^{+0.35}_{-0.35}$	+1.34	$96.44^{+0.07}_{-0.07}$	+0.32
best GNN model	$97.91^{+0.12}_{-0.13}$		$99.82^{+0.02}_{-0.02}$		$82.48^{+0.34}_{-0.34}$		$96.76^{+0.06}_{-0.06}$	
Exp. 26 Run 1894								
cut-based	$83.68^{+0.32}_{-0.32}$	+8.86	$99.41^{+0.03}_{-0.03}$	−0.14	$60.78^{+0.44}_{-0.44}$	+15.49	$90.63^{+0.10}_{-0.10}$	+3.87
mva	$88.01^{+0.28}_{-0.28}$	+4.53	$99.24^{+0.04}_{-0.04}$	+0.03	$68.07^{+0.42}_{-0.42}$	+8.20	$92.40^{+0.09}_{-0.10}$	+2.10
best GNN model	$92.54^{+0.22}_{-0.23}$		$99.27^{+0.04}_{-0.04}$		$76.27^{+0.38}_{-0.38}$		$94.50^{+0.08}_{-0.08}$	
Exp. 0 Run 0								
cut-based	$38.75^{+0.42}_{-0.42}$	+50.46	$95.10^{+0.09}_{-0.09}$	+4.55	$21.44^{+0.37}_{-0.37}$	+53.98	$73.89^{+0.16}_{-0.16}$	+20.16
mva	$76.69^{+0.36}_{-0.37}$	+12.52	$99.47^{+0.03}_{-0.03}$	+0.18	$55.31^{+0.45}_{-0.45}$	+20.11	$88.62^{+0.11}_{-0.11}$	+5.43
best GNN model	$89.21^{+0.27}_{-0.27}$		$99.65^{+0.02}_{-0.03}$		$75.42^{+0.38}_{-0.39}$		$94.05^{+0.08}_{-0.08}$	
$B\bar{B}$								
Exp. 22 Run 26								
cut-based	$65.74^{+0.19}_{-0.19}$	+0.28	$84.96^{+0.06}_{-0.06}$	+0.40	$62.16^{+0.23}_{-0.23}$	+0.26	$80.24^{+0.06}_{-0.06}$	+0.37
mva	$65.81^{+0.19}_{-0.19}$	+0.21	$85.12^{+0.06}_{-0.06}$	+0.24	$62.47^{+0.22}_{-0.23}$	−0.05	$80.40^{+0.06}_{-0.06}$	+0.21
best GNN model	$66.02^{+0.19}_{-0.19}$		$85.36^{+0.06}_{-0.06}$		$62.42^{+0.23}_{-0.23}$		$80.61^{+0.06}_{-0.06}$	
Exp. 26 Run 1894								
cut-based	$54.77^{+0.20}_{-0.20}$	+5.28	$81.73^{+0.06}_{-0.06}$	+1.26	$48.53^{+0.23}_{-0.23}$	+6.34	$75.00^{+0.06}_{-0.06}$	+2.27
mva	$58.90^{+0.19}_{-0.20}$	+1.15	$82.50^{+0.06}_{-0.06}$	+0.49	$53.07^{+0.23}_{-0.23}$	+1.80	$76.57^{+0.06}_{-0.06}$	+0.70
best GNN model	$60.05^{+0.19}_{-0.19}$		$82.99^{+0.06}_{-0.06}$		$54.87^{+0.23}_{-0.23}$		$77.27^{+0.06}_{-0.06}$	
Exp. 0 Run 0								
cut-based	$25.74^{+0.17}_{-0.17}$	+27.81	$73.52^{+0.07}_{-0.07}$	+8.21	$22.24^{+0.19}_{-0.19}$	+29.74	$62.36^{+0.07}_{-0.07}$	+12.85
mva	$46.14^{+0.20}_{-0.20}$	+7.41	$80.06^{+0.07}_{-0.07}$	+1.67	$43.92^{+0.23}_{-0.23}$	+8.06	$72.17^{+0.06}_{-0.06}$	+3.04
best GNN model	$53.55^{+0.20}_{-0.20}$		$81.73^{+0.06}_{-0.06}$		$51.98^{+0.23}_{-0.23}$		$75.21^{+0.06}_{-0.06}$	

A.1.2.1. Charge efficiency for different detector regions over p_T^{MC}, $B\bar{B}$



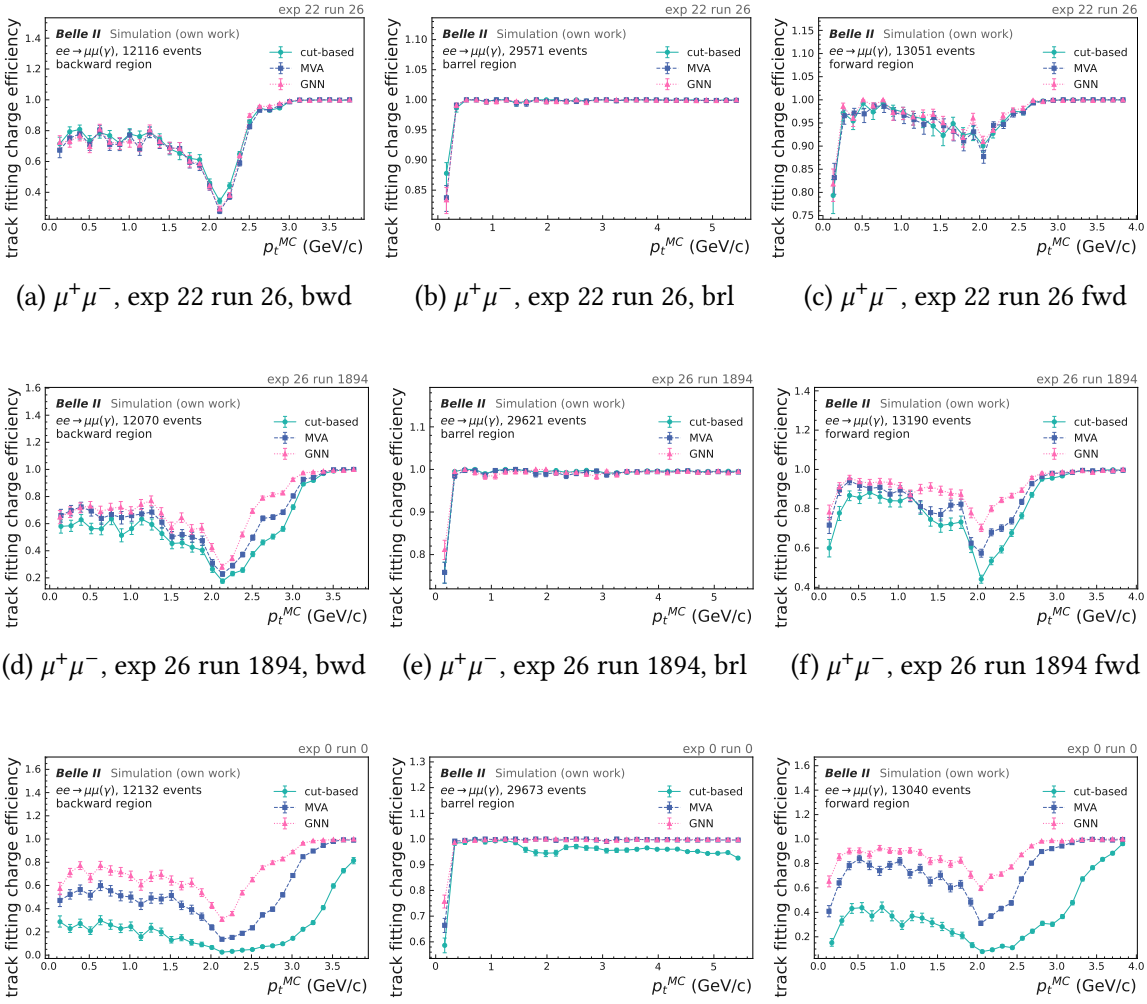
(a) $B\bar{B}$, exp 22 run 26, bwd (b) $B\bar{B}$, exp 22 run 26, brl (c) $B\bar{B}$, exp 22 run 26 fwd

(d) $B\bar{B}$, exp 26 run 1894, bwd (e) $B\bar{B}$, exp 26 run 1894, brl (f) $B\bar{B}$, exp 26 run 1894 fwd

(g) $B\bar{B}$, exp 0 run 0, bwd (h) $B\bar{B}$, exp 0 run 0, brl (i) $B\bar{B}$, exp 0 run 0 fwd

Figure A.2.: Track fitting charge efficiency over the transverse momentum p_T^{MC} for the three different detector regions forward (fwd), barrel (brl) and backward (bwd) comparing cut-based filtering (green) and MVA filtering (blue) with GNN filtering (magenta) for 50 000 $B^0\bar{B}^0$ events and three different background levels.

A.1.2.2. Charge efficiency for different detector regions over p_T^{MC}, $\mu^+\mu^-$ (γ)



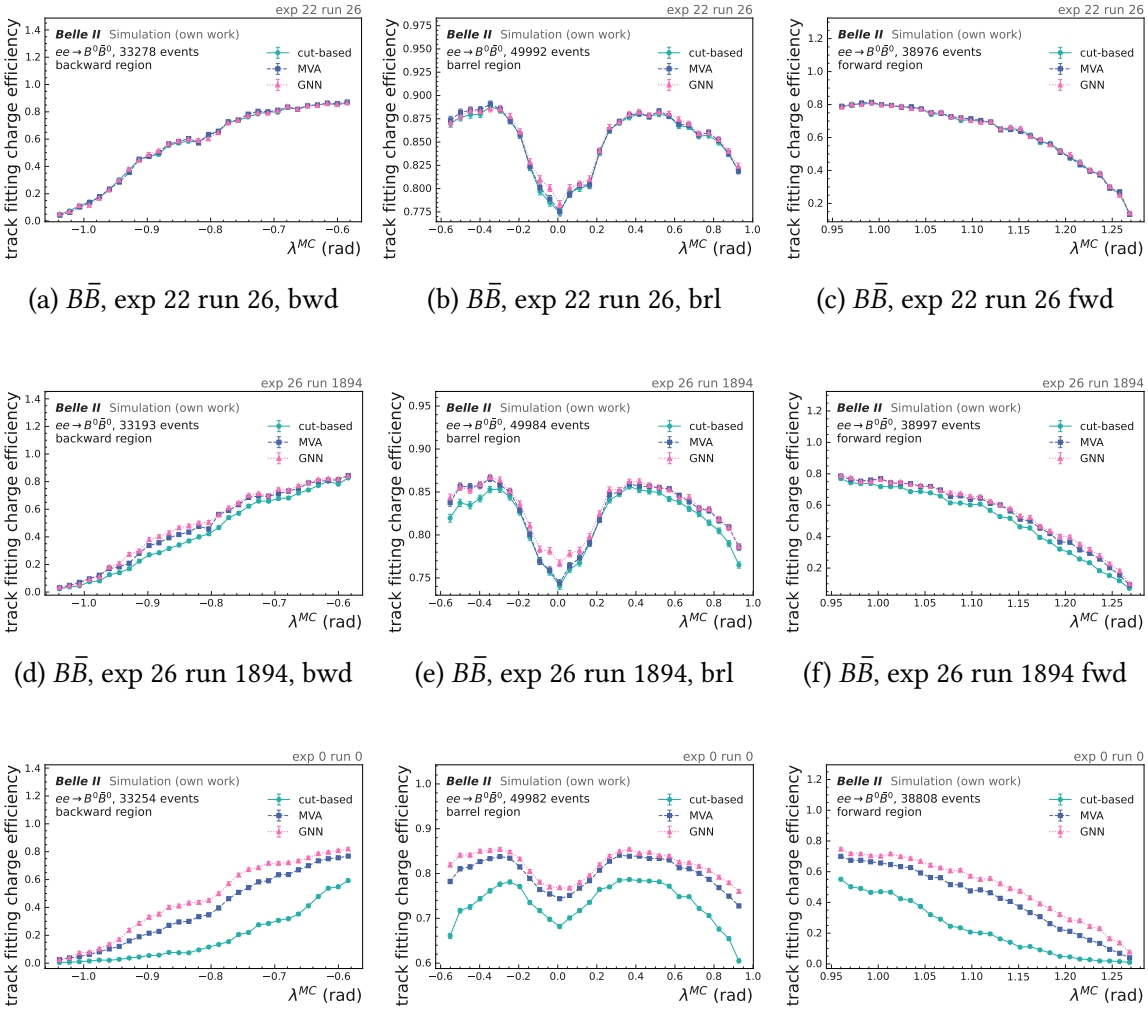
(a) $\mu^+\mu^-$, exp 22 run 26, bwd (b) $\mu^+\mu^-$, exp 22 run 26, brl (c) $\mu^+\mu^-$, exp 22 run 26 fwd

(d) $\mu^+\mu^-$, exp 26 run 1894, bwd (e) $\mu^+\mu^-$, exp 26 run 1894, brl (f) $\mu^+\mu^-$, exp 26 run 1894 fwd

(g) $\mu^+\mu^-$, exp 0 run 0, bwd (h) $\mu^+\mu^-$, exp 0 run 0, brl (i) $\mu^+\mu^-$, exp 0 run 0 fwd

Figure A.3.: Track fitting charge efficiency over the transverse momentum p_T^{MC} for the three different detector regions forward (fwd), barrel (brl) and backward (bwd) comparing cut-based filtering (green) and MVA filtering (blue) with GNN filtering (magenta) for 50 000 $\mu^+\mu^-(\gamma)$ events and three different background levels.

A.1.2.3. Charge efficiency for different detector regions over λ^{MC}, $B\bar{B}$



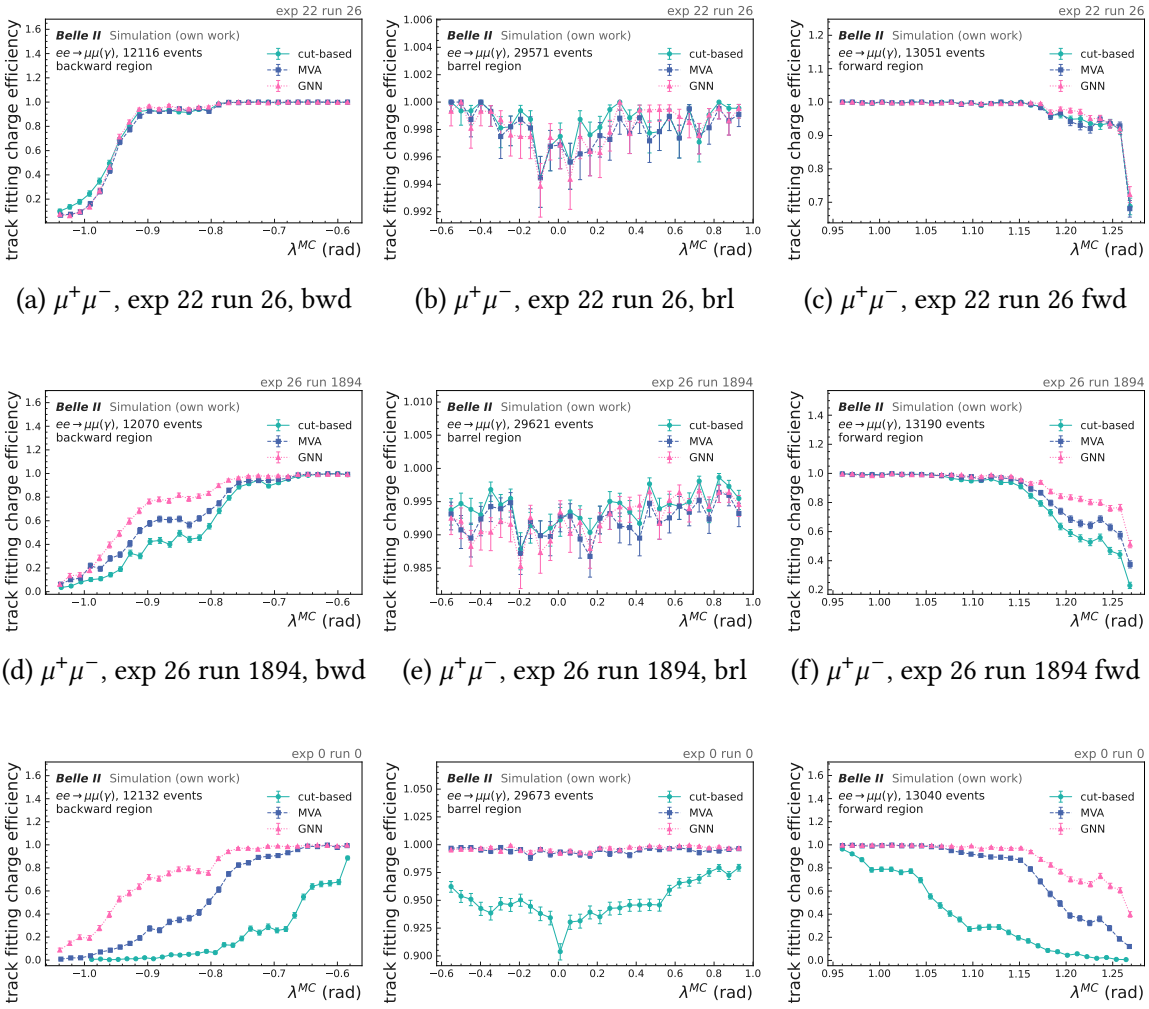
(a) $B\bar{B}$, exp 22 run 26, bwd (b) $B\bar{B}$, exp 22 run 26, brl (c) $B\bar{B}$, exp 22 run 26 fwd

(d) $B\bar{B}$, exp 26 run 1894, bwd (e) $B\bar{B}$, exp 26 run 1894, brl (f) $B\bar{B}$, exp 26 run 1894 fwd

(g) $B\bar{B}$, exp 0 run 0, bwd (h) $B\bar{B}$, exp 0 run 0, brl (i) $B\bar{B}$, exp 0 run 0 fwd

Figure A.4.: Track fitting charge efficiency over the dip angle λ^{MC} for the three different detector regions forward (fwd), barrel (brl) and backward (bwd) comparing cut-based filtering (green) and MVA filtering (blue) with GNN filtering (magenta) for 50 000 $B^0\bar{B}^0$ events and three different background levels.

A.1.2.4. Charge efficiency for different detector regions over λ^{MC}, $\mu^+\mu^-$ (γ)



(a) $\mu^+\mu^-$, exp 22 run 26, bwd (b) $\mu^+\mu^-$, exp 22 run 26, brl (c) $\mu^+\mu^-$, exp 22 run 26 fwd

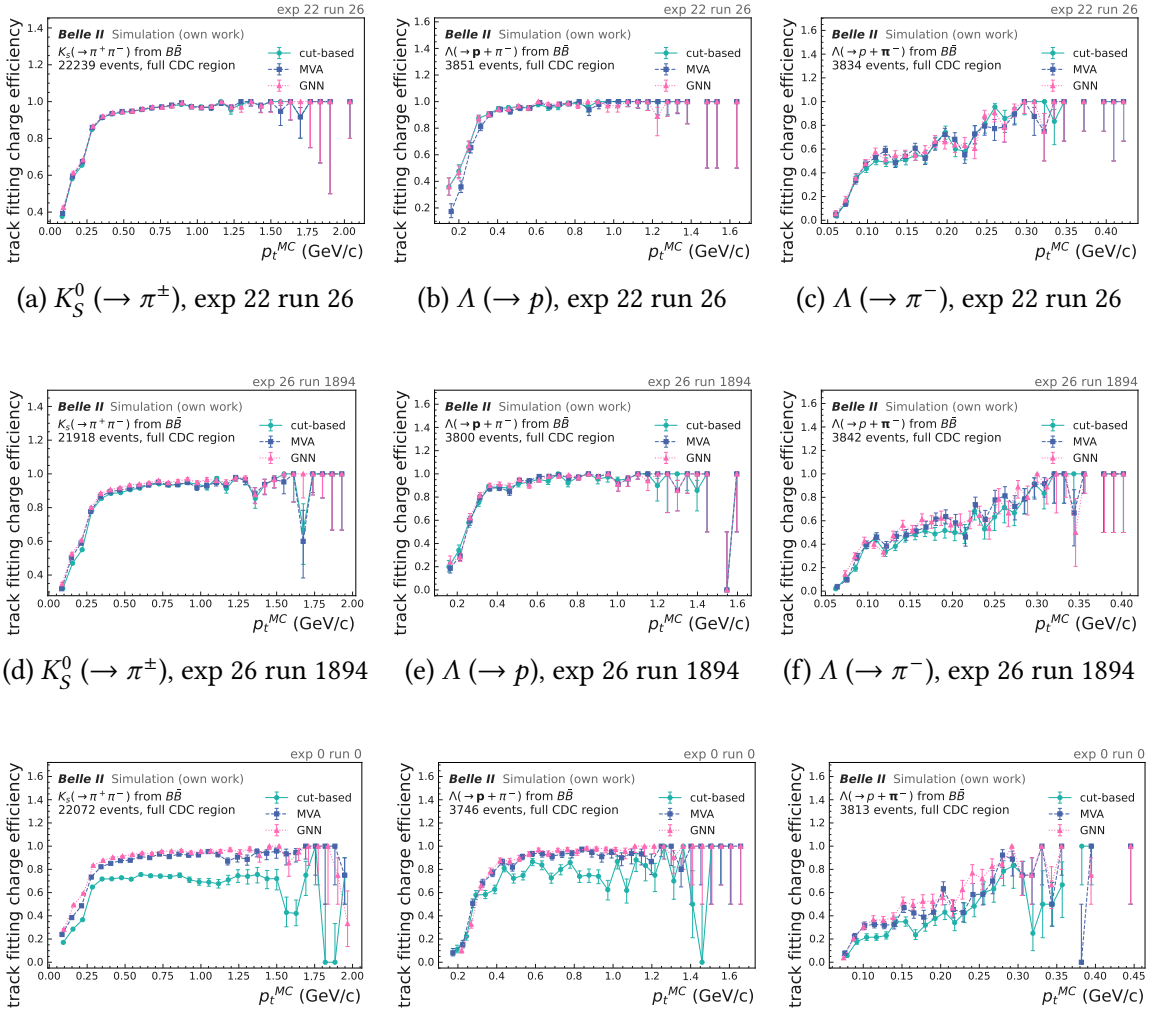
(d) $\mu^+\mu^-$, exp 26 run 1894, bwd (e) $\mu^+\mu^-$, exp 26 run 1894, brl (f) $\mu^+\mu^-$, exp 26 run 1894 fwd

(g) $\mu^+\mu^-$, exp 0 run 0, bwd (h) $\mu^+\mu^-$, exp 0 run 0, brl (i) $\mu^+\mu^-$, exp 0 run 0 fwd

Figure A.5.: Track fitting charge efficiency over the dip angle λ^{MC} for the three different detector regions forward (fwd), barrel (brl) and backward (bwd) comparing cut-based filtering (green) and MVA filtering (blue) with GNN filtering (magenta) for 50 000 $\mu^+\mu^-(\gamma)$ events and three different background levels.

A.1.3. Charge efficiencies for K_S^0 and Λ decays

A.1.3.1. Charge efficiencies for K_S^0 and Λ decays over p_T^{MC}



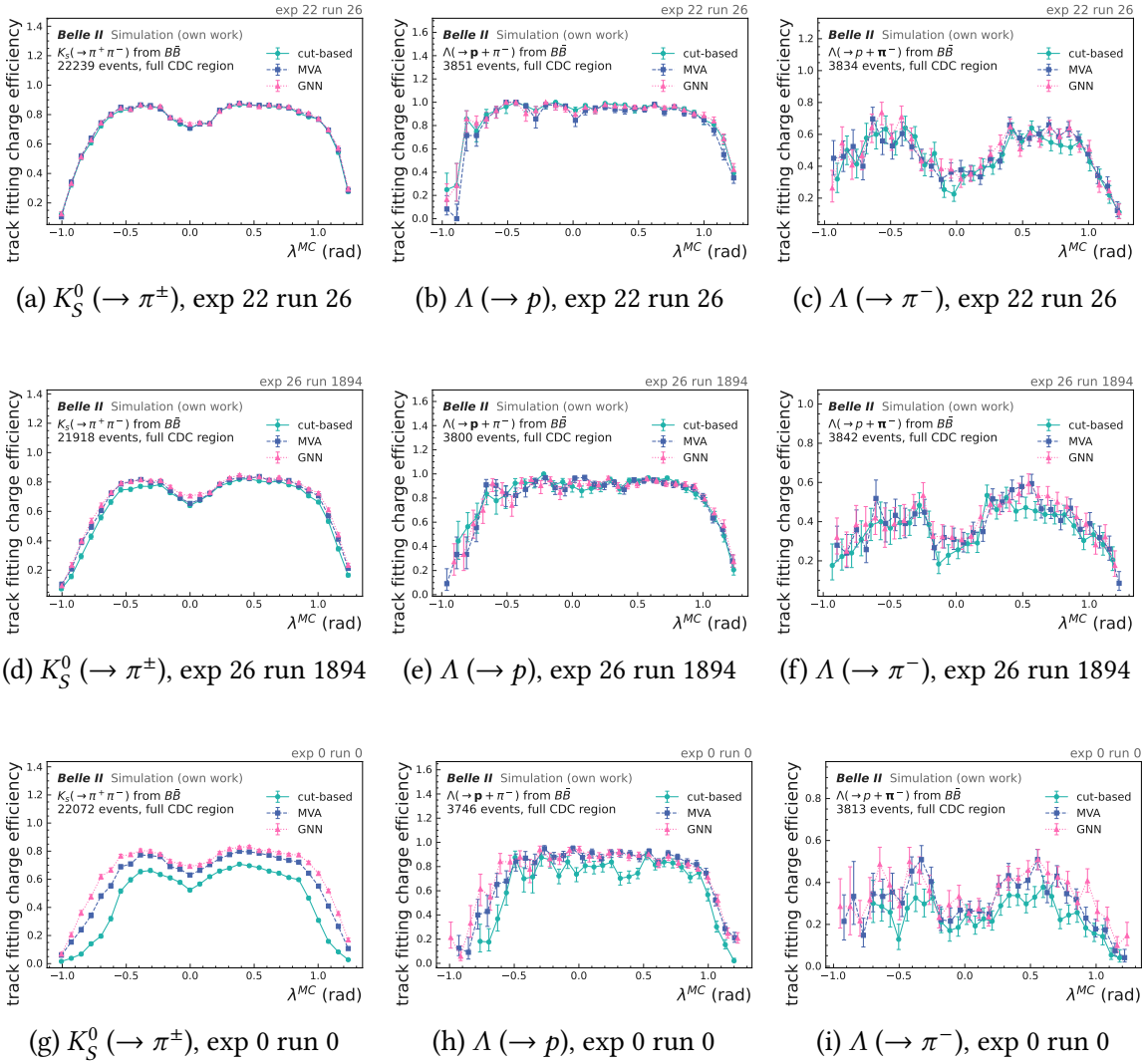
(a) K_S^0 ($\to \pi^\pm$), exp 22 run 26 (b) Λ ($\to p$), exp 22 run 26 (c) Λ ($\to \pi^-$), exp 22 run 26

(d) K_S^0 ($\to \pi^\pm$), exp 26 run 1894 (e) Λ ($\to p$), exp 26 run 1894 (f) Λ ($\to \pi^-$), exp 26 run 1894

(g) K_S^0 ($\to \pi^\pm$), exp 0 run 0 (h) Λ ($\to p$), exp 0 run 0 (i) Λ ($\to \pi^-$), exp 0 run 0

Figure A.6.: Track fitting charge efficiency over the transverse momentum p_T^{MC} for pion and proton tracks from K_S^0 and Λ decays from 50 000 $B^0\bar{B}^0$ events comparing cut-based filtering (green) and mva filtering (blue) with GNN filtering (magenta) for for three different background levels. The following selections are applied: $n_{\text{CDCHitsperTrack}} \geq 7$.

A.1.3.2. Charge efficiencies for K_S^0 and Λ decays over λ^{MC}



(a) K_S^0 ($\to \pi^\pm$), exp 22 run 26 (b) Λ ($\to p$), exp 22 run 26 (c) Λ ($\to \pi^-$), exp 22 run 26

(d) K_S^0 ($\to \pi^\pm$), exp 26 run 1894 (e) Λ ($\to p$), exp 26 run 1894 (f) Λ ($\to \pi^-$), exp 26 run 1894

(g) K_S^0 ($\to \pi^\pm$), exp 0 run 0 (h) Λ ($\to p$), exp 0 run 0 (i) Λ ($\to \pi^-$), exp 0 run 0

Figure A.7.: Track fitting charge efficiency over the transverse momentum p_T^{MC} for pion and proton tracks from K_S^0 and Λ decays from 50 000 $B^0\bar{B}^0$ events comparing cut-based filtering (green) and mva filtering (blue) with GNN filtering (magenta) for for three different background levels. The following selections are applied: $n_{\text{CDCHitsperTrack}} \geq 7$.

A.2. Online hit filtering

A.2.1. Trigger bits

Table A.2.: Full list of CDC trigger output bits, prescale factors, and conditions (index ranges indicate inclusive ORs over inputs). Each bit is accompanied by a bhabha and injection veto condition.

trigger bit(s)	prescale	condition	trigger bit(s)	prescale	condition
bf	0	$t2_{0-3}$	ff30	0	$t2_{1-3}$ & f2f30
bffo	0	$t2_{1-3}$ & $cdc_{open,90}$	ffb	0	$t2_{1-3}$ & $b2b_5$
bfyo	50	$t2_{1-3}$ & ty_{0-3} & $cdc_{open,90}$	fff	0	$t2_{2-3}$
bs	0	ts_{0-3}	fffo	0	$t2_{2-3}$ & $cdc_{open,90}$
by	0	ty_{0-3}	ffo	0	$t2_{1-3}$ & $cdc_{open,90}$
bz	0	$t3_{0-3}$	ffoc	0	$t2_{1-3}$ & $clst_{0-3}$ & $cdc_{open,90}$
f	20000	$t2_{0-3}$	ffs	0	$t2_{1-3}$ & ts_{0-3}
ff	0	$t2_{1-3}$	ffy	1	$t2_{2-3}$ & ty_{0-3}
			ffyo	0	$t2_{2-3}$ & ty_{0-3} & $cdc_{open,90}$
ffz	0	$t2_{2-3}$ & $t3_{0-3}$	fy	0	$t2_{1-3}$ & ty_{0-3}
fs	0	$t2_{0-3}$ & ts_{0-3}	fy30	1	$t2_{1-3}$ & ty_{0-3} & f2f30
fs30	0	$t2_{0-3}$ & ts_{0-3} & s2f30	fyb	1	$t2_{1-3}$ & ty_{0-3} & $b2b_5$
fsb	0	$t2_{0-3}$ & ts_{0-3} & s2f5	fyo	1	$t2_{1-3}$ & ty_{0-3} & $cdc_{open,90}$
fso	0	$t2_{0-3}$ & ts_{0-3} & s2fo	fyy	0	$t2_{2-3}$ & ty_{1-3}
fz	0	$t2_{1-3}$ & $t3_{0-3}$	fzb	0	$t2_{1-3}$ & $t3_{0-3}$ & $b2b_5$
fzo	0	$t2_{1-3}$ & $t3_{0-3}$ & $cdc_{open,90}$	s	4000	ts_{0-3}
fzz	0	$t2_{2-3}$ & $t3_{1-3}$	ss	0	ts_{1-3}
ss30	0	ts_{1-3} & s2s30	ssb	10	ts_{1-3} & s2s5
sso	0	ts_{1-3} & s2so	sss	0	ts_{2-3}
stt	1	typ	stt4	0	typ4
stt5	0	typ5	stt6	1000	typ6
syb	1	ts_{0-3} & ty_{0-3} & s2f5	syo	1	ts_{0-3} & ty_{0-3} & s2fo
yy	0	ty_{1-3}	yyv	0	ty_{1-3}
yyy	0	ty_{2-3}	z	0	$t3_{0-3}$
zz	0	$t3_{1-3}$	zzz	0	$t3_{2-3}$

A.2.2. Trigger rate calculation

In the following, I describe the methodology for determining the individual contributions to the total trigger rate, along with the rate extrapolation to future conditions at SuperKEKB, following the approach outlined in [79].
The total trigger rate, R_{total}, measured at the final output stage of the trigger system, can be decomposed into a contribution from physics processes, R_{phys}, and several beam-related background components as

$$R_{\text{total}} = R_{\text{phys}} + R_{\text{Touschek}} + R_{\text{beam-gas}} + R_{\text{bhabha}} + R_{2\gamma} + R_{\text{cosmic}}, \tag{A.1}$$

where the dominant beam background contributions arise from Touschek scattering, beam-gas interactions, and two-photon processes.

A.2.2.1. Single beam background

The single beam contributions for the low energy ring (LER) and the high energy ring (HER), are composed of

$$R_{\text{beam}} = R_{\text{Touschek}} + R_{\text{beam-gas}} + R_{\text{cosmic}}, \tag{A.2}$$

where each background source exhibits a characteristic dependence on the machine parameters. Touschek scattering, arising from intra-bunch Coulomb scattering, produces a rate proportional to the single-beam current density and thus scales linearly with the single-beam current I and inversely with the horizontal, vertical, and bunch length $\sigma_{x,y,z}$:

$$R_{\text{Touschek}} \propto \frac{I^2}{n_b\, \sigma_z\, \sigma_x\, \sigma_y}, \tag{A.3}$$

where n_b is the number of bunches. Beam-gas interactions, caused by inelastic scattering of beam particles off residual gas molecules in the vacuum pipe, scale linearly with the single-beam current I and the ring average effective residual gas pressure seen by the beam P_{eff}:

$$R_{\text{beam-gas}} \propto I P_{\text{eff}}. \tag{A.4}$$

The overall single-beam background can than be described as

$$R_{\text{beam}} = a_B \times I P_{\text{eff}} + a_T \times \frac{I^2}{n_b\, \sigma_z\, \sigma_x\, \sigma_y} + R_{\text{cosmic}} \tag{A.5}$$

where a_B and a_T are free parameters that will be obtained from fitting the distribution to a linear function after subtracting the constant cosmic ray trigger rate.

A.2.2.2. Luminosity background

Physics trigger rate, two-photon processes and radiative Bhabha events are proportional to the instantaneous luminosity $\mathcal{L}$ and combined into the luminosity dependent trigger rate contribution:

$$R_{\text{lumi}} = R_{\text{phys}} + R_{\text{bhabha}} + R_{2\gamma} \propto \mathcal{L}. \tag{A.6}$$

A.2.2.3. Separation of background contributions

To disentangle the individual contributions, dedicated single-beam and collision datasets are exploited. The method relies on measuring the trigger rate under three distinct machine conditions:

1. **LER single-beam**: only the LER is filled; $R_{\text{beam,LER}}$ is dominated by LER Touschek and LER beam-gas backgrounds.
2. **HER single-beam**: only the HER is filled; R_{HER} is dominated by HER Touschek and HER beam-gas backgrounds.
3. **Collision**: both beams are colliding; R_{coll} includes all contributions plus luminosity-dependent processes.

The luminosity-dependent contribution can then be isolated as

$$R_{\text{lumi}} = R_{\text{total}} - R_{\text{beam,LER}} - R_{\text{beam,HER}}, \tag{A.7}$$

where it is assumed that interference terms between LER and HER backgrounds are negligible.

A.2.2.4. Rate extrapolation

To project the trigger rates to the design luminosity of SuperKEKB at $\mathcal{L}_{\text{design}} = 6 \cdot 10^{35}/(\text{cm}^2\,\text{s})$, a parametric extrapolation is performed. The total rate is modeled as a sum of the individual contributions, each scaled according to the functional dependencies described in Equation A.3–Equation A.6:

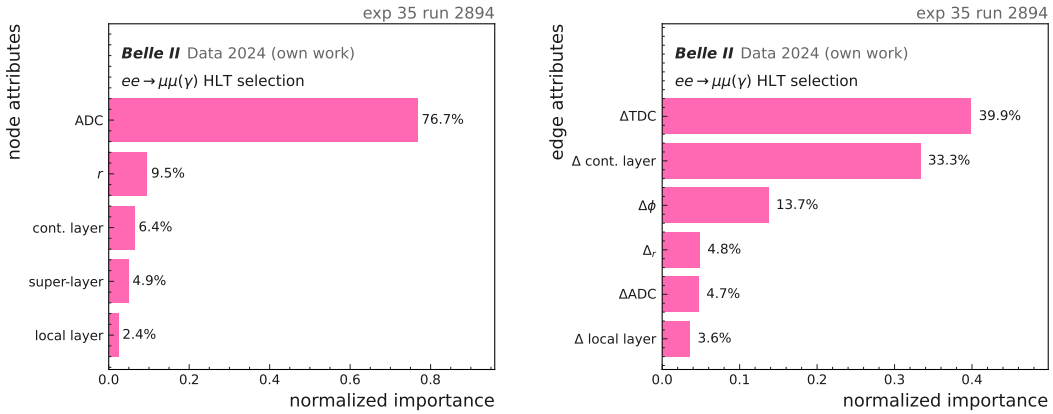
$$\begin{aligned} R_{\text{total}} = \; & a_{T,\text{LER}} \cdot \frac{I_{\text{LER}}^2}{n_b\,\sigma_{xyz,\text{LER}}} + a_{B,\text{LER}} \cdot I_{\text{LER}} P_{\text{eff,LER}} \\ & + a_{T,\text{HER}} \cdot \frac{I_{\text{HER}}^2}{n_b\,\sigma_{xyz,\text{HER}}} + a_{B,\text{HER}} \cdot I_{\text{HER}} P_{\text{eff,HER}} \\ & + a_{\mathcal{L}} \cdot \mathcal{L} \\ & + R_{\text{cosmic}}, \end{aligned} \tag{A.8}$$

where a_T, a_{BG}, and $a_{\mathcal{L}}$ are fitted coefficients extracted from the data taken at different machine parameter settings. The fit is performed separately for each trigger bit of interest. Without algorithmic or hardware upgrades, the CDC stand-alone `stt` and `ff30` rate are projected to exceed this threshold well before the design luminosity is reached.

A.2.2.5. Systematic uncertainties

The main sources of systematic uncertainty in the rate extrapolation include variations in the vacuum pressure and beam optics between fill conditions, non-linearities in the Touschek lifetime at high bunch currents, and trigger dead time corrections. These are assessed by repeating the fits under varied assumptions and comparing the resulting extrapolated rates.

A.2.3. Model feature importance



(a) node-level

(b) edge-level

Figure A.8.: Feature importance for the Level-1 trigger (L1 trigger) application model displaying node- and edge-level features evaluated by masking the model input during inference after application of the first feature-pruning step.

A.2.4. Trigger simulation hyper-parameters

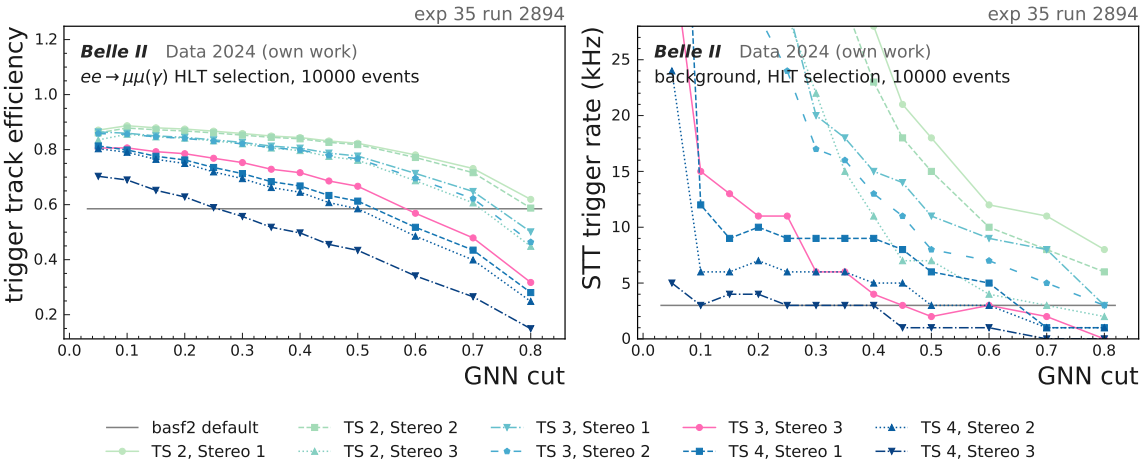


Figure A.9.: Evaluation of the impact of different minimum hit layer requirements in the outer super-layers during TS construction as well as the required number of aligned stereo TSs. The configuration "TS 3, stereo 3", which requires three hits for a TS to be built and three aligned stereo track segments for a track to be reconstructed, achieves trigger rates comparable to the basf2 baseline for GNN cut values above 0.4, while simultaneously providing higher track reconstruction efficiency for cut values up to 0.6.

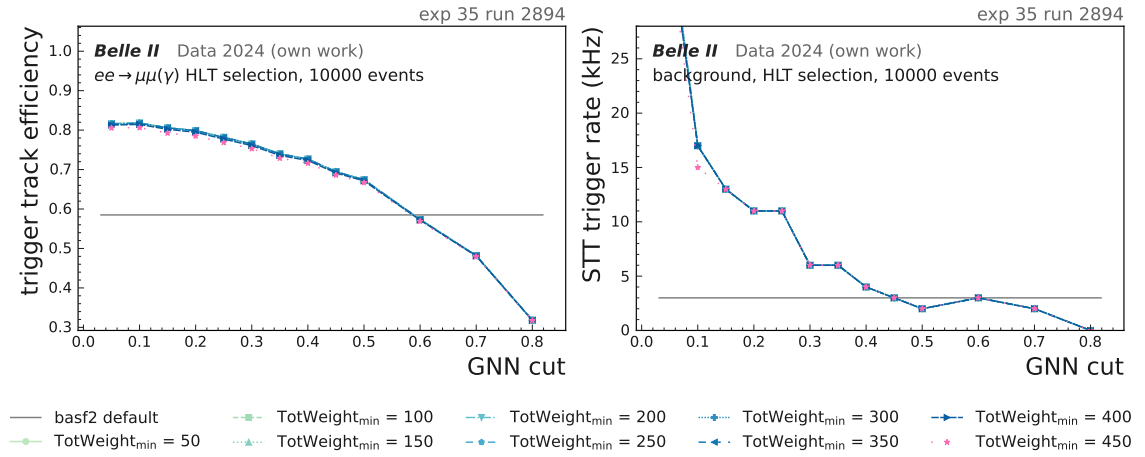


Figure A.10.: The total weight of a Hough cluster in the range between $\text{TotWeight}_{\text{min}} \in [50, 450]$ has no effect on the trigger track efficiency and the STT trigger rate.

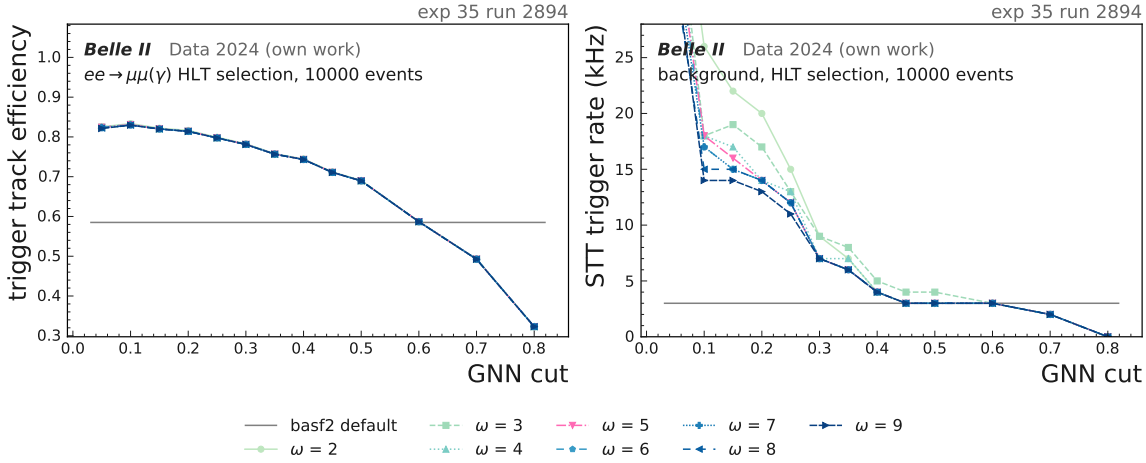


Figure A.11.: The cluster shape of a Hough cluster in the ω dimension range between $\omega \in [2, 9]$ has no effect on the trigger track efficiency and only a marginal effect on the STT trigger rate.

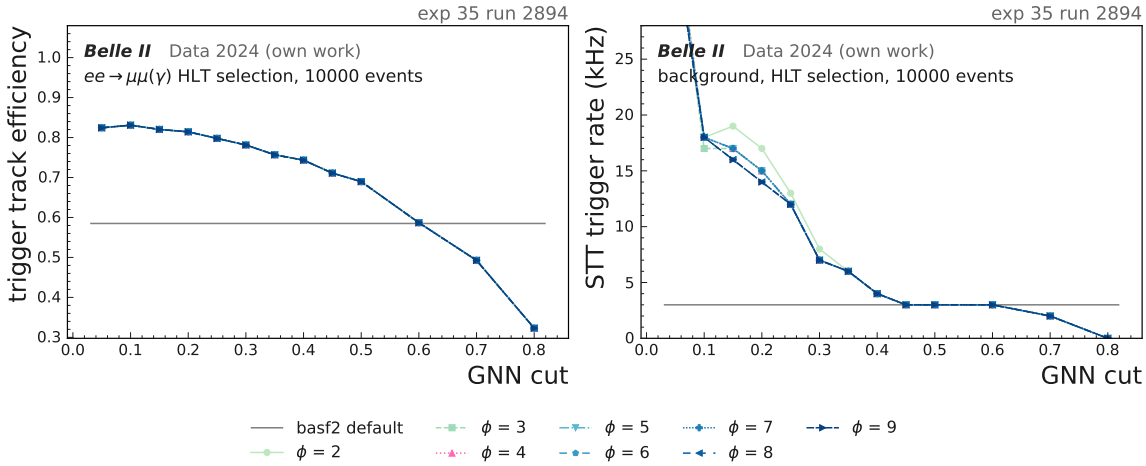


Figure A.12.: The cluster shape of a Hough cluster in the ϕ dimension range between $\phi \in [2, 9]$ has no effect on the trigger track efficiency and only a marginal effect on the STT trigger rate.

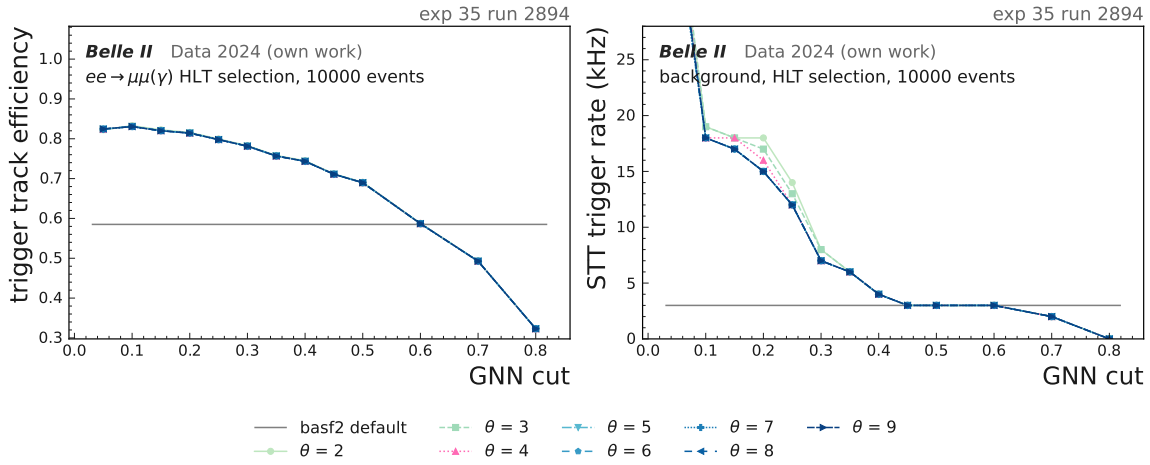


Figure A.13.: The cluster shape of a Hough cluster in the θ dimension range between $\theta \in [2, 9]$ has no effect on the trigger track efficiency and only a marginal effect on the STT trigger rate.

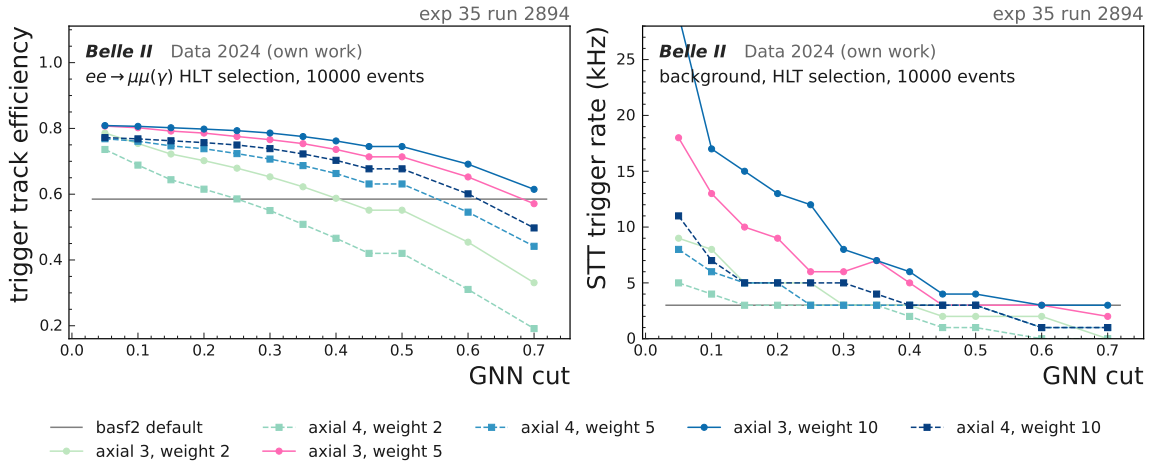


Figure A.14.: The weight used in the weighted binary cross-entropy (BCE) loss with logits has a significant effect on the track-level performance of the GNN filter. For different weights between 2 and 10 the best compromise between efficiency and low background rate is found for a weight of 5.

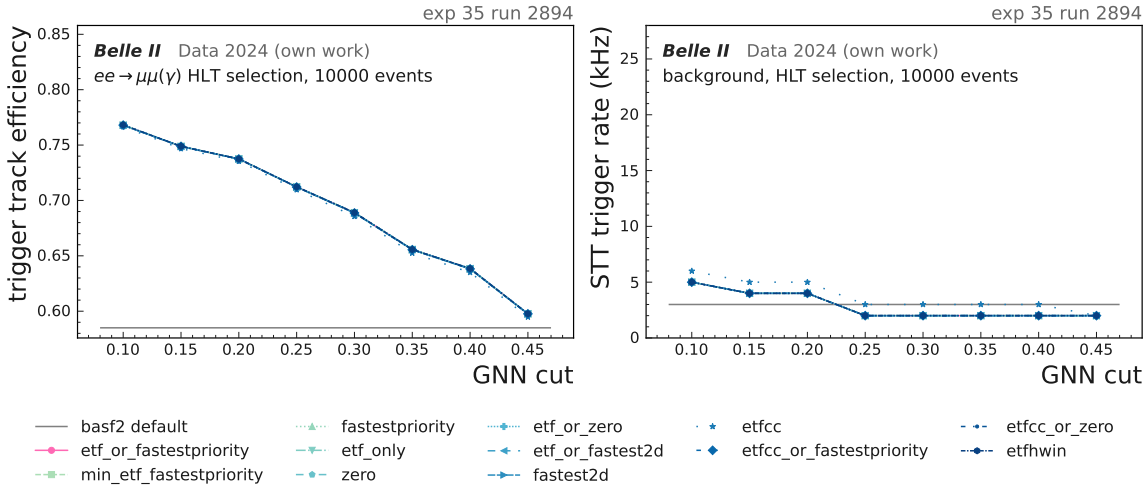


Figure A.15.: Different event time finder (ETF) outputs and combinations of outputs can be used by the neural network fitter. The effect of this choice is marginal.

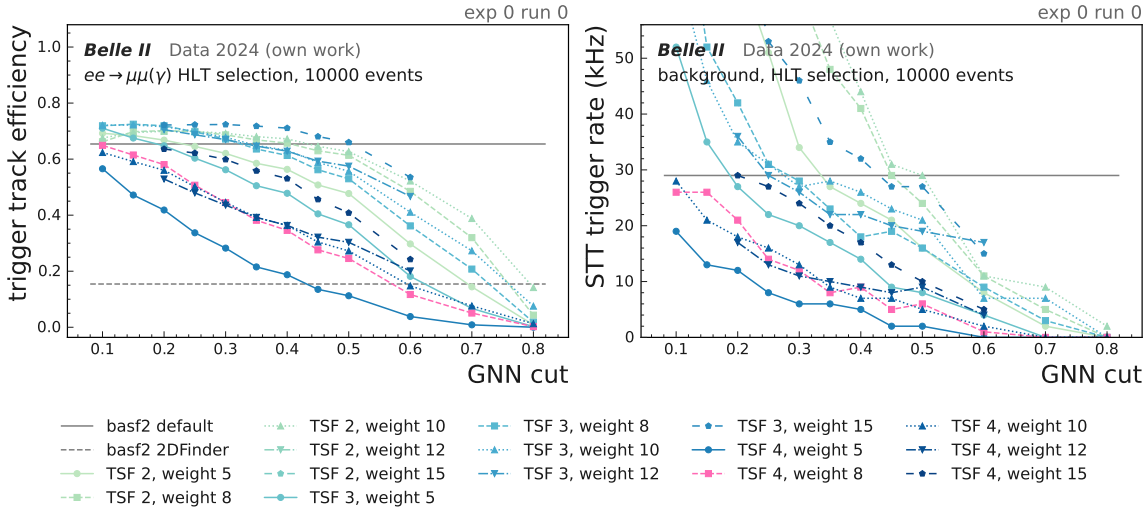


Figure A.16.: When evaluating the parameter space containing the number of hits per TS and the weight used in the weighted BCE loss with logits, the best compromise between efficiency and low fake trigger rate is identified for the combination "TSF 4, weight 8".

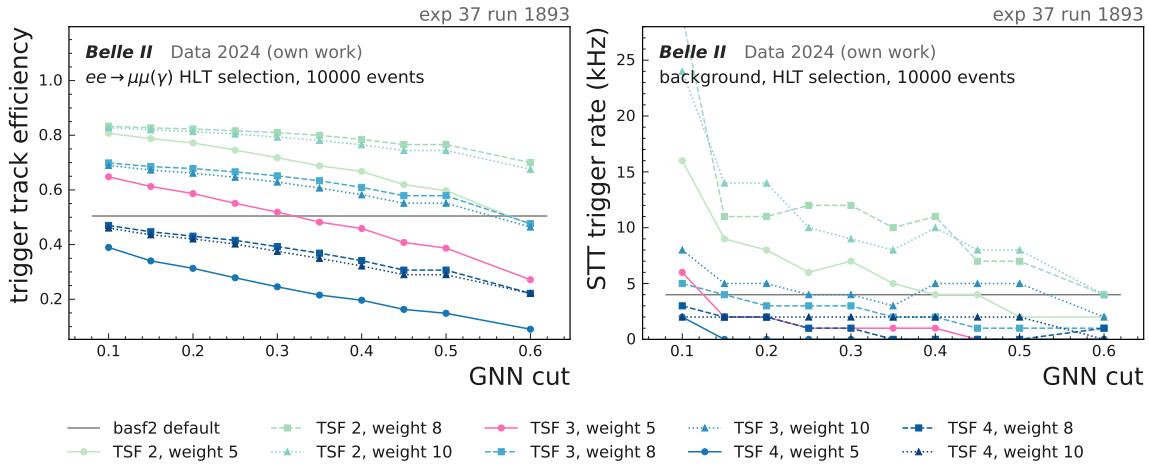
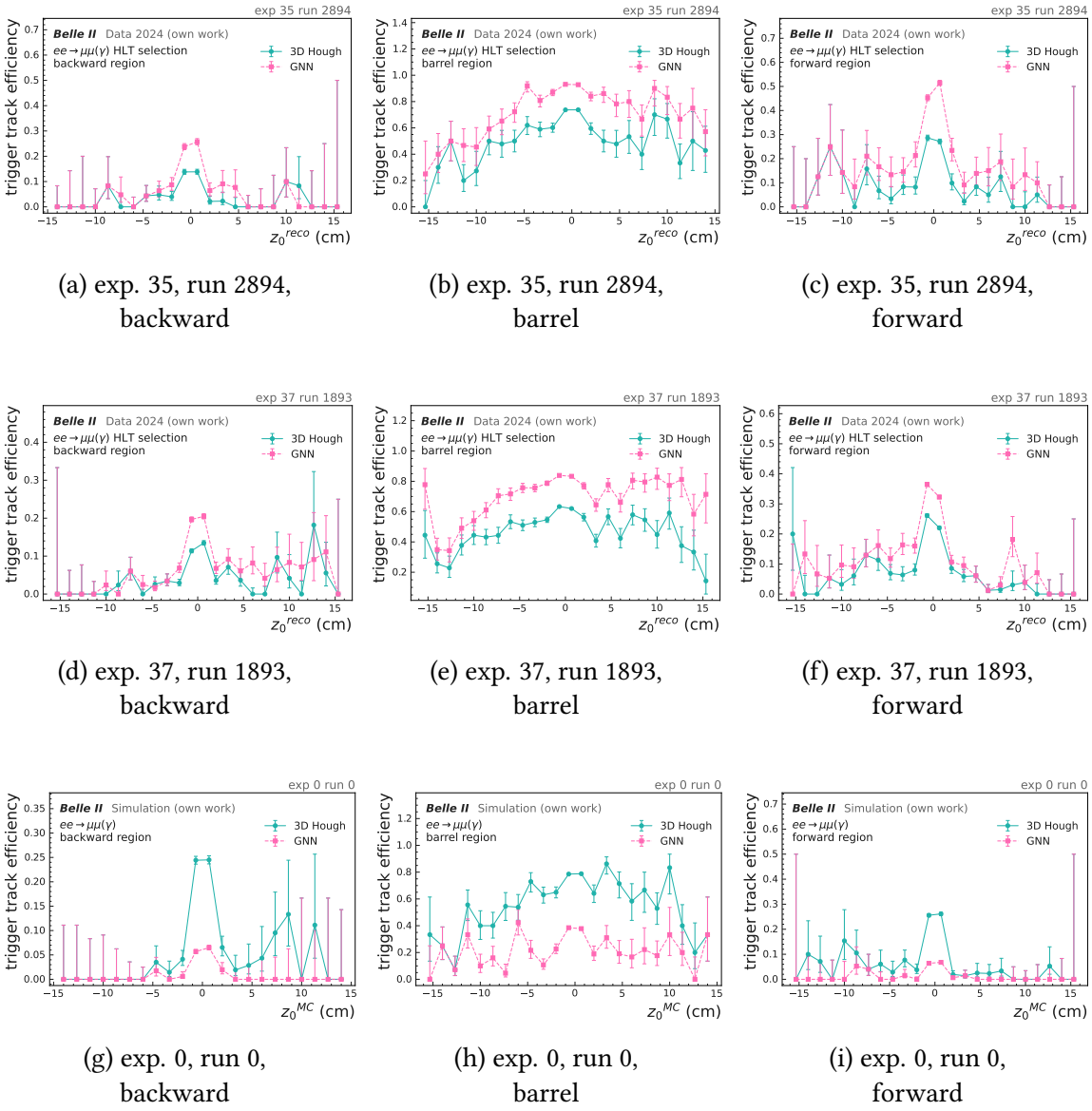


Figure A.17.: For experiment 37, the same configuration as for experiment 35 is used ("TSF 3, weight 5").

A.2.5. Trigger track efficiency for different regions over z_0



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

(c) exp. 35, run 2894, forward

(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward

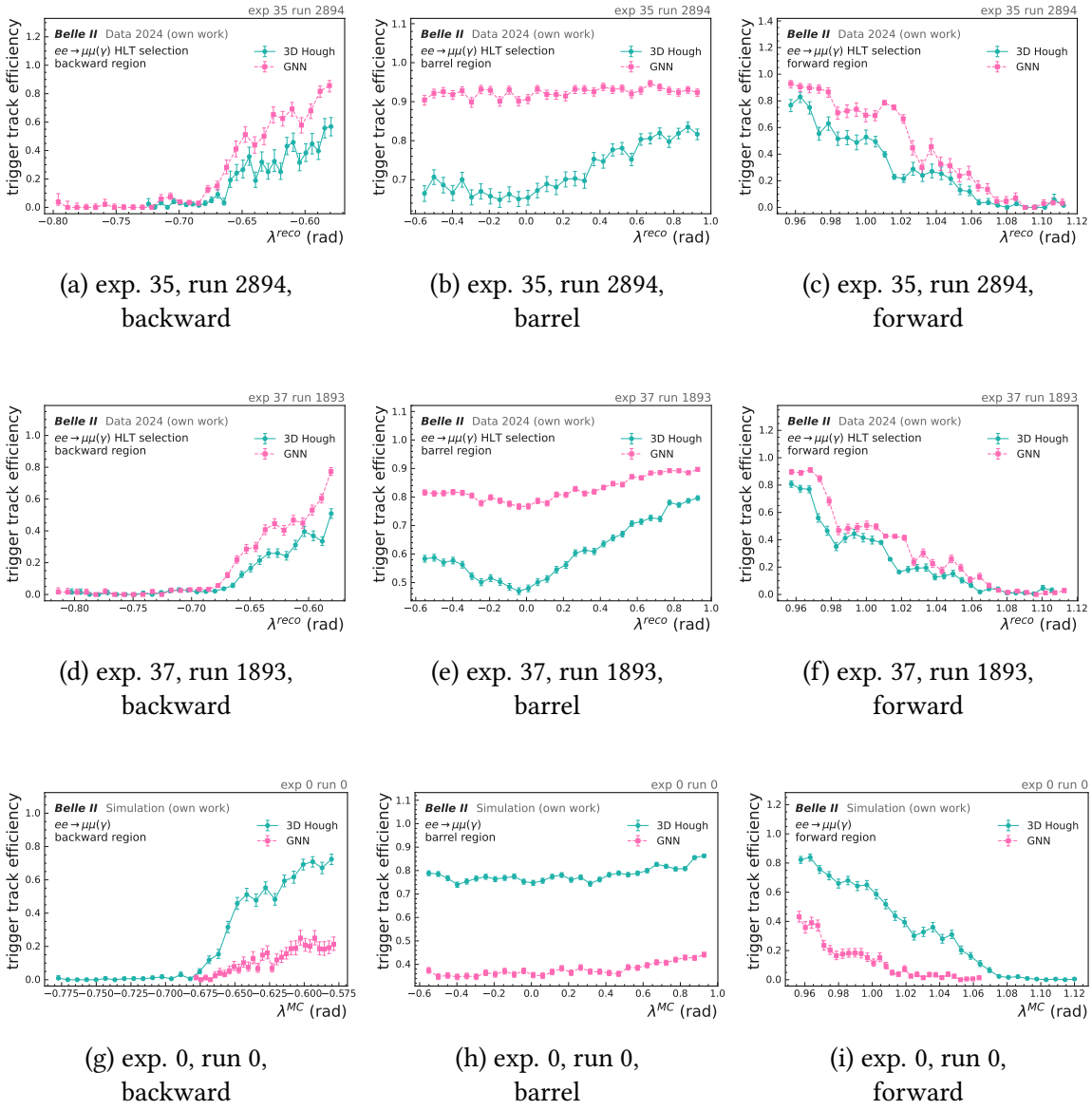
(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.18.: Trigger track fitting efficiency over impact parameter z_0 for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

A.2.6. Trigger track efficiency for different regions over λ



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

(c) exp. 35, run 2894, forward

(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward

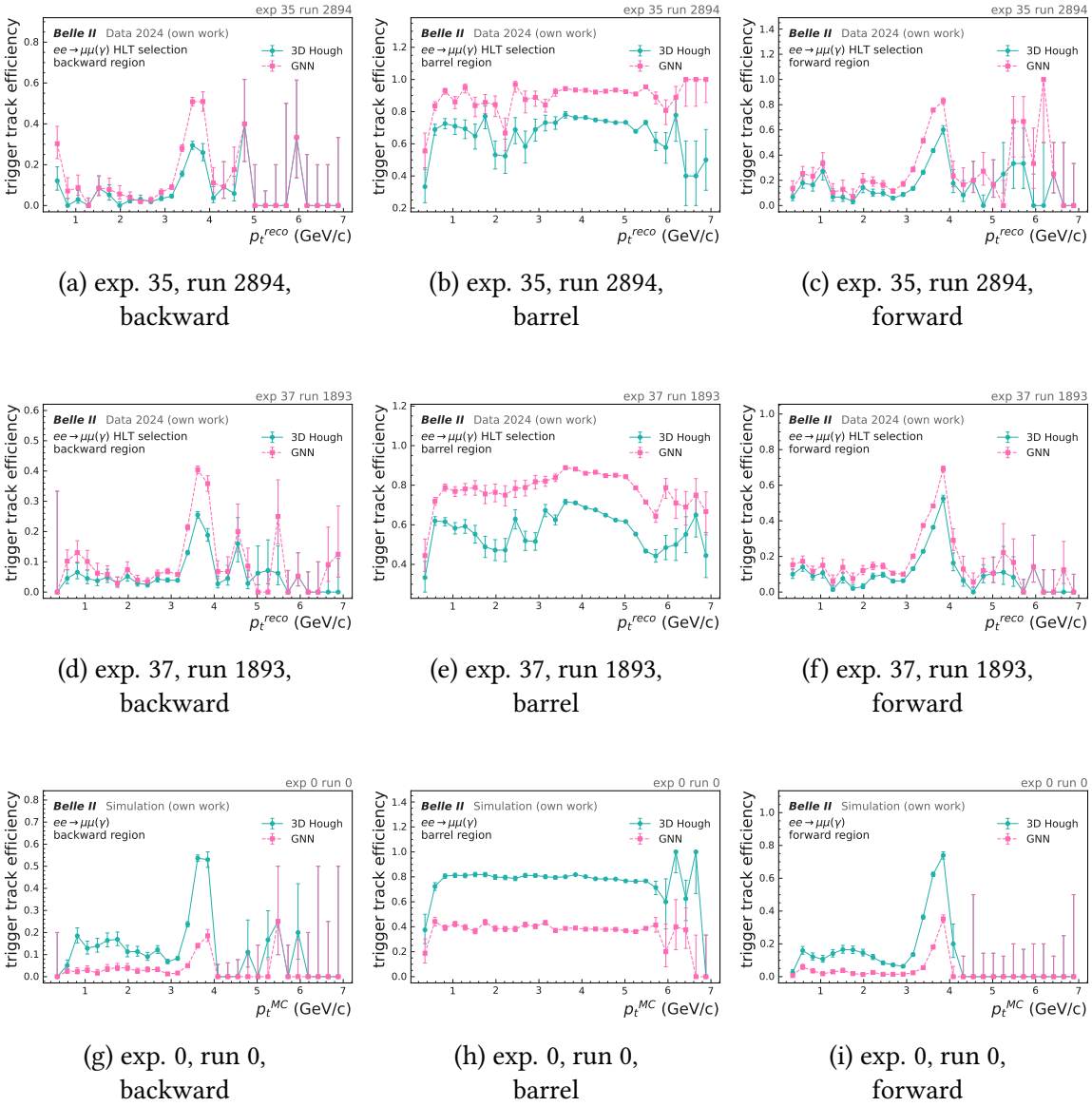
(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.19.: Trigger track fitting efficiency over the dip angle λ for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

A.2.7. Trigger track efficiency for different regions over p_T



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

(c) exp. 35, run 2894, forward

(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward

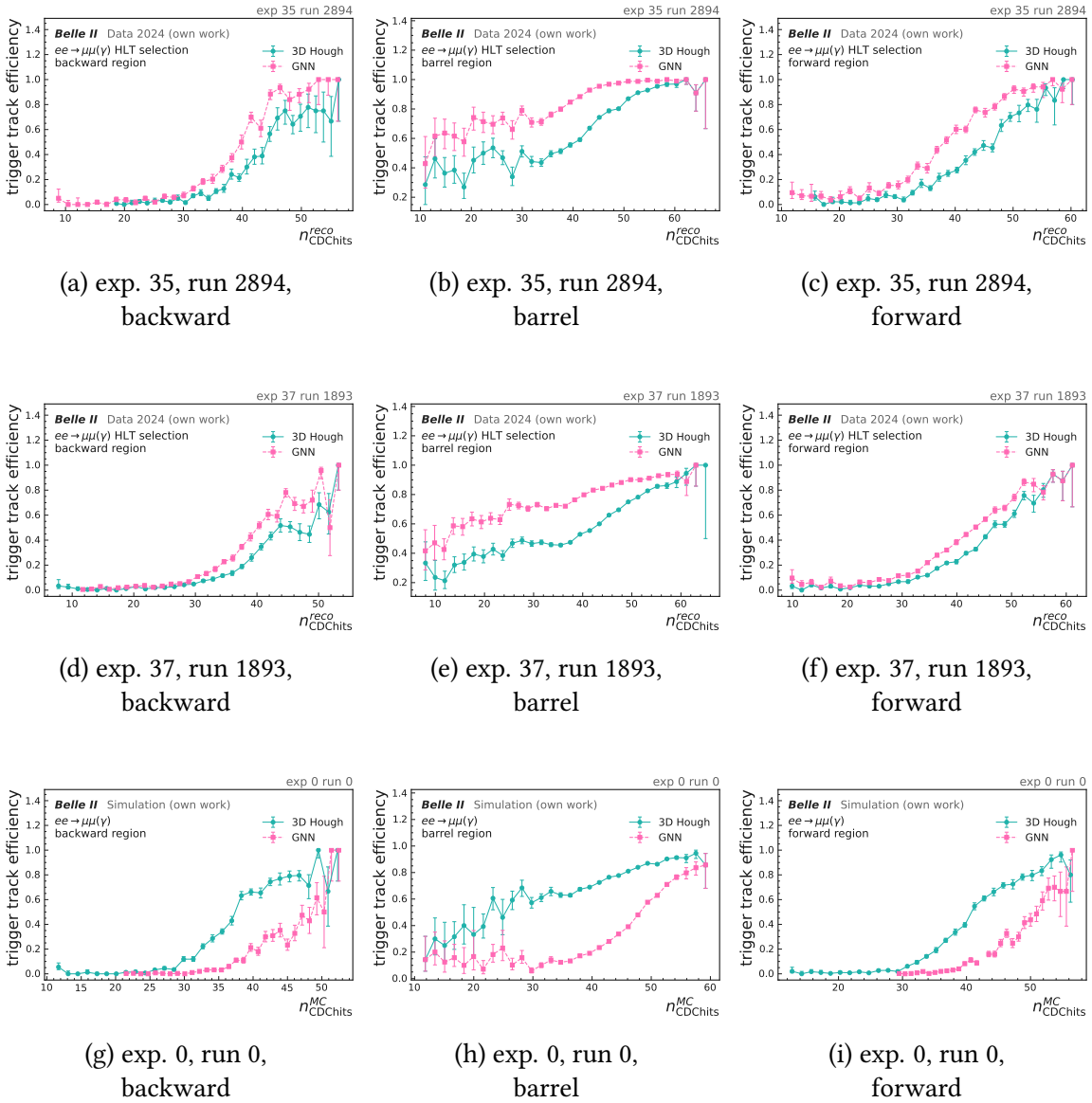
(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.20.: Trigger track fitting efficiency over the transverse momentum p_T for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for exp 0.

A.2.8. Trigger track efficiency for different regions over number of hits



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

(c) exp. 35, run 2894, forward

(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward

(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

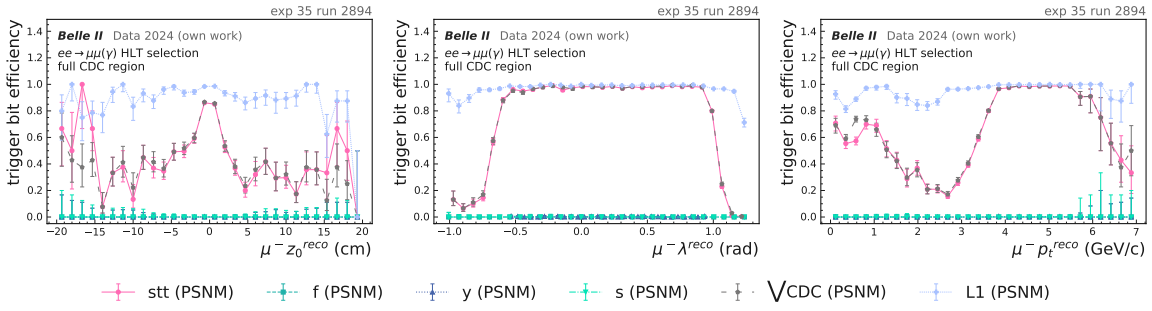
Figure A.21.: Trigger track fitting efficiency over the number of related hits to the offline reconstructed track for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

A.2.9. FTDL trigger rates

Table A.3.: Estimated FTDL trigger rates for selected lines before pre-scaling (bfyo, f, s, ssb, stt6, y) for the adjusted 3DHough + GNN configuration compared to the basf2 2DHough and 3DHough tracking. Rates are given in kHz with statistical uncertainties.

Trigger bit rates	**bfyo (kHz)**	**f (kHz)**	**s (kHz)**	**ssb (kHz)**	**stt6 (kHz)**	**y (kHz)**
Exp. 35, 2894						
basf2 2DHough	$3.80^{+0.98}_{-0.78}$	$177.00^{+5.99}_{-5.80}$	$3.40^{+0.93}_{-0.73}$	$0.00^{+0.20}_{-0.00}$	$55.80^{+3.43}_{-3.23}$	$166.40^{+5.82}_{-5.62}$
basf2 3DHough	$0.40^{+0.40}_{-0.20}$	$33.80^{+2.70}_{-2.50}$	$1.20^{+0.60}_{-0.40}$	$0.00^{+0.20}_{-0.00}$	$4.40^{+1.04}_{-0.84}$	$23.00^{+2.24}_{-2.05}$
adj. 3DHough + GNN	$2.60^{+0.83}_{-0.63}$	$70.20^{+3.83}_{-3.64}$	$5.00^{+1.10}_{-0.90}$	$0.20^{+0.32}_{-0.12}$	$9.20^{+1.46}_{-1.26}$	$43.40^{+3.04}_{-2.84}$
Exp. 37, 1893						
basf2 2DHough	$8.40^{+1.40}_{-1.20}$	$374.20^{+8.58}_{-8.40}$	$8.40^{+1.40}_{-1.20}$	$0.00^{+0.20}_{-0.00}$	$124.00^{+5.05}_{-4.85}$	$358.60^{+8.41}_{-8.22}$
basf2 3DHough	$1.40^{+0.64}_{-0.44}$	$57.00^{+3.47}_{-3.27}$	$2.80^{+0.85}_{-0.65}$	$0.00^{+0.20}_{-0.00}$	$7.60^{+1.34}_{-1.14}$	$38.00^{+2.85}_{-2.65}$
adj. 3DHough + GNN	$2.40^{+0.80}_{-0.60}$	$89.40^{+4.31}_{-4.11}$	$5.60^{+1.16}_{-0.96}$	$0.00^{+0.20}_{-0.00}$	$9.80^{+1.50}_{-1.30}$	$55.40^{+3.42}_{-3.22}$
Exp. 0, 0						
basf2 2DHough	$5462.80^{+22.26}_{-22.27}$	$7219.60^{+19.99}_{-20.08}$	$61.80^{+3.60}_{-3.41}$	$0.40^{+0.40}_{-0.20}$	$4479.80^{+22.25}_{-22.23}$	$7066.60^{+20.32}_{-20.40}$
basf2 3DHough	$18.40^{+2.02}_{-1.82}$	$386.00^{+8.71}_{-8.52}$	$65.20^{+3.70}_{-3.50}$	$2.00^{+0.74}_{-0.54}$	$45.40^{+3.11}_{-2.91}$	$205.00^{+6.43}_{-6.24}$
adj. 3DHough + GNN	$2.20^{+0.77}_{-0.57}$	$74.20^{+3.94}_{-3.74}$	$3.40^{+0.93}_{-0.73}$	$0.00^{+0.20}_{-0.00}$	$11.00^{+1.59}_{-1.39}$	$40.40^{+2.94}_{-2.74}$

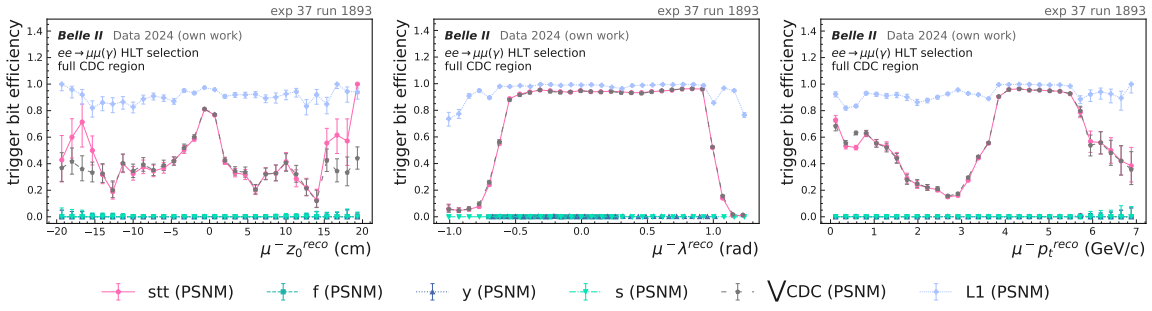
A.2.10. Trigger bit efficiencies



(a) exp. 35, run 2894, impact parameter z_0

(b) exp. 35, run 2894, dip angle λ

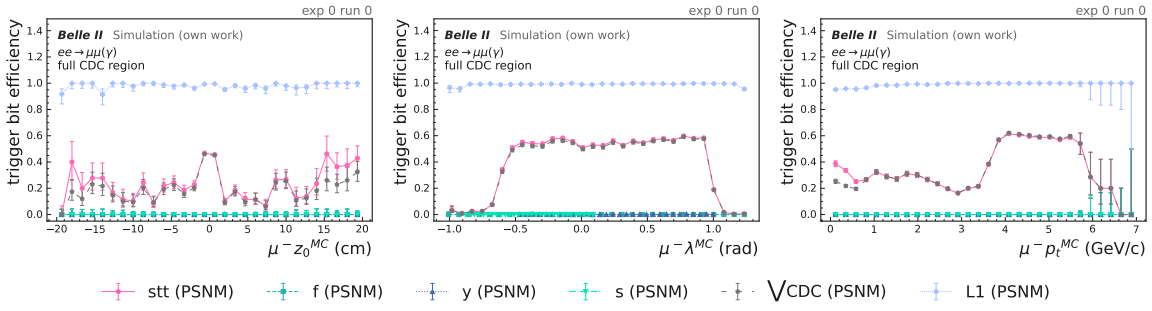
(c) exp. 35, run 2894, transv. momentum p_T



(d) exp. 37, run 1893, impact parameter z_0

(e) exp. 37, run 1893, dip angle λ

(f) exp. 37, run 1893, transv. momentum p_T



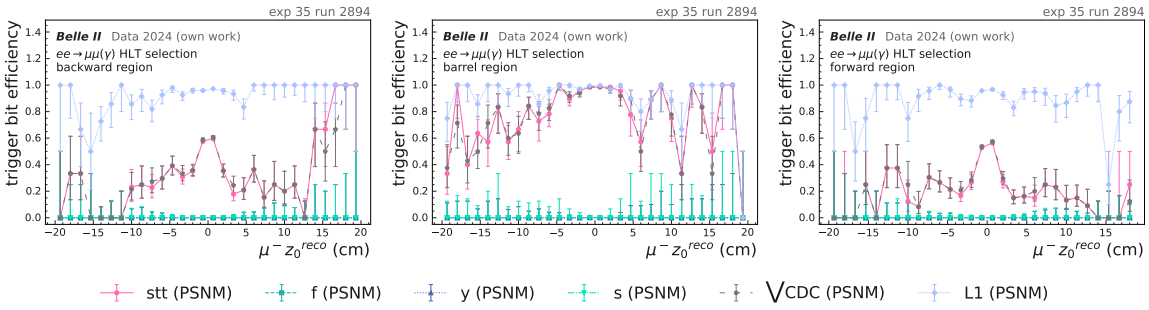
(g) exp. 0, run 0, impact parameter z_0

(h) exp. 0, run 0, dip angle λ

(i) exp. 0, run 0, transv. momentum p_T

Figure A.22.: PSNM trigger bit efficiencies for single-track trigger lines (stt, f, y, s) after pre-scaling over impact parameter z_0, the dip angle λ and transverse momentum p_T for three different experiments with increasing background levels for the GNN configuration evaluated on the μm of 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

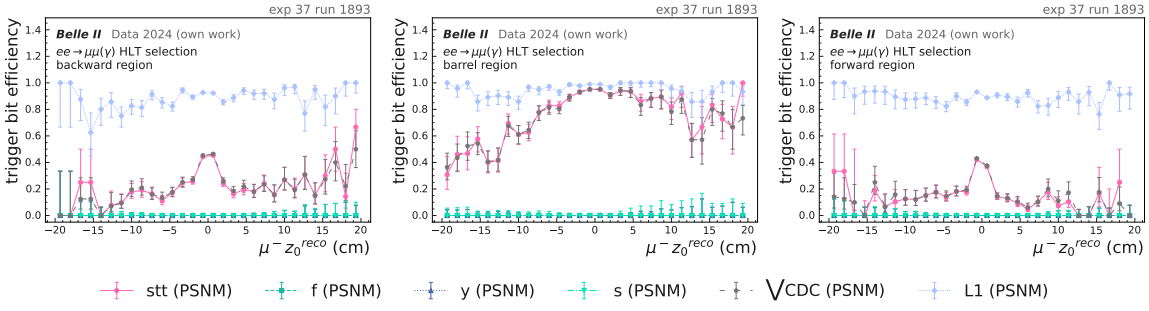
A.2.10.1. Trigger bit efficiency for different regions over z_0



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

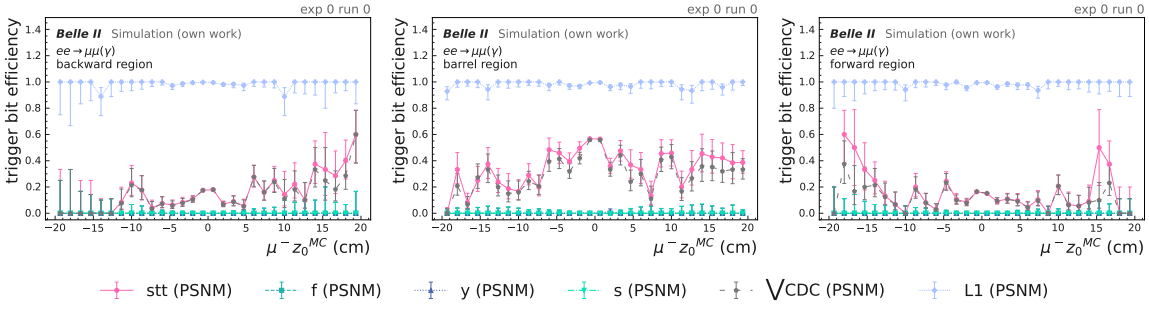
(c) exp. 35, run 2894, forward



(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward



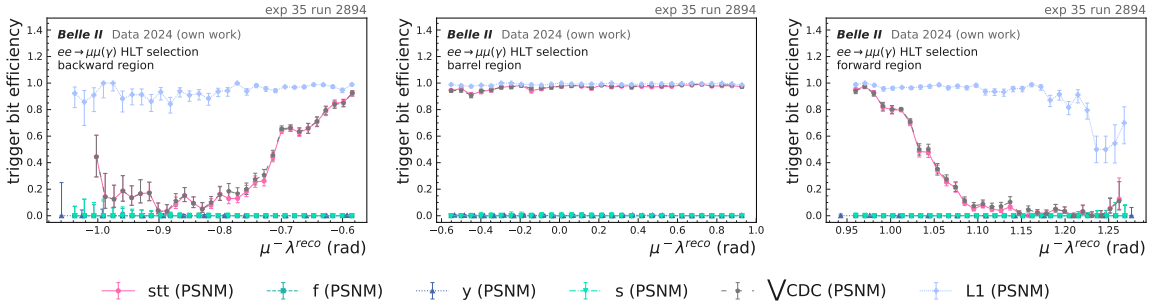
(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.23.: Trigger bit efficiencies over impact parameter z_0 for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

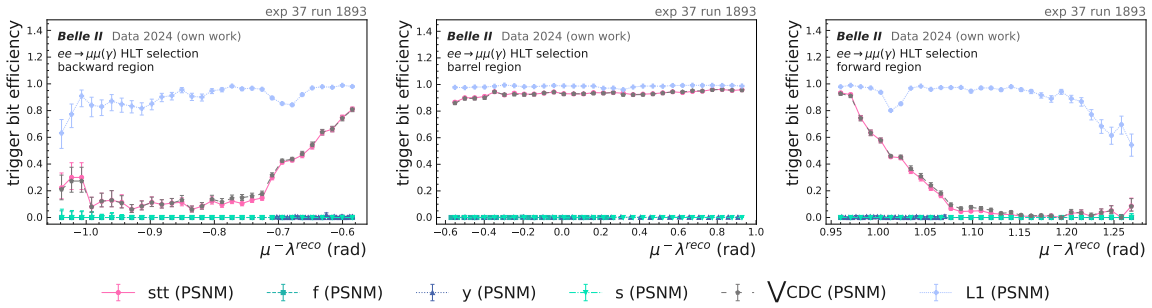
A.2.10.2. Trigger bit efficiency for different regions over λ



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

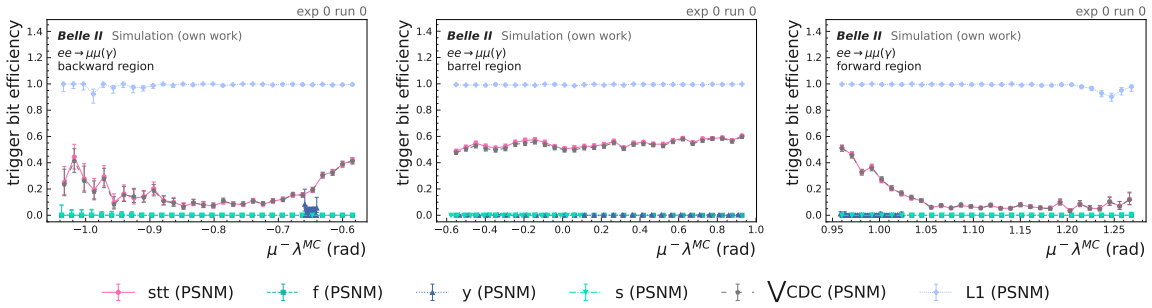
(c) exp. 35, run 2894, forward



(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward



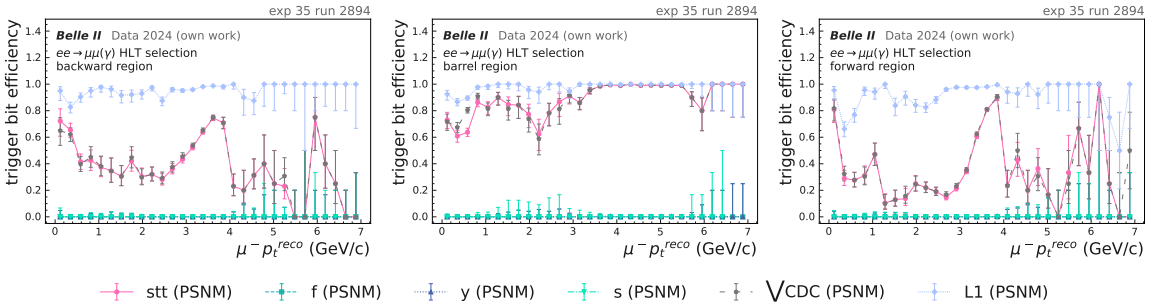
(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.24.: Trigger bit efficiencies over the dip angle λ for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

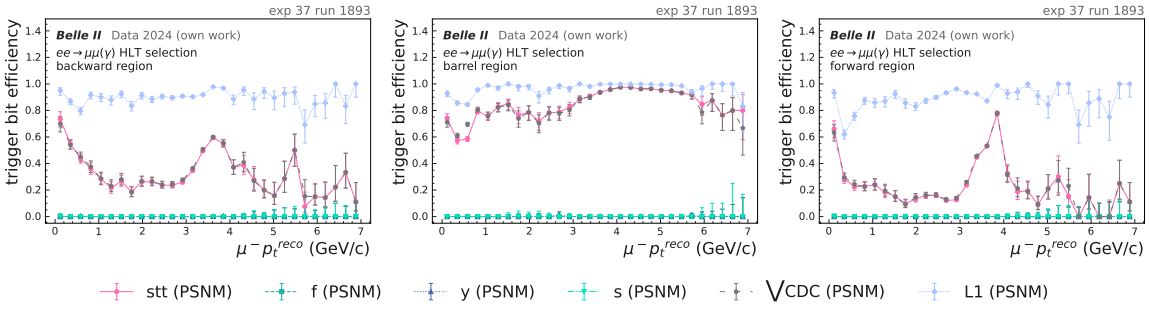
A.2.10.3. Trigger bit efficiency for different regions over p_T



(a) exp. 35, run 2894, backward

(b) exp. 35, run 2894, barrel

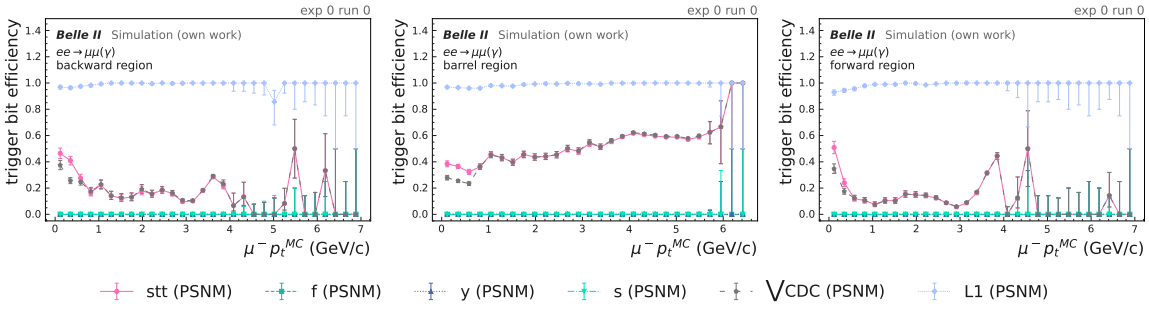
(c) exp. 35, run 2894, forward



(d) exp. 37, run 1893, backward

(e) exp. 37, run 1893, barrel

(f) exp. 37, run 1893, forward



(g) exp. 0, run 0, backward

(h) exp. 0, run 0, barrel

(i) exp. 0, run 0, forward

Figure A.25.: Trigger bit efficiencies over the transverse momentum p_T for the different detector regions (backward, barrel and forward) and for three different experiments with increasing background levels comparing the basf2 3D Hough finder configuration (green) with GNN filtering applied (magenta) for 50 000 HLT-selected $\mu^+\mu^-(\gamma)$ events for experiment 35 and 37 and generated $\mu^+\mu^-$ pairs for experiment 0.

List of Acronyms

ADC analog-to-digital converter. 12–14, 24, 25, 28, 42, 55, 56, 70, 75, 76, 80, 83, 98, 120, 123–128, 134–137, 152–154, 160, 205, 208, 210–213, 220

ALP axion-like particles. 5

ARICH aerogel-based ring-imaging Cherenkov counter. 9

AUC area under the curve. 61, 121–133, 135–139, 160, 166, 211–213

basf2 Belle II Analysis Software Framework. 1, 16, 27, 34, 51, 55, 59, 64, 67–69, 97, 98, 100, 102, 105, 112, 113, 116, 119, 123, 141–149, 160, 165, 182, 187–191, 193–195, 207, 213–217, 220–222

BCE binary cross-entropy. 86–88, 98, 143, 184, 185, 210, 216

BOP number of bit operation. 66, 121–133, 138, 139, 158, 160, 161, 167, 211–213

BRAM block random access memory. 17

CDC central drift chamber. 1, 3, 5, 9–15, 18, 20–25, 27–36, 38, 39, 41–46, 48, 49, 52, 53, 55–57, 59, 60, 62, 66–68, 70, 81, 83, 103, 106, 109, 114, 116, 119–126, 134, 136, 139–141, 148–150, 152, 153, 155, 156, 160, 161, 163, 165–167, 178, 180, 202, 205–207, 210–213, 219, 221, 222

CKF combinatorial Kalman filter. 27–29

CLB configurable logic block. 16, 17

clk clock. 17

CNN convolutional neural network. 3

CPU central processing unit. 1, 16, 17

DAF deterministic annealing filter. 29

DAQ data acquisition. 12–14, 18, 20, 23, 24, 36, 39, 120, 205

DNN deep neural network. 25

DR damping ring. 6

DSP digital signal processor. 17, 24, 157, 161, 167

ECL electromagnetic calorimeter. 9, 12, 15, 36, 37, 55, 148

ETF event time finder. 25, 31, 34, 185, 206, 216

FEE front-end electronic. 10, 12–14, 16, 23, 24, 30, 31, 119, 152, 155, 160, 166, 206, 214

FF flip-flop. 17, 156–158, 161, 167, 214, 222

FIFO first-in-first-out buffer. 155, 157, 214

FN false negative. 89

FP false positive. 89

FPGA Field-Programmable Gate Array. i, 1–3, 16, 17, 27, 30, 120, 121, 123, 152, 155, 157, 158, 160, 161, 163, 166, 167, 204, 205, 214

FPR false positive rate. 61

FTDL final trigger decision logic. 38, 66, 150

FTL fast timing layer. 25

GDL global decision logic. 15, 31, 35, 36, 206

GNN graph neural network. 1–3, 28, 31, 41–43, 46–49, 51, 52, 59–61, 67–70, 73, 75, 76, 78, 80–117, 119, 120, 127, 139–152, 157, 158, 160, 161, 163, 165–167, 169–177, 182, 184, 187–195, 206–211, 213–217, 220–222

GPU graphics processing unit. 16

GRL global reconstruction logic. 15, 31, 33, 35, 36, 206

HEP high-energy physics. 3, 41

HER high energy ring. 6, 7, 38, 179, 180

HLS high-level synthesis. 155

HLT high-level trigger. 1, 12, 16, 27, 54, 55, 69, 70, 112, 115, 116, 119, 122, 134, 137–139, 141, 147–149, 151, 160, 166, 187–190, 192–195, 207, 211–214, 216, 217, 219, 221

I/O input/output. 17

IN interaction network. 47–49, 131, 132

IP interaction point. 7–9, 20, 22, 25, 27, 29, 53, 55, 73, 107

ITT inner tracking and timing detector. 25

KLM K-long and muon detector. 9, 15, 18, 36, 148

L1 trigger Level-1 trigger. i, 1–3, 5, 12, 14–17, 20, 23–25, 27, 30–33, 35, 36, 38, 56, 66, 119–121, 123, 125, 148–150, 152, 153, 156, 160, 161, 163, 166, 167, 181, 206, 215, 219, 221

LER low energy ring. 6, 7, 38, 179, 180

LINAC injector linear accelerator. 6

LS2 long shutdown 2. 23

LUT look-up table. 16, 17, 30, 32, 140, 156–158, 161, 167, 206, 214, 222

MAC multiply-accumulate. 66, 120, 131

MC Monte Carlo. 42, 44–46, 51, 54, 55, 59, 67, 69–72, 125, 143, 145, 146, 207, 219

ML machine learning. 1, 3, 16, 51

MLP multi-layer perceptron. 35, 47, 48, 66, 127, 129, 130, 155, 158, 207, 214

MUX multiplexer. 17

MVA multi-variate analysis. 27, 28, 41, 43, 60, 64, 67–70, 76, 97, 100, 101, 104, 106–110, 112, 114–117, 165, 171–175, 207, 210, 214, 215, 220–222

PE processing element. 155, 157, 214

POCA point of closest approach. 27

PSNM pre-scaled and masked. 38, 66, 149

PXD silicon-pixel detector. 9, 16, 25, 27, 28, 163

QAT quantization-aware training. 128

ROC receiver operating characteristic. 61, 121–123, 125–127, 139, 160, 211, 213

RTL register-transfer level. 155, 214

SB switching block. 17

SEU single-event upset. 23

SoC system-on-chip. 17

STT single track trigger. 25, 37–39, 119, 120, 140, 142–144, 146, 148–151, 160, 167, 182–184, 213–216, 221

SVD silicon-strip vertex detector. 9, 14, 25, 27, 28, 163

TDC time-to-digital Converter. 12, 13, 24, 25, 28, 42, 46, 55, 70, 73, 75, 76, 80, 98, 120, 123–125, 136, 153, 154, 205, 210–212, 220

TOP time-of-propagation counter. 9, 15, 18, 36, 148

TOT time-over-threshold. 12–14, 28, 42, 55, 70, 73, 75, 76, 80, 98, 125, 153, 154, 205, 210, 220

TP true positive. 89

TPR true positive rate. 61

TRG trigger. 13, 24, 30, 34, 47, 56, 59, 62, 64, 119, 123, 125, 134, 138, 156, 160, 205, 211

TS track segment. 30, 32–35, 62, 65, 119, 124, 140–146, 160, 166, 182, 185, 206, 213, 215, 216, 221

TSF track segment finder. 30, 31, 62, 119, 123, 140, 143, 144, 152, 160, 166, 206

VTX vertex detector. 25

Glossary

(g-2) The anomalous magnetic moment of a charged lepton, defined as the deviation of its gyromagnetic factor g from the Dirac value 2, providing a precision test of the Standard Model (SM) and a probe of possible new physics. 5

EvtGen A Monte Carlo Generator suited for the decay of heavy flavour particles. See [118]. 51, 53, 54

GEANT4 A toolkit for simulating the passage of particles through matter using a wide variety of phenomenological models. See [119]. 51

GENFIT2 An experiment-independent framework for track reconstruction in particle and nuclear physics. See [75]. 28, 29

KKMC A Monte Carlo generator specifically for lepton and quark pair production at lepton colliders. See [120]. 53, 54

ASIC Application-Specific Integrated Circuit, a microchip designed for a particular use or application rather than general-purpose use. 12, 16, 28

Cabibbo–Kobayashi–Maskawa (CKM) The CKM matrix is a unitary matrix in the SM that describes the mixing and transition probabilities between different quark flavors in weak interactions. 5, 7

CP violation A phenomenon in particle physics where the combined symmetries of charge conjugation (C), which swaps particles with their antiparticles, and parity (P), which inverts spatial coordinates, are not conserved. CP violation implies that the laws of physics are not exactly the same for matter and antimatter and plays a crucial role in explaining the observed matter-antimatter asymmetry in the universe. 5, 6, 8

cross-talk In the presence of a substantial charge deposition in a single readout channel of the CDC, it is possible that adjacent channels are also spuriously activated as a consequence of charge leakage into neighbouring channels. Cross-talk results in large cluster-like patterns. 28, 56, 75, 81

FastBDT FastBDT is a high-performance, optimized implementation of boosted decision trees designed for fast training and evaluation, particularly in large-scale data analysis tasks. 28, 29

Flavour changing neutral current (FCNC) Processes in which the flavour of a fermion (such as a quark or lepton) changes without altering its electric charge, mediated by neutral gauge bosons. In the Standard Model such transitions are forbidden at tree level and can only occur via higher-order loop diagrams, making them highly suppressed and sensitive probes of physics beyond the Standard Model. 5

KEKB The electron-positron collider at which the Belle experiment was located. 7, 19

Legendre transformation The Legendre transformation is a change of variables that replaces a function's dependence on one variable x by a dependence on its conjugate "slope" variable p. It is widely used, *e.g.* to go from the Lagrangian $L(q, \dot{q})$ to the Hamiltonian $H(q, p)$ or from one thermodynamic potential to another. For a differentiable, convex function $f : \mathbb{R} \to \mathbb{R}$, the Legendre transform is typically defined by

$$f^*(p) = \sup_{x \in \mathbb{R}} (px - f(x)).$$

Equivalently, if $p = f'(x)$ can be inverted to give $x(p)$, the Legendre transform can be written as

$$f^*(p) = px(p) - f\left(x(p)\right).$$

In track finding, the "original" space corresponds to the r-ϕ (or x-y) plane, where circles describe the particle trajectories and the Legendre parameter space (ρ, θ) corresponding to straight lines plays the role of the conjugate variable. A CDC hit is not represented by a single point but by a drift circle of radius r_{drift} centered at (x_0, y_0),

$$(x - x_0)^2 + (y - y_0)^2 = r_{\text{drift}}^2. \tag{A.9}$$

Any track compatible with this hit, parametrized in the transverse plane by the Legendre representation

$$\rho = x \cos\theta + y \sin\theta, \tag{A.10}$$

must be tangent to the corresponding drift circle. Imposing the tangency condition yields the Legendre-space relation for this hit,

$$\rho = x_0 \cos\theta + y_0 \sin\theta \pm r_{\text{drift}}, \tag{A.11}$$

which results in two curves associated with the two possible tangents (on the left and right sides of the sense wire). Consequently, the track-finding problem is reduced to the identification of regions of high density in the (ρ, θ) parameter space. An analogous construction is used to incorporate stereo-layer information: for each stereo hit, the reconstructed position along the track is expressed as a straight line in the $(z_0, \tan\lambda)$ space,

$$z_0 = z_{\text{rec}} - \tan\lambda \cdot s_{\text{rec}},$$

where s_{rec} is the path length to the stereo hit and z_{rec} its reconstructed z-coordinate. Lines corresponding to different stereo hits intersect in $(z_0, \tan\lambda)$, and the point of highest intersection density defines the longitudinal track parameters, in direct analogy to the Legendre approach in the transverse plane.. 27, 29, 202

lepton flavour violation (LFV) Lepton flavour violation is any process in which a charged lepton changes its flavour (e, μ, τ) without conserving the individual lepton flavour numbers, e.g. $\mu \longrightarrow e\gamma$, which is forbidden in the SM with massless neutrinos. 5

Malter effect Insulating deposits on cathode surfaces due to prolonged exposure to ioniziing radiation lead to self-sustaining currents [65]. 21, 22

nano-beam scheme The nano beam scheme is a collider design approach that uses tightly focused beams at the interaction point to increase luminosity primarily by reducing the beam size rather than by increasing beam current. 7, 18, 19

Poisson A discrete probability distribution that gives the probability of a given number of events occurring in a fixed interval of time or space, assuming the events occur independently and at a constant average rate. 52, 219

PYTHIA8 A general-purpose Monte Carlo generator used to describe hard and soft interactions, parton distributions, initial- and final-state parton showers, multiparton interactions, fragmentation and decay. See [121]. 51

Quantum electrodynamics (QED) A relativistic quantum field theory of the electromagnetic interaction, describing how light (photons) interacts with charged particles such as electrons and positrons. 6, 19, 38

Run I Run I denotes the first long physics data-taking period of Belle II, spanning from early 2019 until June 2022, during which SuperKEKB and the detector were commissioned and gradually ramped to stable operation at luminosities up to $4.6 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and a total integrated luminosity of $424\,\mathrm{fb}^{-1}$. 20, 22, 57, 203, 206

Run II Belle II Run II is defined as starting with the post–LS1 physics data taking following the long shutdown and detector/accelerator upgrades. It started in January 2024 (exp. 30) and continuing onward; it does not yet have a fixed end run or experiment, as data taking is ongoing. The luminosities up to $4.4 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and a total integrated luminosity at the time of this thesis of $608\,\mathrm{fb}^{-1}$ including Run I. 20, 22, 57, 206

SAD SAD (Strategic Accelerator Design) is a KEK-developed software framework for accelerator physics design, simulation, and operation support. It offers tools for beam optics calculations, lattice design, beam dynamics studies, and online accelerator modeling, and is widely used for designing, commissioning, and optimizing electron and positron storage rings and linear accelerators. SAD also provides scripting, numerical analysis routines, and interfaces to control systems for both theoretical studies and practical machine operation [86]. 51

Standard Model of particle physics (SM) The theoretical framework in particle physics that describes all known fundamental particles and three of the four fundamental forces (electromagnetic, weak, and strong interactions), but not gravity. 1, 5, 52, 201, 203

SuperKEKB An upgrade of the KEKB electron-positron collider and the accelerator at which the Belle II experiment is located. 1, 5–8, 12, 18, 19, 23, 39, 166, 179, 180, 203, 205

UT The universal trigger (UT) board has been jointly developed for all sub-trigger systems to realize the core trigger logic. Table A.4 summarizes a comparison of the third to fifth UT generations. An upgrade program for the UT is currently in progress, aiming to enhance the FPGA resource capacity and the optical data transmission rate, thereby enabling the implementation of more sophisticated trigger algorithms that exploit increased information from the detector.
At present, the third-generation (UT3) and fourth-generation (UT4) boards are deployed for physics data taking. The number of UT4 boards is planned to be expanded in the short- to medium-term. The development and production of the next-generation UT board (UT5) are scheduled for the medium- to long-term period spanning 2024–2032. A high-end FPGA, Xilinx Versal, has been selected for UT5 in order to implement machine-learning-based trigger logic utilizing its large number of DSP blocks and integrated AI engine [65]. . 30

Table A.4.: Specification of the universal trigger boards taken from [7].

UT generation	UT3	UT4	UT5
Main FPGA (Xilinx)	Virtex6 XC6VHX380-565	Virtex Ultrascale XCVU080-190	Versal
Sub FPGA (Xilinx)	—	Artex7	Artex7, Zynq
# Logic gate	500k	2000k	8000k
Optical transmission rate	8 Gbps	25 Gbps	58 Gbps
# UT boards	30	30	10
Cost per a board (k$)	15	30	50
Time schedule	2014-	2019-2026	2024-2032

UT3 Third-generation universal trigger board. 16, 17, *see* UT

UT4 Fourth-generation universal trigger board. 16, 17, 24, 120, 158, *see* UT

UT5 Fifth-generation universal trigger board. 24, *see* UT

List of Figures

List of Tables

Bibliography

[1] P. Dorwarth. "Graph-Building and Input Feature Analysis for Edge Classification in the Central Drift Chamber at Belle II". MA thesis. Karlsruhe Institute of Technology (KIT), 2023.

[2] J. Eppelt. *Beam Background to ETP (BB2ETP)*. https://gitlab.etp.kit.edu/jeppelt/BB2ETP. Accessed: 2026-04-02.

[3] L. Reuter et al. "End-to-End Multi-track Reconstruction Using Graph Neural Networks at Belle II". In: *Comput. Softw. Big Sci.* 9.1 (2025), p. 6. DOI: 10.1007/s41781-025-00135-6. arXiv: 2411.13596 [physics.ins-det].

[4] G. Heine et al. "Hardware-accelerated GNN-based hit filtering for the Belle II Level-1 trigger". In: *J. Instrum.* 21.02 (2026), p. C02007. DOI: 10.1088/1748-0221/21/02/C02007. arXiv: 2511.04731 [physics.ins-det].

[5] G. Heine et al. "Hardware-Aware Design of a GNN-Based Hit Filtering Algorithm for the Belle II Level-1 Trigger". In: *preprint* (2026). arXiv: 2602.17761 [hep-ex].

[6] T. Abe et al. "Belle II Technical Design Report". In: *preprint* (2010). arXiv: 1011.0352 [physics.ins-det].

[7] H. Aihara et al. "The Belle II Detector Upgrades Framework Conceptual Design Report". In: *preprint* (2024). arXiv: 2406.19421 [hep-ex].

[8] Y. -T. Lai et al. "Design of the Global Reconstruction Logic in the Belle II Level-1 Trigger system". In: *Nucl. Instrum. Methods Phys. Res. A* 1078 (2025), p. 170577. DOI: 10.1016/j.nima.2025.170577. arXiv: 2503.02192 [hep-ex].

[9] Y.-X. Liu et al. "Development of deep neural network first-level hardware track trigger for the Belle II experiment". In: *Nucl. Instrum. Methods Phys. Res. A* 1084 (2026), p. 171248. DOI: 10.1016/j.nima.2025.171248.

[10] H. Qu, C. Li, and S. Qian. "Particle Transformer for Jet Tagging". In: *preprint* (2022). arXiv: 2202.03772 [hep-ph].

[11] V. Belis, P. Odagiu, and T. K. Aarrestad. "Machine learning for anomaly detection in particle physics". In: *Rev. Phys.* 12 (2024), p. 100091. DOI: 10.1016/j.revip.2024.100091. arXiv: 2312.14190 [physics.data-an].

[12] A. J. Larkoski, I. Moult, and B. Nachman. "Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning". In: *Phys. Rep.* 841 (2020), pp. 1–63. DOI: 10.1016/j.physrep.2019.11.001.

[13] D. Guest, K. Cranmer, and D. Whiteson. "Deep Learning and its Application to LHC Physics". In: *Ann. Rev. Nucl. Part. Sci.* 68 (2018), pp. 161–181. DOI: 10.1146/annurev-nucl-101917-021019. arXiv: 1806.11484 [hep-ex].

[14] T. Keck et al. “The Full Event Interpretation”. In: *Comput. Softw. Big Sci.* 3.1 (2019), p. 6. DOI: 10.1007/s41781-019-0021-8. arXiv: 1807.08680 [hep-ex].

[15] X. Ju et al. “Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors”. In: *preprint* (Mar. 2020). arXiv: 2003.11603 [physics.ins-det].

[16] J. Kieseler. “Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data”. In: *Eur. Phys. J. C* 80.9 (2020), p. 886. DOI: 10.1140/epjc/s10052-020-08461-2.

[17] G. DeZoort et al. “Charged Particle Tracking via Edge-Classifying Interaction Networks”. In: *Comput. Softw. Big Sci.* 5.1 (2021), p. 26. DOI: 10.1007/s41781-021-00073-z. arXiv: 2103.16701 [hep-ex].

[18] J. Duarte and J.-R. Vlimant. “Graph Neural Networks for Particle Tracking and Reconstruction”. In: *Artificial Intelligence for High Energy Physics*. 2022. Chap. Chapter 12, pp. 387–436. DOI: 10.1142/9789811234033_0012.

[19] C. Biscarat et al. “Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC”. In: *EPJ Web Conf.* 251 (2021), p. 03047. DOI: 10.1051/epjconf/202125103047.

[20] S. Farrell et al. “Novel deep learning methods for track reconstruction”. In: *preprint* (Oct. 2018). arXiv: 1810.06111 [hep-ex].

[21] A. Akram et al. “Application of Geometric Deep Learning for Tracking of Hyperons in a Straw Tube Detector”. In: *Comput. Softw. Big Sci.* 9.1 (2025), p. 17. DOI: 10.1007/s41781-025-00146-3. arXiv: 2503.14305 [hep-ex].

[22] J. Cao et al. “Object Detection Based on CNN and Vision-Transformer: A Survey”. In: *IET Comput. Vis.* 19.1 (2025), e70028. DOI: 10.1049/cvi2.70028.

[23] T. Shehzadi et al. “Object Detection with Transformers: A Review”. In: *Sensors* 25.19 (2025), p. 6025. DOI: 10.3390/s25196025.

[24] P. Reiser et al. “Graph neural networks for materials science and chemistry”. English. In: *Commun. Mater.* 3.1 (2022), p. 93. DOI: 10.1038/s43246-022-00315-6.

[25] M. Vaida and Z. Huang. “Multimodal graph neural networks in healthcare: a review of fusion strategies across biomedical domains”. In: *Front. Artif. Intell.* 8 (2025). DOI: 10.3389/frai.2025.1716706.

[26] G. Carleo et al. “Machine learning and the physical sciences”. In: *Rev. Mod. Phys.* 91.4 (2019), p. 045002. DOI: 10.1103/RevModPhys.91.045002. arXiv: 1903.10563 [physics.comp-ph].

[27] S. Dittmeier. “Online track reconstruction with graph neural networks on FPGAs for the ATLAS experiment”. In: *EPJ Web Conf.* 337 (2025), p. 01042. DOI: 10.1051/epjconf/202533701042.

[28] J. Kvapil et al. “Intelligent experiments through real-time AI: Fast Data Processing and Autonomous Detector Control for sPHENIX and future EIC detectors”. In: *PoS* ICHEP2024 (2025), p. 1033. DOI: 10.22323/1.476.1033. arXiv: 2501.04845 [physics.ins-det].

[29] A. Elabd et al. “Graph Neural Networks for Charged Particle Tracking on FPGAs”. In: *Front. Big Data* 5 (2022), p. 828666. DOI: 10.3389/fdata.2022.828666. arXiv: 2112.02048 [physics.ins-det].

[30] Z. Que et al. “JEDI-linear: Fast and Efficient Graph Neural Networks for Jet Tagging on FPGAs”. In: *preprint* (2025). arXiv: 2508.15468 [hep-ex].

[31] K. Akai, K. Furukawa, and H. Koiso. “SuperKEKB collider”. In: *Nucl. Instrum. Methods Phys. Res. A* 907 (2018), p. 188. DOI: 10.1016/j.nima.2018.08.017. arXiv: 1809.01958 [physics.acc-ph].

[32] E Kou et al. “The Belle II Physics Book”. In: *Prog. Theor. Exp. Phys.* 2019 (2019). [Erratum: Prog. Theor. Exp. Phys. 2020, 029201 (2020)], p. 123C01. DOI: 10.1093/ptep/ptz106.

[33] R. L. Workman et al. “Review of Particle Physics”. In: *Prog. Theor. Exp. Phys.* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.

[34] I. Adachi et al. “Evidence for $B^+ \to K^+ \nu\bar{\nu}$ decays”. In: *Phys. Rev. D* 109 (11 June 2024), p. 112006. DOI: 10.1103/PhysRevD.109.112006.

[35] P. Ecker. “Search for a dark Higgs boson produced in association with inelastic dark matter at the Belle II experiment”. PhD thesis. Karlsruher Institut für Technologie (KIT), 2024. 215 pp. DOI: 10.5445/IR/1000176696.

[36] S. Jia, W. Xiong, and C. Shen. “Status and Prospects of Exotic Hadrons at Belle II”. In: *Chin. Phys. Lett.* 40.12 (2023), p. 121301. DOI: 10.1088/0256-307X/40/12/121301. arXiv: 2312.00403 [hep-ex].

[37] T. Abe et al. “Achievements of KEKB”. In: *Prog. Theor. Exp. Phys.* 2013 (2013), 03A001. DOI: 10.1093/ptep/pts102.

[38] M. Satoh, Y. Ohnishi, and KEK K. Furukawa SuperKEKB Commissioning Group. *SuperKEKB 24-Hour Operation Summary*. Access date: 2026-04-13. Mar. 2026. URL: https://www-linac.kek.jp/skekb/snapshot/dailysnap.html.

[39] J. Brodzicka et al. “Physics Achievements from the Belle Experiment”. In: *Prog. Theor. Exp. Phys.* 2012 (2012), p. 04D001. DOI: 10.1093/ptep/pts072. arXiv: 1212.5342 [hep-ex].

[40] M. Bona et al. “SuperB: A High-Luminosity Asymmetric e+ e- Super Flavor Factory. Conceptual Design Report”. In: *preprint* (May 2007). arXiv: 0709.0451 [hep-ex].

[41] D. Matvienko. “The Belle II experiment: status and physics program”. In: *EPJ Web Conf.* 191 (2018), p. 02010. DOI: 10.1051/epjconf/201819102010.

[42] F. Bernlochner et al. “The Belle II Experiment at SuperKEKB – Input to the European Particle Physics Strategy”. In: *preprint* (Mar. 2025). arXiv: 2503.24155 [hep-ex].

[43] F. Becherer et al. “The new two-layer Belle II PiXel Detector”. In: *J. Instrum.* 20.07 (July 2025), p. C07060. DOI: 10.1088/1748-0221/20/07/C07060.

[44] K. Adamczyk et al. "The design, construction, operation and performance of the Belle II silicon vertex detector". In: *J. Instrum.* 17.11 (2022), P11042. DOI: 10.1088/1748-0221/17/11/P11042. arXiv: 2201.09824 [physics.ins-det].

[45] V. Bertacchi et al. "Track finding at Belle II". In: *Comput. Phys. Commun.* 259 (2021), p. 107610. DOI: 10.1016/j.cpc.2020.107610. arXiv: 2003.12466 [physics.ins-det].

[46] E. Torassa. "TOP detector for particle identification at Belle II". In: *Int. J. Mod. Phys. A* 39.26n27 (2024), p. 2442014. DOI: 10.1142/s0217751x24420144.

[47] Y. Yusa. "The ARICH detector at Belle II experiment". In: *PoS* EPS-HEP2013 (2013), p. 556. DOI: 10.22323/1.180.0556.

[48] V. Aulchenko et al. "Electromagnetic calorimeter for Belle II". In: *J. Phys. Conf. Ser.* 587.1 (Feb. 2015), p. 012045. DOI: 10.1088/1742-6596/587/1/012045.

[49] C. Ketter et al. "Design and commissioning of readout electronics for a K_L^0 and μ detector at the Belle II experiment". In: *Nucl. Instrum. Methods Phys. Res. A* 1082 (2026), p. 170893. DOI: 10.1016/j.nima.2025.170893.

[50] N. Taniguchi. "Central Drift Chamber for Belle-II". In: *J. Inst.* 12.06 (2017), p. C06014. DOI: 10.1088/1748-0221/12/06/C06014.

[51] Y. Iwasaki et al. "Level 1 Trigger System for the Belle II Experiment". In: *IEEE Trans. Nucl. Sci.* 58 (2011). Ed. by Sascha Marc Schmeling, pp. 1807–1815. DOI: 10.1109/TNS.2011.2119329.

[52] M. T. Prim et al. "Design and Performance of the Belle II High Level Trigger". In: *PoS* ICHEP2020 (2021), p. 769. DOI: 10.22323/1.390.0769.

[53] T. Bilka et al. "Alignment for the first precision measurements at Belle II". In: *EPJ Web Conf.* 245 (2020), p. 02023. DOI: 10.1051/epjconf/202024502023.

[54] T. V. Dong et al. "Calibration and alignment of the Belle II central drift chamber". In: *Nucl. Instrum. Methods Phys. Res. A* 930 (2019), pp. 132–141. DOI: 10.1016/j.nima.2019.03.072.

[55] I. Haide. "A Real-Time Graph Neural Network Trigger Algorithm for the Belle II Electromagnetic Calorimeter". PhD thesis. Karlsruher Institut für Technologie (KIT), 2025. 251 pp. DOI: 10.5445/IR/1000184927.

[56] S. Bähr et al. "The neural network first-level hardware track trigger of the Belle IIexperiment". In: *Nucl. Instrum. Methods Phys. Res. A* 1073 (2025), p. 170279. DOI: 10.1016/j.nima.2025.170279. arXiv: 2402.14962 [hep-ex].

[57] S. H. Kim et al. "Status of the Electromagnetic Calorimeter Trigger system at the Belle II experiment". In: *J. Instrum.* 12.09 (2017), p. C09004. DOI: 10.1088/1748-0221/12/09/C09004.

[58] L. Macchiarulo et al. "A probability-optimized fast timing trigger for the Belle II time of propagation detector". In: *IEEE Nuclear Science Symposium & Medical Imaging Conference*. 2010, pp. 630–635. DOI: 10.1109/NSSMIC.2010.5873835.

[59] S Lee et al. “Belle-II High Level Trigger at SuperKEKB”. In: *J. Phys. Conf. Ser.* 396.1 (2012), p. 012029. DOI: 10.1088/1742-6596/396/1/012029.

[60] A. Natochii et al. “Measured and projected beam backgrounds in the Belle II experiment at the SuperKEKB collider”. In: *Nucl. Instrum. Methods Phys. Res. A* 1055 (2023), p. 168550. DOI: 10.1016/j.nima.2023.168550. arXiv: 2302.01566 [hep-ex].

[61] H. Nakayama et al. “Beam Background Measurements at SuperKEKB/Belle-II in 2020”. In: *12th International Particle Accelerator Conference*. Aug. 2021. DOI: 10.18429/JACoW-IPAC2021-WEXA07.

[62] Belle II Collaboration. *Mirabelle Webpage*. https://mirabelle.belle2.org. Accessed: 2026-02-05. 2024.

[63] L. Reuter. “Track Finding with Graph Neural Networks in the Belle II Drift Chamber”. PhD thesis. Karlsruher Institut für Technologie (KIT), 2026. 287 pp. DOI: 10.5445/IR/1000189859.

[64] L. R. Koller and R. P. Johnson. “Visual Observations on the Malter Effect”. In: *Phys. Rev.* 52 (5 1937), pp. 519–523. DOI: 10.1103/PhysRev.52.519.

[65] S. H. Karpus et al. “Secondary Electron Emission From Thin Aluminium Foils Produced by High Energy Electron Beams”. In: *Prob. Atomic Sci. Technol.* 2021.6 (2021), pp. 38–41.

[66] T. Koga. *Private communication*. 2025. URL: https://inspirehep.net/authors/1351157.

[67] S. Skambraks et al. “A 3D track finder for the Belle II CDC L1 trigger”. In: *J. Phys. Conf. Ser.* 1525.1 (2020), p. 012102. DOI: 10.1088/1742-6596/1525/1/012102.

[68] K. L. Unger et al. “A multi-Hough-based displaced vertex track trigger for the Belle II experiment”. In: *J. Instrum.* 20.02 (2025), p. C02051. DOI: 10.1088/1748-0221/20/02/C02051.

[69] D. Auguste et al. “Upgrade of the Belle II Vertex Detector with Depleted Monolithic Active Pixel Sensors”. In: *J. Instrum.* 20.10 (Oct. 2025), p. C10013. DOI: 10.1088/1748-0221/20/10/C10013.

[70] J. Ott. *Inner Tracking and Timing*. https://indico.belle2.org/event/15311/contributions/101114/attachments/37421/55587/JOtt_ITT_trigWorkshop.pdf. Accessed: 2026-04-06.

[71] T. Kuhr et al. “The Belle II Core Software”. In: *Comput. Softw. Big Sci.* 3.1 (2019), p. 1. DOI: 10.1007/s41781-018-0017-9. arXiv: 1809.04299 [physics.comp-ph].

[72] Belle II collaboration. *Belle II Analysis Software Framework (basf2)*. https://doi.org/10.5281/zenodo.5574115.

[73] T. Alexopoulos et al. “Implementation of the Legendre Transform for track segment reconstruction in drift tube chambers”. In: *Nucl. Instrum. Methods Phys. Res. A* 592 (2008), pp. 456–462. DOI: 10.1016/j.nima.2008.04.038.

[74] A. Glazov et al. "Filtering tracks in discrete detectors using a cellular automaton". In: *Nucl. Instrum. Methods Phys. Res. A* 329 (1993), pp. 262–268. DOI: 10.1016/0168-9002(93)90945-E.

[75] T. Bilka et al. "Implementation of GENFIT2 as an experiment independent track-fitting framework". In: *preprint* (2019). arXiv: 1902.04405 [physics.data-an].

[76] T. Keck. "FastBDT: A Speed-Optimized Multivariate Classification Algorithm for the Belle II Experiment". In: 1.1 (2017), p. 2. DOI: 10.1007/s41781-017-0002-8.

[77] Jojosito and others. *GenFit/GenFit*. https://doi.org/10.5281/zenodo.10301439.

[78] K. L. Unger et al. "Realization of a state machine based detection for Track Segments in the Trigger System of the Belle II Experiment". In: *PoS* TWEPP2019 (2020), p. 145. DOI: 10.22323/1.370.0145.

[79] Y. Xue. *Trigger rate extrapolation using the data on April 16th 20242*. Internal Note, accessed: 2026-03-28. 2024. URL: https://docs.belle2.org/files/451/BELLE2-NOTE-TE-2024-022/1/BELLE2-NOTE-TE-2024-022.pdf.

[80] P. W. Battaglia et al. "Interaction Networks for Learning about Objects, Relations and Physics". In: *preprint* (2016). arXiv: 1612.00222 [cs.AI].

[81] Z. Que et al. "LL-GNN: Low Latency Graph Neural Networks on FPGAs for High Energy Physics". In: *ACM Trans. Embed. Comput. Syst.* 23.2 (2024). DOI: 10.1145/3640464.

[82] A. Elabd et al. "Graph Neural Networks for Charged Particle Tracking on FPGAs". In: *Front. Big Data* 5 (2022), p. 828666. DOI: 10.3389/fdata.2022.828666.

[83] Z. Que et al. "JEDI-linear: Fast and Efficient Graph Neural Networks for Jet Tagging on FPGAs". In: *preprint* (2025). arXiv: 2508.15468 [hep-ex]. URL: https://arxiv.org/abs/2508.15468.

[84] A. F. Agarap. "Deep Learning using Rectified Linear Units (ReLU)". In: *preprint* (2019). arXiv: 1803.08375 [cs.NE].

[85] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *preprint* (2017). arXiv: 1412.6980 [cs.LG].

[86] Belle II collaboration. *Strategic accelerator design (SAD)*. http://acc-physics.kek.jp/SAD. Accessed: 2026-02-05.

[87] T. Uchida et al. "Readout Electronics for the Central Drift Chamber of the Belle-II Detector". In: *IEEE Trans. Nucl. Sci.* 62.4 (2015), pp. 1741–1746. DOI: 10.1109/TNS.2015.2435747.

[88] E. B. Wilson. "Probable Inference, the Law of Succession, and Statistical Inference". In: *J. Am. Statist. Assoc.* 22.158 (1927), pp. 209–212. DOI: 10.1080/01621459.1927.10502953.

[89] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 12 (2011), pp. 2825–2830.

[90] C. Baskin et al. “UNIQ: Uniform Noise Injection for Non-Uniform Quantization of Neural Networks”. In: *ACM Trans. Comput. Syst.* 37.1–4 (2021). DOI: 10.1145/3444943.

[91] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *preprint* (2015). arXiv: 1502.03167 [cs.LG].

[92] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *preprint* (2019). arXiv: 1912.01703 [cs.LG].

[93] T.-Y. Lin et al. “Focal Loss for Dense Object Detection”. In: *preprint* (2018). arXiv: 1708.02002 [cs.CV].

[94] Y. Cui et al. “Class-Balanced Loss Based on Effective Number of Samples”. In: *preprint* (2019). arXiv: 1901.05555 [cs.CV].

[95] S. S. Mohseni Salehi, D. Erdogmus, and A. Gholipour. “Tversky loss function for image segmentation using 3D fully convolutional deep networks”. In: *preprint* (2017). arXiv: 1706.05721 [cs.CV].

[96] H. Kervadec and M. de Bruijne. “On the dice loss gradient and the ways to mimic it”. In: *preprint* (2023). arXiv: 2304.04319 [cs.CV].

[97] M. Neu et al. “Real-Time Graph Building on FPGAs for Machine Learning Trigger Applications in Particle Physics”. In: *Comput. Softw. Big Sci.* 8.1 (2024), p. 8. DOI: 10.1007/s41781-024-00117-0. arXiv: 2307.07289 [hep-ex].

[98] M. Fey and J. E. Lenssen. “Fast Graph Representation Learning with PyTorch Geometric”. In: *preprint* (2019). arXiv: 1903.02428 [cs.LG].

[99] G. Franco, A. Pappalardo, and N. J Fraser. *Xilinx/brevitas*. 2025. DOI: 10.5281/zenodo.3333552. URL: https://doi.org/10.5281/zenodo.3333552.

[100] C. N. Coelho et al. “Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors”. In: *Nature Mach. Intell.* 3 (2021), pp. 675–686. DOI: 10.1038/s42256-021-00356-5. arXiv: 2006.10159 [physics.ins-det].

[101] J. Bai et al. *ONNX: Open Neural Network Exchange*. 2019. URL: https://github.com/onnx/onnx.

[102] H. Cheng, M. Zhang, and J. Q. Shi. “A Survey on Deep Neural Network Pruning-Taxonomy, Comparison, Analysis, and Recommendations”. In: *preprint* (2024). arXiv: 2308.06767 [cs.LG].

[103] V. Natesh and H. T. Kung. “PQS (Prune, Quantize, and Sort): Low-Bitwidth Accumulation of Dot Products in Neural Network Computations”. In: *preprint* (2025). arXiv: 2504.09064 [cs.LG].

[104] Arm Limited. *AMBA AXI-Stream Protocol Specification*. IHI 0051B. 2021. URL: https://developer.arm.com/documentation/ihi0051/latest/.

[105] J. Bachrach et al. “Chisel: Constructing Hardware in a Scala Embedded Language”. In: *Proceedings of the 49th Annual Design Automation Conference*. 2012, pp. 1216–1225. DOI: 10.1145/2228360.2228584.

[106] AMD. *Vitis Unified Software Platform*. https://www.amd.com/de/products/software/adaptive-socs-and-fpgas/vitis.html. Version 2024.2, accessed 2025-10-28. 2025.

[107] M. Neu et al. “Real-Time Graph-based Point Cloud Networks on FPGAs via Stall-Free Deep Pipelining”. In: *preprint* (2025). arXiv: 2507.05099 [eess.SP].

[108] FastML Team. *fastmachinelearning/hls4ml*. Version v0.8.1. 2023. DOI: 10.5281/zenodo.1201549. URL: https://github.com/fastmachinelearning/hls4ml.

[109] AMD. *Vitis Unified Software Platform*. https://www.amd.com/de/products/software/adaptive-socs-and-fpgas/vivado.html. Version 2024.2, accessed 2025-10-28. 2025.

[110] S. Hodgson et al. *cocotb: Python-based chip (RTL) verification*. https://github.com/cocotb/cocotb. Accessed: 2025-05-13. 2025.

[111] AMD. *ModelSim HDL simulator*. https://eda.sw.siemens.com/en-US/ic/modelsim/. Version 2023.4, accessed 2025-05-13. 2025.

[112] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *preprint* (2017). arXiv: 1609.02907 [cs.LG]. URL: https://arxiv.org/abs/1609.02907.

[113] Kalyan Cherukuri and Aarav Lala. “Low-Rank Matrix Approximation for Neural Network Compression”. In: *preprint* (2025). arXiv: 2504.20078 [cs.LG]. URL: https://arxiv.org/abs/2504.20078.

[114] Abdolmaged Alkhulaifi, Fahad Alsahli, and Irfan Ahmad. “Knowledge distillation in deep learning and its applications”. In: *PeerJ Computer Science* 7 (Apr. 2021), e474. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.474. URL: http://dx.doi.org/10.7717/peerj-cs.474.

[115] Chang Sun et al. “HGQ: High Granularity Quantization for Real-time Neural Networks on FPGAs”. In: *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. ACM, Feb. 2026, pp. 79–91. DOI: 10.1145/3748173.3779200. URL: http://dx.doi.org/10.1145/3748173.3779200.

[116] Marta Andronic and George A. Constantinides. “NeuraLUT-Assemble: Hardware-aware Assembling of Sub-Neural Networks for Efficient LUT Inference”. In: *preprint* (2025). arXiv: 2504.00592 [cs.LG]. URL: https://arxiv.org/abs/2504.00592.

[117] Belle II Collaboration. *Belle II Grid and Computing: Sustainability at KEK*. https://indico.belle2.org/event/16060/contributions/100544/attachments/37075/55027/251002-b2gm-sustain-kek.pdf. Accessed: 2026-03-25. 2025.

[118] D. J. Lange. “The EvtGen particle decay simulation package”. In: *Nucl. Instrum. Methods Phys. Res. A* 462.1-2 (2001), pp. 152–155. DOI: 10.1016/S0168-9002(01)00089-4.

[119] S. Agostinelli et al. "GEANT4—a simulation toolkit". In: *Nucl. Instrum. Methods Phys. Res. A* 506 (2003), p. 250. DOI: 10.1016/S0168-9002(03)01368-8.

[120] S. Jadach, B. F. L. Ward, and Z. Was. "The precision Monte Carlo event generator KK for two-fermion final states in e^+e^- collisions". In: *Comput. Phys. Commun.* 130 (2000), p. 260. DOI: 10.1016/S0010-4655(00)00048-5.

[121] T. Sjöstrand et al. "An introduction to PYTHIA 8.2". In: *Comput. Phys. Commun.* 191 (2015), pp. 159–177. DOI: 10.1016/j.cpc.2015.01.024.