

# Automating L1 Trigger Performance Monitoring at Belle II

Arvid Kammann

Bachelor's Thesis

27 September 2024

Institute of Experimental Particle Physics (ETP)

Reviewer:	Prof. Dr. Torben Ferber
Second Reviewer:	Dr. Giacomo De Pietro
Advisor:	Patrick Ecker

Editing time: 14 May 2024 – 27 September 2024



# Automatisches Monitoring der L1-Trigger-Qualität bei Belle II

Arvid Kammann

Bachelorarbeit

27. September 2024

Institut für Experimentelle Teilchenphysik (ETP)

Referent: Prof. Dr. Torben Ferber  
Korreferent: Dr. Giacomo De Pietro  
Betreuer: Patrick Ecker

Bearbeitungszeit: 14. Mai 2024 – 27. September 2024



---

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

**Karlsruhe, 27. September 2024**

.....  
**(Arvid Kammann)**



# Preface

This thesis incorporates the use of Artificial Intelligence (AI) tools to help with grammatical improvement of text, and program code creation.

Grammarly\* is utilised throughout the thesis for spell and grammar checks. I have approved all suggested changes.

GitHub Copilot<sup>†</sup> is used to aid the development of Python code by automatically making suggestions inside the code editor to complete predictable code chunks and to find errors in my code.

GitHub Copilot and ChatGPT<sup>‡</sup> are used to aid the development of HTML code to create a website displaying the results (Ch. 5) which does not constitute the core scientific work of this thesis.

I have approved and tested all suggestions to provide robust and reliable results.

The work of this thesis adds a feature to the already existing Validation Interface for the Belle II Experiment (VIBE) [1] by Patrick Ecker. This is done within the Belle II Analysis Software Framework (basf2) [2].

The data files and Monte Carlo (MC) simulation files are taken and produced by the Belle II collaboration and downloaded by Dr. Giacomo De Pietro.

When plotting the trigger line efficiencies (Ch. 3), the uncertainties are calculated using Patrick Ecker's *DM-Analysis-Tools* [3] and the fit is done using the `optimize.curve_fit` function from SciPy [4].

The histograms in Ch. 4 are created using a Matplotlib style sheet based on the Belle II Plotting Style Guide [5] provided to me by Greta Heine.

The website displaying the results (Ch. 5) is built together with Simon Weber in order to use consistent design choices between our two projects.

Dr. Giacomo De Pietro and Patrick Ecker proof-read this thesis and suggested improvements of which many are included in the final version.

Prof. Dr. Torben Ferber, Dr. Giacomo De Pietro and Patrick Ecker proposed the studies conducted in this thesis.

I want to thank them and the entire Belle II group at ETP for welcoming me into their group and for their support during my time with them.

---

\*Grammarly Inc: Grammarly. <https://www.grammarly.com/>

<sup>†</sup>GitHub and OpenAI: Copilot. <https://github.com/features/copilot/>

<sup>‡</sup>OpenAI: ChatGPT. <https://chatgpt.com>



# Contents

<b>Preface</b>	<b>VII</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. The Belle II Experiment</b>	<b>3</b>
2.1. SuperKEKB Collider . . . . .	3
2.2. Belle II Detector . . . . .	3
2.3. Belle II Trigger System . . . . .	5
2.4. Belle II Analysis Software Framework . . . . .	6
<b>3. Trigger Line Efficiency Monitoring</b>	<b>7</b>
3.1. Trigger Line Efficiencies . . . . .	7
3.2. Turn-on Curves . . . . .	8
3.3. Efficiency of Neural 3D CDC Track Triggers . . . . .	11
3.4. Discrepancy Between Data and Simulation . . . . .	14
<b>4. Run-Dependent L1 Efficiency Monitoring</b>	<b>17</b>
4.1. Dataset . . . . .	17
4.2. Analysis . . . . .	17
4.2.1. hie Performance Monitoring . . . . .	18
4.2.2. fyo Performance Monitoring . . . . .	22
<b>5. Displaying the Results on a Website</b>	<b>25</b>
<b>6. Summary and Outlook</b>	<b>29</b>
<b>A. Appendix</b>	<b>31</b>
A.1. Additional Results of the Monitoring . . . . .	31



# 1. Introduction

The Standard Model (SM) of particle physics is the most successful description of nature on the smallest known scale. While it predicts the effects of three of the four fundamental forces with staggering accuracy, cosmological observations show the existence of Dark Matter (DM) which does not appear in the SM. Theorists have proposed many different models covering a wide mass range, but no experiments were able to verify any of these theoretical predictions. The search for DM is one of the most pressing pursuits in modern particle physics. It is performed by a vast number of experiments using entirely different approaches.

Collider experiments such as Belle II look for very rare processes where ordinary matter couples non-gravitationally to DM. To achieve meaningful statistics even for these events, Belle II aims for record-breaking luminosities which in turn leads to an extremely high amount of measured data of which not all can be stored due to limitations in computing power and storage capacity.

The Belle II trigger system selects which events to store and which to disregard permanently. It needs to come to a decision in a very short amount of time because, during its selection, more events are still measured and need to be judged by the system. Its performance is crucial for the experiment as it influences the data available for analyses.

Running the experiment with a faulty trigger system wastes money and resources and leads to unnecessary energy consumption. This calls for a thorough monitoring process.

The process needs to be automated, so it can be run over all data-taking periods and investigate all aspects of the trigger system without spending the working hours it would otherwise take. Additionally, its results need to be displayed in an easily accessible way to quickly bring attention to possible errors.

This thesis presents an automated monitoring of the Belle II trigger system. In Ch. 2, the collider and the detector as well as the trigger system and its software framework are introduced. Chapter 3 describes trigger line efficiencies and how they are monitored which lays the groundwork for the automated monitoring (Ch. 4). To make the results easily accessible for the collaboration, a website is created which is presented in Ch. 5. Chapter 6 gives a summary of the obtained results and points at possible subsequent studies.



## 2. The Belle II Experiment

Belle II is a detector at the particle collider SuperKEKB at the KEK complex in Tsukuba, Japan.

### 2.1. SuperKEKB Collider

SuperKEKB is an asymmetric, circular  $e^+e^-$  collider with a centre-of-mass (COM) energy of 10.580 GeV [6]. Before entering the collider rings, the electrons and positrons are accelerated in the injector linear accelerator (LINAC) to an energy of 7 GeV and 4 GeV, respectively. From there, they are sent into two different rings until they collide at the interaction point (IP) inside the Belle II detector. A sketch of the accelerator complex can be seen in Fig. 2.1.

The COM energy is chosen to match the mass of the  $\Upsilon(4S)$  resonance<sup>1</sup> which mainly (> 96%) decays into two  $B$  mesons [7]. For this reason, SuperKEKB produces a high number of  $B$  mesons and is called a  $B$ -factory.

The asymmetry in energy of the electrons and positrons provides a boost to all emerging particles. This helps detect particles with a small lifetime in the laboratory frame due to relativistic time dilation.

### 2.2. Belle II Detector

The Belle II detector is positioned at the Tsukuba straight section at SuperKEKB right at the IP. It contains multiple layers of detectors that are built in cylindrical shape around the beam pipe. An overview of Belle II can be seen in Fig. 2.2.

The innermost layer is the Vertex Detector (VXD) which consists of the Pixel Detector (PXD) and the Silicon Vertex Detectors (SVD). It measures the position of particles with very high precision which is crucial to reconstruct the particles' trajectories (often called "tracks") near the IP.

The next layer around the VXD is the Central Drift Chamber (CDC) through which the particles propagate revealing their charge and momentum. Outside of the CDC, there are the Aerogel Ring-imaging Cherenkov (ARICH) and Time-of-Propagation (TOP) detectors, both delivering data useful for particle identification.

Finally, there are the Electromagnetic Calorimeter (ECL) and the  $K_L^0$  and Muon Detector (KLM). The ECL measures energy depositions of leptons, hadrons and photons. The KLM

---

<sup>1</sup>Some data-taking periods are intentionally run "off-resonance" meaning the COM energy is not equal to the mass of the  $\Upsilon(4S)$  resonance. This is discussed in Sec. 4.1.

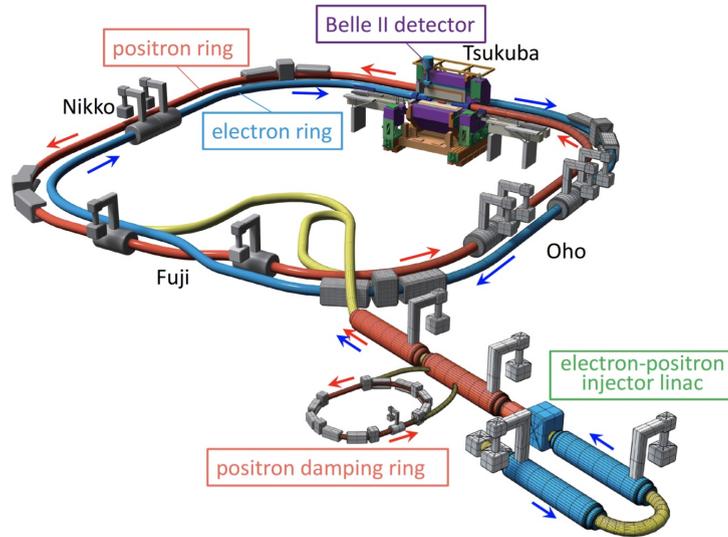


Figure 2.1.: An overview of the SuperKEKB accelerator. The  $e^+e^-$  injector linear accelerator is shown on the bottom right. From there, the electrons travel in the high-energy ring (HER) and the positrons in the low-energy ring (LER) (depicted in blue and red, respectively). The Belle II detector is positioned at the Tsukuba straight section. Image taken from [6].

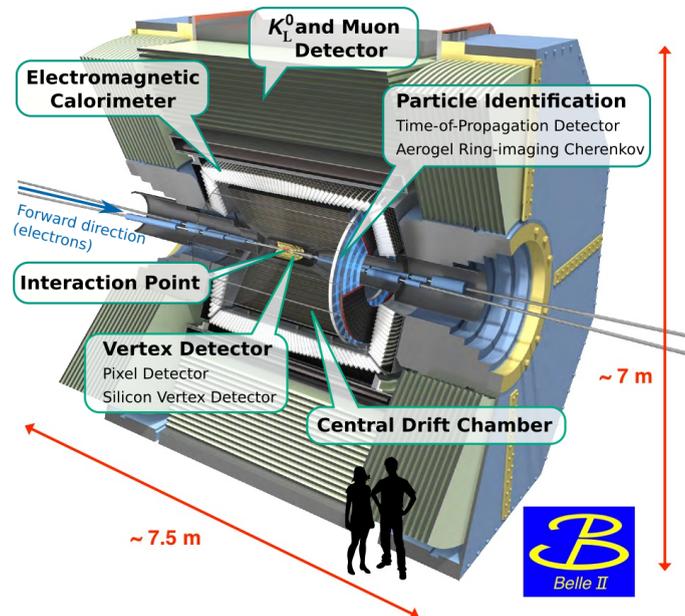


Figure 2.2.: Overview of the Belle II detector. It is built asymmetrically to match the forward direction given by the direction of the electrons which are higher-energy than the positrons. Image taken from [8], annotated by me.

detects and identifies  $K_L^0$  and muons behind the ECL and a 1.5 T superconducting magnet. The coordinate system used for Belle II analyses is defined to have its origin at the IP and the  $z$ -axis pointing in the forward direction of the electrons. The  $x$ -axis points horizontally away from the centre of the collider ring. Finally, the  $y$ -axis is defined to point vertically upwards creating a right-handed coordinate system. Because of the cylindrical shape of the detector, it is often natural to use cylindrical coordinates. Therefore, the azimuth  $\phi$  is defined in the  $xy$ -plane ranging from  $-180^\circ$  to  $180^\circ$  where the  $x$ -axis fixes  $\phi = 0$  [9]. Sometimes, spherical coordinates come in handy, as well. In this case, the origin is defined at the IP and the polar angle  $\theta$  ranges from 0 to  $180^\circ$  with  $\theta = 0$  being the forward direction of the electrons.

## 2.3. Belle II Trigger System

Computing power and storage capacity limit the amount of measured data Belle II can store. Events that happen quite often (mostly Bhabha scattering events) would take away resources needed to save events of more interest. Therefore, all measured events need to be somehow assessed as to how interesting they are for analyses. This is where the Belle II trigger system comes in. It assigns a binary value to every event depending on whether or not the event is worth storing, 1 ("triggered") meaning the event is stored and 0 ("not triggered") meaning it is permanently deleted.

The Belle II trigger system consists of two parts: the Level 1 trigger (TRG), sometimes just called L1, and the high level trigger (HLT).

### Level 1 Trigger

When selecting which events should be stored, time is a limiting factor. The system needs to temporarily store all events that happen during the selection to have the chosen events available once the selection is finished. Therefore, it does not have time to send all data to an external computing centre and do thorough calculations there.

Instead TRG uses field-programmable gate arrays (FPGAs) that sit right at the detector (this is why it is called Level 1 trigger). To be exact, these FPGAs are part of the subdetectors CDC, ECL and KLM. They create input trigger bits which are binary signals corresponding to certain signatures which either are of interest for analysis or considered unnecessary to be stored. For example, the `bha_veto` trigger bit sends "1" if the event is Bhabha scattering and "0" if it is not.

Output triggers (final trigger decision logic, FTDL) combine these trigger bits to make trigger decisions. Many of them use the negated `bha_veto` decision to filter out Bhabha scattering.

The FTDL is then prescaled to produce the prescale and mask (PSNM) trigger decision. Prescaling means that certain FTDL triggers do not store all events they detect but only every  $n$ th, where  $n$  is called the prescale factor. For example, the prescale factor of the trigger line corresponding to Bhabha events is 100, so Belle II only stores 1 out of 100 Bhabha events [10].

If at least one of the PSNM triggers gives a positive result for a certain event, it is then processed further by the HLT. TRG turns an input rate around 45 kHz (always depending on current luminosity) into an output rate of no more than 30 kHz. Figure 2.3 shows the TRG logic.

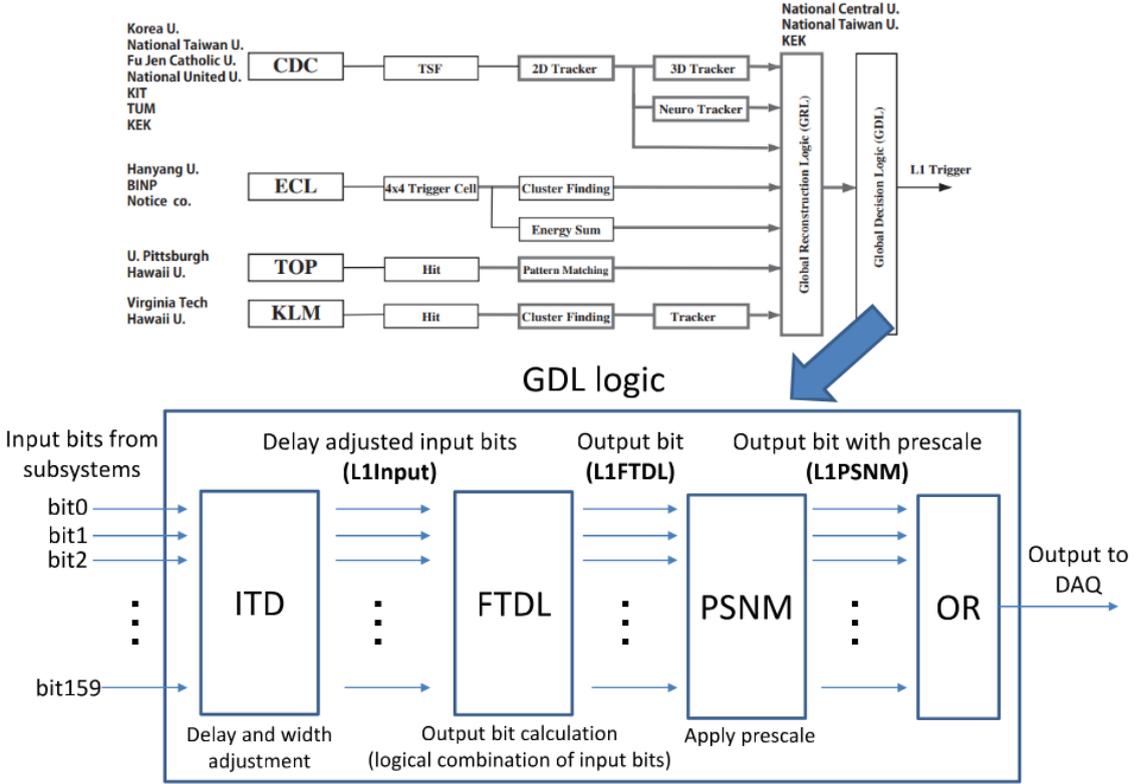


Figure 2.3.: The logic of TRG. The subdetectors form the input trigger bit decisions. These decisions are processed by the global decision logic (GDL) via the input trigger delay module (ITD) forming output bits (FTDL) which are then prescaled (PSNM). If at least one PSNM trigger bit returns 1 ("is triggered"), the event is given to data acquisition (DAQ). Image taken from [11].

## High Level Trigger

The HLT consists of about 10 000 CPU cores which immediately reconstruct each triggered event. Depending on the results of the reconstruction, it then applies a further selection and stores the remaining events on actual hard drives. The maximum output rate of the HLT is 10 kHz.

## 2.4. Belle II Analysis Software Framework

The Belle II Analysis Software Framework (basf2, [2] [12]) is C++ and python software which can deal with a very high amount of data to perform a wide range of tasks. It is used both for offline analyses by individual collaboration members and large-scale jobs by systems such as the HLT. It handles tasks ranging from event generation to track reconstruction and physics analysis.

For the full documentation of the release `light-2403-persian`, which is used for offline analysis throughout this thesis, see [13].

## 3. Trigger Line Efficiency Monitoring

In this chapter, I investigate trigger line efficiencies for single experiment runs to introduce important concepts which are necessary for the monitoring process discussed in Ch. 4.

### 3.1. Trigger Line Efficiencies

The trigger system needs to function well for the experiment to succeed as its decisions determine which data are stored and end up being used in analyses and more importantly which data are not stored and therefore permanently discarded. To monitor the trigger system, I study the efficiency of one trigger line at a time, meaning the percentage of events this specific trigger line decided to store.

Before starting to monitor trigger line efficiencies, a proper definition is needed. The naive way would be to calculate the efficiency  $\epsilon_{\text{trg}}$  as

$$\epsilon_{\text{trg}} = \frac{N(\text{trg})}{N_{\text{all}}} , \quad (3.1)$$

where  $N(\text{trg})$  is the number of events where the trigger line is triggered, and  $N_{\text{all}}$  is the number of all events. However, when studying measured data,  $N_{\text{all}}$  is unknown because, by design, only events which triggered at least one trigger line are registered.

This problem is easily solved by defining one or multiple reference lines. Then, the efficiency is calculated as

$$\epsilon_{\text{trg}} = \frac{N(\text{trg AND ref})}{N(\text{ref})} \quad (3.2)$$

with  $\text{trg}$  being the trigger line decision of interest and  $\text{ref} = (\text{ref}_1 \text{ OR } \text{ref}_2 \text{ OR } \dots \text{ OR } \text{ref}_n)$  being the OR connection of all  $n$  reference lines.

Note that FTDL and PSNM outputs are combinations of several input trigger bits (see [10] for the full trigger bit table). Different output trigger lines can therefore be correlated when using the the same input bit to form a decision. Correlated trigger lines should not be used as reference lines since this will naturally bias the sample used to study the trigger line. In this thesis, I only analyse ECL and CDC triggers. Usually, it is a good idea to take an ECL line as a reference to a CDC line and vice versa. Still, there are input bits used by both ECL and CDC trigger lines, so one needs to be careful.

Table 3.1.: All selection criteria used throughout this thesis.

Final State Particle	Selection Criteria
$\mu^\pm$	$ dr  < 0.5 \text{ cm}$ $ dz  < 2 \text{ cm}$ $\text{thetaInCDCAcceptance} (17^\circ \leq \theta \leq 150^\circ)$ $\text{muonID\_noSVD} > 0.9$
$\gamma$	$\text{thetaInECLAcceptance} = 2 (32.2^\circ \leq \theta \leq 128.7^\circ)$
ECL Energy Depositions	$4 \leq \text{clusterThetaID} \leq 58$ $\text{clusterUncorrE} > 0.1 \text{ GeV}$ $\text{clusterTiming} < 200 \text{ ns}$
Additionally	$\text{nCleanedTracks}(\mu^\pm \text{ selection}) = 2$ $E_{\text{event}} < 11 \text{ GeV}$

### 3.2. Turn-on Curves

Many trigger lines are designed to be triggered if a specific variable exceeds a certain value. For example, the high energy trigger (hie) uses the total energy  $E_{\text{tot}}$  that is deposited into a certain area of the ECL (approximately  $22^\circ \leq \theta \leq 128^\circ$ ) and is triggered when it exceeds 1 GeV. To monitor this, I bin the energy deposition and plot the efficiency of all events in each bin using the `DM-Analysis-Tools` [3] to calculate uncertainties. This can be seen in Fig. 3.1. In this plot and throughout the thesis, I study the process  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  with the selection criteria listed in Tab. 3.1 (for the definition of each variable, see section 7.3. in [13]).

I select two muons in the acceptance region of the CDC ( $17^\circ \leq \theta \leq 150^\circ$ ) with a vertex close to the IP and a photon within the barrel region of the ECL ( $32.2^\circ \leq \theta \leq 128.7^\circ$ ). Particles in the ECL are only considered if they are measured within a certain angular region and if they exceed the energy threshold of 100 MeV. The `clusterTiming` selection filters out cases where the cluster is hit too long after a reference time  $t_0$  as this is mostly caused by beam background.

Since SuperKEKB only supplies the  $e^+e^-$  pair with a total energy of 11 GeV, any event with a higher energy is disregarded.

Ideally, the distribution of Fig. 3.1 should follow a step function, being 0 for all values below what is called the turn-on point, and 1 for all above. For hie, this means that the efficiency of events with  $E_{\text{tot}} < 1 \text{ GeV}$  should be 0, and for  $E_{\text{tot}} \geq 1 \text{ GeV}$ , it should approach 1 with a very sharp transition.

In practice, however, it takes off a bit before the turn-on point and reaches the plateau a bit later. This is due to the fact that the high-level variable  $E_{\text{tot}}$  on the  $x$ -axis is only an approximation of the variable used by TRG on the lowest level. One difference is the resolution of the polar angle region which is considered. In the offline reconstruction, a selection on the polar angle of the highest-energy ECL crystal is used. The crystals around the highest-energy one still measure some deposition which will be recognised by TRG if it is

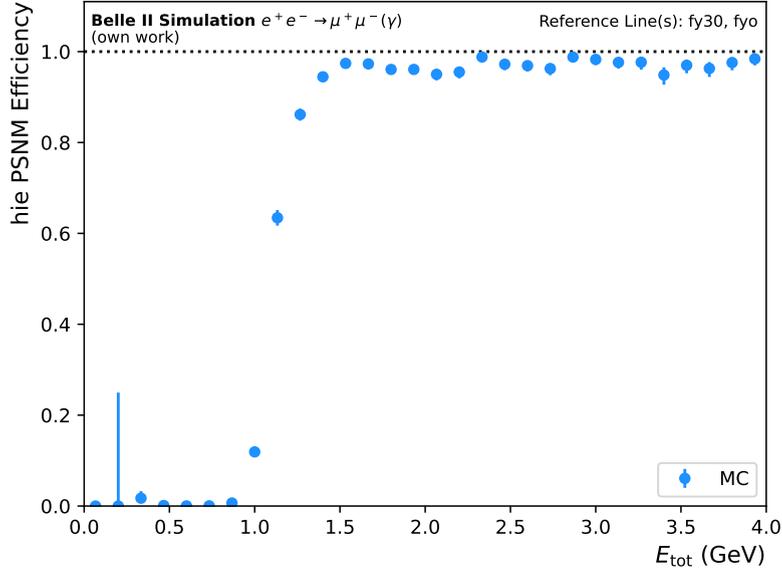


Figure 3.1.: PSNM efficiency of hie for different values of the total energy deposition in the ECL. Simulation of experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

within the area hie uses (even if the highest-energy crystal is outside this area). This leads to a non-vanishing hie efficiency for  $E_{\text{tot}}$  values slightly below 1 GeV. Additionally, the energy resolution and timing selection differ between the two because the offline reconstruction has more data available and defines a different reference point  $t_0$  than the low-level variable.

To monitor the turn-on behaviour, I define the fit function

$$f(x, m, s, k, y_0) = \left( \frac{1}{2} \cdot \operatorname{erf}(s \cdot (x - m)) + \frac{1}{2} \right) \cdot (k - y_0) + y_0 \quad (3.3)$$

where the error function erf is given by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (3.4)$$

with the mathematical constants  $\pi$  and  $e$ .

In addition to the variable  $x$ , the fit function in Eq. 3.3 depends on the fit parameters, namely

- the turn-on point  $m$ ,
- the steepness  $s$ ,
- the asymptotic value  $k$  and
- the starting value  $y_0$ .

In the limit  $s \rightarrow \infty$ ,  $k \rightarrow 1$  and  $y_0 \rightarrow 0$ , the fit function resembles the ideal step function. For most trigger lines, there is a known ideal turn-on point that is monitored by comparing

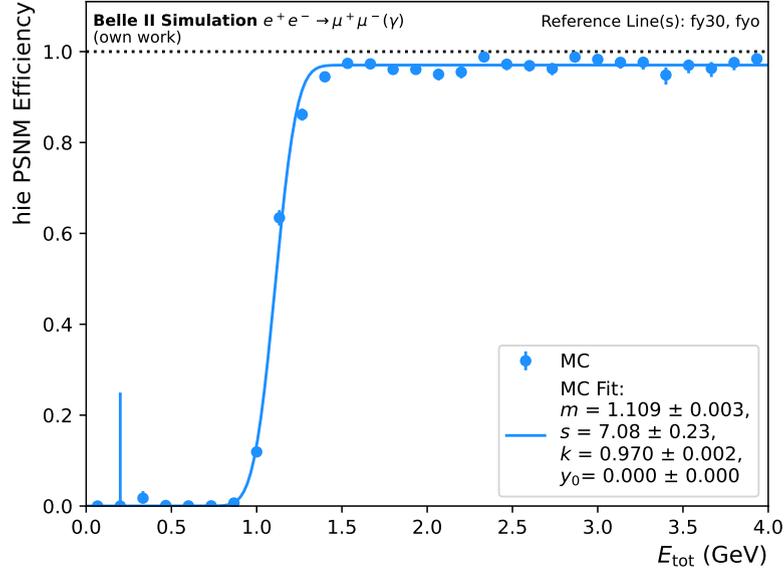


Figure 3.2.: PSNM efficiency of hie against the total energy in the ECL. Fitted with the fit function defined in Eq. 3.3. Simulation of experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

with the fit parameter  $m$ .  $k$  and  $y_0$  are important to monitor the actual performance of the trigger line as they are the true positive rate and false positive rate (for values far away from the turn-on point), respectively. The steepness parameter  $s$  shows how well the trigger line is performing around the turn-on point.

In Fig. 3.2, a fit is performed using the `optimize.curve_fit` function from SciPy [4] which relies on the `optimize.leastsq` method. This will be used for fitting throughout the thesis. The fit function has the problem of overestimating the steepness behind the turn-on point while underestimating the steepness in front of it. To tackle this problem, a double-sided fit can be performed. This is done by first using the fit function from Eq. 3.3, and then fixing the results of  $m$ ,  $k$  and  $y_0$  to define the double-sided fit function

$$f_{\text{DS}}(x, s_1, s_2) = \begin{cases} f(x, m, s_1, k, y_0) & x \leq m \\ f(x, m, s_2, k, y_0) & x > m \end{cases}, \quad (3.5)$$

where  $f$  is the symmetric fit function defined in Eq. 3.3.

The design of  $f_{\text{DS}}$  ensures it is continuous in  $x = m$ . The results of a double-sided fit are shown in Fig. 3.3. As expected, the result for  $s_1$  is higher than  $s$  from the fit in Fig. 3.2 and  $s_2$  is lower.

For the purpose of monitoring the turn-on curves, however, just one parameter that measures steepness suffices and a perfect fit is not necessary. Therefore, I only use the symmetric fit in the automated monitoring process described in Ch. 4.

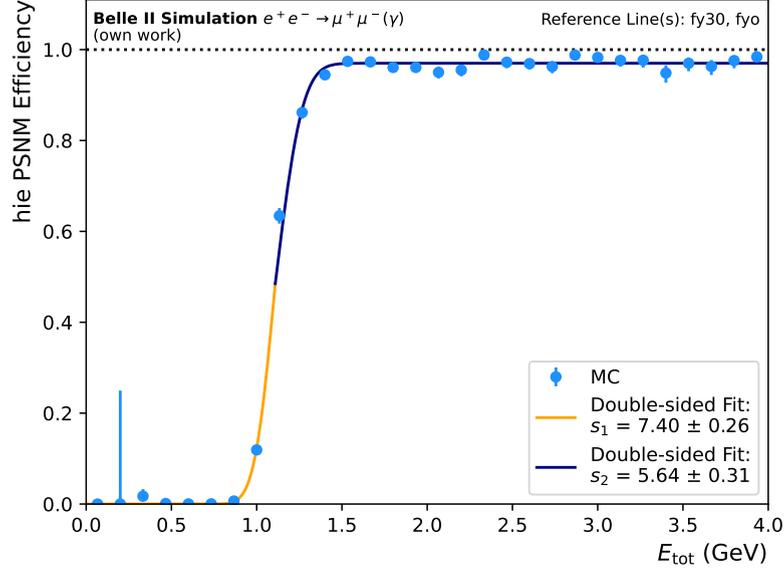


Figure 3.3.: PSNM efficiency of hie against the total energy in the ECL. Fitted with the fit function defined in Eq. 3.5. The values for  $m$ ,  $k$  and  $y_0$  are taken from the fit in Fig. 3.2. Simulation of experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

### 3.3. Efficiency of Neural 3D CDC Track Triggers

Now, I investigate neural 3D CDC track triggers, namely the trigger lines fy30 and fyo (the discussion also applies to their predecessors ff30 and ffo). They are designed to trigger if  $\Delta\phi_{\max}$  is higher than  $30^\circ$  and  $90^\circ$ , respectively, where  $\Delta\phi_{\max}$  is the maximum azimuthal opening angle between any two tracks of the same event.

As depicted in Fig. 3.4 and Fig. 3.5, their turn-on curves have a turn-on point at the values  $30^\circ$  and  $90^\circ$ , respectively, after which the efficiency stagnates until it rises again for the last few data points. This behaviour differs from the fit function from Sec. 3.2. Therefore, fyo and fy30 need different treatment.

One possibility, which I use for other trigger lines in Ch. 4, is to perform the fit once to get a rough estimate of the turn-on point parameter  $m$ . Then I zoom in around this calculated turn-on point and perform the fit again, disregarding points that are too far away from the turn-on point. However, for fy30 and fyo this procedure is not stable enough which is why in Ch. 4, I zoom in around the expected turn-on point of the trigger line. In Fig. 3.6 this is done for the trigger line fyo which has an expected turn-on point at  $90^\circ$ , so the fit ranges from  $45^\circ$  to  $135^\circ$  disregarding the events with a high opening angle.

Another possibility is to manually declare the troublemaking data points as outliers and disregard them for the fit (Fig. 3.7). To properly fit all data, I fit a second turn-on curve for the last data points. I do this by selecting a point where the first curve ends and the second begins. Then, I fit the first curve and take the value for  $k$  as the value for  $y_0$  for the second curve, determining only the remaining three fit parameters. This is done in Fig. 3.8. Note that all of these options succeed in producing a more accurate fit but need some

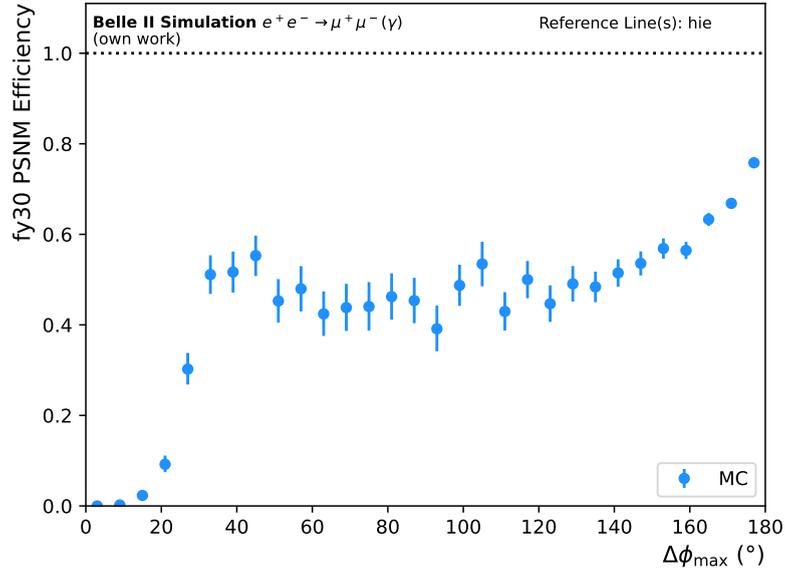


Figure 3.4.: PSNM efficiency of fy30 for different values of the maximal opening angle between two tracks. Simulation of experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

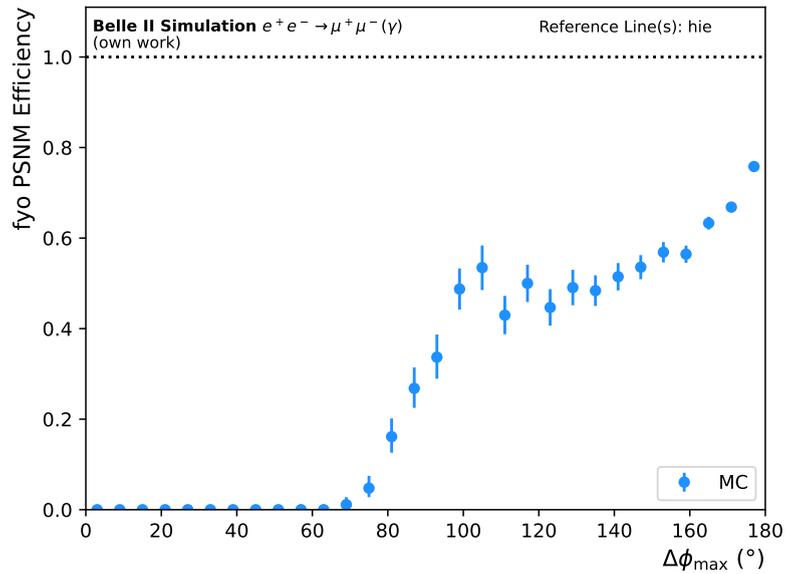


Figure 3.5.: PSNM efficiency of fyo for different values of the maximal opening angle between two tracks. Simulation of experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

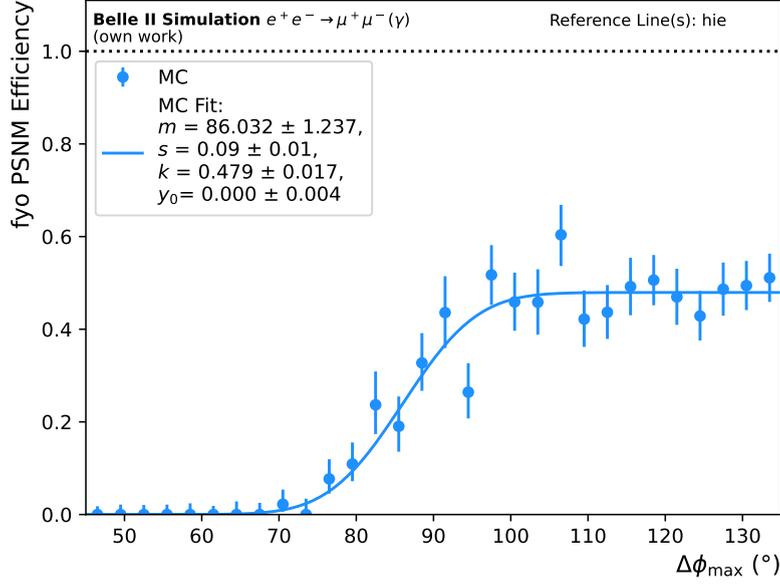


Figure 3.6.: PSNM fyo efficiency as a function of maximal opening angle between two tracks for a range of  $90^\circ$  around the expected turn-on point ( $90^\circ$ ). Standard fitting method from Sec. 3.2. Simulation from experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

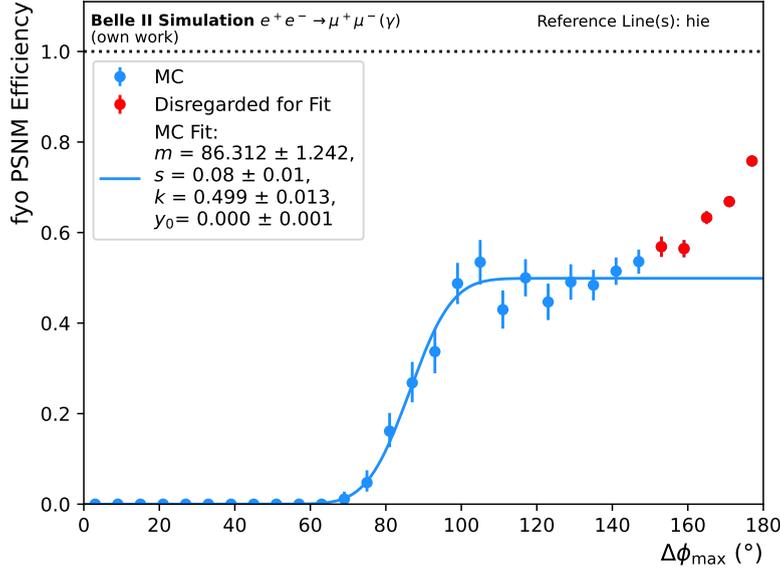


Figure 3.7.: PSNM fyo efficiency as a function of maximal opening angle between two tracks. The last five data points are manually disregarded for the fit. Simulation from experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

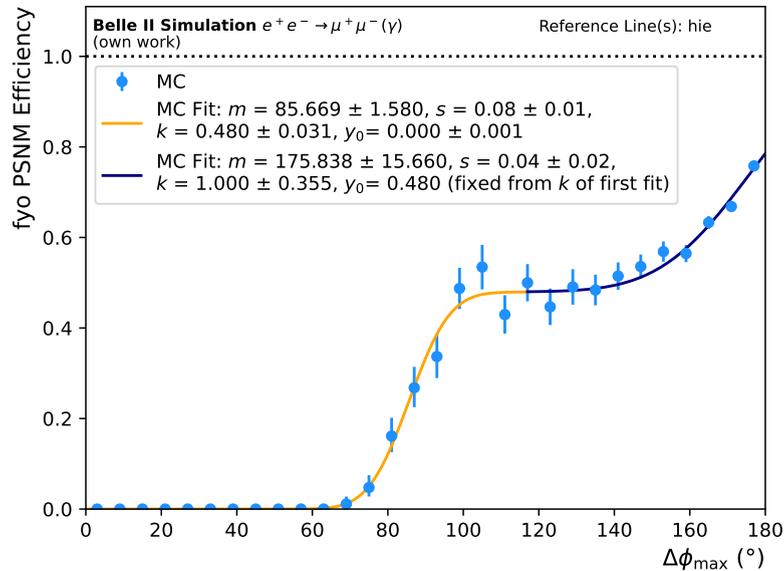


Figure 3.8.: PSNM fyo efficiency as a function of maximal opening angle between two tracks. Two different turn-on curves are fitted to the data points. Simulation from experiment 26, run 898 (only  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  final state).

manual selection (the data points I disregard, the point around which I "zoom in", and the point where I separate the two curves). Which data points to disregard and where to separate the curves, changes from run to run. Therefore, these two do not qualify for an automated monitoring of thousands of runs and I "zoom in" around the expected turn-on point in the monitoring (Ch. 4).

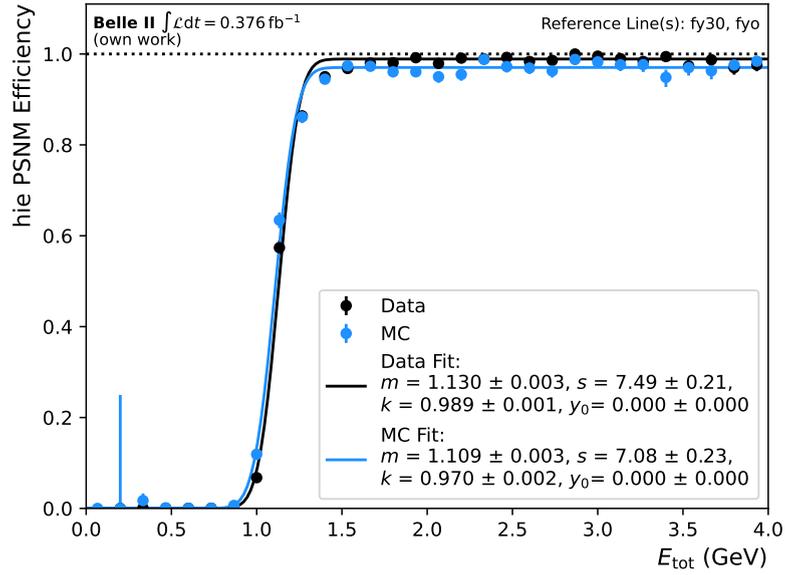
### 3.4. Discrepancy Between Data and Simulation

In modern particle physics, Monte Carlo (MC) simulation is used to make predictions and hint scientists at where to look for new physics. Naturally, one wants to compare data and MC and therefore also trigger decisions in the simulation are necessary. This is where trigger simulation (TSIM) comes in.

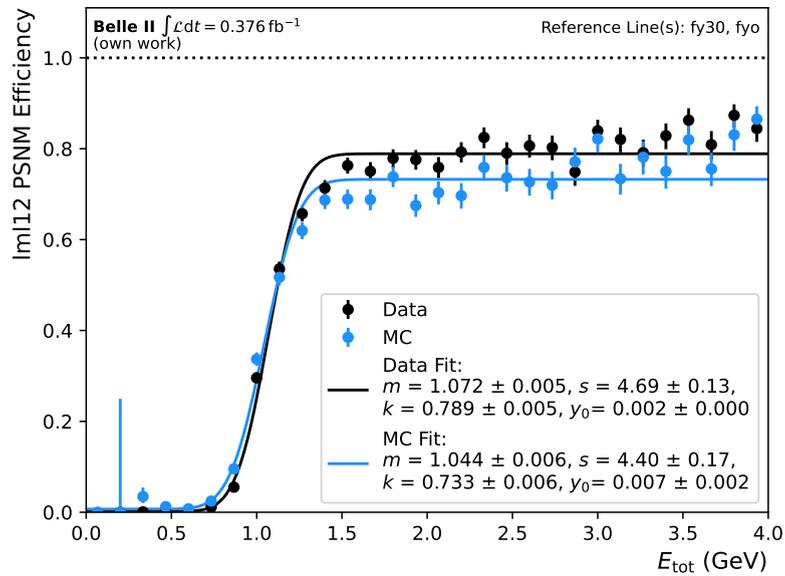
As the name suggests, TSIM simulates the L1 trigger system of Belle II. In this section, I investigate a discrepancy of TSIM and TRG for ECL trigger lines such as hie. Figure 3.9 shows comparisons between data and MC for hie and the low multiplicity (lml) trigger line lml12.

In both plots, the efficiency of data is a few percent higher than that of TSIM, which is an unexpected issue especially for higher energies as these ECL trigger lines should conceptually be simulated quite accurately. The discrepancy arises for all ECL trigger lines and has been studied internally by the collaboration. The error in the TSIM code has been found and will be fixed in an upcoming release.

This shows that monitoring trigger line efficiencies in both data and simulation is an important task for the collaboration. Issues like this can impact analyses if they go unnoticed for too long.



(a) hie trigger line



(b) lml12 trigger line

Figure 3.9.: Efficiency of ECL trigger lines as a function of the total energy that is deposited into the ECL for both data and simulation of experiment 26, run 898. Data show a higher asymptotic value by a few percent.



## 4. Run-Dependent L1 Efficiency Monitoring

During the lifetime of an experiment like Belle II, the collaboration keeps maintaining, changing and improving the experimental setup. For example, a part of a detector might break and need replacement, the experiment is performed at different energies or more particles are accelerated to reach a higher luminosity.

Every time a component of the detector is changed, it can have an impact on trigger efficiencies as the input trigger bits use data measured by many subdetectors. To find sources of errors and inaccuracies, the efficiencies need to be investigated for all data-taking periods.

### 4.1. Dataset

In this thesis, I only monitor those runs in the dataset (up until June 2022) which use a COM energy equal to the mass of the  $\Upsilon(4S)$  resonance. This excludes some data-taking periods which consist of energy scan runs and off-resonance runs. For all relevant experiments, the integrated luminosity  $\int \mathcal{L} dt$  used in this thesis can be found in Tab. 4.1.

All available samples of MC simulation are used to get the best possible description of collected data. However, due to selecting only the process  $e^+e^- \rightarrow \mu^+\mu^-(\gamma)$  (see Sec. 3.2 for the selection criteria), other final states only play a marginal role. The only other relevant contribution comes from the process  $e^+e^- \rightarrow e^+e^-\mu^+\mu^-$  when the electron and positron travel too closely to the beam pipe to be reconstructed.

### 4.2. Analysis

I now monitor the L1 efficiencies of all runs within a certain experiment, one single trigger line at a time. For that, I perform a fit of the turn-on curve (see Sec. 3.2) for every run

Table 4.1.: Integrated luminosity of all experiment runs used in this thesis. Determined using the `b2info-luminosity` tool from `basf2` [2].

Experiment	7	8	10	12	14	16
$\int \mathcal{L} dt$ (fb <sup>-1</sup> )	0.506	1.663	3.655	54.573	16.500	10.294
	17	18	20	22	24	26
	10.715	89.900	3.788	32.060	85.642	54.795

and analyse the results of the fit parameters. To get a better look at the performance of the trigger line around the turn-on point, I also perform an additional fit for a smaller range around the turn-on point that was determined with the full-range fit. This leads to a finer binning and a narrower fitting range in that area. Therefore, the parameters  $m$  and  $s$  (turn-on point and steepness) are determined more accurately than with the full-range fit. The parameters  $k$  and  $y_0$  are influenced mainly by data points in the tails of the curve. Therefore, they are determined more accurately without the additional close-range fit. The main work of this thesis is creating a mode in the Validation Interface for the Belle II Experiment (VIBE) [1] which can do this procedure automatically for any trigger line and any variable. The source code of my contribution can be found in [14]. As an example, I show the performance of the trigger line `hie` for experiment 22. Since the neural 3D CDC track triggers `fy30` and `fyo` call for a slightly different treatment (see Sec. 3.3), the `fyo` monitoring of experiment 22 is also shown. More results can be found in Sec A.1.

#### 4.2.1. `hie` Performance Monitoring

The  $x$ -axis of `hie` turn-on curves is the total energy deposition in the ECL. Its full range is chosen to be from 0 to 4 GeV where the upper limit is a compromise between a large range to have enough data points for the parameters  $k$  and  $y_0$  and having high statistics in every entry. The narrow range is then defined to reach from 0.5 GeV below the determined turn-on point to 0.5 GeV above it. Here, I use `fy30` and `fyo` as reference lines (for older experiment runs `ff30` and `ffo` are used).

The results of the automated monitoring are the fitting parameters of every run of the experiment, which are visualised by histograms. Figures 4.1a and 4.1b show histograms of the turn-on point parameter  $m$  for simulation of experiment 22 and data of experiment 22, respectively. As discussed above, the close-range fit is more accurate for this parameter as it uses more data points in the relevant region around the turn-on point. The figures show both fitting ranges for comparison.

As can be seen in both Fig. 4.1a and Fig. 4.1b, the close-range fit has a much lower standard deviation than the full-range fit. This is mainly due to outliers in the full-range fit that get corrected by fitting again in a region around the turn-on point. Note that the mean that is given in the figures should be understood as a quick cross-check of the procedure and to study differences between data and simulation. It is not an accurate measure for the average result as it includes outliers and does not weight the values with respect to the luminosity of the corresponding run.

To investigate how well MC simulation agrees with data, 2D histograms which show the correlation between the two come in handy. As can be seen in Fig. 4.1c, there is a correlation between data and simulation for the fit parameter  $m$  (at this point, I only use the fit results from the close-range fit) with most high-number bins being along the diagonal of the plot. This means that the run-dependent MC matches the data reasonably well for most runs.

I repeat the same procedure for the steepness parameter  $s$ . Again, I find a value for  $s$  for the first full-range fit and for the narrower fitting range around the detected turn-on point. Figures 4.2a and 4.2b show histograms of both ranges for simulation and data, respectively.

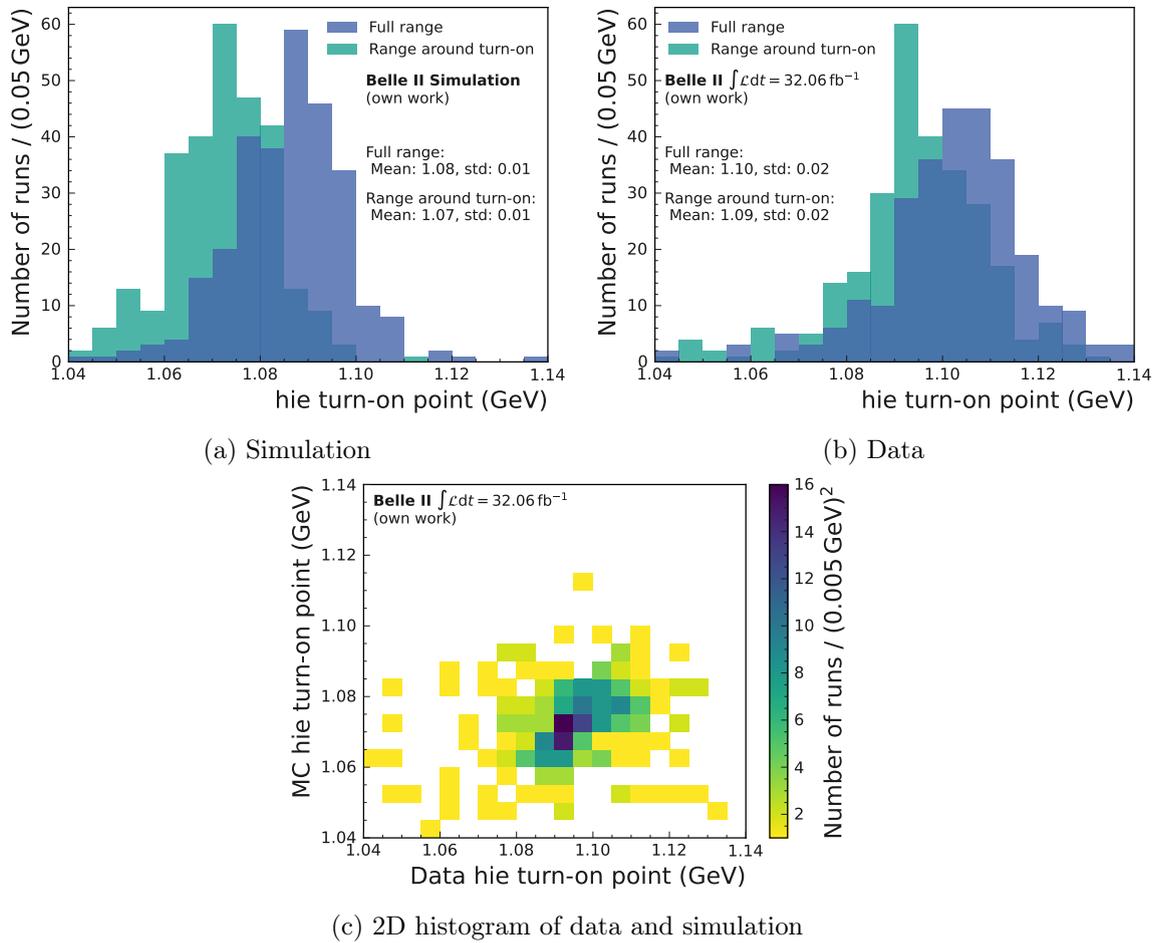


Figure 4.1.: Data-MC comparison of the turn-on point of hie. The 1D histograms show the distribution of turn-on points for simulation and data, respectively. The 2D histogram shows the correlation between the two.

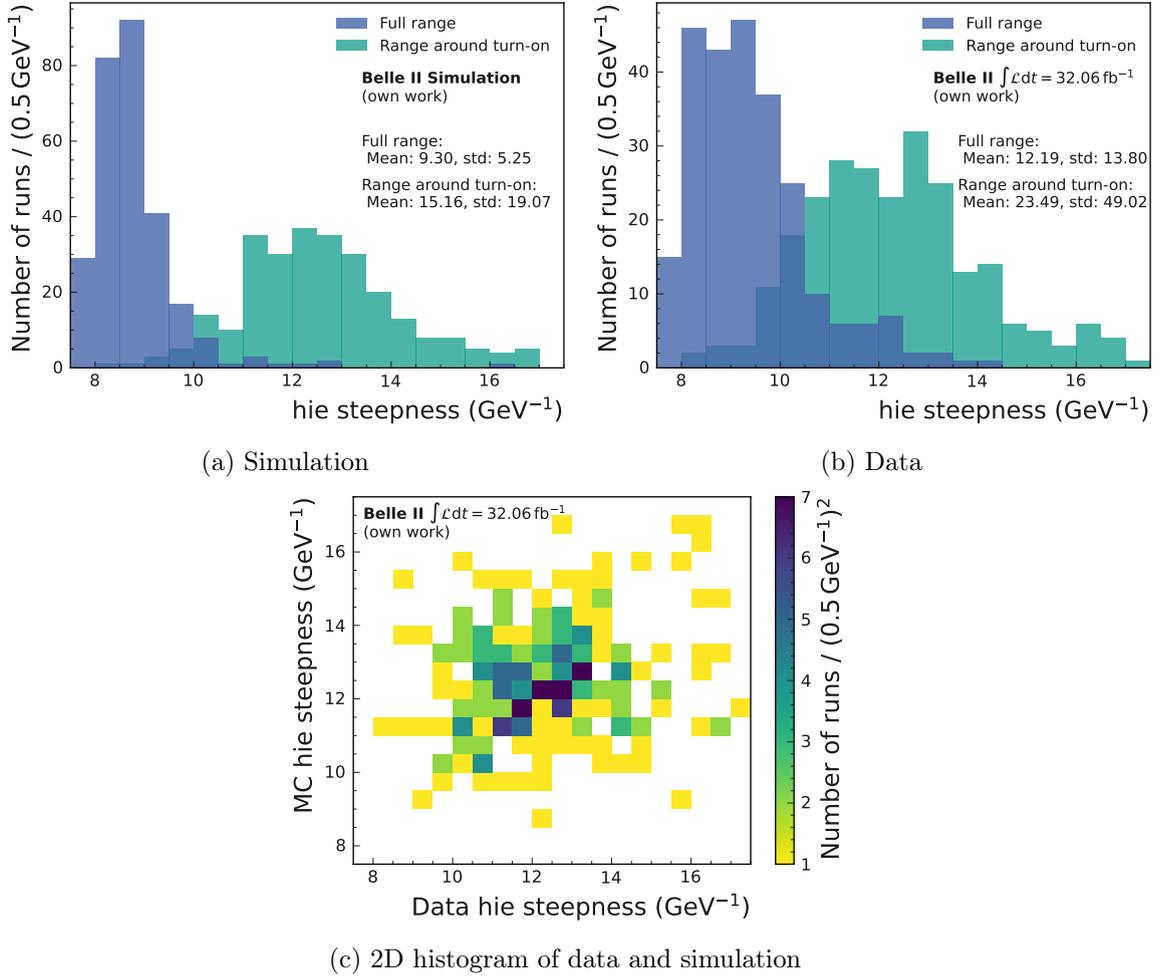


Figure 4.2.: Data-MC comparison of the steepness parameter of hie. The 1D histograms show the distribution of the parameter for simulation and data, respectively. The 2D histogram shows the correlation between the two.

Note that there is no upper limit for the parameter which means there are a few outliers very far away from the shown range. This leads to a high mean and standard deviation. The steepness depends on data points around the turn-on point. Therefore, I use the results of the narrow-range fit for the 2D histogram 4.2c showing the correlation between data and simulation. As with the turn-on point parameter  $m$ , most high-number bins live on the diagonal meaning the run-dependent MC matches the data well for these runs. However, there are many bins with one or two runs far away from the diagonal which is to be expected as  $s$  has a much higher standard deviation than  $m$ .

Next up is the fitting parameter for the plateau value of the efficiency  $k$ . As discussed in Sec. 3.4, simulation shows a slightly lower value than data which can be seen in Fig. 4.3a and Fig. 4.3b. Again, the 1D histograms show the results of both fitting ranges while the 2D histogram only shows the more accurate one which this time is the full-range fit (as the

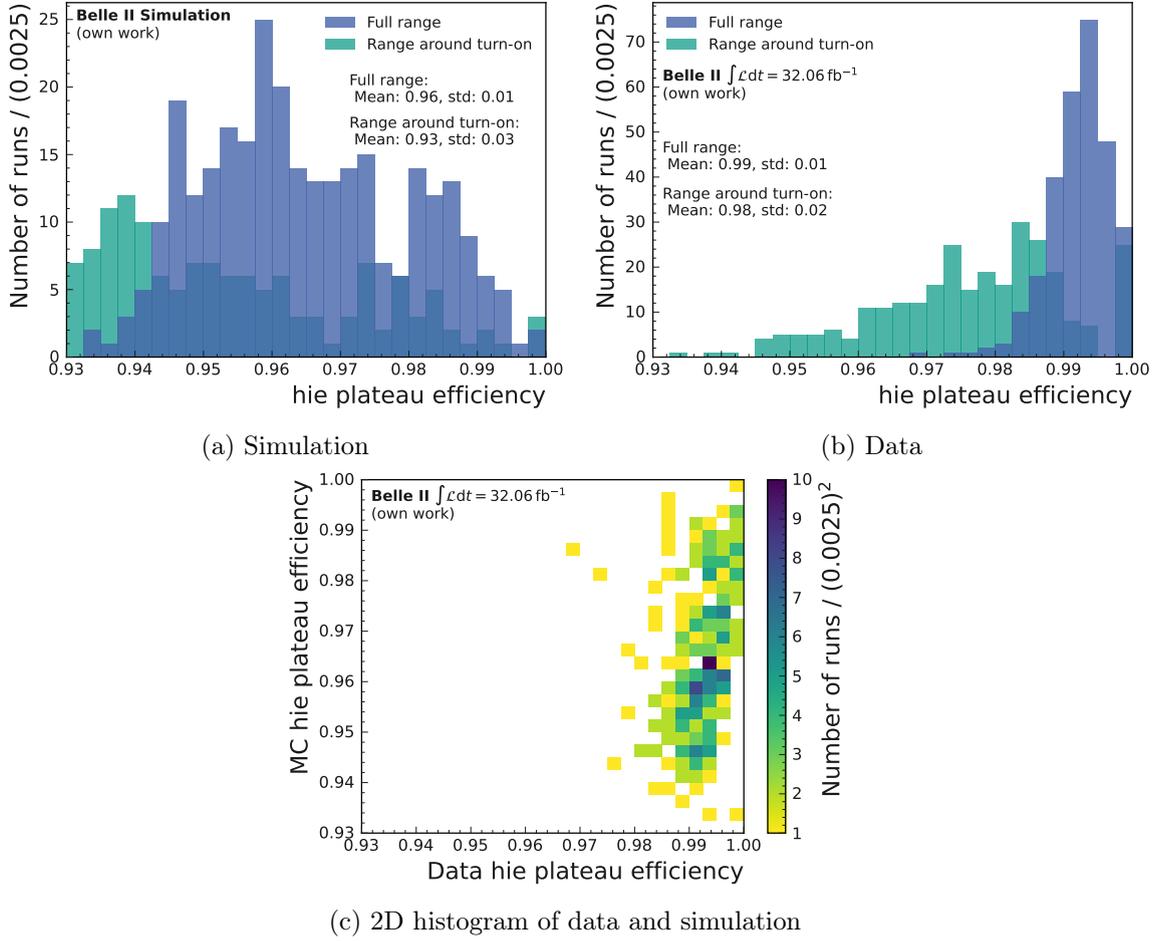


Figure 4.3.: Data-MC comparison of plateau efficiency of hie. The 1D histograms show the distribution of the plateau efficiency for simulation and data, respectively. The 2D histogram shows the correlation between the two.

plateau efficiency manifests itself in values far above the turn-on point). Even with this difference, Fig. 4.3c shows a clear correlation between simulation and data. This is because the bug in the simulation code leads to a constant offset independent of the data-taking period. The only different contribution which leads to a varying trigger line efficiency is run-dependent beam background which is quite well-modelled by the simulation. Therefore, it is no surprise that the 2D histogram shows a correlation between data and simulation.

The final fitting parameter  $y_0$  can be understood as a cross-check how well the fits are working. As the efficiency should be consistently 0 below the turn-on point, a value for  $y_0$  that is noticeably different from 0 is a hint that something went wrong.

Figure 4.4 shows histograms for data and simulation. All values except for very few outliers are of order  $10^{-4}$  or smaller which is a good sign for the stability of the monitoring.

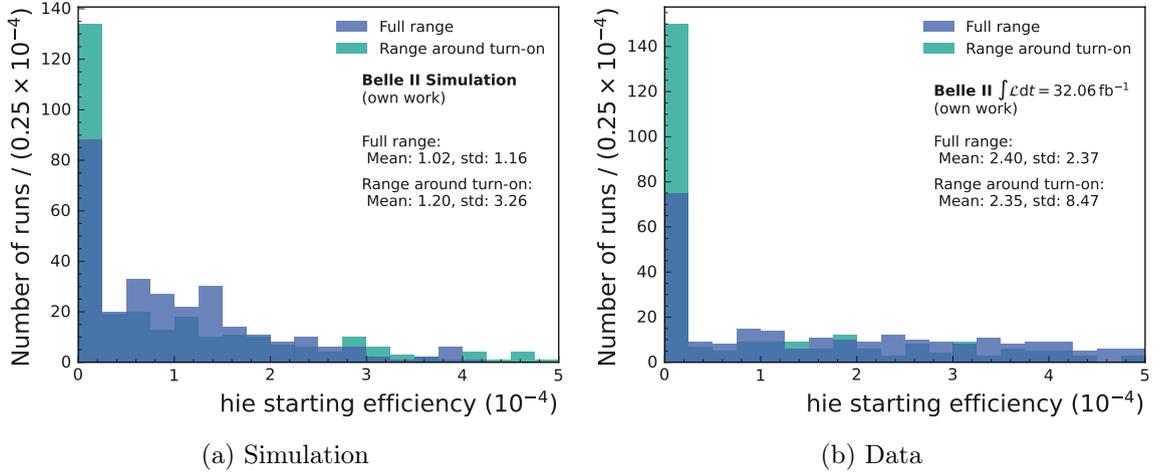


Figure 4.4.: Starting efficiency parameter  $y_0$  of data and MC. The histograms show the distribution of the parameter for simulation and data, respectively.

#### 4.2.2. fyo Performance Monitoring

For the neural 3D CDC track triggers fy30 and fyo, the  $x$ -axis is the maximal 2D azimuthal opening angle  $\Delta\phi_{\max}$  between any two tracks of the same event. It ranges from  $0^\circ$  to  $180^\circ$ . For the monitoring, I use hie as a reference line.

The trigger lines fy30 and fyo (ff30 and ffo for older experiment runs) are harder to monitor than lines such as hie because their turn-on curves are not constant after the turn-on point (see Sec. 3.3). Therefore, the full-range fit produces very inaccurate results and since the close-range fit depends on a somewhat accurate turn-on point approximation, the procedure I use for other trigger lines is too unstable for fy30 and fyo.

Instead, I restrict the range to be  $45^\circ \leq \Delta\phi_{\max} \leq 135^\circ$  for fyo (which is designed to have the turn-on at  $90^\circ$ ) and  $0^\circ \leq \Delta\phi_{\max} \leq 60^\circ$  for fy30 (which is designed to have the turn-on at  $30^\circ$ ). These ranges include enough data to perform meaningful fits while excluding the troubling regions. I then do not perform an additional fit.

Figures 4.5 and 4.6 show the distributions of the fit parameters  $m$  and  $s$  for simulation and data. Simulation produces much more stable results for both parameters. As before, the mean and standard deviation are heavily shifted by outliers outside of the shown ranges. Therefore, they should not be considered as a meaningful statement about the average result. They are handy, however, to notice immediately that the results of data are much broader spread than those of simulation.

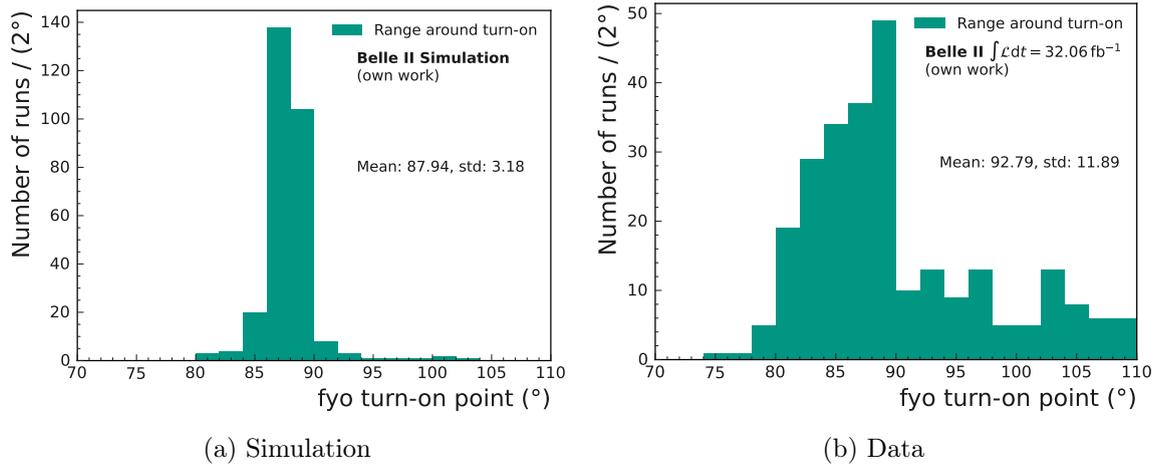


Figure 4.5.: Data-MC comparison of turn-on point of  $f_{y0}$ . The histograms show the distribution of turn-on points for simulation and data, respectively.

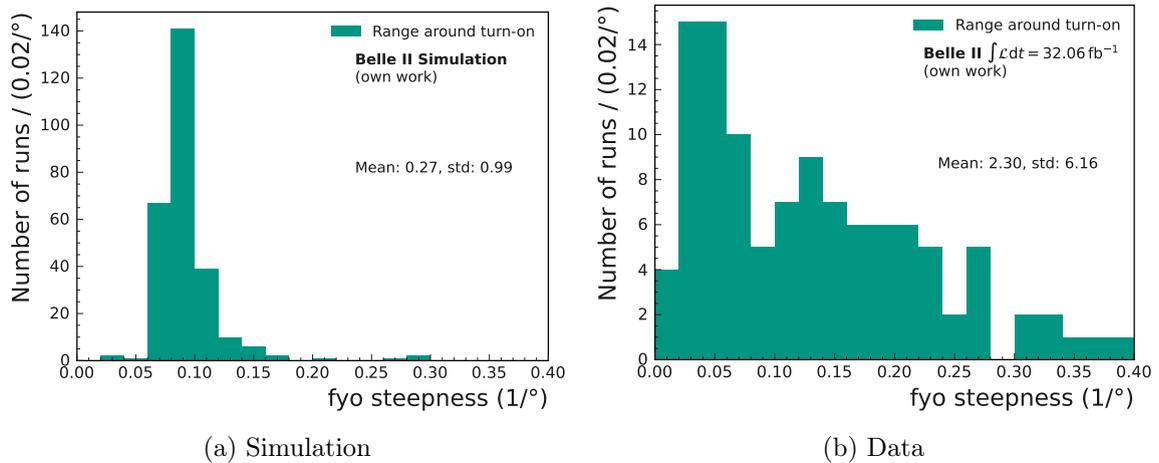


Figure 4.6.: Data-MC comparison of steepness parameter of  $f_{y0}$ . The histograms show the distribution of the parameter for simulation and data, respectively. As there is no upper limit for this parameter, outliers far outside of the shown range heavily influence the mean and standard deviation.



## 5. Displaying the Results on a Website

The run-dependent monitoring (Ch. 4) helps find problems in TRG and TSIM quickly. To be useful to collaboration members, its results need to be displayed in an easily accessible way. This is done by creating a website based on HTML, CSS and JavaScript which dynamically displays the results of Simon Weber's automated  $K_S^0$  monitoring and my work. As described in the preface, the AI tools ChatGPT [15] and GitHub Copilot [16] helped with code creation.

The starting page lets the user decide between the  $K_S^0$  monitoring and the trigger line monitoring. After choosing the trigger line monitoring, they see a page describing the procedure and linking the GitLab page of work (Fig. 5.1).

From there, an experiment can be chosen and the monitoring page will be displayed (Fig. 5.2). Here, the user can pick a trigger line and one of the four fit parameters to be displayed for every run of the chosen experiment. By default, both data and simulation are shown which can be turned off, by clicking on one of the icons in the legend. It is also possible to zoom to compare data and simulation thoroughly run by run.

Since the results are displayed dynamically using JavaScript, the website can easily be adjusted and expanded if more trigger lines and experiments are studied. Also, it can be used as a structure to display the results of more monitoring work with different approaches.

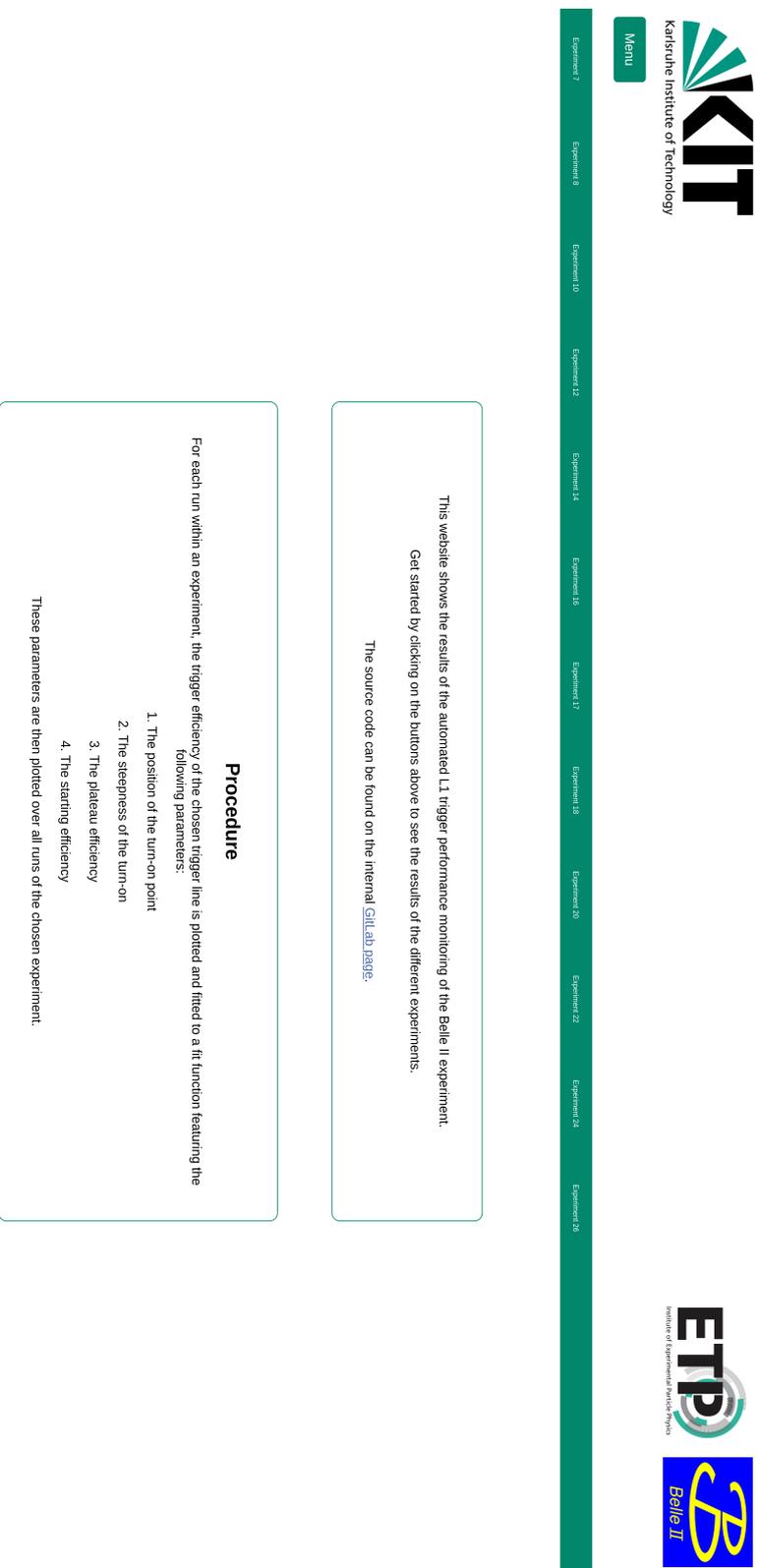
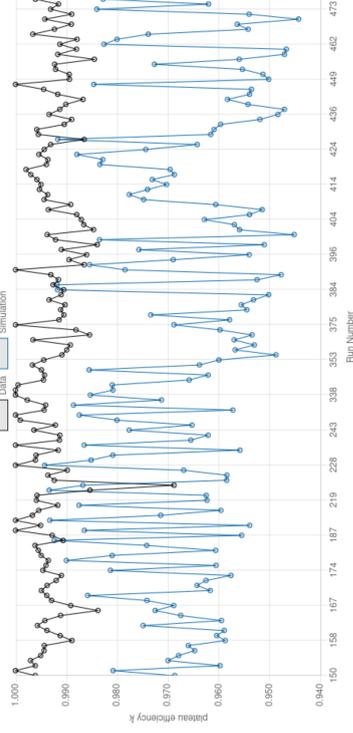


Figure 5.1.: The Introductory page for the trigger line monitoring. The user can click on the experiments in the bar at the top to see the monitoring for that experiment.



Trigger Line

- hie
- stt
- fyo

Fit Parameter

- turn-on point  $m$
- steepness parameter  $s$
- plateau efficiency  $k$
- starting efficiency  $y_0$

Figure 5.2.: The page showing the result of the trigger line monitoring. The user can select the trigger line and the fit parameter on the left. They also can turn off data or simulation and click on a different experiment at the top.



## 6. Summary and Outlook

The thesis presents an automated monitoring process of the L1 trigger performance which is applied to all physics runs up until experiment 26. The monitoring compares the turn-on curves of different trigger lines for data and simulation. In particular, it investigates the turn-on point, the turn-on steepness, and the asymptotic efficiency of the trigger line which are then compared to the ideal values.

The process is used for all experiment runs to find problematic ones that may show issues due to detector or software updates. It is important to discover issues of the trigger system quickly as running the experiment uses a lot of resources which are wasted if the trigger system does not work properly.

Data and MC show correlations between their results which means the differences in runs are well-modelled by the simulation. However, there is a lot of runs where the results differ between data and run-dependent MC. These runs in particular could be the target of subsequent studies to find out if the reason lies in statistics, the fitting process, or an actual mismodelled run-dependent simulation.

The `trigger_line_efficiency` mode which is added to VIBE makes it easy to monitor any run and trigger line which have not been studied in the thesis. It also lays the groundwork for trigger line studies of different processes or with different approaches. For example, one could study the correlation between trigger line efficiencies and observables which do not directly influence the trigger line decision.

A website makes the result of the monitoring process accessible. It can be easily expanded to contain results of different studies or to allow for more customisation if a different monitoring process is performed.

In conclusion, the L1 trigger system works well and its simulation TSIM does a good job simulating the trigger decisions for single runs. The results show no obvious flaws except for the run-dependent MC mismodelling the ECL trigger efficiencies (see Sec. 3.4) where the TSIM results are consistently a few percent lower than TRG. This issue is known and will be fixed in upcoming releases of the software framework. The monitoring process presented in this thesis can be used to verify the expected results in the new releases.



# A. Appendix

## A.1. Additional Results of the Monitoring

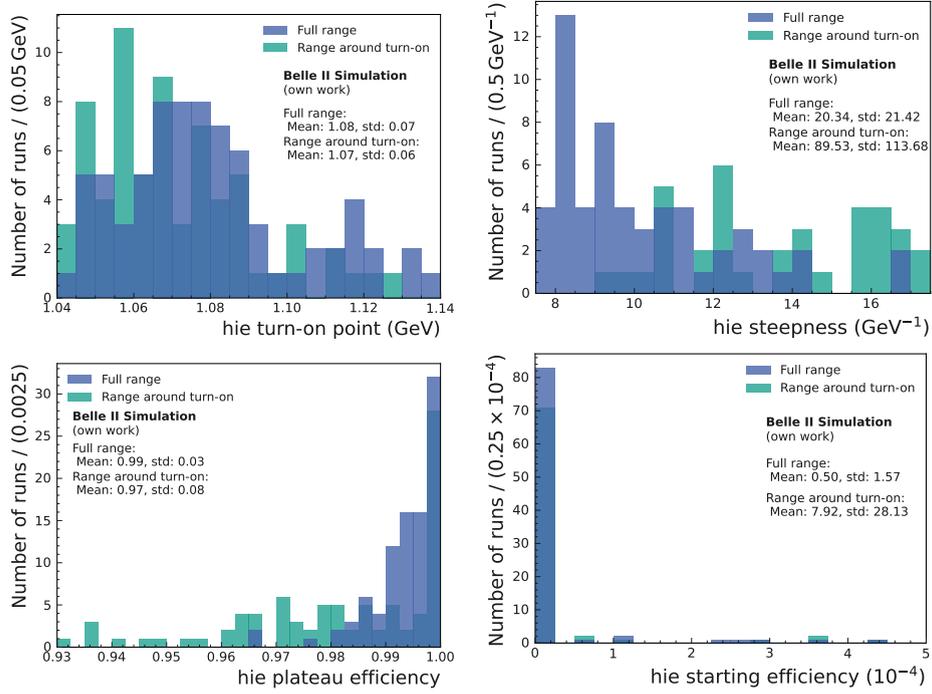


Figure A.1.: Histograms of fit parameters of trigger line hie for simulation of experiment 7.

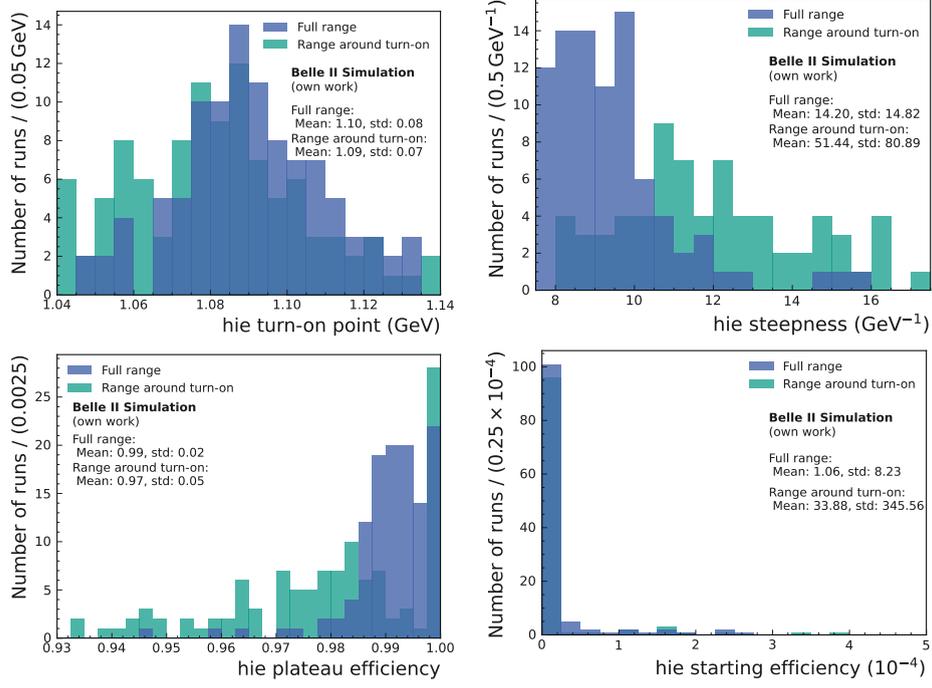


Figure A.2.: Histograms of fit parameters of trigger line hie for simulation of experiment 8.

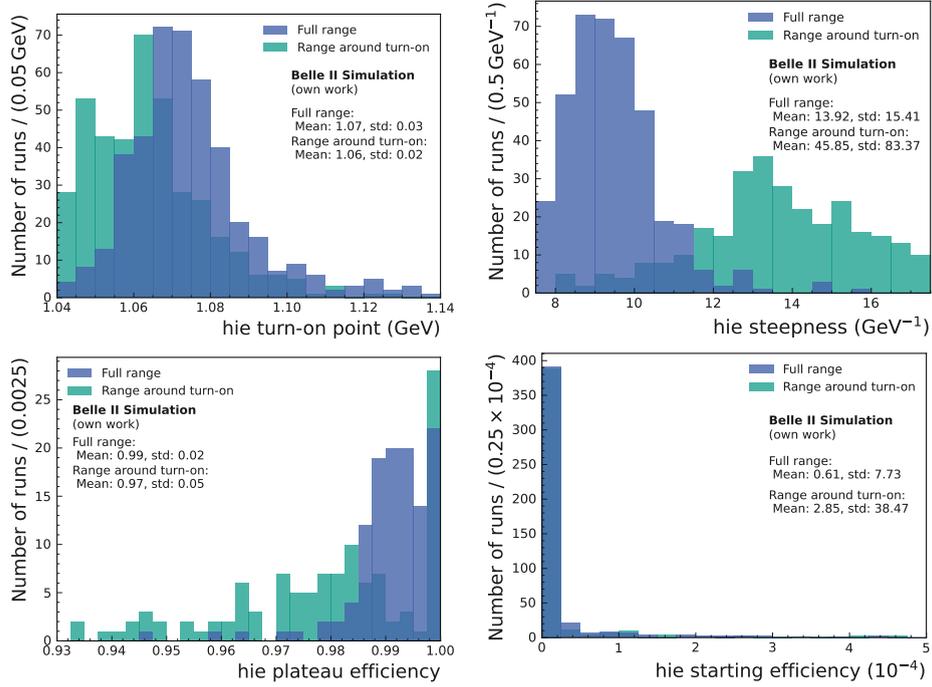


Figure A.3.: Histograms of fit parameters of trigger line hie for simulation of experiment 10.

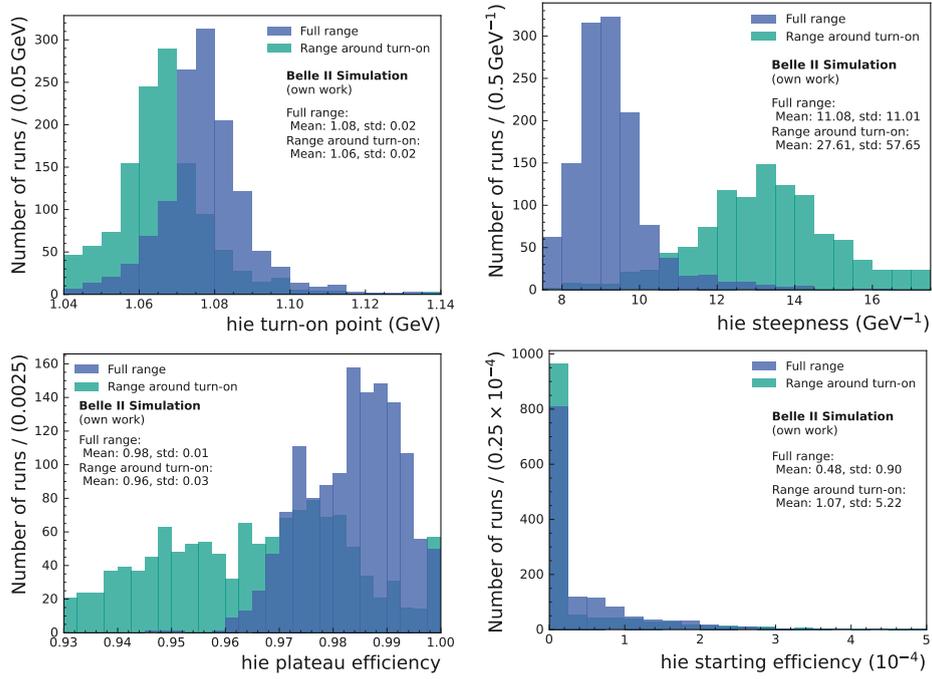


Figure A.4.: Histograms of fit parameters of trigger line hie for simulation of experiment 12.

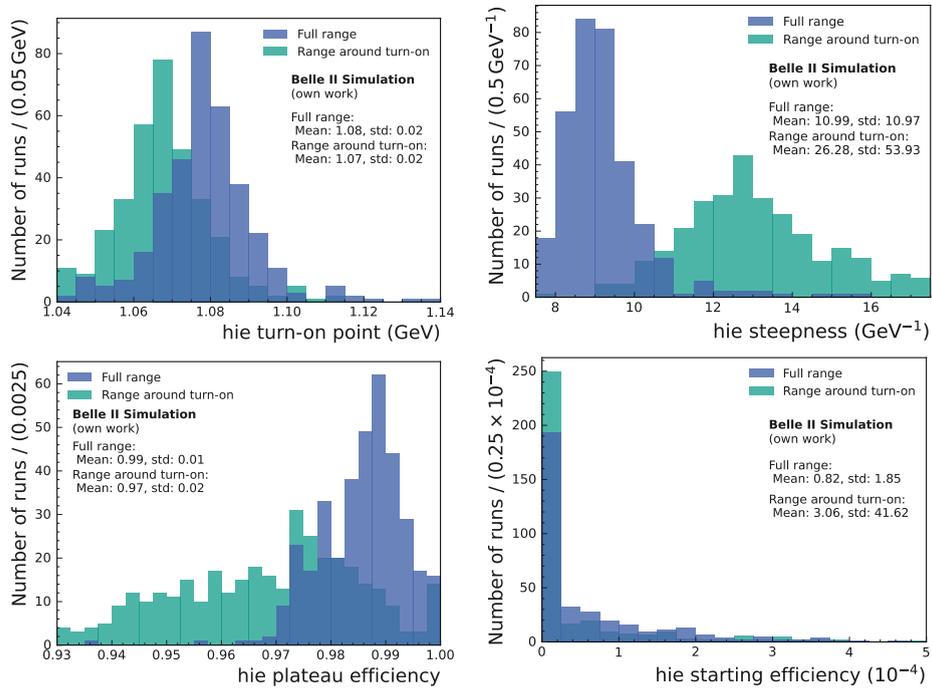


Figure A.5.: Histograms of fit parameters of trigger line hie for simulation of experiment 14.

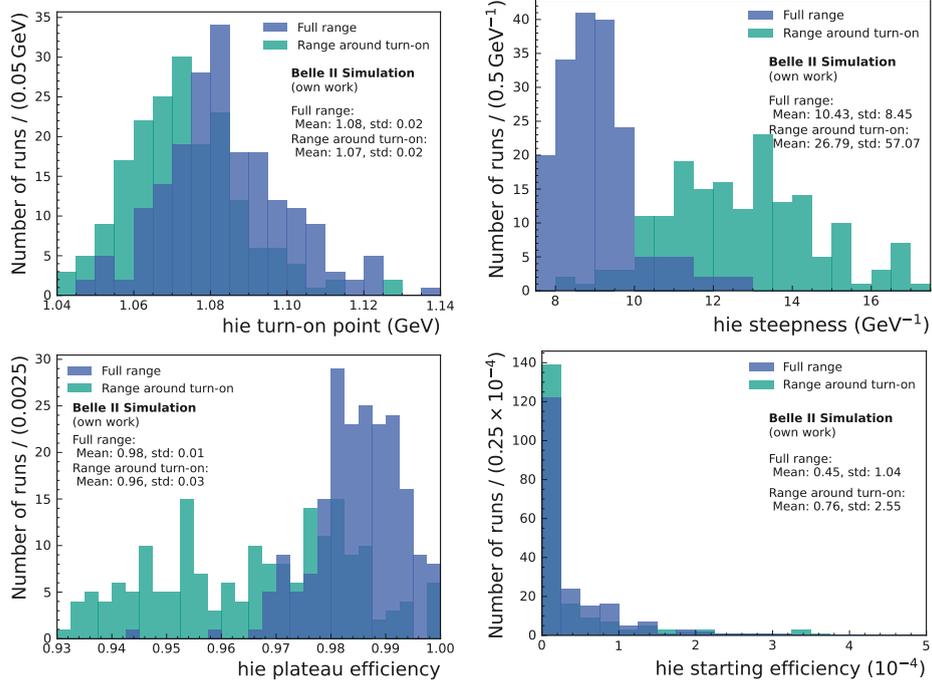


Figure A.6.: Histograms of fit parameters of trigger line hie for simulation of experiment 16.

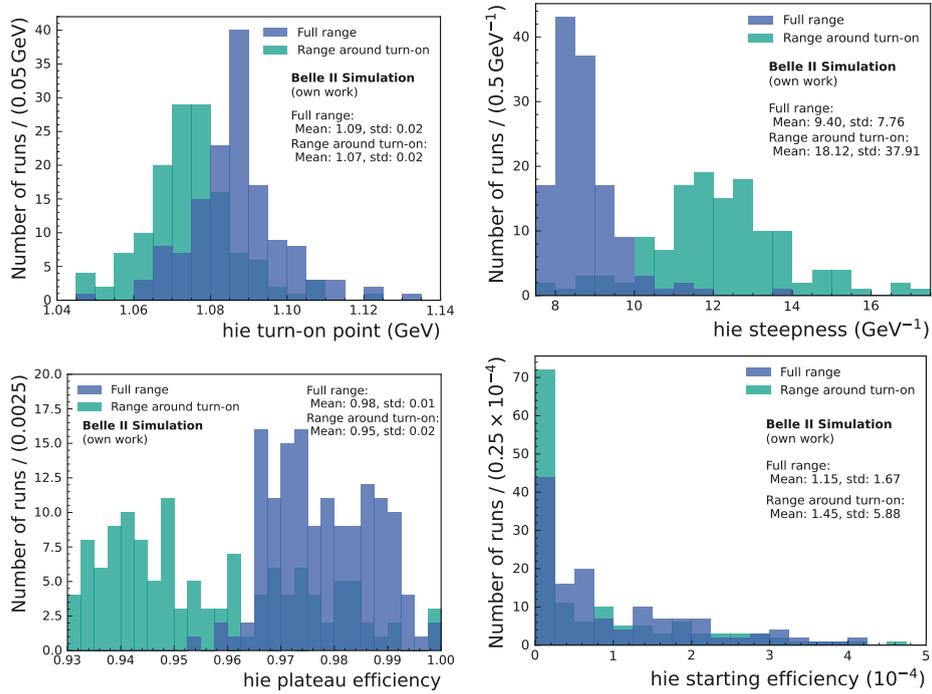


Figure A.7.: Histograms of fit parameters of trigger line hie for simulation of experiment 17.

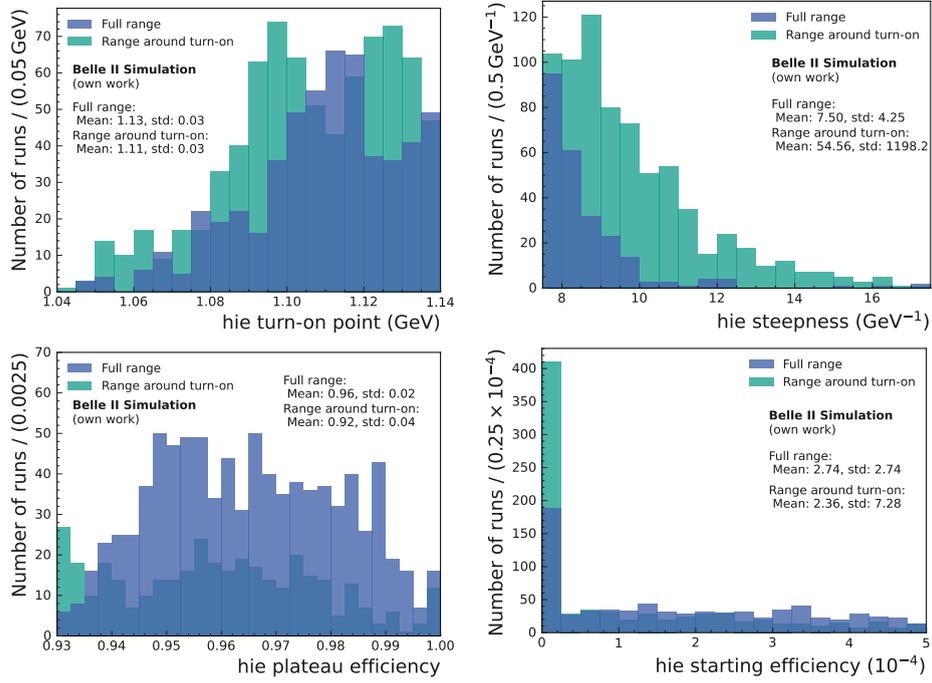


Figure A.8.: Histograms of fit parameters of trigger line hie for simulation of experiment 26.

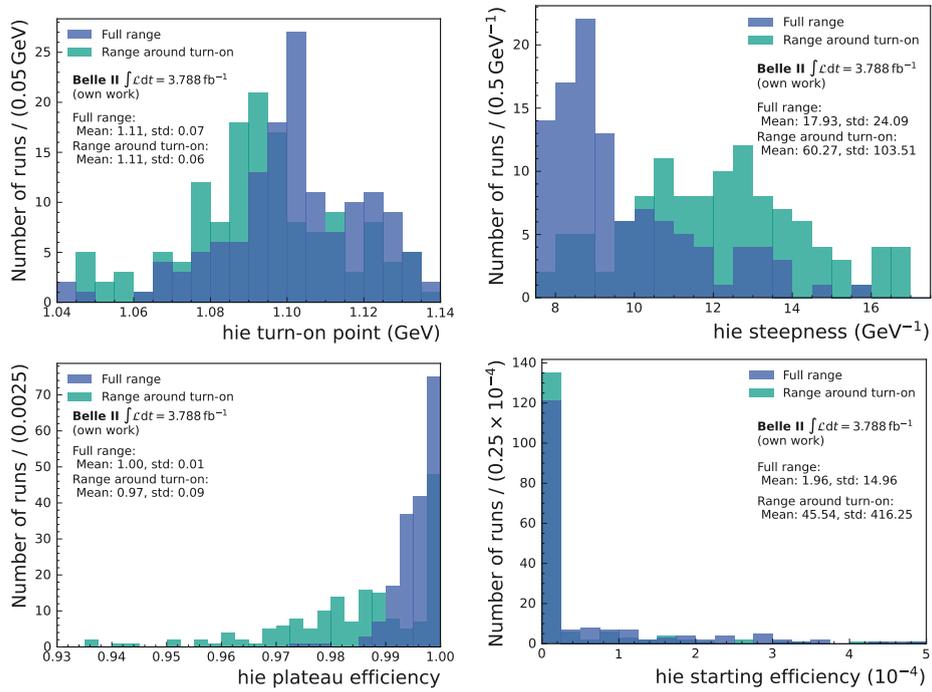


Figure A.9.: Histograms of fit parameters of trigger line hie for data of experiment 20.

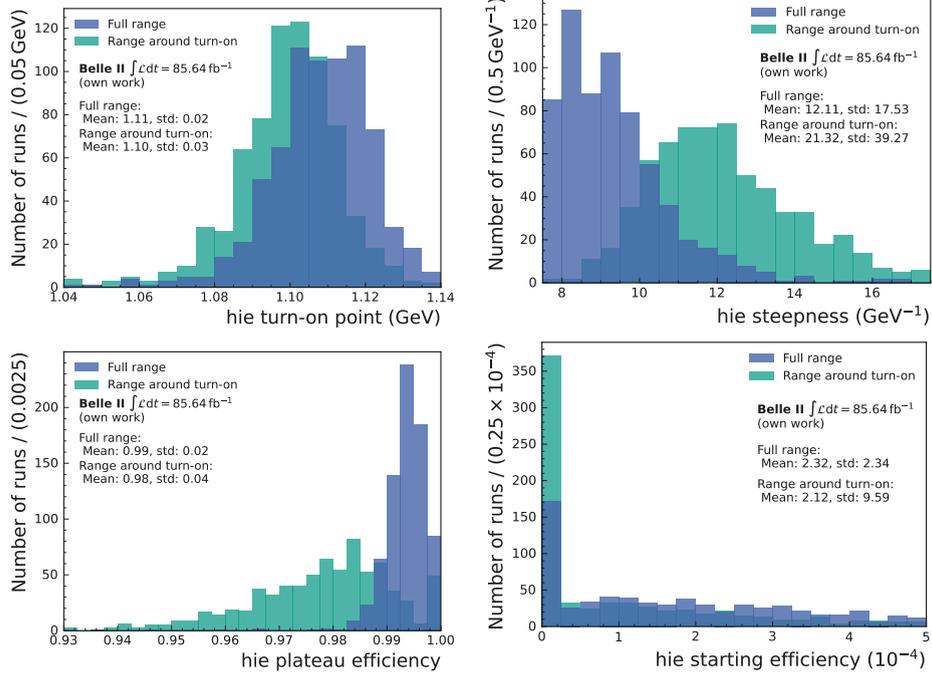


Figure A.10.: Histograms of fit parameters of trigger line hie for data of experiment 24.

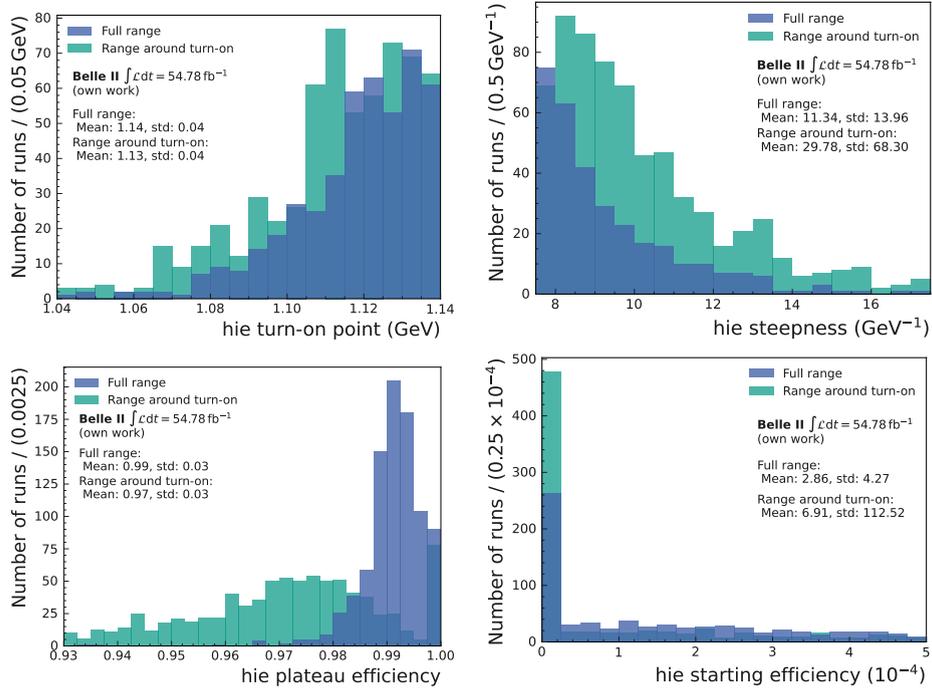


Figure A.11.: Histograms of fit parameters of trigger line hie for data of experiment 26.

# Bibliography

- [1] P. Ecker *et al.* (Belle II Collaboration), “Validation Interface for the Belle II Experiment,” 2022.  
<https://gitlab.desy.de/belle2/data-production/validation/vibe>.
- [2] T. Kuhr, C. Pulvermacher, M. Ritter, T. Hauth, and N. Braun, “The Belle II Core Software,” *Computing and Software for Big Science* **3** no. 1, (2018) 1.
- [3] P. Ecker, J. Eppelt, and G. De Pietro, “DM-Analysis-Tools,” 2022.  
<https://github.com/eckerpatrick/DM-Analysis-Tools>.
- [4] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods* **17** (2020) 261–272.
- [5] Belle II Collaboration, “Belle II Plotting Style Guide,” 2023.  
<https://gitlab.desy.de/belle2/software/belle2style>.
- [6] K. Akai, K. Furukawa, and H. Koiso, “SuperKEKB collider,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **907** (2018) 189.
- [7] S. Navas *et al.* (Particle Data Group), “Review of Particle Physics,” *Phys. Rev. D* **110**, 030001 (2024) 2054.
- [8] I. H. de la Cruz, “The Belle II experiment: fundamental physics at the flavor frontier,” *Journal of Physics: Conference Series* **761** (2016) 12017.
- [9] T. Hara, T. Kuhr, and Y. Ushiroda, “Belle II Coordinate System and Guideline of Belle II Numbering Scheme,” 2011. <https://indico.mpp.mpg.de/event/2308/contributions/4092/attachments/3414/3799/Belle2NumberingScheme.pdf>.
- [10] Belle II Software Group, “Belle II XWiki page: Trigger Bit Table,” 2024.  
<https://xwiki.desy.de/xwiki/bin/view/BI/Belle%20II%20Internal/Detector%20WebHome/Trigger%20WebHome/TriggerBitTable/>.
- [11] Belle II Software Group, “basf2 release-08-01-07 software documentation: 23. TRG,” 2024.  
<https://software.belle2.org/release-08-01-07/sphinx/trg/doc/index.html>.
- [12] Belle II Collaboration, “Belle II Analysis Software Framework (basf2) (release-06-00-09),” *Zenodo* (2022) .

- [13] Belle II Software Group, “basf2 light-2403-persian software documentation,” 2024.  
<https://software.belle2.org/light-2403-persian/sphinx/index.html>.
- [14] A. Kammann *et al.* (Belle II Collaboration), “Validation Interface for the Belle II Experiment: Level 1 Trigger Monitoring,” 2024.  
<https://gitlab.desy.de/kit-etp/l1-trigger-monitoring>.
- [15] OpenAI, “ChatGPT (GPT-4 Turbo),” 2024. [www.chatgpt.com](http://www.chatgpt.com).
- [16] GitHub and OpenAI, “GitHub Copilot (GPT-4 Turbo),” 2024.  
<https://github.com/features/copilot>.