

# **Graph-Building and Input Feature Analysis for Edge Classification in the Central Drift Chamber at Belle II**

Master Thesis of

Philipp Benjamin Dorwarth

at the Department of Physics  
Institute of Experimental Particle Physics (ETP)

Reviewer: Prof. Dr. Torben Ferber  
Second reviewer: Prof. Dr. Markus Klute  
Advisor: Dr. Slavomira Stefkova

May 16, 2022 – May 16, 2023



# **Graphkonstruktion für die Analyse von Eingabemerkmale für die Kantenklassifizierung in der zentralen Driftkammer bei Belle II**

Masterthesis von

Philipp Benjamin Dorwarth

an der Fakultät für Physik  
Institut für Experimentelle Teilchenphysik (ETP)

Referent: Prof. Dr. Torben Ferber  
Korreferent: Prof. Dr. Markus Klute  
Betreuender Mitarbeiter: Dr. Slavomira Stefkova

16. Mai 2022 – 16. Mai 2023



---

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, May 16, 2023**

.....  
Philipp Benjamin Dorwarth



# Disclaimer

The research studies presented in this thesis were proposed by my supervisor, Professor Torben Ferber.

The tracking pipeline, which is a central aspect of this thesis, was developed by Lea Reuter.

This thesis analysis features particle collision events produced using the Belle II Analysis Software Framework (basf2), developed by the Belle II Software Group [30]. This software was employed for the generation of all events used in this thesis and for the reconstruction of experimental detector data.

Simulations of the Inelastic Dark Matter with a Dark Higgs model are simulated by Lea Reuter using a simulation written by Patrick Ecker. The background overlay files were centrally provided by the data production group of the Belle II experiment.

For chapter 6, the Dimuon events were personally simulated using basf2, with the background overlay files also centrally provided by the data production group of the Belle II experiment.

The corresponding plots of the results from the studies were created with the Matplotlib package [20], along with the seaborn package [36].

The event displays were created using a framework developed by Lea Reuter, adapted to this thesis's requirements.

Please note that any internal references may be attributed by name. All individuals mentioned are esteemed members of the Belle II collaboration.



# Abstract

The Belle II experiment at the SuperKEKB accelerator in Japan offers unique opportunities for particle physics research. However, the current tracking algorithms have limitations in managing highly displaced vertices and escalating beam backgrounds. This research investigates several components of a Graph Neural Network (GNN)-based pipeline, including the exploration of input features for real-time pattern recognition algorithms, a systematic approach for graph-building based on hits in the Central Drift Chamber (CDC), the primary tracking detector of the Belle II experiment, and the use of the Interaction Network (IN) for edge classification and background clean-up.

The study found that Analog-to-Digital Converter (ADC) count and the Time-to-Digital Converter (TDC) count, representing deposited energy and associated timing information in a CDC cell, provide orthogonal discrimination power, making them both valuable for distinguishing signal from background.

Graph-building models were analyzed, utilizing different patterns for possible connections to neighboring sense wires in the CDC, using simulated events of an Inelastic Dark Matter with a Dark Higgs (IDMDH) physics model. The aim was to construct graphs that effectively encapsulate crucial information about signal particle tracks to accurately predict their hits in a subsequent classification task. Metrics are introduced to aid in balancing between capturing essential edges connecting signal hits and excluding those associated with background.

The IN is employed as a classifier in the GNN-based pipeline, performing edge classification on the constructed graphs. This process allows for the identification of signal hits and the execution of a background clean-up. The classification and the clean-up tasks yield promising results, with their effectiveness significantly influenced by the graph-building model. The clean-up tasks correctly identified up to  $(80.6 \pm 0.4) \%$  signal hits in the CDC while maintaining a purity of  $(67.4 \pm 0.4) \%$  in the hit selection. Furthermore, an initial analysis towards real-time implementation, aligning input feature resolution with anticipated resolutions at the Level 1 Trigger (L1 Trigger) stage, is conducted.

This Thesis provides encouraging evidence that a GNN-based pipeline offers a viable solution to the challenges posed by highly displaced vertices and increased beam background conditions.



# Contents

<b>Disclaimer</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. The Belle II Experiment</b>	<b>3</b>
2.1. SuperKEKB Accelerator and Belle II Detector . . . . .	3
2.2. Central Drift Chamber . . . . .	5
2.2.1. Geometry and Structure . . . . .	6
2.2.2. Readout Electronics . . . . .	7
2.2.3. Background in the Central Drift Chamber . . . . .	9
2.3. Data Taking With the Belle II Online System . . . . .	10
2.4. Belle II Analysis Software Framework . . . . .	11
2.4.1. Tracking in the Central Drift Chamber . . . . .	11
<b>3. Motivation for Advancements in Particle Tracking Algorithms</b>	<b>13</b>
3.1. Inelastic Dark Matter with a Dark Higgs . . . . .	14
3.2. Displaced Vertices at Belle II . . . . .	15
<b>4. Graph Neural Network-Based Track and Vertex Finding</b>	<b>19</b>
4.1. Pipeline for a Graph Neural Network-Based Tracking Approach . . . . .	19
4.2. Interaction Network . . . . .	21
4.2.1. Mathematical Description of the Implemented Interaction Network . . . . .	22
4.2.2. Pytorch Geometric . . . . .	23
4.2.3. Training Theory . . . . .	23
<b>5. Monte Carlo Simulation and Data Sampling for Event Analysis</b>	<b>25</b>
5.1. Introduction to Monte Carlo Simulation at Belle II . . . . .	25
5.2. Dimuon Sample Generation for Feature Analysis with Monte Carlo Simulation . . . . .	26
5.3. Data Sample Reconstruction for Feature Analysis . . . . .	27

5.4. Simulating Inelastic Dark Matter With a Dark Higgs Signal Using Monte Carlo Simulation . . . . .	28
<b>6. Investigation of Central Drift Chamber Signal Characteristics as Potential Input Features for Neural Networks</b>	<b>31</b>
6.1. Methodology for Comparing Signal Information of the Central Drift Chamber in Monte Carlo Simulation and Belle II Data . . . . .	31
6.1.1. Selection of Events . . . . .	33
6.1.2. Comparison of Track and Hit Feature Distributions . . . . .	35
6.2. Results . . . . .	38
6.2.1. Evaluating the Discriminatory Potential of Analog-to-Digital Converter Counts . . . . .	39
6.2.2. Angular Dependence of the Analog-to-Digital Converter count	42
6.2.3. Evaluating the Discriminatory Potential of Time-to-Digital Converter Counts . . . . .	45
6.2.4. Orthogonality of Analog-to-Digital Converter and Time-to-Digital Converter Distributions . . . . .	46
6.2.5. Spatial Distribution of Central Drift Chamber Hits . . . . .	49
6.2.6. Implementing Analog-to-Digital Converter and Time-to-Digital Converter Information in Event Analysis . . . . .	50
<b>7. Graph-Building</b>	<b>53</b>
7.1. Signal and Background . . . . .	54
7.2. Wire and Layer Distance . . . . .	55
7.3. Metrics for Evaluating Graph-Building . . . . .	58
7.3.1. Complete Set of True Edges . . . . .	58
7.3.2. The True Graph . . . . .	60
7.3.3. Overcount Metric . . . . .	63
7.4. Graph-Building Models . . . . .	64
7.4.1. Characterization of Graph-Building Models . . . . .	67
7.4.2. True Edge Purity and True Edge Efficiency . . . . .	68
7.4.3. True Graph Purity and True Graph Efficiency . . . . .	72
7.4.4. Overcount . . . . .	73
7.4.5. Recurring Patterns in the Metrics . . . . .	73
<b>8. Background Clean-Up in then Central Drift Chamber Using the Interaction Network</b>	<b>77</b>
8.1. Interaction Network Training and Dataset Description . . . . .	78
8.1.1. Implementation in PyTorch Geometric . . . . .	79
8.1.2. Training of the Interaction Networks . . . . .	81
8.2. Metrics for the Classification Task . . . . .	82
8.2.1. Definition of Classified Purities . . . . .	84
8.2.2. Definition of Classified Efficiencies . . . . .	84
8.2.3. Threshold Determination Using the F1 Score . . . . .	85
8.2.4. Classification of Nodes . . . . .	86

---

8.3.	Performance Measurements . . . . .	86
8.3.1.	F1 Score of the True Edges . . . . .	86
8.3.2.	F1 Score of the True Graph . . . . .	88
8.3.3.	Beam Background Clean-Up and Node Classification . . . . .	92
8.3.4.	Conclusion on the Background Clean-Up With the Interaction Network . . . . .	93
8.4.	Discussion on Real-Time Implementation . . . . .	96
8.4.1.	Floating Point Optimization . . . . .	98
8.4.2.	Performance Assessment of Edge Classification With Down-scaled Resolution Inputs . . . . .	99
<b>9.</b>	<b>Conclusion and Outlook</b>	<b>103</b>
	<b>Acronyms</b>	<b>107</b>
	<b>Bibliography</b>	<b>111</b>
<b>A.</b>	<b>Appendix: Input Feature Analysis</b>	<b>117</b>
A.1.	Input Feature Analysis . . . . .	117
A.2.	Run-Dependent Analysis . . . . .	118
A.3.	Experiment 20, Run 672 . . . . .	123
A.4.	Experiment 20, Run 874 . . . . .	128
A.5.	Experiment 21, Run 116 . . . . .	133
A.6.	Experiment 21, Run 128 . . . . .	138
A.7.	Experiment 22, Run 32 . . . . .	143
A.8.	Experiment 22, Run 546 . . . . .	148
<b>B.</b>	<b>Appendix: Graph-Building</b>	<b>153</b>
B.1.	Additional Information on the Graph-Building With <i>Nominal Phase 3</i> Background . . . . .	153
B.2.	Graph-Building Analysis with <i>early Phase 3</i> Background . . . . .	155
<b>C.</b>	<b>Appendix: Background Clean-Up</b>	<b>165</b>
C.1.	Background Clean-Up on Early Phase 3 Background . . . . .	165
C.2.	Edge Classification on Nominal Phase 3 Background With Reduced Input Features . . . . .	173



# List of Figures

2.1.	The SuperKEKB accelerator. . . . .	4
2.2.	The Belle II detector with highlighted components. . . . .	5
2.3.	Technical illustration of the Central Drift Chamber. . . . .	6
2.4.	Axial and stereo layers of the Central Drift Chamber . . . . .	7
2.5.	Trigger wires in stereo and axial layers after Long Shutdown 2 . . . . .	8
3.1.	Theoretical Feynman diagram of a Inelastic Dark Matter with a Dark Higgs signature. . . . .	15
3.2.	Offline reconstruction efficiency of the Belle II Analysis Software Framework . . . . .	16
3.3.	Online Level 1 Trigger efficiency of the Single Track Trigger. . . . .	17
4.1.	Illustration of a Graph Neural Network-based Track- and Vertexfinding pipeline . . . . .	20
6.1.	Angular track distribution comparison before and after L1 Trigger bit selection. . . . .	34
6.2.	Transversal momentum distribution of reconstructed tracks. . . . .	36
6.3.	Longitudinal momentum distribution of reconstructed tracks. . . . .	36
6.4.	Dimuon invariant mass distribution. . . . .	37
6.5.	Muon Particle Identification distribution. . . . .	37
6.6.	Analog-to-Digital Converter count distributions for signal and background hits. . . . .	41
6.7.	Two-dimensional histogram of Analog-to-Digital Converter counts in relation to the polar angle $\theta$ for Experimental Data. . . . .	42
6.8.	Two-dimensional histogram of Analog-to-Digital Converter counts in relation to the polar angle $\theta$ for simulation. . . . .	43
6.9.	Illustration of the space charge effect in the Central Drift Chamber. . . . .	43
6.10.	Ratio of experimental and simulated Analog-to-Digital Converter counts relative to the polar angle $\theta$ . . . . .	44
6.11.	Time-to-Digital Converter distributions for signal and background hits. . . . .	46
6.12.	Two-dimensional histogram of Analog-to-Digital Converter and Time-to-Digital Converter counts for Central Drift Chamber hits in experimental data. . . . .	47
6.13.	Two-dimensional histogram of Analog-to-Digital Converter and Time-to-Digital Converter counts for Central Drift Chamber hits in simulation. . . . .	47

6.14. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter and Time-to-Digital Converter counts for Central Drift Chamber hits. . . . .	48
6.15. Spatial distribution of Central Drift Chamber hits in $x$ direction. . .	49
6.16. Spatial distribution of Central Drift Chamber hits in $y$ direction. . . .	50
7.1. Detail from the Central Drift Chamber showing signal and background hits, and edge classifications. . . . .	55
7.2. Schematic for layer and wire distance in the Central Drift Chamber for graph-building. . . . .	57
7.3. True edges with different layer distances. . . . .	60
7.4. Illustration of the true graph estimation process. . . . .	61
7.5. Event display after with graph edges and true graph edges. . . . .	62
7.6. Process steps of the true graph calculation. . . . .	63
7.7. Wires used for the Track Segment Finder and wires used for the graph-building models. . . . .	65
7.8. Illustration of the graph-building algorithm. . . . .	66
7.9. Ten different graph-building models. . . . .	66
7.10. The <i>true edge pur.</i> in relation to the number of edges for ten graph-building models. . . . .	69
7.11. The <i>true edge pur.</i> in relation to the maximum number of allowed connections for ten graph-building models. . . . .	70
7.12. The <i>true edge pur.</i> in relation to the <i>true edge eff.</i> for ten graph-building models. . . . .	71
7.13. The <i>true edge pur.</i> in relation to the number of edges for ten graph-building models. . . . .	71
7.14. The <i>true graph pur.</i> in relation to the <i>true graph eff.</i> for ten graph-building models. . . . .	72
7.15. The <i>true graph pur.</i> in relation to the <i>overcount</i> for ten graph-building models. . . . .	74
7.16. The <i>overcount</i> in relation to the median number of edges for ten graph-building models. . . . .	75
7.17. Scatter plot of the <i>true graph eff.</i> in relation to the opening angle . .	75
7.18. Histogrammed <i>true graph eff.</i> distribution. . . . .	76
8.1. Illustration of the Interaction Network implemented for this thesis. . .	79
8.2. Train loss for training with nominal Phase 3 background. . . . .	83
8.3. Validation loss for training with nominal Phase 3 background. . . . .	83
8.4. F1 score of the classified true edges for different dark Higgs masses. . .	87
8.5. The <i>class. true edge eff.</i> in relation to the <i>class. true edge pur.</i> . . . .	89
8.6. F1 score of the classified true graph edges for different dark Higgs masses.	90
8.7. The <i>class. true graph eff.</i> in relation to the <i>class. true graph pur.</i> . . .	91
8.8. Event display showing the background clean-up with the Interaction Network. . . . .	94

---

8.9.	The <i>node efficiency</i> in relation to the <i>node purity</i> . . . . .	95
8.10.	Correlation of the introduced metrics for nominal phase 3 background. . . . .	96
A.1.	Distribution of the momentum $p_x$ of reconstructed tracks. . . . .	117
A.2.	Distribution of the momentum $p_y$ of reconstructed tracks. . . . .	118
A.3.	Transversal momentum distribution of reconstructed tracks (experiment 24, run 888, run-dependent simulation). . . . .	119
A.4.	Longitudinal momentum distribution of reconstructed tracks (experiment 24, run 888, run-dependent simulation). . . . .	119
A.5.	Muon Particle Identification (PID) distribution (experiment 24, run 888, run-dependent simulation). . . . .	120
A.6.	Dimuon invariant mass distribution (experiment 24, run 888, run-dependent simulation). . . . .	120
A.7.	Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 24, run 888, run-dependent simulation). . . . .	121
A.8.	Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 24, run 888, run-dependent simulation). . . . .	121
A.9.	Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 24, run 888, run-dependent simulation). . . . .	122
A.10.	Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 24, run 888, run-dependent simulation). . . . .	122
A.11.	Transversal momentum distribution of reconstructed tracks (experiment 20, run 672, run-independent simulation). . . . .	123
A.12.	Longitudinal momentum distribution of reconstructed tracks (experiment 20, run 672, run-independent simulation). . . . .	124
A.13.	Muon Particle Identification (PID) distribution (experiment 20, run 672, run-independent simulation). . . . .	124
A.14.	Dimuon invariant mass distribution (experiment 20, run 672, run-independent simulation). . . . .	125
A.15.	Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 20, run 672, run-independent simulation). . . . .	125
A.16.	Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 20, run 672, run-independent simulation). . . . .	126
A.17.	Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 20, run 672, run-independent simulation). . . . .	127
A.18.	Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 20, run 672, run-independent simulation). . . . .	127

A.19. Transversal momentum distribution of reconstructed tracks (experiment 20, run 874, run-independent simulation). . . . .	128
A.20. Longitudinal momentum distribution of reconstructed tracks (experiment 20, run 874, run-independent simulation). . . . .	129
A.21. Muon Particle Identification (PID) distribution (experiment 20, run 874, run-independent simulation). . . . .	129
A.22. Dimuon invariant mass distribution (experiment 20, run 874, run-independent simulation). . . . .	130
A.23. Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 20, run 874, run-independent simulation). . . . .	130
A.24. Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 20, run 874, run-independent simulation). . . . .	131
A.25. Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 20, run 874, run-independent simulation). . . . .	132
A.26. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 20, run 874, run-independent simulation). . . . .	132
A.27. Transversal momentum distribution of reconstructed tracks (experiment 21, run 116, run-independent simulation). . . . .	133
A.28. Longitudinal momentum distribution of reconstructed tracks (experiment 21, run 116, run-independent simulation). . . . .	134
A.29. Muon Particle Identification (PID) distribution (experiment 21, run 116, run-independent simulation). . . . .	134
A.30. Dimuon invariant mass distribution (experiment 21, run 116, run-independent simulation). . . . .	135
A.31. Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 21, run 116, run-independent simulation). . . . .	135
A.32. Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 21, run 116, run-independent simulation). . . . .	136
A.33. Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 21, run 116, run-independent simulation). . . . .	137
A.34. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 21, run 116, run-independent simulation). . . . .	137
A.35. Transversal momentum distribution of reconstructed tracks (experiment 21, run 128, run-independent simulation). . . . .	138
A.36. Longitudinal momentum distribution of reconstructed tracks (experiment 21, run 128, run-independent simulation). . . . .	139
A.37. Muon Particle Identification (PID) distribution (experiment 21, run 128, run-independent simulation). . . . .	139

A.38. Dimuon invariant mass distribution (experiment 21, run 128, run-independent simulation). . . . .	140
A.39. Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 21, run 128, run-independent simulation). . . . .	140
A.40. Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 21, run 128, run-independent simulation). . . . .	141
A.41. Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 21, run 128, run-independent simulation). . . . .	142
A.42. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 21, run 128, run-independent simulation). . . . .	142
A.43. Transversal momentum distribution of reconstructed tracks (experiment 22, run 32, run-independent simulation). . . . .	143
A.44. Longitudinal momentum distribution of reconstructed tracks (experiment 22, run 32, run-independent simulation). . . . .	144
A.45. Muon Particle Identification (PID) distribution (experiment 22, run 32, run-independent simulation). . . . .	144
A.46. Dimuon invariant mass distribution (experiment 22, run 32, run-independent simulation). . . . .	145
A.47. Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 22, run 32, run-independent simulation). . . . .	145
A.48. Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 22, run 32, run-independent simulation). . . . .	146
A.49. Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 22, run 32, run-independent simulation). . . . .	147
A.50. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 22, run 32, run-independent simulation). . . . .	147
A.51. Transversal momentum distribution of reconstructed tracks (experiment 22, run 546, run-independent simulation). . . . .	148
A.52. Longitudinal momentum distribution of reconstructed tracks (experiment 22, run 546, run-independent simulation). . . . .	149
A.53. Muon Particle Identification (PID) distribution (experiment 22, run 546, run-independent simulation). . . . .	149
A.54. Dimuon invariant mass distribution (experiment 22, run 546, run-independent simulation). . . . .	150
A.55. Ratio of experimental and simulated Analog-to-Digital Converter (ADC) counts relative to the polar angle $\theta$ (experiment 22, run 546, run-independent simulation). . . . .	150

A.56. Analog-to-Digital Converter (ADC) count distributions for signal and background hits (experiment 22, run 546, run-independent simulation).	151
A.57. Time-to-Digital Converter (TDC) count distributions for signal and background hits (experiment 22, run 546, run-independent simulation).	152
A.58. Ratio of two-dimensional histograms of experimental and simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits (experiment 22, run 546, run-independent simulation).	152
B.1. The <i>true edge pur.</i> in relation to the <i>number of edges</i> for ten graph-building models.	157
B.2. The <i>true edge pur.</i> in relation to the <i>maximum number of possible connections</i> for ten graph-building models.	158
B.3. The <i>true edge pur.</i> in relation to the <i>true edge eff.</i> for ten graph-building models.	159
B.4. The <i>true edge pur.</i> in relation to the <i>number of edge</i> for ten graph-building models.	160
B.5. The <i>true graph pur.</i> in relation to the <i>true graph eff.</i> for ten graph-building models.	161
B.6. The <i>true edge pur.</i> in relation to the <i>overcount</i> for ten graph-building models.	162
B.7. The <i>overcount</i> in relation to the <i>number of edges</i> for ten graph-building models.	163
C.1. Correlation of the metrics for <i>early Phase 3</i> background.	166
C.2. F1 score of the classified true edges for different dark Higgs masses.	168
C.3. The <i>class. true edge eff.</i> in relation to the <i>class. true edge pur.</i>	169
C.4. F1 score of the classified true graph edges for different dark Higgs masses.	169
C.5. The <i>class. true graph eff.</i> in relation to the <i>class. true graph pur.</i>	170
C.6. Event display showing the background clean-up with the Interaction Network (IN) with <i>early Phase 3</i> background.	171
C.7. The <i>node efficiency</i> in relation to the <i>node purity</i>	172
C.8. F1 score of the classified true edges for different dark Higgs masses with Level 1 Trigger (L1 Trigger) information.	174
C.9. The <i>class. true edge eff.</i> in relation to the <i>class. true edge pur.</i> with Level 1 Trigger (L1 Trigger) information.	175
C.10. F1 score of the classified true graph edges for different dark Higgs masses with Level 1 Trigger (L1 Trigger) information.	175
C.11. The <i>class. true graph eff.</i> in relation to the <i>class. true graph pur.</i> with Level 1 Trigger (L1 Trigger) information.	176

# List of Tables

2.1.	SuperKEKB machine parameters, from [28, 1]. . . . .	4
2.2.	Detailed information on the superlayers of the Central Drift Chamber. Taken from [1]. . . . .	7
2.3.	Current and anticipated information from the front-end electronics of the Central Drift Chamber. . . . .	8
5.1.	Parameters and Conditions for the Run-Independent Dimuon Sample.	27
5.2.	Parameters and Conditions for the Run-Dependent Dimuon Sample. .	27
5.3.	Reconstruction Conditions for the Experimental Data. . . . .	28
5.4.	Parameters and Conditions for the Inelastic Dark Matter with a Dark Higgs simulation. . . . .	29
5.5.	Current optics and beam conditions of the Belle II experiment. . . . .	29
5.6.	Comparison of anticipated beam parameters of the Belle II experiment.	29
6.1.	Summary of the <code>mumu_tight</code> High Level Trigger skim. . . . .	32
6.2.	Selection efficiency comparison for the skimmed data sample of experi- ment 24, run 888 and simulation. . . . .	34
6.3.	Experimental data to Monte Carlo simulation ratios of hits and tracks after selection. . . . .	39
7.1.	Median edges for the ten different graph-building models for <i>nomi- nal Phase 3</i> background. . . . .	68
7.2.	Model-independent median number of edges in different layer and wire distances for <i>nominal Phase 3</i> background. . . . .	68
7.3.	<i>true edge eff.</i> and <i>true edge pur.</i> of the ten graph-building models. . .	69
7.4.	The <i>true graph eff.</i> and <i>true graph pur.</i> of the ten graph-building models for <i>nominal Phase 3</i> background. . . . .	73
8.1.	Model summary of the Interaction Network implemented in PyTorch.	81
8.2.	Hyperparameters for training of the Interaction Network. . . . .	82
8.3.	The <i>true edge eff.</i> , the <i>class. true edge eff.</i> , and the <i>total true edge eff.</i>	89
8.4.	The <i>true graph eff.</i> , the <i>class. true graph eff.</i> , and the <i>total true graph eff.</i>	91
8.5.	The <i>node efficiency</i> and the <i>node purity</i> . . . . .	93
8.6.	Median estimated Floating Point Operations. . . . .	98
8.7.	Summary of the adapted interaction network for reduced resolution inputs. . . . .	100

8.8.	The <i>class. true edge eff.</i> and the <i>class. true edge pur.</i> with reduced resolution inputs. . . . .	101
8.9.	The <i>true graph eff.</i> , and the <i>total true graph eff.</i> with reduced resolution inputs. . . . .	102
A.1.	Selection efficiency comparison for the skimmed data and simulation (experiment 24, run 888, run-dependent simulation). . . . .	118
A.2.	Selection efficiency comparison for the skimmed data and simulation (experiment 20, run 672, run-independent simulation). . . . .	123
A.3.	Selection efficiency comparison for the skimmed data and simulation (experiment 20, run 874, run-independent simulation). . . . .	128
A.4.	Selection efficiency comparison for the skimmed data and simulation (experiment 21, run 116, run-independent simulation). . . . .	133
A.5.	Selection efficiency comparison for the skimmed data and simulation (experiment 21, run 128, run-independent simulation). . . . .	138
A.6.	Selection efficiency comparison for the skimmed data and simulation (experiment 22, run 32, run-independent simulation). . . . .	143
A.7.	Selection efficiency comparison for the skimmed data and simulation (experiment 22, run 546, run-independent simulation). . . . .	148
B.1.	Median number of edges for each superlayer in the Central Drift Chamber.	154
B.2.	Median edges for the ten different graph-building models for <i>early Phase 3</i> background. . . . .	155
B.3.	Model-independent median number of edges with different wire and layer distances for <i>early Phase 3</i> background. . . . .	155
B.4.	The <i>true edge eff.</i> and <i>true edge pur.</i> with <i>early Phase 3</i> background.	156
B.5.	The <i>true graph eff.</i> and <i>true graph pur.</i> with <i>early Phase 3</i> background.	156
C.1.	Median estimated Floating Point Operations with <i>early Phase 3</i> background. . . . .	165
C.2.	The <i>true edge eff.</i> , <i>class. true edge eff.</i> and the combined <i>total true edge eff.</i> with <i>early Phase 3</i> background. . . . .	167
C.3.	The <i>class. true graph eff.</i> , <i>true graph eff.</i> and the combined <i>total true graph eff.</i> with <i>early Phase 3</i> background. . . . .	167
C.4.	The <i>node efficiency</i> and <i>node purity</i> with <i>early Phase 3</i> background.	168
C.5.	Median estimated Floating Point Operations, using <i>nominal Phase 3</i> background and reduced input feature resolution. . . . .	173
C.6.	The <i>true edge eff.</i> , and the <i>total true edge eff.</i> , using <i>nominal Phase 3</i> background and reduced resolution inputs. . . . .	173
C.7.	The <i>class. true graph eff.</i> and the <i>class. true graph pur.</i> , using <i>nominal Phase 3</i> background and reduced resolution inputs. . . . .	174

# 1. Introduction

For years, scientists have been striving to gain a deeper understanding of the fundamental forces in our universe through experimental and theoretical research in particle physics. One of the leading experiments in this field is the Belle II experiment, located at Japan's SuperKEKB accelerator facility. The high-precision experiment operates at the  $\Upsilon(4S)$  resonance, where electrons and positrons collide at asymmetric beam energies in a clean environment [29, 1]. With the SuperKEKB achieving a new world record instantaneous luminosity of  $4.7 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  in June 2022 and an ambitious target data set of  $50 \text{ ab}^{-1}$ , Belle II provides unparalleled opportunities in searches for new physics in rare processes at the intensity frontier.

However, these achievements come with unique challenges. The Belle II collaboration continuously pushes the limits of the instantaneous luminosity of the SuperKEKB accelerator even further by increasing beam currents and reducing beam size at the Interaction Point (IP). This results in an increased beam background, affecting the performance of established particle tracking algorithms which are fundamental for the physics analysis. Common tracking algorithms, like the combinatorial Kalman filters and the Legendre transform algorithms, scale in their computational complexity worse than linear with increased background hits in the detector [7, 33, 25, 17].

Furthermore, the existing tracking algorithm of the Belle II experiment is optimized for particles originating from the IP, such as  $B$ -decays. However, some particles of the Standard Model of Particle Physics (SM), like  $\Lambda$  and  $K_s^0$ , as well as potential Beyond the Standard Model (BSM) particles, can originate from displaced vertices. These particles decay in the detector spatially separated from the IP due to their comparably high lifetimes. The Belle II reconstruction algorithm is not optimized for displaced vertices. Thus, it is known to have reduced efficiency for these particles. Moreover, the efficiency of the Belle II tracking algorithm drops for low transverse momentum tracks.

The application of deep learning algorithms and, more specifically, Geometrical Deep Learning (GeoDL) applications, such as Graph Neural Networks (GNNs), have recently gained attention in the field of particle tracking [5, 25, 16]. This thesis explores the steps of such an algorithm to address the challenges for particle tracking in the Belle II experiment. This investigation forms part of a broader research initiative to establish a fully GNN-based tracking pipeline, with the aim of improving the detection and tracking of particles [44, 18, 34].

Particle tracking involves the identification of the trajectories (tracks) particles take as they move through a detector. This process can be thought of as pattern recognition, where Two-Dimensional (2D) or Three-Dimensional (3D) position measurements (hits) of the detector are used to identify trajectories. This task becomes progressively more complex as the number of background hits increases, making distinguishing the signal hits from the background difficult. Chapter 6 addresses this problem by analyzing additional detector outputs of the primary tracking detector of the Belle II experiment, the Central Drift Chamber (CDC). The goal is to identify beneficial detector outputs which provide discriminatory power.

Employing a GNN-based pipeline, the initial step includes building the input graph for the network. Chapter 7 discusses graph-building algorithms for the CDC and introduces several metrics to assess their performance in the context of particle physics models. Finally, in chapter 8, a GNN algorithm, namely, the Interaction Network (IN), is utilized for the pattern recognition in the CDC to classify specific signal edges of a graph corresponding to the path of a signal particle. The chapter evaluates different graph-building models on the edge classification task and analysis their performance in a background clean-up of hits in the CDC. A discussion on the comparability of the results in real-time implementation follows the analysis.

The overall structure of the study takes the form of nine chapters. Following an introduction to the essential physics concepts and the motivation for a novel displaced vertex trigger (chapter 2 and chapter 3), a comprehensive overview of the tracking pipeline under discussion is provided (chapter 4), along with the details on Monte Carlo (MC) event generation and event reconstruction used in the studies (chapter 5). Subsequent chapters delve into the primary analysis, presenting analyses of raw detector outputs, the graph-building algorithms with several metrics, and finally delve into the study evaluating various graphs resulting from the graph-building concerning the edge classification task and the background clean-up task in the CDC. The concluding chapter offers a summary and future prospects of the studies.

## 2. The Belle II Experiment

The Belle II Experiment is located at Tsukuba, Japan, in the High Energy Accelerator Research Organization (KEK). This national research facility for high energy physics hosts the SuperKEKB accelerator, where the Belle II experiment is conducted. The main goals of Belle II are to improve high-precision measurements of parameters of the Standard Model of Particle Physics (SM) and to search for new physics in the flavor sector [28]. This will be achieved through the analysis of the target dataset of  $50 \text{ ab}^{-1}$ , which contains approximately fifty times more data than its predecessor, the Belle experiment.

### 2.1. SuperKEKB Accelerator and Belle II Detector

The following section is based on the work of [28, 1]. The SuperKEKB accelerator, the successor to the KEKB accelerator, is operated at a center of mass energy from below the  $\Upsilon(1S)$  resonance at 9.46 GeV to the  $\Upsilon(6S)$  resonance at 11.24 GeV. Most of the data is taken at the  $\Upsilon(4S)$  resonance at  $(10\,579.4 \pm 1.2) \text{ MeV}$  [53], where the accelerator can produce a  $B$ -meson pair without remnant particles. SuperKEKB consists of the positron ring, also referred to as Low-Energy Ring (LER), and the electron ring, also referred to as the High-Energy Ring (HER). In those approximately 3 km long rings, electrons and positrons with asymmetric beam energies, 7 GeV and 4 GeV respectively, collide at the Interaction Point (IP) in the Belle II detector [31]. The asymmetric acceleration boosts the center-of-mass system, large enough to allow precise measurements of lifetimes, mixing parameters, and Charge Conjugation Parity Symmetry (CP) violation of  $B$  or  $D$  mesons.

The SuperKEKB accelerator consists of a Linear Particle Accelerator (LINAC), which injects the electrons to the HER and the positrons to the LER. The pre-injector generates the positrons at the beginning of the LINAC, where electrons irradiate a tungsten target. The positrons are led into a damping ring which reduces the emittance by radiation damping. Afterward, the positrons are reinjected to the LINAC.

The target instantaneous luminosity of the SuperKEKB accelerator is 40 times greater than the peak luminosity recorded by KEKB. This is realized mainly through a reduced beam size at the collision point, and an increased current ("nano-beam" scheme) [28, 9, 1]. The absolute delivered instantaneous luminosity is measured with Bhabha scattering. The nano-beam scheme leads to growing emittance through

## 2. The Belle II Experiment

intra-beam scattering and a short lifetime due to the Touschek effect [1]. Further technical design parameters are listed in table 2.1.

Table 2.1.: SuperKEKB machine parameters, from [28, 1].

	LER ( $e^+$ )	HER ( $e^-$ )	
Energy	4.000	7.007	GeV
Beta function at IP ( $\beta_x^*/\beta_y^*$ )	32/0.27	25/0.30	mm
Horizontal Beam Size $\sigma_x^*$	$10.2 \pm 10.1$	$7.75 \pm 7.58$	$\mu\text{m}$
Vertical Beam Size $\sigma_y^*$	59	59	nm
Horizontal emittance	3.2	4.6	nm
Luminosity	$8 \times 10^{35}$		$\text{cm}^{-2} \text{s}^{-1}$

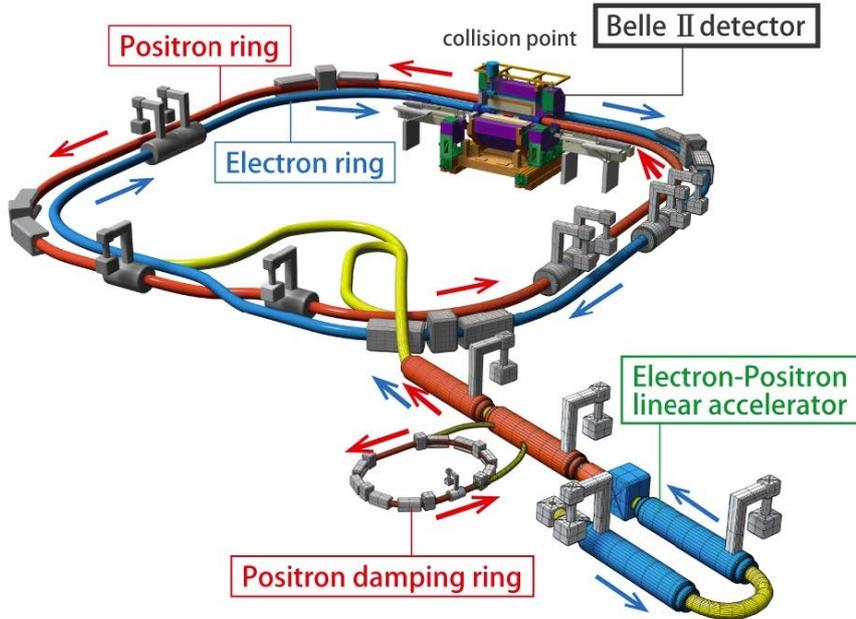


Figure 2.1.: The SuperKEKB accelerator. Taken from [52].

The Belle II detector is shown in fig. 2.2. The hermetic detector consists of five main detector systems.

- (i) The Vertex Detectors (VXD), comprising a Pixel Detector (PXD) with two layers and a Silicon Vertex Detector (SVD) consisting of a four-layer double-sided silicon-strip detector.
- (ii) The central tracking device is the Central Drift Chamber (CDC). The CDC is capable of reconstructing a full Three-Dimensional (3D) helix track. This it is the fundamental detector for this thesis.

- (iii) The Particle Identification (PID) system, consisting of a Time-of-Propagation (TOP) counter used in the barrel region. It is a special Cherenkov detector, using two-dimensional information of Cherenkov rings and the reconstructed momentum to determine the identity of the particles [41]. In the forward endcap region, an Aerogel Ring Imaging Cherenkov (ARICH) counter, a proximity focusing Cherenkov ring imaging detector with aerogel as Cherenkov radiator is used [28, 54].
- (iv) The Electromagnetic Calorimeter (ECL) detector, built out of an array of thallium-doped cesium iodide CsI(Tl) crystals. This system detects gamma rays and identifies electrons.
- (v) The  $K_L$  Muon Detector (KLM) system, placed outside of the superconducting solenoid. It composes of a sandwich of iron plates and active detector elements. The iron plates return the magnetic flux and provide additional interaction lengths.

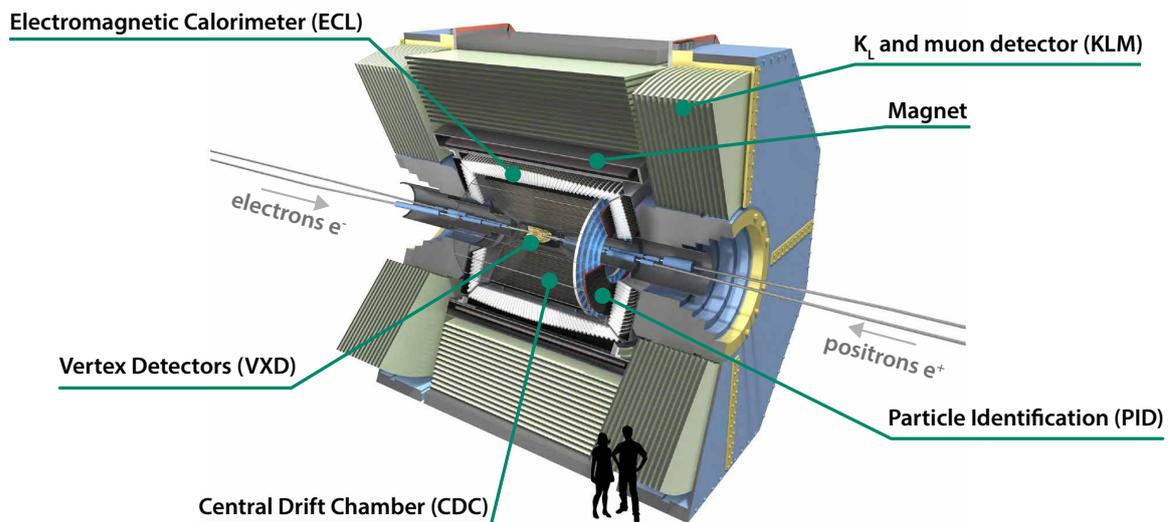


Figure 2.2.: The Belle II experiment with crucial detector components. Adapted from [52].

## 2.2. Central Drift Chamber

The following chapter is based on [28, 1]. The Central Drift Chamber (CDC) is the main tracking system of the Belle II detector. It is a cylindrical chamber containing sense wires and is filled with a gas mixture. If a charged particle traverses the detector, it ionizes the gas mixture inside the Central Drift Chamber (CDC). The electric field present within the chamber accelerates these charged particles, leading to further ionization of the gas and creating an avalanche effect. The cascade of charged particles is accumulated in the sense wires. Eventually, this leads to the detection of a signal hit.

## 2. The Belle II Experiment

Using 14 336 sense wires, a high spatial resolution can be archived, and the momenta of the passing particles can be measured. Furthermore, the CDC provides information on particle identity by measuring the energy loss ( $dEdx$ ) in the gas volume. This is especially useful for low momenta tracks, which do not reach the PID devices. The structure of the CDC will be explained in detail in the following section. Additionally, details of the readout electronics, the measurement information, and backgrounds in the CDC are presented in the following subsections.

### 2.2.1. Geometry and Structure

The following section is based on [1]. A technical drawing of the CDC is given in fig. 2.3. The CDC covers the full range of the azimuthal angles and has a polar angular acceptance of  $17^\circ$  to  $150^\circ$ . The chamber has an inner radius of 160 mm and an outer radius of 1130 mm.

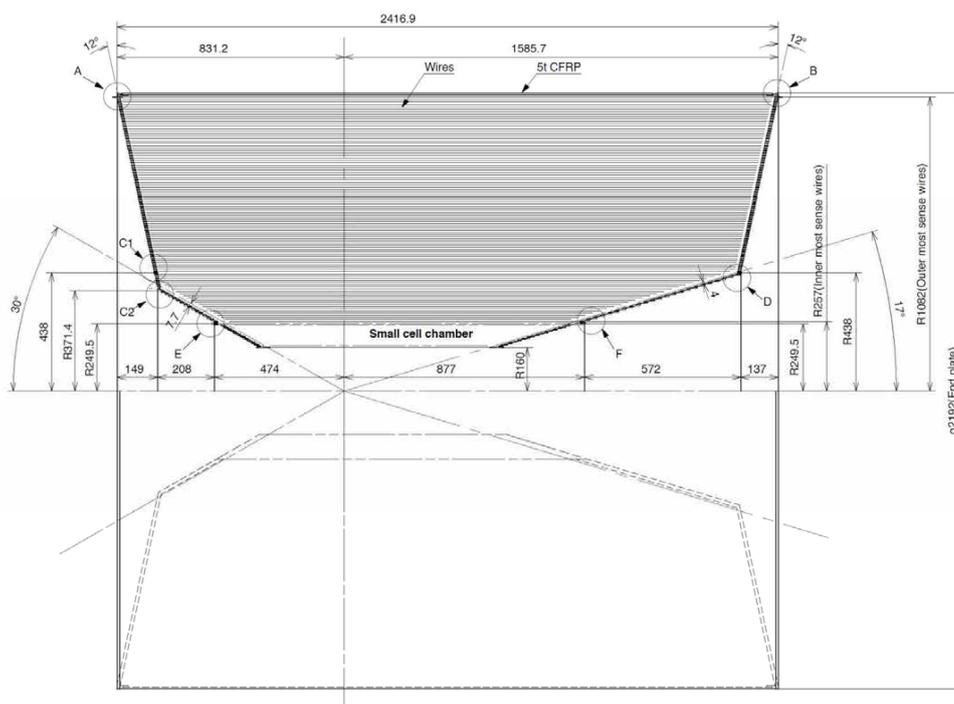


Figure 2.3.: Technical drawing of the Central Drift Chamber, displaying its structure. Taken from [1].

The 14 336 sense wires with a diameter of  $30\ \mu\text{m}$  made from tungsten are arranged around the Interaction Point (IP). Each is surrounded by a drift cell, and the electric field is generated with additional 42 240 field wires made from aluminum. The sense wires are arranged into 56 layers, further compartmentalized into nine superlayers. Each superlayer comprises six layers, and the innermost superlayer has two additional active guard layers. The superlayers are distinguished into axial ("A") and stereo

layers ("U", "V"), where the first ones are aligned, and the latter ones are skewed to the solenoidal magnetic field. This design allows the CDC to reconstruct a full 3D helix track. The two different wire types are illustrated in fig. 2.4, and additional information on the nine superlayers is displayed in table 2.2.

The chamber of the CDC is filled with a gas mixture of 50 % Helium and 50 % Ethan where a drift velocity of  $3.3 \text{ cm } \mu\text{s}^{-1}$  with a maximum drift time of 350 ns for a 17 mm drift cell is achieved. The size of the drift cells is  $\sim 18.2 \text{ mm}$  for the outer superlayers and 10 mm for the innermost superlayer.

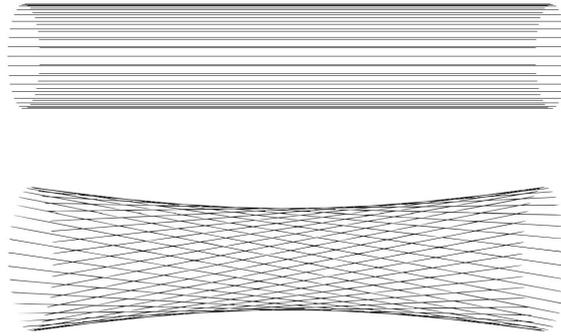


Figure 2.4.: Illustrative drawing of the axial layer (top) and the stereo layer (bottom) in the Central Drift Chamber of Belle II. Taken from [7].

Table 2.2.: Detailed information on the superlayers of the Central Drift Chamber. Taken from [1].

no.	superlayer type	layers	wires / layer	radius (mm)	stereo angle (mrad)
1	Axial	8	160	168.0 – 238.0	0
2	Stereo U	6	160	257.0 – 348.0	45.4 – 45.8
3	Axial	6	192	365.2 – 455.7	0
4	Stereo V	6	224	476.9 – 566.9	-55.3 – -64.3
5	Axial	6	256	584.1 – 674.1	0
6	Stereo U	6	288	695.3 – 785.3	63.1 – 70.0
7	Axial	6	320	802.5 – 892.5	0
8	Stereo V	6	352	913.7 – 1003.7	-68.5 – -74.0
9	Axial	6	384	1020.9 – 1111.4	0

### 2.2.2. Readout Electronics

The readout electronics of the CDC are situated close to the backward endplate and consist of a front-end digitizer [50]. It provides two main pieces of information to the Data Acquisition (DAQ) system. One is the digitized analog signal, the Analog-to-Digital Converter (ADC) count, which refers to the deposited energy of a

## 2. The Belle II Experiment

measurement in the CDC. The other is the associated time information, the Time-to-Digital Converter (TDC) count. TDC and ADC information is passed on to the trigger system to process relevant signals quickly. However, due to the high-speed requirements and limited time window of the trigger system, it has a constraint available bandwidth, and not all signals can be passed on. In table 2.3, the available information for the DAQ and the trigger system is listed. As illustrated in fig. 2.5, signals are passed from a subset of the sense wires, the trigger wires. If this Thesis refers to trigger information, it refers to the information available after the Long Shutdown 2 (LS2).

Table 2.3.: Current and anticipated information from the front-end electronics of the Central Drift Chamber passed to the Level 1 Trigger and the Data Acquisition. The projections are preliminary and may change based on future hardware considerations and decisions (Koga-san, T., personal communication, 03.11.2022).

	DAQ	TRG	TRG LS1	TRG LS2
Time window (ns)	800	32	32	32
TDC resolution (ns)	1 ns	2 ns	2 ns	2 ns
ADC resolution (points)	25	-	3	3
Hit info (layers)	6/6 (8/8)	5/6 (5/8)	5/6 (5/8)	5/6 (5/8)
TDC count (layers)	6/6 (8/8)	1/6 (1/8)	1/6 (1/8)	5/6 (5/8)
ADC count (layers)	6/6 (8/8)	-	5/6 (5/8)	5/6 (5/8)

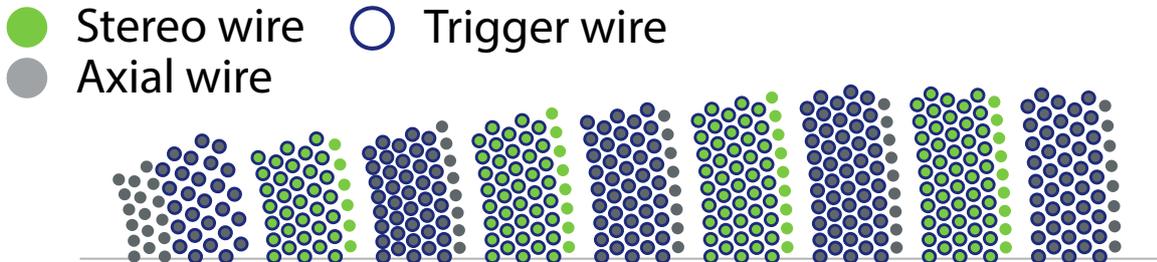


Figure 2.5.: Illustration of a fraction of the Central Drift Chamber with stereo and axial layers. The marked trigger wires are assumed to be available to the Level 1 Trigger system after Long Shutdown 2.

The Time-to-Digital Converter (TDC) count  $\text{TDC}_{\text{count}}$  is the time information of hits in the central drift chamber and has a resolution of  $\tau \approx 1$  ns. The time window for the  $\text{TDC}_{\text{count}}$  is set in the front-end electronics considering the L1-trigger latency and differs for the innermost superlayer in the CDC from the outer layers. The window is defined with  $[t_1, t_2]$ , where  $t_1$  equals the trigger delay and  $t_2$  is the delay added by the window width. The time of measurement of a hit in the CDC has a temporal shift to its actual time. The TDC count is derived from the following equation

$$\tau \cdot \text{TDC}_{\text{count}} = T_{\text{stop}} - T_{\text{evt}} - T_{\text{tof}} - T_{\text{drift}} - T_{\text{prop-delay}} - T_{\text{time-walk}}, \quad (2.1)$$

where the event time  $T_{\text{evt}}$  is referring to the actual event time. In Monte Carlo (MC) this is set to zero for a signal event. The time of flight  $T_{\text{tof}}$  is the time it takes for the particle leading to the hit in the CDC to the primary event happening in the IP. The drift time  $T_{\text{drift}}$  describes the time it takes for free charges from the gas ionization in the drift cell of a sense wire to reach the sense wire. The  $x - t$  (time-to-distance) relation is described as a polynomial function or linear function, depending on the drift region, and provides an approximation between drift time and distance [7, 1]. The  $T_{\text{prop-delay}}$  is the time, the signal propagates along the sense wire, and  $T_{\text{time-walk}}$  is the pulse-height dependent timing shift.

The Analog-to-Digital Converter (ADC) is calculated as a truncated mean of a time window to minimize the Landau tail's contribution and reduce the data size [50, 7]. The ADC value is sampled with a frequency of approximately 32 MHz and digitized. For the DAQ, this information is sampled from approximately 25 points, whereas it is anticipated to be calculated from 3 points for the trigger system with a reduced resolution.

### 2.2.3. Background in the Central Drift Chamber

Various background processes are expected in the CDC. Additional wire hits increase the difficulty of pattern recognition of charged signal tracks as they are expected from future luminosity increasements of the SuperKEKB accelerator [38, 39]. They impact the accuracy of measurements and the reconstruction efficiency of the Belle II experiment. In the following, some of the major background sources are listed. The section is based on [38].

**Cross-talk** Cross-talk occurs by interference from different channels in a detector [47].

In the context of the CDC, cross-talk refers to the impact on signals in sense wires caused by electrical coupling between spatially close sense wires. This coupling can influence or induce signals in adjacent sense wires.

**Beam Background** The high-luminosity beam of the SuperKEKB is expected to generate additional hits in the CDC. Collimation adjustment and shielding are critical for reducing beam background.

**Bhabha Scattering** In the IP, also undesirable collision processes can happen. Radiative Bhabha scattering ( $e^+e^- \rightarrow e^+e^-\gamma$ ), for example, increases the Belle II occupancy due to high cross-section but is of minor interest.

**Touschek Effect** The Touschek effect describes the loss of particles in the beam due to intrabeam scattering. It is the main limiting factor for the beam's lifetime.

**Injection Background** Due to a limited lifetime, the particle beam must be continuously refilled to keep a stable current. Particle bunches do not enter the storage ring ideally because of optical mismatches and possible injection kicker

errors. The increased amplitudes of the particles in the beamline lead to large backgrounds. Therefore, events shortly after injection are vetoed from the trigger system of Belle II.

**Synchrotron Radiation** When electrons and positrons pass the deflecting magnets close to the IP, they emit synchrotron radiation.

**Cosmic Events** Cosmic rays can generate background events in the Belle II experiment.

### 2.3. Data Taking With the Belle II Online System

In the Belle II detector, a vast number of events occur, but only a fraction is of interest for further analysis. Events like Bhabha scattering ( $e^+e^- \rightarrow e^+e^-$ ) are of secondary interest for the experiment [22]. Hence, it is not desired to save all events. In general, each subdetector transmits its signal information continuously, but the information is only saved if the online system triggers an event.

The following section is based on [1]. The trigger system consists of the Level 1 Trigger (L1 Trigger), and the High Level Trigger (HLT). The primary objective is to reduce the rate of events recorded for storage and analysis while retaining the most interesting ones for physics analysis. The system is designed to reduce the amount of data before it reaches the first storage hard disk. The L1 Trigger utilizes information from the CDC, the ECL, and the  $K_L$  Muon Detector (KLM) in a low resolution to identify relevant events. It is designed to make decisions near real-time using a Field-Programmable Gate Arrays (FPGAs), enabling fast but also configurable hardware. The L1 Trigger comprises several sub-triggers which send their signal to a Global Decision Logic (GDL), which acts as the final arbiter. The GDL uses summary information of  $\mathcal{O}(100)$  bits and fine information of  $\mathcal{O}(4000)$  bits of the sub-triggers. It processes the information and, eventually, issues trigger signals for physical events and calibration purposes. The overall latency of the L1 Trigger trigger is in the order of  $5 \mu\text{s}$  with an expected output rate of 30 kHz at target luminosity.

The online system incorporates a DAQ, which reads out the detector information and transfers the data through several steps of data processing, and finally leads to the storage system. One critical component of the system is the HLT, a more sophisticated trigger system performing fast reconstructions on Central Processing Units (CPUs). The software on the HLT is the same as the offline reconstruction software to avoid introducing additional systematics different from that in offline processing. Detailed information on the software framework of Belle II can be found in the following section.

## 2.4. Belle II Analysis Software Framework

The Belle II experiment employs a software framework known as Belle II Analysis Software Framework (basf2) [30]. It is a comprehensive software solution designed to serve several purposes required for a High Energy Physics (HEP) experiment. Belle II Analysis Software Framework (basf2) offers a high-level analysis framework to analyze data produced in the Belle II experiment and is capable of handling a wide range of tasks like generating simulated events, unpacking real raw data, and reconstructing complex events.

One of the software's key aspects is its modular structure, comprised of efficient C++ modules. Users typically create a steering file, which provides a convenient way of declaring a basf2 path to manage the modules used for the data analysis. An online database supplies the Belle II experiment conditions for each dataset. The conditions are defined as binary objects called payloads and organized within a globaltag. A globaltag offers a convenient way to manage the configuration of the analysis tools.

Event reconstruction relies heavily on particle tracking to determine particle properties, such as charge and momentum. For this, the basf2 tracking algorithm utilizes information from multiple detectors. The existing track-finding algorithm for the CDC will be discussed in greater detail below, as they are of greater interest to this thesis.

### 2.4.1. Tracking in the Central Drift Chamber

The basf2 algorithm for track finding in the CDC consists of two main modules. A global track-finding algorithm and a local track-finding algorithm. The following section is based on [7].

The global algorithm utilizes a Legendre transform method for tracking, achieving high track-finding efficiency with low fake rates. It begins by identifying patterns of hits consistent with helix trajectories. Then the position information in the axial layers is determined using the drift time information for each hit in the CDC. In this process,  $T_{tof}$  and  $T_{prop-delay}$  are initially estimated and later revised. Further details on timing information in the CDC can be found in subsection 2.2.2.

Circles in the  $r, \phi$ -plane can be mapped to a conformal space, where tracks starting at the origin are represented as straight lines, and drift circles remain circles. Thus, the particle tracks are represented as straight lines tangential to the circles. These tangents can be represented using two Legendre parameters  $\rho$  and  $\theta$  as

$$\rho = x_0 \cos \theta + y_0 \sin \theta \pm R_{dr}. \quad (2.2)$$

The drift circles are represented using their position  $(x_0, y_0)$  and radius  $R_{dr}$ . Assuming tracks to originate from the IP, track finding eventually involves identifying densely populated regions in this Two-Dimensional (2D) space. This way, multiple track

candidates can be found. In the final step, the  $z$  information is added to the  $r - \phi$  tracks by incorporating the information from the stereo layers.

The basf2 also offers a local track-finding algorithm designed to detect short tracks and tracks with displaced vertices using the cellular automaton concept. Finally, results from both algorithms can be combined, considering their specific benefits. Generally, the global track-finding algorithm is used as a baseline, while the local track-finding algorithm is not in use to date due to its high fake rates.

### 3. Motivation for Advancements in Particle Tracking Algorithms

The Standard Model of Particle Physics (SM) describes fundamental findings of particle physics with great success to this day. It provides a theoretical framework describing three of the four known fundamental forces, electromagnetic, weak, and strong interactions. Even though the model proclaims itself with precise predictions, it cannot explain all observed phenomena. Questions about the genesis of the matter-to-antimatter asymmetry in the universe, the formation and evolution of galactic structures, or the nature of the neutrino mass are unanswered until today.

Pioneering work by Vera Rubin [45] in 1980 demonstrated that the rotational velocities of spiral galaxies could not only be explained from the visible matter. The study by Rubin discusses the idea of the existence of non-luminous matter. Nowadays, this concept is well known as Dark Matter (DM). The existence of DM is supported by strong evidence over a wide range of astrophysical scales, but its particle physics nature has not yet been established. Indications of DM's existence have also been found in further observations of cosmological phenomena like gravitational lensing by galaxy clusters, the famous bullet cluster [8], and Cosmic Microwave Background (CMB) anisotropies.

The detection of DM has become one of the ultimate goals in experimental physics. With growing attention on light DM and the associated dark sector states with masses in the  $\text{MeV } c^{-2}$  to  $\text{GeV } c^{-2}$  range, low-energy, high-intensity facilities such as  $B$ -factories offer unique advantages for the exploration of such scenarios. The Belle II detector is one such facility, designed to study the properties of  $B$ -mesons and search for various DM models.

However, the baseline basf2 tracking algorithm has three limitations. First, the algorithm is optimized for tracks originating from the Interaction Point (IP), where most particles in  $B$ -meson decays will originate. This limits its ability to track SM particles with longer lifetimes, such as  $\Lambda$  and  $K_s^0$ , with significantly displaced vertices. Moreover, Beyond the Standard Model (BSM) mediator candidates can have longer lifetimes. In the following section 3.1, the concept of inelastic DM is presented, which can include such long-lived mediators. Subsequently, section 3.2 discusses possible signatures of it in the Belle II detector, highlighting the need for novel tracking algorithms to address this challenge. Second, the basf2 tracking algorithm is not yet optimized for high beam backgrounds, which will increase with future upgrades of

the SuperKEKB, for an increased instantaneous luminosity [38]. Lastly, the *basf2* tracking algorithm is not yet fully efficient for low transverse momentum ( $p_T$ ) tracks.

### 3.1. Inelastic Dark Matter with a Dark Higgs

This section delves into the concept of the Inelastic Dark Matter with a Dark Higgs (IDMDH) model and the important signatures that the Belle II detector may detect. The model discussed in this section was first presented in [12], and a more detailed explanation can be found here. Thermal DM is well motivated and has long captivated the scientific community with searches in collider experiments as in direct and indirect searches.

The IDMDH model introduces two states of DM, the  $\chi_1$  and a heavier state  $\chi_2$ . The mass splitting  $\Delta = m_{\chi_2} - m_{\chi_1}$  can be enabled with a dark Higgs field, introducing a dark Higgs  $h'$  which mixes with the Standard Model of Particle Physics (SM) Higgs. In this scenario, the inelastic DM can couple to a massive gauge boson, a dark photon  $A'$ , which mixes with the SM photon. The signature in the Belle II experiment can then be identified as two lepton tracks, respectively electrons or muons, or two hadronic tracks originating from a displaced vertex.

The model consists of seven free parameters.

1. The mass of the dark photon  $m_{A'}$
2. The mixing angle of the SM photon to the dark photon  $\epsilon$
3. The mixing angle of the SM Higgs to the dark Higgs  $\theta$
4. The mass of the dark Higgs  $m_{h'}$
5. The mass of the  $\chi_1$   $m_{\chi_1}$
6. The coupling of  $\chi_1$  and  $\chi_2$  to the dark Higgs  $f$
7. The coupling of  $\chi_1$  and  $\chi_2$  to the dark Photon  $g_X$

The simultaneous progress of expansion and cool-down of the early universe led to a drop in particle production and annihilation. Ultimately, this progress is expected to leave the thermal equilibrium of the Weakly Interacting Massive Particles (WIMPs) like the hypothetical DM. This thermal freeze-out is expected to leave distinct signatures in the CMB. Precise measurements of the CMB like these from the Planck satellite can be used to constrain the parameters of the IDMDH. These constraints can be summarized as

$$\frac{1}{4}\alpha_f^2 m_{\chi_1} < m_{h'} \lesssim m_{\chi_1} < m_{A'}, \quad (3.1)$$

with the coupling constant  $\alpha_f = f/4\pi$ .

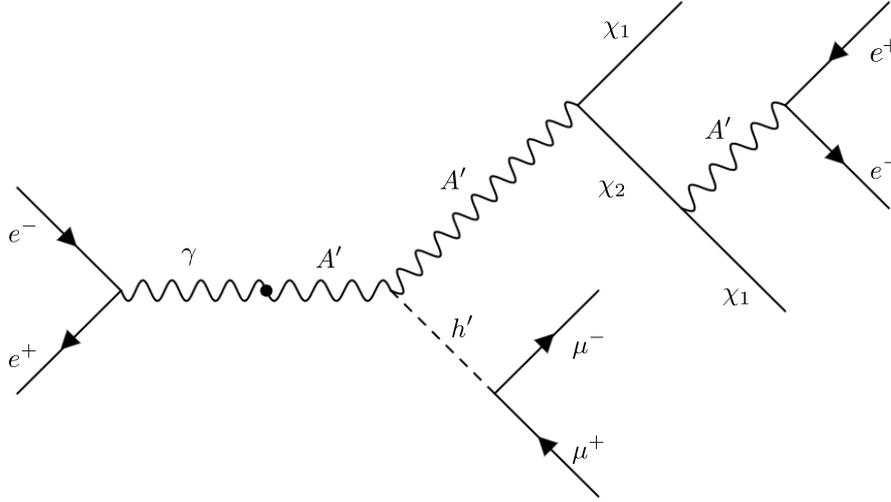


Figure 3.1.: A possible signature of the Inelastic Dark Matter with a Dark Higgs. The Standard Model of Particle Physics photon mixes with the dark photon, which couples to a dark Higgs, which decays into two muons which is the searched signal. The dark photon finally couples to the Dark Matter particles  $\chi_1$  and  $\chi_2$ .  $\chi_1$  is assumed to be stable and the heavier state  $\chi_2$  decays into a  $\chi_1$  with an  $e^+e^-$  signature. Adapted from [12].

Figure 3.1 presents one possible feynman diagram of the IDMDH. The signature of this diagram in a detector is four charged leptons as a final state, which can decay displaced from the Interaction Point (IP). The diagram sets the foundation for the signatures simulated for this thesis and details on this can be found in section 5.4. Searches for signatures of these displaced tracks have been shown to significantly gain sensitivity with a new displaced vertex trigger for the Belle II experiment [13].

### 3.2. Displaced Vertices at Belle II

Displaced vertices refer to the points in space where particles decay or interact, located at a significant distance from the primary IP. These displaced vertices commonly originate from long-lived particles of the SM but can also originate from exotic decay processes that are not accommodated by the SM. Identifying and analyzing displaced vertices can provide crucial insights into new physics phenomena, such as inelastic dark matter or other BSM scenarios, and potentially aid precise measurements of the SM.

The current experimental configuration of Belle II may miss key signatures that appear generically in a number of models, consisting of a displaced pair of electrons, muons, or hadrons. This becomes clear when investigating the reconstruction efficiency of Belle II Analysis Software Framework (basf2). Figure 3.2 shows the reconstruction

### 3. Motivation for Advancements in Particle Tracking Algorithms

efficiency of basf2 on a simulation with uniformly displaced. Details on the simulation are presented in section 5.4. The efficiency of the reconstruction software is satisfactory for particles with a vertex close to the IP. For particles with a larger displacement in the transverse plane  $\rho = \sqrt{x^2 + y^2}$ , the reconstruction efficiency decreases significantly.

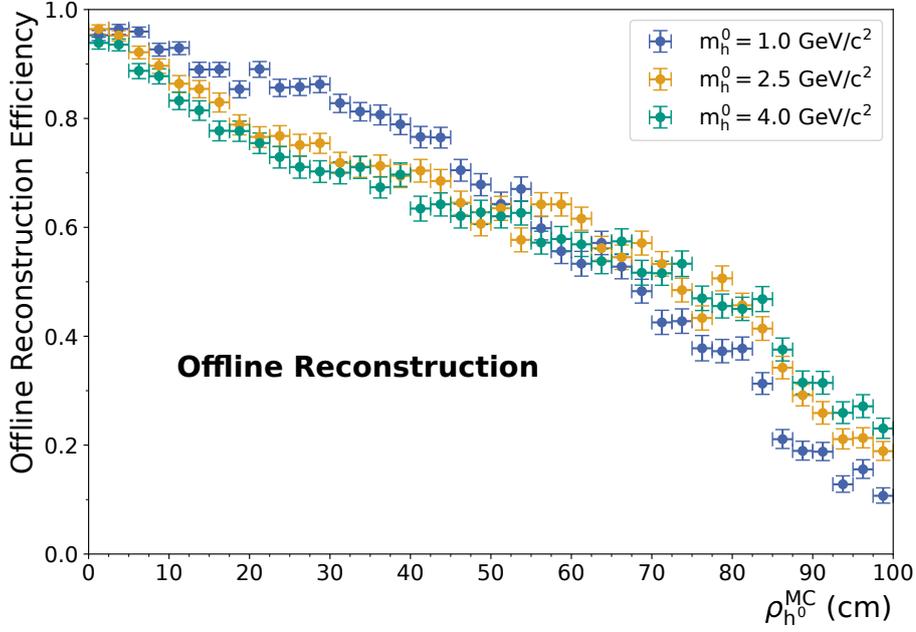


Figure 3.2.: Offline reconstruction efficiency of Belle II Analysis Software Framework in dependency of the transversal displacement. Courtesy of Lea Reuter [44].

This becomes even more obvious when examining trigger efficiency. Since the majority of SM processes are expected to have tracks originating from the IP at Belle II, most triggers are designed to trigger on such tracks. This is also necessary due to background suppression. It follows that the trigger efficiency is dropping for displaced tracks. Figure 3.3 presents the trigger efficiency of the Single Track Trigger (`stt`) which is designed to identify single tracks in the Belle II using a neural network approach [48]. In fact, the trigger efficiency of the `stt` for tracks with a displacement  $\rho > 40$  cm is close to zero.

Given the limitations of the current reconstruction software and trigger efficiency for detecting displaced vertices, it is clear that a new algorithm is needed for both online and offline identification of these vertices. Such an algorithm would improve the sensitivity of the Belle II experiment to inelastic dark matter and beyond the Standard Model scenarios involving displaced vertices [13].

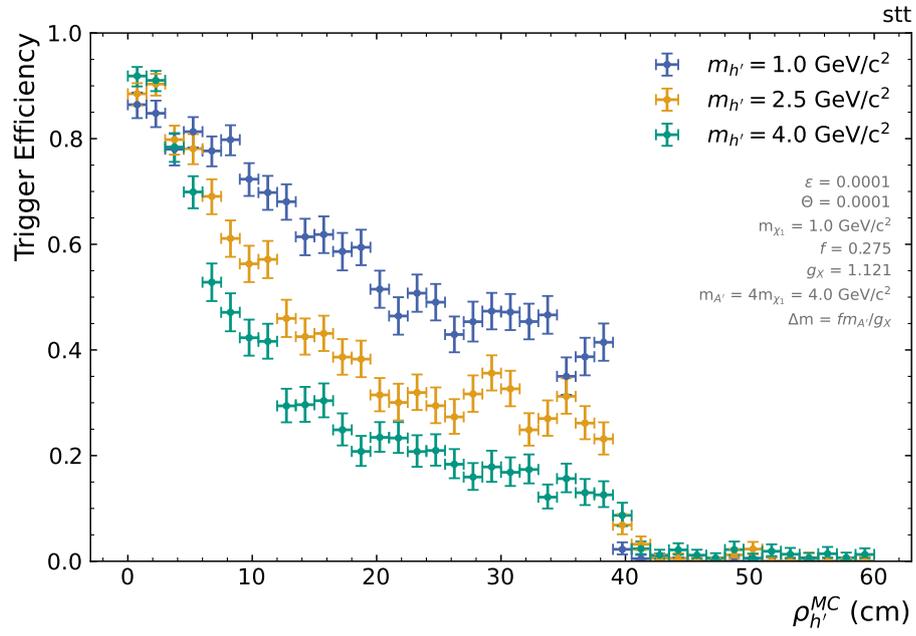


Figure 3.3.: Online Level 1 Trigger efficiency of the Single Track Trigger trigger bit in dependency of the transversal displacement. Courtesy of Lea Reuter [44].



## 4. Graph Neural Network-Based Track and Vertex Finding

In the field of High Energy Physics (HEP), the accurate representation and analysis of complex and heterogeneous data from detectors are crucial for understanding the underlying physical processes. Graph Neural Networks (GNNs) recently have emerged as a powerful and adaptable approach for representing and processing this data [25, 26, 43, 46, 10]. Particularly in particle tracking, the application of such networks has shown promising results [25].

Graph Neural Networks (GNNs) are a class of neural networks that operate on graph structures, consisting of nodes and connections between the nodes called edges. The graph-based representation can capture relationships between different entities, such as Two-Dimensional (2D) and Three-Dimensional (3D) measurements in the tracking detector. The versatility of graphs enables embedding a flexible depth of information as node features or edge attributes. Therefore, a graph can comprehensively represent the information provided by a tracking detector like the CDC at the Belle II experiment.

In the main tracking detector of Belle II the CDC, these measurements refer to measured energy deposition in sense wires called hits. These hits can result from signal particles and several background sources like beam background, cross-talk, and others. A detailed overview of the major backgrounds in the CDC is presented in subsection 2.2.3.

A novel graph neural network track and vertex finding pipeline is proposed by *Lea Reuter* and *Torben Ferber* to address the growing challenge of the track finding task [44, 18, 34]. The *GNN-pipeline* is introduced in the following section. The Interaction Network (IN) is an integral part of the *GNN-pipeline* and is used for the analysis in this thesis. It is introduced in detail in section 4.2.

### 4.1. Pipeline for a Graph Neural Network-Based Tracking Approach

The baseline basf2 tracking algorithm, presented in subsection 2.4.1, exhibits three main non-optimal features that the new *GNN-pipeline* aims to address.

#### 4. Graph Neural Network-Based Track and Vertex Finding

- (i) This algorithm is optimized for tracks originating from the IP, which is not ideal for Standard Model of Particle Physics (SM) particles with longer lifetimes, such as  $\Lambda$  and  $K_s^0$ , that can have significantly displaced vertices.
- (ii) The baseline algorithm is not yet optimized for high beam backgrounds as they are expected with the increasing luminosity.
- (iii) The tracking algorithm is not fully efficient for low transverse momentum ( $p_T$ ) tracks.

The *GNN-pipeline*, illustrated in fig. 4.1, is intended for both online trackings, referring to real-time tracking during data taking in the Belle II experiment, and offline tracking, for post-data-taking analysis. Each application must address distinct challenges. The online application requires low latency and suitable computational demand, while the offline approach must deliver precise results, competing against the established and accurate baseline algorithms.

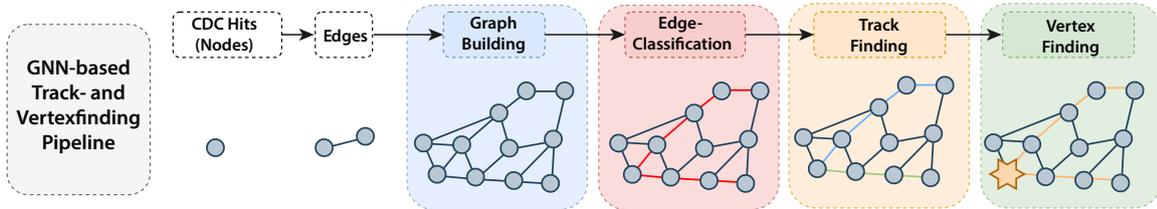


Figure 4.1.: Illustration of the Graph Neural Network-based Track- and Vertexfinding pipeline. Adapted with courtesy of Lea Reuter [44].

The *GNN-pipeline* is separated into four modular steps, ensuring flexibility and adaptability:

**Graph-building** The initial step is constructing the input graph for the Graph Neural Network (GNN). The graph-building process is essential for transforming raw data into a structured graph, selectively connecting nodes and edges to strike a balance between incorporating valuable physical knowledge, adhering to hardware limitations for efficient processing, and maximizing accuracy in subsequent steps of the GNN-pipeline.

**Edge-Classification and Background clean-up** Background clean-up plays a pivotal role in analyzing particle collisions, as it ensures the quality and reliability of results by isolating imprints of signal particles in the measurements. The background clean-up is realized using the Interaction Network (IN), which classifies edges of an input graph to identify nodes corresponding to hits in the CDC corresponding to a signal particle and determine cleaned events.

**Track-Finding** In the third step, an object condensation [26] method classifies hits in the CDC, identifying individual particles. Simultaneously this GNN-based algorithm determines track parameters. The first implementation of this algorithm for Belle II is presented in [18].

**Vertex-Finding** In the final step, an advanced algorithm for vertex-finding is intended to be employed. To this date, it is unclear if it can be incorporated into the preceding step.

The innovative Graph Neural Network-based Track and Vertex Finding pipeline represents a leap forward in high energy physics research, providing a powerful tool to overcome the challenges the Belle II experiment faces. This thesis concentrates on advancing research in graph-building by investigating various graph-building methods and determining suitable metrics for evaluating resulting graphs. Additionally, a background clean-up using the IN is implemented and examined to deepen the understanding of graph-building. While the remaining two steps of the proposed pipeline are not this thesis’s primary focus, advancements in graph-building and the in-depth analysis of the IN contribute to the overall improvement and development of the *GNN-pipeline*.

## 4.2. Interaction Network

The goal is to clean up events in the CDC by isolating hits that can be associated with a specific particle of interest against background hits. These signal hits are generally energy depositions in the CDC resulting from the passage of signal particles, such as those produced in the decay or interaction processes being studied, and not caused by background processes like beam-induced noise or detector artifacts. Traditional tracking algorithms have several drawbacks, such as their iterative nature, which prevents parallelization, the increased difficulty of pattern recognition with more background hits [38], and a complexity that is worse than non-linear.

To address these issues, Geometrical Deep Learning (GeoDL) is utilized, allowing for parallelization on hardware and simultaneous analysis of entire events. GeoDL aims to generalize and adapt the core principles of deep learning to work with non-Euclidean data structures, such as graphs and other irregular structures. Graph Neural Networks (GNNs) are a popular example of GeoDL, designed specifically to operate on graph-structured data.

One example of a GNN is the Interaction Network (IN), first introduced in [6] and later adapted for particle tracking by [14], with its initial application in the *TrackML* challenge [5, 4], showing promising results.

The motivation for utilizing the Interaction Network (IN) for the background clean-up in the context of the Belle II experiment stems from its ability to represent and reason about objects and relations in intricate systems. The IN combines a structured model with deep learning, allowing it to exploit explicit knowledge of relations among objects while providing highly scalable learning. This adaptability is achieved through adjustable and trainable parameter space. Additionally, the network’s flexibility in input sizes and dimensions allows for versatile use cases, such

as online or offline applications, by adjustment of the available input information. Furthermore, the tracking at the CDC can benefit from flexible input sizes, as patterns of spatially distributed hits in the CDC can vary significantly due to the involvement of different particles and background sources in a single event picture.

In summary, the IN addresses the limitations of traditional tracking algorithms in the Belle II experiment by leveraging the power of GeoDL. This versatile and adaptable approach may significantly improve pattern recognitions and the subsequent particle tracking in the Belle II experiment.

#### 4.2.1. Mathematical Description of the Implemented Interaction Network

This section presents the mathematical foundation of the IN implemented in this thesis. The IN is a specific class of GNNs called message-passing networks, which are designed to learn and generalize from structured graph data. The core idea behind message-passing networks is the iterative exchange of information between neighboring nodes (receivers and senders) to accumulate information in the receiving nodes. This process can be extended by incorporating relational information between two nodes in the form of edge attributes. The IN implementation focuses on predicting edge states, which are then used for edge classification and background clean-up in the *GNN-pipeline*.

The following section provides a general mathematical introduction to the IN and is based on [6, 14]. Detailed information regarding the implementation in this thesis is presented in subsection 8.1.1.

Initially, the input to the IN is a graph  $G = (V, R)$  with  $N_V$  nodes  $V$  and  $N_R$  directed edges  $R$ . They are defined as

$$V = \{v_i\}_{i=1,\dots,N_V}, \quad R = \{r_k\}_{k=1,\dots,N_R}. \quad (4.1)$$

Each node  $v_i$  and each edge  $r_k$  is defined as an input vector with a fixed dimension. The IN uses a relation-centric function

$$f_R(b_k) = r'_k, \quad \text{where } b_k = (v_i, v_j, r_{i,j})_k \quad (4.2)$$

to describe the future states of a single edge  $r'_k$ , focusing on the relations between objects. The index  $k$  denotes the iteration over all neighboring nodes  $\mathcal{N}(v_i, v_j)$ , connected by a directed edge  $r_{i,j} = r_k$ . The interaction between two nodes is represented using  $b_k$ . The relational model

$$\phi_R(B) = R' = \{r'_k\}_{k=1,\dots,N_R}, \quad \text{with } B = \{b_k\}_{k=1,\dots,N_R}, \quad (4.3)$$

applies  $f_R$  to each  $b_k$  and gives the future state of all edges. Now the output of the relational model is aggregated by a function

$$a(\phi_R(B)) = \tilde{R}' = \{\tilde{r}'_i\}, \quad \text{with } \tilde{r}'_i = \sum_{j \in \mathcal{N}(i)} r'_{ij} \quad (4.4)$$

collecting all effects applying to a receiver (node) by collecting all relational information from the connected nodes  $\mathcal{N}(i)$ . In this Thesis, a summation function is used in the aggregation step. After the aggregation, an object-centric model

$$\phi_O(V, \tilde{R}') = V' \quad (4.5)$$

is used to compute an output for each node. The object-centric model focuses on the individual objects (nodes) in the graph. Finally, a second relation model  $\phi_{R,2}$  is applied, which classifies the edges

$$\phi_{R,2}(B'), \text{ with the input } B' = (v'_i, v'_j, r'_{i,j})_k. \quad (4.6)$$

The second model uses the output of the first relational model and the output of the object model as an input. The IN model can be described as

$$\text{IN}(G) = \phi_{R,2}(\phi_R(B), \phi_O(a(G, \phi_R(B)))) \quad (4.7)$$

leading to a final one-dimensional output with  $N_R$  edge weights. The final output can be used to classify edges by defining a threshold.

### 4.2.2. Pytorch Geometric

PyTorch Geometric (PyG) is an extensive library of the deep learning framework PyTorch, focused on geometric deep learning tasks. It allows making usage of GPUs, which allows for highly efficient parallelization of computational tasks and a wide range of pre-implemented base classes for the ease of an own implementation of a GNN. PyTorch and the PyG are used to embed the proposed *GNN-pipeline* for analysis use-cases. The whole IN is embedded in a `MessagePassing` network, a base class from PyG, which helps in creating GNN by automatically taking care of message propagation. Further details on the implementation can be found in subsection 8.1.1.

### 4.2.3. Training Theory

The primary objective of training a neural network is to learn the probability distribution  $P$  that represents the true underlying distribution of the problem being analyzed. In this context, a sample of data points with known labels,  $y$ , is provided by the simulation. The network aims to learn a hypothesis  $Q$ , which approximates the true label distribution  $P$ . The goal is to minimize the divergence of the two distributions  $P$  and  $Q$ . The Kullback-Leibler divergence, a well-known measure of the divergence between two distributions, is defined as

$$D_{\text{KL}}(P||Q) = \sum_i (P_i \log(P_i) - P_i \log(Q_i)). \quad (4.8)$$

In most training cases, the distribution  $P$  is not entirely known. For binary classification problems, the true label distribution  $P$  can be represented by a Bernoulli

distribution with the parameter  $y$ . Similarly, the predicted probability distribution can be represented as the Bernoulli distribution with the parameter  $p$ . There are only two possible outcomes. Success (1) with probability  $y$  and failure (0) with probability  $1 - y$ . The predicted probabilities for these outcomes are  $p$  and  $1 - p$ , respectively, the output of the sigmoid function with a value between zero and one. Thus, the Kullback-Leibler divergence formula in this discrete case can be expressed as:

$$D_{\text{KL}}(P||Q) = y \log(y) - y \log(p) + (1 - y) \log(1 - y) - (1 - y) \log(1 - p). \quad (4.9)$$

As the first and third terms can be assumed to be constant, minimizing the Kullback-Leibler divergence is equivalent to minimizing the binary cross-entropy loss defined as

$$L_{\text{cross-entropy}} = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)). \quad (4.10)$$

By minimizing the binary cross-entropy loss, the divergence between the true label distribution  $P$  and the predicted probability distribution  $Q$  is effectively minimized.

## 5. Monte Carlo Simulation and Data Sampling for Event Analysis

This Thesis employs Monte Carlo (MC) simulated events for various studies, including the input feature comparison, the graph-building analysis, and for training and evaluation of the Interaction Network (IN).

In this thesis, a MC simulation of dimuon events is performed for the input feature analysis. Section 5.2 presents the dimuon simulation and reconstruction, while section 5.3 presents details of the reconstruction used for the corresponding experimental data.

Section 5.4 outlines the MC simulation of the Inelastic Dark Matter with a Dark Higgs (IDMDH)-model introduced in section 3.1, employed for evaluating the graph-building and the training of the IN. The truth information supplied by the MC simulation is a fundamental aspect of the analysis.

### 5.1. Introduction to Monte Carlo Simulation at Belle II

For the MC simulation at the Belle II experiment, Belle II Analysis Software Framework (basf2) is used. The MC simulation can be divided into three steps.

1. Event generation, where particles are simulated based on an underlying physics theory, such as the Standard Model of Particle Physics (SM). This begins with the interaction of the positron and the electron at the Interaction Point (IP). There are various generators for different use cases depending on the simulated models and involved particles.
2. Simulation of particle's propagation through the various components of the Belle II detector. This step includes a simulation of the energy deposition in the detector, as well as the response of the detector. Detailed information on the complex geometry of the detector is needed for that step. Geant4 simulation software is used to simulate particle interaction with the virtual detector [2], such as ionization, bremsstrahlung, pair production, Cherenkov radiation, and others.
3. Reconstruction of the raw data. The event simulation is followed by a reconstruction where the generated raw data is processed and reconstructed. The

basf2 mainly uses the same algorithms for reconstructing real detector data and simulated data, ensuring a consistent analysis process. This step may include the reconstruction of particle properties.

Adding background to the physics processes is achieved by applying overlays to the simulated detector response. The naming of background overlays of the Belle II experiment can be assigned to major commissioning phases [39].

**Phase 1** This phase focused on the commissioning of the SuperKEKB accelerator.

**Phase 2** From March 2018 to July 2018, the full Belle II detector was installed except for the Vertex Detector (VXD). First background measurements ensured a safe installation of the sensitive VXD [35].

**Phase 3** The ongoing phase of the Belle II experiment started in March 2019 and is dedicated to physics data taking.

## 5.2. Dimuon Sample Generation for Feature Analysis with Monte Carlo Simulation

The signal consists of dimuon events simulated with the standard dimuon generator of basf2, the KKMC generator[23, 24] in version 4.19. Specifically, the basf2 implementation with the module `add_kkmc_generator`, with a final state of  $\mu^+\mu^-$ . The latest available pre-release background overlays were used in this analysis, namely the pre-release-07 backgrounds. They are expected to provide the most accurate simulation of Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts of the Central Drift Chamber (CDC). One drawback is that a pre-release background might still undergo some verification tests and possible improvements before the official release. This analysis, therefore, does not use an official simulation of the Belle II experiment.

Recent updates in background simulation include a more refined physical description of the collimators and beam pipe, as well as a first implementation of collimator tip scattering embedded into a new sequential tracking framework [37]. These modifications significantly improve agreement between data and simulation for the CDC hit-wire rate, which is now within one order of magnitude for luminosity and beam backgrounds.

The Belle II experiment continuously collects data over an extended period. The detector's performance, calibration, and environmental conditions may change throughout this time. To account for these, the Belle II experiment uses run-dependent background files. These are background files generated from real data to account for varying backgrounds that may result from varying background conditions. Utilizing these background files offers a robust verification of the analysis results. Therefore two simulations were performed, one with run-independent background files and one

using run-dependent background files. Details on the configuration of basf2 are listed in table 5.1 and in table 5.2. The number of simulated events was adjusted to the number of available background files to prevent a re-usage of the available background files. For the run-independent analysis, 80 000 events are simulated, and for the run-dependent analysis, 14 778 events are simulated.

Table 5.1.: Simulation parameters employed for the run-independent dimuon samples with the Belle II Analysis Software Framework simulation.

Parameter	Value
Basf2 release	82bd9e2155301de47454988d73f0169cfdcf27563 (development branch)
Conditions global tag	mc_production_MC15ri_a, main_2022-07-05
Experiment number	1003 (early-phase-3)
Background files	pre-release-07-00-00a

Table 5.2.: Simulation parameters employed for the run-dependent dimuon samples with the Belle II Analysis Software Framework simulation.

Parameter	Value
Basf2 release	release-06-01-08
Conditions global tag	hlt_filters_exp24, mc_production_MC15rd_a_exp24_bucket30, data_reprocessing_prompt, online
Experiment number	24
Run number	888
Background files	run-dependent background

### 5.3. Data Sample Reconstruction for Feature Analysis

The Belle II uses two identifiers to organize the data collected during the operation of the experiment. The experiment number does refer to a group of data collected under similar conditions within a distinct phase of the experiment with a specific detector and accelerator configuration. The run number refers to a distinct period of continuous data taking, operating with stable beam and detector conditions. Several runs from different experiments are investigated for the input feature analysis. The selection on the runs is described in chapter 6. The unpacking of the raw data is handled by basf2, as is the reconstruction of the data. Table 5.3 does present details on the configuration on basf2 used.

Table 5.3.: Reconstruction parameters employed for reconstruction of the Belle II raw data with the Belle II Analysis Software Framework.

Parameter	Value
Basf2 release	release-06-00-14
Conditions global tag	<code>data_reprocessing_prompt</code>
Experiment (run) number	20 (672,874), 21 (116,128), 22 (32,546), 24 (888)

## 5.4. Simulating Inelastic Dark Matter With a Dark Higgs Signal Using Monte Carlo Simulation

One of the main objectives of the *GNN-pipeline* is to optimize the online and offline reconstruction efficiency of particles with displaced vertices. Therefore, a signal MC simulation with simulated displaced vertices is used to evaluate the graph-building and the edge classification performance of the Graph Neural Network (GNN)-pipeline. The events were generated on the basis of the IDMDH model presented in section 3.1 with additional constraints. The coupling of the Dark Matter (DM) particles to the dark Higgs is fixed to

$$f = \sqrt{4\pi\alpha_f} \approx 0.2476, \quad (5.1)$$

and the coupling constant to the dark photon is set to

$$g_X = \sqrt{4\pi\alpha_X} = 1.12, \quad (5.2)$$

where  $\alpha_X$  is the dimensionless coupling constant. Additionally, the total mass of the considered particles in an event is the constraint to the beam energy of SuperKEKB accelerator

$$m_{\chi_1} + m_{\chi_2} + m_{h'} < 10.58 \text{ GeV } c^{-2}. \quad (5.3)$$

This work focuses on the dilepton pair created in the dark Higgs decay. To reduce the complexity of the events, the lifetime of the  $\chi_2$  is set to a high value in the simulation to guarantee its decay outside of the Belle II detector. For the event generation of the purely leptonic final states, MadGraph5 is used [3]. For this work, solely dimuon final states are simulated and investigated. They have comparatively high branching fractions to other leptonic final states in this model and are the primary expected leptonic signal. After generating the events, they are simulated and reconstructed using basf2 with the same settings as in the official MC15 campaign. The corresponding settings are listed in table 5.4.

This analysis uses background overlays with *early Phase 3* and *nominal Phase 3* background. The *early Phase 3* overlay thus refers to background conditions expected in the early stages of the Belle II experiment. The *nominal Phase 3* overlay refers to background conditions met at design luminosity. Table 5.5 does list optics and beam conditions of the Belle II from June 2021. In contrast, table 5.6 shows the expected beam parameters, with largely increased instantaneous Luminosity.

5.4. *Simulating Inelastic Dark Matter With a Dark Higgs Signal Using Monte Carlo Simulation*

---

Table 5.4.: Simulation parameters employed for the Inelastic Dark Matter with a Dark Higgs simulation with Belle II Analysis Software Framework.

Parameter	Value
Basf2 release	release-06-00-08
Conditions globaltag	mc_production_MC15ri_a, release-06-00-07
Experiment number	1003 (early-phase-3)
Experiment number	0 (nominal-phase-3)
Background files	release-05-01-15

Table 5.5.: Parameters for the optics and beam conditions of the Belle II experiment, from the luminosity background study on June 16, 2021, taken from [39].

Parameters	LER	HER
Beam current (mA)	732.6	647.2
Number of bunches	1174	1174
Beam size $\sigma_x$ ( $\mu\text{m}$ )	184.6	151.0
Beam size $\sigma_y$ ( $\mu\text{m}$ )	60.7	36.2
Beam size $\sigma_z$ (mm)	6.5	6.8
$\beta_x^*/\beta_y^*$ at IP (mm)	32/0.27	25/0.30
Pressure (nPa)	88.7	24.3
Luminosity $\mathcal{L}$	$2.6 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$	

Table 5.6.: Comparison of expected beam parameters for SuperKEKB before Long Shutdown 2 (Januar 2027) and target parameters (2031), taken from [39].

Parameters	Jan 2027	Jan 2031
Beam current (LER/HER) (A)	2.52/1.82	2.80/2.00
Number of bunches	1576	1761
Beam size $\sigma_z$ (LER/HER) (mm)	8.27/7.60	8.25/7.58
Emittance $\epsilon_x$ (LER/HER) (nm)	4.6/4.5	3.3/4.6
Emittance $\epsilon_y/\epsilon_x$ (LER/HER) (%)	1/1	0.27/0.28
$\beta_x^*$ at IP (LER/HER) (mm)	60/60	32/25
$\beta_y^*$ at IP (LER/HER) (mm)	0.6/0.6	0.27/0.3
Luminosity $\mathcal{L}$ ( $\text{cm}^{-2} \text{ s}^{-1}$ )	$2.8 \times 10^{35}$	$6.3 \times 10^{35}$

The simulated dark Higgs masses used for this analysis where the range of  $m_{h'} = [1.1, 1.2, \dots, 4, 0]$   $\text{GeV c}^{-2}$  for the *nominal Phase 3* background. The additional analysis provided for *early Phase 3* background used a simulated Higgs mass in the range of  $m_{h'} = [0.6, 0.7, \dots, 4, 0]$   $\text{GeV c}^{-2}$ . The lifetime of the dark Higgs  $h'$  determines the origin of the dimuon signal as they originate from its decay vertex. The simulation contains uniformly displaced vertices in a radial distance from the IP of  $r = [0, 100]$  cm. The vertices point to the IP, meaning that the vector sum of the momenta of the two signal muons points back to the IP. The simulated Higgs mass influences the signal in the Belle II detector in two ways. Firstly, the energy of the signal muons is directly affected by the mass of the dark Higgs. Secondly, the opening angle between the two muons is influenced by the dark Higgs mass, with an increased Higgs mass leading to a larger opening angle in the lab frame.

MC truth matching refers to the association of hits in a particle detector to the corresponding MC particle generated in the simulation. This is an established method to determine the performance of reconstruction algorithms, to evaluate the accuracy of analysis, or to examine the efficiency of selections. In MC truth matching, the term *truth* refers to the information about the generated particle, like its type, energy, and momentum. In the context of the CDC, the probability of overlap of a signal particle hit and a background hit is non-negligible, particularly in the *nominal Phase 3* background scenario with a higher number of background hits. This makes the truth-matching process more challenging and necessitates the development of more sophisticated methods for handling such situations. To address this issue, a new feature, storing up to five different particles for each wire hit, was implemented to build graphs and benchmark the GNN algorithm more robustly, considering the challenges posed by the particle interactions within the CDC.

## 6. Investigation of Central Drift Chamber Signal Characteristics as Potential Input Features for Neural Networks

Machine learning algorithms heavily rely on input features to make accurate predictions. However, in real-time reconstruction scenarios, input feature selection becomes particularly important due to limitations, such as time constraints, limited memory on the Field-Programmable Gate Arrays (FPGAs), and limited data rates. To ensure accurate and efficient real-time reconstruction, it is crucial to select input features that possess high relevance and good discriminatory power. Additionally, the input features should be orthogonal to each other to avoid redundancies, as adding an input feature involves a trade-off between computational overhead and information gain. In a previous study, the use of digitized signal information as additional input features has shown great improvements for a classification task in the Central Drift Chamber (CDC) [34]. More precisely, Time-to-Digital Converter (TDC) count and Analog-to-Digital Converter (ADC) count information for each hit was included. Resolution and scope of the features were taken, as available at the analysis level. However, it has not been shown that this information provides discriminatory information in real data. The following chapter investigates the discriminatory potential of the two detector signals and evaluates the consistency of their behavior in Monte Carlo (MC) simulation to data. It is also discussed how both ADC and TDC information can be used at the trigger level with reduced resolution.

### 6.1. Methodology for Comparing Signal Information of the Central Drift Chamber in Monte Carlo Simulation and Belle II Data

In this comparison, ADC and TDC information of dimuon ( $ee \rightarrow \mu\mu$ ) events were investigated. For this, a small subset of the available Belle II data runs is used to compare real data with simulated MC events. The runs are selected based on the availability of the skim `mumu_tight` as a starting point for a tight selection on dimuon samples. The skim already provides some tight constraints on event properties, as highlighted in table 6.1.

Table 6.1.: Summary of the `mumu_tight` High Level Trigger skim [30].

Condition	Description
$nTracks = 2$	Events with two reconstructed tracks.
$enECLTrack1 < 0.5$ $enECLTrack2 < 0.5$	Each track with a cluster in the ECL with an energy deposition smaller than 0.5 GeV.
$EMumutot < 2 \text{ GeV}$	The combined total energy of the ECL clusters and the energy of the two tracks in the ECL are restricted to a maximum deposited energy.
$  \theta_+^* + \theta_-^*  - 180^\circ  < 10^\circ$ $ 180^\circ +  \phi_+^* - \phi_-^*   < 10^\circ$	Without additional particles in the interaction, the Muons are expected to fly back-to-back in the rest frame of the dimuon system.
$p_{P1} > 0.5 \text{ GeV}/c$ $p_{P2} > 0.5 \text{ GeV}/c$	The skim demands the momentum of the tracks in the CMS frame to be larger than 0.5 GeV/c each.

In the following subsection 6.1.1, the selection of events given by the skim `mumu_tight` will be discussed. In subsection 6.1.2, track and hit distributions of the selection are investigated.

### 6.1.1. Selection of Events

For the purpose of conducting a fast and broad comparison with runs from multiple experiments, events with two data files per run have been chosen. However, it is important to note that these shorter runs and small subsets have the drawback that they may be less representative of events in Belle II. This is due to potential instabilities that could lead to these shorter data collections, which are not accounted for in simulations. Nonetheless, limited computing resources make this selection necessary for this analysis. The runs presented in this thesis were randomly selected from available data of the Belle II experiment in order to address this issue. Self-generated MC events are compared to real data. Additional simulation details can be found in chapter 5.

A strict selection process is necessary to obtain a clean sample of real data with dimuon events. The MC events generated for this analysis include a signal  $ee \rightarrow \mu\mu$  signature with a simulated background. Events taken from the skimmed data may contain events without a dimuon signal. Therefore, higher efficiency is expected when selecting the signal simulation compared to the data.

Events are selected based on low-level information like Level 1 Trigger (L1 Trigger) signals and raw detector measurements, and high-level, processed information. This high-level information is derived from the reconstruction using the Belle II Analysis Software Framework (basf2) framework and through calculations.

Each L1 Trigger output bit represents a decision taken from the L1 Trigger system on whether an event meets specific event criteria [22]. More information on the Belle II online systems can be found in section 2.3. The Belle II detector provides many bits, each covering a specific range of physics events. The trigger bit `stt` is triggering on events that contain at least one track with estimated momentum  $p > 0.7 \text{ GeV}/c$  found from the Neuro 3D track trigger [40]. The `stt` bit can be vetoed from the ECL Bhabha veto signal and the SuperKEKB injection veto.

The trigger is expected to be better described by simulation in the barrel region, where higher efficiency is anticipated. Therefore, the reconstructed tracks' angle  $\theta_{\text{track}}$  are constrained to be in the range of  $32.2^\circ < \theta_{\text{track}} < 128.7^\circ$ . The track-based efficiency of the `stt` bit selection is shown in fig. 6.1.

The events that result from reconstruction may consist of various reconstructed particles, not just two reconstructed muon tracks. To address this, only events with two tracks, each with a muon Particle Identification (PID)  $P(\mu) > 0.9$ , are selected. The reconstructed invariant mass of the dimuon system is selected to be  $m_{\mu\mu} > 8 \text{ GeV } c^{-2}$ , to suppress dimuon events involving intermediate particles.

For the analysis, hits in the CDC are selected, which have a TDC count in a window between  $4256 \leq \text{TDC}_{\text{count}} \leq 5024$ . This is the same TDC window that is currently used during data-taking at Belle II [30]. Any hits outside this window are excluded from the comparison. Table 6.2 shows exemplary selection efficiencies for

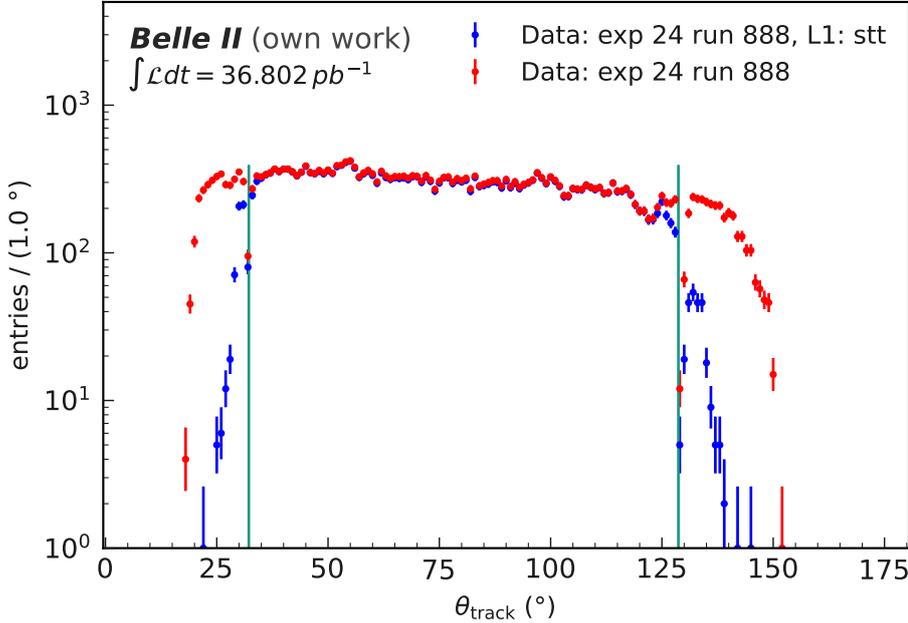


Figure 6.1.: Comparison of the angular track distribution, before and after Level 1 Trigger bit selection of experiment 24, run 888. The green lines denote the edges of the barrel region. The TDC values are in the interval  $[4256, 5024]$ , the dimuon mass is  $m_{\gamma\gamma} > 8 \text{ GeV}$  and the muon PID is  $P(\mu) > 0.9$ .

experiment 24, run 888, and the MC simulation. As expected, the selection shows a higher efficiency for the MC simulation. The selection efficiency of other runs can be found in appendix A. No corrections, such as PID correction, tracking correction, and energy calibration, are applied to the events. Thus, this analysis may show differences when comparing the reconstructed properties of simulation and data.

Table 6.2.: Selection efficiency comparison for the skimmed data sample of experiment 24, run 888 and simulation. The data consists of  $36.802 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 24, run 888 (%)	MC simulation (%)
1	Theta cut	$76.40^{+0.27}_{-0.27}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection <i>stt</i>	$94.91^{+0.16}_{-0.16}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$87.80^{+0.24}_{-0.24}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$94.69^{+0.18}_{-0.18}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.37^{+0.19}_{-0.19}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$56.9^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

### 6.1.2. Comparison of Track and Hit Feature Distributions

To evaluate the accuracy and reliability of this analysis, it is essential to compare the track parameters of the reconstructed Muons in simulation and data. This comparison serves several purposes:

- (i) It helps to identify discrepancies between the experimental and the simulated data sets.
- (ii) It allows to review and refine the event selection criteria.
- (iii) It aids in verifying the accuracy of the simulation.

This section compares the reconstructed tracks' properties for MC simulation and data to evaluate their agreement. The data used in this analysis come from experiment 24, run 888. This specific run is used as an example, but the thesis analyzes multiple runs, and their distributions can be found in appendix A. All histograms in this analysis are normalized to the integrated luminosity  $\int \mathcal{L} dt$  of the corresponding run. Each MC event  $n_i$  is weighted following the equation

$$w_i = \frac{\int \mathcal{L}_{\text{data}} dt \cdot N_{\text{sim\_events}}}{\sigma_{ee \rightarrow \mu\mu}}, \quad (6.1)$$

with a cross-section of  $\sigma_{ee \rightarrow \mu\mu} = 1.148$  nb. The statistical error is approximated for a bin  $N_k$  with  $n$  entries as

$$\sigma_{N_k} = \sqrt{\sum_i^n w_i^2}, \quad (6.2)$$

with the corresponding weight  $w_i$  of each bin entry.

The longitudinal momentum distribution  $p_z$  and the transversal momentum  $p_T$  are present in fig. 6.3 and fig. 6.2, respectively. For completeness, the individual momentum distributions  $p_x$  and  $p_y$  are available in appendix A.1. The  $p_z$  distribution is symmetrical for the dimuon events but shifted due to the asymmetrical beam energies of the SuperKEKB accelerator. The transversal momentum  $p_T$  distribution peaks just above 5 GeV/c corresponding to half the center-of-momentum energy of the SuperKEKB accelerator. This peak is anticipated since the collision energy is distributed over the two tracks.

Figure 6.4 shows the invariant mass distribution of the dimuon system. The invariant mass distribution of the dimuon system is expected to peak at the center-of-mass energy of the electron-positron collision because the collision energy is expected to be transferred to the two muons in the analyzed process. The Belle II experiment operates at a collision energy of approximately 10.58 GeV. The PID distribution of the muon probability  $P(\mu)$  is shown in fig. 6.5. After the selection, no tracks with a PID  $P(\mu) < 0.9$  remain in the selection. The PID should increase towards one, underlining the tight selection on the muons.

6. Investigation of Central Drift Chamber Signal Characteristics as Potential Input Features for Neural Networks

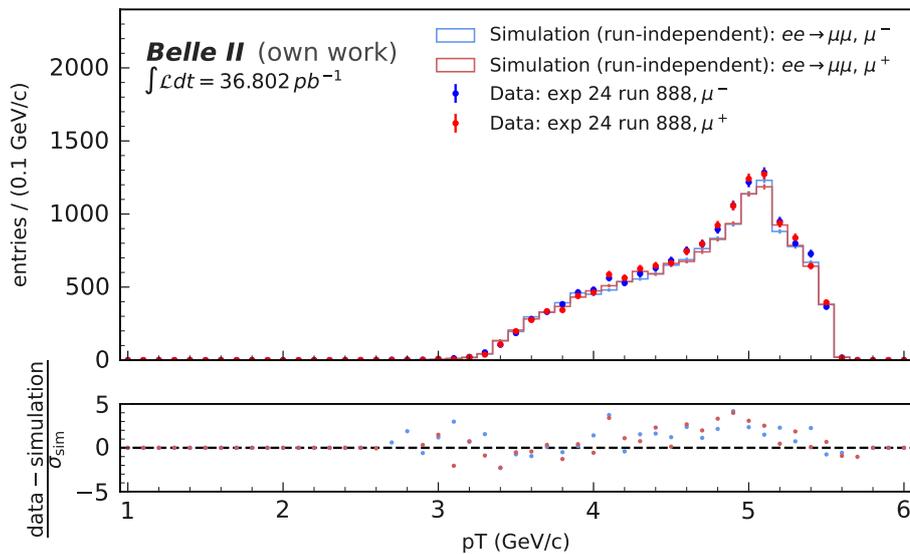


Figure 6.2.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 24, run 888, and run-independent Monte Carlo simulation after selection.

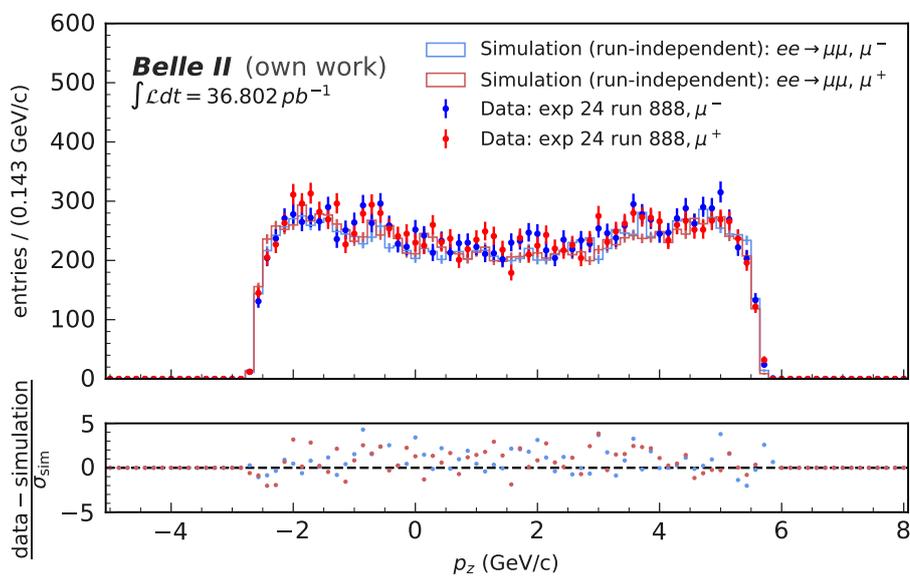


Figure 6.3.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 24, run 888, and run-independent Monte Carlo simulation after selection.

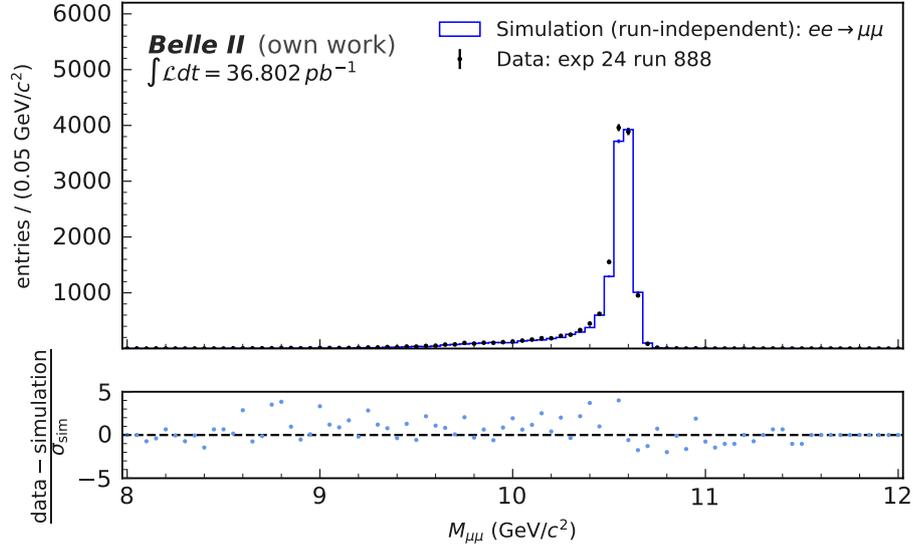


Figure 6.4.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection.

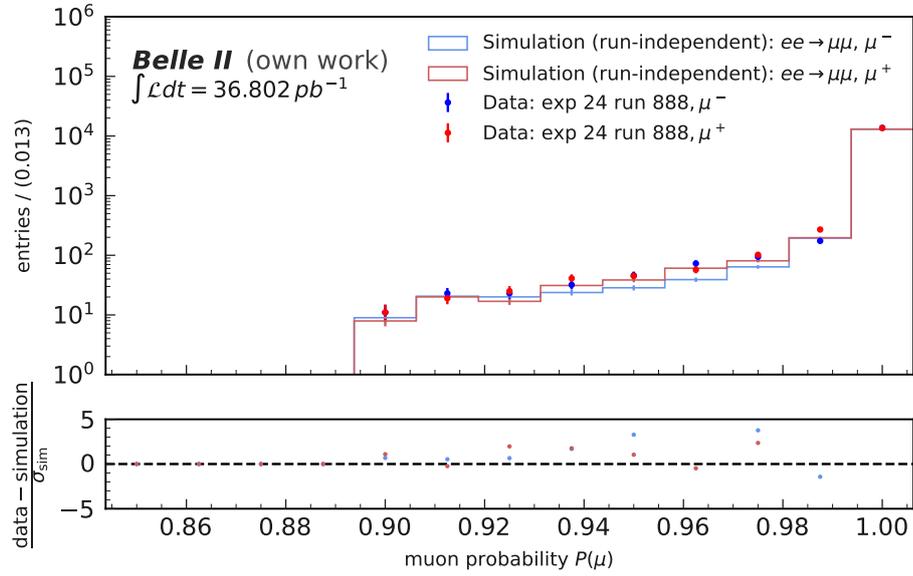


Figure 6.5.: Reconstructed muon Particle Identification  $P(\mu)$  distributions of experiment 24, run 888 and run-independent MC after selection.

The distributions exhibit a general agreement in shape when comparing the simulations and experimental data. However, some notable deviations are observed, which may be attributed to missing corrections or discrepancies between the data and simulation. It is important to recognize that this is an initial comparison, and the primary goal is to evaluate the overall consistency between the ADC and TDC distributions. Considering this objective, the agreement between the simulation and experimental data is considered satisfactory and provides a solid foundation for further analysis. The subsequent section presents and evaluates the distributions of ADC and TDC counts.

## 6.2. Results

The data and simulation are expected to contain a similar number of tracks and hits. In table 6.3 track ratios

$$\eta_{\text{tracks}} = \frac{n_{\text{tracks\_data}}}{n_{\text{tracks\_sim}}}, \quad (6.3)$$

hit ratios  $\eta_{\text{hits}}$  and the respective signal ratios  $\eta_{\text{signal\_hits}}$  and background ratios  $\eta_{\text{bkg\_hits}}$ , are listed for several runs. The expected ratio is  $\eta_{\text{tracks}} = 1$ . Some investigated runs show a deviation of 20% or more, indicating a substantial deviation from the expected ratio. The absence of corrections on the selected events could partially account for this. Additionally, the number of hits shows considerable differences between the simulation and real data. Surprisingly, the presented runs show a negative correlation for the ratio  $\eta_{\text{total\_hits}}$  with the rising experiment number. This result is somewhat counterintuitive, as the expected beam background is rising due to continuously increased instantaneous luminosity over the time period of the Belle II experiment, which implies an opposite behavior.

The background in the CDC is influenced by the machine parameters of the SuperKEKB. Collimators, optics, beam current, and other machine parameters significantly affect the background. The most important backgrounds in the CDC are discussed in subsection 2.2.3. It is possible that the differences in ratios are due to the short runs used for comparison. The injection backgrounds may significantly impact these runs, considerably affecting the expected number of background hits in the CDC. A measurable parameter that offers insight into the injection background of Belle II is the leakage current of the CDC. The corresponding average leakage current per layer  $I_{\text{leak}}$  for the runs are shown in table 6.3. The leakage current increases with higher injection background [42] and can be correlated with the expected background hits in the CDC. A positive relationship was found between the background hit ratio  $\eta_{\text{bkg\_hits}}$  and the leakage current  $I_{\text{leak}}$ , with a Pearson correlation coefficient of  $\rho = 0.96$ , underlining, that the injection background can at least partially explain the observed decrease in background hits.

Table 6.3.: Experimental data to Monte Carlo simulation ratios of hits and tracks after selection.

exp.	run	$I_{\text{leak}}$ in $\mu\text{A}$	$\int \mathcal{L} dt$ in $\text{nb}^{-1}$	$\eta_{\text{total\_hits}}$	$\eta_{\text{signal\_hits}}$	$\eta_{\text{bkg\_hits}}$	$\eta_{\text{tracks}}$
20	672	110.8	47573.42	1.909	0.797	2.226	0.789
20	874	83.72	44006.91	1.835	0.938	2.091	0.926
21	116	101.28	47118.00	1.850	0.838	2.138	0.833
21	128	103.74	37740.47	1.852	0.831	2.144	0.825
22	32	58.55	39656.24	1.087	1.062	1.094	1.030
22	546	73.48	38852.53	1.462	0.974	1.601	0.962
24	888	27.12	36802.19	0.843	1.086	0.774	1.054

### 6.2.1. Evaluating the Discriminatory Potential of Analog-to-Digital Converter Counts

The histogram of the ADC distribution is shown in fig. 6.6. Hits assigned to a reconstructed signal track are represented in blue, while the remaining hits are labeled as background and colored red. Values below an ADC count of two are shifted in the zero bin, and the distribution is shown up to 600 ADC counts. The histogram reveals two distinctive bumps in the background distribution in regions that are anticipated to be dominated by cross-talk. Further information on this background effect is elaborated in subsection 2.2.3. The effect leads to background hits with a characteristic ADC distribution. The first occurs in the low ADC regions below 20 ADC Counts. The second bump is present around 150 ADC counts. In both regions, the background count exceeds the signal count, making this a possible distinguishing feature for the background to signal discrimination. These features are present in simulation and real data. However, comparing the two distributions reveals a large deviation in the number of hits in the real data background compared to the simulation. A large excess is present in the bins lower than 20 ADC Counts. This excess could be found in all investigated data. The cross-talk modeling in the simulation strongly influences the low ADC region as it is one of the dominating contributions. Therefore, the deviations may result from discrepancies in the cross-talk modeling for the low ADC region compared to the data. These discrepancies are not fully understood yet and should be investigated in future research in detail.

Conversely, the shape of the histogram exhibits a similar curve in the remaining ADC region for simulation and data for the background hits. Still, it shows a shortage of hits compared to the simulation, which is in agreement with the overall background hit ratio  $\eta_{\text{bkg\_hits}} = 0.774$ . Again, providing evidence of the deviations in the background modeling for the ADC count.

Delving into the distribution of the signal hits, it is worth reminding that ADC value refers to the deposited energy in the CDC. The expected energy deposition  $dE/dx$  of a particle in the CDC is influenced by particle properties. For muons with

energy below about 100 GeV, ionization is the dominant energy loss mechanism [51]. The average energy loss can be described by the Bethe-Bloch formula, which mainly depends on

$$\beta\gamma = \frac{v/c}{\sqrt{1 - (v/c)^2}}. \quad (6.4)$$

As a result, the energy deposition of an ionization point in the CDC is expected to be Landau distributed for the dimuon events. Consequently, the energy deposition in the signal hits in the CDC is also expected to be roughly Landau distributed. It is important to note that several additional factors can influence the ADC distribution, including the number of ionization points in a drift cell and the detector resolution. Given the anticipated Landau distribution for the ADC signal, a dominating region for the dimuon events in the ADC distribution is expected. The observed signal shows a peak in the range of 30 ADC counts to 100 ADC counts where the signal dominates the total hit distribution. The agreement between simulated and real data events is good for most of the observed range, particularly in the peak region. However, deviations are present in the region of 120 ADC counts to 500 ADC counts, where the simulation shows a shortage of hits. Several factors may contribute to this discrepancy, given that the overall data-to-simulation ratio is good in terms of the overall signal hit ratio and the signal track ratio. These sources of influence include inaccurate modeling of the ADC count distribution in the simulation, as detector efficiency differences in simulation and data. These differences in the signal may also result from the missing corrections.

Nonetheless, it is clear that both signal and background exhibit a range of ADC counts with distinguishing features. Therefore, the ADC count provides a powerful separation potential between data and simulation and should thus be utilized as an input feature for analysis.

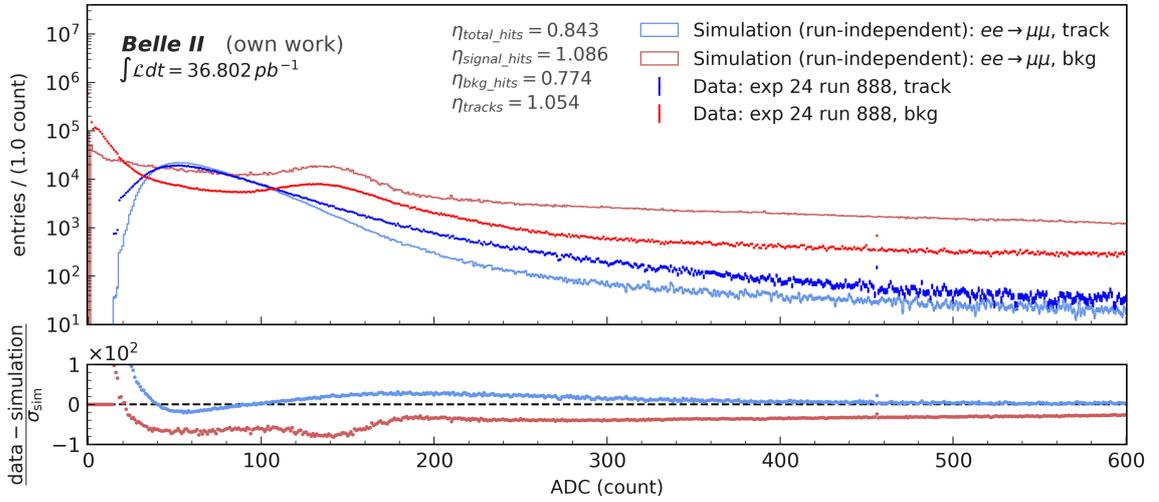


Figure 6.6.: Comparison of Analog-to-Digital Converter count distributions for signal and background hits in experiment 24, run 888 and simulation. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. Distinctive bumps in the background distribution can be observed below 20 ADC counts and around 150 ADC counts, with the signal dominating the total hit distribution in the range of 30 ADC counts to 100 ADC counts. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis.

### 6.2.2. Angular Dependence of the Analog-to-Digital Converter count

After the reconstruction of events, each reconstructed track in the CDC can be related to the associated hits. Therefore hits can be associated with reconstructed track parameters like the polar angle. Figure 6.7 presents the ADC counts of hits of the data in a Two-Dimensional (2D) histogram in dependency to the reconstructed signal tracks angle  $\theta$ , which is the polar angle measured from the beam axis. Figure 6.8 presents the corresponding histogram for the simulated distribution. The distribution of hits in  $\theta$  is expected to be slightly asymmetric due to the asymmetrical arrangement of the CDC around the Interaction Point (IP) and the asymmetrical beam energies.

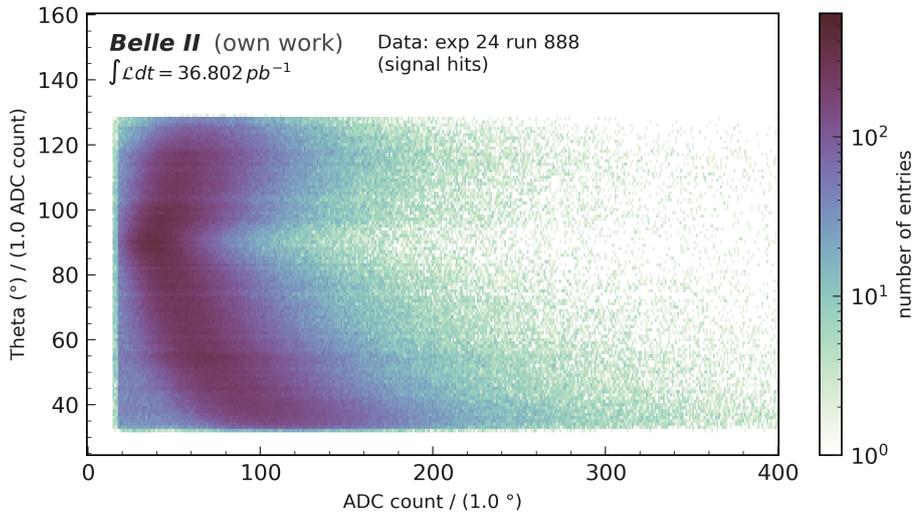


Figure 6.7.: A two-dimensional histogram showing the Analog-to-Digital Converter counts of signal hits associated with a reconstructed track as a function of the polar angle  $\theta$  derived from experimental data. The angle  $\theta$  is measured from the beam axis.

Tracks with a larger  $\theta$  angle are observed to have a correlated ADC distribution. This is expected due to space charge effects for tracks orthogonal to the sense wire [15]. The space charge effect occurs when ionization points of charged particles inside the CDC influence the electric field, affecting the signal produced by the particles. The space charge effect is more pronounced for tracks orthogonal to the sense wire, leading to a correlation between the angle  $\theta$  and the ADC distribution. A detailed illustration is shown in fig. 6.9. The sense wires of the CDC are arranged orthogonal to the  $\theta$  direction.

The ratio between data and simulation is presented in fig. 6.10. The agreement between data and simulation is good in most regions. As shown in table 6.3, the ratio of signal hits  $\eta_{\text{signal\_hits}}$  has a relatively good agreement. Therefore, a high agreement is anticipated. The histograms show that regions with the majority of hits have only minor deviations, while bins with fewer hits exhibit larger deviations.

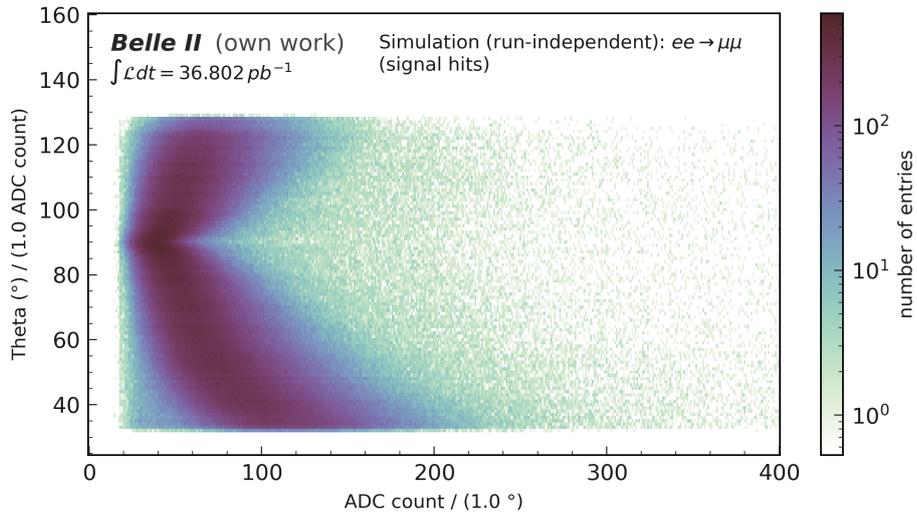


Figure 6.8.: A two-dimensional histogram showing the Analog-to-Digital Converter counts of signal hits associated with a reconstructed track as a function of the polar angle  $\theta$  derived from simulation. The angle  $\theta$  is measured from the beam axis.

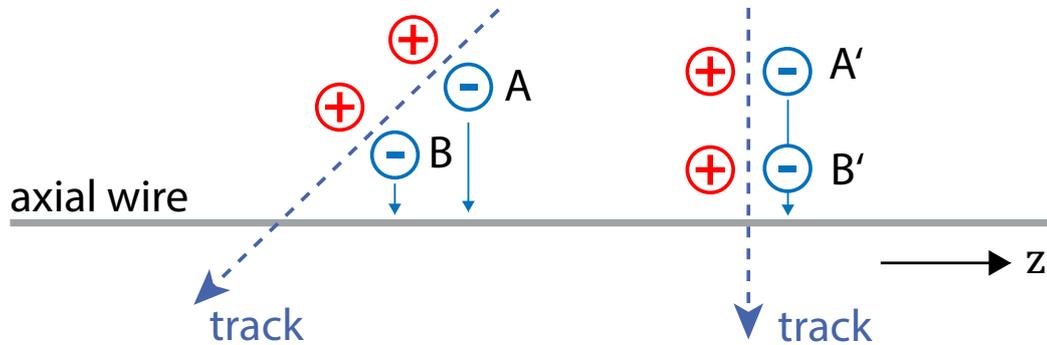


Figure 6.9.: Illustration of the space charge effect in the Central Drift Chamber (CDC), leading to a positive correlation of ADC and increased angle in  $|\theta|$ . Two particle tracks are illustrated, traversing an axial sense wire of the CDC. The left illustration shows a track with a shallow angle to the sense wire. The two ionization points, A and B do not influence their signal in the axial wire. The orthogonal track on the right also leads to two ionization points A' and B'. Point B's electron avalanche lowers the electric field at A', leading to a smaller overall signal. Based on illustrations by Mirei Aoyama.

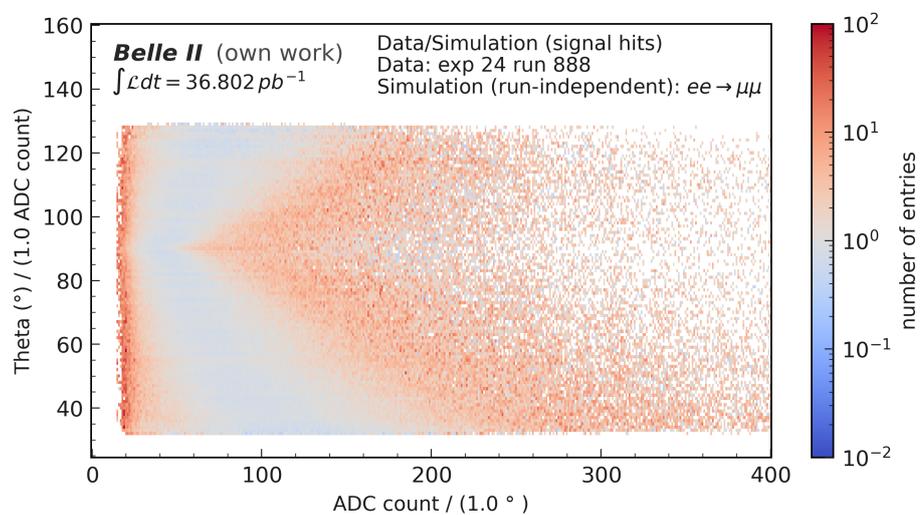


Figure 6.10.: This plot demonstrates the ratio of the experimental (experiment 24, run 888) and simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

### 6.2.3. Evaluating the Discriminatory Potential of Time-to-Digital Converter Counts

The TDC count distribution shown in fig. 6.11 demonstrates overall a good agreement in shape for the background distribution between simulation and data. The number of backgrounds in the data is expected to show an excess in line with the background hit ratio of  $\eta_{\text{bkg\_hits}} = 0.774$ . These additional hits are anticipated to be attributed to the increased injection background. This type of background is expected to be randomly distributed in time. The difference in background hits between data compared to the simulation reveals a nearly uniformly distributed negative pull across the entire TDC range, which is consistent with this hypothesis. However, the background pull shows increased fluctuations around 5000 TDC counts, where the background simulation does not accurately represent the actual data distributions, resulting in more significant deviations.

The signal hits from a single event are expected to be clustered within a small time window, which generally depends on the trigger signal. This implies a correlation between the time window and the trigger signal properties, suggesting that the TDC signal may exhibit a region with increased signal hits. This, in turn, introduces discriminatory potential within the signal and background. The ratio of signal hits  $\eta_{\text{signal\_hits}} = 1.086$  implies an overall good agreement in the TDC count.

However, there is a shortage of CDC hits observed in the range of 4250 TDC counts to 4550 TDC counts. Although the number of signal hits in this range is small compared to background hits, it may still slightly affect signal-to-background discrimination. On the other hand, there is a good match in the TDC window of 4600 TDC counts to 4880 TDC counts between simulation and data, where the ratio of signal to background hits increases, enabling discriminating signal to background. In the right area of this peak, between 4880 TDC counts to 4950 TDC counts, there are again noticeable deviations present due to a much more smeared-out signal peak in the data as present in the simulation, suggesting disparities in the simulation. These differences in the signal may be the result of the missing corrections.

It is important to note that the real data's TDC distribution exhibits a consistent oscillation in the histogram, evident in both the signal and background. However, the cause of this phenomenon needs to be fully comprehended and requires further investigation in future research.

Overall, the TDC count shows discriminatory power between signal and background for data as well as for simulation. Hence, it is a suitable input feature and should be used.

## 6. Investigation of Central Drift Chamber Signal Characteristics as Potential Input Features for Neural Networks

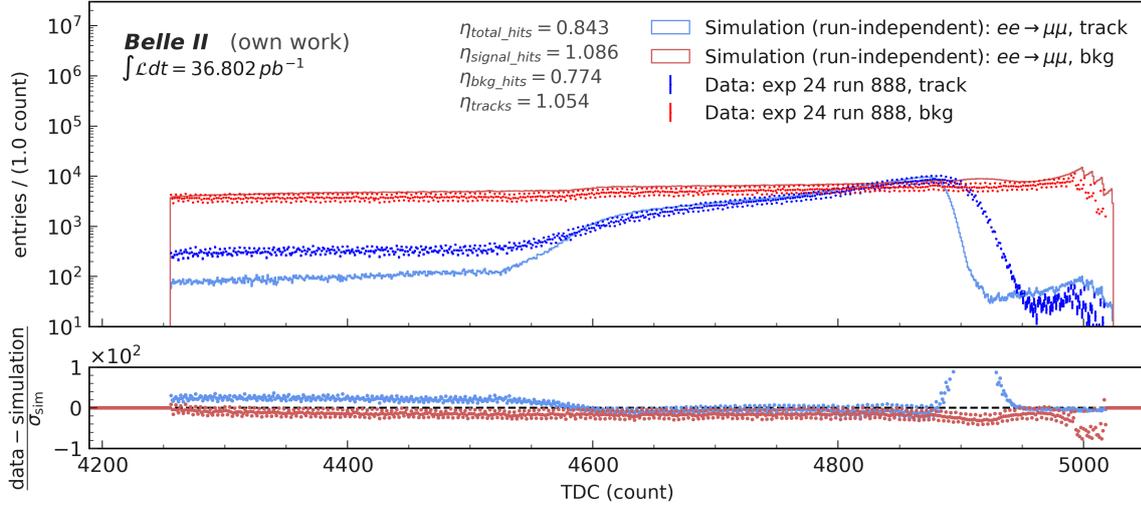


Figure 6.11.: Comparison of Time-to-Digital Converter (TDC) distributions for signal and background hits in experiment 24, run 888 and simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution, with noticeable deviations between simulation and data. Oscillations observed in the data’s TDC distribution require further investigation.

### 6.2.4. Orthogonality of Analog-to-Digital Converter and Time-to-Digital Converter Distributions

To understand how the ADC and TDC count distributions complement each other in separating signal hits from background hits, their orthogonality is examined. Figure 6.12, fig. 6.13, and Figure 6.14 show a 2D histogram of TDC and ADC values of all hits for data, simulation, and the ratio between them. It shows that for the signal region with 30 ADC count to 150 ADC count and 4800 TDC count to 4950 TDC count, the counts are predominantly uncorrelated, as is the major background region. This observation agrees with the analysis presented in [34], where the combination of both features significantly improves the edge classification for the CDC.

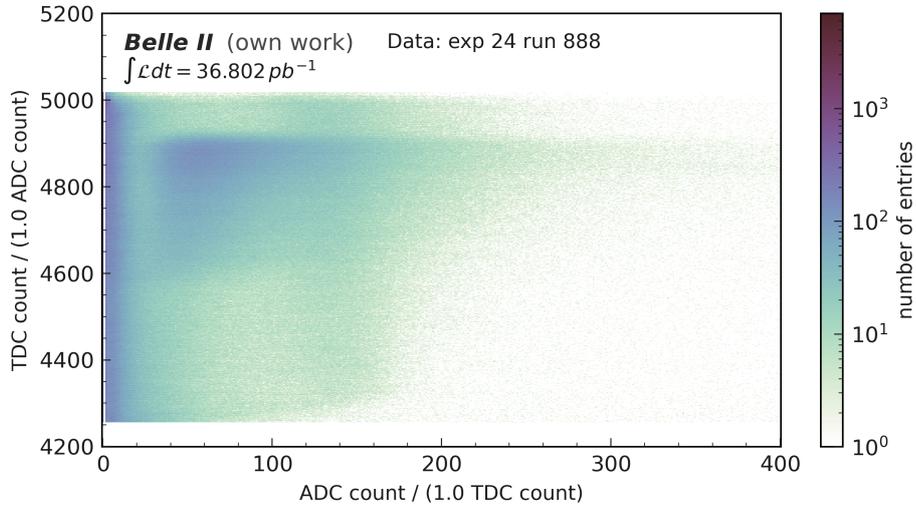


Figure 6.12.: This two-dimensional histogram presents the Analog-to-Digital Converter and Time-to-Digital Converter counts of Central Drift Chamber hits from experimental data (experiment 24, run 888). The signal region, with Analog-to-Digital Converter counts ranging from 30 to 150 and Time-to-Digital Converter counts from 4800 to 4950, predominantly exhibits uncorrelated counts.

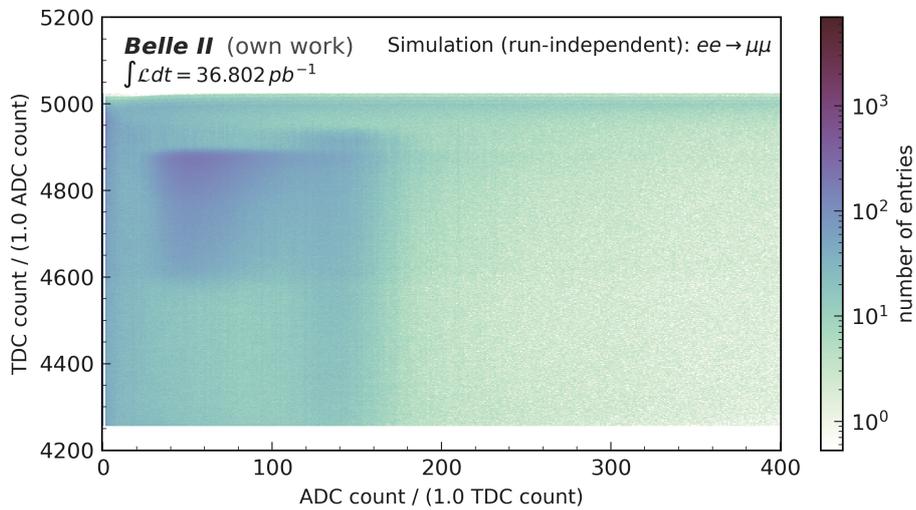


Figure 6.13.: This two-dimensional histogram presents the Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts of Central Drift Chamber (CDC) hits from simulated data. Mirroring the experimental data, the signal region, with ADC counts ranging from 30 to 150 and TDC counts from 4800 to 4950, predominantly exhibits uncorrelated counts.

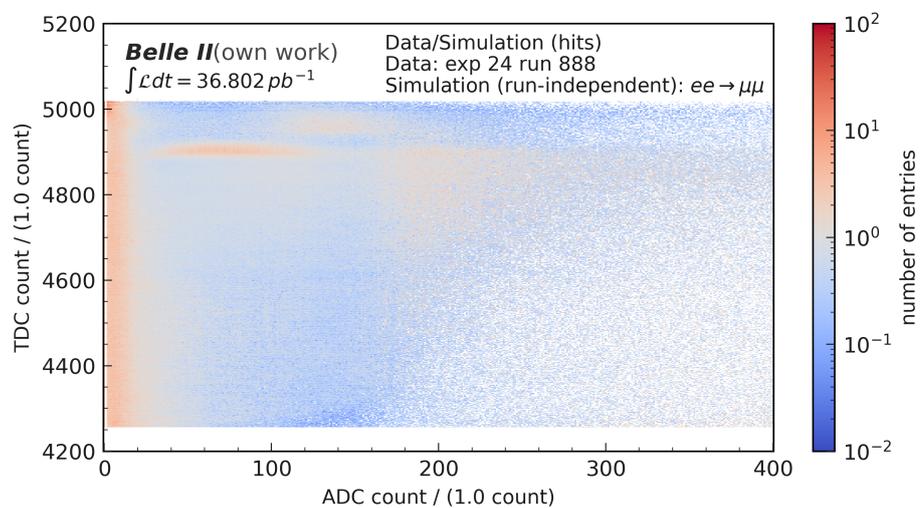


Figure 6.14.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter and Time-to-Digital Converter counts for Central Drift Chamber hits, providing a method for evaluating the relative agreement between the two distributions.

### 6.2.5. Spatial Distribution of Central Drift Chamber Hits

In addition to the ADC and TDC counts, the spatial information of the CDC hits is another input feature for the Graph Neural Network (GNN)-pipeline. The two muons originate at the IP and travel through the CDC, with their trajectories in the  $x - y$  plane concentrated around the central region of the Belle II detector. As the muons can be boosted in  $z$ -direction, they can leave the CDC in this direction, resulting in a lower hit density in the outer regions of the detector. Furthermore, the innermost superlayer of the CDC features more closely spaced sense wires, leading to a denser distribution of expected hits in that region. fig. 6.15 and fig. 6.16 show the distribution of the hits in the  $x$  and  $y$  direction, in good agreement with the expected distribution. Again, it is observed that there is a shortage of hits in the real data, which is approximately consistent over the entire range of  $x$  and  $y$  values but is more strongly present close to the IP. Apart from this discrepancy, the alignment of the  $x$  and  $y$  distributions are found to be consistent. The background hits show a much higher pull compared to the signal simulation, which is consistent with the discrepancies found in the ADC and TDC distribution.

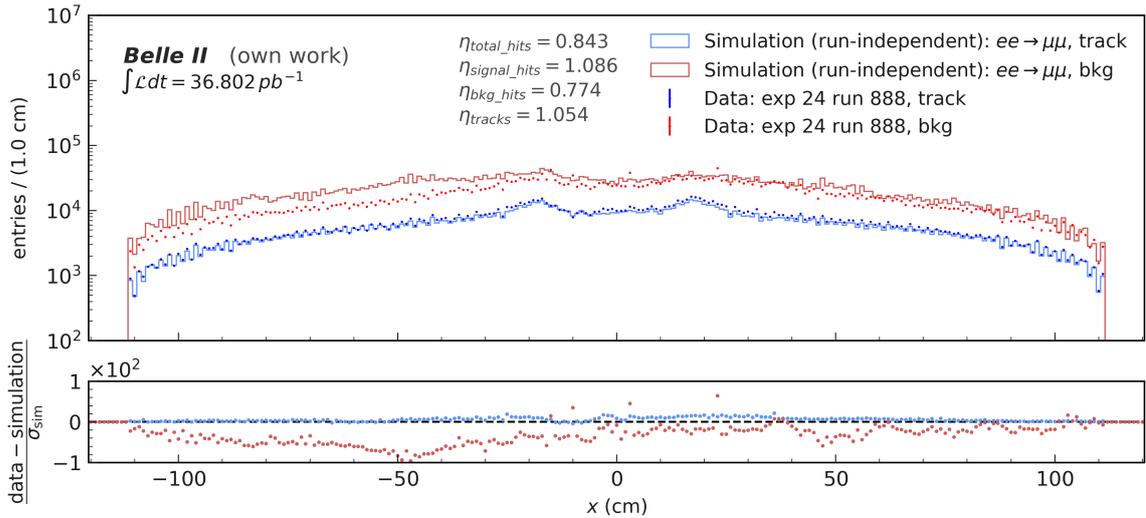


Figure 6.15.: This plot shows the spatial distribution of Central Drift Chamber hits in the  $x$  direction. The distribution of the experimental data (experiment 24, run 888) agrees well with the simulation, although a shortage of hits is observed in experimental data.

## 6. Investigation of Central Drift Chamber Signal Characteristics as Potential Input Features for Neural Networks

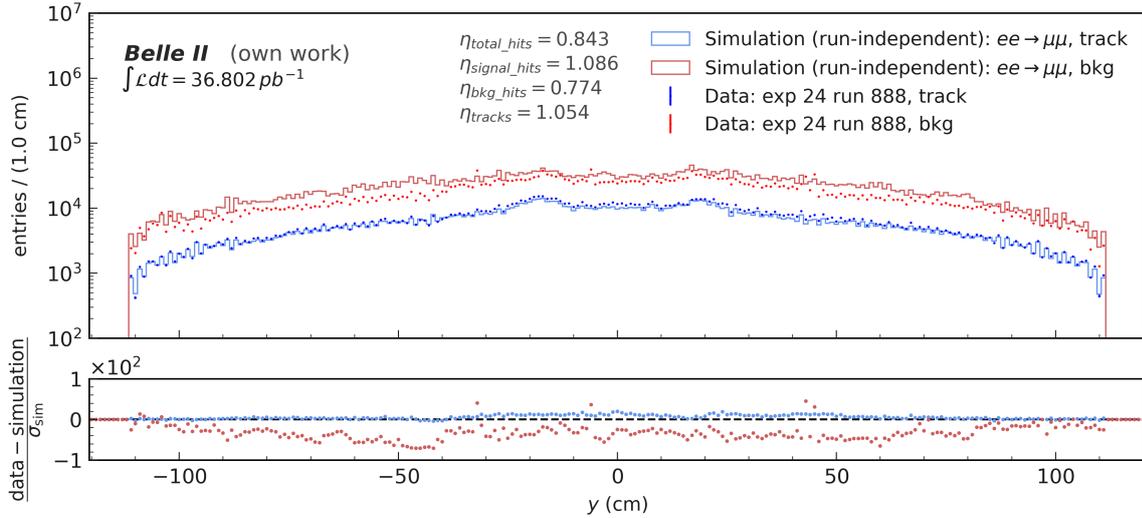


Figure 6.16.: This plot shows the spatial distribution of Central Drift Chamber hits in the  $y$  direction. The distribution of the experimental data (experiment 24, run 888) agrees well with the simulation, although a shortage of hits is observed in experimental data.

### 6.2.6. Implementing Analog-to-Digital Converter and Time-to-Digital Converter Information in Event Analysis

The current analysis of the ADC and TDC count distributions are the first data to MC comparison with a tight selection on dimuon events. It is shown that both distributions are discriminatory and have the potential to be used as input features for further analyses in the Belle II detector. The comparison of the signal track hits shows a close agreement between the data and simulation, with a ratio close to one. The discrepancy between the data and simulation mainly comes from the non-signal hits. Furthermore, run-dependent MC studies were investigated, resulting in a high agreement for the background, as can be viewed in appendix A.2. Following this analysis, it is anticipated that the ADC information for each trigger wire will be available by fall 2023 after the Long Shutdown 1 (LS1). However, the resolution will be significantly lower. The ADC information is expected to be utilized using three flags. In order to effectively use the discrimination potential of the ADC count the analysis suggests using a range of 0 ADC counts to 30 ADC counts as a first flag, the 30 ADC counts to 150 ADC counts as a second flag and 150 ADC counts to 600 ADC counts as a third flag. This is a first suggestion, and further investigations into the performance of the edge classification may make it necessary to change these ranges.

The definition of the TDC changes for the L1 Trigger level since the event time  $T_{\text{evt}}$  is unknown. For this reason, absolute values of the TDC count hold no significance. The expected resolution is 2 ns. Therefore, it may be useful to calculate

TDC differences between two hits and use this information as an edge attribute instead of utilizing unprocessed TDC information as an input feature. Further results on the edge classification can be found in chapter 8. Moving forward, further understanding the background modeling will be necessary to improve the simulation's accuracy and better understand the background. This incorporates taking into account the granularity and specificity of the ADC and TDC count as a detector output, which might require more detailed modeling in the simulation to capture subtle variations and correlations that could influence their distribution and also their discriminatory power, which is crucial to understand for the analysis. Subtle differences in the distributions may influence the performance of any machine learning algorithm which is trained on the simulated samples. Therefore differences in the hit distribution are crucial to be understood in future applications of TDC and ADC. In addition, an analysis with longer and more stable runs could improve the background hit comparison in the CDC.



## 7. Graph-Building

In High Energy Physics (HEP), detectors provide their data in a heterogenous format, with each subdetector generating different information about an event using intricate sensors. Accurate representation of each measurement is essential for analyzing experimental results, particularly for tracking detectors like the Central Drift Chamber (CDC) at the Belle II experiment.

The CDC has a complex structure with an irregular format. The 56 layers with sense wires are arranged concentrically around the Interaction Point (IP) and have increasing numbers of wires per layer. Each of the wires can provide information for particle tracking. The core CDC data thus consists of a variable number of Two-Dimensional (2D) position measurements, along with additional information for each measurement, such as timing information and deposited energy information.

Graphs can be an effective way to represent this data, as they capture relationships between different entities. In the context of the CDC, the entities are the measurements, and the spatial structure can be represented as the relations. Nodes are therefore used to capture individual measurements, and edges can connect nodes based on spatial proximity or other relationships. Additional information, such as timing and deposited energy, can be added as node features or edge attributes. The graphs in this analysis contain  $x$  and  $y$  position of the wires, timing information Time-to-Digital Converter (TDC), and energy information Analog-to-Digital Converter (ADC) as node features. The edge attributes of the graphs include the distance in the  $x - y$ -plane of measurements in the detector  $\Delta\rho$  and the angular distance  $\Delta\phi$ .

The graph-based representation is well-suited for embedding details on a single wire level of the CDC, providing a flexible and intuitive way to analyze the data.

Graph-building is the process of transforming raw data into the structured form of a graph. It is particularly important to include relevant physical relationships, while it is desirable to exclude unphysical ones. As the number of hits  $n_{hits}$  in the CDC is expected to increase considerably due to increased beam background, the potential number of edges in a fully connected graph will grow at  $\mathcal{O}(n_{hits}^2)$ . This results in a large number of connections, many of which provide little physical knowledge for the tracking of the signal particle. Therefore, the additional background connections may negatively influence the analysis with the *GNN-pipeline* due to increased noise.

The computing load for the *GNN-pipeline* is expected to depend on the size of the input graphs. Particularly in the online application of the *GNN-pipeline*, tight

latency constraints in the sub-microsecond timescale must be met. On the specialized hardware, this requires optimizing the parallel data throughput where constraints on the size of the graphs may apply. Consequently, nodes should be connected purposefully to maintain a balance between incorporating valuable physical knowledge and adhering to hardware limitations for efficient real-time processing.

This chapter investigates various graph-building methods and analyzes their performance in the context of the Belle II CDC. The goal is to develop a systematic approach for graph-building that connects nodes selectively, using domain knowledge and physical expectations, to improve tracking accuracy with the Belle II detector. To accomplish this, metrics are introduced to assess the quality of graph-building, which will help in selecting appropriate graph-building models for specific tasks within the Belle II experiment.

### 7.1. Signal and Background

In the scope of Belle II, an event refers to a specific instance when particles interact, decay, or are produced in the detector, which was captured for analysis. The stored data for an event consists of accumulated information over a defined time period. As a result, a hit in a CDC wire during this time period may contain multiple incidents. Typical Belle II simulated events consist of a mixture of signal and background hits. A hit in one CDC wire is defined as a signal hit if any simulated signal particle deposited energy in that wire. Monte Carlo (MC) identities refer to such signal particles in this simulation. If a node of the graph is matched to a signal particle, this does refer to deposited energy in the corresponding CDC wire. One node can also be matched to multiple MC identities if multiple signal particles share a hit. The remaining nodes are considered background. fig. 7.1 illustrates a fraction of the CDC, showing hits originating from signal particle tracks and background. It also displays true and false edges resulting from an arbitrary graph-building method. Edges between two nodes matching the same MC particle are regarded as true edges. All other edges are defined as false. This characterization allows for a clear, model-independent way to distinguish the nodes and the edges.

In an intermediate step of the *GNN-pipeline*, the Interaction Network (IN) is utilizing the graphs from the graph-building to classify edges. These classified edges are eventually used to clean up events in the CDC from background hits. True edges provide information about the relation of signal hits in the CDC. Therefore they introduce essential knowledge to the *GNN-pipeline* for particle tracking. Thus, selecting the true edges incorporated in the graph-building step is essential to alleviate the classification task. In theory, an event with  $n$  true nodes can compose  $n \cdot (n - 1)$  true edges when building a fully connected graph. This number scales in  $\mathcal{O}(n^2)$  for a large number of signal hits in the CDC. Even though these true edges may

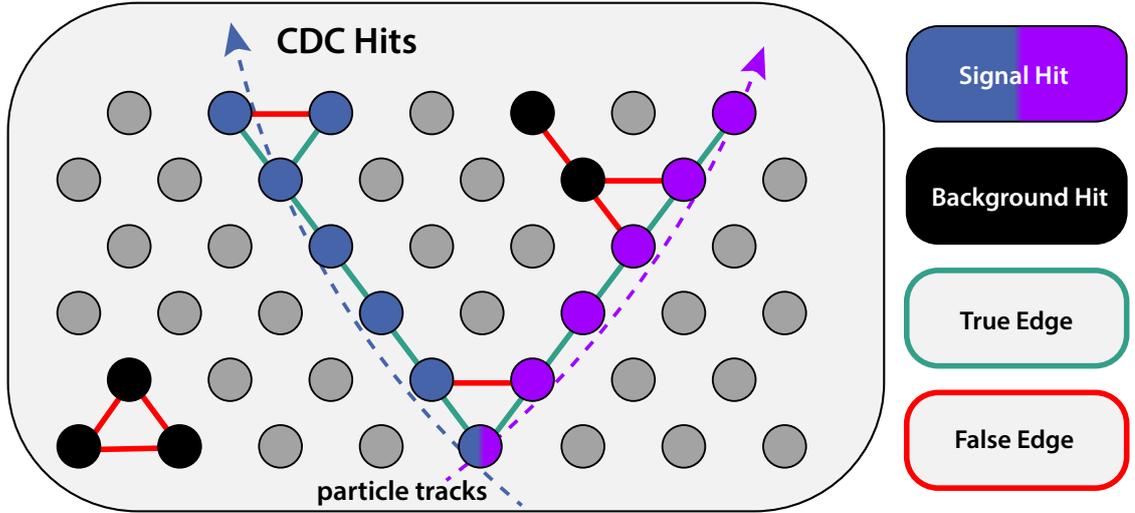


Figure 7.1.: Illustration of a detail from the Central Drift Chamber, showcasing particle tracks, signal and background hits, and their associated true and false edges. Signal hits are those matched to a signal particle, while background hits are unassociated with any signal particle. The lowest signal node is associated with two signal particles, emphasizing the possibility of multiple signal particles sharing a hit. True edges (marked in green) connect nodes to match the same Monte Carlo particles, while false edges (marked in red) do not.

provide relational information between the signal hits, not all of them may be similarly important for the background clean-up.

## 7.2. Wire and Layer Distance

Two parameters are considered to define possible spatial connections between two nodes  $v_i$ , and  $v_j$ ,

$$\text{the layer distance} \quad \Delta l_{ij} \quad (7.1)$$

$$\text{and the wire distance} \quad \Delta w_{ij}. \quad (7.2)$$

These two parameters are advantageous against spatial distances measured in units such as centimeters. They are independent of the specific layout of the wires in the CDC and provide a more generalized approach to describing connections between hits. The numbering scheme is illustrated in fig. 7.2. The wire distance,  $\Delta w_{ij}$ , is calculated considering the periodic boundary conditions of the CDC.

$$\Delta w_{ij} = \text{sign}(w_j - w_i) \cdot \min_{k \in \{-1, 0, 1\}} |w_i - w_j + kN|, \quad (7.3)$$

where  $w_i$  and  $w_j$  are the wire indices of hits  $v_i$  and  $v_j$ , respectively, and  $N$  is the total number of wires in a single layer.  $\text{sign}(x)$  is the sign function, which returns the sign

of the input value  $x$ :

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases} \quad (7.4)$$

The wires of the CDC are shifted in every other layer to each other. Therefore one single node has two neighboring nodes in a neighboring layer. The wire distance of each nearest neighbor with a layer distance of  $\Delta l_{ij} = 1$  is defined with a wire distance of

$$\Delta w_{ij} = 0. \quad (7.5)$$

Similarly, the connections to the next-to-next neighbors in the same layer distance are defined with a wire distance of

$$\Delta w_{ij} = \pm 1. \quad (7.6)$$

This does not fully match the Belle II Analysis Software Framework (basf2) numbering between two nodes  $v_i$  and  $v_j$  with a layer distance of  $\Delta l_{ij} = 1$ . The basf2 defines the wire distances to the neighboring nodes as

$$\begin{aligned} \Delta w_{ij} &= 0, -1, \text{ for } i \text{ even and } sl = 0 \\ \Delta w_{ij} &= 0, +1, \text{ for } i \text{ odd and } sl = 0 \\ \Delta w_{ij} &= 0, -1, \text{ for } i \text{ odd and } sl > 0 \\ \Delta w_{ij} &= 0, +1, \text{ for } i \text{ even and } sl > 0. \end{aligned}$$

With the number of the superlayer in the range of  $sl = [0, 1, \dots, 8]$ . The numbering scheme is redefined for ease of the metric and model definitions, which will be defined symmetrically. Furthermore, this thesis will use a numbering scheme from SL1 to SL9 referring to the nine superlayer of the CDC.

For connections in the same layer and connections with a layer distance of  $\Delta l = 2$ , the neighbors are defined in alignment with the basf2 wire numbering scheme as shown in the following equations

$$\Delta l_{ij} = 0 \text{ and } \Delta w_{ij} = \pm 1 \quad (7.7)$$

$$\Delta l_{ij} = 2 \text{ and } \Delta w_{ij} = 0, \pm 1. \quad (7.8)$$

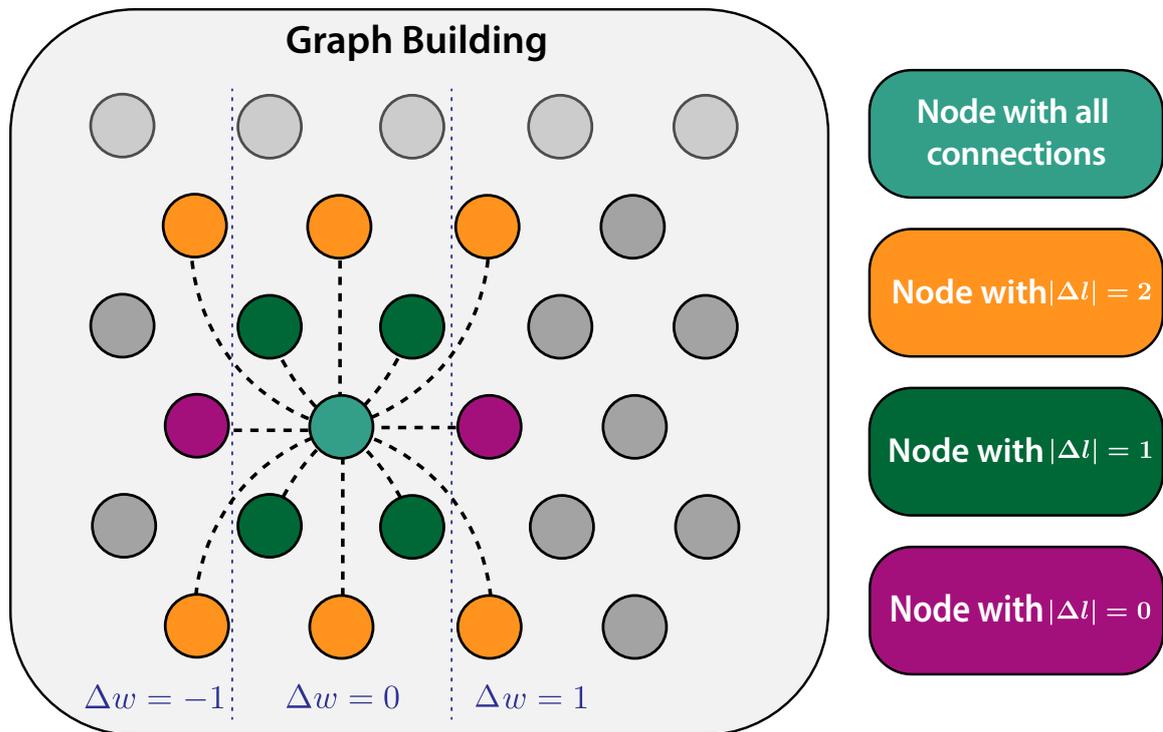


Figure 7.2.: Schematic of layer and wire distance in the CDC for graph-building. The numbering scheme of the Belle II Analysis Software Framework is adjusted to ensure symmetrical wire distances for neighboring layers.

### 7.3. Metrics for Evaluating Graph-Building

In order to analyze physical events represented in a graph, it is crucial to include

- (i) all relevant nodes that provide relevant position measurements of a traversing signal particle and complementary detector information, and
- (ii) all relevant edges providing relational knowledge between these measurements.

Consequently, it is essential to define edges of physical significance. Section 7.1, discusses how to classify hits and edges of a graph for the CDC using simulated events. Following this, two methods are introduced to delineate a set of physically interesting edges to analyze the performance and quality of a graph-building algorithm. Method one, presented in subsection 7.3.1, is introduced to describe a set of potentially relevant edges by using geometrical information. Method two, presented in subsection 7.3.2, uses an algorithm to estimate a particle's track in the CDC, to construct an underlying graph to create a minimal subset of physically relevant edges. The overhead metric introduced in subsection 7.3.3 discusses the trade-off between the two subsets of physically interesting connections.

#### 7.3.1. Complete Set of True Edges

To ensure the efficiency of the *GNN-pipeline*, it is crucial to carefully select the connections in the graphs and keep their number to a minimum. This requires identifying the true edges in the CDC that hold significance for the *GNN-pipeline*. The complete set of true edges is determined by utilizing spatial information in the CDC to isolate these substantial edges from all possible true edges. Hits of a single particle traversing the CDC are anticipated to form along the particle's path, typically within the close proximity of one to two wires. It is important to note that these hits are generally far apart, as the particle's hits are spread out along its entire path through the detector. Connecting these distant hits could introduce more noise to the graph with little additional information for the track-finding process. Therefore, the maximum layer distance for a true edge in the CDC is restricted to be

$$|\Delta l_{ij}| \leq 2, \quad (7.9)$$

for the complete set of true edges. For particles with a low transversal momenta  $p_T$ , this configuration can lead to a high number of true edges. As these shallow tracks traverse the CDC with a low angle or can even curl in the CDC. This means they have a comparatively high number of true nodes within the layer distance of  $\Delta l = 0, 1, 2$ . Thus they have a large number of true edges within this layer distance. To adjust the number of true edges for these shallow tracks in the complete set, the maximum wire distance is limited to

$$|\Delta w_{ij}| \leq 6. \quad (7.10)$$

By implementing two simple geometrical constraints,  $|\Delta l_{ij}| \leq 2$  and  $|\Delta w_{ij}| \leq 6$  respectively, the complete set of true edges is defined. It is now possible to specify

the fraction of true edges resulting from a graph-building method that is included in the complete set of true edges. The number of those included true edges is formally defined as

$$n_{\text{incl. true edges}} = |\{x | x \in \text{complete set of true edges} \wedge x \in \text{built graph edges}\}|. \quad (7.11)$$

With this, two metrics for the quality measurement of the graph-building are defined. The *true edge pur.* is given by

$$\text{true edge pur.} = \frac{n_{\text{incl. true edges}}}{n_{\text{total edges}}}, \quad (7.12)$$

and the *true edge eff.* is given by

$$\text{true edge eff.} = \frac{n_{\text{incl. true edges}}}{n_{\text{true edges}}}. \quad (7.13)$$

The interpretation of the *true edge eff.* and the *true edge pur.* is not straightforward and depends on the requirements of the classification task. High *true edge pur.* indicates that most part of the edges in the input graph is true edges. This is desirable because it means that the graph is mainly composed of valuable connections for the *GNN-pipeline*. This may enhance the classification with the IN and provide mainly useful relational information for tracking. The amount of noise is small in such a graph.

High *true edge eff.* refers to the fraction of true edges included in the graph compared to the complete set of true edges. Generally, a high *true edge eff.* is desired. Nonetheless, including as many true edges as possible may not always be the optimal solution case, as this also implies a larger graph. Some connections may also be redundant and may not include additional information.

This case is shown in fig. 7.3. The illustrations show a graph highlighting two redundant connections between the outer two true nodes, with and without an intermediate true node. The green edges depict connections with a layer distance of  $\Delta l = 1$ , and the purple edge demonstrates a connection with a layer distance of  $\Delta l = 2$ . Both connections are defined as true edges as both represent connections between two true nodes. Using the IN for classification of the edges, information is only passed from the lower node to the upper node if connections with  $\Delta l = 2$  are included in the graph-building. It is not a priori clear if both connections are beneficial for the edge classification task. Therefore, it is essential to strike a balance between *true edge eff.* and the complexity of the graph to optimize the performance of the *GNN-pipeline*. To further understand the impact of additional true edges in graph-building, a simple graph with a minimum amount of connections is defined in the next section to benchmark the overhead of different graph-building methods.

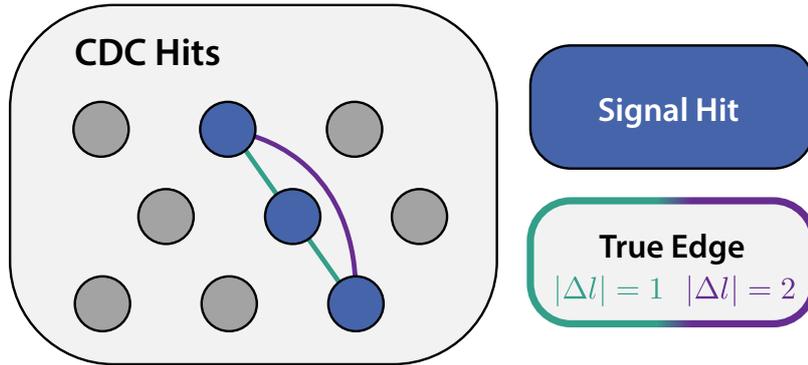


Figure 7.3.: Three signal hits of a signal particle. The outer nodes can be connected using an intermediate connection with the middle hit, resulting in two true edges (colored green). Additionally, the outer hits can be connected using a single true edge (colored purple), skipping the intermediate hit.

### 7.3.2. The True Graph

A straightforward way to construct a true graph is to connect all true nodes in the order of their hit in the CDC from the signal particle. However, the hit order information was not available in the simulation data used in this thesis. Alternatively, the order of hits could be calculated by determining the helix of the signal particle in CDC. This would require storing the simulated true  $z$ -information for each CDC hit. This information was also missing in this study as this information is not available in the used simulation samples. Thus, it is not possible to calculate the particle's exact true path for the simulation samples used. Therefore, a method to estimate the true graph is defined. Three boundary conditions are used for the true graph calculation.

- (i) Only signal nodes with the same MC identity are connected, and
- (ii) each true node may have a maximum of one incoming and one outgoing edge, and
- (iii) not all true nodes must be included in the graph.

The last condition is introduced to account for scenarios where a particle causes multiple hits in close proximity, making it difficult to determine the correct order of hits. A reasonable estimate of the true graph may already be obtained with a subset of the true signal nodes. fig. 7.4 gives an illustration of this idea. The graphic shows a particle traversing the CDC where hits are triggered in close proximity, making it challenging to ascertain the order of hits in a single layer.

Additionally, curling tracks in the CDC can result in multiple hits in single layers, as depicted in fig. 7.5. Connections between hits resulting from different traversals of a layer should be omitted. Only one path is built per superlayer. For curling tracks, multiple true paths exist per superlayer, and only one of them is intended to be included in the true graph estimation. Consequently, the other paths are missing,

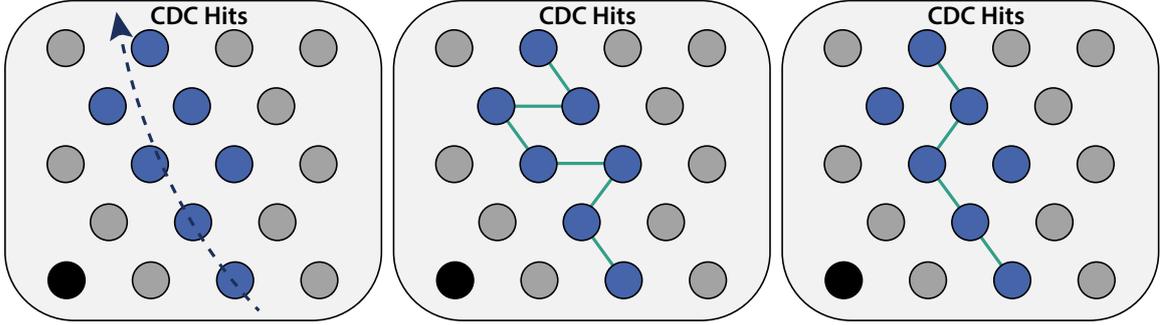


Figure 7.4.: An illustration of the true graph estimation. Blue dots represent signal hits in the Central Drift Chamber, and green lines represent the true graph edges. The middle panel shows a scenario where all true nodes are included, even when they are in close proximity, resulting in a more complex graph. In contrast, the right panel depicts a more streamlined representation, where not all nodes are connected despite being true hits. This approach may provide a better illustration of the underlying trajectory pattern of the signal particle.

and some true graph edges are missing in the estimation of such events. A potential solution involves using information about the correct path of a particle, which is intended for future performance measurements.

The working principle of the used algorithm is shown in figure 7.6. In the first step of the algorithm, all nodes with a hit from one signal MC particle are selected. The next step calculates a distance matrix for each superlayer in the CDC. With this, a path graph is built, starting with a node from the innermost layer with hits. A path graph is a graph where the nodes are arranged in a linear sequence, and each node is connected to its next node by a single edge. The starting node is selected randomly if there is more than one hit in the innermost layer. Afterward, the path is created by recursively connecting the node to the next neighbor.

The distance matrix is calculated using the wire and layer distance. Therefore, the distance between two nodes  $i$  and  $j$  is defined as  $dr_{ij}^2 = \Delta w_{ij}^2 + \Delta l_{ij}^2$ , and for same layer connections  $\Delta l_{ij} = 0$ , there is a bias added to the distance  $\Delta l_{ij} \rightarrow \Delta l'_{ij} = \Delta l_{ij} + 1$  to favor connections between layers. This is implemented as most simulated signal particle tracks used for this analysis point outwards of the CDC.

Again it is possible to define the fraction of true edges resulting from the graph-building that is included in the true graph estimate.

$$n_{\text{incl. true graph edges}} = |\{x | x \in \text{true graph edges} \wedge x \in \text{built graph edges}\}|. \quad (7.14)$$

With this, the true graph purity can be defined as

$$\text{true graph pur.} = \frac{n_{\text{incl. true graph edges}}}{n_{\text{total edges}}}, \quad (7.15)$$

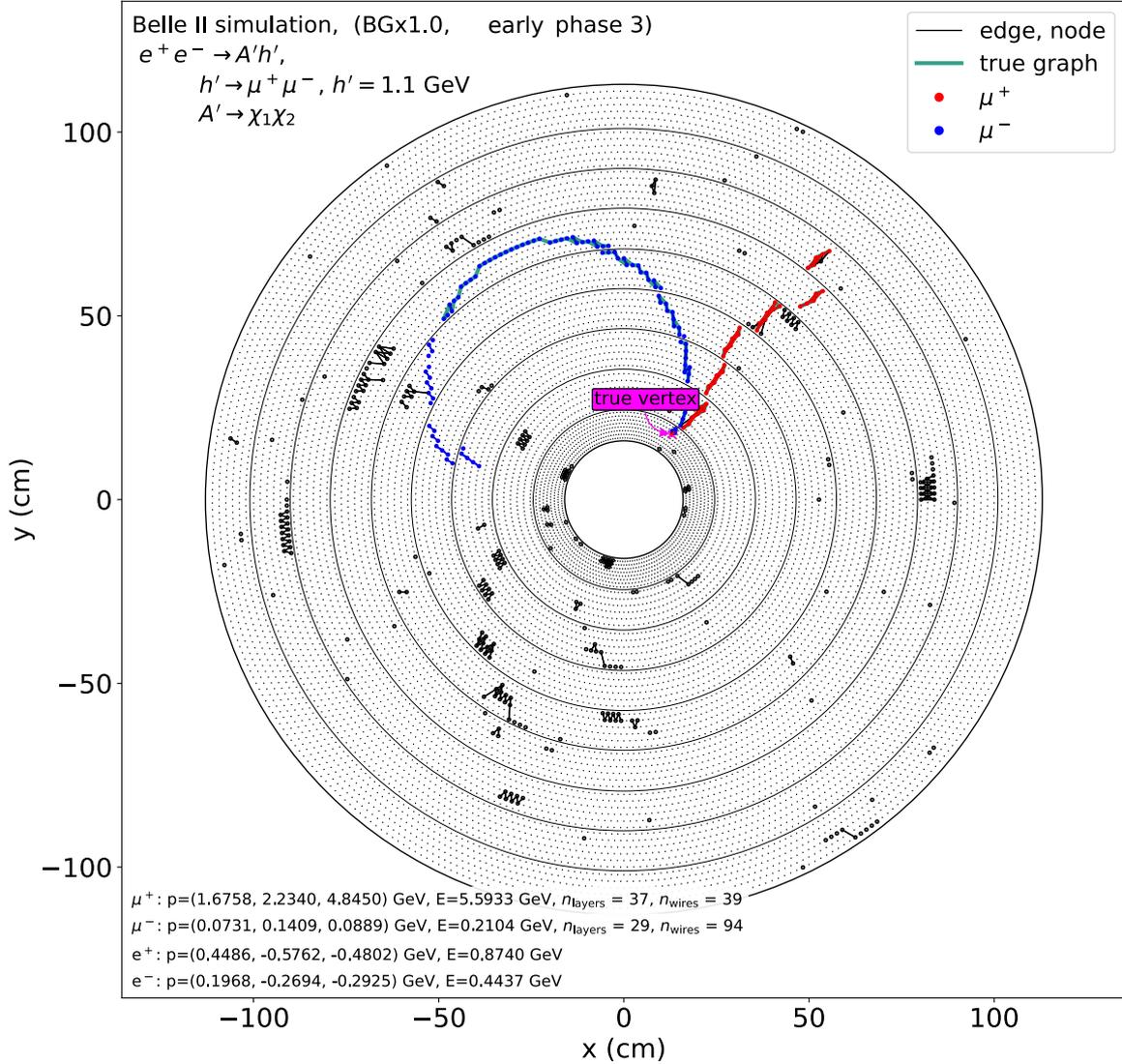


Figure 7.5.: Event display after graph-building. The hits of the two signal muons are colored red and blue. Background hits are colored black. The graph-building incorporated true and false edges, which are colored accordingly. The true graph is colored green. Only one true graph is created per superlayer and particle, resulting in an incomplete true graph for the curling track.

and the true graph efficiency is defined as

$$\text{true graph eff.} = \frac{n_{\text{incl. true graph edges}}}{n_{\text{true graph edges}}}. \quad (7.16)$$

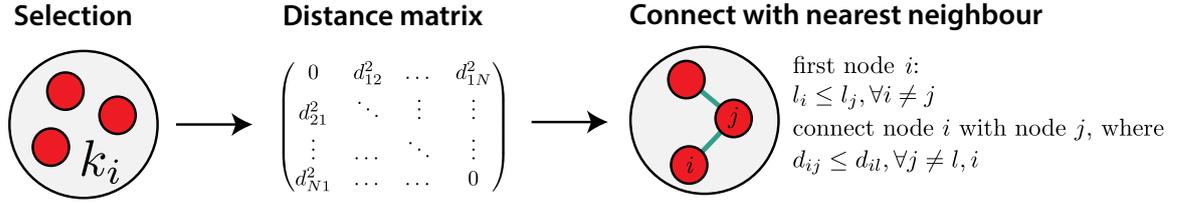


Figure 7.6.: The true graph calculation. First, the set of nodes  $\{v_i\}$  matched to one signal Monte Carlo identity are selected per superlayer. A distance matrix between the nodes is calculated, and finally, a path graph is built by connecting nodes with their nearest neighbor. The first node of the path is selected as the innermost node, with the lowest layer number  $l$ .

High *true graph eff.* is generally desired in graph-building, as it indicates effective capturing of true graph edges, which provide crucial relational information between signal nodes for subsequent steps in the *GNN-pipeline*. However, alternative true graph paths may exist and are not accounted for by the true graph estimation and the *true graph eff.* calculation, implying that other true edges may offer valuable, albeit redundant, information.

Moreover, in some cases, the *GNN-pipeline* might benefit from additional true edges, as they can provide supplementary information to the graph. Consequently, achieving the highest *true graph pur.* may not be the optimal solution in the graph-building step. On the other hand, a high *true graph pur.* demonstrates the ability to minimize false edges and redundant or additional true edges in the graph, ensuring that the information in the graph is concisely reduced to the most important edges. This contributes to better background noise reduction. It is a matter of background clean-up with the IN to find the right balance between the additional true edges and the true graph edges.

Consequently, achieving perfect *true graph eff.* and *true graph pur.* should not be the sole criterion for determining the performance of a graph-building method. In practice, a method with slightly lower *true graph eff.* could still perform well if it captures the true nodes of a particle's paths and forms sufficient true edges providing the necessary relational information for the background clean-up in the *GNN-pipeline*. In the following subsection, the overhead of the true graph edges is defined to measure the additional true edges from a graph-building method.

### 7.3.3. Overcount Metric

Due to the computational constraints, it is advantageous if the graph contains fewer edges for the input in the *GNN-pipeline*. While the true graph estimates the minimum number of edges that are needed to connect the true hits of a signal particle, additional

connections may be useful for the classification task. The *overcount* metric

$$\text{overcount} = \frac{n_{\text{incl. true edges}}}{n_{\text{true graph edges}}}, \quad (7.17)$$

is estimating the number of these additional edges. Values greater than one refer to additional edges but do not necessarily mean that the *true graph eff.* of a graph is one. In an ideal scenario, a graph-building method should maximize the *true graph eff.* but also the *true edge pur.*. A lower *overcount* value, together with a constant *true graph eff.*, indicates that the graph-building method is more selective in including additional true edges, thus potentially reducing the computational complexity of the subsequent classification task.

## 7.4. Graph-Building Models

Because background hits in the CDC are expected to be approximately two orders of magnitude higher than signal hits [32], connecting all nodes in the CDC in a fully connected graph, results in a large number of false edges yielding a low *true edge pur.* and *true graph pur.* Thus, one of the objectives of graph-building is to minimize the number of edges connecting background hits.

Particles traversing the CDC are expected to leave hits in close proximity and with distinct patterns. Previous trigger algorithms make use of these expected patterns. The Belle II experiment uses a Track Segment Finder (TSF) [21, 49] at the trigger level, employing hourglass-shaped segments to trigger particle tracks. Details on the tracking algorithms can be found in subsection 2.4.1.

In fig. 7.7, the left side displays an hourglass-shaped segment from the TSF. The current trigger is optimized for pointing tracks originating directly from the IP. However, for searches for new physics, the expected tracks may originate from displaced vertices, which may be non-pointing to the IP. Therefore, the expected patterns in the CDC may differ.

In this section, graph-building models are introduced and analyzed, designed for the simulation presented in section 5.4. The simulation involves two signal muons that are uniformly displaced in the CDC in  $r = \sqrt{x^2 + y^2 + z^2}$  in the range of 0 cm to 100 cm. Hits originating from these particles are signal hits, while other hits in the CDC are considered background. The graph-building is performed on the simulation with two types of backgrounds. The *nominal Phase 3* background refers to the full expected background for Belle II, and the *early Phase 3* background refers to current background conditions. Details on the two simulated background conditions are given in section 5.4.

This chapter focuses on the *nominal Phase 3* backgrounds, as the *GNN-pipeline* is expected to be employed under this background condition. Additional information

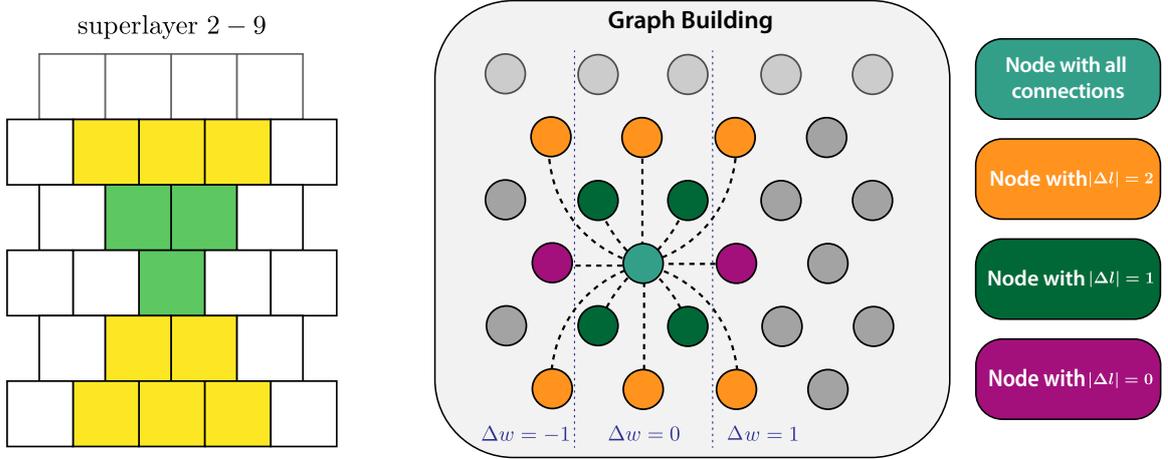


Figure 7.7.: Left: wires used for the Track Segment Finder [1] in the superlayer 2 to 9. Right: Wires considered for the graph-building models.

for the graph-building on the *early Phase 3* background can be found in appendix B. This analysis is performed using all CDC wires, thus referring to the full available information after data taking. The following analysis for the *nominal Phase 3* background is performed on 30 000 simulated events with a simulated dark Higgs mass in the range of  $m_{h'} = [1.1, 1.2, \dots, 4, 0] \text{ GeV } c^{-2}$ . Each of investigated masses is regarded with 1000 simulated events.

The two signal muons are expected to leave signal hits in spatially close sense wires. Thus, it is useful to connect spatially neighboring nodes. Adding connections to spatially further separated sense wires in the graph-building may include additional relational information in the graph. Therefore, different models are analyzed to determine the necessary allowed connections in the graph-building. The aim is to construct graphs that capture the relevant information about particle tracks in the CDC while minimizing false edges and maintaining high *true graph eff.* Figure 7.7 presents the connections considered in this analysis on the right.

First, wires with hit information in the CDC are selected for each event. Then, hits are grouped by layers, and edges are built by comparing pairs of layers iteratively. The spatial distance of two compared hits decides whether an edge is built. For this, a distance matrix is calculated using wire distances  $\Delta w$  and wire distances  $\Delta l$ .

fig. 7.8 shows exemplary the graph-building algorithm. Each node  $v_i$  in a layer in the CDC is compared to the nodes from the other layer in the layer pair with a layer distance  $\Delta l = 0, 1, 2$ . The connections are defined as undirected edges. Therefore, for connections in the same layer, only one neighbor with

$$\Delta l_{ij} = 0 \text{ and } \Delta w_{ij} = +1, \quad (7.18)$$

must be considered in the graph-building, as otherwise, the connection within the same layer would be doubled.

## 7. Graph-Building

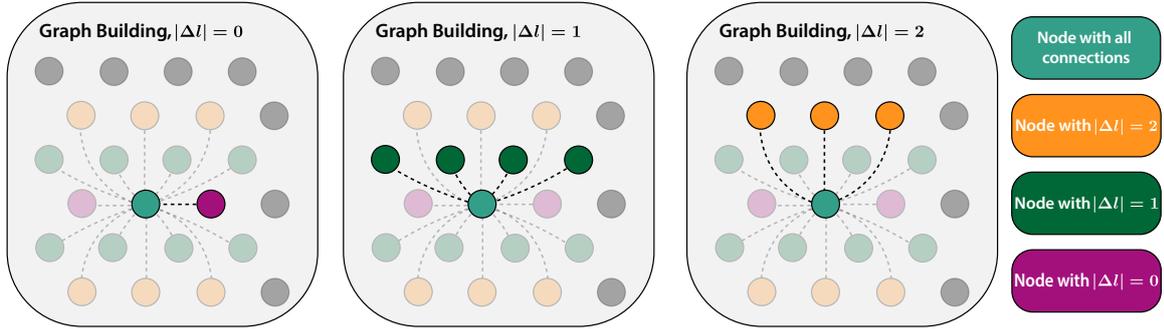


Figure 7.8.: Illustration of the implemented graph-building algorithm, here displayed for model 10. The light green central node is connected within layer pairs with  $\Delta l = 0, 1, 2$ . The graph-building process is iteratively repeated for each layer pair in the Central Drift Chamber (CDC). As the connections are built undirected, the final graph allows possible connections of the central node also downwards.

Ten different graph-building models are investigated for this analysis. They are illustrated in fig. 7.9. Starting with model 01, which introduces a graph-building algorithm allowing connections with a layer distance  $\Delta l = 1$  and a wire distance of  $\Delta w = 0$ , the different models progressively add more connections, such as connections with a wire distance of  $\Delta l = 2$  and connections in the same layer with  $\Delta l = 0$ . The latter connections are allowed to address also low  $p_T$  particles with shallow tracks in the CDC.

The best graph-building model out of the presented one is ultimately determined by analyzing its performance in the *GNN-pipeline*. Note that this thesis focuses on the edge classification with the IN, therefore evaluating the performance on this step of the *GNN-pipeline* in chapter 8.

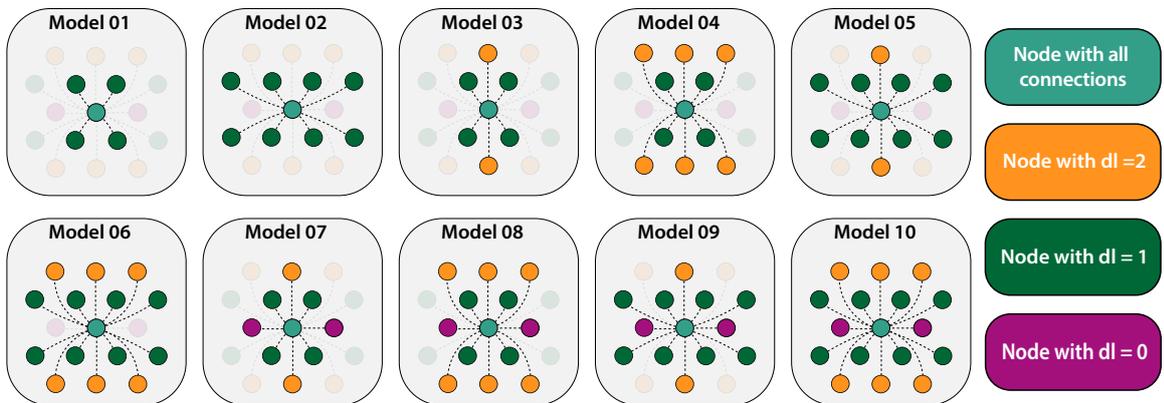


Figure 7.9.: Ten different graph-building patterns analyzed in this thesis, showing possible connections for a node in the Central Drift Chamber.

### 7.4.1. Characterization of Graph-Building Models

The ten different graph-building models are characterized based on the information provided in the tables 7.1 and 7.2 below. Values are given using the median to measure the central tendency of the events, as these contain outliers with very different amounts of signal and background hits. The median is more resistant to such outliers. Errors are approximated using the standard error on the median

$$\sigma_{\text{median}} = 1.253 \cdot \sigma_{\text{mean}}, \quad (7.19)$$

with the standard error of the mean  $\sigma_{\text{mean}}$ .

The number of edges per event generated by each graph-building method is crucial when implementing the *GNN-pipeline*. An increased number of edges generally implies increased computational demands and bandwidth requirements. This is discussed in more detail for the interaction network in section 8.4. Table 7.1 shows the median number of edges obtained for each model in total and the number of incl. graph connections and included true edge connections. The latter two are discussed in more detail in the following two sections. The total number of edges is expected to correlate directly to the maximum number of possible connections allowed by the graph-building method. table 7.1 shows the allowed connections in the last column. A Pearson correlation  $p_r = 0.9625$  between these two columns provides good evidence for this. Additionally, the number of edges is also dependent on the type of connections. Models 04 and 05 both allow a maximum of 10 neighboring nodes for a single node in the graph, but they still show significant differences in the median number of total edges. Table 7.2 shows the number of edges obtained for each possible distance, the edges within the distance of  $\Delta l = 0, 1, 2$  respectively. Indeed, the table demonstrates that the total number of edges obtained is dependent on the investigated layer distance  $\Delta l$ . In general, the total number of edges is slightly decreasing for connections with larger layer distances. One potential reason for this result may be that the graphs are built for each superlayer independently. This leads to a dependency on the permitted connections on the layer position within the superlayer. Investigating a single type of connection with  $\Delta l > 0$ , this behavior leads to a decrease of the total number of edges while  $\Delta l$  increases, as outer nodes in the superlayer have decreasing possible connections.

Table 7.1.: Median edges for the ten different graph-building models for *nominal Phase 3* background. Each event possesses a median of  $66.00 \pm 0.21$  true graph edges and  $155.0 \pm 1.9$  true edges. The table lists the total edges created from each model, including true graph edges and true edges. Additionally, the maximum number of edges one single node can have in the respective model is shown in the last column.

model	total number of edges	incl. graph	incl. true edges	max. connections
01	$3100.0 \pm 2.2$	$55.00 \pm 0.19$	$63.00 \pm 0.26$	4
02	$5810 \pm 5$	$58.00 \pm 0.19$	$74.0 \pm 0.4$	8
03	$4017 \pm 3$	$56.00 \pm 0.19$	$86.0 \pm 0.4$	6
04	$5834 \pm 5$	$57.00 \pm 0.19$	$107.0 \pm 0.5$	10
05	$6727 \pm 5$	$59.00 \pm 0.19$	$99.0 \pm 0.5$	10
06	$8545 \pm 7$	$60.00 \pm 0.19$	$120.0 \pm 0.6$	14
07	$6435 \pm 5$	$61.00 \pm 0.20$	$102.0 \pm 0.5$	8
08	$8253 \pm 6$	$62.00 \pm 0.20$	$123.0 \pm 0.6$	12
09	$9145 \pm 7$	$64.00 \pm 0.20$	$113.0 \pm 0.6$	12
10	$10961 \pm 8$	$65.00 \pm 0.21$	$134.0 \pm 0.7$	16

Table 7.2.: Model-independent median number of edges with different wire and layer distances for *nominal Phase 3* background.

layer distance	wire distance	total number of edges	max. connections
$\Delta l = 0$	$\Delta w = 1$	$2418.0 \pm 1.5$	2
$\Delta l = 1$	$\Delta w = 0$	$3100.0 \pm 2.2$	4
$\Delta l = 1$	$\Delta w = 0, 1$	$5810 \pm 5$	8
$\Delta l = 2$	$\Delta w = 0$	$917 \pm 9$	2
$\Delta l = 2$	$\Delta w = 0, 1$	$2737.0 \pm 2.5$	6

#### 7.4.2. True Edge Purity and True Edge Efficiency

The values for *true edge eff.* and *true edge pur.* obtained for the graph building models are listed in table 7.3. As anticipated, the obtained *true edge pur.* is relatively low due to the expected high background in the events, resulting in a considerably large fraction of false edges. The *true edge pur.* of all models is in the order of  $\mathcal{O}(10^{-2})$ .

Figure 7.10 shows the *true edge pur.* plotted against the total number of edges. In general, the number of false edges is expected to correlate to the total number of edges. Therefore, the *true edge pur.* should decrease with the increasing total number of edges. This behavior is confirmed in the plot, but still, some models achieve higher *true edge pur.* with a similar number of connections. In this comparison, model 04 does show a significantly higher *true edge pur.* than model 02, with a similar number

Table 7.3.: *true edge eff.* and *true edge pur.* of the ten graph-building models.

model	<i>true edge eff.</i> (%)	<i>true edge pur.</i> (%)
01	$47.44 \pm 0.10$	$2.024 \pm 0.008$
02	$51.92 \pm 0.07$	$1.274 \pm 0.006$
03	$65.47 \pm 0.17$	$2.166 \pm 0.009$
04	$83.45 \pm 0.18$	$1.845 \pm 0.008$
05	$69.58 \pm 0.14$	$1.471 \pm 0.007$
06	$88.10 \pm 0.15$	$1.401 \pm 0.007$
07	$72.29 \pm 0.14$	$1.584 \pm 0.007$
08	$90.48 \pm 0.15$	$1.489 \pm 0.007$
09	$76.47 \pm 0.11$	$1.238 \pm 0.006$
10	$95.24 \pm 0.13$	$1.221 \pm 0.006$

of edges. Therefore model 04 is significantly better at catching true edges compared to model 02.

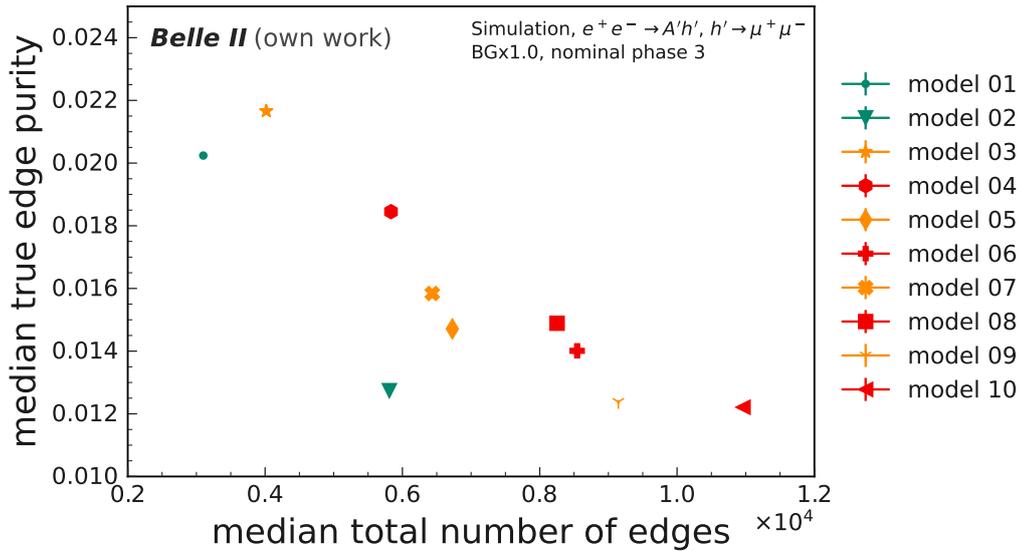


Figure 7.10.: The event median of the *true edge pur.* and the median of the total number of edges created from each graph-building model. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

The *true edge pur.* is also expected to correlate with the number of maximal allowed connections for a single node, as these are also an indicator of the increased total number of edges. This correlation is well visible in fig. 7.11. The plot also shows that the *true edge pur.* is not only dependent on the number of connections but also,

some models perform better with a similar number of maximal connections. This highlights the importance of the type of connection. The ratio of *true edge pur.* over the total number of edges is, therefore, a good indicator to understand the importance of connection types.

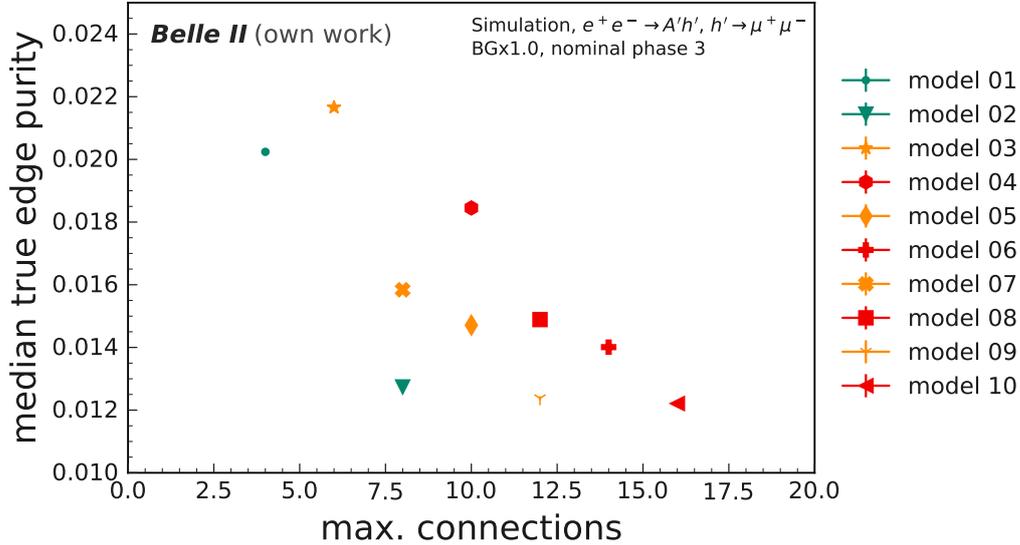


Figure 7.11.: The *true edge pur.* plotted against the maximum number of allowed connections for a single node in the graph-building models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

Figure 7.12 shows the *true edge pur.* plotted against the *true edge eff.* for the proposed graph-building models. The plot shown in this comparison features the separation of the models into three clusters, highlighted in green, yellow, and red. Each color denotes the allowed edges in a layer distance of  $\Delta l = 2$ . Introducing more allowed connections in this layer distance significantly increases the *true edge eff.*. This effect is partially explained by the calculation of the complete set of true edges, which includes connections with a layer distance of  $\Delta l = 2$ . Consequently, a method including these connections is expected to have an increased *true edge eff.*.

Figure 7.13 shows the efficiency plotted against the total number of edges, where again, it is visible that some models are far more efficient in including the true edges while keeping a similar total number of edges. Again edges in  $\Delta l = 2$  show significant importance.

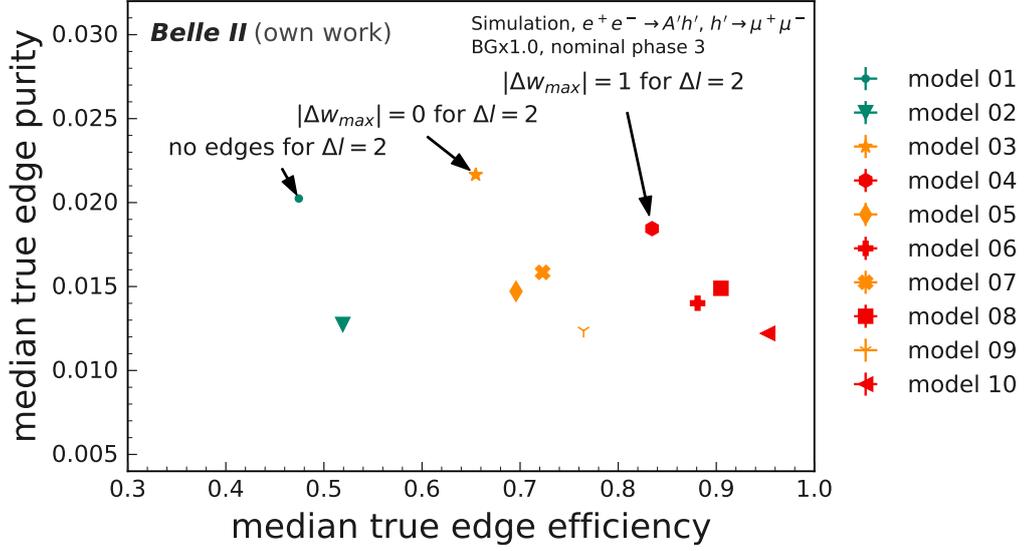


Figure 7.12.: The event median of the *true edge eff.* and *true edge pur.* The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

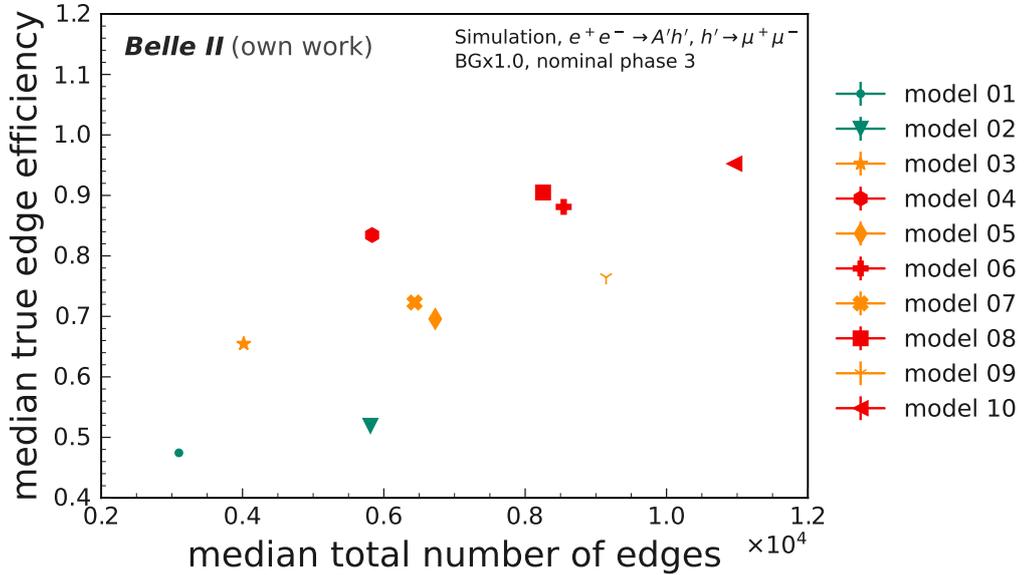


Figure 7.13.: The median of *true edge eff.* and *total number of edges* calculated for each graph-building model. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

### 7.4.3. True Graph Purity and True Graph Efficiency

Turning now to the investigation of the true graph edges, fig. 7.14 shows a scatter plot of the *true graph eff.* and the *true graph pur.* and table 7.4 shows the values tabulated. As the expected number of true graph edges is smaller than the expected number of true edges, shown in table 7.1, the *true graph pur.* is expected to be lower than the *true edge pur.* Naturally, one would expect a trade-off between *true graph eff.* and *true graph pur.* for the graph-building methods. Higher *true graph eff.* would typically lead to a decrease in *true graph pur.* and vice versa. This trade-off should manifest as a curve or line in the *true graph pur.-true graph eff.* plane where all models are located. Indeed, this behavior is approximately demonstrated in the fig. 7.14. In contrast to the plot for the true edges, the separation into three clusters is less distinct. However, a similar tendency for the three groups of models is present, and the highest *true graph eff.* is again achieved by Model 10 with an *true graph eff.* of  $(98.667 \pm 0.013) \%$ . This indicates that model 10 is the graph-building method, which is the most successful in incorporating true graph edges. The highest *true graph pur.* is achieved by Model 01 with  $(1.785 \pm 0.006) \%$ , which implies that it is the model with not only the smallest amount of edges but also the best model for including the least false edges among the considered models for this simulation. It is now subject to the edge classification task identifying the best model for the clean-up task with the interaction network.

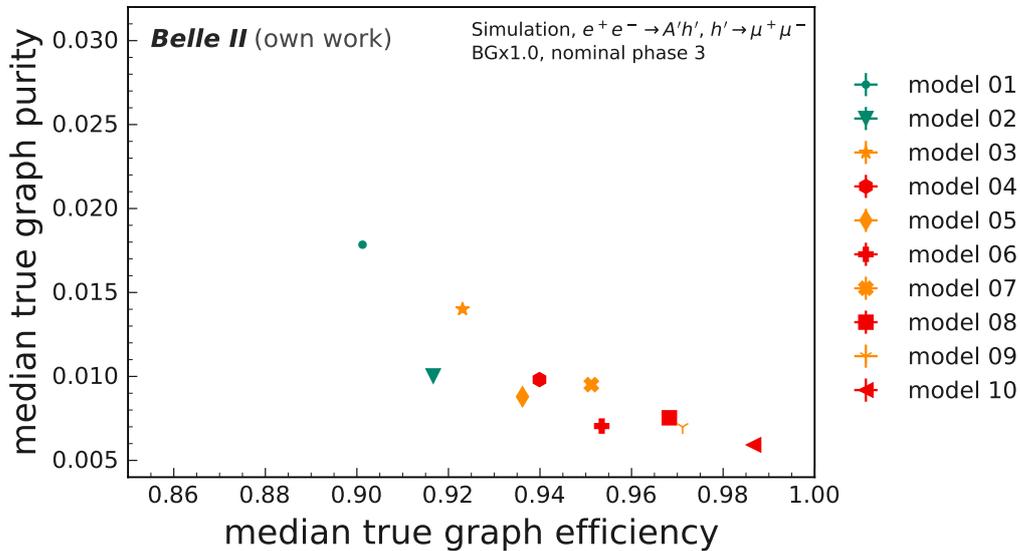


Figure 7.14.: The event median of the *true graph eff.* and *true graph pur.*. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

Table 7.4.: The *true graph eff.* and *true graph pur.* of the ten graph-building models for *nominal Phase 3* background.

Model	<i>true graph eff.</i> (%)	<i>true graph pur.</i> (%)	<i>overcount</i>
01	90.12 $\pm$ 0.10	1.785 $\pm$ 0.006	1.0087 $\pm$ 0.0013
02	91.67 $\pm$ 0.07	1.003 $\pm$ 0.004	1.0935 $\pm$ 0.0019
03	92.31 $\pm$ 0.10	1.401 $\pm$ 0.005	1.4393 $\pm$ 0.0024
04	93.99 $\pm$ 0.10	0.981 $\pm$ 0.004	1.7805 $\pm$ 0.0027
05	93.62 $\pm$ 0.07	0.8793 $\pm$ 0.0028	1.5492 $\pm$ 0.0023
06	95.35 $\pm$ 0.07	0.7045 $\pm$ 0.0023	1.8442 $\pm$ 0.0027
07	95.12 $\pm$ 0.05	0.9513 $\pm$ 0.0030	1.6053 $\pm$ 0.0022
08	96.83 $\pm$ 0.04	0.7536 $\pm$ 0.0024	1.8889 $\pm$ 0.0027
09	97.115 $\pm$ 0.018	0.6990 $\pm$ 0.0022	1.7123 $\pm$ 0.0029
10	98.667 $\pm$ 0.013	0.5924 $\pm$ 0.0019	1.989 $\pm$ 0.004

#### 7.4.4. Overcount

Figure 7.15 shows the *true graph pur.* of the graphs in relation to the *overcount* metric. It is anticipated that the *overcount* would be correlated to the number of edges generated by a graph-building method. Introducing more relational information between signal nodes but also increasing the number of backgrounds in the CDC. Therefore, leading to an increased background level in the graph but also introduces valuable knowledge.

fig. 7.16 generally confirms this expected correlation. However, it also highlights that some models have an increased *overcount* than others with a comparable number of edges. For instance, model 04 exhibits a higher *overcount* than model 02, despite both containing a similar total number of edges per event. This suggests that using model 04 yields more true edges compared to model 02, which is useful for the classification task as more relational information between the true nodes is provided for the IN.

#### 7.4.5. Recurring Patterns in the Metrics

The CDC has a specific pattern of wire distribution around the IP, resulting in recurring patterns in graph-building metrics. The scatter plot of the 30 000 graphs with the *true graph eff.* as a function of the opening angle between the two signal muons in fig. 7.17 displays distinctive signatures with densely populated lines at specific values. In this plot, the *true graph eff.* is evaluated for superlayer 2, which consists of six layers.

Since expected tracks in the CDC are typically pointing outward, a particle traversing a superlayer is expected to leave roughly one hit in each layer. Consequently, the true graph is anticipated to have approximately five edges for each signal particle

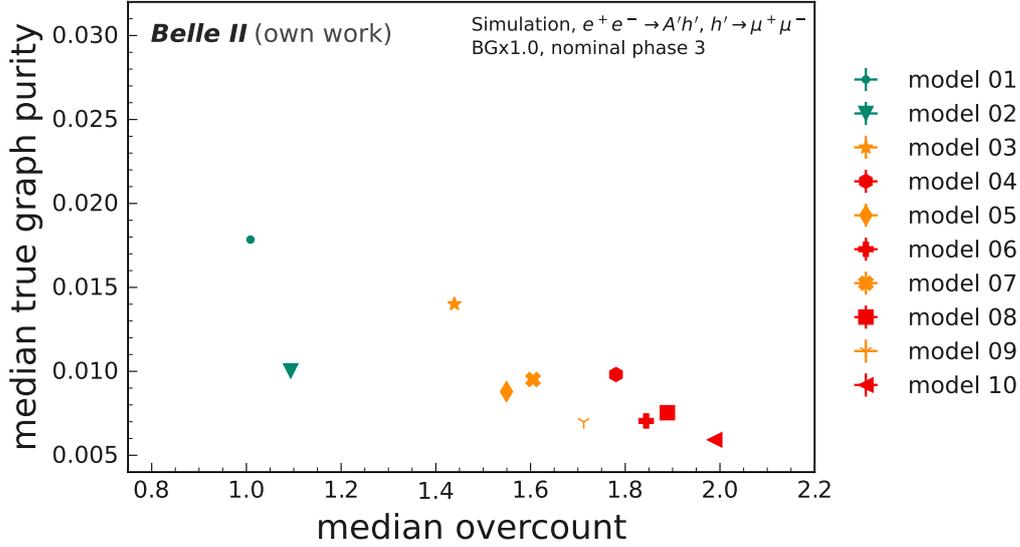


Figure 7.15.: The event median of the *true graph pur.* and the *overcount*. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

within superlayer 2. Since the *true graph eff.* calculation does take into account the included graph edges over the total number of true graph edges, a sparse distribution between zero and one is expected. Mathematically this can be expressed as

$$true\ graph\ eff. = \frac{k}{n_{true\ graph\ edges}}, \quad k \in \mathbb{N}, \quad (7.20)$$

with an expected number of  $n_{true\ graph\ edges} = 10$  for two signal particles in superlayer 2. This is verified in fig. 7.18, which shows the *true graph eff.* for superlayer 2 of the 30 000 events. The plot shows several distinct peaks. A large fraction of the events reach a *true graph eff.* of one. As tracks may curl, or originate within a specific layer, the number of  $n_{true\ graph\ edges}$  may vary for some events, allowing several other efficiency values.

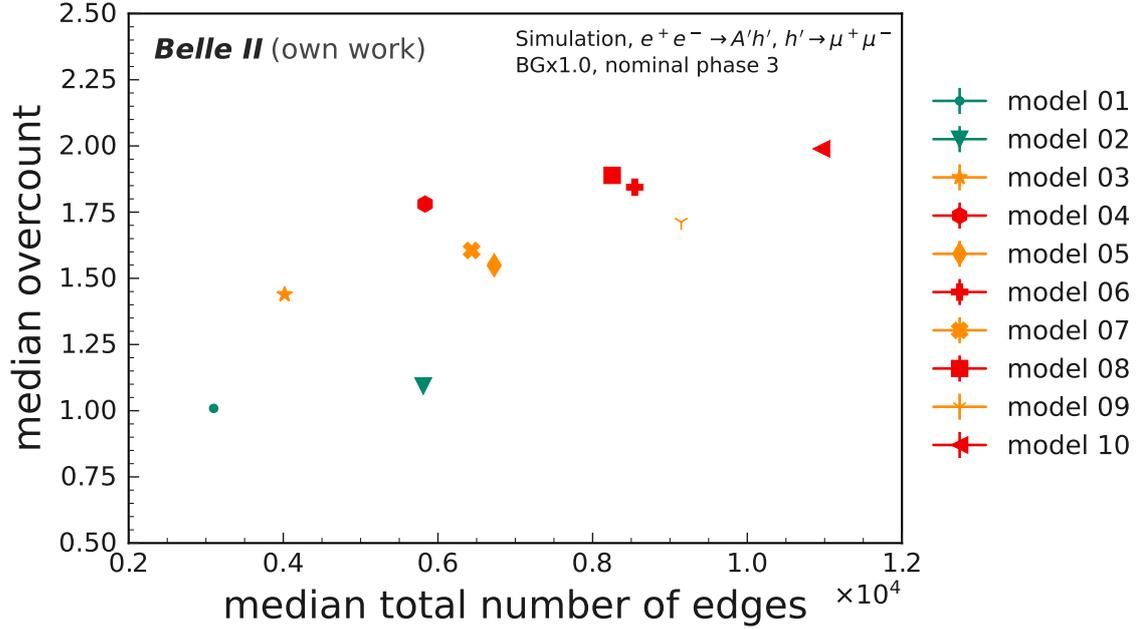


Figure 7.16.: The event median of the *overcount* plotted against the median number of edges per event of the graph-building method. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

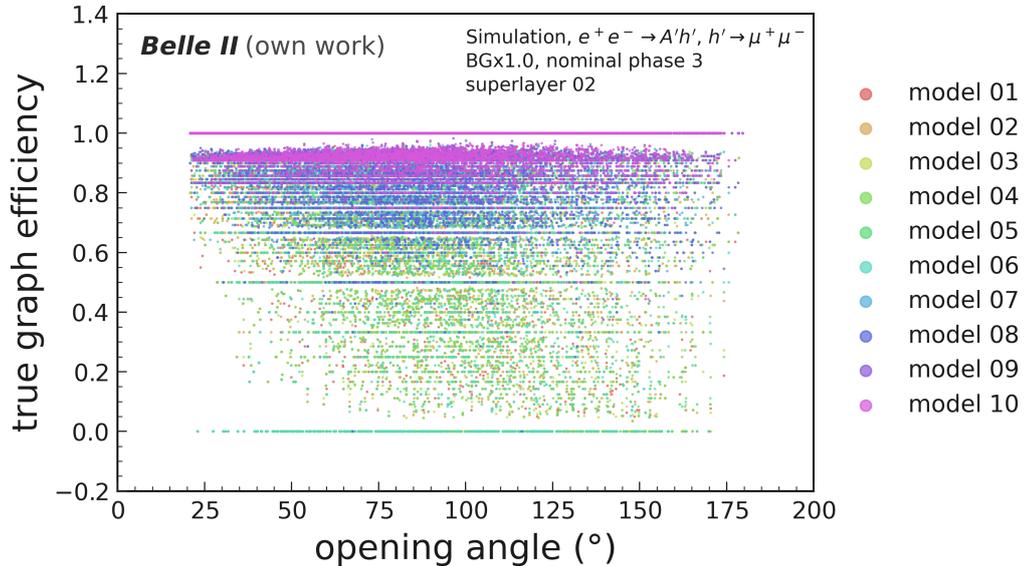


Figure 7.17.: Scatter plot of the ten graph-building models for superlayer 2. The graphic shows the simulated dimuon signal's *true graph eff.* in relation to the opening angle.

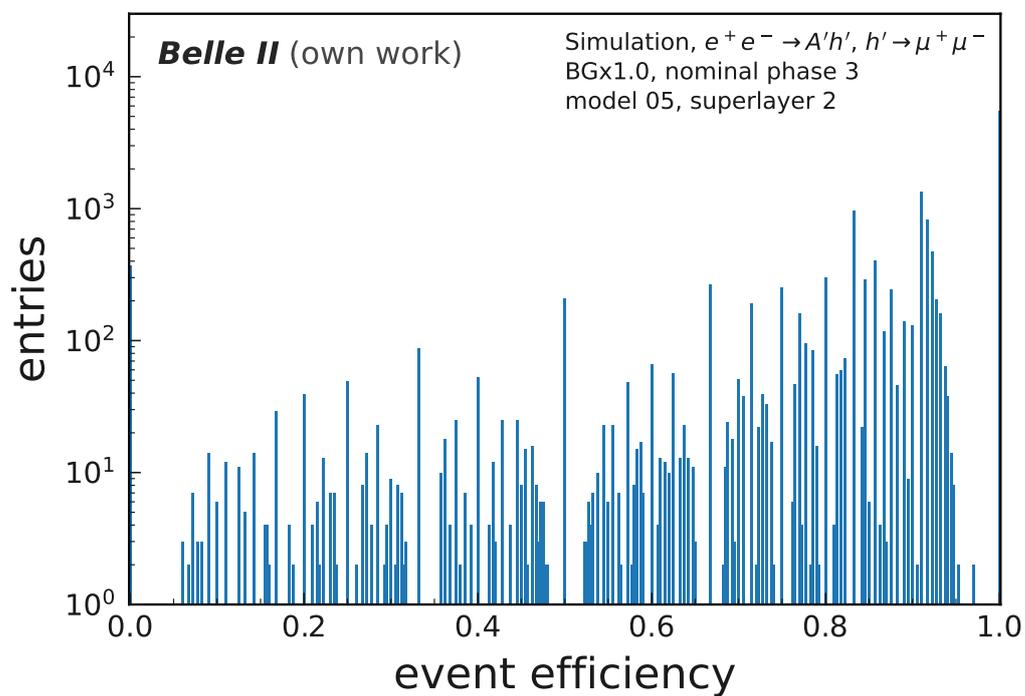


Figure 7.18.: The histogrammed *true graph eff.* shows a sparse distribution for model 05 in superlayer 2.

## 8. Background Clean-Up in then Central Drift Chamber Using the Interaction Network

In order to obtain accurate and reliable results when analyzing particle collisions, it is crucial to perform background clean-up. This process involves isolating the traces of the specific particle of interest from extraneous signals that may interfere with the later analysis. In the context of the Central Drift Chamber (CDC), the background clean-up refers to the discrimination of hits, which can be associated with a signal particle and the remaining hits, classified as background hits. In the context of machine learning, this task can be understood as pattern recognition because it involves detecting specific patterns or structures within the detector data that correspond to the underlying physics of particle collisions. A more comprehensive explanation of signal and background in the CDCs is presented in section 7.1, where the concept of true and false edges is also introduced. The tracking *GNN-pipeline*'s background clean-up fulfills three primary functions:

- (i) Reducing the computational costs for subsequent tasks of the *GNN-pipeline* by filtering the input graph, which is especially crucial in the online application.
- (ii) Increasing accuracy when utilizing the Object Condensation (OC) approach [26] to determine track parameters of signal particles in the CDC.
- (iii) The facilitation of discoveries, where effective background clean-up may enable new searches of rare signals and allow for more precise measurements, improving sensitivity.

In chapter 4, the background clean-up is introduced as part of the *GNN-pipeline*, utilizing the Interaction Network (IN). The Graph Neural Network (GNN) performs a classification task, a machine learning process that identifies the category or class to which a particular input belongs. In this context, the IN is trained to classify edges of an input graph, which correspond to either true or false edges. True edges represent connections between signal hits in the CDC, while false edges correspond to connections between background hits or a mix of signal and background hits. By successfully classifying edges, the IN is used to subsequently identify nodes in the graph that correspond to signal hits.

This chapter does focus on the analysis of the success of the classification task on the ten graph-building models presented in chapter 7. For this, an IN is trained on each graph-building model to discriminate the corresponding true and false edges. Section 8.1 presents the configuration of the IN, with respect to the available input features and section 8.2 presents metrics for the evaluation of the classification task. In the following section 8.3, metrics are evaluated to measure the performance of the classification task and the overall performance of the *GNN-pipeline* to the point of the background clean-up.

A real-time implementation of the IN must be employed on specialized hardware to process events in the sub-microsecond timescale. Therefore, the IN will be implemented on designated Field-Programmable Gate Arrays (FPGAs), which imposes certain limitations on the network and input size due to constrained computing resources. A larger input size may require more bandwidth and may affect the latency and throughput of an FPGA, while larger networks could exhaust the FPGA's memory. To gain first insights into the real-time implementation, the total number of floating point operations required by each graph-building method is estimated. This helps to understand the computational overhead better and sets the foundation for an initial discussion of implementing the IN on an FPGA. Additionally, a first comparison of the classification efficiency is presented, using input features with a reduced resolution, aligning the anticipated resolution for the Level 1 Trigger (L1 Trigger) trigger.

## 8.1. Interaction Network Training and Dataset Description

The IN is trained on graphs generated by the ten different graph-building methods proposed in chapter 7, predicting true and false edges corresponding to the definition in section 7.1. The same dataset was used for the analysis of the INs as for the graph-building. Details regarding the simulation are presented in section 5.4. This chapter analyzes the IN with the full expected beam background, respectively *nominal Phase 3* background. Studies performed on *early Phase 3* background can be found in appendix C. The dataset includes 30 000 simulated events, with a simulated dark Higgs mass in the range of  $m_{h'}$  = [1.1, 1.2, ...4, 0] GeV. Each of the investigated masses is taken into account with 1000 simulated events.

The dataset is randomly split into a training set with 24 000 events, a validation set with 3000 events, and an evaluation set with 3000 events. The random splitting of the dataset leads to a varying number of events in the evaluation dataset for each mass in the specified range, resulting in a diverse representation of masses. This diversity is taken into account by calculating the median value of the events in the evaluation dataset and then determining the uncertainties based on the statistical error derived from the events' distribution. The error is estimated by

$$\sigma_{\text{median}} = 1.253 \cdot \sigma_{\text{mean}}. \quad (8.1)$$

Input features for the classification task are the spatial position of the CDC hits,  $x$ , and  $y$ , time information of a hit in the form of the Time-to-Digital Converter (TDC) count, and information about the deposited energy in a hit, the Analog-to-Digital Converters (ADCs) count. The spatial information is valuable because hits of a particle in the CDCs are expected to be close, forming a path of consecutive hits in the CDC. The TDC count and the ADC provide additional orthogonal discriminatory information between signal and background hits, which has been shown in the chapter 6.

The edge attributes of the input graphs represent relational spatial information between the connected nodes in cylindrical coordinates, specifically  $\Delta\rho$  and  $\Delta\phi$ . Here,  $\rho$  does correspond to the spatial distance in the  $x - y$  plane of the detector, and  $\phi$  is the respective angle in the plane.

### 8.1.1. Implementation in PyTorch Geometric

In this PyTorch implementation, the IN consists of two Relational Models and one Object Model, comprising a total of 1987 trainable parameters. In the IN, information flows from a sender to a receiver through directed processes. As a result, the edges generated during the graph-building process are changed into undirected edges. In PyTorch Geometric (PyG), this is achieved by duplicating the edges with opposite directions. The layer sizes of the IN were selected to keep the number of parameters relatively small.

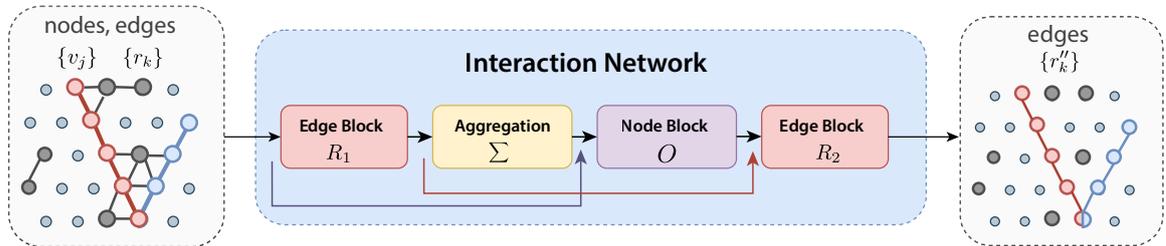


Figure 8.1.: An illustration of the Interaction Network implemented for this thesis. The network comprises two relational models and one object model. The input to the Graph Neural Network is a graph consisting of nodes  $v_i$  and edges  $r_k$ , and the output delivers classified edges  $r''_k$ . This figure is adapted courtesy of Lea Reuter [44] and is based on [25].

The same model architecture is used for *nominal Phase 3* and *early Phase 3* samples. Figure 8.1 shows an illustration of the IN implemented for this analysis and in table 8.1 the model is summarized. A detailed mathematical description of the IN is given in subsection 4.2.1. The network consists of three submodels and an aggregation block. Each of the four components implemented in PyG is introduced below:

**Relational Model  $R_1$**  The model  $R_1$  is responsible for computing the effects of pairwise interactions between the nodes in the system. It does compute updated edge

states  $r'_k$ . It consists of three fully connected layers with Rectified Linear Unit (ReLU) activation functions in between. The first linear layer has 220 parameters, followed by a second linear layer with 420 parameters, and a final linear layer with 42 parameters.

**Aggregation** The aggregation step is implemented with a summation function. In this step, the updated edges (messages)  $r'_k$  from the relational network  $R_1$  are aggregated for each node. This step intrinsically uses directed edges, and therefore it is important that the input graph has undirected edges to use the relational information for each node. The output for each node is denoted as  $\tilde{r}'_i$ .

**Object Model  $O$**  The object model  $O$  processes the effects computed by the relational model and calculates updated object states  $v'_i$  accordingly. The input of the network is the aggregation out  $\tilde{r}'_i$  along with the initial input nodes  $v_i$ . This network also comprises three fully connected layers with ReLU activation functions in between. The first linear layer has 140 parameters, followed by a second linear layer with 420 parameters, and a final linear layer with 84 parameters.

**Relational Model  $R_2$**  The second relational model  $R_2$  computes pairwise interactions, considering the updated object states  $v'_i$ . The structure is similar to the first relational model, with three fully connected layers and ReLU activation functions. The first linear layer has 220 parameters, the second has 420 parameters, and the third has 21 parameters. This model gives a one-dimensional output where a sigmoid function is applied, creating the final output.

Each model has one hidden layer, an input layer, and an output layer. The size of the hidden dimension is 20 for the hidden layers, and the input dimension is determined from the input graph, as is the output dimension of the intermediate blocks, as they are aligned with the input graph dimensions.

Training the network aims to adjust the model's weights and biases to reduce the discrepancy between its predictions and the true labels. For this, a loss function is defined, which does represent the divergence of prediction and labels. More details on the motivation for training are presented in subsection 4.2.3. Backpropagation is performed to minimize the loss function using gradient descent. This updates the model's weights and biases by iteratively moving in the direction of the steepest decrease of the loss function.

The learning rate is a parameter that determines the step size an optimizer performs during the gradient descent toward the loss function. It does influence the speed and convergence of the optimization process. For the IN, the learning rate is adjusted using the learning rate scheduler `ReduceLROnPlateau` from PyTorch. This scheduler adjusts the learning rate if no loss improvement is detected for a defined number of epochs. Two parameters are specified, the *decay factor*, by which the learning rate is reduced, and the *patience*, which defines the number of epochs the scheduler takes into account before reducing the learning rate.

During training, an optimizer is used to maintain an individual learning rate for each parameter. For this training, the Adaptive Moment Estimation (Adam) optimizer [27] is used. The Adam optimizer computes adaptive learning rates based on the first and second moments of the gradients. It combines properties of two other optimization techniques, AdaGrad [11] and RMSProp [19].

Table 8.1.: Model summary of the Interaction Network implemented in PyTorch with a total of 1.987 trainable parameters.

Block	Layer Type	Number of Parameters
Relational Model 1		
	Linear	220
	ReLU	
	Linear	420
	ReLU	
	Linear	42
Object Model 1		
	Linear	140
	ReLU	
	Linear	420
	ReLU	
	Linear	84
Relational Model 2		
	Linear	220
	ReLU	
	Linear	420
	ReLU	
	Linear	21

### 8.1.2. Training of the Interaction Networks

The training of the IN is carried out with a batch size of 1024 events for graph-building models 01 to 05 and model 07 and a batch size of 512 events for the remaining models. The batch size was adjusted to accommodate the memory constraints on the hardware, as some models contained larger graphs that required more memory on the Graphics Processing Units (GPUs). The hyperparameters for the training are displayed in table 8.2. Ten separate trainings were conducted for each of the proposed graph-building models from chapter 7, resulting in ten trained IN.

One training epoch corresponds to the training of the IN on the entire training data set. The backpropagation of the network is performed after each batch iteration using the mean cross-entropy loss. It is important to note that the size of the training set varies as the number the edges depends on the graph-building method. Due to

this, the mean loss is calculated from a very different number of edges. Additionally, due to the different batch sizes, the number of updates performed on the parameters differs for the different IN.

Verifying the convergence of each model’s loss is therefore essential, as this is typically taken as an indication that the network has reached maximum learning potential. Figure 8.2 shows the losses on the training datasets after each epoch and fig. 8.3 shows the losses of the corresponding validation datasets. Both presented loss functions exhibit good convergence. The validation loss is particularly important, as it measures the network’s generalization capability. This is because events from the validation dataset are not used to optimize the IN. This importance becomes evident when analyzing overfitting. Overfitting corresponds to a network that learns not only the underlying patterns of a dataset but also the noise, causing the models to learn the training data too well. As a result, the model performs poorly on unseen data. This effect may be noticeable during training as an increase in validation loss and a continuous decrease in training loss. Therefore, monitoring the relationship between the two losses is essential.

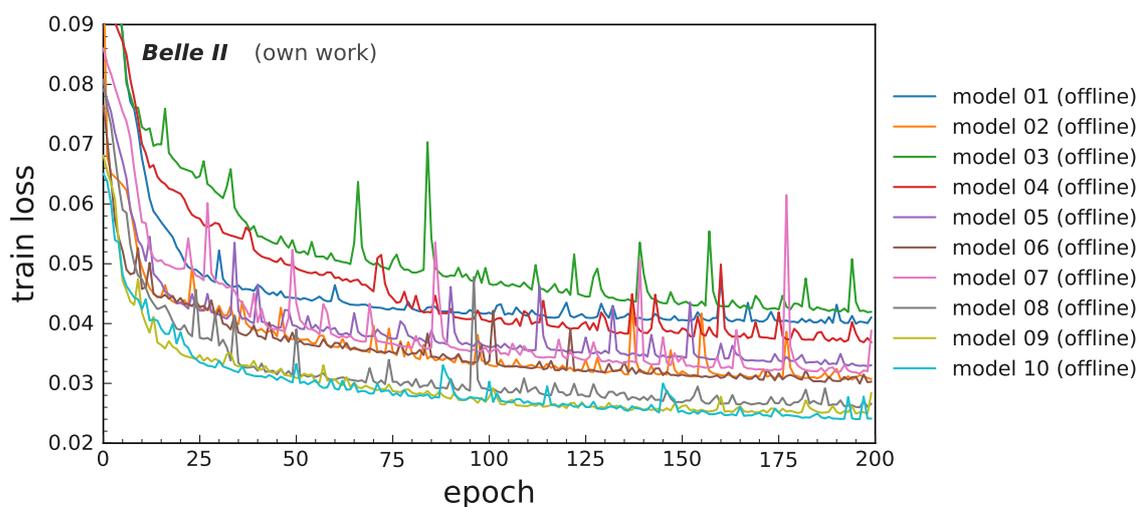
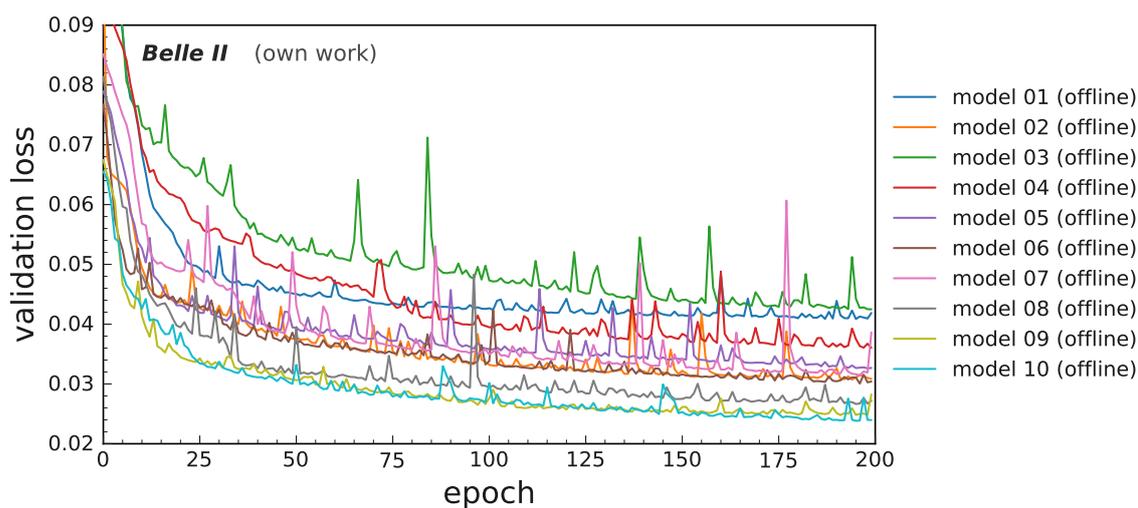
After 100 to 150 epochs, both losses show minimal improvement, indicating well-trained models. Artifacts like the peaks in the losses may result from a slightly too high learning rate, which allows the model to overshoot the loss function’s minima. Nevertheless, the loss functions demonstrate adequate training of the ten networks. Throughout the training process of the ten IN, the optimal model is determined by selecting the one that has the lowest validation loss across all epochs.

Table 8.2.: Hyperparameters for training of the Interaction Network.

Hyperparameter	Value
Object Model Hidden Size	20
Relational Model 1 Hidden Size	20
Relational Model 2 Hidden Size	20
learning rate	0.0075
learning rate scheduler factor	0.2
learning rate scheduler patience	15
Number of Epochs	200
Batch size	1024 & 512

## 8.2. Metrics for the Classification Task

Evaluating the performance of the IN is important to determine how well it cleans up the background and retains valuable information for the *GNN-pipeline*. In the following subsection, various metrics are introduced for assessing the classification

Figure 8.2.: Train loss the training with *nominal Phase 3* background.Figure 8.3.: Validation loss the training with *nominal Phase 3* background.

performance of the ten models based on the metrics used before for the graph-building in chapter 7.

In section 7.3, several metrics are introduced to investigate the performance of the graph-building. The metrics defined in the graph-building determine purities and efficiencies of the graph-building by differences of the edges included before and after the graph-building. A similar approach is employed for assessing the classification performance of the INs. First, the analogon to the included true edges after the classification is defined as the included predicted true edges, with its number defined

as

$$n_{\text{incl. pred. true edges}} = |\{x|x \in \text{complete set of true edges} \wedge x \in \text{predicted true edge}\}|. \quad (8.2)$$

Similarly, the included predicted true graph edges are defined as

$$n_{\text{incl. pred. true graph edges}} = |\{x|x \in \text{true graph edges} \wedge x \in \text{predicted true edge}\}|. \quad (8.3)$$

These two numbers reflect the fraction of true edges and true graph edges included after the classification step. Regarding the classification task performed on the graphs,  $n_{\text{incl. pred. true edges}}$  can also be referred to as the true positives.

### 8.2.1. Definition of Classified Purities

In a classification task, purity is a metric that measures the proportion of correctly classified positive instances out of all instances classified as positive. In other words, purity represents the fraction of true positive predictions among all positive predictions. To account for this interpretation, definitions of the *true graph pur.* and *true edge pur.* from chapter 7 are revised.

The included number of edges  $n_{\text{incl. true edges}}$  and  $n_{\text{incl. true graph edges}}$  are replaced with  $n_{\text{incl. pred. true edges}}$  and  $n_{\text{incl. pred. true graph edges}}$ , respectively. Furthermore, the number of edges in the denominator is represented by  $n_{\text{total predicted edges}}$ . The purities after classification are given as

$$\text{class. true edge pur.} = \frac{n_{\text{incl. pred. true edges}}}{n_{\text{total predicted edges}}}, \quad (8.4)$$

$$\text{class. true graph pur.} = \frac{n_{\text{incl. pred. true graph edges}}}{n_{\text{total predicted edges}}}. \quad (8.5)$$

A high purity indicates that the classifier successfully discriminates true edges and true graph edges, which is essential for the effectiveness of the *GNN-pipeline*. A low purity indicates a noisy classifier selection, as many false edges are included in its predictions.

### 8.2.2. Definition of Classified Efficiencies

The definition of *true graph eff.* and *true edge eff.* are also revised for the performance measurement of the classification. Typically efficiency is determined by calculating the proportion of a subset that successfully passes a classification task from an entire set. For the classification with the IN, this entire set is defined as the included fraction of the true graph edges and true edges in the graphs. Therefore, the classified true

edge efficiency is defined as

$$\begin{aligned} \text{class. true edge eff.} &= \frac{n_{\text{incl. pred. true edges}}}{n_{\text{incl. true edges}}} \\ &= \frac{n_{\text{incl. pred. true edges}}}{n_{\text{true edges}}} \cdot \frac{1}{\text{true edge eff.}}, \end{aligned} \quad (8.6)$$

and the classified true graph efficiency

$$\begin{aligned} \text{class. true graph eff.} &= \frac{n_{\text{incl. pred. true graph edges}}}{n_{\text{incl. true graph edges}}} \\ &= \frac{n_{\text{incl. pred. true graph edges}}}{n_{\text{true graph edges}}} \cdot \frac{1}{\text{true graph eff.}}. \end{aligned} \quad (8.7)$$

Both newly defined efficiencies not only make use of the prediction output of the classification but also change the set of available edges for the calculation to the input of the classification, respectively  $n_{\text{incl. true edges}}$  and the  $n_{\text{incl. true graph edges}}$ . This approach allows the efficiency to focus more on the performance of the classification task itself without directly involving the underlying graph-building method.

Overall efficiency can be determined by multiplying each classification efficiency with their counterparts from the graph-building to get insights into the efficiency of incorporating the true graph edges and true edges of the complete sets presented in section 7.3.

A high efficiency indicates that the classifier successfully finds true edges or the true graph edges. This is crucial for the *GNN-pipeline*, as these edges are expected to be helpful in the following OC task. A low efficiency implies that the classifier misses a significant fraction of true edges or the true graph edges.

### 8.2.3. Threshold Determination Using the F1 Score

The output of the IN is continuous, with a value between zero and one for each edge. The output is converted into a discrete class type using a threshold value. It is important to note that the choice of the threshold does influence *class. true edge pur.* and *class. true graph eff.* The final choice involves finding a balance based on prioritized goals. Since this decision depends on the subsequent requirements of the OC and the whole *GNN-pipeline*, this initial analysis focuses on optimizing the *F1 score*, thus optimizing the harmonic mean of the two metrics. The *F1 score* is defined as

$$F1_{\text{true\_edge}} = 2 \frac{\text{class. true edge pur.} \times \text{class. true edge eff.}}{\text{class. true edge pur.} + \text{class. true edge eff.}}. \quad (8.8)$$

To determine the optimal threshold  $t_{\text{class}}$ , the *F1 score* is calculated for a range of thresholds in  $t_{\text{class}} = [0.01, 0.02, \dots, 0.99]$ . The threshold that maximizes the *F1 score* of the validation dataset is considered to be the optimal threshold.

### 8.2.4. Classification of Nodes

The previous sections focused on metrics for evaluating the classified edges, which is the output of the IN. These metrics target an intermediate output of the background clean-up. As the emphasis now shifts to the primary goal of this chapter, the following section introduced metrics, focusing on the background clean-up. In the final step, nodes connected to a true edge are kept, while nodes without connections to a predicted node are classified as background. For the final analysis, two metrics are considered to evaluate the performance of the node classification. The node efficiency

$$node\ efficiency = \frac{n_{\text{predicted signal}}}{n_{\text{signal nodes}}} \quad (8.9)$$

with the number of true nodes  $n_{\text{predicted signal}}$ , included after the background clean-up task, and the total number of signal nodes in the event  $n_{\text{signal nodes}}$ . The node purity is defined as the fraction of predicted true nodes out of the total number of predicted nodes  $n_{\text{predicted nodes}}$

$$node\ purity = \frac{n_{\text{predicted signal}}}{n_{\text{predicted nodes}}}. \quad (8.10)$$

The *node efficiency* quantifies the effectiveness of the *GNN-pipeline* to retain the signal nodes up to the background clean-up stage, which is critical for the success of the subsequent track-finding and track-fitting processes. Thus, achieving a high *node efficiency* is essential for ensuring accurate and high-quality results. The *node purity*, on the other hand, gives insights into the output of the classification task. Its result reflects the *GNN-pipeline's* ability to discriminate between background and signal nodes. It is important to note that the *node efficiency* generally has a more direct and significant impact on the accuracy and quality of track reconstruction. This is especially true, as subsequent steps of the *GNN-pipeline* also provide additional background discrimination.

## 8.3. Performance Measurements

Accurate performance measurements not only help evaluate the effectiveness of the IN but also contribute to optimizing the track-finding task, which is one of the ultimate goals of the *GNN-pipeline*. The following section aims to investigate the performance of the ten INs to understand which metrics are powerful to determine the performance of the INs. Ultimately, the performance of the background clean-up using the classified edges of the INs is determined.

### 8.3.1. F1 Score of the True Edges

The ten graph-building methods produce input graphs with an unbalanced dataset, as the majority of included edges are not the edges of interest. This results in low *true*

*edge pur.*. When evaluating the classification performance of such a dataset, using the well-established accuracy metric alone may not provide enough information, as high accuracy could be achieved by identifying a high number of false edges without accurately classifying the true edges. In this case, the *F1 score* is a more useful metric, as it measures the purity and efficiency of the minority class, which is the true edges. This section evaluates the *F1 score* defined in eq. (8.8) for the true edge classification over different Higgs masses for the ten models. Figure 8.4 shows the median *F1 score*, evaluated over the different higgs masses. A high *F1 score* indicates that both the *class. true edge pur.* and the *class. true edge eff.* are well-balanced. The IN trained with the ten different graph-building methods presents an *F1 score* with values between 0.6 and 0.8. These results remain consistent over the investigated dark Higgs masses, with a slight decrease of the *F1 score* for higher simulated Higgs masses.

The decline in *F1 score* for higher masses may be due to the increased complexity of events with larger Higgs masses, which are expected to have wider opening angles and, therefore, can have two signal tracks with more complex displaced signatures. The IN trained on model 08 consistently demonstrates the best *F1 score* performance of the ten models.

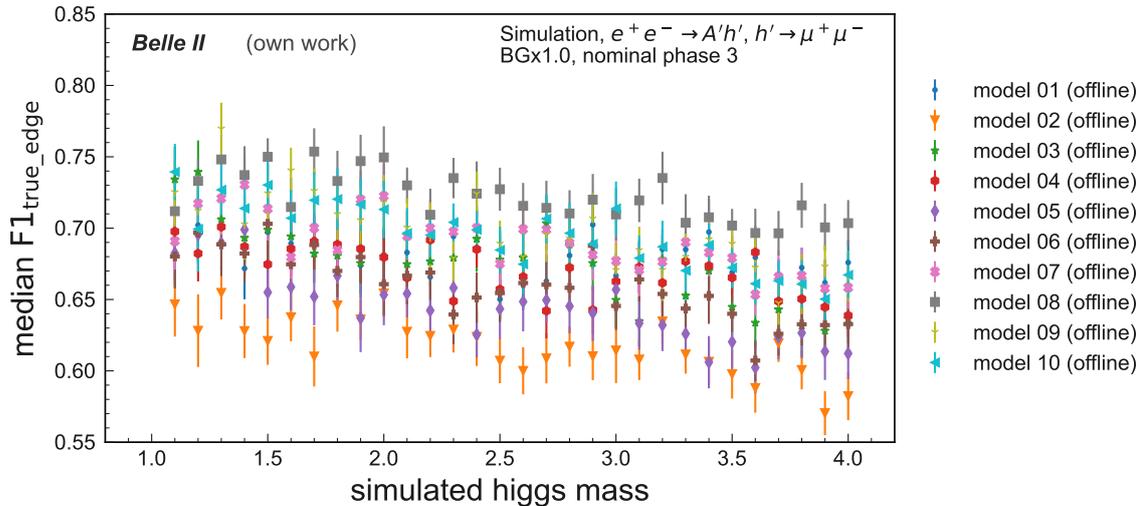


Figure 8.4.: Median *F1 score* for the true edges of the classification task for different simulated Higgs masses. The ten Interaction Networks provide consistent results across the investigated dark Higgs masses with a slight decrease for higher simulated masses. Model 08 consistently demonstrates the best *F1 score* performance among the ten models.

When examining an IN with a relatively low *F1 score*, like the one trained on model 02, it is helpful to examine the efficiency-purity plane to determine whether low purity or efficiency is the primary contributing factor. Figure 8.5 shows both values for each trained neural network. The ten INs align along a line with a strong positive

correlation between *class. true edge eff.* and *class. true edge pur.* of  $p = 0.953$ . This indicates that when one of these metrics improves, the other is likely to improve. This confirms the training of the networks on the harmonic mean of both and makes it evident that model 08 is the best-performing network, achieving the highest *class. true edge pur.* and *class. true edge eff.*. The highest *F1 score* can be achieved from networks trained on models 06 to model 10, including same-layer connections to their next neighbor. This underlines the importance of same-layer connections for analyzing the displaced dark Higgs samples. The increased efficiency may be due to particles with concentric tracks to the Interaction Point (IP), which can leave hits with parallel signatures to the layers of the CDC. These signatures can occur with low  $p_T$  tracks but can also result from displaced tracks, such as the ones simulated for this analysis. Therefore, the additional same-layer connections might be specifically helpful for finding hits of tracks originating from displaced vertices.

Model 10 incorporates the same and additional connections as model 08, but the network trained on this graph-building model shows a slightly worse *F1 score*, with lower *class. true edge pur.* and *class. true edge eff.* than model 08. This implies that the IN used for this analysis does not benefit from the additional relational information in the nodes with a wire distance of  $|\Delta w| = 1$  in the layer distance  $\Delta l = 1$ . In fact, the additional edges of model 10 seem to increase the difficulty of the classification process for the IN.

A more detailed view of the performance of the *GNN-pipeline* up to the classification task can be obtained by calculating the total efficiency. The *total true edge eff.* is given as the product of the *true edge eff.* and the *class. true edge eff.*

$$\text{total true edge eff.} = \text{true edge eff.} \times \text{class. true edge eff.} \quad (8.11)$$

$$= \frac{n_{\text{incl. pred. true edges}}}{n_{\text{true edges}}}. \quad (8.12)$$

The *total true edge eff.* provides a detailed overview of the remaining true edges from the complete set of true edges after completing the classification task. It allows a detailed comparison of the ten models. Table 8.3 displays three metrics, the *true edge eff.*, which measures efficiency in graph-building, the *class. true edge eff.*, which measures efficiency in classification tasks and the *total true edge eff.*, which measures the efficiency of the combined tasks. The results indicate that model 08 and model 10 provide the highest *total true edge eff.* scores, and model 10 does outperform the efficiency of model 08 slightly. Although model 10 does not benefit from more connections than model 08 in the classification task, it does so in the combined step of graph-building and edge classification.

### 8.3.2. F1 Score of the True Graph

The IN is trained on predicting the true edge in the graph. Nevertheless, it is interesting to investigate whether these true edges correspond to true graph edges,

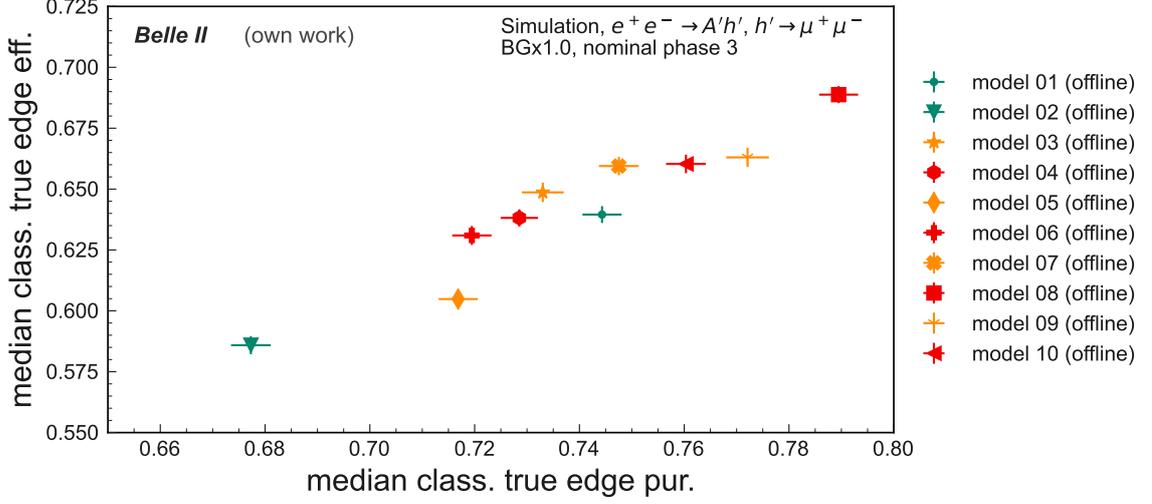


Figure 8.5.: Median *class. true edge eff.* and *class. true edge pur.* of the ten investigated Interaction Network. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ . Model 08 achieves the highest *class. true edge pur.* and *class. true edge eff.*.

Table 8.3.: The *true edge eff.*, the *class. true edge eff.*, and the *total true edge eff.* for models 01 to 10. Values are represented as percentages.

Model	<i>true edge eff.</i> (%)	<i>class. true edge eff.</i> (%)	<i>total true edge eff.</i> (%)
01	$47.44 \pm 0.10$	$64.0 \pm 0.4$	$30.79 \pm 0.21$
02	$51.92 \pm 0.07$	$58.6 \pm 0.4$	$32.61 \pm 0.22$
03	$65.47 \pm 0.17$	$64.9 \pm 0.4$	$41.0 \pm 0.4$
04	$83.45 \pm 0.18$	$63.8 \pm 0.4$	$52.2 \pm 0.4$
05	$69.58 \pm 0.14$	$60.5 \pm 0.5$	$42.65 \pm 0.30$
06	$88.10 \pm 0.15$	$63.1 \pm 0.4$	$56.7 \pm 0.4$
07	$72.29 \pm 0.14$	$65.9 \pm 0.4$	$47.6 \pm 0.4$
08	$90.48 \pm 0.15$	$68.9 \pm 0.4$	$62.7 \pm 0.4$
09	$76.47 \pm 0.11$	$66.3 \pm 0.4$	$51.5 \pm 0.4$
10	$95.24 \pm 0.13$	$66.0 \pm 0.4$	$65.2 \pm 0.4$

meaning edges associated with the particle's estimated path. The *F1 score* definition for the true graph edges is given as

$$F1_{\text{true\_graph}} = 2 \frac{\text{class. true graph pur.} \times \text{class. true graph eff.}}{\text{class. true graph pur.} + \text{class. true graph eff.}}. \quad (8.13)$$

## 8. Background Clean-Up in then Central Drift Chamber Using the Interaction Network

Figure 8.6 shows the  $F1$  score as a function of the simulated dark Higgs mass. Interestingly, the graphic presents a strong contrast in the  $F1$  score. Model 01 and model 02 have significantly better scores compared to the remaining models, contrasting with the  $F1_{\text{true\_edge}}$ , where model 02 achieved the lowest scores.

After the graph-building process, the *class. true graph pur.* is expected to be lower than the *class. true edge pur.*, as the former considers a subset of the latter. This is due to the incl. true graph edges being a subset of the incl. true edges, formally

$$\text{incl. true graph edges} \subseteq \text{incl. true edges.} \quad (8.14)$$

This is a result of the used graph-building methods, which only consider edges, which can be included in both, the true graph and the complete set of true graph edges. By default, each true graph edge is also a true edge.

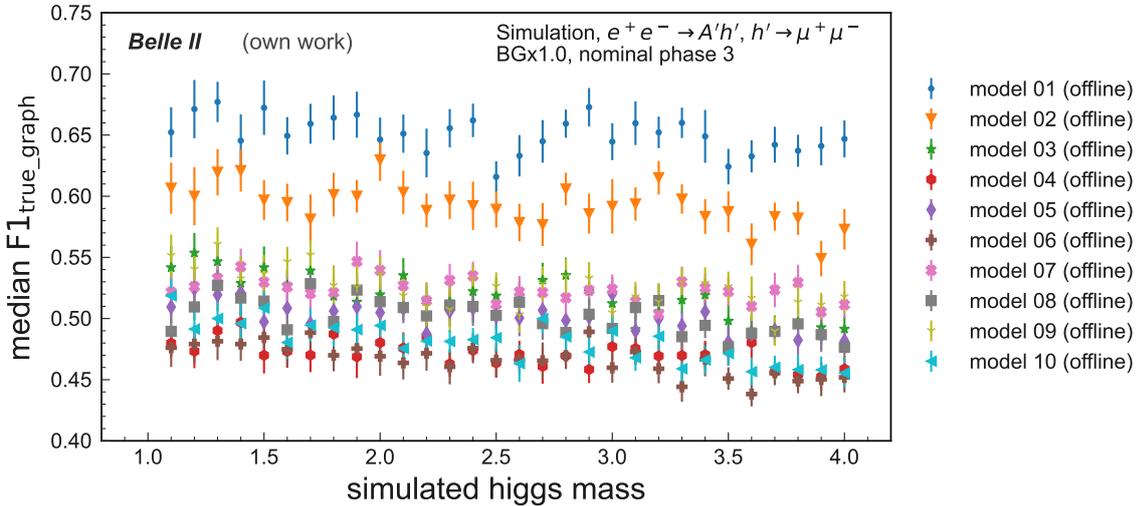


Figure 8.6.: Median  $F1$  score for the true graph edges of the classification task for different simulated Higgs masses.

Figure 8.7 presents the efficiency-purity plane. The models' purity shows large differences, with models 01 and 02 having a much higher *class. true graph pur.*, which explains the higher  $F1$  score of these models. This behavior can be explained by the fact that the networks are not optimized for the best  $F1_{\text{true\_graph}}$  score but for the highest  $F1_{\text{true\_edge}}$  score, as the target edges for the classification are all true edges for the final background clean-up. Therefore, the model is not optimized to minimize the predicted edges to the true graph edges but to a superset of these. The network is not trained to improve the *class. true graph pur.*, as all true edges are utilized for the subsequent background clean-up. The ten models are clearly separated into three regions where *class. true edge pur.* decreases significantly with an increased allowed wire distance in the layer distance  $\Delta l = 2$ . This implies that most included edges at this layer distance are not true graph edges.

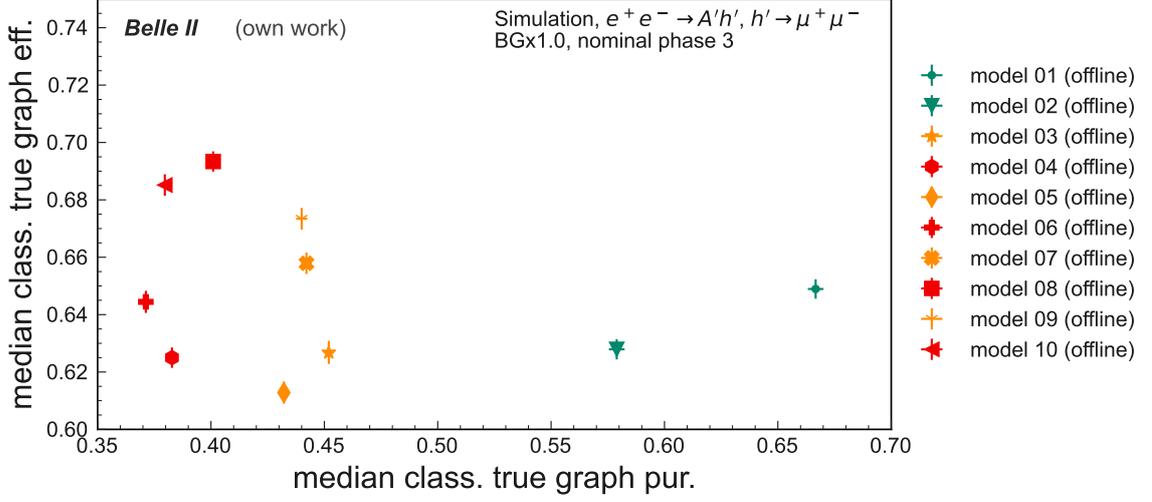


Figure 8.7.: Median *class. true graph eff.* and *class. true graph pur.* of the ten investigated models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models have no allowed edges, yellow ones allow edges up to a wire distance of  $\Delta w = 1$ , and red-labeled models allow connections with a maximum wire distance of  $\Delta w = 2$ .

As the incl. true graph edges are a subset of the incl. true edges, the IN is implicitly trained to increase the *class. true edge eff.*. To analyze the predicted true graph edges, *class. true graph eff.* is the metric of interest. Again, model 08 presents the highest classification efficiency.

Table 8.4.: The *true graph eff.*, the *class. true graph eff.*, and the *total true graph eff.* for models 01 to 10. Values are represented as percentages.

model	<i>class. true graph eff.</i> (%)	<i>true graph eff.</i> (%)	<i>total true graph eff.</i> (%)
01	$64.9 \pm 0.4$	$90.12 \pm 0.10$	$58.5 \pm 0.4$
02	$62.8 \pm 0.4$	$91.67 \pm 0.07$	$57.6 \pm 0.4$
03	$62.7 \pm 0.5$	$92.31 \pm 0.10$	$57.9 \pm 0.5$
04	$62.5 \pm 0.4$	$93.99 \pm 0.10$	$58.7 \pm 0.4$
05	$61.3 \pm 0.4$	$93.62 \pm 0.07$	$57.4 \pm 0.4$
06	$64.4 \pm 0.4$	$95.35 \pm 0.07$	$61.4 \pm 0.4$
07	$65.8 \pm 0.4$	$95.12 \pm 0.05$	$62.6 \pm 0.4$
08	$69.3 \pm 0.4$	$96.83 \pm 0.04$	$67.1 \pm 0.4$
09	$67.3 \pm 0.4$	$97.115 \pm 0.018$	$65.4 \pm 0.4$
10	$68.5 \pm 0.4$	$98.667 \pm 0.013$	$67.6 \pm 0.4$

For further investigation of the true graph, a total true graph efficiency is calculated

$$total\ true\ graph\ eff. = true\ graph\ eff. \times class.\ true\ graph\ eff. \quad (8.15)$$

$$= \frac{n_{incl.\ pred.\ true\ graph\ edges}}{n_{true\ graph\ edges}}. \quad (8.16)$$

This efficiency provides a detailed overview of the remaining true graph edges resulting from the true graph estimation after the classification. Table 8.4 shows the calculated efficiencies. Model 08 and model 10 again achieve the highest efficiency.

Comparing models 03 to 06 with models 07 to 10, the added same-layer connections led to a significant increase in *class. true graph eff.* with a comparably small influence on *class. true graph pur.*. Thus, connections with a layer distance of  $\Delta l = 0$  again seem to be crucial for improving the classification task. This additional performance can be determined by calculating the fraction

$$\delta g_{08-04} = \frac{\varepsilon_{08}\tilde{\varepsilon}_{08}T - \varepsilon_{04}\tilde{\varepsilon}_{04}T}{\tilde{\varepsilon}_{08}T - \tilde{\varepsilon}_{04}T} = \frac{\varepsilon_{08}\tilde{\varepsilon}_{08} - \varepsilon_{04}\tilde{\varepsilon}_{04}}{\tilde{\varepsilon}_{08} - \tilde{\varepsilon}_{04}} = 2.0 \pm 0.5 \quad (8.17)$$

where for easier readability, the set of true graph edges  $\varepsilon$  indicates the *class. true graph eff.*,  $\tilde{\varepsilon}$  denotes the *true graph eff.* and  $T$  denotes the total number of true graph edges resulting from the true graph estimation. The fraction  $\delta g$  measures the ratio of additional true edges added and predicted by the *GNN-pipeline* over the additional true edges included by graph-building. Therefore, a significant value greater than one indicates that a graph-building model not only includes more edges but is also capable of improving its performance on the subset included in the compared graph-building model. This formula can only be employed and understood in this manner if one graph-building model produces a graph that is a subset of the other. Model 08's graph is a superset of model 04's graph as model 08 adds the same-layer connections. These edges seem to capture important local information. Through the IN, this information is passed on to neighboring edges, which in turn improves the prediction accuracy on these.

### 8.3.3. Beam Background Clean-Up and Node Classification

The final goal of the edge classification is to clean up events in the CDC from their background by removing hits that cannot be associated with a signal particle. This step can be understood as pattern recognition because it involves identifying and connecting signal hits in the CDC that follow a specific pattern, respectively, the trajectory of the signal particle. The predicted nodes are selected by determining all edges passing the threshold determined from the training. Each node connected to an edge passing the threshold is kept. Other nodes are considered classified as false.

Figure 8.9 presents *node efficiency* and *node purity* after the selection. Model 08 exhibits the highest *node purity*, and a high *node efficiency*, indicating superior

performance in the background clean-up process. Its *node efficiency* is only surpassed by Model 10. On the other hand, Model 01 stands out as an outlier with high purity and reasonable efficiency.

The initial results of the OC condensation approach have shown promising results in discriminating background nodes from signal nodes [18]. Therefore, high *node purity* might not be the primary goal for the tracking. However, the OC must be optimized to be accommodated on an FPGA. This optimization may include parameter reduction of the network, which may influence its performance in the discrimination task. Furthermore, one major objective of the background clean-up is reducing the input size for the OC. Therefore it is important to provide a high *node purity*. In the end, finding a good trade-off between *node efficiency* and *node purity* for the *GNN-pipeline* is important.

Subsection 8.3.3 provides a visual representation of the background clean-up for an event in the Central Drift Chamber with an input graph of model 10. The picture shows the possibilities of the IN impressively. The central event display demonstrates the output of the IN network showcasing the predicted edges with their predicted values. The right side of the image shows the final cleaned-up event after applying the threshold and selecting the nodes. The remaining hits are noticeably streamlined and show a clear pattern that follows the trajectory of the signal particles. The display also shows some contamination from additional background hits.

Table 8.5.: Efficiency and Purity for the true nodes after selection. Values are presented as percentages.

model	<i>node efficiency</i> (%)	<i>node purity</i> (%)
01	$67.8 \pm 0.4$	$69.6 \pm 0.4$
02	$68.3 \pm 0.4$	$64.1 \pm 0.4$
03	$67.8 \pm 0.5$	$63.0 \pm 0.4$
04	$71.2 \pm 0.5$	$64.0 \pm 0.4$
05	$69.4 \pm 0.5$	$63.5 \pm 0.4$
06	$74.3 \pm 0.5$	$64.3 \pm 0.4$
07	$76.2 \pm 0.4$	$66.1 \pm 0.4$
08	$80.0 \pm 0.4$	$71.6 \pm 0.4$
09	$78.9 \pm 0.4$	$69.7 \pm 0.5$
10	$80.6 \pm 0.4$	$67.4 \pm 0.4$

#### 8.3.4. Conclusion on the Background Clean-Up With the Interaction Network

In this study, the performance of the IN is evaluated for edge classification and background clean-up. Additionally, the performance on the true graph edges was

## 8. Background Clean-Up in then Central Drift Chamber Using the Interaction Network

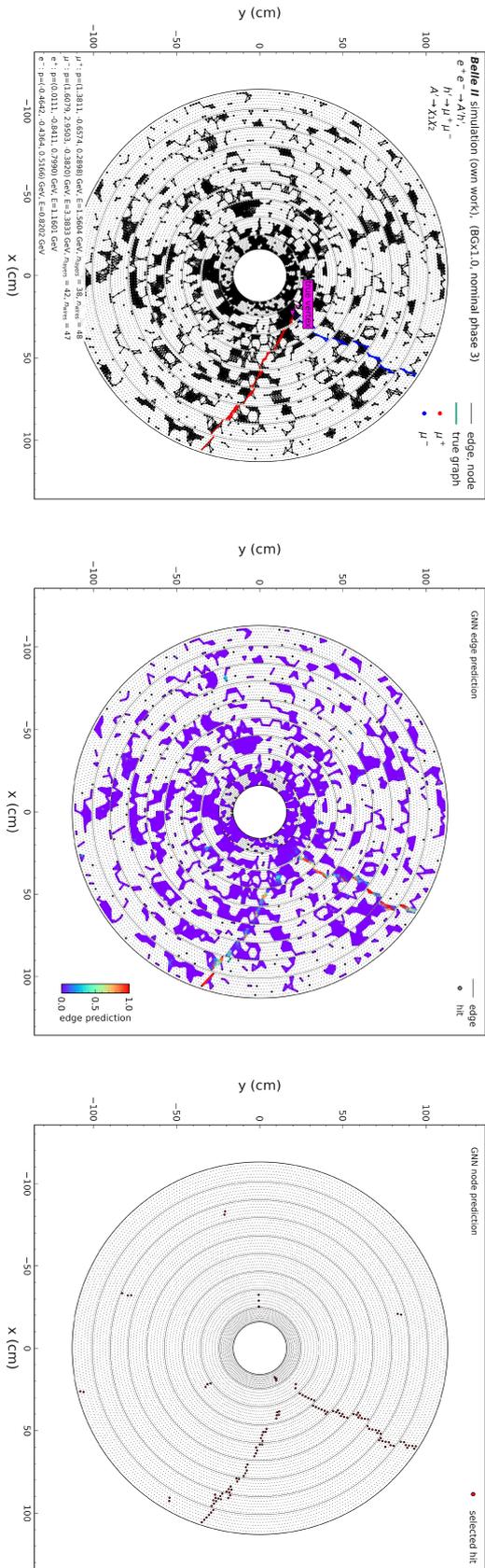


Figure 8.8: An example of the background clean-up. The event display shows all Central Drift Chamber hits connected in a graph using the graph-building method 10. The central picture shows the output of the Interaction Network network, the predicted edges. The picture on the right shows the final cleaned-up event with the remaining hits after threshold application and node selection.

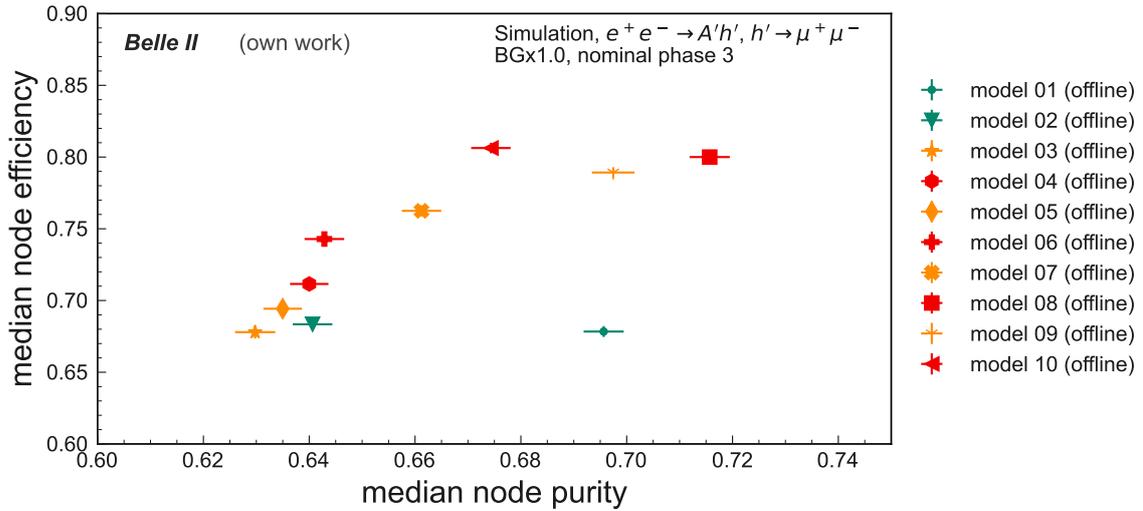


Figure 8.9.: Median *node efficiency* and *node purity* of the ten investigated models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

analyzed. The metrics are found to have interdependencies with each other. The correlation between the metrics is investigated in dependency on the ten analyzed models. The resulting correlations are presented in fig. 8.10. The following interesting relationships can be found:

**True Edges and True Graph Edges** Choosing a model with high *class. true edge eff.* is likely to positively impact the *true graph eff.*, as the incl. true graph edges are a subset of the incl. true edges in this analysis.

**True Edges and Signal Nodes** Choosing one of ten models with improved *class. true edge pur.*, positively impacts the *node purity*, as it helps dismiss background nodes connected to edges not passing the threshold.

**True Graph Edges and Signal Nodes** True graph edges appear most important for the classification task. A model with a high *total true graph eff.* will likely significantly impact the *node efficiency*. This assertion is supported by the strong correlation of  $p = 0.98$  between the two metrics. The high correlation between *class. true graph eff.* and *node efficiency* in the ten models underlines the importance of incorporating true graph edges in graph-building. This may be due to the increased information they provide for the classification task, as they are edges estimating the particle's true path. The *class. true graph pur.* negatively correlates with node efficiency, suggesting that true graph edges are essential but not the only crucial edges for high *node efficiency*. The network can benefit from additional true edges.

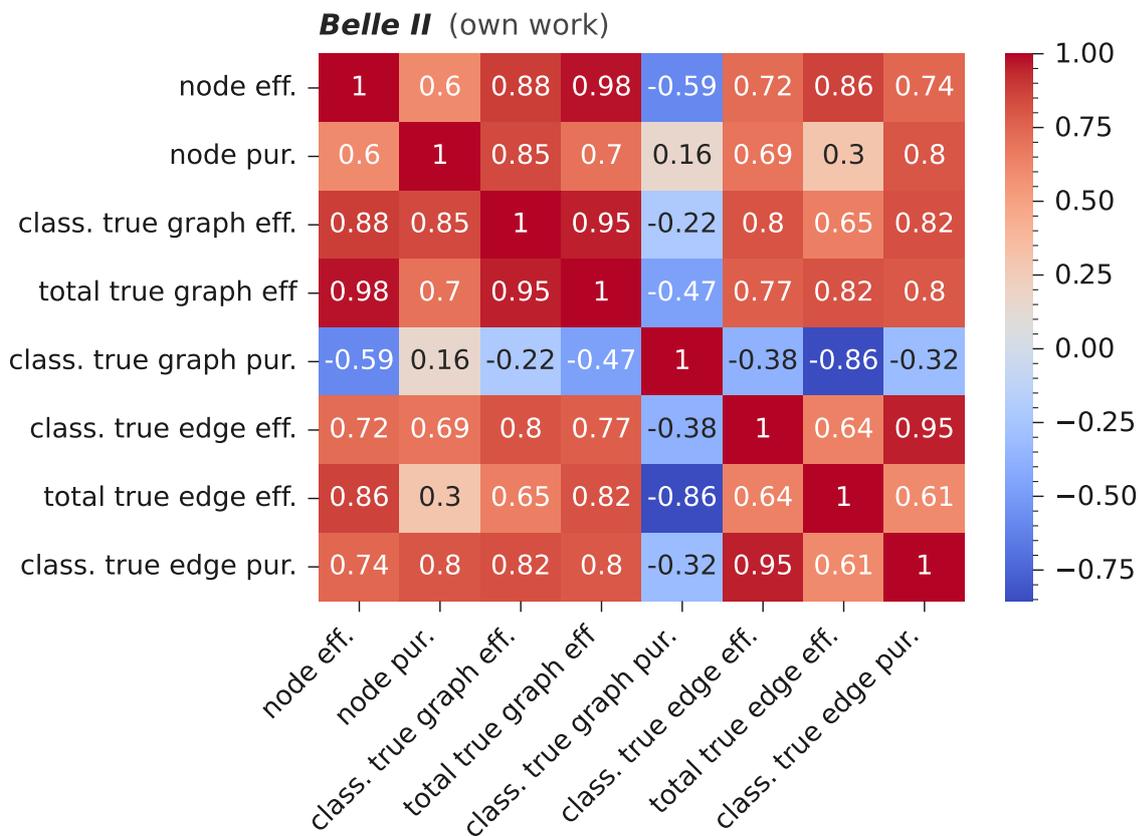


Figure 8.10.: Correlation of the metrics presented in this chapter for the *nominal Phase 3* background with the ten analyzed graph-building models.

Among the models evaluated, Model 08 demonstrated the most robust performance of the analyzed models. This suggests that the underlying architecture and training of the IN contribute to its success in the background clean-up task.

## 8.4. Discussion on Real-Time Implementation

Implementing the *GNN-pipeline* in real-time (online) on the L1 Trigger level may be a crucial step in improving the efficiency and sensitivity of particle collision analysis in future data-taking periods of the Belle II experiment. More selective data acquisition, an anticipated outcome of this implementation, could significantly enhance the experiment's sensitivity. For instance, a displaced vertex trigger for the Inelastic Dark Matter with a Dark Higgs (IDMDH) model discussed in this thesis may increase the sensitivity to this model for the Belle II experiment [12]. Thus, the implementation on L1 Trigger level may not only tackle challenges posed by the increased beam backgrounds condition but also increase the feasibility of novel particle discoveries or physics discoveries.

This section focuses on the potential real-time, or online, implementation of this system. Here, "real-time" emphasizes the necessity for immediate processing and response, allowing the system to react instantaneously to changing conditions in the detector, while "online" refers to continuous, uninterrupted processing. These terms underscore the need for a trade-off between increased input sizes incorporating additional information on the physics event for the IN and the computational overhead.

It is crucial to understand the comparability of this thesis analysis with an online implementation, where specialized hardware like FPGAs are used, and to recognize the limitations when comparing the results to the potential performance on an FPGA. The following points highlight some of these limitations:

**Precision** The PyG implementation employs high-precision (32 Bit) floating-point operations. However, for the implementation on an FPGA, lower precision arithmetic (e.g., fixed-point or half-precision floating point) may be used to reduce the utilization on the FPGA. This change in precision may influence the network's accuracy.

**Resource consideration** To meet latency and throughput needs, space multiplexing is used instead of time multiplexing, which is common in software implementations on traditional hardware architectures, such as Central Processing Units (CPUs). This choice increases resource usage on the FPGA, directly affected by the size of the IN and the size of the input graphs. Therefore, it is important to consider FPGA resources and hardware requirements for the implementation of the *GNN-pipeline*.

**Memory limitations** Memory available on an FPGA is usually limited to a few megabytes, and relying on external memory does introduce significantly increased latency. Thus, optimizing the size of input graphs and neural network models for the FPGAs to maintain low latency levels is crucial.

**Clock frequency and parallelization** As the FPGA clock frequency is typically lower than that of GPUs and CPUs, the latency is highly dependent on efficient pipelining and parallelization on the FPGA. With regard to the IN, pipelining the graphs through the FPGA may be especially problematic due to layer skipping, where information must be retained or forwarded to a register.

Although implementing the IN in PyG can offer valuable insights when comparing the classification performance of the input graphs, the comparison does not account for the computational overhead of different methods, a critical factor for the implementation. The hardware must be efficient enough to build and process the graphs in a sub-microsecond timescale. Additionally, the results may be impacted by the reduced available resolution in the online application.

To address these challenges, the following two sections discuss a potential method to estimate the computational complexity of the IN and present a first analysis on the reduced input features.

### 8.4.1. Floating Point Optimization

To better understand the computational complexity of the IN and the potential performance on an FPGA, the number of Floating Point Operations (FLOPs) used from the IN is compared for the ten different graph-building models. FLOPs is a common metric for quantifying the computational workload of an algorithm. However, it is essential to recognize that the FLOP count derived from the profiling tool may not directly correspond to the actual performance on an FPGA. Due to the hardware differences, variations in how efficiently the operations are executed may affect the overall computational workload. Furthermore, the implementation on the hardware level may vary from the implementation in PyG, and therefore the FLOPs may not be directly comparable. Furthermore, the Pytorch profiler used for the FLOP count analysis accounts only for additions combined with matrix multiplications, like in the fully connected layers. The scatter addition, performed during the aggregation step, is not accounted for. Despite these limitations, the FLOP count can still provide valuable insights into the computational complexity of the IN and serve as an approximate comparison metric between the different graph-building models. Table 8.6 shows the

Table 8.6.: Median estimated Floating Point Operations of the Interaction Network for models 01 to 10. FLOPs are estimated using the PyTorch profiler.

model	total FLOPs (MFLOP)
01	$20.23 \pm 0.05$
02	$33.94 \pm 0.08$
03	$24.79 \pm 0.06$
04	$34.03 \pm 0.08$
05	$38.51 \pm 0.09$
06	$47.53 \pm 0.12$
07	$37.11 \pm 0.08$
08	$46.15 \pm 0.10$
09	$50.53 \pm 0.11$
10	$59.97 \pm 0.14$

estimated flops for each model. It has been shown that model 10 does present a slightly better *node efficiency* and *node purity* than model 08. From the table, it is directly visible that this slight increase comes with a vastly increased computational complexity. Given these considerations, model 08 from the ten graph-building models studied offers a promising balance between performance, as measured in *node efficiency* and *node purity*, and computational complexity, represented by the FLOPs.

### 8.4.2. Performance Assessment of Edge Classification With Downscaled Resolution Inputs

A fundamental consideration in the development and real-time implementation of the graph-based machine learning models for the Belle II experiment is not only the computational constraints of the FPGAs utilized but also the reduced available resolution of the input features. Details on the expected information and resolutions, available in future data-taking periods of the Belle II experiment, can be found in subsection 2.2.2.

This section presents an initial investigation to assess the edge classification performance with the same IDMDH simulation, using the same *nominal Phase 3* background overlays, but with a post-processing downscaled resolution of input features to address the anticipated resolution available after Long Shutdown 2 (LS2). The reduced resolutions concerned are:

- (i) Reduced sense wire information. Only the information from a fraction of the sense wires in the CDC is provided to the L1 Trigger.
- (ii) Reduced resolution of the TDC information and an unknown event time  $T_{\text{evt}}$ . Therefore the absolute values of the TDC count lose their meaning in an event. The resolution is anticipated to be reduced from 1 ns to 2 ns.
- (iii) Reduced resolution of the ADC count and a reduced sampling rate of the ADCs count. The latter may yield differing values from the highly sampled information passed to the Data Acquisition (DAQ) system. The anticipated resolution is to be within a few bits.

It's worth noting that the anticipated resolutions and information availability after LS2 detailed above are preliminary and non-official. They are subject to change based on decisions related to hardware use, configuration, and other system design aspects for the following period.

In an effort to adjust to these constraints, several changes were made to the data processing techniques. Instead of using all available wire information, only trigger wires were utilized. The resolution of ADC counts was utilized with three flags in the range of  $[0, 30]$  ADC counts,  $[30, 150]$  ADC counts and  $[150, 600]$  ADC counts to emulate the expected drop in resolution. The TDC resolution was reduced by binning the TDC counts in bins of two TDC counts, referring to the halved resolution.

Finally, the original input features were adjusted to contain the low resolution ADC information and the spatial information  $x, y$ . To address the unknown absolute values of the TDC, it was not employed as a node feature but utilized as an additional edge attribute  $\Delta\text{TDC}$ . This approach allows using the potentially informative relative differences between the TDC values. The metrics for the evaluation are generally adopted for the graph-building from section 7.3 and for the edge classification task

section 8.2. The underlying sets of true graph edges and true edges were matched to the available sense wires at the L1 Triggers level.

A slightly adjusted model of the IN was utilized as shown in table 8.7 to accommodate the adjusted input feature dimensions. The models were trained with a batch size of 1024.

Table 8.7.: Summary of the adapted Interaction Network implemented in PyTorch Geometric for reduced resolution inputs. The model contains 1.947 trainable parameters.

Block	Layer Type	number of parameters
Relational Model 1		
	Linear	220
	ReLU	
	Linear	420
	ReLU	
	Linear	63
Object Model 1		
	Linear	140
	ReLU	
	Linear	420
	ReLU	
	Linear	63
Relational Model 2		
	Linear	200
	ReLU	
	Linear	420
	ReLU	
	Linear	21

Table 8.8 shows the classification metrics for the true edges and table 8.9 presents the *total true graph eff.*, additional metrics are listed in appendix C.2. Comparing the model’s performance before and after the reduction in resolution indicates a decrease in efficiency and purity across the board. For instance, the *class. true edge eff.* decreased from values ranging between 58.6 % to 68.9 % to a range of 47.2 % to 57.6 %. Similarly, *class. true edge eff.*, *class. true edge pur.*, and *class. true graph pur.* all experienced similar drops. Model 08 again achieves the highest efficiency. It is important to note that these results still hold considerable promise for online applications as the model maintains the capability to learn and predict edges accurately. This capability is particularly encouraging, given the high beam background and the fact that the model has not been optimized regarding parameter space and optimal training. Furthermore, achieving higher efficiency at the expense of a drop in purity is possible and, therefore, a suitable configuration for the whole tracking *GNN-pipeline*.

It has been demonstrated that the overall *total true graph eff.* plays a crucial role in the background clean-up. The graph-building model heavily influences this efficiency. Thus, graph-building plays a key role in achieving high *node efficiency* in background clean-up.

As the same graph-building models were employed, similar efficiency in the graph-building is expected on the considered fraction of the CDC. Therefore, a cautious forecast of the background clean-up implies similar drops in the *node efficiency* and *node purity* as for the other investigated metrics.

In essence, the challenges of real-time implementation of IN on FPGA, such as hardware limitations and reduced resolution of input features, are indeed significant obstacles. However, with the use of appropriate graph-building models and the application of the IN, effective background clean-up may be achievable, and sufficient resolution may be achieved in the subsequent particle tracking task of the *GNN-pipeline*. This underscores the potential of these methods to significantly increase the experiment's sensitivity, paving the way for new particle or physics discoveries.

Table 8.8.: The *class. true edge eff.* and the *class. true edge pur.* of the ten Interaction Networks with *nominal Phase 3* background and the reduced resolution inputs with Level 1 Trigger information.

Model	<i>class. true edge eff.</i> (%)	<i>class. true edge pur.</i> (%)
01	53.7 ± 0.4	62.1 ± 0.5
02	47.2 ± 0.5	49.3 ± 0.4
03	57.1 ± 0.5	66.2 ± 0.5
04	54.5 ± 0.5	63.5 ± 0.5
05	54.0 ± 0.5	63.9 ± 0.5
06	50.6 ± 0.5	61.8 ± 0.5
07	57.0 ± 0.5	66.7 ± 0.5
08	57.6 ± 0.5	65.3 ± 0.5
09	54.8 ± 0.5	65.9 ± 0.5
10	55.6 ± 0.5	65.4 ± 0.5

Table 8.9.: The *true graph eff.*, and the *total true graph eff.* of the ten Interaction Networks with *nominal Phase 3* background and the reduced resolution inputs with Level 1 Trigger information.

Model	<i>true graph eff.</i> (%)	<i>total true graph eff.</i> (%)
01	86.36 $\pm$ 0.10	46.9 $\pm$ 0.4
02	88.00 $\pm$ 0.08	45.2 $\pm$ 0.5
03	90.24 $\pm$ 0.11	48.6 $\pm$ 0.5
04	93.15 $\pm$ 0.11	49.9 $\pm$ 0.5
05	91.67 $\pm$ 0.08	48.9 $\pm$ 0.5
06	94.44 $\pm$ 0.08	48.6 $\pm$ 0.5
07	94.20 $\pm$ 0.05	52.9 $\pm$ 0.5
08	97.14 $\pm$ 0.05	55.7 $\pm$ 0.5
09	96.296 $\pm$ 0.024	53.7 $\pm$ 0.5
10	98.630 $\pm$ 0.018	56.0 $\pm$ 0.5

## 9. Conclusion and Outlook

The aim of the present research was to determine the feasibility of a *GNN-pipeline* for particle tracking in the main tracking detector, the Central Drift Chamber (CDC), of the Belle II experiment. For this, the thesis set out three primary goals.

- (i) Examine the discriminatory power of detector outputs when employed as input features for a neural network. Focusing on a first comparison of differences between simulation and real detector data, as this can have a significant impact on the performance of the *GNN-pipeline*.
- (ii) Explore the influence of different graph-building methods, gaining a better understanding of important connections in the graphs.
- (iii) Assess the feasibility of a background clean-up by classifying edges of the graphs, resulting from the graph-building with the Interaction Network (IN), to find patterns of the signal particles and discriminate them against the background.

This study has identified that the Analog-to-Digital Converter (ADC) count, which refers to the deposited energy in a cell of the CDC, and the Time-to-Digital Converter (TDC) count, referring to associated timing information, offer orthogonal discrimination power for signal-to-background separation. This makes them valuable features for pattern recognition in the CDC, additional to spatial information. This information is crucial for the Belle II experiment's future data-taking periods. An increase in beam background is expected, which will pose a challenge to hit discrimination and affect the performance of the current pattern recognition systems.

This thesis examined the application of graph-building in the context of a Graph Neural Network (GNN)-based approach for track finding within the CDC. Graph-building models were analyzed using different patterns for possible connections to neighboring sense wires in the CDC. The analysis used simulated events of a Inelastic Dark Matter with a Dark Higgs (IDMDH) physics model, which encompasses displaced vertices in the CDC. It is necessary to create graphs for the *GNN-pipeline* that effectively capture crucial information about particle tracks to reconstruct tracks and determine track parameters accurately, while it is beneficial to exclude extraneous edges. This thesis elaborates on two sets of essential metrics to assess this task by defining a subset of edges with relational information between signal hits in the CDC.

- (i) The complete set of true edges represents an extensive set of edges.

- (ii) The true graph represents a minimal set of true edges by estimating the path of a particle.

Subsequently, the graph-building models were used as input for analyzing the IN, using ADC and TDC as additional input features to classify edges. The thesis emphasizes that a high *true graph eff.* in the graph building and a high *class. true graph eff.* in the edge classification ensure that the essential edges are present, connecting signal hits of a particle. The combined efficiency metric *total true graph eff.* highlights this significance with a strong correlation to the *node efficiency*, representing the fraction of remaining signal hits after the background clean-up.

Additional true edges to the minimal set can offer supplemental information to a certain extent but generally, a trade-off between purity and efficiency must be considered in graph-building. To aid this decision, the fraction  $\delta g$  is introduced, which measures the ratio of additional true graph edges added and predicted by the *GNN-pipeline* over the additional true graph edges included by graph-building. This metric offers an assessment of the classification task's benefit of additional true graph edges when one graph-building model produces a graph that is a subset of the other.

Choosing the optimal graph-building model for a physical model relies on the specific analysis being conducted. In this thesis, the *GNN-pipeline* was assessed using a IDMDH simulation. Model 08 and model 10 emerged as the most promising, achieving the highest *node efficiency*. Both models permit connections within the same layer of the CDC and use connections within a layer distance of  $\Delta l = 2$ . Model 10 uses the same connections as model 08 but introduces additional connections in the layer distance of  $\Delta l = 2$ . This introduces significantly more edges to the overall *GNN-pipeline* while leading to slightly higher *node efficiency*.

Background removal employing the IN showed promising results. The IN successfully learned the discrimination between true and false edges. The *node purity* outcomes ranged from  $(63.0 \pm 0.4) \%$  to  $(71.6 \pm 0.4) \%$  across the ten analyzed models. The *node efficiency* could reach a value of up to  $(80.6 \pm 0.4) \%$  with the full expected background of Belle II.

This thesis also presented an initial analysis towards real-time implementation. For this, the input to the graph-building and the edge classification was adjusted to align the anticipated resolutions on Level 1 Trigger (L1 Trigger) level in the future data-taking periods of Belle II. It finds that the increase in computational complexity can be estimated by the number of Floating Point Operations (FLOPs), which is particularly relevant for the real-time application as this faces several computational constraints and aids the discussion for the implementation on Field-Programmable Gate Arrays (FPGAs). The classification showed reduced efficiencies and purities. For instance, the *class. true edge eff.* decreased from values ranging between 58.6% to 68.9% to a range of 47.2% to 57.6%. However, the networks could still learn and predict edges, presenting promising results for online implementation.

---

In summary, the results obtained in this thesis are encouraging, indicating that the *GNN-pipeline* can be effectively applied to the Belle II experiment.

Several limitations to this study need to be acknowledged.

The input feature analysis was designed with the primary goal of a broad, albeit fast, comparison. A limited number of runs with a small dataset of the Belle II experiment were analyzed. Thus, further investigations could benefit from additional data for a more comprehensive understanding. Additionally, no corrections were applied to the reconstructed signal, which introduce additional uncertainty in the selection.

The analysis of the graph-building models was constrained by a narrow scope of ten models, and the metrics were only evaluated on these models. A further limitation lies in the specificity of the simulations used. The graph-building, the edge classification, and the background clean-up processes were only analyzed with a IDMDH, thus potentially limiting the generalizability of these findings.

Lastly, the online implementation of the *GNN-pipeline* on an FPGA introduces several challenges and may make an adjustment of resolution necessary. The results presented in this study are not directly applicable to a real-time implementation but serve as a basis for in-depth studies.

In the future, several aspects of the current work can be explored and expanded upon. The input feature analysis can be improved by applying corrections to the signal tracks and refining the run selection with runs with increased luminosity. This may help understand the details of further modeling optimization to map the simulation to the real detector signal. This will be crucial for optimized training of the *GNN-pipeline*. More broadly, research is also needed to determine if simulations accommodate the expected increased beam background, as this will be the primary application of the *GNN-pipeline*. The findings of this study and related studies have encouraged the Belle II collaboration to make efforts to make ADC and TDC information available on the trigger level for future real-time applications.

A reduced resolution of the ADC and TDC values are expected for the implementation on L1 Trigger. Future studies could access the optimal utilization of this reduced resolution. This includes finding optimal flags for the ADC count. This would help maximize the performance of the *GNN-pipeline* under these constraints and ensure the best results in real-world applications.

A first implementation of the graph-building on an FPGA is ongoing. A detailed analysis of the subsequent performance of the IN on the hardware will give more detailed insights into optimization requirements regarding the input size of the networks. The IN model used for this analysis has a rather small number of parameters. Future studies should therefore examine the hardware's limits in terms of larger parameter space of the IN and investigate its benefit on the classification efficiencies.

## 9. Conclusion and Outlook

---

A key aspect will also be to evaluate the effect of *node efficiency* and *node purity* on the accuracy of the track parameter estimation in subsequent *GNN-pipeline* steps to understand the background clean-up requirements in more detail.

# Acronyms

**stt** Single Track Trigger. 16

**2D** Two-Dimensional. 2, 11, 19, 42, 46, 53

**3D** Three-Dimensional. 2, 4, 7, 19

**Adam** Adaptive Moment Estimation. 81

**ADC** Analog-to-Digital Converter. iii, 7–9, 26, 31, 38–40, 42, 46, 49–51, 53, 79, 99, 103–105, 121, 122, 126, 127, 131, 132, 136, 137, 141, 142, 146, 147, 151, 152

**ARICH** Aerogel Ring Imaging Cherenkov. 5

**basf2** Belle II Analysis Software Framework. 11–13, 15, 16, 19, 25–28, 33, 56

**BSM** Beyond the Standard Model. 1, 13, 15

**CDC** Central Drift Chamber. iii, 2, 4–11, 19–22, 26, 30, 31, 33, 38–40, 42, 45, 46, 49, 51, 53–58, 60, 61, 64–66, 73, 77, 79, 88, 92, 99, 101, 103, 104, 121, 122, 125, 127, 130, 132, 135, 137, 140, 142, 145, 147, 150, 152

**CMB** Cosmic Microwave Background. 13, 14

**CP** Charge Conjugation Parity Symmetry. 3

**CPU** Central Processing Unit. 10, 97

**DAQ** Data Acquisition. 7–10, 99

**DM** Dark Matter. 13, 14, 28

**ECL** Electromagnetic Calorimeter. 5, 10, 32, 33

**FLOP** Floating Point Operation. 98, 104

**FPGA** Field-Programmable Gate Array. 10, 31, 78, 93, 97–99, 101, 104, 105

**GDL** Global Decision Logic. 10

**GeoDL** Geometrical Deep Learning. 1, 21, 22

- GNN** Graph Neural Network. iii, 1, 2, 19–23, 28, 30, 49, 53, 54, 58, 59, 63, 64, 66, 67, 77, 78, 82, 84–86, 88, 92, 93, 96, 97, 100, 101, 103–106
- GPU** Graphics Processing Unit. 23, 81, 97
- HEP** High Energy Physics. 11, 19, 53
- HER** High-Energy Ring. 3, 29
- HLT** High Level Trigger. 10
- IDMDH** Inelastic Dark Matter with a Dark Higgs. i, iii, 14, 15, 25, 28, 96, 99, 103–105, 165, 173
- IN** Interaction Network. iii, 2, 19–23, 25, 54, 59, 63, 66, 73, 77–88, 91–93, 96–98, 100, 101, 103–105, 165, 168, 169, 171, 173–175
- IP** Interaction Point. 1, 3, 4, 6, 9–11, 13, 15, 16, 20, 25, 30, 42, 49, 53, 64, 73, 88
- KEK** High Energy Accelerator Research Organization. 3
- KLM**  $K_L$  Muon Detector. 5, 10
- L1 Trigger** Level 1 Trigger. iii, 10, 33, 50, 78, 96, 99, 100, 104, 105, 174–176
- LER** Low-Energy Ring. 3, 29
- LINAC** Linear Particle Accelerator. 3
- LS1** Long Shutdown 1. 50
- LS2** Long Shutdown 2. 8, 99
- MC** Monte Carlo. 2, 9, 25, 28, 30, 31, 33–35, 50, 54, 60, 61
- OC** Object Condensation. 77, 85, 93
- PID** Particle Identification. 5, 6, 33–35
- PXD** Pixel Detector. 4
- PyG** PyTorch Geometric. 23, 79, 97, 98
- ReLU** Rectified Linear Unit. 80
- SM** Standard Model of Particle Physics. 1, 3, 13–16, 20, 25
- SVD** Silicon Vertex Detector. 4

**TDC** Time-to-Digital Converter. iii, 8, 26, 31, 33, 38, 45, 46, 49–51, 53, 79, 99, 103–105, 122, 127, 132, 137, 142, 147, 152

**TOP** Time-of-Propagation. 5

**TSF** Track Segment Finder. 64

**VXD** Vertex Detector. 4, 26

**WIMP** Weakly Interacting Massive Particle. 14



## Bibliography

- [1] T. Abe et al. “Belle II Technical Design Report”. In: *KEK Report* (). Edited by: Z. Doležal and S. Uno. URL: <https://arxiv.org/pdf/1011.0352>.
- [2] S. Agostinelli et al. “Geant4—a simulation toolkit”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003). PII: S0168900203013688, pp. 250–303. ISSN: 01689002. DOI: 10.1016/S0168-9002(03)01368-8. URL: <https://www.sciencedirect.com/science/article/pii/S0168900203013688>.
- [3] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. *MadGraph 5: going beyond*. PII: 2359. 2011. DOI: 10.1007/JHEP06(2011)128. URL: <https://arxiv.org/pdf/1106.0522>.
- [4] Sabrina Amrouche et al. *The Tracking Machine Learning challenge : Throughput phase*. submitted to Computing and Software for Big Science. 2021. URL: <https://arxiv.org/pdf/2105.01160>.
- [5] Sabrina Amrouche et al. “The Tracking Machine Learning Challenge: Accuracy Phase”. In: *In: Escalera S., Herbrich R. (eds) The NeurIPS* (), pp. 231–264. DOI: 10.1007/978-3-030-29135-8\_9. URL: <https://arxiv.org/pdf/1904.06778>.
- [6] Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. *Interaction Networks for Learning about Objects, Relations and Physics*. Published in NIPS 2016. 2016. URL: <https://arxiv.org/pdf/1612.00222>.
- [7] Valerio Bertacchi et al. “Track Finding at Belle II”. In: *Computer Physics Communications* 259.3 (2021). *Comput.Phys.Commun.* 259 (2021) 107610, p. 107610. ISSN: 00104655. DOI: 10.1016/j.cpc.2020.107610. URL: <http://arxiv.org/pdf/2003.12466v2>.
- [8] Douglas Clowe, Maruša Bradač, Anthony H. Gonzalez, Maxim Markevitch, Scott W. Randall, Christine Jones, and Dennis Zaritsky. “A Direct Empirical Proof of the Existence of Dark Matter”. In: *The Astrophysical Journal* 648.2 (2006), pp. L109–L113. ISSN: 0004-637X. DOI: 10.1086/508162. URL: <https://arxiv.org/pdf/astro-ph/0608407>.
- [9] SuperB Collaboration. *SuperB: A High-Luminosity Asymmetric  $e^+e^-$  Super Flavor Factory. Conceptual Design Report*. 524 pages, presented to INFN, March 2007. Due to arXiv file size limitations, the arXiv and printed versions are not identical. A printed, bound copy of this CDR is available from INFN Publishing Services, INFN-Pisa, L.go Pontecorvo, 3, I-56127, Pisa, Italy. 2007. URL: <https://arxiv.org/pdf/0709.0451>.

- [10] Javier Duarte and Jean-Roch Vlimant. “Graph Neural Networks for Particle Tracking and Reconstruction”. In: (2022). *Artificial Intelligence for High Energy Physics*, Chapter 12, pp. 387–436 (2022) 44 pages, 20 figures. Submitted for review. To appear in “Artificial Intelligence for Particle Physics”, World Scientific Publishing, pp. 387–436. DOI: 10.1142/9789811234033\_0012. URL: <http://arxiv.org/pdf/2012.01249v2>.
- [11] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159. URL: <http://jmlr.org/papers/v12/duchi11a.html>.
- [12] Michael Duerr, Torben Ferber, Camilo Garcia-Cely, Christopher Hearty, and Kai Schmidt-Hoberg. *Long-lived dark Higgs and inelastic dark matter at Belle II*. PII: 15404. 2021. DOI: 10.1007/JHEP04(2021)146. URL: <https://arxiv.org/pdf/2012.08595>.
- [13] Michael Duerr, Torben Ferber, Christopher Hearty, Felix Kahlhoefer, Kai Schmidt-Hoberg, and Patrick Tunney. “Invisible and displaced dark matter signatures at Belle II”. In: *Journal of High Energy Physics* 2020.2 (2020). PII: 12308. DOI: 10.1007/JHEP02(2020)039. URL: <https://arxiv.org/pdf/1911.03176>.
- [14] Abdelrahman Elabd et al. “Graph Neural Networks for Charged Particle Tracking on FPGAs”. In: *Frontiers in Big Data* 5 (2022). *Front. Big Data* 5 (2022) 828666 28 pages, 17 figures, 1 table, published version, p. 673. ISSN: 2624-909X. DOI: 10.3389/fdata.2022.828666. URL: <http://arxiv.org/pdf/2112.02048v3>.
- [15] Keiko Emi et al. “Study of a measurement and the gas-gain saturation by a prototype drift chamber for the BELLE-CDC”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 379.2 (1996). PII: 0168900296007164, pp. 225–231. ISSN: 01689002. DOI: 10.1016/0168-9002(96)00716-4.
- [16] Steven Farrell et al. *Novel deep learning methods for track reconstruction*. CTD 2018 proceedings. 2018. URL: <https://arxiv.org/pdf/1810.06111>.
- [17] Rudolf Frühwirth and Are Strandlie. “Track Finding”. In: *Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors*. Ed. by Rudolf Frühwirth and Are Strandlie. Particle Acceleration and Detection. Cham: Unspecified, 2021, pp. 81–102. ISBN: 978-3-030-65770-3. DOI: 10.1007/978-3-030-65771-0\_5.
- [18] *Graph Neural Network based Track Finding in the Central Drift Chamber at Belle II*. 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP). 2023. URL: <https://indico.jlab.org/event/459/contributions/11761/> (visited on 05/03/2023).
- [19] Geoffrey Hinton and Tijmen Tieleman. *Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude*. Lecture Slides in Neural Networks for Machine Learning. 2012. URL: [https://www.cs.toronto.edu/%C2%A0tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/%C2%A0tijmen/csc321/slides/lecture_slides_lec6.pdf).
- [20] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

- 
- [21] Y. Iwasaki, ByungGu Cheon, Eunil Won, Xin Gao, L. Macchiarulo, K. Nishimura, and G. Varner. “Level 1 Trigger System for the Belle II Experiment”. In: *IEEE Transactions on Nuclear Science* 58.4 (2011), pp. 1807–1815. ISSN: 0018-9499. DOI: 10.1109/TNS.2011.2119329.
- [22] Yoshihito Iwasaki, ByungGu Cheon, Eunil Won, and Gary Varner. “Level 1 trigger system for the Belle II experiment”. In: *2010 17th IEEE-NPSS Real Time Conference. Conference record : 24-28 May, 2010, Lisbon, Portugal*. 2010 17th Real-Time Conference - IEEE-NPSS Technical Committee on Computer Applications in Nuclear and Plasma Sciences (RT 2010) (Lisbon, Portugal, May 24–28, 2010). [Piscataway, N.J.]: IEEE, 2010, pp. 1–9. ISBN: 978-1-4244-7108-9. DOI: 10.1109/RTC.2010.5750454.
- [23] S. Jadach, B. F. L. Ward, and Z. Wa s. “Coherent exclusive exponentiation for precision Monte Carlo calculations”. In: *Physical Review D* 63.11 (2001). ISSN: 2470-0010. DOI: 10.1103/PhysRevD.63.113009. URL: <https://arxiv.org/pdf/hep-ph/0006359>.
- [24] S. Jadach, B.F.L. Ward, and Z. Wa s. “The precision Monte Carlo event generator for two-fermion final states in collisions”. In: *Computer Physics Communications* 130.3 (2000). PII: S0010465500000485, pp. 260–325. ISSN: 00104655. DOI: 10.1016/S0010-4655(00)00048-5. URL: <https://arxiv.org/pdf/hep-ph/9912214>.
- [25] Xiangyang Ju et al. “Performance of a geometric deep learning pipeline for HL-LHC particle tracking”. In: *The European Physical Journal C* 81.10 (2021). PII: 9675. ISSN: 1434-6044. DOI: 10.1140/epjc/s10052-021-09675-8.
- [26] Jan Kieseler. “Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph, and image data”. In: *The European Physical Journal C* 80.9 (2020). PII: 8461. ISSN: 1434-6044. DOI: 10.1140/epjc/s10052-020-08461-2. URL: <https://arxiv.org/pdf/2002.03605>.
- [27] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 2014. URL: <https://arxiv.org/pdf/1412.6980>.
- [28] E. Kou et al. “The Belle II Physics Book”. In: *Progress of Theoretical and Experimental Physics* 2019.12 (2019). DOI: 10.1093/ptep/ptz106.
- [29] E. Kou et al. “The Belle II Physics Book”. In: *Progress of Theoretical and Experimental Physics* 2020.2 (2020). DOI: 10.1093/ptep/ptaa008.
- [30] T. Kuhr, C. Pulvermacher, M. Ritter, T. Hauth, and N. Braun. “The Belle II Core Software”. In: *Computing and Software for Big Science* 3.1 (2019). PII: 17. ISSN: 2510-2036. DOI: 10.1007/s41781-018-0017-9. URL: <https://arxiv.org/pdf/1809.04299>.
- [31] S. Kurokawa and E. Kikutani. “Overview of the KEKB accelerators”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 499.1 (2003). PII: S0168900202017710, pp. 1–7. ISSN: 01689002. DOI: 10.1016/S0168-9002(02)01771-0.
- [32] Simon Kurz. “Performance of Belle II tracking on collision data”. In: (2020). Jun.

- [33] S. Lantz et al. “Speeding up particle track reconstruction using a parallel Kalman filter algorithm”. en. In: *Journal of Instrumentation* 15.09 (2020), P09030–P09030. DOI: 10.1088/1748-0221/15/09/P09030. URL: <https://iopscience.iop.org/article/10.1088/1748-0221/15/09/P09030>.
- [34] Lea Reuter, Philipp Dorwarth, Slavomira Stefkova, and Torben Ferber. “GNN-based Track and Vertex Finding”. In: *Belle II Germany Meeting 2022*.
- [35] Z. Liptak et al. “Measurements of beam backgrounds in SuperKEKB Phase 2”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1040 (2022). PII: S0168900222005149, p. 167168. ISSN: 01689002. DOI: 10.1016/j.nima.2022.167168. URL: <https://www.sciencedirect.com/science/article/pii/S0168900222005149>.
- [36] Michael L. Waskom. “seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.
- [37] A. Natochii, S. E. Vahsen, H. Nakayama, T. Ishibashi, and S. Terui. “Improved simulation of beam backgrounds and collimation at SuperKEKB”. In: *Physical Review Accelerators and Beams* 24.8 (2021). DOI: 10.1103/PhysRevAccelBeams.24.081001. URL: <https://arxiv.org/pdf/2104.02645>.
- [38] A. Natochii et al. *Beam background expectations for Belle II at SuperKEKB*. Contribution to Snowmass 2021. 25 pages, 9 figures, 2 tables. For the update v1 -> v2, we put the corrected Snowmass topical group number AF06 -> AF05. For the update v2 -> v3, we corrected Table 2 for Setup-2 (CW ON -> OFF). 2022. URL: <http://arxiv.org/pdf/2203.05731v3>.
- [39] A. Natochii et al. *Measured and projected beam backgrounds in the Belle II experiment at the SuperKEKB collider*. 28 pages, 17 figures, 9 tables. 2023. URL: <http://arxiv.org/pdf/2302.01566v1>.
- [40] S. Neuhaus et al. “A neural network z -vertex trigger for Belle II”. In: *Journal of Physics: Conference Series* 608 (2015), p. 012052. ISSN: 1742-6588. DOI: 10.1088/1742-6596/608/1/012052. URL: <https://arxiv.org/pdf/1410.1395>.
- [41] K. Nishimura. “The time-of-propagation counter for BelleII”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 639.1 (2011). PII: S0168900210021984, pp. 177–180. ISSN: 01689002. DOI: 10.1016/j.nima.2010.09.164. URL: <https://arxiv.org/pdf/1009.0876>.
- [42] A. Paladino. “Beam background evaluation at SuperKEKB and Belle II”. In: *Journal of Instrumentation* 15.07 (2020), pp. C07023–C07023. DOI: 10.1088/1748-0221/15/07/C07023.
- [43] Shah Rukh Qasim, Jan Kieseler, Yutaro Iiyama, and Maurizio Pierini. “Learning representations of irregular particle-detector geometry with distance-weighted graph networks”. In: *The European Physical Journal C* 79.7 (2019). PII: 7113. ISSN: 1434-6044. DOI: 10.1140/epjc/s10052-019-7113-9. URL: <https://arxiv.org/pdf/1902.07987>.
- [44] Lea Reuter, Philipp Dorwarth, Slavomira Stefkova, and Torben Ferber. *GNN-based Track and Vertex Finding at Belle II*. Talk presented at 26. Deutsche

- 
- Physikerinnentagung 2022 (26th German Conference of Women in Physics), KIT Campus South, 27 Nov 2022. 2022. URL: <https://indico.scc.kit.edu/event/2604/contributions/11656/>.
- [45] V. C. Rubin, N. Thonnard, and W. K. Ford JR. “Rotational properties of 21 SC galaxies with a large range of luminosities and radii, from NGC 4605 /R = 4kpc/ to UGC 2885 /R = 122 kpc”. en. In: *The Astrophysical Journal* 238 (1980), p. 471. ISSN: 0004-637X. DOI: 10.1086/158003.
- [46] Hadar Serviansky, Nimrod Segol, Jonathan Shlomi, Kyle Cranmer, Eilam Gross, Haggai Maron, and Yaron Lipman. *Set2Graph: Learning Graphs From Sets*. 2020. URL: <https://arxiv.org/pdf/2002.08772>.
- [47] Shoichi Shimazaki, Takashi Taniguchi, Tomohisa Uchida, Masahiro Ikeno, Nanae Taniguchi, and Manobu M. Tanaka. “Front-end electronics of the Belle II drift chamber”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 735 (2014). PII: S0168900213012849, pp. 193–197. ISSN: 01689002. DOI: 10.1016/j.nima.2013.09.050.
- [48] S. Skambraks et al. “A  $Z^0$ -Vertex Trigger for Belle II”. In: *IEEE Transactions on Nuclear Science* 62.4 (2015), pp. 1732–1740. ISSN: 0018-9499. DOI: 10.1109/TNS.2015.2439617. URL: <https://arxiv.org/pdf/1406.3319>.
- [49] Sebastian Skambraks, Steffen Bähr, Jürgen Becker, Christian Kiesling, Sara McCarney, Felix Meggendorfer, Raynette van Tonder, and Kai Lukas Unger. “A 3D track finder for the Belle II CDC L1 trigger”. In: *Journal of Physics: Conference Series* 1525.1 (2020), p. 012102. ISSN: 1742-6588. DOI: 10.1088/1742-6596/1525/1/012102.
- [50] Nanae Taniguchi, Masahiro Ikeno, Yoshihito Iwasaki, Masatoshi Saito, Shoichi Shimazaki, Manobu Tanaka, Tomohisa Uchida, and Shoji Uno. “All-in-one readout electronics for the Belle-II Central Drift Chamber”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 732 (2013). PII: S0168900213009509, pp. 540–542. ISSN: 01689002. DOI: 10.1016/j.nima.2013.06.096. URL: <https://www.sciencedirect.com/science/article/pii/S0168900213009509>.
- [51] Mark Thomson. *Modern particle physics*. 7th printing. Thomson, Mark (Autor.) Cambridge: Cambridge University Press, 2021. XVI, 554 strony. ISBN: 978-1-107-03426-6.
- [52] Torben Ferber. “Electromagnetic calorimeter reconstruction in Belle II”. In: *19th International Workshop on Advanced Computing and Analysis Techniques in Physics Research*.
- [53] R. L. Workman et al. “Review of Particle Physics”. In: *PTEP* 2022 (2022), p. 083C01. DOI: 10.1093/ptep/ptac097.
- [54] Y. Yusa. “ARICH for Belle II”. In: *Journal of Instrumentation* 9.10 (2014), pp. C10015–C10015. DOI: 10.1088/1748-0221/9/10/C10015.



# A. Appendix: Input Feature Analysis

This appendix presents additional plots of the input feature analysis, a run-dependent analysis of the data, and additional runs for the input feature analysis.

## A.1. Input Feature Analysis

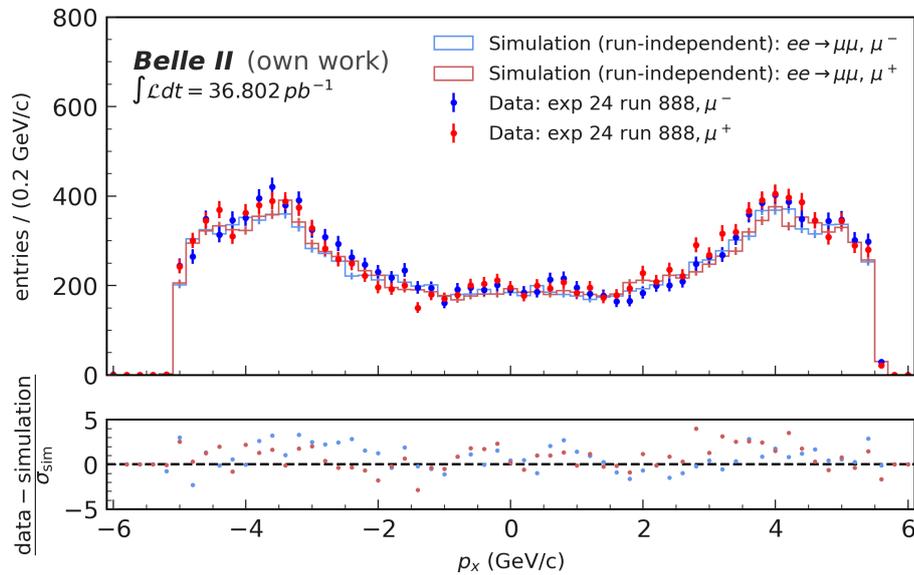


Figure A.1.: Distribution of the momentum  $p_x$  of reconstructed tracks for experiment 24, run 888, and run-independent Monte Carlo simulation after selection.

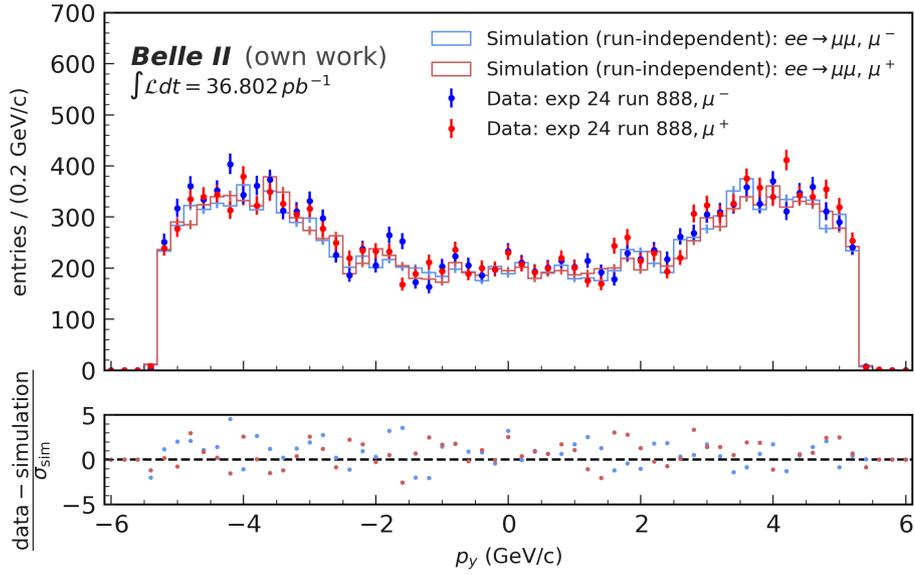


Figure A.2.: Distribution of the momentum  $p_y$  of reconstructed tracks for experiment 24, run 888, and run-independent Monte Carlo simulation after selection.

## A.2. Run-Dependent Analysis

This section presents figures corresponding to Experiment 24 with Run 888, with an MC simulation using run-dependent background files.

Table A.1.: Selection efficiency comparison for the skimmed data sample of experiment 24, run 888, and run-dependent simulation. The data consists of  $36.802 \text{ pb}^{-1}$ , and the simulation consists of 14 778 events.

Order No	Selection	Exp. 24, run 888 (%)	MC simulation (%)
1	Theta cut	$76.40^{+0.27}_{-0.27}$	$75.8^{+0.5}_{-0.5}$
2	L1 selection $stt$	$94.91^{+0.16}_{-0.16}$	$98.28^{+0.17}_{-0.18}$
3	Dimuon track selection	$87.80^{+0.24}_{-0.24}$	$94.0^{+0.3}_{-0.3}$
4	Dimuon mass cut	$94.69^{+0.18}_{-0.18}$	$99.87^{+0.4}_{-0.6}$
5	PID cut	$94.37^{+0.19}_{-0.19}$	$95.10^{+0.29}_{-0.30}$
	Total efficiency	$56.9^{+0.3}_{-0.3}$	$66.5^{+0.5}_{-0.5}$

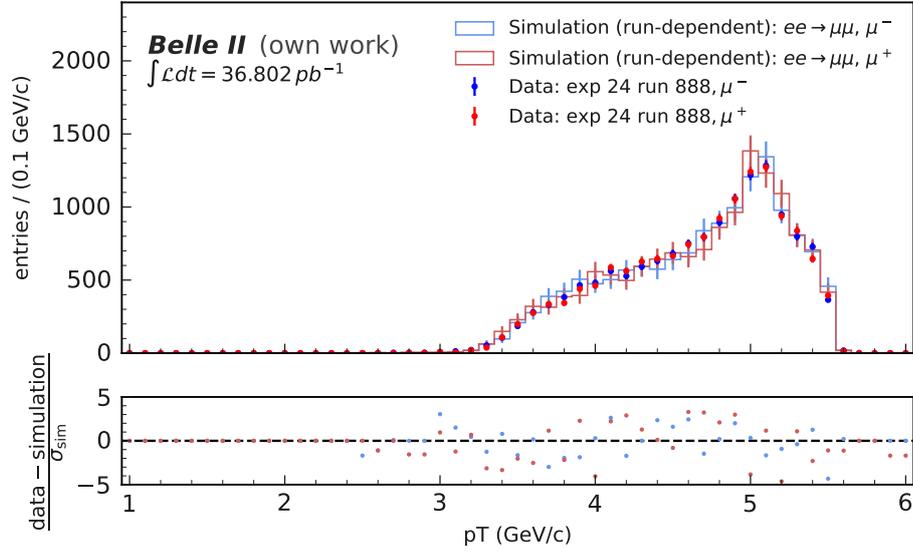


Figure A.3.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 24, run 888, and run-dependent Monte Carlo simulation after selection.

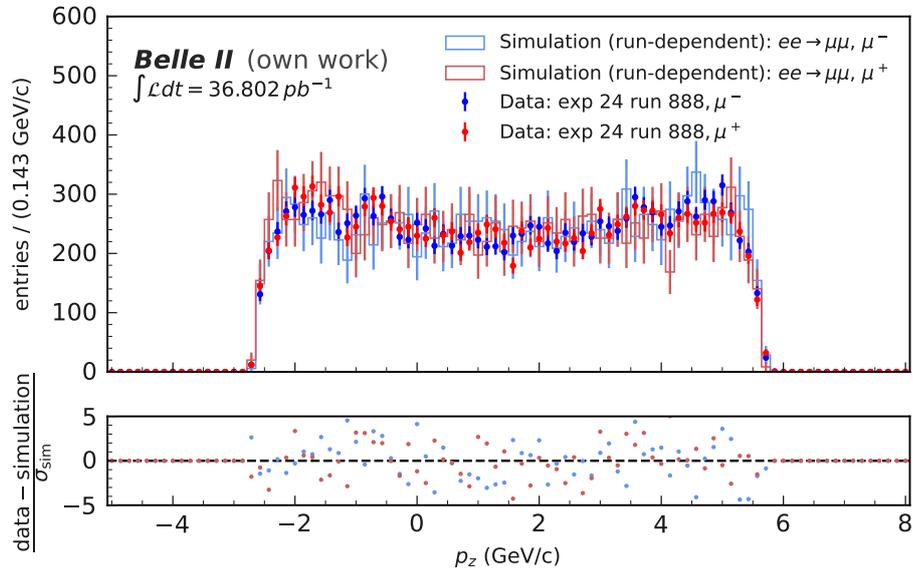


Figure A.4.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 24, run 888, and run-dependent Monte Carlo simulation after selection.

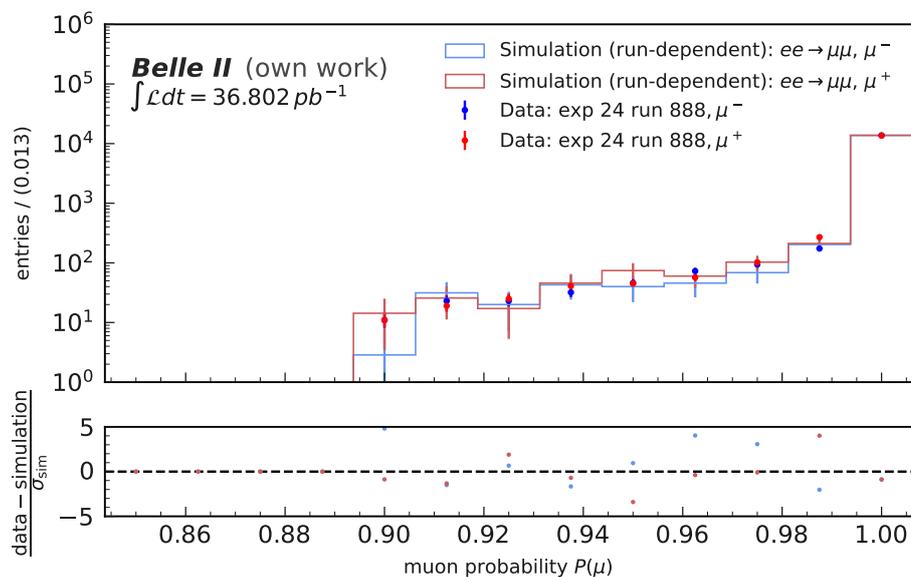


Figure A.5.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 24, run 888, and run-dependent Monte Carlo simulation after selection.

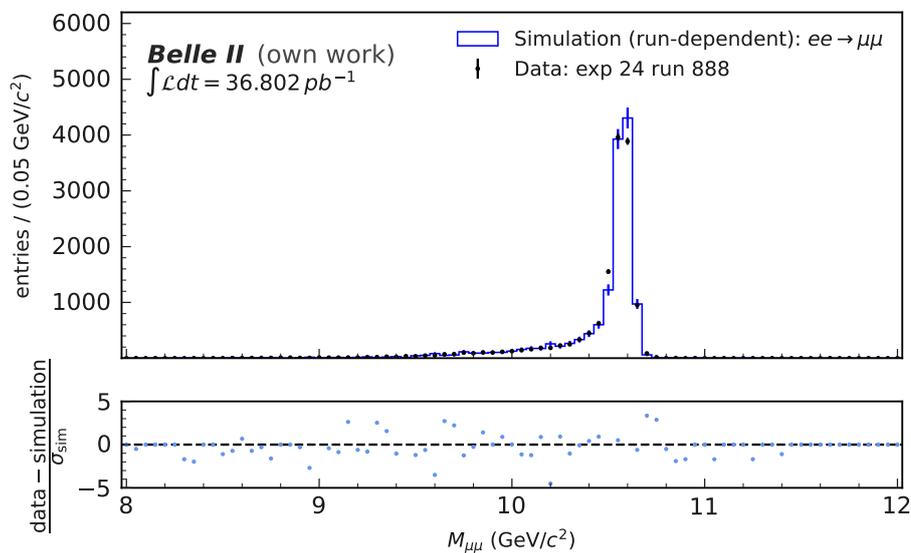


Figure A.6.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 24, run 888, and run-dependent simulation.

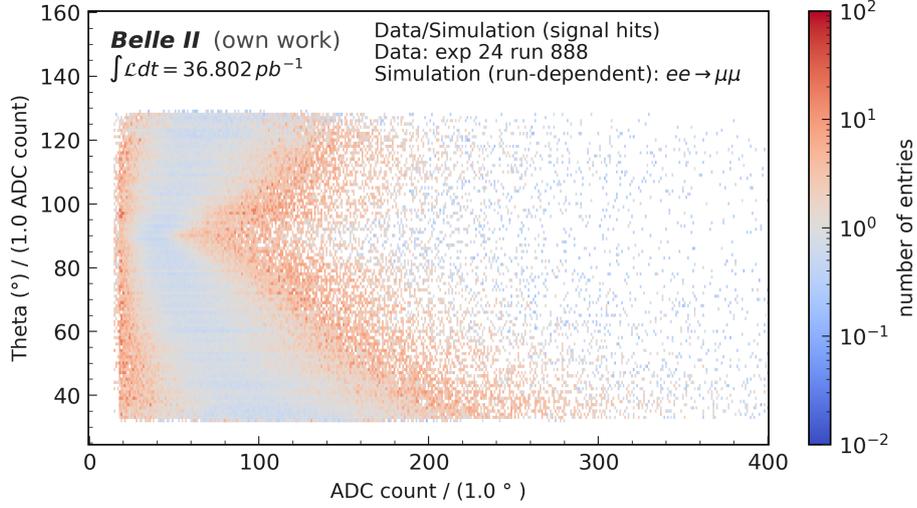


Figure A.7.: This plot demonstrates the ratio of the experimental (experiment 24, run 888) and run-dependent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

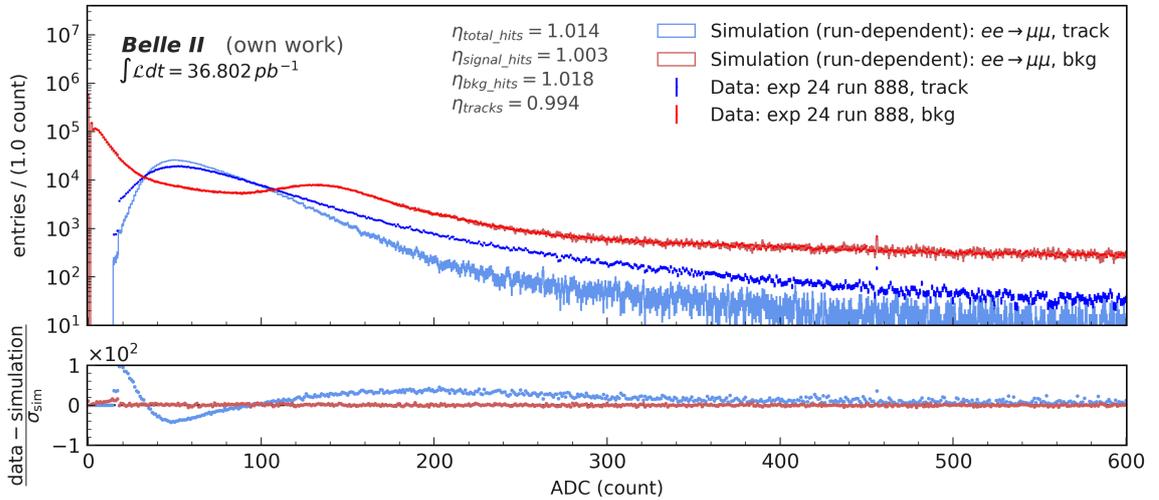


Figure A.8.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 24, run 888, and run-dependent Monte Carlo simulation.

## A. Appendix: Input Feature Analysis

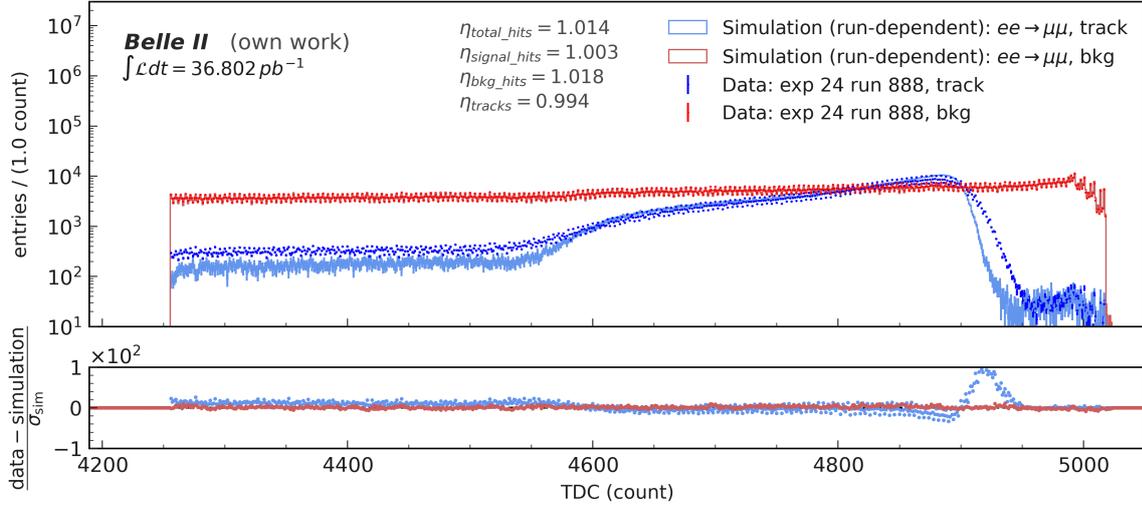


Figure A.9.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 24, run 888, and run-dependent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

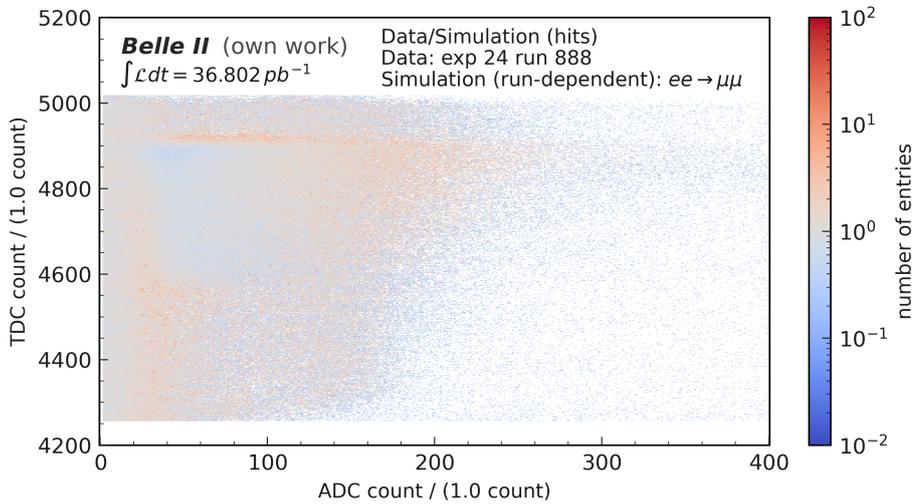


Figure A.10.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 24, run 888, and run-dependent Monte Carlo simulation.

### A.3. Experiment 20, Run 672

This section presents figures corresponding to Experiment 20 with Run 672.

Table A.2.: Selection efficiency comparison for the skimmed data sample of experiment 20, run 672, and run-independent simulation. The data consists of  $47.573 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 20, run 672 (%)	MC simulation (%)
1	Theta cut	$74.79^{+0.27}_{-0.27}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection <i>stt</i>	$94.12^{+0.17}_{-0.17}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$87.14^{+0.25}_{-0.25}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$94.76^{+0.18}_{-0.18}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.05^{+0.19}_{-0.20}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$54.7^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

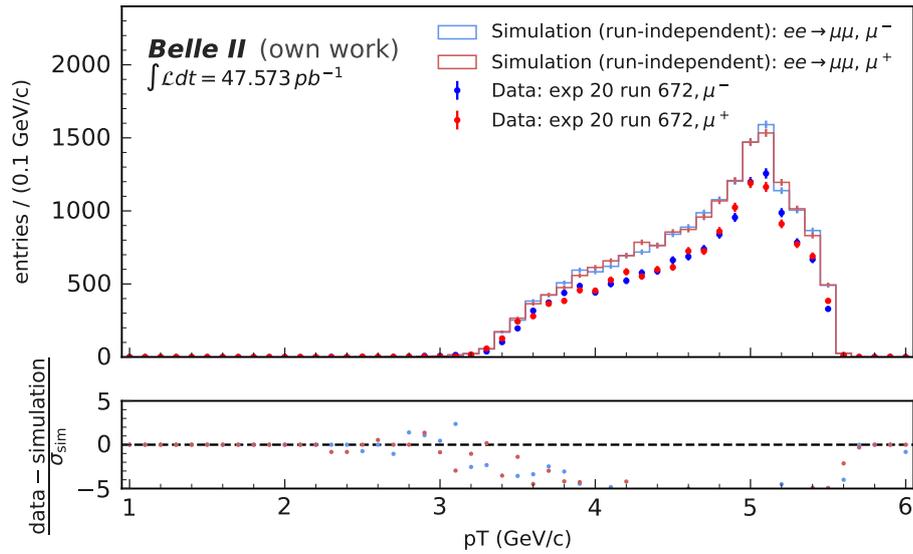


Figure A.11.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 20, run 672, and run-independent Monte Carlo simulation after selection.

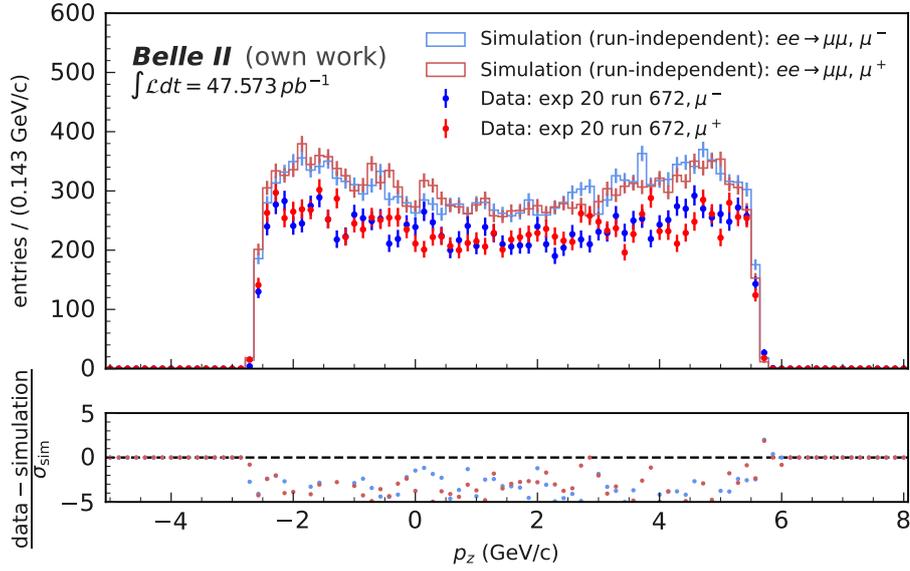


Figure A.12.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 20, run 672, and run-independent Monte Carlo simulation after selection.

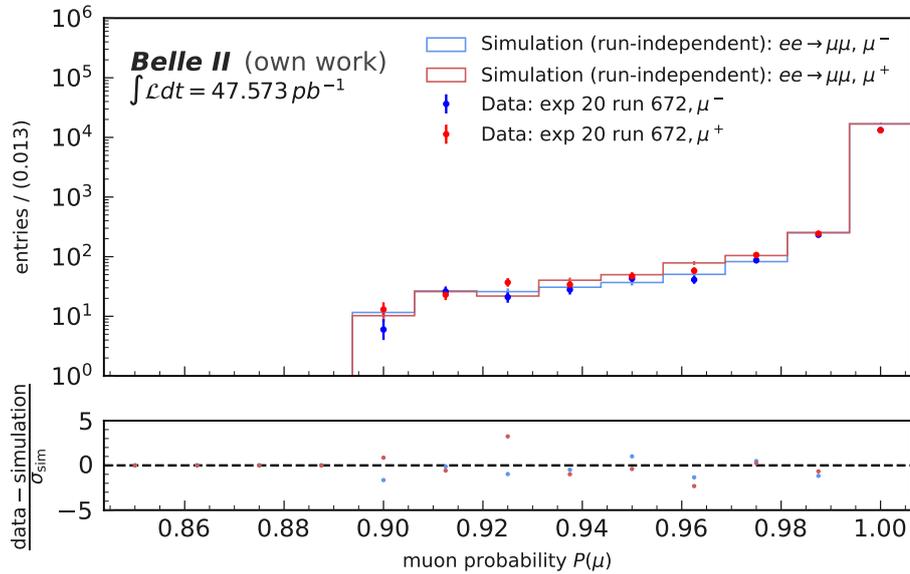


Figure A.13.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 20, run 672, and run-independent Monte Carlo simulation after selection.

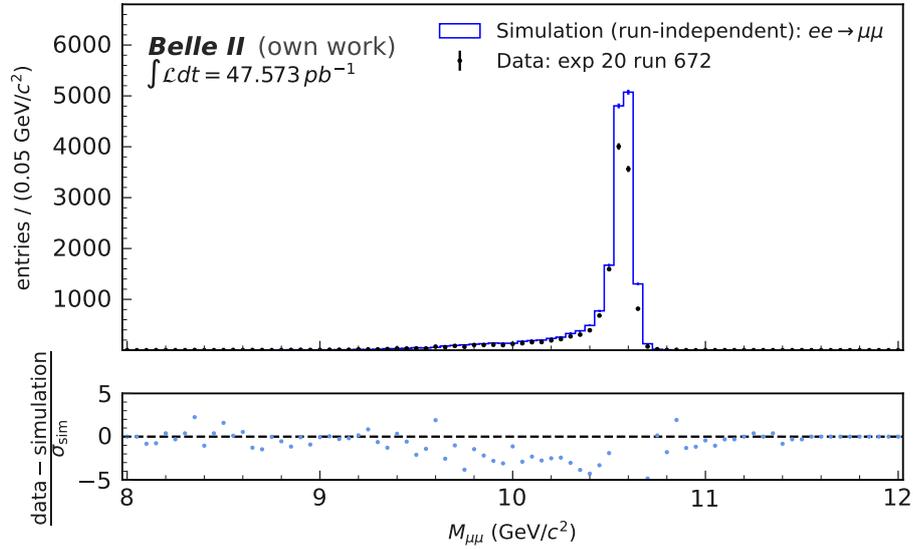


Figure A.14.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 20, run 672, and run-independent simulation.

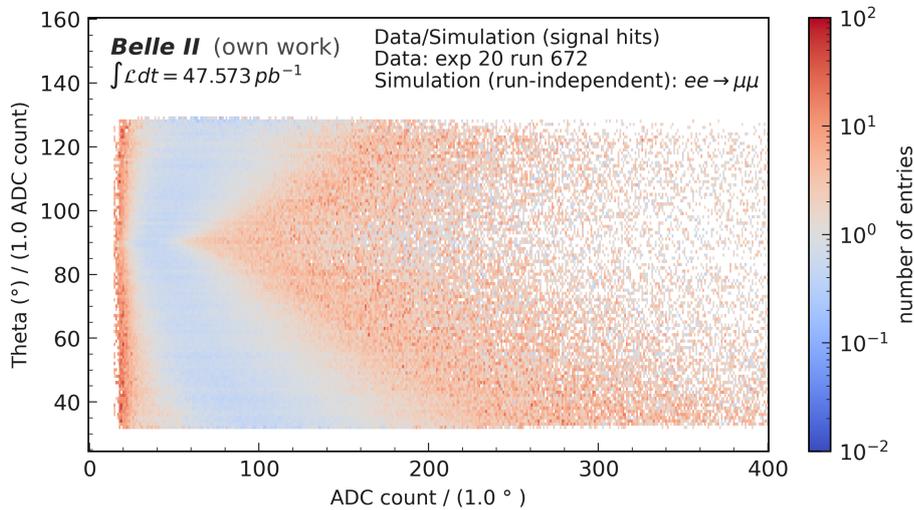


Figure A.15.: This plot demonstrates the ratio of the experimental (experiment 20, run 672) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

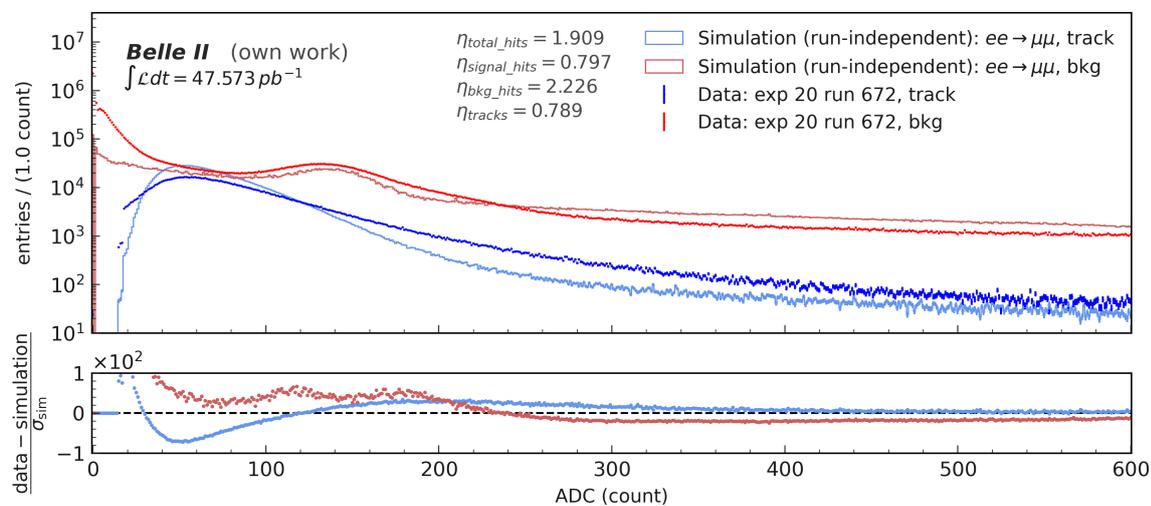


Figure A.16.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 20, run 672, and run-independent Monte Carlo simulation.

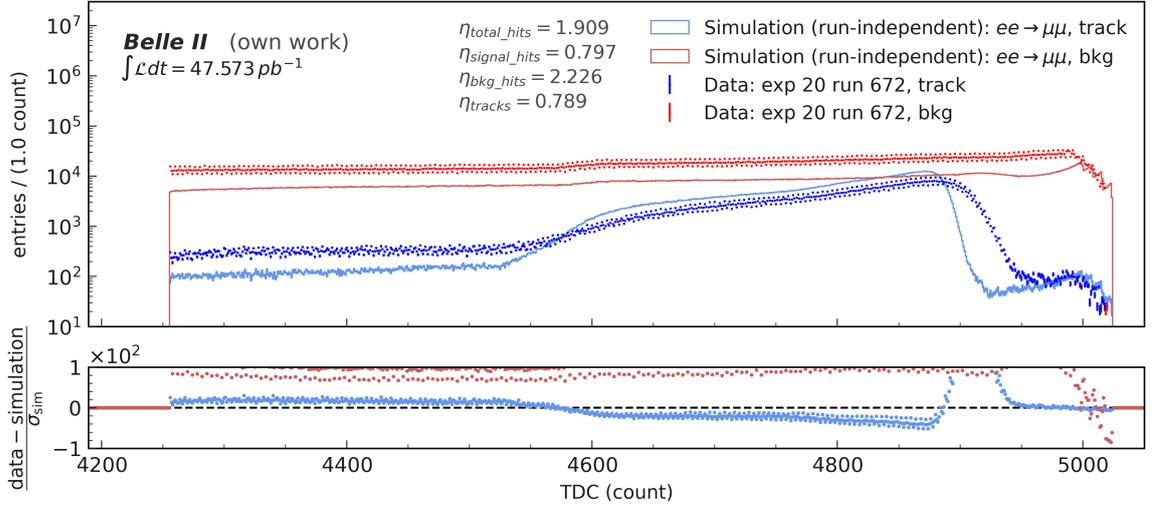


Figure A.17.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 20, run 672, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

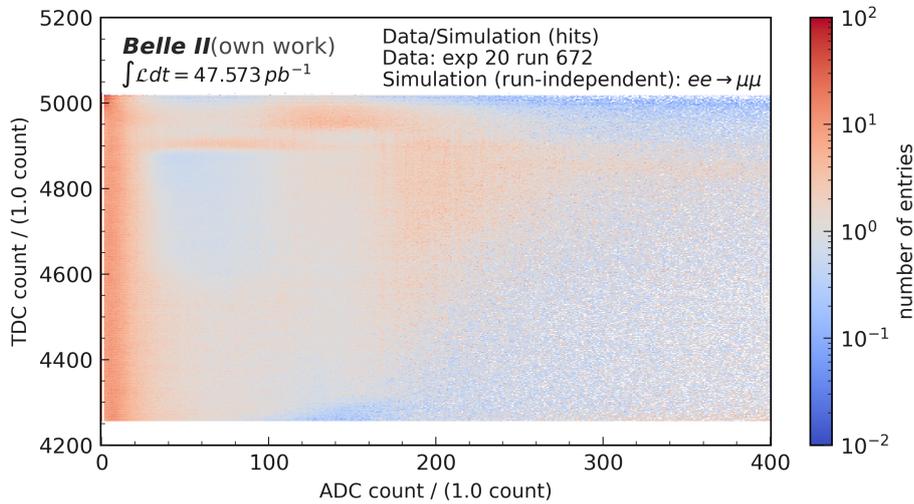


Figure A.18.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 20, run 672, and run-independent Monte Carlo simulation.

## A.4. Experiment 20, Run 874

This section presents figures corresponding to Experiment 20 with Run 874.

Table A.3.: Selection efficiency comparison for the skimmed data sample of experiment 20, run 874, and run-independent simulation. The data consists of  $44.007 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 20, run 874 (%)	MC simulation (%)
1	Theta cut	$75.57^{+0.26}_{-0.26}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection $stt$	$94.33^{+0.16}_{-0.16}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$87.77^{+0.24}_{-0.24}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$94.89^{+0.17}_{-0.17}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.58^{+0.18}_{-0.18}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$56.2^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

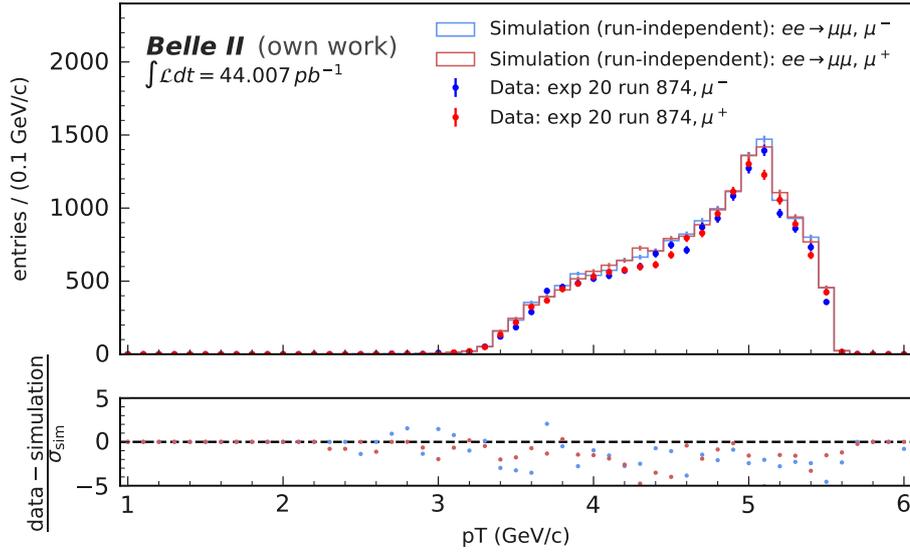


Figure A.19.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 20, run 874, and run-independent Monte Carlo simulation after selection.

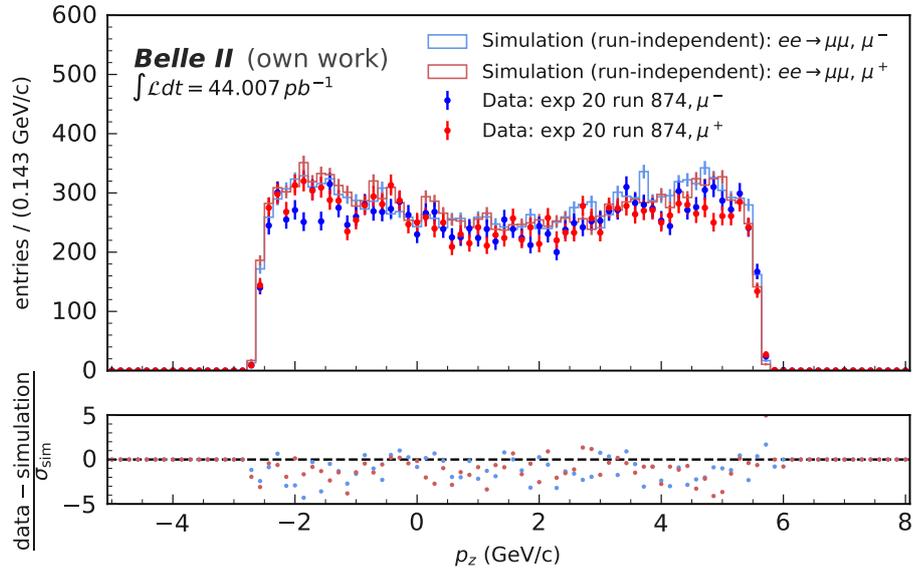


Figure A.20.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 20, run 874, and run-independent Monte Carlo simulation after selection.

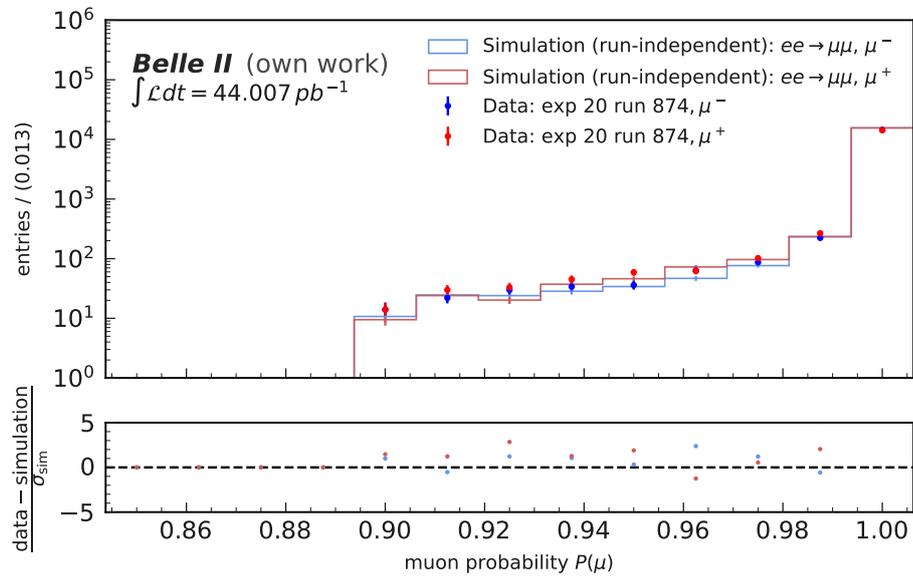


Figure A.21.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 20, run 874, and run-independent Monte Carlo simulation after selection.

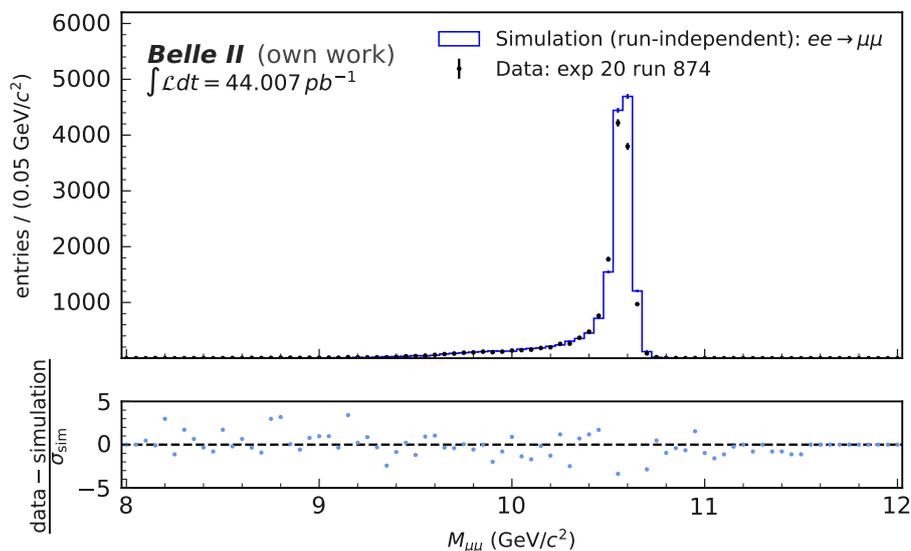


Figure A.22.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 20, run 874, and run-independent simulation.

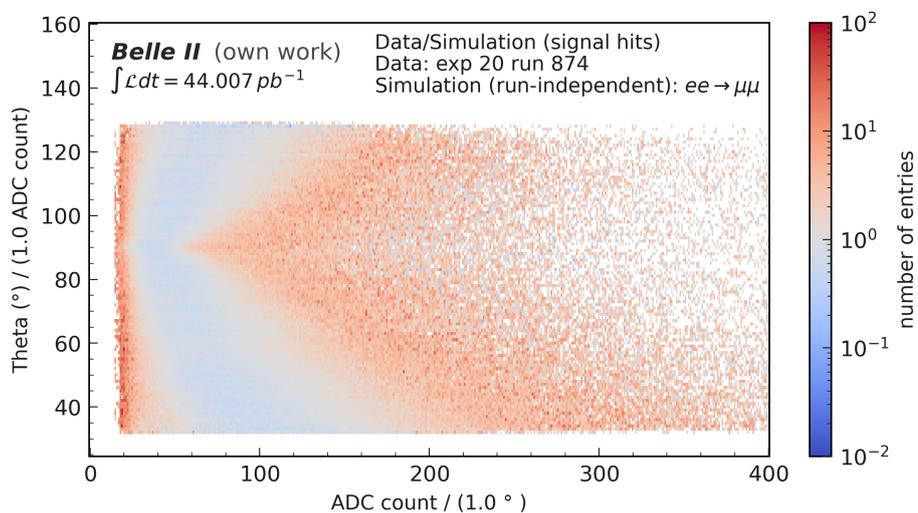


Figure A.23.: This plot demonstrates the ratio of the experimental (experiment 20, run 874) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

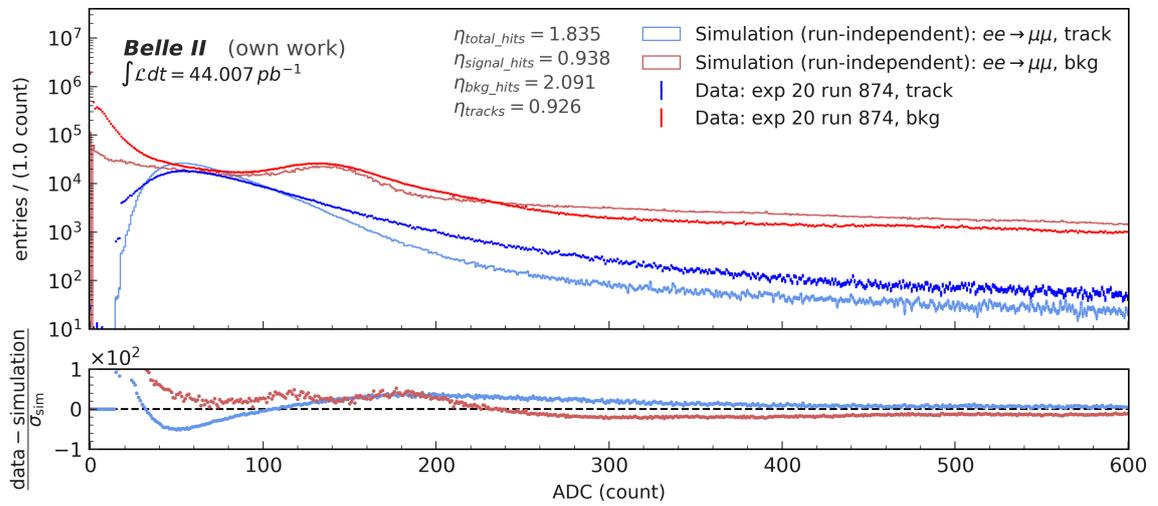


Figure A.24.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 20, run 874, and run-independent Monte Carlo simulation.

## A. Appendix: Input Feature Analysis

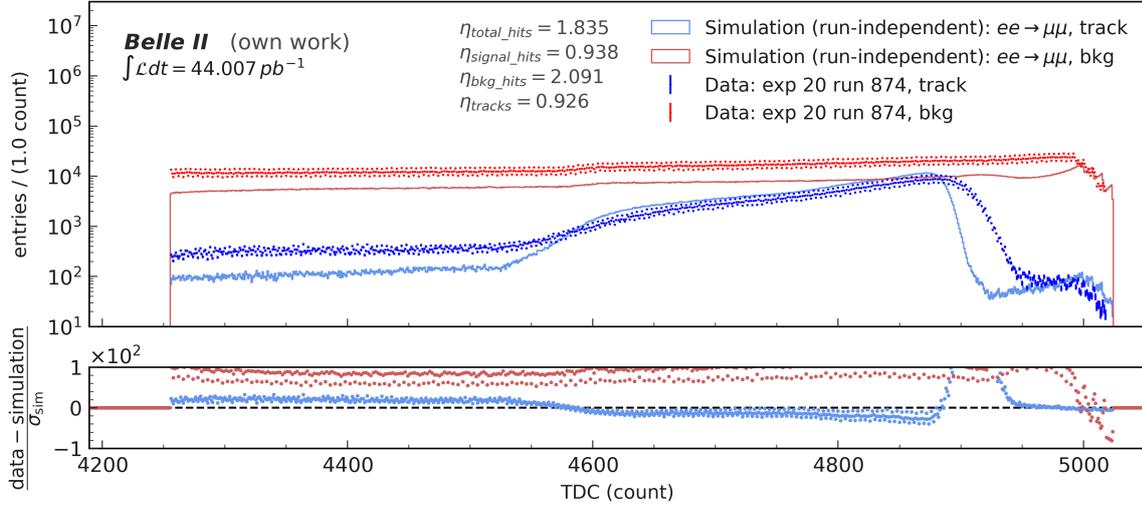


Figure A.25.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 20, run 874, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

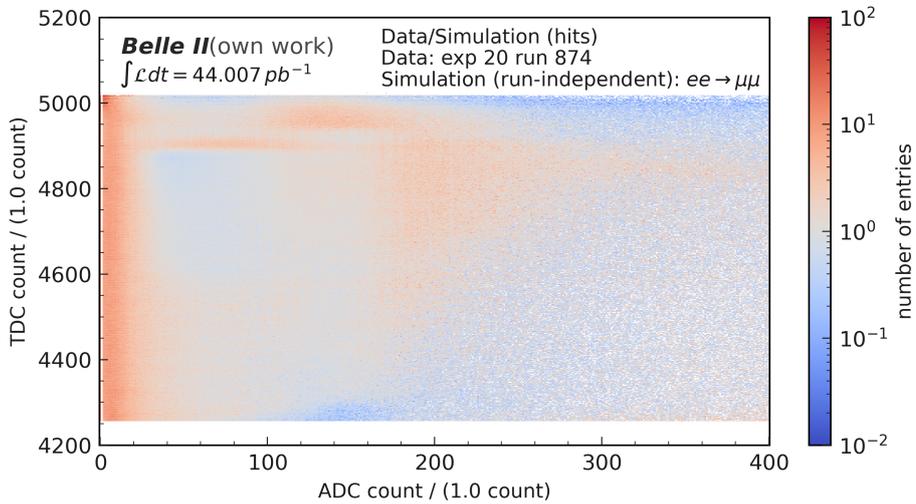


Figure A.26.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 20, run 874, and run-independent Monte Carlo simulation.

## A.5. Experiment 21, Run 116

This section presents figures corresponding to Experiment 21 with Run 116.

Table A.4.: Selection efficiency comparison for the skimmed data sample of experiment 21, run 116, and run-independent simulation. The data consists of  $47.118 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 21, run 116 (%)	MC simulation (%)
1	Theta cut	$75.40^{+0.27}_{-0.27}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection <i>stt</i>	$94.33^{+0.16}_{-0.17}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$87.82^{+0.24}_{-0.24}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$95.05^{+0.17}_{-0.17}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.30^{+0.19}_{-0.19}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$56.0^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

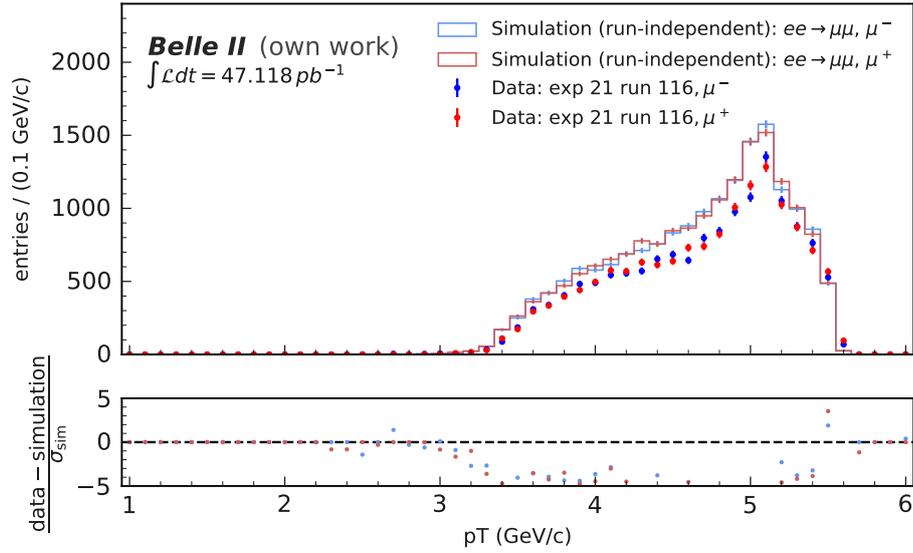


Figure A.27.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 21, run 116, and run-independent Monte Carlo simulation after selection.

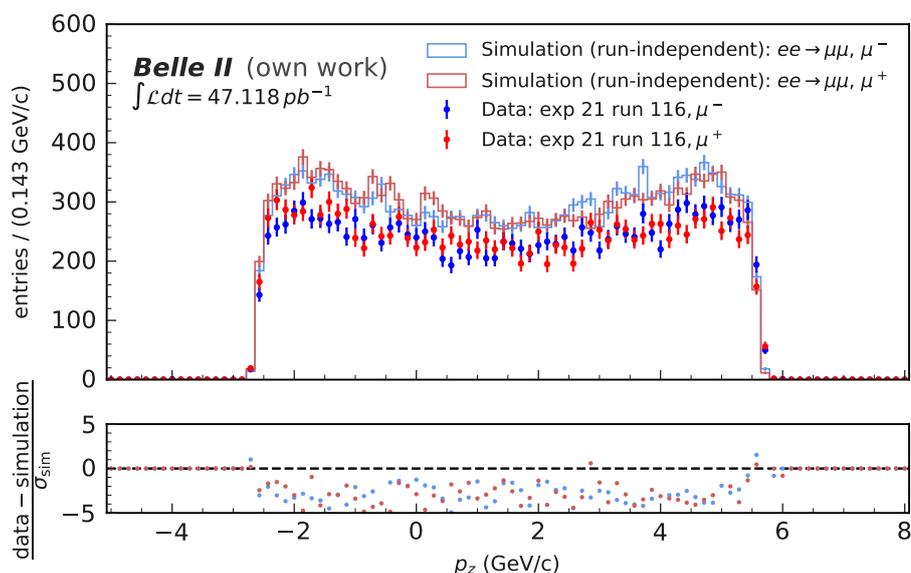


Figure A.28.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 21, run 116, and run-independent Monte Carlo simulation after selection.

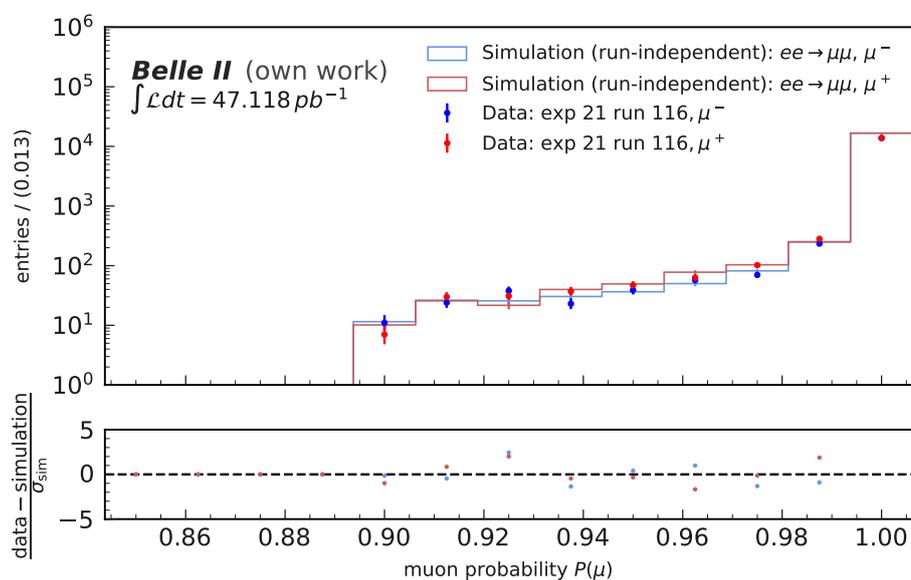


Figure A.29.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 21, run 116, and run-independent Monte Carlo simulation after selection.

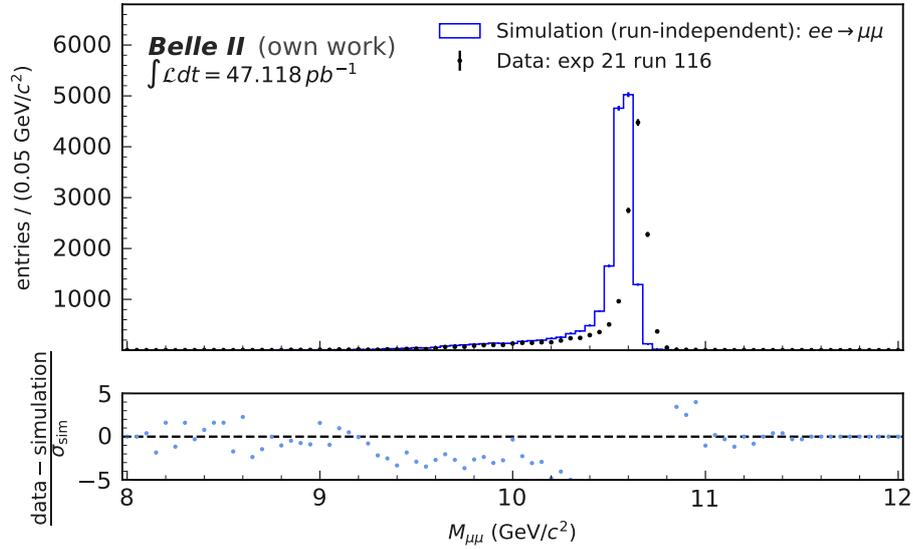


Figure A.30.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 21, run 116, and run-independent simulation.

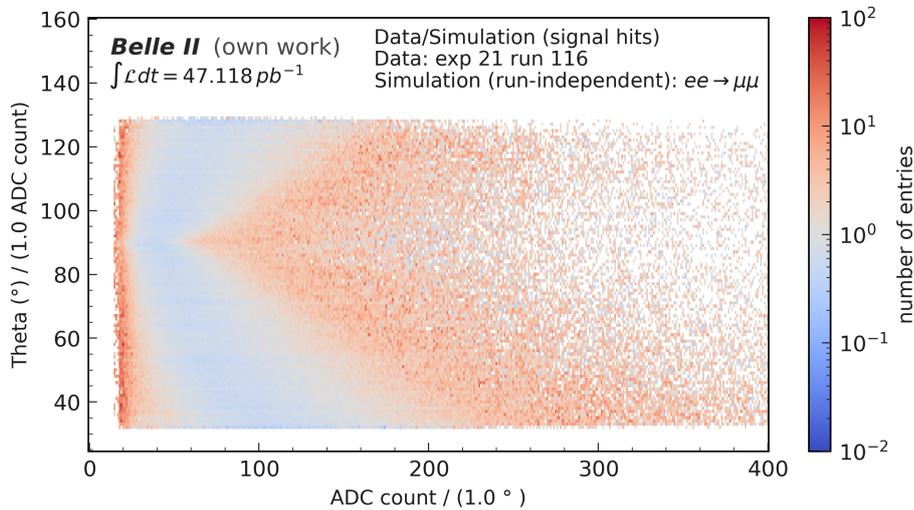


Figure A.31.: This plot demonstrates the ratio of the experimental (experiment 21, run 116) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

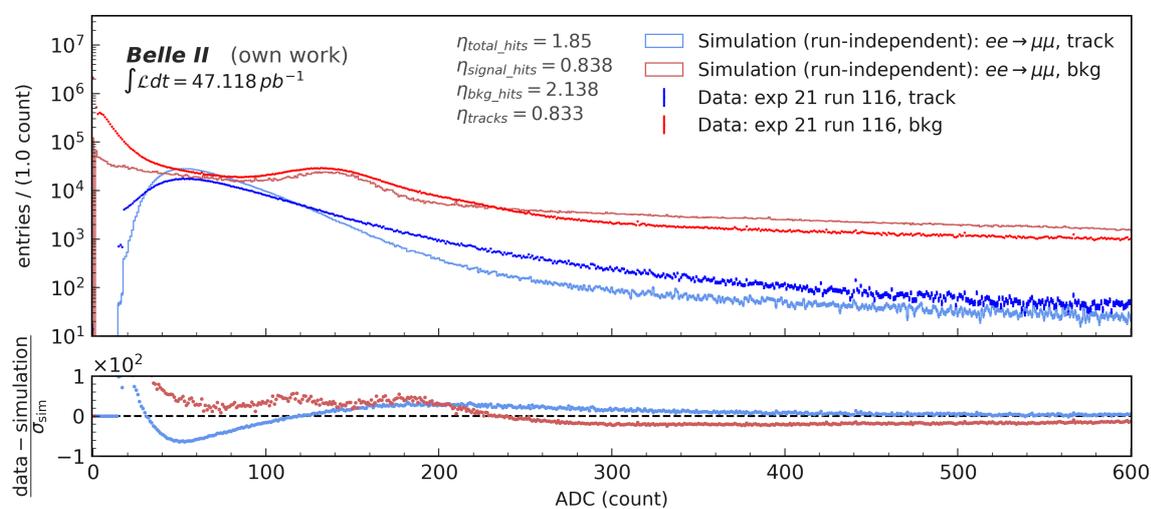


Figure A.32.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 21, run 116, and run-independent Monte Carlo simulation.

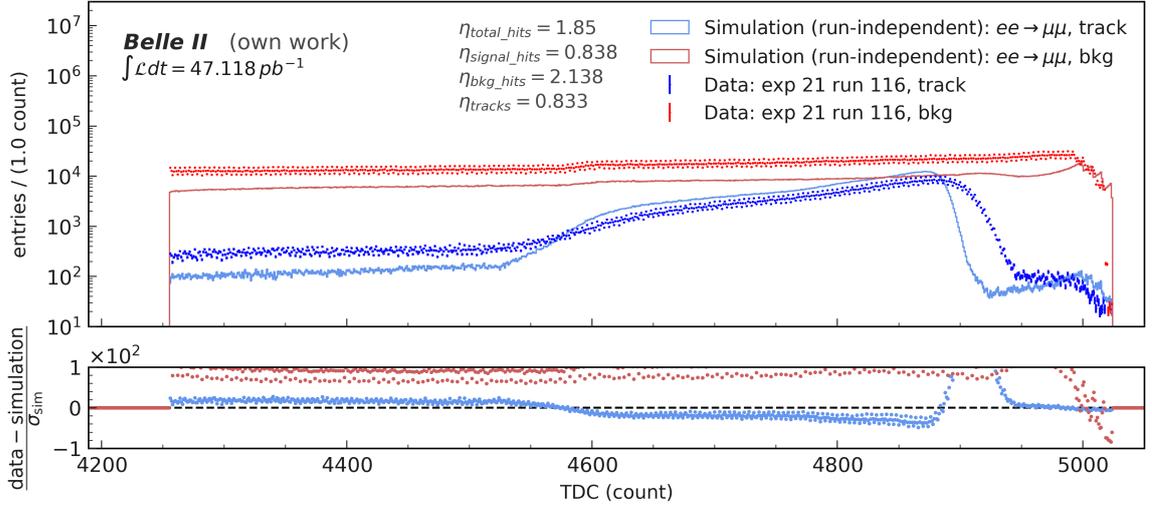


Figure A.33.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 21, run 116, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

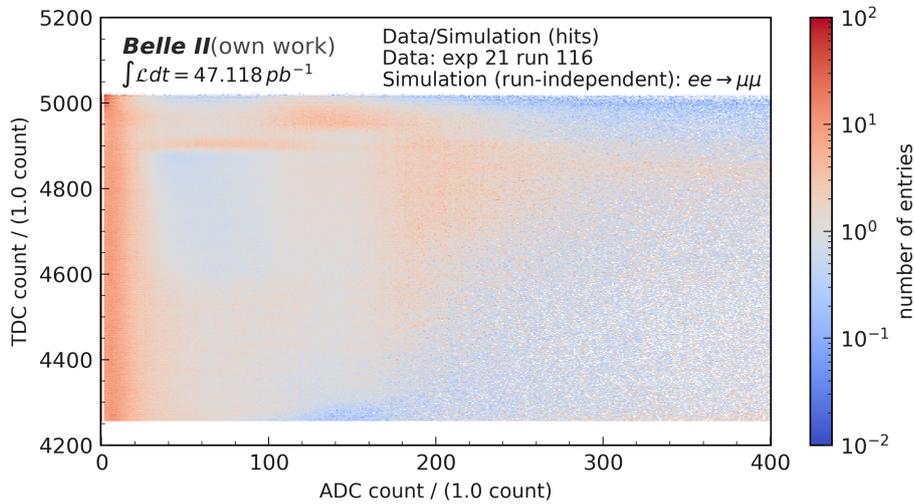


Figure A.34.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 21, run 116, and run-independent Monte Carlo simulation.

## A.6. Experiment 21, Run 128

This section presents figures corresponding to Experiment 21 with Run 128.

Table A.5.: Selection efficiency comparison for the skimmed data sample of experiment 21, run 128, and run-independent simulation. The data consists of  $37.740 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 21, run 128 (%)	MC simulation (%)
1	Theta cut	$75.2^{+0.3}_{-0.3}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection $stt$	$94.70^{+0.18}_{-0.18}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$86.91^{+0.28}_{-0.28}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$94.76^{+0.20}_{-0.20}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.74^{+0.20}_{-0.21}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$55.6^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

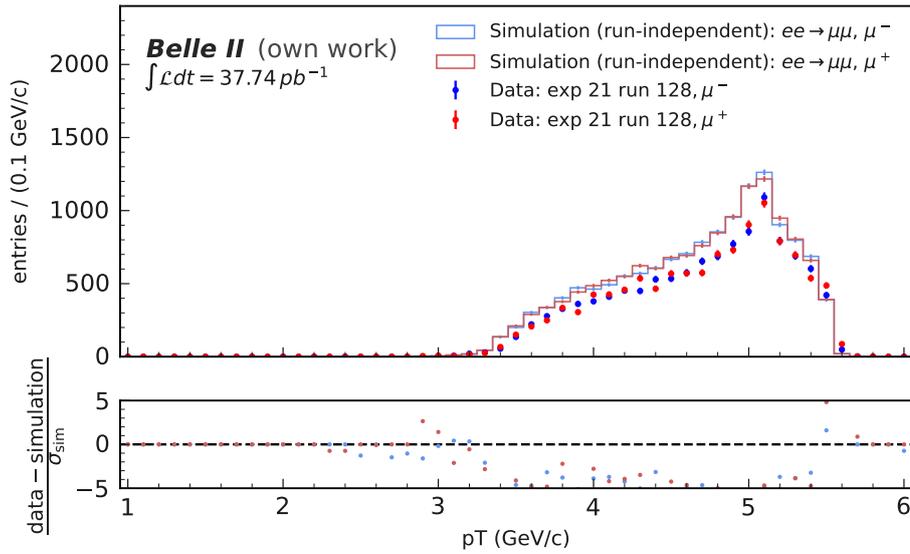


Figure A.35.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 21, run 128, and run-independent Monte Carlo simulation after selection.

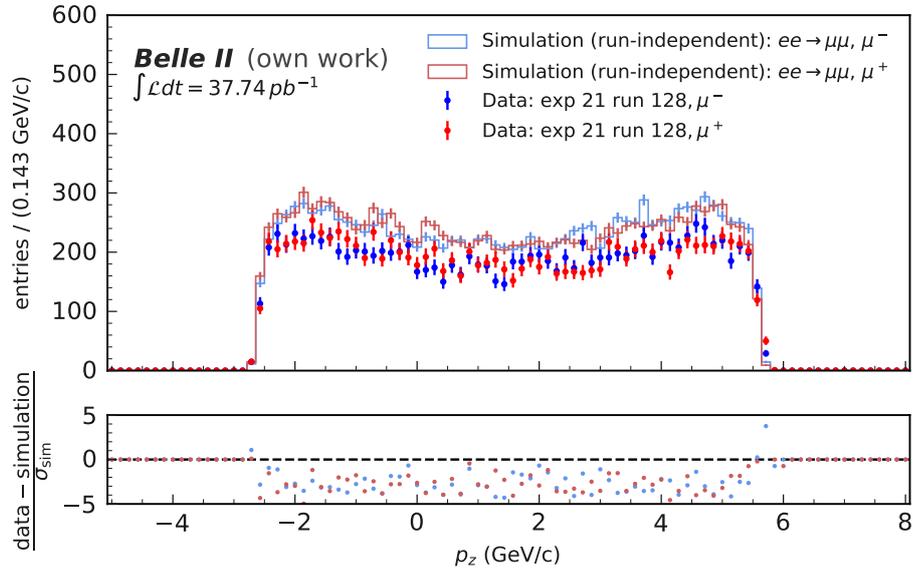


Figure A.36.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 21, run 128, and run-independent Monte Carlo simulation after selection.

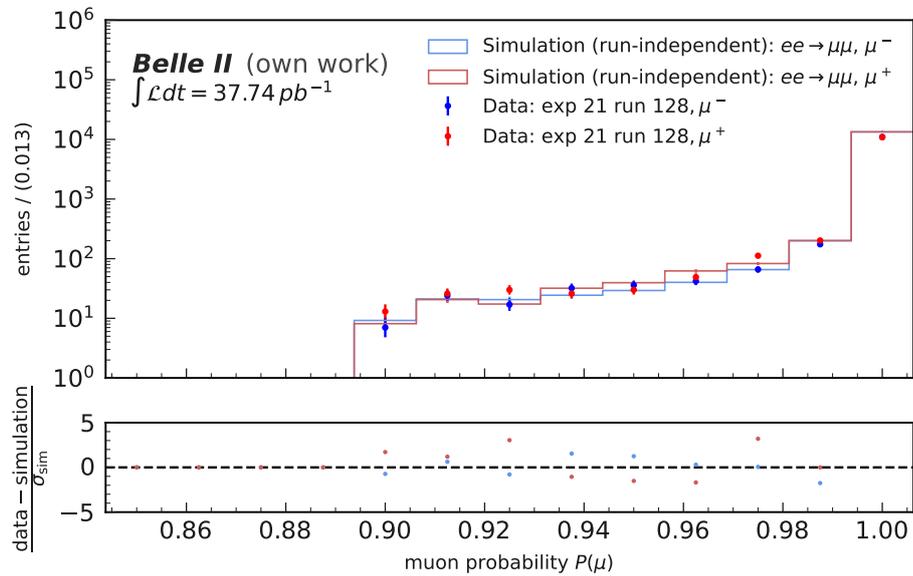


Figure A.37.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 21, run 128, and run-independent Monte Carlo simulation after selection.

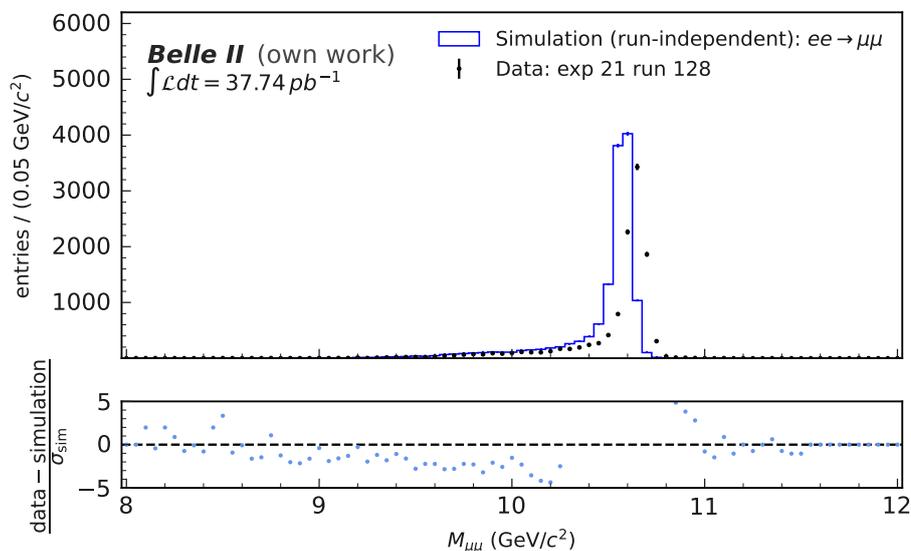


Figure A.38.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 21, run 128, and run-independent simulation.

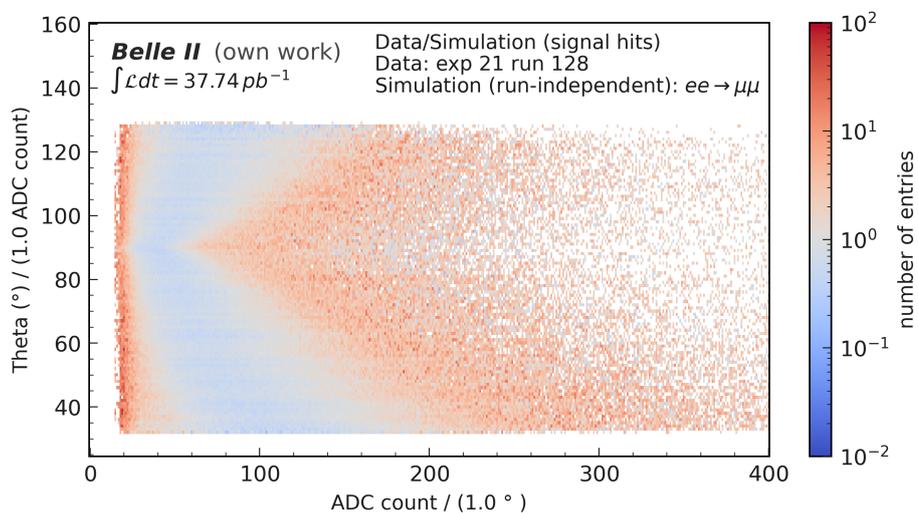


Figure A.39.: This plot demonstrates the ratio of the experimental (experiment 21, run 128) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

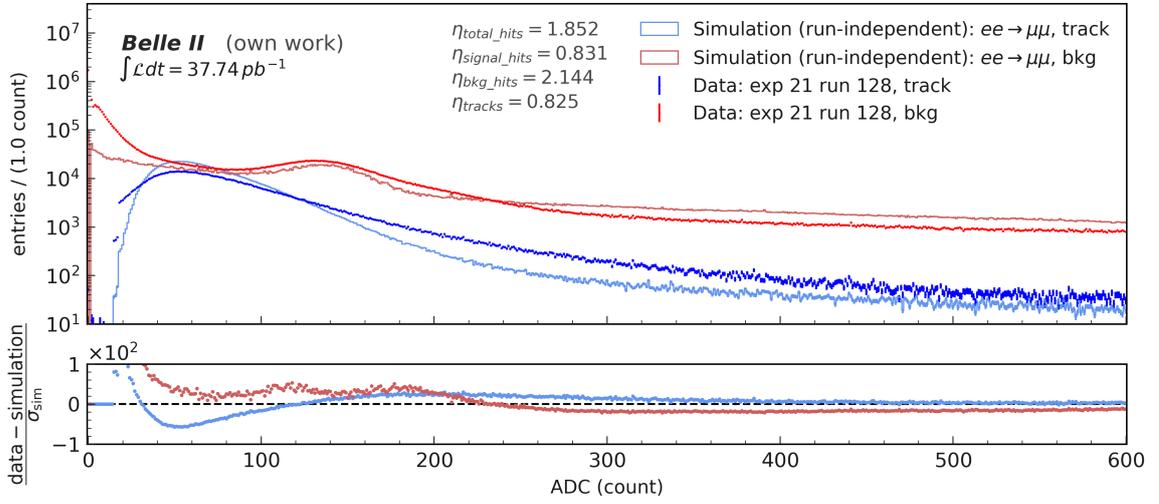


Figure A.40.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 21, run 128, and run-independent Monte Carlo simulation.

## A. Appendix: Input Feature Analysis

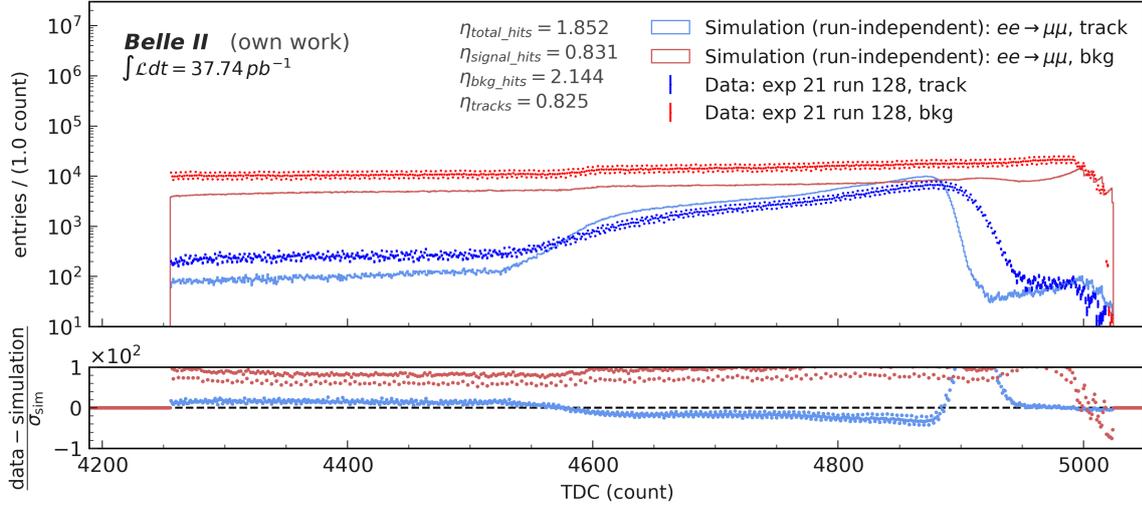


Figure A.41.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 21, run 128, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

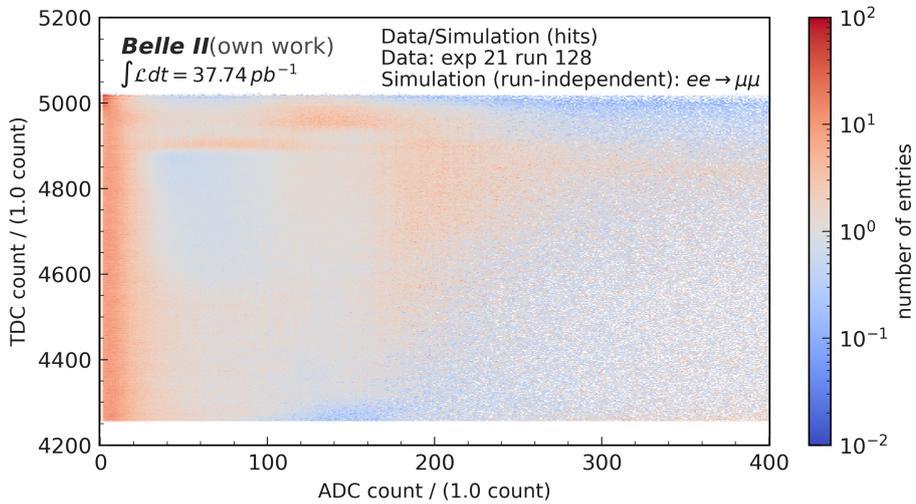


Figure A.42.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 21, run 128, and run-independent Monte Carlo simulation.

## A.7. Experiment 22, Run 32

This section presents figures corresponding to Experiment 22 with Run 32.

Table A.6.: Selection efficiency comparison for the skimmed data sample of experiment 22, run 32, and run-independent simulation. The data consists of  $39.656 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 22, run 32 (%)	MC simulation (%)
1	Theta cut	$74.62^{+0.26}_{-0.27}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection $stt$	$94.33^{+0.16}_{-0.16}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$88.41^{+0.23}_{-0.23}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$94.64^{+0.17}_{-0.18}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.42^{+0.18}_{-0.18}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$55.61^{+0.3}_{-0.4}$	$68.39^{+0.24}_{-0.24}$

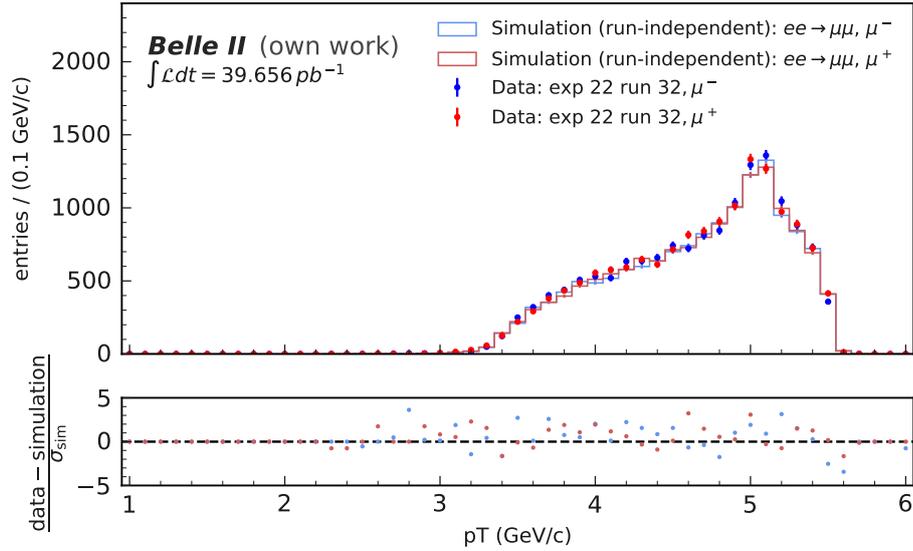


Figure A.43.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 22, run 32, and run-independent Monte Carlo simulation after selection.

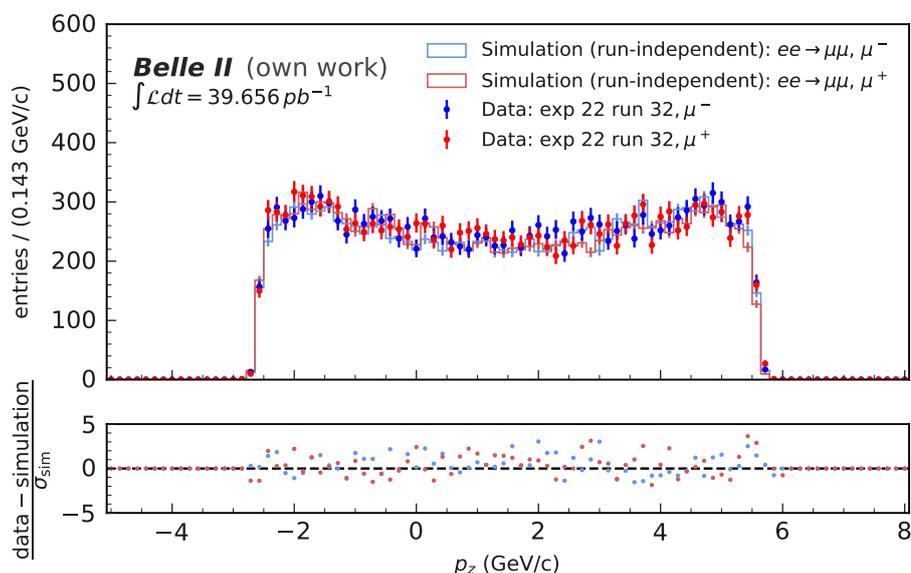


Figure A.44.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 22, run 32, and run-independent Monte Carlo simulation after selection.

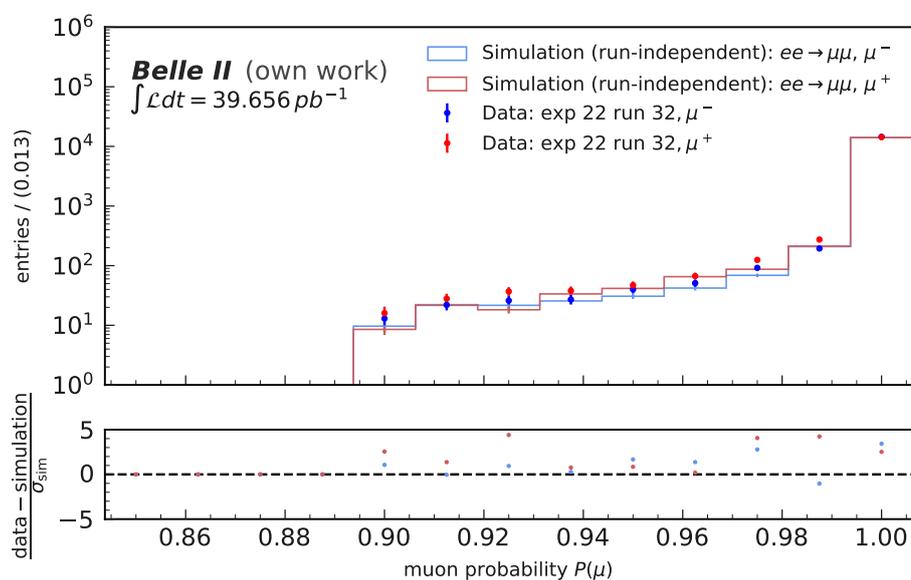


Figure A.45.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 22, run 32, and run-independent Monte Carlo simulation after selection.

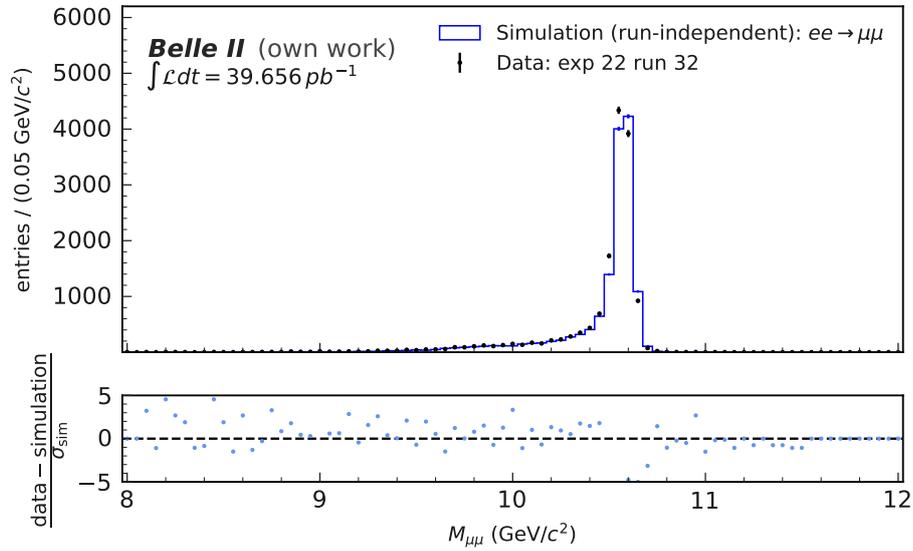


Figure A.46.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 22, run 32, and run-independent simulation.

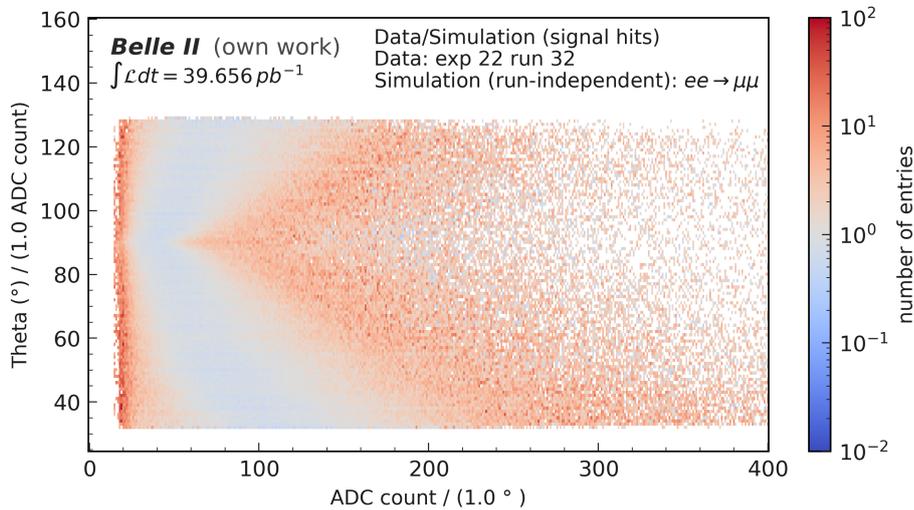


Figure A.47.: This plot demonstrates the ratio of the experimental (experiment 22, run 32) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

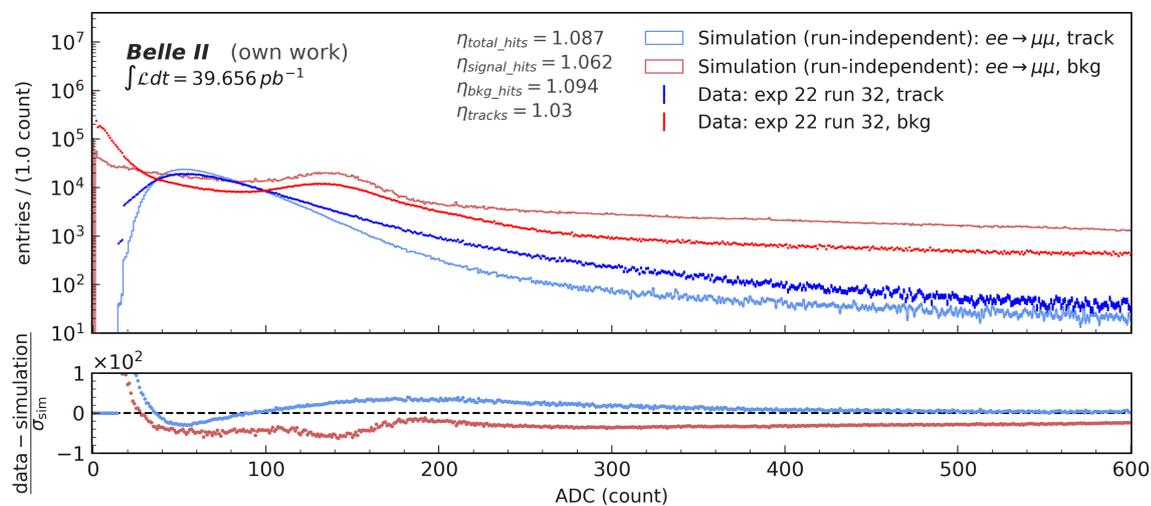


Figure A.48.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 22, run 32, and run-independent Monte Carlo simulation.

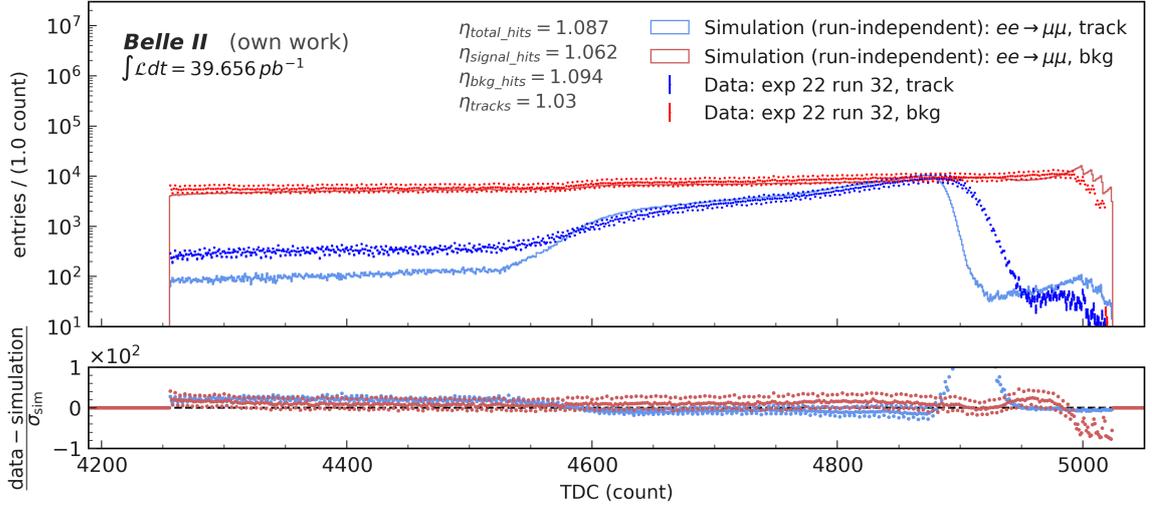


Figure A.49.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 22, run 32, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

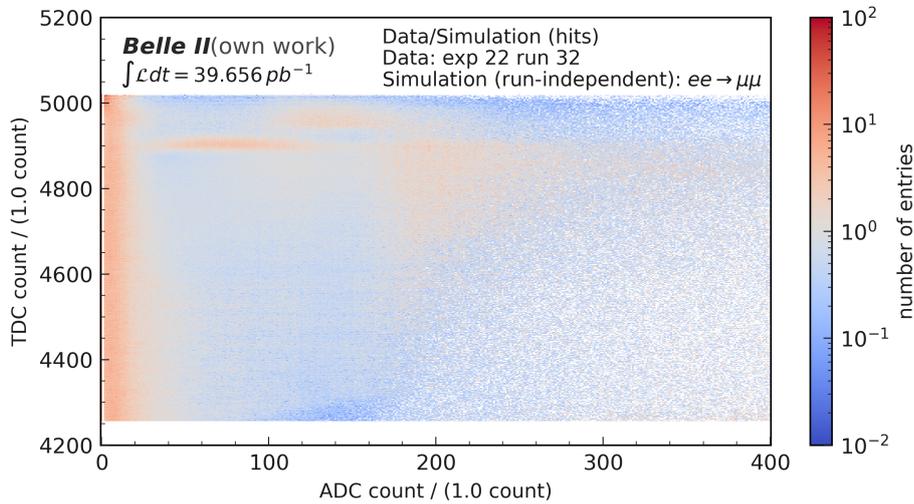


Figure A.50.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 22, run 32, and run-independent Monte Carlo simulation.

## A.8. Experiment 22, Run 546

This section presents figures corresponding to Experiment 22 with Run 546.

Table A.7.: Selection efficiency comparison for the skimmed data sample of experiment 22, run 546, and run-independent simulation. The data consists of  $38.853 \text{ pb}^{-1}$ , and the simulation consists of 80 000 events.

Order No	Selection	Exp. 22, run 546 (%)	MC simulation (%)
1	Theta cut	$74.46^{+0.28}_{-0.28}$	$75.46^{+0.22}_{-0.22}$
2	L1 selection $stt$	$93.92^{+0.17}_{-0.18}$	$99.56^{+0.4}_{-0.4}$
3	Dimuon track selection	$87.65^{+0.25}_{-0.25}$	$95.50^{+0.12}_{-0.12}$
4	Dimuon mass cut	$95.12^{+0.17}_{-0.18}$	$99.839^{+0.023}_{-0.026}$
5	PID cut	$94.05^{+0.19}_{-0.20}$	$95.47^{+0.13}_{-0.13}$
	Total efficiency	$54.8^{+0.3}_{-0.3}$	$68.39^{+0.24}_{-0.24}$

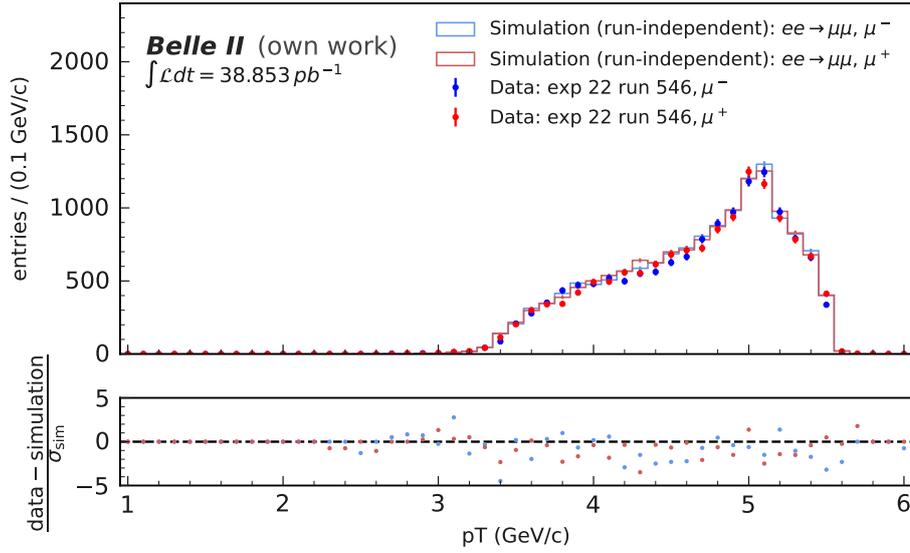


Figure A.51.: Distribution of the transversal momentum  $p_T$  of reconstructed tracks for experiment 22, run 546, and run-independent Monte Carlo simulation after selection.

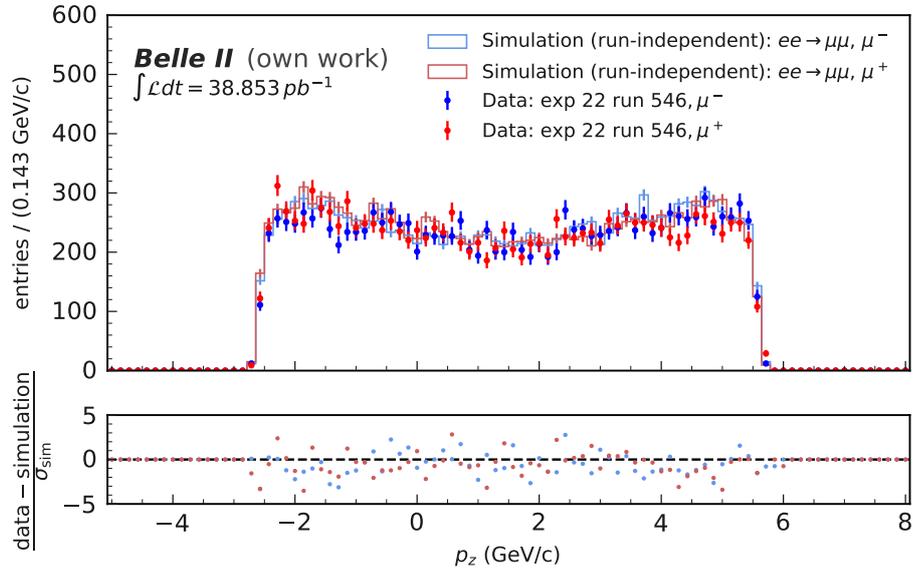


Figure A.52.: Distribution of the longitudinal momentum  $p_z$  of reconstructed tracks for experiment 22, run 546, and run-independent Monte Carlo simulation after selection.

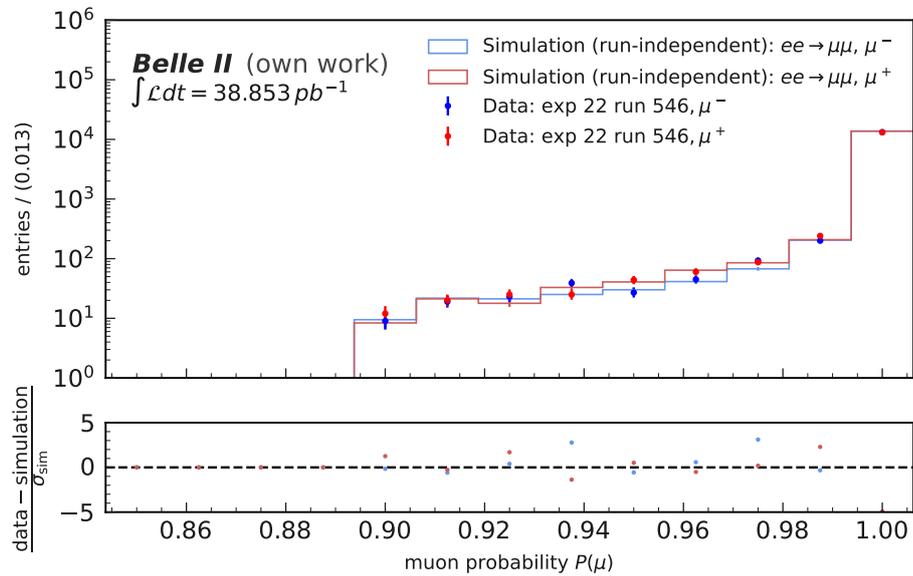


Figure A.53.: Reconstructed muon Particle Identification  $P(\mu)$  distributions for experiment 22, run 546, and run-independent Monte Carlo simulation after selection.

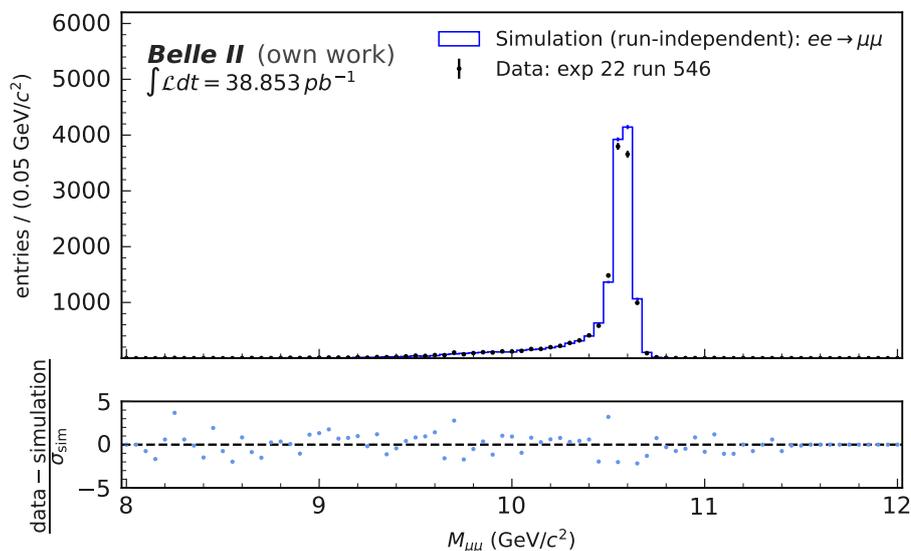


Figure A.54.: Distribution of the reconstructed invariant Dimuon mass  $M_{\mu\mu}$  after selection. for experiment experiment 22, run 546, and run-independent simulation.

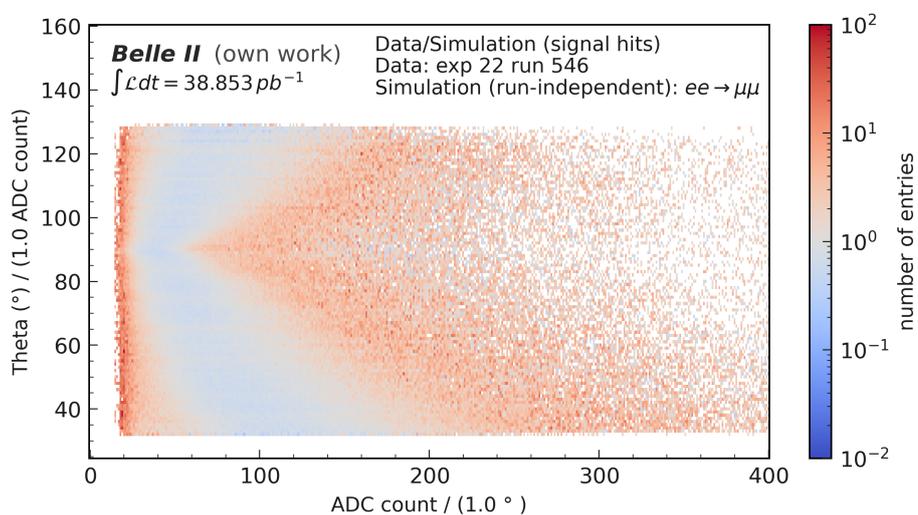


Figure A.55.: This plot demonstrates the ratio of the experimental (experiment 22, run 546) and run-independent simulated Analog-to-Digital Converter counts of Central Drift Chamber hits as a function of the polar angle  $\theta$ . The angle  $\theta$  is measured from the beam axis. The plot illustrates the relative agreement between the data and simulated distributions.

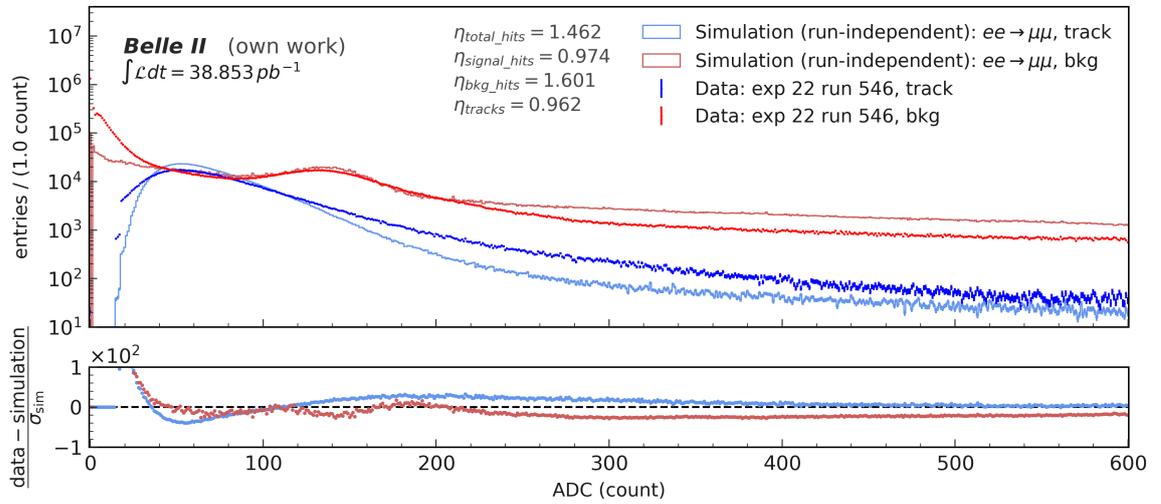


Figure A.56.: Comparison of Analog-to-Digital Converter (ADC) count distributions for signal and background hits. Two blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the two red histograms denote the background hits. The figure highlights the separation potential between signal and background using ADC counts as an input feature for analysis. Experimental data from experiment 22, run 546, and run-independent Monte Carlo simulation.

A. Appendix: Input Feature Analysis

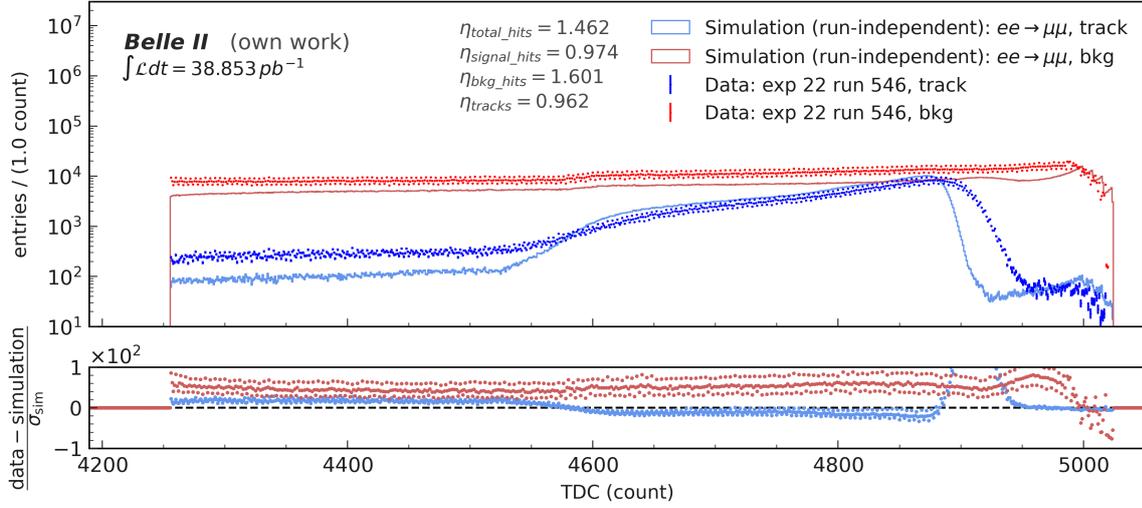


Figure A.57.: Comparison of Time-to-Digital Converter distributions for signal and background hits in experiment 22, run 546, and run-independent simulation. The blue histograms represent hits assigned to reconstructed signal tracks for data and simulation, while the red histograms denote the background hits. The figure highlights a discriminatory potential between signal and background TDC counts due to the distinctive characteristics in the signal distribution. Oscillations observed in the data's TDC distribution require further investigation.

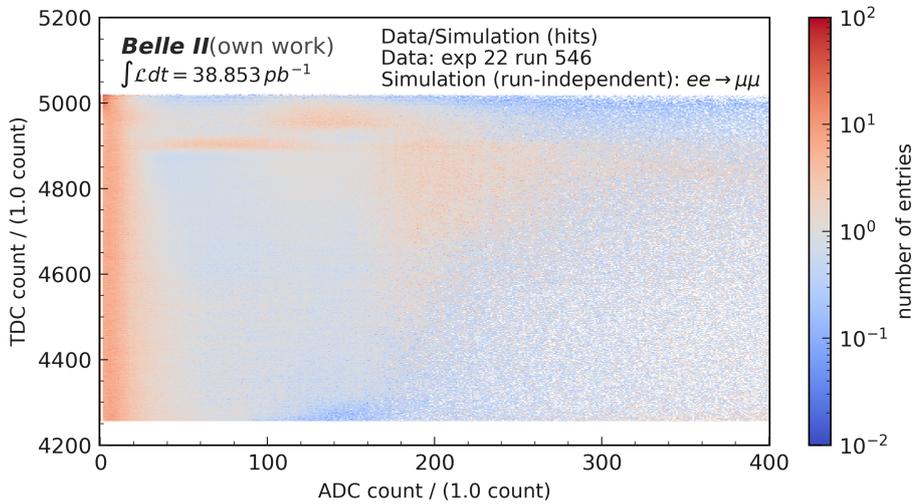


Figure A.58.: This plot depicts the ratio of experimental to simulated Analog-to-Digital Converter (ADC) and Time-to-Digital Converter (TDC) counts for Central Drift Chamber (CDC) hits, providing a method for evaluating the relative agreement between the two distributions. Experimental data from experiment 22, run 546, and run-independent Monte Carlo simulation.

## **B. Appendix: Graph-Building**

This appendix presents additional information for the analysis of the graph-building model with a *nominal Phase 3* background overlay on the signal simulation and an analysis with *early Phase 3* background.

### **B.1. Additional Information on the Graph-Building With *Nominal Phase 3* Background**

This section presents additional tables and figures corresponding to the graph-building analysis with *nominal Phase 3* background overlays.

Table B.1.: Median number of edges resulting from the ten different graph-building models with *nominal Phase 3* background for each superlayer of the Central Drift Chamber.

Model	Total	SL 1	SL 2	SL 3	SL 4	SL 5	SL 6	SL 7
01	3100 ± 2	604 ± 1	384 ± 1	321 ± 1	344 ± 1	345 ± 1	272 ± 1	305 ± 1
02	5810 ± 5	1148 ± 2	747.0 ± 1.2	599 ± 1	652 ± 1	641 ± 1	505 ± 1	558 ± 1
03	4017 ± 3	828 ± 1	510.0 ± 0.8	416 ± 1	440 ± 1	439 ± 1	342 ± 1	382 ± 1
04	5834 ± 5	1261 ± 2	762 ± 1	604 ± 1	635 ± 1	626 ± 1	482 ± 1	535 ± 1
05	6727 ± 5	1371 ± 2	873 ± 1	693 ± 1	749 ± 1	734 ± 1	575 ± 1	635 ± 1
06	8545 ± 7	1805 ± 3	1125 ± 2	881 ± 2	943 ± 2	921 ± 2	715 ± 1	789 ± 1
07	6435 ± 5	1244 ± 2	810 ± 1	655 ± 1	718 ± 1	711 ± 1	570 ± 1	630 ± 1
08	8253 ± 6	1678.0 ± 2.6	1062.0 ± 1.6	843.0 ± 1.3	912.0 ± 1.5	899.0 ± 1.4	710.0 ± 1.2	783.0 ± 1.2
09	9145 ± 7	1788.0 ± 2.7	1173.0 ± 1.8	933.0 ± 1.4	1026.0 ± 1.6	1006.0 ± 1.5	803.0 ± 1.4	882.0 ± 1.4
10	10961 ± 8	2221 ± 4	1424.0 ± 2.2	1121.0 ± 1.8	1221.0 ± 2.0	1194.0 ± 1.8	942.0 ± 1.7	1036.0 ± 1.6
Model	SL 8	SL 9						
01	256.5 ± 0.5	245 ± 4						
02	470 ± 1	442 ± 1						
03	320 ± 1	306 ± 1						
04	445 ± 1	428 ± 1						
05	533 ± 1	503 ± 1						
06	658 ± 1	625 ± 1						
07	536 ± 1	515 ± 1						
08	660.0 ± 1.2	637.0 ± 1.0						
09	749.0 ± 1.3	713.0 ± 1.1						
10	874.0 ± 1.6	835.0 ± 1.3						

## B.2. Graph-Building Analysis with *early Phase 3* Background

This section presents figures and tables corresponding to the graph-building analysis with *Early Phase 3* background overlays.

Table B.2.: Median edges for the ten different graph-building models for *early Phase 3* background. Each event possesses a median of  $66.00 \pm 0.20$  true graph edges and  $149.0 \pm 2.2$  true edges. The table lists the total edges created from each model, including true graph edges and true edges. The maximum number of edges one single node can have in the respective model is shown in the last column.

model	total number of edges	incl. graph	incl. true edges	max. connections
01	$231.0 \pm 0.6$	$56.00 \pm 0.17$	$63.00 \pm 0.27$	4
02	$365.0 \pm 1.1$	$58.00 \pm 0.18$	$73.0 \pm 0.5$	8
03	$282.0 \pm 0.8$	$57.00 \pm 0.18$	$88.0 \pm 0.4$	6
04	$349.0 \pm 1.1$	$58.00 \pm 0.18$	$108.0 \pm 0.5$	10
05	$416.0 \pm 1.3$	$59.00 \pm 0.18$	$99.0 \pm 0.6$	10
06	$483.0 \pm 1.6$	$60.00 \pm 0.18$	$119.0 \pm 0.7$	14
07	$468.0 \pm 1.2$	$61.00 \pm 0.18$	$102.0 \pm 0.5$	8
08	$535.0 \pm 1.5$	$62.00 \pm 0.19$	$122.0 \pm 0.7$	12
09	$602.0 \pm 1.7$	$63.00 \pm 0.19$	$112.0 \pm 0.7$	12
10	$669.0 \pm 2.0$	$65.00 \pm 0.19$	$132.0 \pm 0.8$	16

Table B.3.: Model-independent median number of edges with different wire and layer distances for *early Phase 3* background.

layer distance	wire distance	total number of edges	max. connections
$\Delta l = 0$	$\Delta w = 1$	$187.0 \pm 0.5$	2
$\Delta l = 1$	$\Delta w = 0$	$231.0 \pm 0.6$	4
$\Delta l = 1$	$\Delta w = 0, 1$	$365.0 \pm 1.1$	8
$\Delta l = 2$	$\Delta w = 0$	$50.00 \pm 0.21$	2
$\Delta l = 2$	$\Delta w = 0, 1$	$118.0 \pm 0.6$	6

Table B.4.: The *true edge eff.* and *true edge pur.* of the ten graph-building models with *early Phase 3* background overlays.

Model	<i>true edge eff.</i> (%)	<i>true edge pur.</i> (%)
01	49.46 ± 0.09	27.22 ± 0.08
02	52.38 ± 0.06	19.60 ± 0.07
03	70.00 ± 0.16	31.90 ± 0.09
04	87.66 ± 0.16	31.18 ± 0.09
05	72.91 ± 0.13	23.84 ± 0.08
06	90.83 ± 0.14	24.40 ± 0.08
07	76.14 ± 0.13	21.92 ± 0.07
08	93.98 ± 0.14	22.65 ± 0.07
09	79.17 ± 0.11	18.51 ± 0.07
10	97.25 ± 0.11	19.43 ± 0.07

Table B.5.: The *true graph eff.* and *true graph pur.* of the ten graph-building models for *early Phase 3* background overlays.

Model	<i>true graph eff.</i> (%)	<i>true graph pur.</i> (%)	<i>overcount</i>
01	91.07 ± 0.09	24.02 ± 0.07	1.0120 ± 0.0013
02	92.16 ± 0.06	15.68 ± 0.05	1.0789 ± 0.0022
03	93.23 ± 0.09	20.19 ± 0.06	1.4902 ± 0.0023
04	95.00 ± 0.09	16.51 ± 0.05	1.7925 ± 0.0027
05	94.20 ± 0.06	14.10 ± 0.05	1.5714 ± 0.0025
06	95.95 ± 0.07	12.33 ± 0.04	1.843 ± 0.004
07	95.65 ± 0.04	12.94 ± 0.04	1.6250 ± 0.0023
08	97.33 ± 0.04	11.49 ± 0.04	1.8876 ± 0.0029
09	97.248 ± 0.016	10.38 ± 0.04	1.712 ± 0.004
10	98.750 ± 0.012	9.471 ± 0.029	1.975 ± 0.004

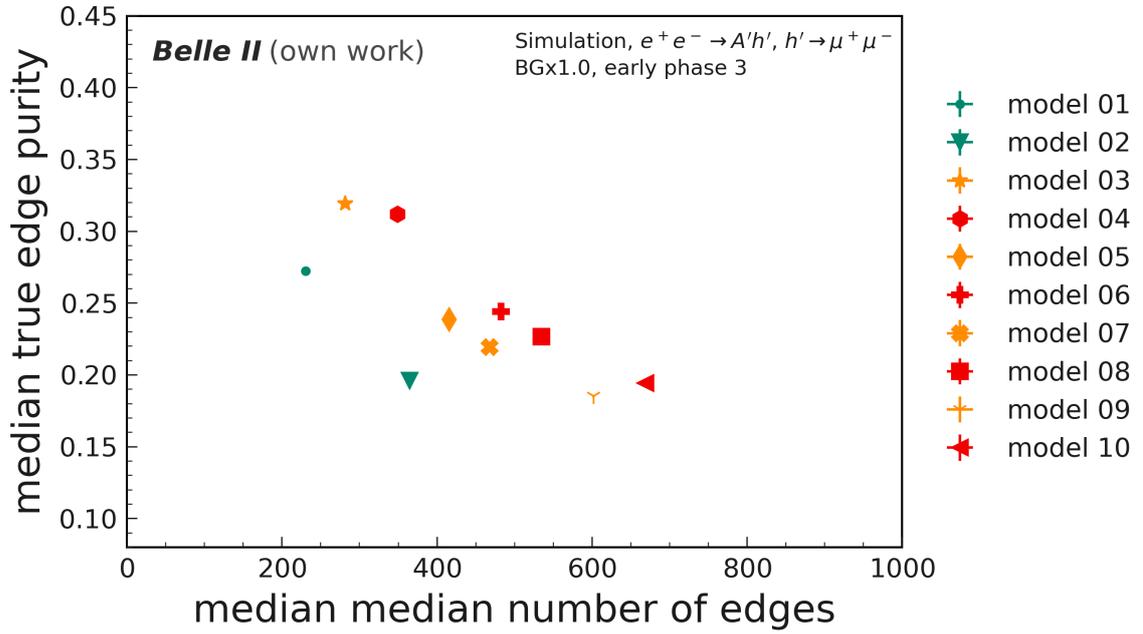


Figure B.1.: The event median of the *true edge pur.* and the median of the *total number of edges* created from each graph-building model. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

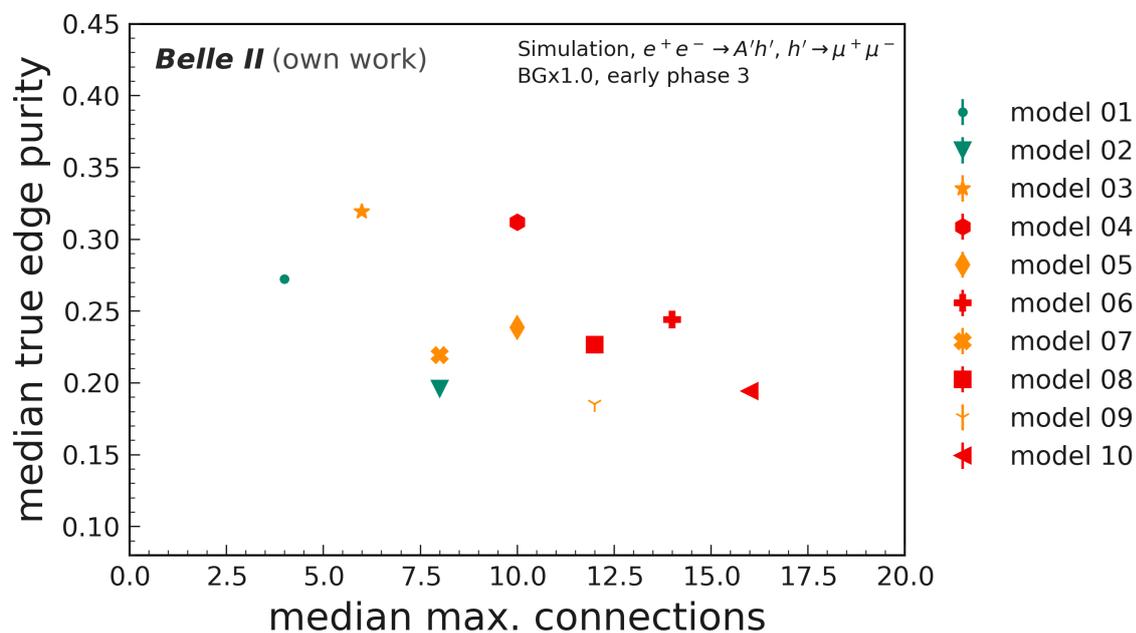


Figure B.2.: *true edge pur.* plotted against the maximum number of possible connections for a single node in a graph-building model. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

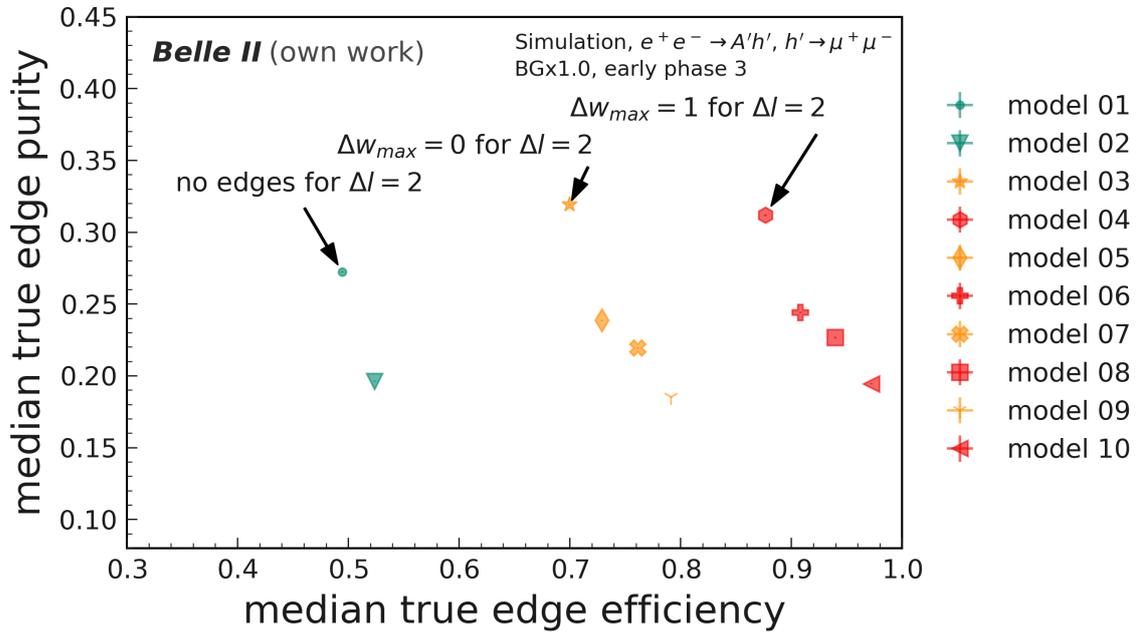


Figure B.3.: The event median of the *true edge eff.* and *true edge pur.* The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

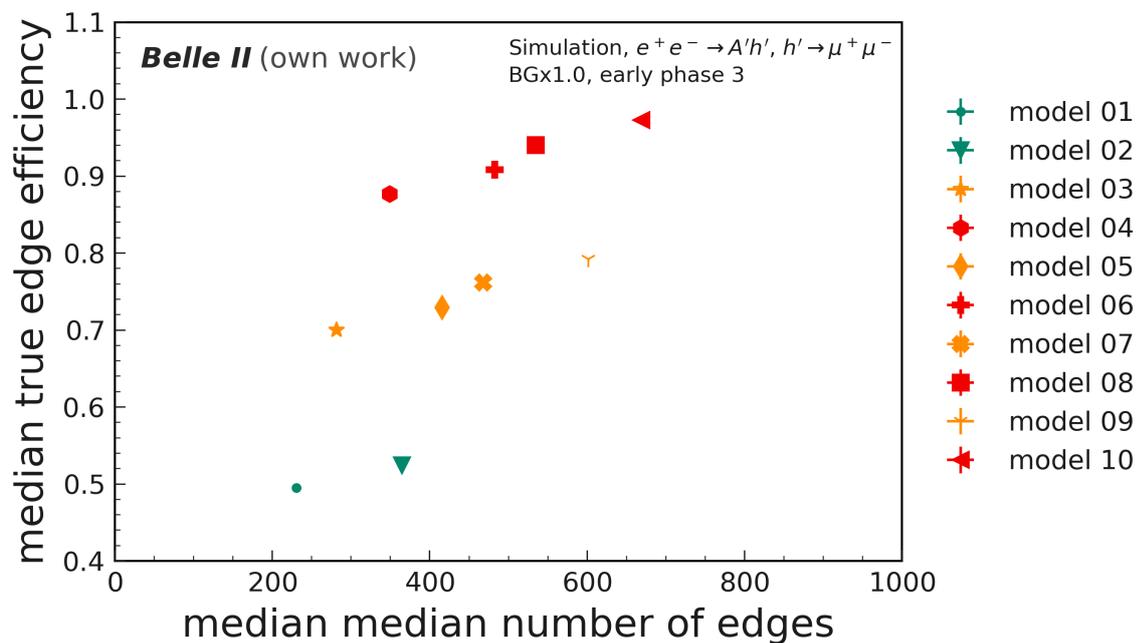


Figure B.4.: The median of *true edge eff.* and *total number of edges* calculated for each graph-building model. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

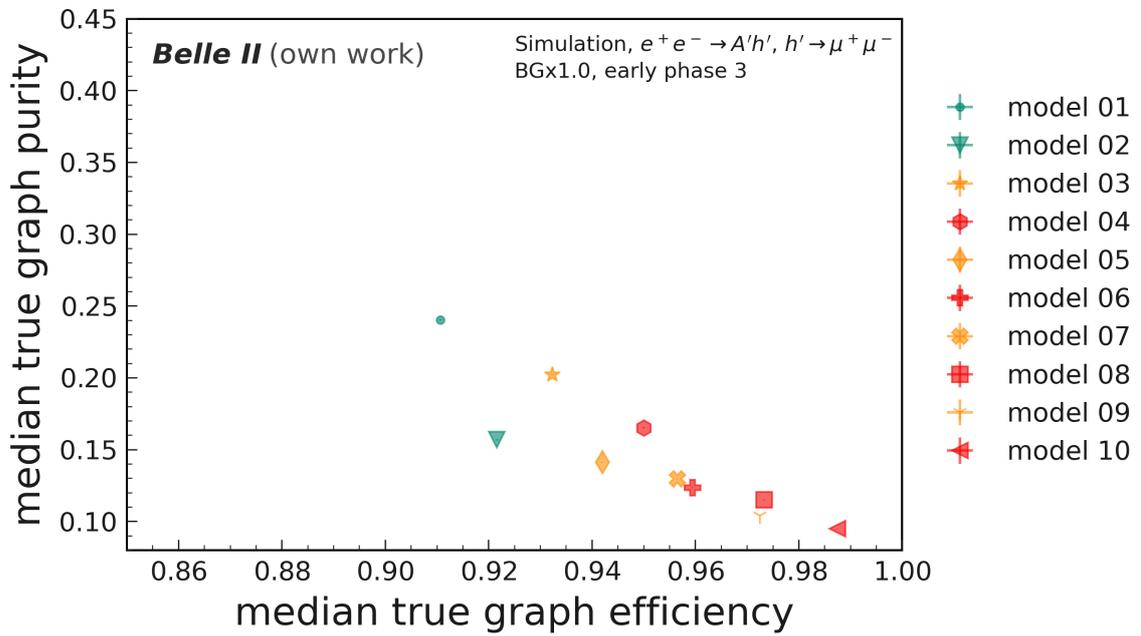


Figure B.5.: The event median of the *true graph eff.* and *true graph pur.*. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

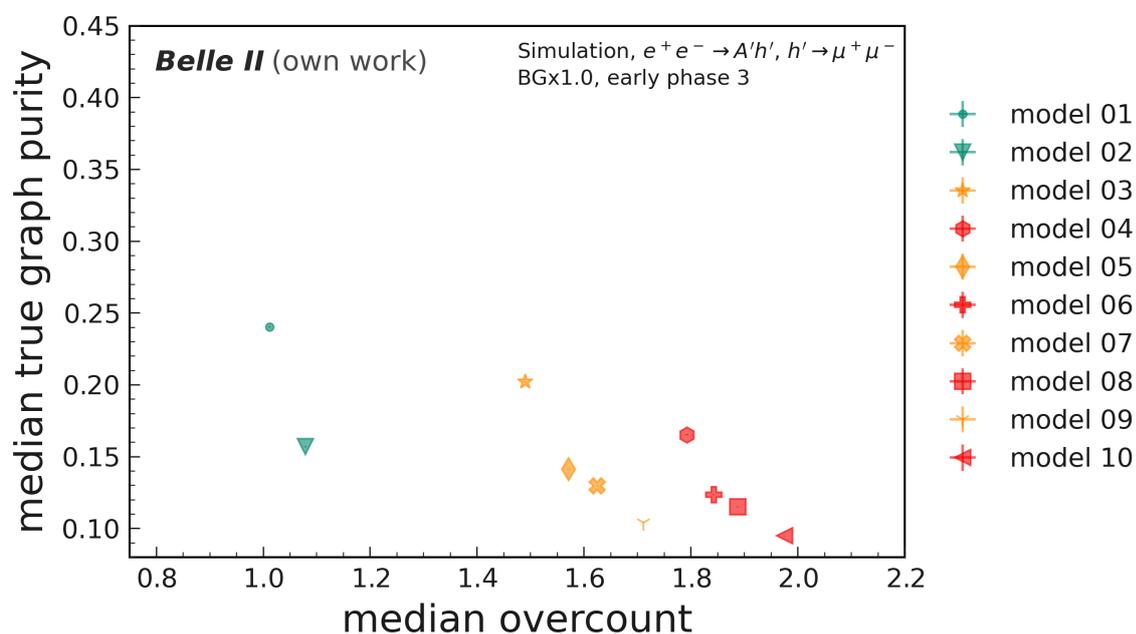


Figure B.6.: The event median of the *true graph pur.* and the *overcount*. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

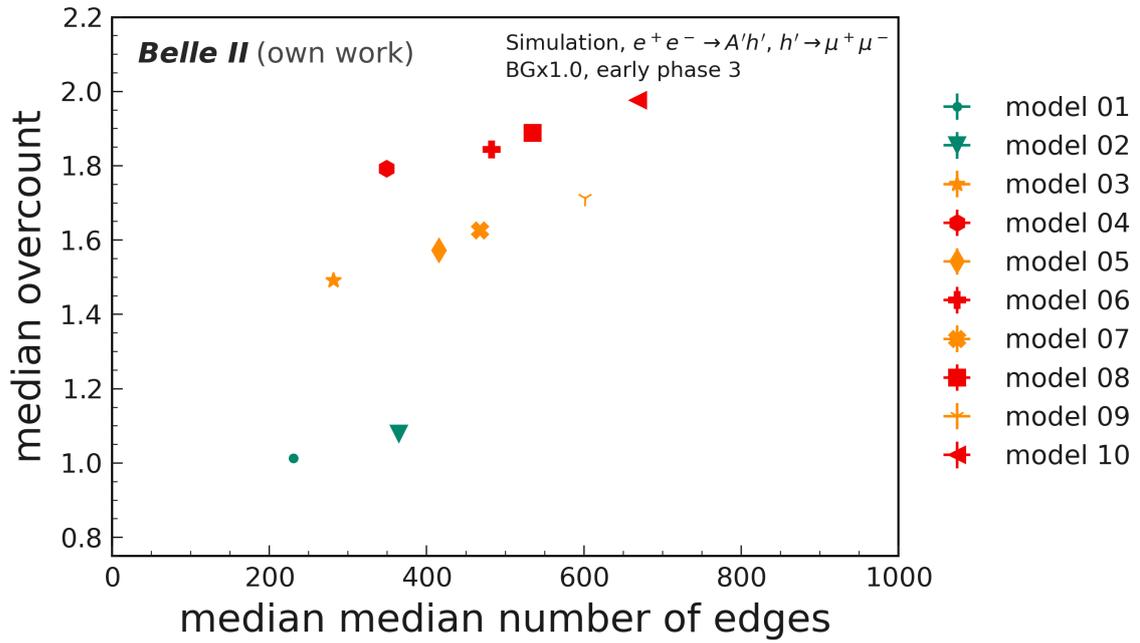


Figure B.7.: The event median of the *overcount* plotted against the median number of edges per event of the graph-building method. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .



## C. Appendix: Background Clean-Up

### C.1. Background Clean-Up on Early Phase 3 Background

This section presents information on the analysis of the IN on the IDMDH simulation with *early Phase 3* background overlays. The models were trained with a batch size of 1024.

Table C.1.: Median estimated Floating Point Operations of the ten Interaction Networks. Floating Point Operation are estimated using the PyTorch profiler with *early Phase 3* background.

Model	Total FLOPs (MFLOP)
01	$1.666 \pm 0.013$
02	$2.352 \pm 0.022$
03	$1.926 \pm 0.017$
04	$2.267 \pm 0.023$
05	$2.605 \pm 0.026$
06	$2.95 \pm 0.04$
07	$2.868 \pm 0.024$
08	$3.209 \pm 0.030$
09	$3.55 \pm 0.04$
10	$3.90 \pm 0.04$

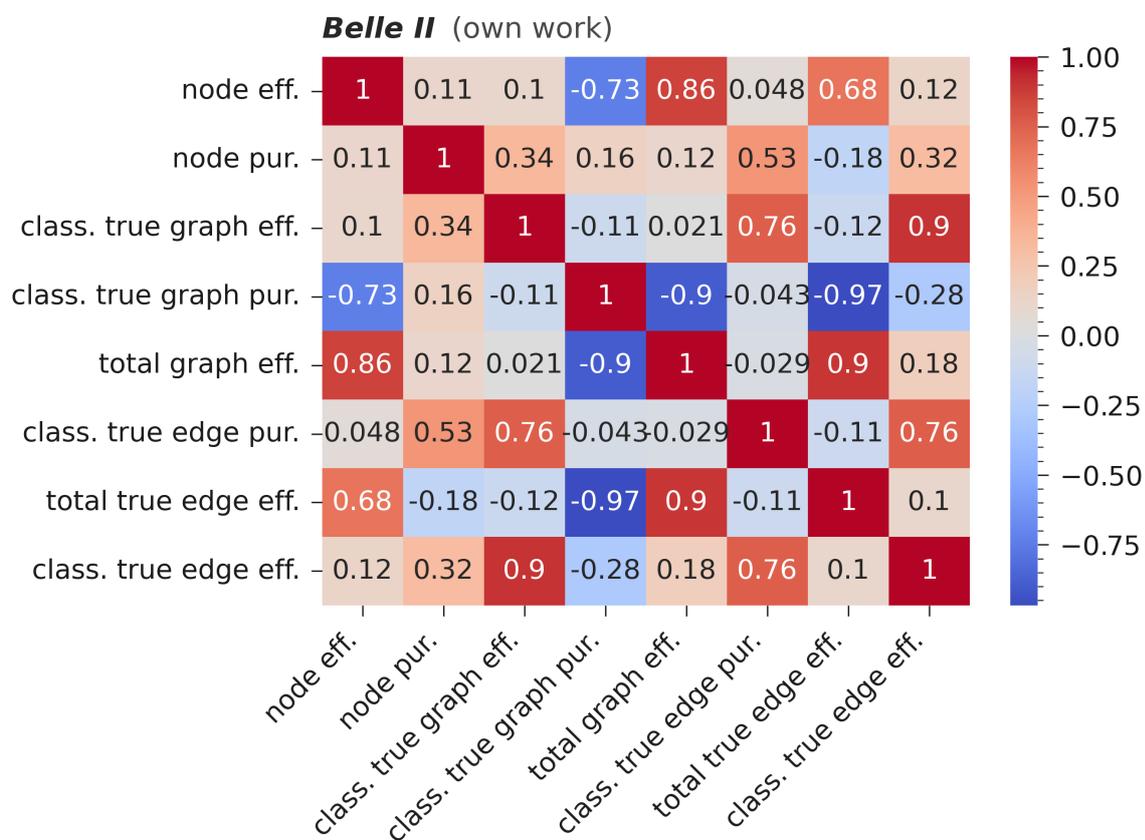


Figure C.1.: Correlation of the metrics presented in this chapter for the *early Phase 3* background with the ten analyzed graph-building models.

Table C.2.: The *true edge eff.*, *class. true edge eff.* and the *total true edge eff.* with *early Phase 3* background for models 01 to 10 with *early Phase 3* background. Values are represented as percentages.

Model	<i>true edge eff.</i> (%)	<i>class. true edge eff.</i> (%)	<i>total true edge eff.</i> (%)
01	$49.46 \pm 0.09$	$96.00 \pm 0.29$	$47.48 \pm 0.17$
02	$52.38 \pm 0.06$	$95.54 \pm 0.30$	$50.04 \pm 0.17$
03	$70.00 \pm 0.16$	$97.34 \pm 0.28$	$68.14 \pm 0.26$
04	$87.66 \pm 0.16$	$96.59 \pm 0.29$	$84.67 \pm 0.30$
05	$72.91 \pm 0.13$	$97.01 \pm 0.29$	$70.73 \pm 0.25$
06	$90.83 \pm 0.14$	$95.63 \pm 0.29$	$86.86 \pm 0.30$
07	$76.14 \pm 0.13$	$96.90 \pm 0.29$	$73.78 \pm 0.26$
08	$93.98 \pm 0.14$	$96.32 \pm 0.29$	$90.5 \pm 0.4$
09	$79.17 \pm 0.11$	$96.67 \pm 0.29$	$76.53 \pm 0.26$
10	$97.25 \pm 0.11$	$96.07 \pm 0.30$	$93.4 \pm 0.4$

Table C.3.: The *class. true graph eff.*, *true graph eff.* and the combined *total true graph eff.* for models 01 to 10 with *early Phase 3* background. Values are represented as percentages.

Model	<i>class. true graph eff.</i> (%)	<i>true graph eff.</i> (%)	<i>total true graph eff.</i> (%)
01	$96.07 \pm 0.29$	$90.12 \pm 0.10$	$87.49 \pm 0.28$
02	$96.23 \pm 0.29$	$91.67 \pm 0.07$	$88.69 \pm 0.28$
03	$96.91 \pm 0.28$	$92.31 \pm 0.10$	$90.35 \pm 0.28$
04	$96.49 \pm 0.29$	$93.99 \pm 0.10$	$91.67 \pm 0.29$
05	$96.89 \pm 0.29$	$93.62 \pm 0.07$	$91.27 \pm 0.28$
06	$95.95 \pm 0.29$	$95.35 \pm 0.07$	$92.06 \pm 0.29$
07	$96.61 \pm 0.29$	$95.12 \pm 0.05$	$92.41 \pm 0.29$
08	$96.15 \pm 0.29$	$96.83 \pm 0.04$	$93.58 \pm 0.29$
09	$96.67 \pm 0.29$	$97.115 \pm 0.018$	$94.01 \pm 0.29$
10	$96.09 \pm 0.29$	$98.667 \pm 0.013$	$94.89 \pm 0.29$

Table C.4.: Efficiency and Purity for the true nodes after the background clean-up, using the Interaction Network.

model	node efficiency (%)	node purity (%)
01	$94.6 \pm 0.4$	$94.7 \pm 0.4$
02	$96.59 \pm 0.30$	$95.8 \pm 0.4$
03	$96.1 \pm 0.4$	$95.8 \pm 0.4$
04	$96.2 \pm 0.4$	$94.6 \pm 0.4$
05	$97.83 \pm 0.30$	$94.5 \pm 0.4$
06	$97.56 \pm 0.30$	$92.9 \pm 0.4$
07	$98.21 \pm 0.28$	$95.6 \pm 0.4$
08	$98.21 \pm 0.28$	$95.3 \pm 0.4$
09	$98.31 \pm 0.28$	$95.6 \pm 0.4$
10	$98.20 \pm 0.28$	$95.8 \pm 0.4$

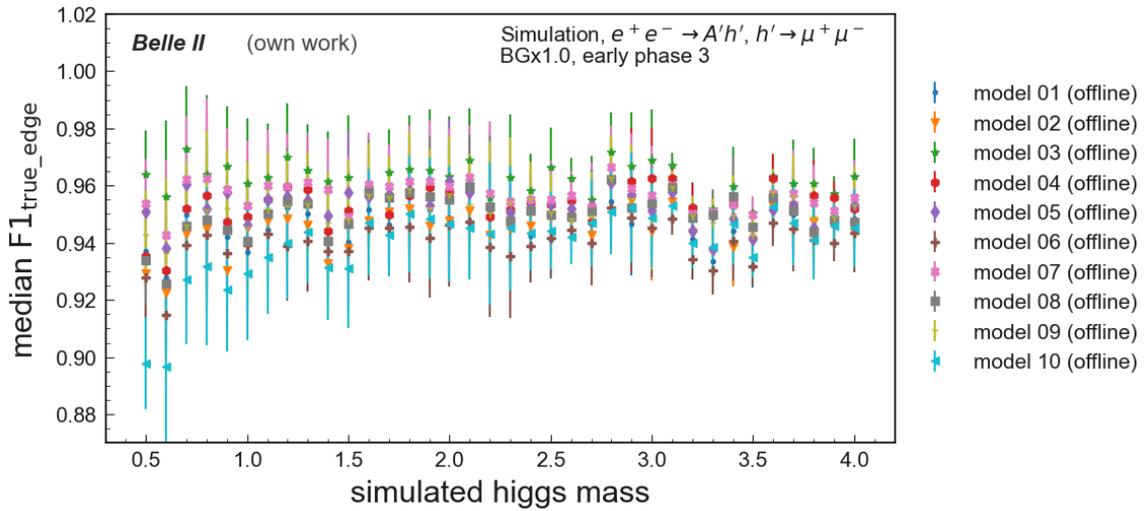


Figure C.2.: Median  $F1$  score for the true edges of the classification task for different simulated Higgs masses. The ten Interaction Networks (INs) provide consistent results across the investigated dark Higgs masses with a slight decrease for higher simulated masses.

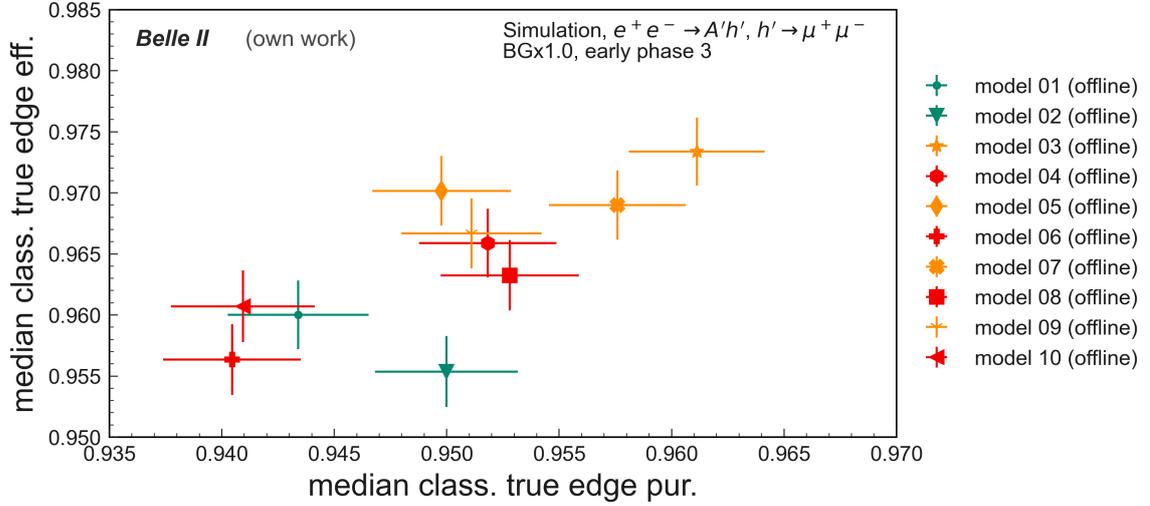


Figure C.3.: Median *class. true edge eff.* and *class. true edge pur.* of the ten investigated Interaction Networks (INs). The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

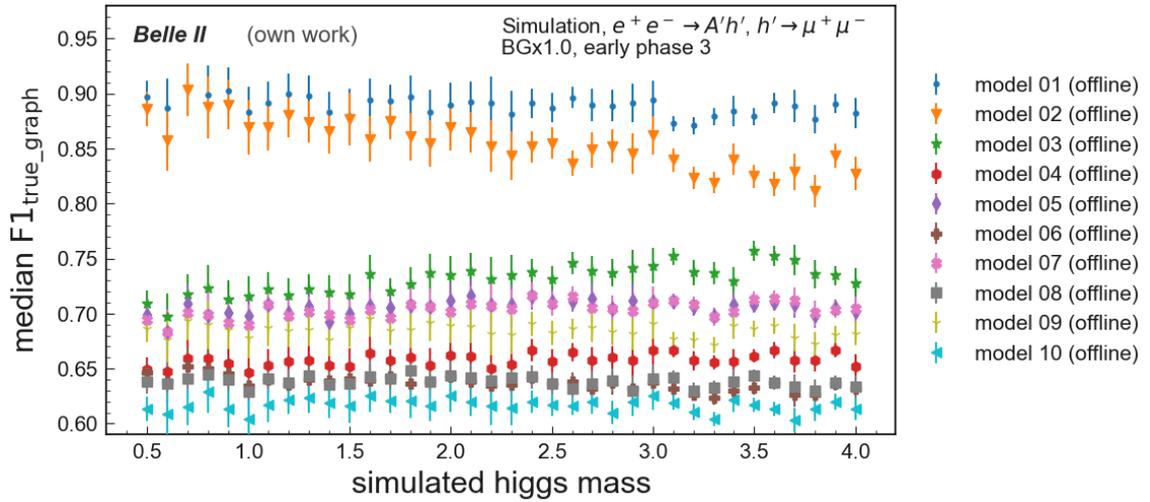


Figure C.4.: Median  $F1$  score for the true graph edges of the classification task for different simulated Higgs masses.

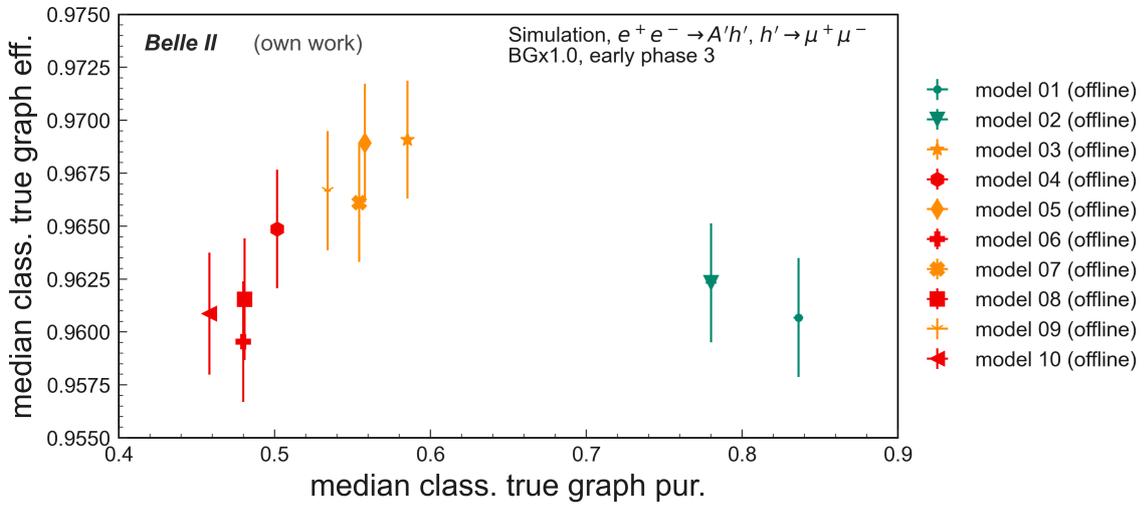


Figure C.5.: Median *class. true graph eff.* and *class. true graph pur.* of the ten investigated models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models have no allowed edges, yellow ones allow edges up to a wire distance of  $\Delta w = 1$ , and red-labeled models allow connections with a maximum wire distance of  $\Delta w = 2$ .

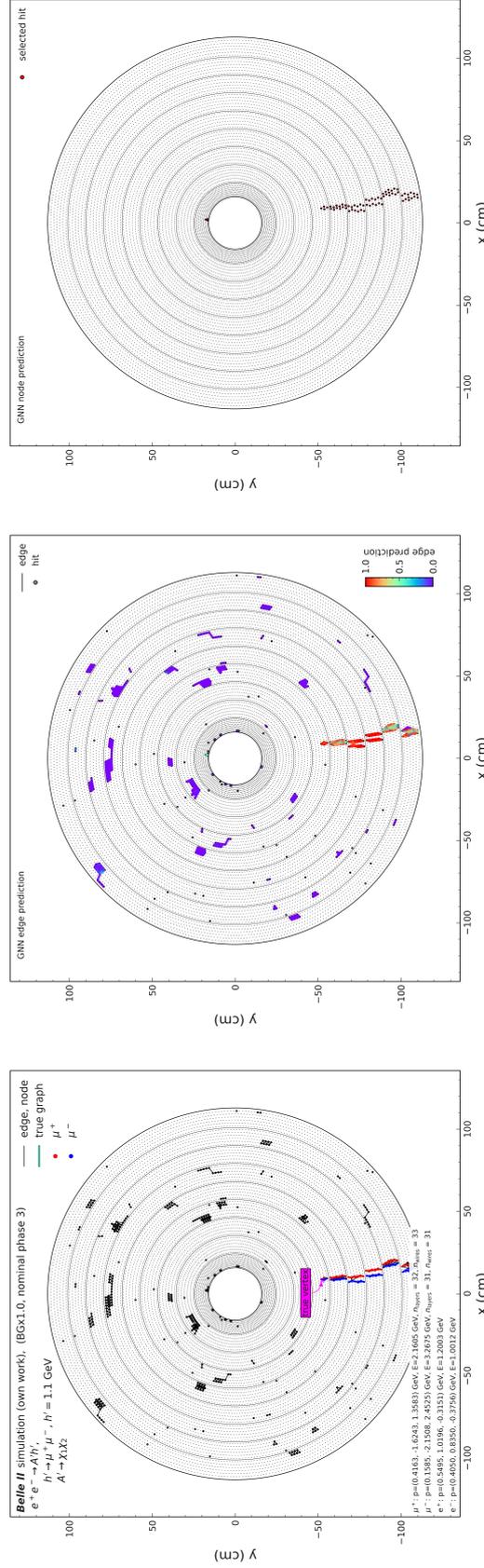


Figure C.6.: An example of the background clean-up. The event display shows all CDC hits connected in a graph using the graph-building method 08. The central picture shows the output of the Interaction Network (IN) network, the predicted edges. The picture on the right shows the final cleaned-up event with the remaining hits after threshold application and node selection.

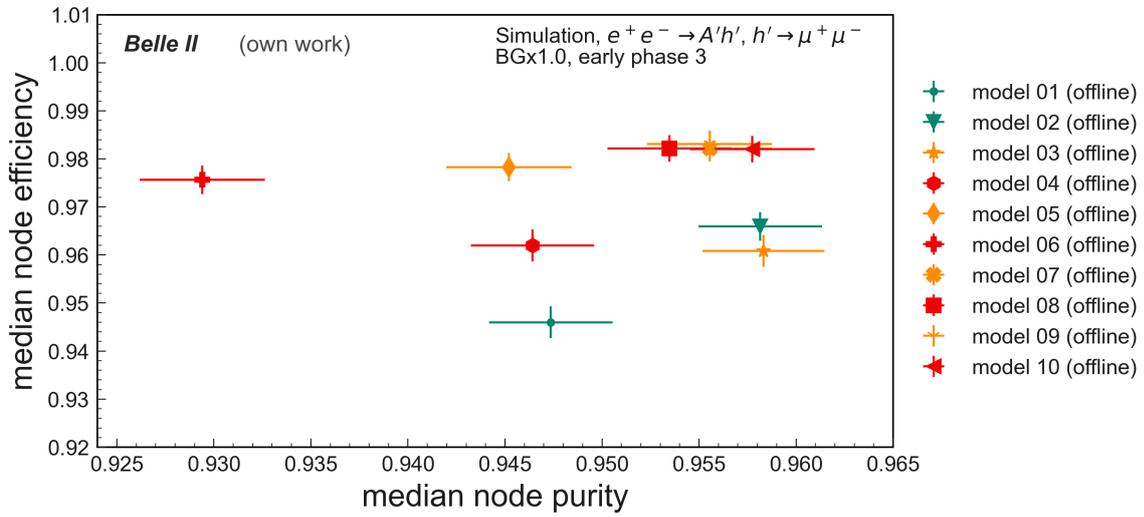


Figure C.7.: Median *node efficiency* and *node purity* of the ten investigated models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ .

## C.2. Edge Classification on Nominal Phase 3 Background With Reduced Input Features

This section presents information on analyzing the IN on the IDMDH simulation with *nominal Phase 3* background overlays. The models were trained with a batch size of 1024.

Table C.5.: Median estimated Floating Point Operations of the ten Interaction Networks. Floating Point Operation are estimated using the PyTorch profiler with *nominal Phase 3* background and reduced input feature resolution.

model	total FLOPs (MFLOP)
01	$9.588 \pm 0.020$
02	$15.45 \pm 0.04$
03	$11.519 \pm 0.026$
04	$15.20 \pm 0.04$
05	$17.31 \pm 0.05$
06	$21.02 \pm 0.06$
07	$17.35 \pm 0.04$
08	$21.05 \pm 0.05$
09	$23.26 \pm 0.06$
10	$26.84 \pm 0.07$

Table C.6.: The *true edge eff.*, and the *total true edge eff.* of the ten Interaction Networks with *nominal Phase 3* background, using reduced input resolution aligned with the anticipated Level 1 Trigger information.

Model	<i>true edge eff.</i> (%)	<i>total true edge eff.</i> (%)
01	$49.33 \pm 0.10$	$26.49 \pm 0.21$
02	$53.33 \pm 0.08$	$25.17 \pm 0.27$
03	$66.90 \pm 0.17$	$38.2 \pm 0.4$
04	$84.34 \pm 0.18$	$46.0 \pm 0.5$
05	$70.79 \pm 0.14$	$38.2 \pm 0.4$
06	$88.64 \pm 0.15$	$44.9 \pm 0.5$
07	$74.17 \pm 0.14$	$42.3 \pm 0.4$
08	$91.84 \pm 0.15$	$52.9 \pm 0.5$
09	$78.11 \pm 0.11$	$42.8 \pm 0.4$
10	$96.23 \pm 0.13$	$53.5 \pm 0.5$

Table C.7.: The *class. true graph eff.* and the *class. true graph pur.* of the ten Interaction Networks with *nominal Phase 3* background and Level 1 Trigger information.

Model	<i>class. true graph eff.</i> (%)	<i>class. true graph pur.</i> (%)
01	54.3 $\pm$ 0.4	56.4 $\pm$ 0.5
02	51.4 $\pm$ 0.5	43.1 $\pm$ 0.4
03	53.9 $\pm$ 0.5	42.01 $\pm$ 0.29
04	53.6 $\pm$ 0.5	35.59 $\pm$ 0.25
05	53.3 $\pm$ 0.5	40.17 $\pm$ 0.28
06	51.5 $\pm$ 0.5	33.75 $\pm$ 0.24
07	56.2 $\pm$ 0.5	41.02 $\pm$ 0.27
08	57.3 $\pm$ 0.5	35.20 $\pm$ 0.23
09	55.8 $\pm$ 0.5	40.57 $\pm$ 0.27
10	56.8 $\pm$ 0.5	35.01 $\pm$ 0.24

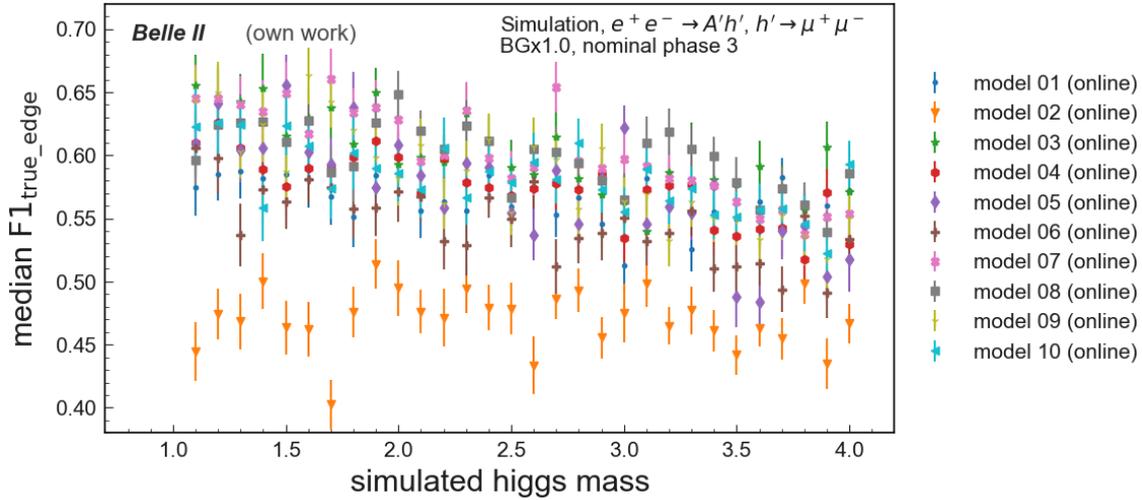


Figure C.8.: Median *F1 score* for the true edges of the classification task for different simulated Higgs masses. The ten Interaction Networks (INs) provide consistent results across the investigated dark Higgs masses with a slight decrease for higher simulated masses. With *nominal Phase 3* background and Level 1 Trigger input features.

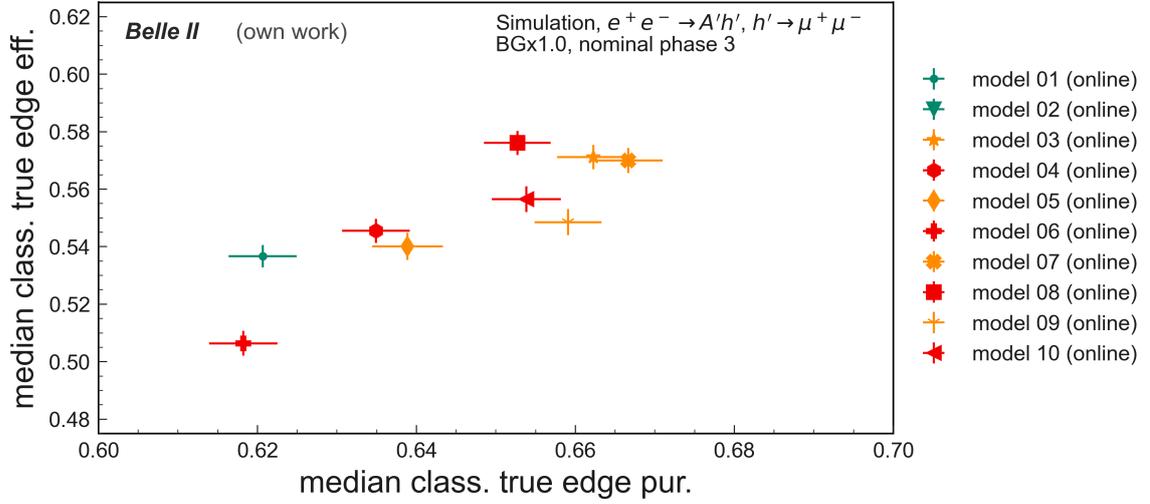


Figure C.9.: Median *class. true edge eff.* and *class. true edge pur.* of the ten investigated Interaction Networks (INs). The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models do not permit edges, and yellow models allow edges with a wire distance of up to  $\Delta w = 1$ . Red-labeled models permit connections with a maximum wire distance of  $\Delta w = 2$ . With *nominal Phase 3* background and Level 1 Trigger input features.

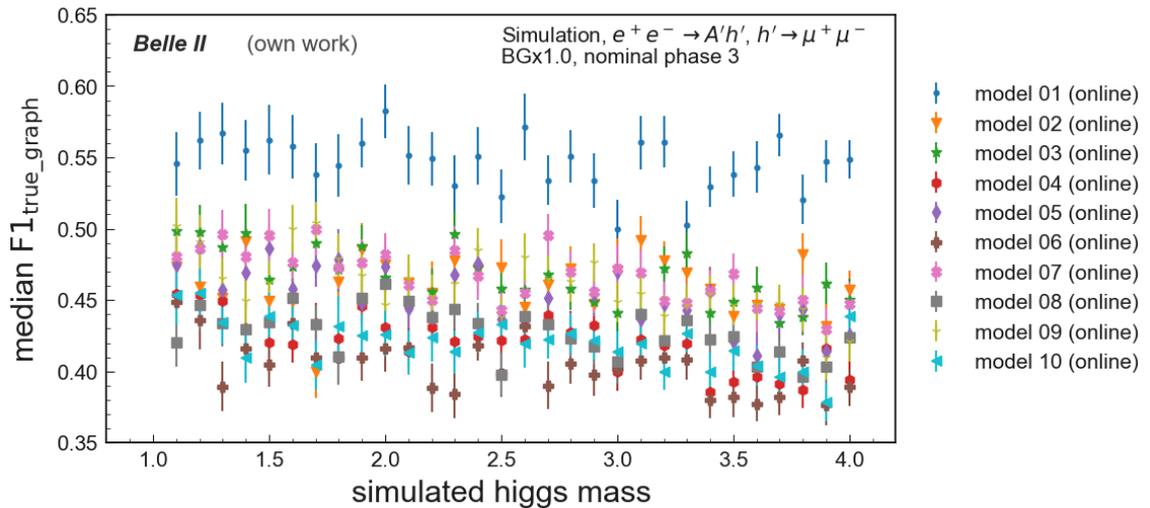


Figure C.10.: Median  $F1$  score for the true graph edges of the classification task for different simulated Higgs masses. With *nominal Phase 3* background and Level 1 Trigger input features.

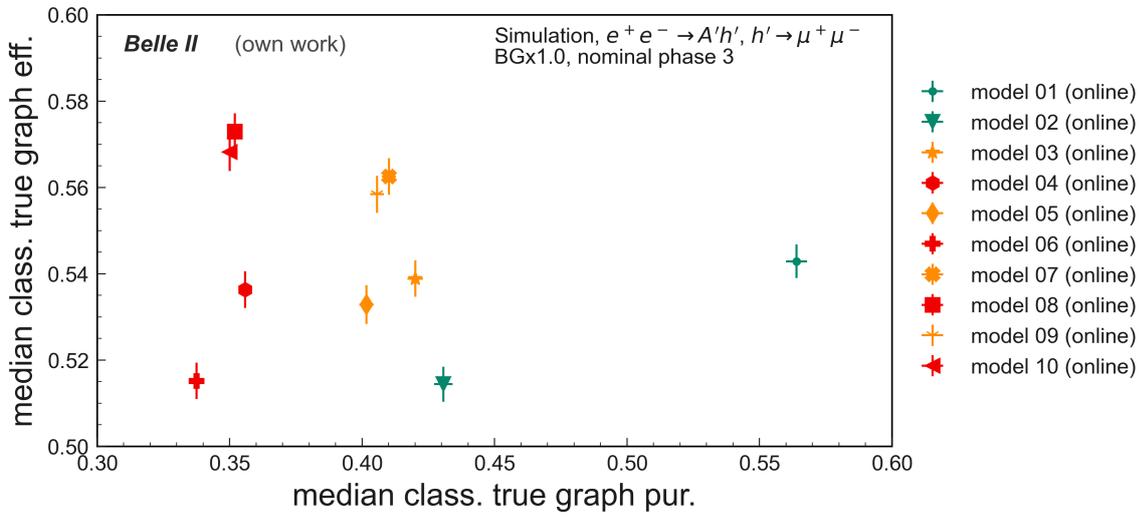


Figure C.11.: Median *class. true graph eff.* and *class. true graph pur.* of the ten investigated models. The models are color-coded based on the allowed connection within a layer distance of  $\Delta l = 2$ . Green models have no allowed edges, yellow ones allow edges up to a wire distance of  $\Delta w = 1$ , and red-labeled models allow connections with a maximum wire distance of  $\Delta w = 2$ . With *nominal Phase 3* background and Level 1 Trigger input features.