

Emerging Jets Search, Triton Server Deployment, and Track Quality Development

Machine Learning Applications in High Energy Physics

by

Claire Savard

M.S., University of Colorado – Boulder, 2020

B.S., University of Michigan – Ann Arbor, 2018

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Physics
2024

Committee Members:

Kevin Stenson, Chair

Keith Ulmer

Eric Zimmerman

Oliver DeWolfe

Tamara Lehman



Savard, Claire (Ph.D., Physics)

Emerging Jets Search, Triton Server Deployment, and Track Quality Development: Machine Learning
Applications in High Energy Physics

Thesis directed by Prof. Kevin Stenson

Machine learning is becoming prevalent in high energy physics, with numerous applications in physics analyses and event reconstruction showing great improvements compared to traditional computing methods. This thesis studies three projects which each propose new avenues for machine learning applications within the high energy physics CMS experiment located at CERN. In the first project, a search for a dark matter signal called “emerging jets” is performed, using graph neural networks to greatly increase sensitivity to the signal’s signature within the data. The result of this dark matter search sets the most stringent exclusion limits to date on theoretical emerging jet models. Motivated by inefficiencies encountered when processing the emerging jet graph neural network at Fermi National Accelerator Laboratory’s computing centers, the second project re-optimizes the computing centers for machine learning inference. This re-optimization uses NVIDIA Triton Inference Servers to process users’ analysis code heterogeneously, therefore achieving high processing throughput and decreasing user time-to-insight. The last project focuses on an upgrade to the CMS experiment’s real-time event selection system which improves physics object reconstruction under harsh processing conditions. A boosted decision tree is used to quickly and efficiently quantify a reconstructed particle’s “track quality” in order to remove particle tracks reconstructed erroneously. In summary, this thesis will not only present examples of how high energy physics can greatly benefit by leveraging machine learning techniques for physics analysis and reconstruction, but will also provide guidance on how the field can prepare for the inevitable increase in machine learning applications.

Dedication

To all the STEM enthusiasts who come after me.

Acknowledgements

This work would not be possible without some key people, both in my professional and personal life, who have kept me motivated and on track.

I must first thank my advisor, Prof. Kevin Stenson, who found out very quickly how passionate and determined I am in much more than physics. He supported my pursuit of a computer science masters in parallel to my physics Ph.D. which has not only given me the expertise to contribute to some of the coolest computational problems in high-energy physics, but will also be invaluable in my future career. He also understood my passion for diversity, equity, and inclusion, and science outreach efforts and was supportive of the time I put aside for that. I would not have had the same fulfilling Ph.D. experience if it were not for Kevin.

I would next like to thank those collaborators who have had a profound impact on this work. To the Colorado CMS team – Kevin, Keith, Bill, Alexx, George, Nick, Jannicke, Emily and Noah – I apologize for my contributions in making group meetings way too long. To the emerging jets team – Yi-Mu, Long, Kevin, Sarah, Jannicke, Guillermo, Alexx, and Scarlett – working with you made the CMS review process slightly less painful. To the triton team – Burt, Lindsey, and Nick – without all of your timely responses, my graph neural network’s processing would have been unmanageable. And to my track quality partner – Chris – one of us needed to work on the hardware, and I’m glad it wasn’t me. And to you all, I appreciate you answering all of my questions thoroughly despite my stubbornness.

Living in Boulder has been an amazing experience, but even better has been sharing wonderful moments in and around this city with some of the best people. To Giaco, Alex, Arnulf, Grace, Ariel,

and many more, I hope we continue to adventure and board-game together for many years to come. To my CERN friends that I will never forget, especially Megan, I hope to see you back on the dance floor ASAP. And to everyone else who has been cheering me on behind the scenes, from my extended family to my childhood friends, I feel it and appreciate it more than know.

And of course, I must acknowledge my family for whom I am incredibly grateful. My parents – Guy and Carole – have always made me feel capable in all that I do. My brothers – Nicolas and Emile – make life fun and silly, and keep me laughing even though they claim I don't have a sense of humor. And I have been extremely lucky to have spent most of my Ph.D. years living within a 5 minute walk of my sister – Léa – who has become my staple weekend adventure partner and the first person I turn to for whatever I may need. You are all the best kind of people, and I hope to give back the joy you all bring to me.

Lastly, I would like to thank my partner, Dalton, who suffered and celebrated with me every step of way. Thank you for being everything that you are, and for loving everything that I am. One of the best parts of my Ph.D. was meeting you.

Contents

Chapter

1	Introduction	1
2	The Standard Model	3
2.1	Introduction	3
2.2	Fundamental Particles	3
2.3	Mathematical Formulation	7
2.3.1	Quantum Electrodynamics	8
2.3.2	Quantum Chromodynamics	9
2.3.3	Electroweak Theory	10
2.3.4	Higgs Mechanism	11
2.4	Dark Matter	12
2.4.1	Popular Theories	13
2.4.2	Dark Sector	14
3	The CMS experiment	16
3.1	The Large Hadron Collider	16
3.2	The CMS Detector	20
3.2.1	Silicon Tracker	23
3.2.2	Calorimeters	25
3.2.3	Superconducting Magnet	28

3.2.4	Muon System	28
3.2.5	Trigger System	30
3.3	Event Reconstruction	31
3.3.1	Fundamental Elements	32
3.3.2	Particle Flow Candidates	35
3.3.3	High-Level Objects	36
4	Emerging Jets Analysis	39
4.1	Introduction	39
4.2	Signal Model	40
4.2.1	Unflavored Down Scenario	41
4.2.2	Flavor-Aligned Down Scenario	42
4.3	Signal and Background Data Samples	44
4.4	Signal Event Selection	47
4.4.1	Physics Object Reconstruction	47
4.4.2	Emerging Jet Tagger	49
4.4.3	Event-Level Variables	63
4.4.4	Selection Criteria Optimization	63
4.4.5	Uncertainties	71
4.5	Background Estimation	72
4.5.1	Estimation Calculation	73
4.5.2	Uncertainties	80
4.5.3	Simulation Closure Tests	82
4.5.4	Data Closure Tests	83
4.6	Results	85
4.6.1	GNN Exclusion Results	88
4.6.2	Comparisons to Cut-Based Exclusion Results	89

5	Triton Server Deployment at Fermilab Computing Facilities	94
5.1	Preface	94
5.2	Introduction	95
5.3	Background	96
5.3.1	Shared computing facilities	96
5.3.2	Common machine learning processors	96
5.3.3	NVIDIA Triton Inference Server	98
5.4	Fermilab Triton Server Implementation	99
5.4.1	Computing facility statistics	99
5.4.2	Typical user workflow	99
5.4.3	Triton server implementation	101
5.5	Benchmarking Tests	104
5.5.1	Timing comparison	104
5.5.2	Increasing workers	106
5.5.3	Multi-model scaling	108
5.6	Limitations	110
5.7	Conclusion	111
6	Level-1 Track Quality Development	113
6.1	Introduction	113
6.2	High Luminosity LHC	114
6.3	Level-1 Trigger Upgrade	114
6.3.1	Subsystem upgrades	115
6.3.2	Architecture	117
6.4	Level-1 Tracker Tracks	118
6.4.1	Reconstruction	119
6.4.2	Use-Cases	120

6.4.3	Fake Tracks	122
6.5	Track Quality Classification	123
6.5.1	Current Method	123
6.5.2	Machine Learning Development	124
6.5.3	Hardware Implementation	127
6.6	Applications	129
6.6.1	Primary Vertex	129
6.6.2	Track Jets	131
6.6.3	Track H_T	133
6.7	Displaced Track Quality	134
6.7.1	Extended Tracking	134
6.7.2	Preliminary Displaced Track Quality Classifier	136
6.8	Final remarks	138
7	Conclusion	139
	References	141
 Appendix		
A	Emerging Jet Specifics	149
A.1	Impact Parameter Transformation	149
A.2	GNN Tagger Generalizability	152
A.3	Cut Based Flavor-Aligned Results	153
B	Triton Server Specifics	155
B.1	Triton server parameters	155
B.2	ParticleNet demo model parameters	158

Tables

Table

2.1	Gauge boson properties	5
2.2	Quark properties	5
2.3	Lepton properties	6
4.1	Unflavored coupling scenario parameters	45
4.2	Flavor-aligned coupling scenario parameters	46
4.3	GNN training parameters	58
4.4	Unflavored cut-based final cutsets	68
4.5	Flavor-aligned cut-based final cutsets	68
4.6	Unflavored and flavor-aligned GNN final cutsets	68
4.7	Signal systematic uncertainties summary	73
4.8	Background uncertainties	82
4.9	Final background estimation results	86
6.1	List of track features from Kalman filter	121
6.2	Standard L1 track cuts	121
6.3	E_T^{miss} track quality cuts	124
6.4	Simulated resources of both TQ ML models on an FPGA	127
6.5	Outdated track cuts applied for track jets	131
6.6	Updated track cuts applied for track jets	131

B.1 Triton server scaling parameters	156
--	-----

Figures

Figure

2.1	Standard Model of particle physics	4
2.2	Dark matter theories map	13
3.1	CERN accelerator complex	17
3.2	CMS Run 2 number of interactions per bunch crossing	18
3.3	CMS Run 2 integrated luminosity	19
3.4	CMS detector	20
3.5	CMS detector slice	21
3.6	CMS detector coordinate system	22
3.7	CMS silicon tracker	24
3.8	CMS electromagnetic calorimeter	26
3.9	CMS hadron calorimeter	27
3.10	Different CMS muon chamber types	29
3.11	Track reconstruction efficiencies	34
3.12	Particle jet example	37
4.1	Dark mediator pair production Feynman diagrams	40
4.2	Emerging jet signal produced from proton-proton collisions	41
4.3	Emerging jet signature in CMS detector	41
4.4	Dark pion lifetime as a function of dark pion mass and quark composition	43

4.5	z residual of primary vertices	48
4.6	Prompt track fraction of events with poorly and non-poorly reconstructed vertices . .	48
4.7	Median d_{xy} of emerging jets versus SM jets	50
4.8	α_{3D} of emerging jets versus SM jets	50
4.9	Number of displaced tracks	51
4.10	Track girth	51
4.11	Graph representation of a jet object	52
4.12	GNN coordinates	54
4.13	GNN input features	56
4.14	Correlation matrix of GNN inputs	57
4.15	GNN validation in training	59
4.16	GNN score distributions	60
4.17	GNN ROC curves	61
4.18	GNN permutation feature importance	62
4.19	GNN ROC curves compared to cut-based tagging performance	63
4.20	Event-level selection variable distributions	64
4.21	Sum of squared distances of k -means clustering GNN optimal cutsets	66
4.22	k -means clustering of cutsets visualization	67
4.23	Unflavored model signal selection efficiency	69
4.24	Flavor-aligned model signal selection efficiency	70
4.25	SR, CR, and FR region comparison	76
4.26	Mistag rate for simulated background jets along p_T and jet flavor	77
4.27	Jet-flavored mistag rates in data versus FR simulation	79
4.28	Background estimation closure test in simulation	83
4.29	Background estimation closure test in data	84
4.30	Final background estimation results	87
4.31	Example of CL_s upper limit cross section results	87

4.32	Example of exclusion results	87
4.33	Unflavored GNN method final results	89
4.34	Flavor-aligned GNN method final results	90
4.35	Unflavored cut-based method final results	91
4.36	Scaling data for cut-based unflavored $m_{\pi_{dark}} = 10$ GeV exclusion limit	92
4.37	Scaling data for cut-based flavor-aligned $m_{\pi_{dark}} = 10$ GeV exclusion limit	92
5.1	Typical use workflow at the LPC	100
5.2	EAF Triton server implementation schematic	101
5.3	Inference rate vs. Triton instances for different scaling parameters	103
5.4	Timing of Triton model vs. local CPU model	104
5.5	Timing test on ResNet50	105
5.6	Timing test on a small BDT	105
5.7	Number of Triton instances vs. number of workers	107
5.8	Queue time per request vs. number of workers	107
5.9	Throughput vs number of workers	108
5.10	Throughput vs. number of background models using instance sharing	109
5.11	Throughput vs. number of background models using unique instances	110
6.1	Current L1T schematic	115
6.2	Phase-2 L1T schematic	115
6.3	Phase-2 L1T physics objects locations	117
6.4	Proposed Phase-2 L1T architecture	118
6.5	Projection of L1 track seeds to other stubs to form a track	119
6.6	Fake vs. real tracks example	123
6.7	Performance of TQ BDT vs. NN vs. E_T^{miss} cuts	126
6.8	BDT performance on real tracks vs. p_T	126
6.9	TQ BDT bins and their performance on different track types	129

6.10	z_0^{PV} residual for different fake track rejection techniques	130
6.11	z_0^{PV} efficiency for different fake track rejection techniques	130
6.12	Track jet finding efficiency vs. true jet p_T for different track selection cuts	132
6.13	Track H_T rate vs. H_T threshold for various track selection cuts	134
6.14	Track H_T trigger efficiency vs. true H_T for various track selection cuts	134
6.15	Tracking efficiency vs. true particle d_0 on displaced muons	135
6.16	Displaced track d_0 distribution before and after weighting	137
6.17	Displaced and prompt TQ BDT ROC curve on displaced tracks	137
6.18	Displaced and prompt BDT performance on real tracks vs. d_0	137
A.1	Impact parameter distributions	150
A.2	Transformed impact parameter distributions	150
A.3	Scaled impact parameter distributions	150
A.4	Transformed vs scaled impact parameter GNN performance comparison	151
A.5	GNN tagger performance generalizability	152
A.6	Flavor-aligned cut based method final results	154
B.1	Scale-up queue time instability example	158
B.2	Throughput during chunk processing instability example	158

Chapter 1

Introduction

In the past decade, physicists have become increasingly interested in the use of machine learning techniques for fundamental research. In high energy physics in particular, many machine learning algorithms applied to event identification and reconstruction have proven to be quite successful. While valid questions exist regarding the appropriate use-cases of such complex techniques, there is no question that many existing applications show marvelous performance increases compared to standard computing techniques. In this thesis, I propose new ways in which machine learning can be applied in particle physics and show how these techniques can improve physics research beyond traditional methods.

High energy physics experiments, such as the CMS experiment at the CERN accelerator facility, study the fundamental particles of the universe. Chapter 2 introduces the current Standard Model of particle physics and its limitations, and Chapter 3 describes the CMS experiment and how it can be used to study the Standard Model and beyond-standard-model theories. In brief, proton bunches are collided at nearly the speed of light within the center of the CMS detector, and these explosive collisions produce a medley of particles which pass through and potentially interact with the detector. These detector interactions are then used to reconstruct and study the full physical process produced by the collisions.

In Chapter 4, I use CMS proton-proton collision data to search for dark matter which manifests in the detector as “emerging jets”. The theory which gives rise to emerging jets is a hidden sector dark quantum chromodynamics model with a heavy dark mediator particle that allows interactions

to occur between the hidden and standard model sectors. I develop graph neural networks to search for these unique dark matter signatures within CMS data, which greatly increases signal sensitivity from more traditional computing algorithms. Comparisons between the graph neural networks and a traditional search algorithm are made to show the ultimate gain that a machine learning technique such as this can provide to physics analyses.

Motivated by computational inefficiencies uncovered during the emerging jets analysis, Chapter 5 focuses on re-optimizing the high energy physics computing facility located at Fermi National Accelerator Laboratory for machine learning inference. Computing facilities used for processing physics analyses are generally constrained to run code on one processor at a time. Analyses using machine learning models, however, benefit greatly from heterogeneous computing, which can be achieved using an NVIDIA Triton Inference Server to offload machine learning inference from CPUs to GPUs. This new computing ecosystem increases data processing throughput while ensuring highly coveted GPU resources are shared amongst all users.

With upgrades to the CERN accelerator facility forthcoming, which will increase CMS data rates and data complexity, the initial real-time event selection component of the detector (Level-1 Trigger) is being improved in order to maintain – and hopefully increase – current physics reach. One major improvement to the trigger is the reconstruction of charged particle tracks which can be used to build useful physics objects. Chapter 6 discusses the development of a new variable that describes the quality of these reconstructed tracks, called “track quality”. I use machine learning to calculate the track quality, which can be used to increase the reconstruction performance of many physics objects built from these tracks.

In brief, this thesis shows examples of when and how machine learning techniques can be applied in high energy physics to advance fundamental research, as well as how high energy physics should prepare for the inevitable increase in machine learning applications throughout the field. I hope to inspire other high energy physicists to explore the enticing prospects machine learning can deliver. Happy reading!

Chapter 2

The Standard Model

2.1 Introduction

The Standard Model (SM) is currently the best theoretical framework for describing the fundamental particles of the universe and their interactions. However, the SM is not yet complete. For example, out of the four fundamental forces that are known to exist, the SM explains all but gravity, which interacts with all things that have mass. In addition to this theoretical limitation of the SM, it also does not describe a widely accepted phenomena known as dark matter, or matter which does not seemingly interact with any fundamental particles except through gravity, making it very hard to study. This chapter will give an overview of the SM, followed by a dark matter discussion and the introduction of a theory that will motivate the emerging jets search in Chapter 4.

2.2 Fundamental Particles

The current SM consists of 17 particles characterized by their intrinsic properties. In brief, the fundamental particles are split by their spin¹ into fermions – the building blocks of matter with half-integer spin – and bosons – the force carriers with whole integer spin. The forces mediated by the bosons are the strong force, weak force, and electromagnetic force, and only certain fermions are influenced by certain forces. Figure 2.1 shows a concise diagram of the SM with the different particles types and properties, which will be helpful to refer back to as each particle is described in

¹Spin is related to the angular momentum of the particle and is used to distinguish fermions from bosons as particles with half versus whole integer spin are seen to have different behavior. More on spin can be found at [1].

Standard Model of Elementary Particles

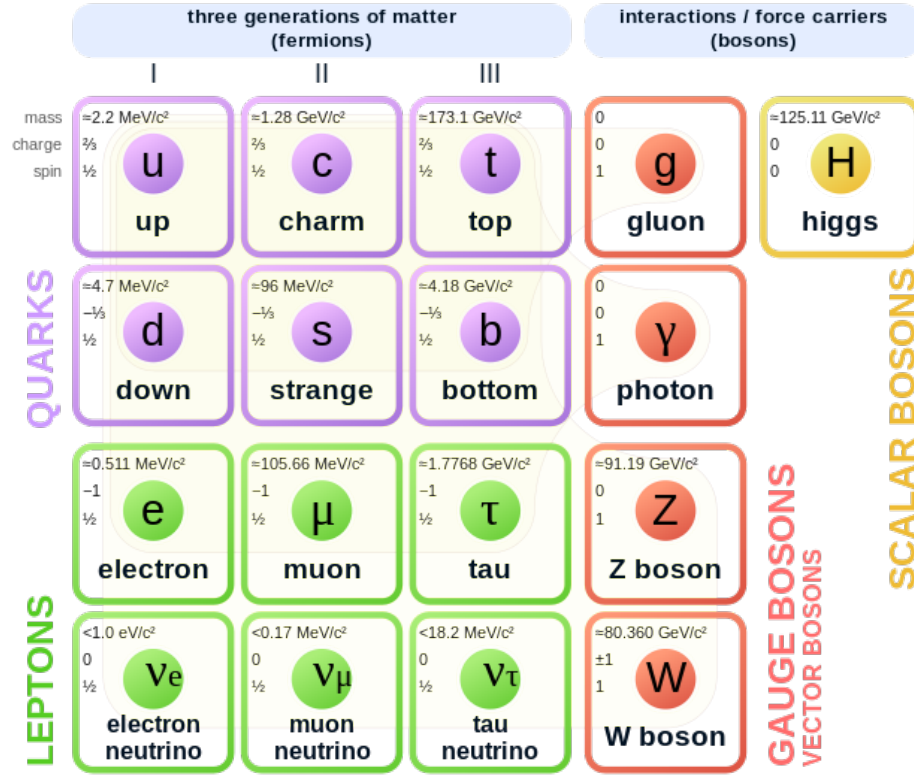


Figure 2.1: The Standard Model of particles physics [2].

further detail below.

Each of the four gauge bosons – the gluon, photon, Z , and W bosons – mediates a specific fundamental force, meaning that two particles interacting via a certain force do so by exchanging a boson associated with that force. The gluon mediates the strong force and has no mass or (electric) charge. A gluon is capable of interacting with any particle that has a color charge (including itself), which does not refer to a physical color but is an analog to how an electric charge is required for the electromagnetic force. The photon mediates the electromagnetic force and also has no mass or charge. The Z boson mediates the weak force and has a mass² of $\sim 91 \text{ GeV}$ and no charge. The W boson also mediates the weak force and has a mass of $\sim 80 \text{ GeV}$ and a charge of ± 1 . Each of these

²A standard choice of units (called “natural units”) in high energy physics is to set the speed of light $c = 1$ such that mass, momentum, and energy are all described in units of energy (generally eV).

Name	Symbol	Force	Charge	Mass [GeV]
gluon	g	strong	0	0
photon	γ	electromagnetic	0	0
W boson	W^{\pm}	weak	± 1	80.38
Z boson	Z	weak	0	91.19

Table 2.1: A list of gauge bosons and their properties. Symbol refers to the character generally used in literature to refer to the boson, and force refers to the force that the boson mediates (or “carries”). All of these particles have a spin of 1.

Name	Symbol	Generation	Charge	Mass
up	u	1	$+2/3$	2.16 MeV
down	b	1	$-1/3$	4.67 MeV
charm	c	2	$+2/3$	1.27 GeV
strange	s	2	$-1/3$	93.4 MeV
top	t	3	$+2/3$	172.7 GeV
bottom	b	3	$-1/3$	4.18 GeV

Table 2.2: A list of quarks and their properties. Symbol refers to the character generally used in literature to refer to the quark, and generation refers to the mass ordering of the quark pairs. All of these particles have a spin of $1/2$.

bosons have spin 1. A list of the gauge bosons and their properties can be seen in Table 2.1.

Quarks are a sub-category of fermions that each have a color charge (either blue, green, or red), and therefore interact with the strong force via gluons. Quarks can be bound together by gluons to form colorless hadrons (a composite particle made up of two or more quarks) like protons and neutrons. The most common quarks found in nuclei are the up and down quarks, which make up the first, and lightest, generation of quarks. The up quark has a charge of $+2/3$ and mass of 2.16 MeV, while the down quark has a charge of $-1/3$ and mass of 4.67 MeV. The second and third quark generations – charm, strange, top and bottom quarks – also consist of pairs of $+2/3$ and $-1/3$ charged quarks with larger masses. All of the quarks and their properties can be found in Table 2.2.

Quarks can also interact via the electromagnetic or weak force.

Name	Symbol	Generation	Charge	Mass
electron	e	1	-1	0.511 MeV
electron neutrino	ν_e	1	0	< 0.8 eV
muon	μ	2	-1	105.66 MeV
muon neutrino	ν_μ	2	0	< 0.8 eV
tau	τ	3	-1	1.777 GeV
tau neutrino	ν_τ	3	0	< 0.8 eV

Table 2.3: A list of quarks and their properties. Symbol refers to the character generally used in literature to refer to the lepton, and generation refers to the mass ordering of the charged leptons. All of these particles have a spin of $1/2$.

The remaining fermions that are colorless are called leptons. Leptons have either a charge of -1 , indicating that they interact with the electromagnetic force, or are neutral. Charged leptons consist of electrons, with a mass of 511 keV, and their heavier siblings, muons and taus. Since electrons are the lightest of the trio, they are the most abundant in nature, just like the up and down quarks. Each charged lepton is also paired with a unique neutral particle, called the neutrino. Neutrinos have a unique property called “oscillation” which allows them to change into different neutrino types while travelling. Since neutrinos do not interact via the electromagnetic force, they can be difficult to study. Because of this, many of their properties are still unknown, such as their mass.³ Pairs of charged and neutral leptons are organized into generations like the quarks, as can be seen in Table 2.3 along with other lepton properties. All leptons interact via the weak force.

The only particle left to discuss is the Higgs boson. This is a scalar boson with spin 0, no charge, and a mass of ~ 125.25 GeV. Unlike the gauge bosons, this scalar boson is not associated with one of the fundamental forces. Instead, the Higgs boson is associated to mass through the Higgs field, which is an energy field that interacts with particles to give them mass. Since gluons and photons are massless, they are the only particles that do not interact with the Higgs field. The weak force is the only fundamental force which influences the Higgs boson.

³Although the neutrino masses are still unknown, neutrino experiments have been able to set upper limits on their mass based on experimental observations [3].

Each particle in the SM is also associated with a corresponding antiparticle which has the same properties and mass, but has an opposite color or electric charge. For example, an anti-electron has an electric charge of $+1$, and an anti-top quark has an electric charge of $-2/3$ and a color charge of anti-red, anti-green, or anti-blue. Particles without any form of charge, like photons, are considered to be their own antiparticles. If particles come in contact with their corresponding antiparticles, they will annihilate to produce photons.

2.3 Mathematical Formulation

The SM is formulated using quantum field theory, which describes all fundamental particles and forces as fields that interact with one another. In this theory, the SM has gauge symmetry $SU(3) \times SU(2) \times U(1)$. The $SU(3)$ term describes the strong interaction, which is explained by quantum chromodynamics (QCD). The remaining two terms $SU(2) \times U(1)$ describe the electromagnetic and weak interactions, described by quantum electrodynamics (QED) and electroweak theory. These symmetries lead to the SM Lagrangian which can be compactly written as

$$\begin{aligned} \mathcal{L}_{SM} = & -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\bar{\psi}\gamma^\mu D_\mu\psi \\ & + \bar{\psi}_i y_{ij} \psi_j \phi + |D_\mu\phi|^2 - V(\phi) \\ & + h.c., \end{aligned} \tag{2.1}$$

where $F_{\mu\nu}$ is the field strength tensor of the gauge fields, ψ are the fermion fields, γ_μ are Dirac matrices, D_μ is the gauge covariant derivative, ϕ is the Higgs field, and $h.c.$ are the hermitian conjugates of all terms. In brief, the first line accounts for the strength of the strong, electromagnetic, and weak forces and how they act on the particles, and the second line describes how the particles get their mass. The following sections will outline how \mathcal{L}_{SM} is built from the QED, QCD, electroweak, and Higgs theories.⁴

⁴A much more in depth derivation of the SM Lagrangian can be found at [4]

2.3.1 Quantum Electrodynamics

QED is the theory of how the electromagnetic force interacts with all charged fermions through the exchange of photons. This is represented by a $U(1)$ gauge symmetry where 1 indicates that there is only one type of electric charge.

The spin-1/2 fermion field ψ can be described using the Dirac Lagrangian

$$\mathcal{L}_{fermion} = \bar{\psi}(i\gamma^\mu\partial_\mu - m)\psi, \quad (2.2)$$

where γ_μ are Dirac matrices [5], ∂_μ are partial derivatives, and m is the mass of the fermion. With the addition of a mass term, this equation is not gauge invariant. Therefore, the mass term will be dropped for the moment and will be returned to when describing the Higgs mechanism in Section 2.3.4. The remaining Lagrangian is invariant under a $U(1)$ global gauge transformation, which geometrically corresponds to a circle which is invariant to rotations such that $\psi(x) \rightarrow e^{iq\alpha}\psi(x)$, where q is the electric charge and e and α are constants. However, this Lagrangian is not invariant under a local gauge transformation where $\alpha = \alpha(\vec{x})$ is a function of space and time.

The introduction of a spin-1 vector field A_μ (the photon) to create a covariant derivative that swaps in for the partial derivative in Eqn. 2.2 fixes local gauge invariance, and is defined as

$$D_\mu = \partial_\mu + ieA_\mu, \quad (2.3)$$

where $A_\mu \rightarrow A_\mu - \partial_\mu\alpha(x)$ is also gauge invariant. The addition of a ieA_μ term allows the fermions to interact with the photon with coupling constant e , equal to the electric charge. The kinetic term of the vector field must also be included in the Lagrangian, and is done so by introducing the electromagnetic tensor

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \quad (2.4)$$

which describes the electromagnetic field strength. The final form of the QED Lagrangian with no

mass term is therefore

$$\mathcal{L}_{QED} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\bar{\psi}\not{D}\psi, \quad (2.5)$$

where $\not{D} = \gamma^\mu D_\mu$. This equation is accounted for by the first line of the SM Lagrangian (Eqn. 2.1).

2.3.2 Quantum Chromodynamics

QCD is the theory of how the strong force binds quarks through the exchange of gluons. This is represented by a $SU(3)$ gauge symmetry where 3 indicates the number of color charges: red, blue, and green.

This Lagrangian can be derived in almost the same way as was done for QED. Starting from a massless $\mathcal{L}_{fermion}$, the covariant derivative is introduced to ensure gauge invariance. This derivative is defined as

$$D_\mu = \partial_\mu + ig t_a A_\mu^a, \quad (2.6)$$

where g is the strong force coupling constant, t_a are the 8 generators of the $SU(3)$ symmetry called Gell-Mann matrices [6], and A_μ^a represents the gluon field. The gluon field tensor is then defined as

$$G_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + gf^{abc} A_\mu^b A_\nu^c, \quad (2.7)$$

where f^{abc} are the structure constants of the $SU(3)$ symmetry group. The big difference between the gluon field tensor and the electromagnetic tensor found in Eqn 2.4 is the addition of the third term in the gluon field tensor. This term refers to the self-interactions gluons can have through the strong force since the gluons themselves have a color charge. This is not present in the electromagnetic tensor since photons do not contain electric charge, and this makes up a large distinction between QED and QCD. The final form of the QCD Lagrangian with no mass term is therefore

$$\mathcal{L}_{QCD} = -\frac{1}{4}G_{\mu\nu}^2 + i\bar{\psi}\not{D}\psi, \quad (2.8)$$

which also contributes to the first line of the SM Lagrangian (Eqn. 2.1).

Unlike the electromagnetic force, the strong force does not weaken as quarks and gluons are spaced farther apart. This leads to a unique property of QCD called “confinement” which says that quarks will always be found in pairs (called mesons) or triplets (called baryons), tied together with gluons. As quarks move farther away from each other, the energy will grow until it is energetically favorable to pair produce more quarks that will combine with the isolated quarks to create mesons and baryons. If these compound particles then have enough energy, this process can continue again, and again, to produce a shower of particles originating from the original quark or gluon. This showering process is called “hadronization”.

2.3.3 Electroweak Theory

The weak force acts on all fermions and allows them to change their type. For example, it can cause a down quark to convert into an up quark through the absorption of a W^+ boson, or it can cause an electron to convert into an electron neutrino through the emission of a W^- boson. A lepton or quark can also emit or absorb neutral Z bosons as they please. This is represented by a $SU(2)$ symmetry, but is not a self-consistent theory as it ties together with the electromagnetic force due to the electric charge of the W bosons. Instead, it is generally presented as a unification of the QED and weak theory, called the electroweak theory, which is represented by a $SU(2) \times U(1)$ symmetry.

In this theory, all fermions have a chirality (left or right-handed) such that charged weak interactions couple only to left-handed fermions (explained with $SU(2)$ symmetry) while neutral weak interactions couple to both left and right-handed ones (explained with $U(1)$ symmetry). A weak isospin quantum number (T_3) moderates the charged weak interactions allowed by defining only left-handed particles as having non-zero isospin. A weak hypercharge quantum number (Y) relates the isospin and electric charge by $q = T_3 + 0.5Y$. Electroweak interactions only occur for particles with non-zero Y , which is the case for all SM fermions.

In order for both left and right-handed fermion fields to be gauge invariant under this symmetry,

two new spin-1 vector fields must be introduced into the covariant derivative:

$$D_\mu = \partial_\mu - i\frac{g'}{2}YB_\mu - i\frac{g_W}{2}\sigma^a W_\mu^a, \quad (2.9)$$

where B_μ and W_μ^a are the vector scalar fields for the $U(1)$ and $SU(2)$ symmetries, respectively, g' and g_W are coupling constants of each field, and σ^a are the Pauli spin matrices [7]. The strength tensor $B_{\mu\nu}$ is related to B_μ like in Eqn. 2.4, and the strength tensor $W_{\mu\nu}^a$ is related to W_μ^a like in Eqn. 2.7 which accounts for the self-interactions the W bosons have because they are charged. The final form of the electroweak Lagrangian with no mass term is therefore

$$\mathcal{L}_{EWK} = -\frac{1}{4}W_{\mu\nu}^2 W_a^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} + i\bar{\psi}_L \not{D}\psi_L + i\bar{\psi}_R \not{D}'\psi_R, \quad (2.10)$$

where D' is equivalent to D without the last term in Eqn. 2.9. This term contributes to the first line of the SM Lagrangian (Eqn. 2.1).

2.3.4 Higgs Mechanism

Up until this point, masses have been ignored. The masses of particles are actually given through their interaction with a complex scalar Higgs field ϕ which transforms under $SU(2)$ gauge transformations. This field was originally theorized to account for non-zero W and Z boson masses via spontaneous symmetry breaking of the electroweak theory, allowing fields to mix and the bosons to acquire mass. The Lagrangian associated with this process is given by

$$\mathcal{L}_\phi = |D_\mu\phi|^2 - V(\phi), \quad (2.11)$$

where D_μ is the same covariant derivative defined in electroweak theory (Eqn. 2.9), and $V(\phi)$ is the Higgs potential given as

$$V(\phi) = -\mu^2\phi^2 + \lambda\phi^4, \quad (2.12)$$

where μ^2 and λ are positive constants. Spontaneous symmetry breaking occurs when the Higgs field is in the ground state of the potential $V(\phi)$ which create W, Z, and Higgs boson mass terms when \mathcal{L}_ϕ is expanded to higher order as

$$m_W^2 = \frac{g^2 \mu^2}{2\lambda}, m_Z^2 = \frac{(g^2 + g'^2) \mu^2}{2\lambda}, \text{ and } m_H = 2\mu^2. \quad (2.13)$$

Thus, \mathcal{L}_ϕ is the term on the second line of the SM Lagrangian Eqn. 2.1 which describes the interactions between the Higgs and the massive gauge bosons.

The interaction between the Higgs field and fermions is carried out through Yukawa coupling, which generally describes an interaction between a scalar field ϕ and Dirac fermion field ψ using the Yukawa potential $V \approx g\bar{\psi}\phi\psi$ [8]. The Lagrangian for the Higgs-fermion interactions can then be written as

$$\mathcal{L}_Y = \bar{\psi}_i y_{ij} \psi_j \phi \quad (2.14)$$

where y_{ij} is the Yukawa coupling matrix between the Higgs boson and each quark and lepton generation. The masses of each fermion are proportional to the corresponding coupling constants. This Lagrangian corresponds to the first term on the second line of the SM Lagrangian and therefore completes the description of Eqn. 2.1.

2.4 Dark Matter

Although the SM provides the best framework to date for particles and their interactions, it is clear that the model is incomplete as many experimental observations are unaccounted for. One significant example is its lack of explanation for dark matter. Many astronomical and cosmological studies [9–11] provide concrete evidence for another form of matter making up $\sim 27\%$ of the universe, whereas the matter represented in the SM only makes up $\sim 5\%$.⁵ Numerous dark matter theories propose particle candidates that can be indirectly observed experimentally, yet none have shown any

⁵The other $\sim 68\%$ of the universe is made up of dark energy, which is associated with the rate at which the universe expands. This topic will not be discussed in this thesis, but more information can be found at [12, 13]

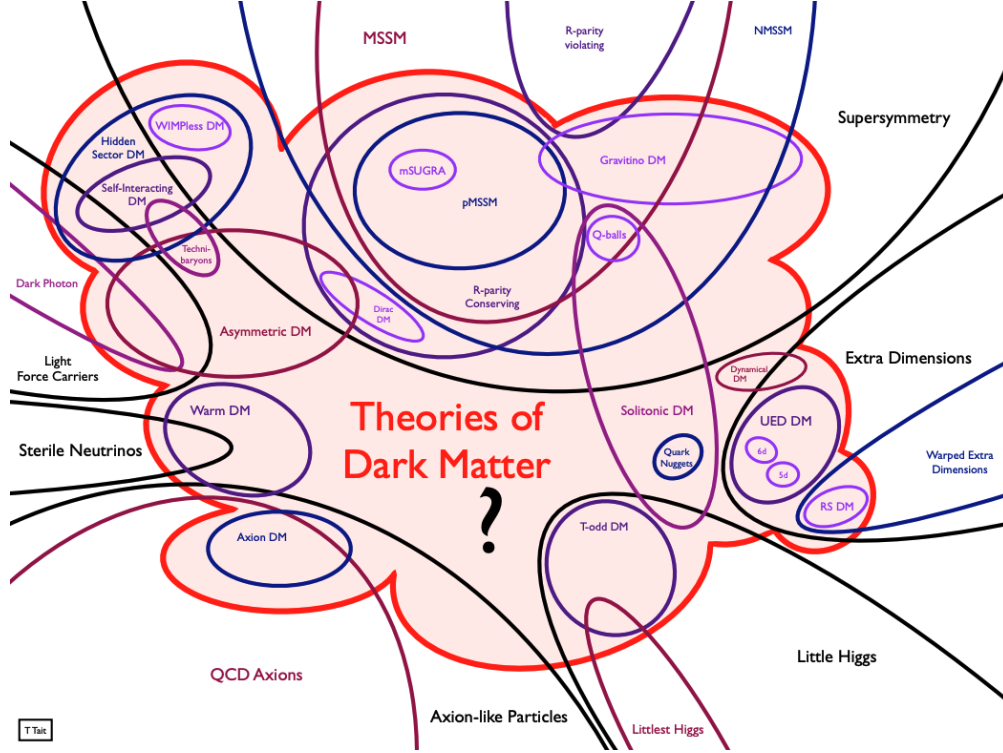


Figure 2.2: A visual map of many of the different dark matter theories and how they relate to one another [14].

significant observations thus far. This section will outline some of the popular dark matter theories that have been tested to date, as well as introduce dark sector theories which will motivate the emerging jets analysis in Chapter 4.

2.4.1 Popular Theories

Weakly interacting massive particles (WIMPs) are one of the most popular dark matter theories out there. In summary, a WIMP is a dark matter particle that interacts with SM particles through gravity and forces as weak as, or weaker than, the weak force. These particles would have a large mass ($\mathcal{O}(100 \text{ GeV})$) and would account for the amount of dark matter seen in the universe. These particles also fit naturally within other beyond-standard-model theories, such as supersymmetry. Experimental searches for WIMPs include the search for excess gamma rays or high-energy neutrinos produced from WIMP annihilation [15], and WIMP-nucleus interaction effects [16].

Axions are hypothetical dark matter particles that are motivated by asymmetries observed experimentally in the SM, called CP violation [17]. This asymmetry is seen everywhere except for the strong force, and axions allow this asymmetry to occur in QCD at very small scales. These particles would interact with the SM particles through gravity and very weak forces, and are theorized to be very light ($\mathcal{O}(1 \mu\text{eV})$). Experimental searches for axions generally include the search for photons produced through axion conversion [18, 19].

Sterile neutrinos are dark matter candidates that are theorized to only interact with the gravitational force. These particles are motivated by the observed non-zero mass of SM neutrinos, indicating the existence of both left and right-handed neutrinos. However, only left-handed neutrinos have been detected, and therefore sterile neutrinos are theorized to be their right-handed counterparts. Given sterile neutrinos are thought to mix with SM neutrinos [20] but evade any sort of detection, searches for these dark matter particles look for anomalies in SM neutrino oscillations [21].

Although each of these popular theories has been studied for decades, no experiment has yet to validate any theory with observation. It is for this reason that more and more theories continue to develop in the hopes that, one day, a dark matter detection will be made. Figure 2.2 shows a map of current dark matter theories being studied and how they overlap with one another.

2.4.2 Dark Sector

Another theory of dark matter postulates that there exists an entirely new sector of particles, called the dark (or hidden) sector, which is analogous to the SM (or light/visible) sector with its own gauge symmetry. This dark sector will have distinct dark particles and dark forces, causing it to be strongly self-interacting which is motivated by several astrophysical anomalies [22–24]. Interactions between the hidden and visible sectors are weak and indirect as the dark particles are not influenced by the SM forces. Any interaction that occurs is instead mediated through gravity or a new particle that acts as a portal between sectors. Mediating particles that are commonly studied include dark photons [25] and exotic Higgs bosons [26].

In these models, dark matter particles are allowed to decay into SM particles through the

exchange of the dark mediator particle. Experimental searches for dark sectors generally include the creation of the dark mediators which interact in the dark sector before turning back into SM particles that can then be studied. In high energy physics, hadron colliders are well suited to create potential dark mediators with TeV scale masses [27]. Several efforts in the past few years have therefore been made to test such theories that would produce signatures which current high energy physics detectors are sensitive to [28]. The dark matter analysis presented in Chapter 4 is based on a dark sector heavy mediator particles theory with a portal between dark and light sector fermions, which creates a unique signature called an “emerging jet”.

Chapter 3

The CMS experiment

3.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is the largest and most powerful particle accelerator in the world, located at the Conseil Européen pour la Recherche Nucléaire (CERN) on the border of France and Switzerland. It sits in a tunnel between 50 – 175 m underground and has a 27 km circumference. Superconducting magnets and accelerating structures are found in LHC ring to guide and accelerate two beams of proton bunches¹ travelling in opposite directions along the accelerator path until the protons reach an energy of 6.5 TeV, corresponding to a proton speed of almost the speed of light. These proton beams are then collided at designated collision points to produce 13 TeV proton-proton collisions, the highest collision energy ever created by humankind.

The LHC is the last accelerator in a chain of particle accelerators located at CERN that help boost the protons to their final energy. This chain of accelerators found at CERN can be seen in Figure 3.1. The proton source is initially created by splitting hydrogen gas into electrons and protons using a strong electric field. The protons are then sent to LINAC 2, a linear accelerator which accelerate the protons to 50 MeV, before being sent to the BOOSTER synchrotron to be accelerated up to 1.4 GeV. The PS and SPS are the next synchrotrons in the chain, increasing the proton energies to 25 GeV and 450 GeV. Lastly, the protons are ejected both clockwise and counterclockwise into the LHC where they are accelerated to their final 6.5 TeV energies before

¹Proton “bunches” refers to a collection of protons travelling together. Oscillating electric fields are used to bunch the protons along a certain frequency.

The CERN accelerator complex *Complexe des accélérateurs du CERN*

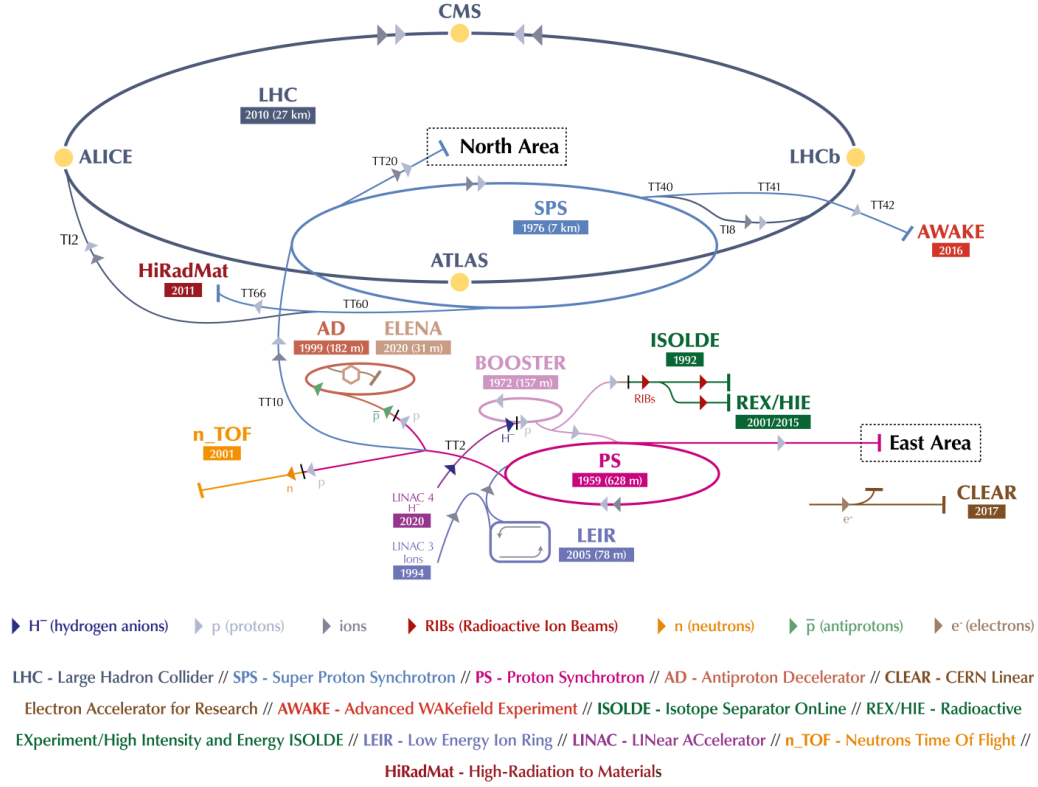


Figure 3.1: A schematic of the CERN accelerator complex [29].

being collided.

There are four collision points located along the LHC ring, and at each of these points lies a particle physics experiment ready to analyze the product of the collisions. The ATLAS [30] and CMS [31] experiments use general-purpose particle detectors located on opposite sides of the LHC to study all physics processes produced by the collisions. The LHCb experiment [32] specializes in b-hadron processes, and the ALICE experiment [33] specializes in heavy-ion collision studies.² Chapter 4 analyzes proton-proton collision data collected by the CMS experiment during the years of 2016 – 2018 (also called “Run 2”) in the search for a dark matter signal called “emerging jets”.

²Although this thesis will focus on proton collisions produced from the LHC, the LHC is also designed to collide heavy nuclei, which it does about one month per year.

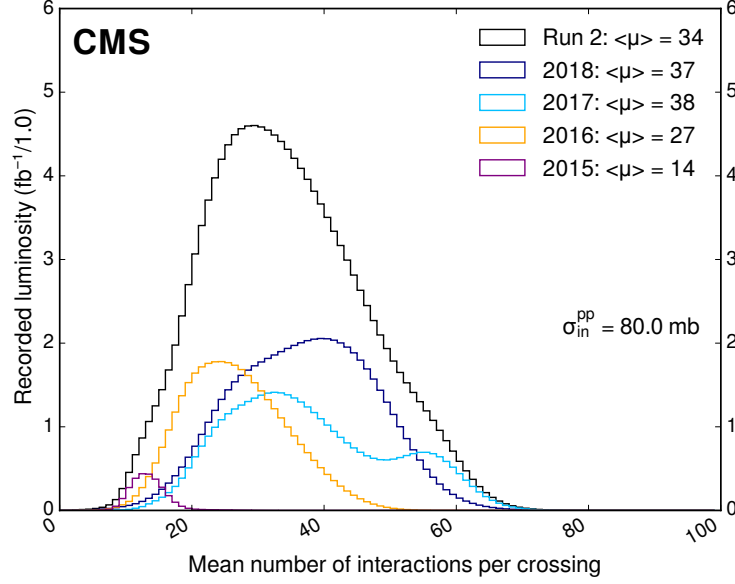


Figure 3.2: The number of proton-proton interactions (with cross-section of 80 mb) per bunch crossing event for the CMS 2015 – 2018 data collection years, as well as for all of Run 2 [37]. $\langle\mu\rangle$ represents a distribution average value.

Each LHC proton bunch contains $\sim 10^{11}$ protons and is about 1 mm wide in the transverse direction and 8 cm long along the beam line. The bunches are then focused further in the transverse direction by each experiment in order to increase the probability of proton-proton collisions. For the CMS experiment during Run 2, the transverse width of the beam was around $\sigma_{x,y} \approx 14 \mu\text{m}$ [34–36], leading to an average of ~ 34 proton-proton collisions per bunch-crossing event, as seen in Figure 3.2. Within each bunch-crossing event, the proton-proton collision that produces the highest energy is called the “hard scattering” interaction, while the rest of the collisions are referred to as “pile-up” interactions.

The rate at which protons within the bunch-crossings collide is measured by the instantaneous luminosity, defined as

$$\mathcal{L}_{inst} = \frac{N_p^2 n_p f}{4\pi\sigma_x\sigma_y} F, \quad (3.1)$$

where N_p is the number of protons per bunch, n_p is the number of bunches per beam, f is the revolution frequency of the beams, and F is a geometric correction factor due to the beams being

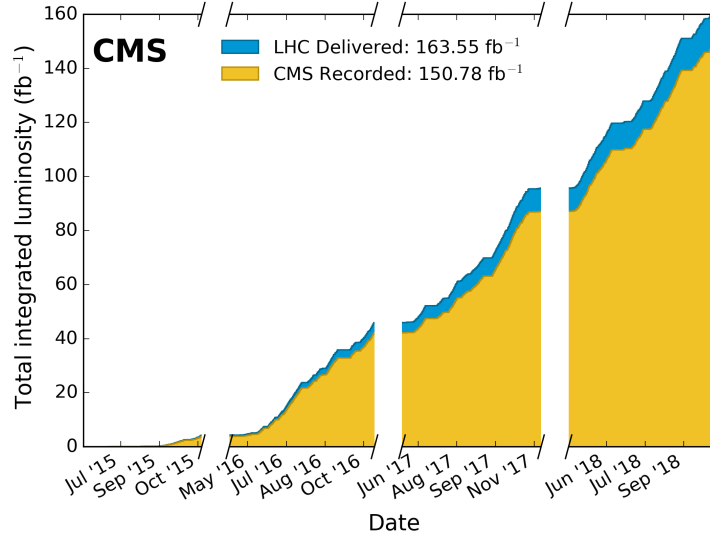


Figure 3.3: The total integrated luminosity during the 2015 – 2018 collection years for the CMS experiment [37]. The gaps along the date axis indicate time periods where no data was being collected.

collided at a slight angle. If the proton bunches are tightly packed, then \mathcal{L}_{inst} is large and there is a high likelihood protons will collide. For a given physics process with a cross section of σ , the total number of events expected during data collection time T can be written as

$$N_{event} = \sigma \int_T \mathcal{L}_{inst} dt = \sigma \mathcal{L}_{integ}, \quad (3.2)$$

where \mathcal{L}_{integ} is the total integrated luminosity. The number of events expected can be increased by either increasing the data collection period T or increasing the instantaneous luminosity \mathcal{L}_{inst} , which is highly desirable in the study of rare physics properties.

The LHC's \mathcal{L}_{inst} has more than doubled its nominal value, recording a maximum of 20 Hz/nb ($2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$) in the 2018 data collection year. Figure 3.3 shows how the integrated luminosity of the CMS detector has increased over the 2015 – 2018 data collection period, recording a total value of 150.78 fb^{-1} , where about 138 fb^{-1} is deemed good for physics and makes up the full Run 2 dataset. Within each year of Run 2 data, the instantaneous luminosity (which can be estimated by

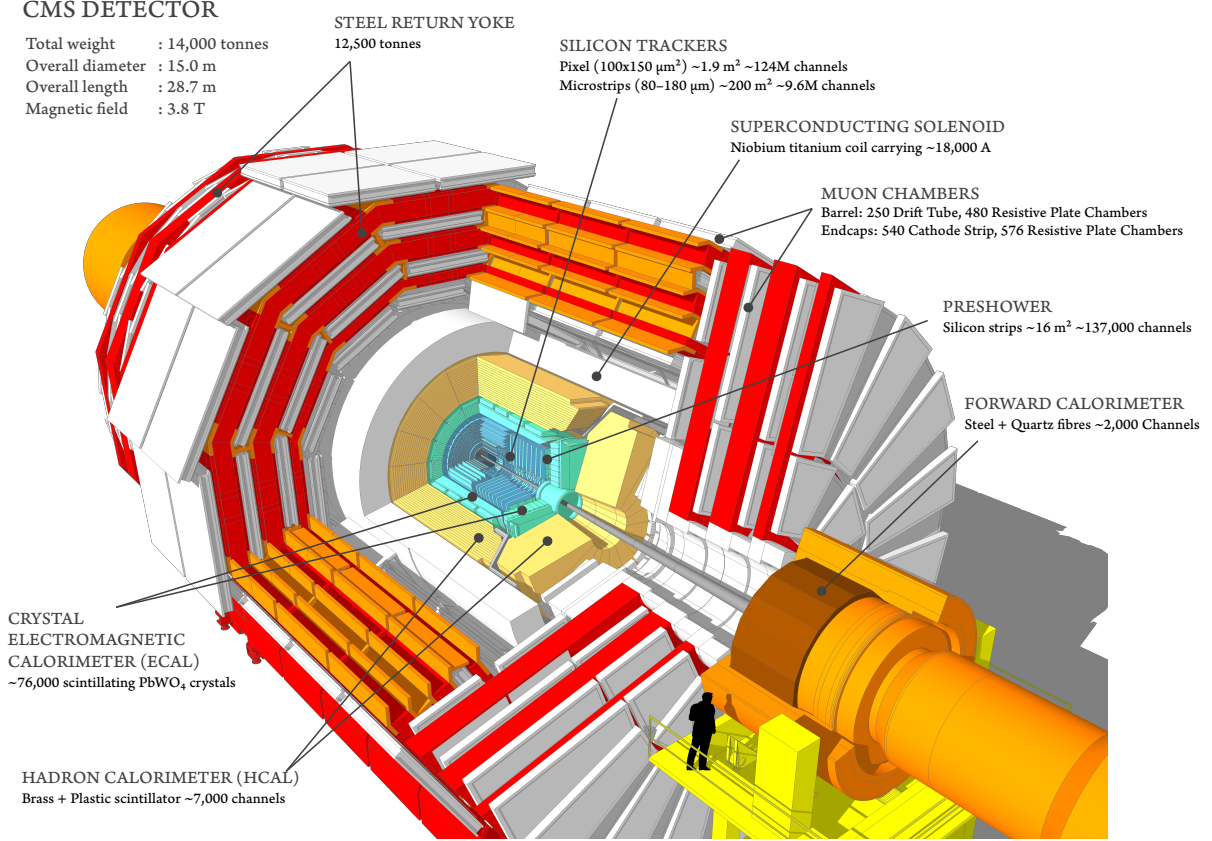


Figure 3.4: A schematic of the CMS detector [38].

the slope of the integrated luminosity distribution) increase mainly due to a stronger focusing of the proton bunches to a small transverse area.

3.2 The CMS Detector

The compact muon solenoid (CMS) detector [31], shown in Figure 3.4, is a general purpose detector designed to observe all physics phenomena produced by LHC proton-proton collisions. It is composed of a variety of different subdetectors, each specializing in distinct measurement and reconstruction techniques, surrounding the collision point. A superconducting solenoid is present within the subdetector layers to help with momentum measurements. Figure 3.5 shows an example of how the various subdetector layers affect different particles moving outward from the collision point, with the solenoid bending the charged particle trajectories as they pass through. In general,

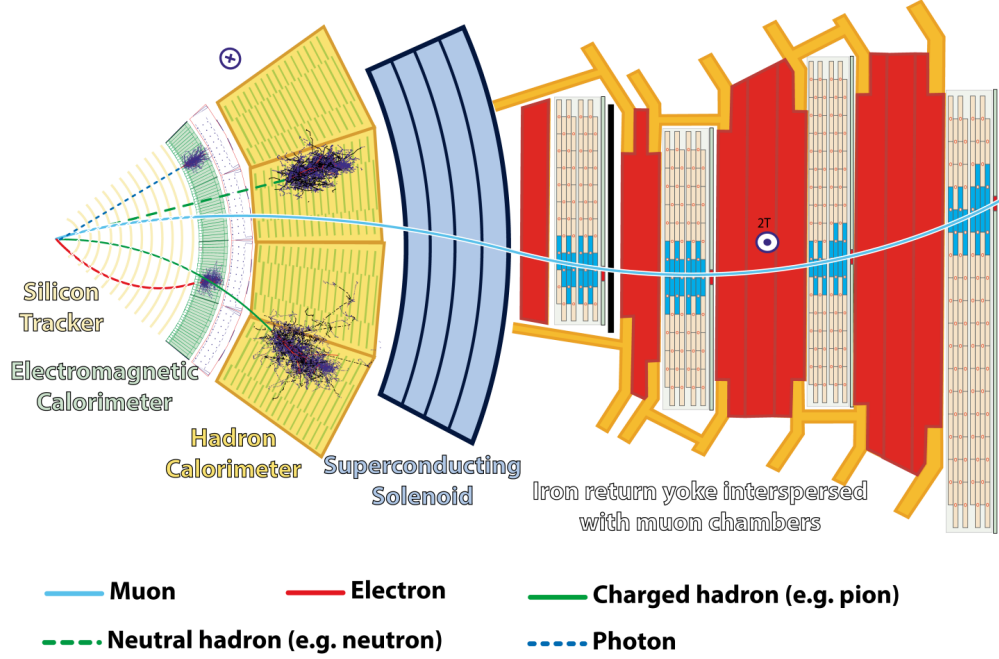


Figure 3.5: A look at how different particles behave as they pass through the different CMS detector layers [39].

all particles passing through CMS are expected to be captured within the solenoid volume except for muons and neutrinos. This section will describe the separate components of the CMS detector, starting from the layer closest to the beamline and working its way out.

Useful Kinematic Variables

Before describing the individual subdetector components, it is important to introduce a couple of CMS-standard kinematic variables that will be used continuously throughout this and the following chapters. Due to the cylindrical shape of the detector, the experiment uses cylindrical coordinates for spatial descriptions. The axis along the beamline is the z -axis, generally measured in cm, and $z = 0$ cm corresponds to the center of the detector, or the nominal collision point. r and ϕ lie in the transverse plane with r measuring the distance from the z -axis and ϕ measuring the azimuthal angle from the axis pointing towards the center of the LHC ring. θ is the polar angle within the rz -plane, but is generally transformed to the pseudorapidity angle η defined as

$$\eta = -\ln \tan \frac{\theta}{2}. \quad (3.3)$$

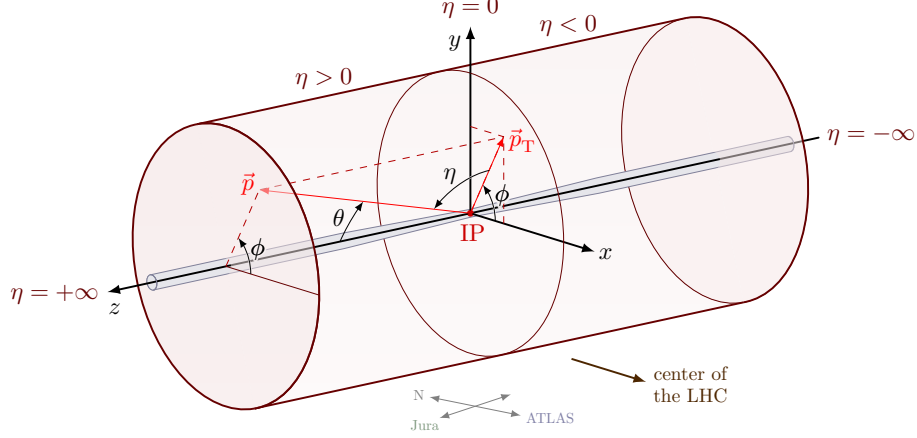


Figure 3.6: A sketch of the CMS detector coordinate system where “IP” stands for the interaction (or collision) point and the cylinder represents the detector volume [40].

A schematic of the coordinate system within the CMS detector can be seen in Figure 3.6. The cylindrical section of the detector is generally referred to as the “barrel”, whereas the ends of the cylinder are referred to as the “endcaps”.

Because the incoming proton beams are almost perfectly aligned with the z -axis, any excess momentum from the quark/gluon collisions within the protons are expected to be along z where there is no detection technology present. Therefore, momentum of physics objects are typically represented just by their transverse momentum p_T which is the magnitude of the momentum in the $r\phi$ -plane (perpendicular to the z -axis) and can be seen in Figure 3.6). p_T is generally measured in GeV, following the particle physics convention of setting the speed of light $c = 1$. The angular separation ΔR of two physics objects is measured in the (η, ϕ) space, defined as

$$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}, \quad (3.4)$$

where $\Delta\eta$ and $\Delta\phi$ are the angular separations between the 2 objects in each of these variables.

3.2.1 Silicon Tracker

The innermost part of the CMS detector is called the silicon tracker and is tasked with providing precise measurements of charged particle paths emanating from the collision point. It is also designed to reconstruct collision vertices which is crucial for identifying pileup and displaced tracks. The tracker is 5.9 m long and 2.5 m in diameter and can detect all charged particles with $p_T > 1$ GeV and $|\eta| < 2.5$. Because of the tracker's placement inside the superconducting solenoid, the charged particle paths will bend as they pass through the tracker, allowing the calculation of p_T as

$$p_T = \frac{R}{qB}, \quad (3.5)$$

where R is the radius of curvature in the transverse plane, q is the charge of the particle, and B is the magnetic field strength. In order to provide a precise R measurement, the tracker consists of around 14 layers of silicon detector technology to provide small spatial resolution.

All silicon sensors found throughout the tracking system use the same detecting principle. A thin layer of silicon is set up as a reverse-biased p-n junction. A charged particle passing through the depletion region ionizes the silicon and creates electron-hole pairs. The electric field applied within the sensor then accelerates the ionized charge towards the readout chips, generating a large, measurable current which indicates that a charged particle has passed through. How the silicon sensors are spaced determines the spatial resolution the tracker can achieve.

Because of the different levels of granularity needed in different regions of the tracker, the tracker is split into two sections. The pixel detector makes up the inner most layers and has finer spatial segmentation to provide precise vertexing information. The silicon strip detector makes up the outer most layers and does not need as fine spatial segmentation as it is only used for general charged particle trajectory detection. Figure 3.7 shows the layout of the pixel and silicon strip tracking systems. Each silicon tracker subsystem is described in further detail below.

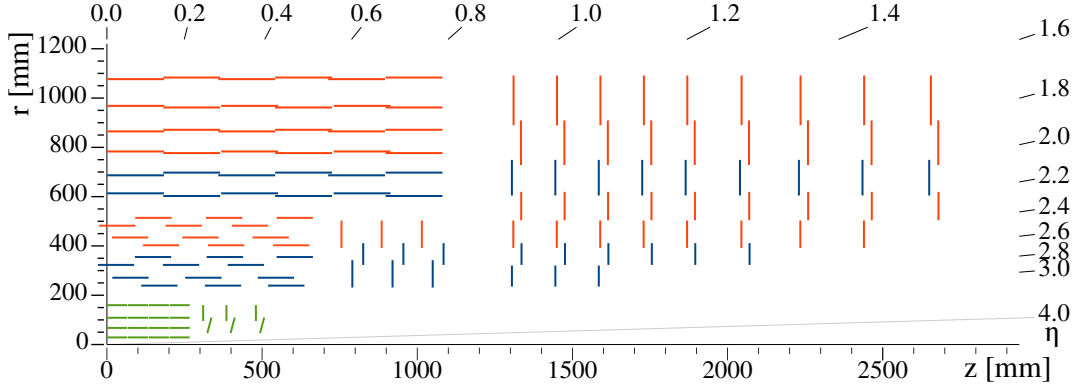


Figure 3.7: A schematic of the CMS silicon tracker (after the pixel detector upgrade) [41]. The dashed lines represent layers of silicon sensors. The pixel detector is shown by the green lines, while the silicon strip detector is shown by the blue and red lines. The red layers use a single layer of silicon sensors, while the blue layers use a double layer of silicon sensors.

3.2.1.1 Pixel Detector

In the pixel detector, the silicon sensors are arranged in a dense grid of pixels, each being $100 \mu\text{m} \times 150 \mu\text{m}$ in size. Three cylindrical layers at r of 4.4, 7.3, and 10.2 cm can be found in the barrel, complemented by two disks of pixel layers at $|z|$ of 35.5 and 48.5 cm in the endcap. This setup provides 3 high precision points for each charged particle trajectory that passes through the pixel detector. With this information, the pixel has small impact parameter resolution which is crucial in the reconstruction of secondary vertices (vertices generated from long-lived particles farther from the collision point).

At the end of 2016, the pixel detector was upgraded [42] to include more layers closer to the beamline, with 4 barrel layers at r of 2.9, 6.8, 10.9, and 16.0 cm, and 3 endcap layers at $|z|$ of 29.1, 39.6, and 51.6 cm. This was motivated by the increase in instantaneous luminosity produced by the LHC causing more proton-proton interactions per bunch crossing, and therefore making the reconstruction of vertices and charged particle paths more difficult. As a result of the upgrade, tracking performance increased altogether. This performance increase includes having smaller vertex resolutions (from $\sim 25 \mu\text{m}$ to $\sim 15 \mu\text{m}$), and higher hit efficiency – the fraction of pixel signals (hits) that are within 1 mm of where the hit is expected – (from $\sim 94\%$ to $> 99\%$).

3.2.1.2 Silicon Strip Detector

In the silicon strip detector, the silicon sensors are arranged into long strips (about 10 cm) varying in width from 80 to 205 μm . The use of long strips allows for good resolution perpendicular to the strip direction while using neighboring tracking modules to infer spatial information along the length of the strip to keep the cost of the tracking system down. 10 layers of strips in the barrel are located between a r of 20 to 110 cm, and 12 layers of strips in the endcaps are located between a $|z|$ of 70 to 270 cm. Some layers in both the barrel and endcap have two silicon strips aligned with a 100 mrad relative angle between them to allow for 3-dimensional tracking information.

3.2.2 Calorimeters

Next in the chain of subdetectors are two calorimeters used to stop and measure the energy of various particles. The electromagnetics calorimeter is specialized for electrons and photons, while the hadron calorimeter is for hadrons. Specifics for each calorimeter can be found below.

3.2.2.1 Electromagnetic Calorimeter

The electromagnetic calorimeter (ECAL) is a homogeneous detector made up of lead tungstate (PbWO_4) crystals. When an electron or photon pass through the crystals, they interact electromagnetically to produce “showers” of Bremsstrahlung photons (from the electron) and electron-positron pairs (from the photon). These showers also produce scintillating light with an intensity proportional to the energy of the original particle. This light is collected by photodiodes and phototriodes placed behind the crystals to measure the total energy.

The PbWO_4 crystals have high density (8.28 g/cm³), a short radiation length (0.89 cm), and a small Molière radius (2.2 cm) which leads to fine granularity within a compact space. They are arranged into two separate sections: a central barrel section (EB) with coverage up to $|\eta| = 1.48$, and the endcap region (EE) which extends the coverage region up to $|\eta| = 3.0$. The crystals in the EB are produced in $2.2 \times 2.2 \times 23 \text{ cm}^3$ blocks and are oriented radially outward from the collision point. This gives a transverse granularity of approximately 0.0175×0.0175 in $(\Delta\eta, \Delta\phi)$. In the

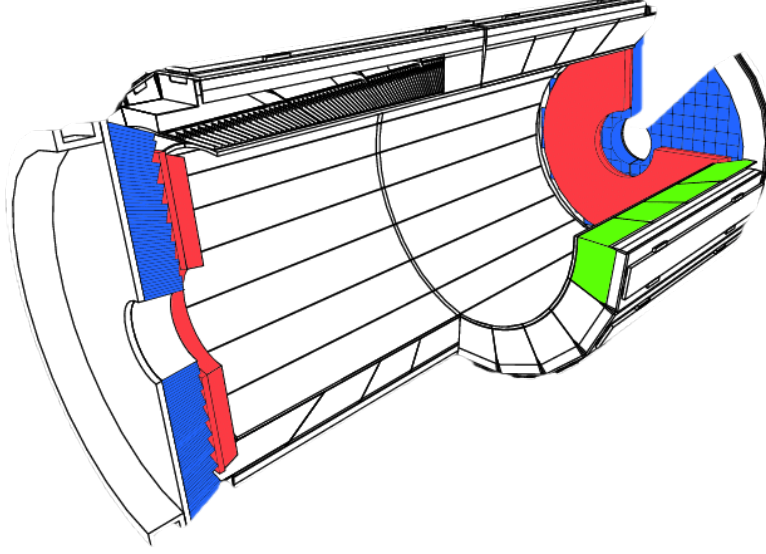


Figure 3.8: A schematic of the CMS ECAL [43] where the blue is the EE, the red is the SE, and the green is the EB.

EE, the crystals are oriented in $3.0 \times 3.0 \times 22 \text{ cm}^3$ blocks and also point towards the collision point. This gives a transverse granularity of up to approximately 0.05×0.05 in $(\Delta\eta, \Delta\phi)$.

In addition to the EB and EE, a preshower detector (SE) is set in front of the EE to improve the distinction between photons generated from the hard scattering process and photons generated from neutral pions decays (most commonly from the $\pi^0 \rightarrow \gamma\gamma$ process). The SE is based on a lead absorbers and 2 mm wide silicon strip sensors which allow for finer granularity. Neutral pion decays tend to result in two closely-spaced low-energy photons which can only be distinguished from a single photon in the SE (not in the EE) due to the finer granularity the SE provides. A full diagram of the ECAL can be seen in Figure 3.8.

3.2.2.2 Hadron Calorimeter

The hadronic calorimeter (HCAL) comes after the electromagnetic calorimeter, and it is a sampling calorimeter tasked with absorbing and measuring hadronic showers. There are two main parts to the HCAL: the absorbing material and scintillating material. The absorbing material is either brass or steel and is used to stop the incident particles along their path. The scintillating

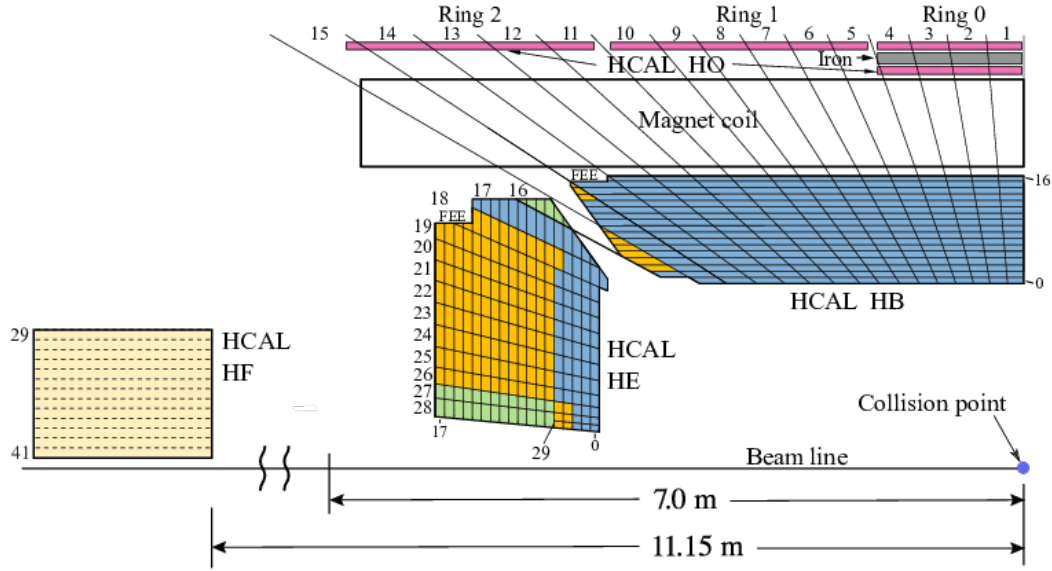


Figure 3.9: A schematic of the CMS HCAL [44] showing the locations of the 4 main regions: the HB, HO, HE, and HF.

material is plastic scintillator tiles which produces light when particle showers generated in the absorber material pass through. This light is then collected and measured by photodiodes.

The HCAL is split up into 4 different regions. The barrel region (HB) has a coverage of up to $|\eta| = 1.4$ and lies between the ECAL and solenoid magnet. Since some high energy particles can make it through the HB and penetrate the magnet, a region in the barrel outside of the solenoid (outer barrel region, HO) with the same $|\eta|$ range is also occupied with a calorimeter. The HCAL endcap (HE) region spans $1.3 < |\eta| < 3$, and an additional calorimeter is also included farther along and closer to the beamline (forward region, HF) to capture those particles in the $3 < |\eta| < 5.2$ region. A diagram of the HCAL can be seen in Figure 3.9.

In the HB, HE and HO, the absorbing layers (combination of brass and steel) are placed parallel to the barrel and endcap surfaces and the scintillating material splits each region along various η values to form “towers”. This results in an $(\Delta\eta, \Delta\phi)$ transverse granularity of $(0.087, 0.087)$ everywhere except for $|\eta| \geq 1.6$ where it is about $(0.17, 0.17)$. Because the HF is placed within a very high η range, it is prone to high radiation from the collision. Therefore, this region uses only thick steel plates as absorbers and quartz fibers as scintillators which are more radiation hard. The

towers in the HF are configured to have a $(\Delta\eta, \Delta\phi)$ transverse granularity of $(0.175, 0.175)$.

3.2.3 Superconducting Magnet

The superconducting solenoid lies in the barrel between the HCAL and muon system and has an inner diameter of 6 m and length of 13 m. It consists of over 2000 coils of superconducting niobium-titanium alloy wires, each carrying over 18 kA of current to produce a 3.8 T magnetic field along the z -axis within the solenoid volume. The strength of the field produced allows for measurements of particle with momentum up to $p_T \sim 1$ TeV with a relative uncertainty of 10%. In order to contain the magnetic field outside of the solenoid, iron return yokes are used. These iron yokes also help stop all remaining particles except for muons and neutrinos.

3.2.4 Muon System

The outermost part of the CMS detector is dedicated to the muons produced from the collisions which are expected to be one of the only particles that will pass beyond the tracker, calorimeters, and solenoid due to their large mass and lack of participation in the strong interaction. The muon system is designed to precisely measure muon trajectories and properties using gaseous chambers paired with an anode and cathode on either ends of the chamber. As the muons pass through the chambers, the gas ionizes and a drift current is created which can then be measured. There are 3 different sorts of gaseous chambers used through the muon system – drift tubes (DT), cathode strip chambers (CSC), and resistive plate chambers (RPC) – which together lead to a muon reconstruction efficiency of $> 95\%$ almost everywhere. Figure 3.10 shows how each of these different chambers are set up, supplemented with a brief description of each below.

DTs are placed in the barrel region and have a coverage of $|\eta| < 1.2$. They are about 4 cm wide, 1 cm tall, and 2-4 m long. A high voltage (anode) wire is placed in the center to pull the ionized electrons towards the center and create a signal. The time that it takes for the electrons to reach the wire help indicate where the muon passed in the chamber. DTs are stacked in layer and in various directions to form a DT chamber which can measure a spatial coordinate in both the

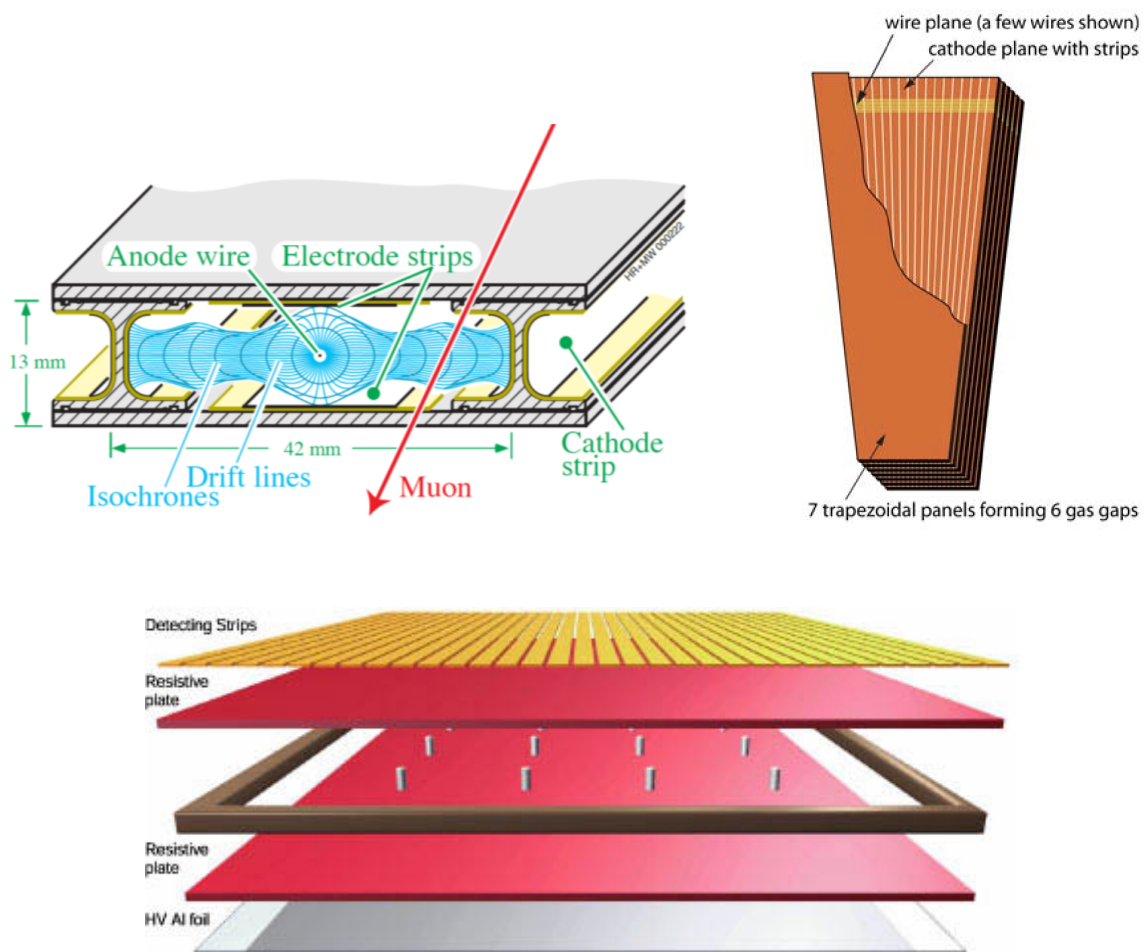


Figure 3.10: A look at the three different muon chambers used by the CMS muon system. The top left is a DT [31], top right is a CSC [31], and bottom is an RPC [45].

$r\phi$ -plane and along the z -axis. There are 4 DT chamber surrounding the beamline at various radii.

In the endcap, CSCs are used to cover a range of $0.9 < |\eta| < 2.4$. They are arranged in panels with constant $\Delta\phi$ around the endcap circle. Multiple anode wires run inside the panels in the azimuthal direction while cathode strips run radially so that they are perpendicular to each other. This allows a spatial position to be extracted by comparing the signal on the anode and cathode without relying on the drift time of the ionized electrons. This is a favorable property of the CSC in the endcap where muon rates are higher and the magnetic field is large and non-uniform.

Because there is uncertainty in the background rates and the muon systems ability to measure a precise collision time, RPCs are scattered throughout both the barrel and endcap to provide better timing resolution. These gaseous chambers have anode-cathode plates separated by a thin layer of gas. A muon passing through will create an avalanche of electrons which are picked up quickly by metallic strips, which can also be used for spatial determination.

3.2.5 Trigger System

Proton-proton collision rates at the LHC can get up to 40 MHz, which corresponds to around 60 TB/s if all events are saved. The CMS detector is unfortunately not capable of processing and storing such massive data rates, and therefore fast decisions must be made on which collision events should be kept. The trigger system is in charge of event selection in real time with the collisions in order to reduce the data rate to ~ 1 kHz. This drastic data reduction is done in two steps; the Level-1 Trigger (L1T) reduces the 40 MHz to 100 kHz, then the High-Level Trigger (HLT) takes in this new rate and further reduces it to the desired 1 kHz rate.

3.2.5.1 Level-1 Trigger

The L1T is a hardware-based system composed mainly of FPGAs for their flexibility, but also of ASICs and programmable memory when speed and density are especially important. When a collision event occurs, energy deposits from the calorimeters and track segments from the muon system are quickly processed to calculate a few important physics objects, such as electron/photon

and muon candidates. These objects are sent to the Global Calorimeter and Global Muon Trigger where they remove low reconstruction quality objects and send a reduced object collection to the Global Trigger. The Global Trigger is then tasked with making the ultimate decision on whether an event is stored or not. The total decision time of the L1T must occur within $3.2 \mu\text{s}$ and the process is pipelined such that the L1T can begin processing a new event while still working on the event preceding it.

3.2.5.2 High-Level Trigger

The HLT [46] is a software-based system held on a CPU farm which can house more complex algorithms. When an event passes the L1T, it is sent on to the HLT for a higher level of processing. The HLT improves the initial L1T objects by using information collected from the tracker which provides a more complete view of the collision event. Particle tracks can also be combined to create vertices and jets. The HLT then makes decisions to target certain event topologies that physicists are interested in studying. If the events do not pass these selections, then they are thrown away.

3.3 Event Reconstruction

After events are selected, the full detector information collected from the event stored in a buffer during trigger selection is then sent off for event reconstruction. Event reconstruction refers to the process of taking the electrical signals recorded by the CMS detector and converting them into a reenactment of the collision event. The reconstructed event contains information about the particles produced, their trajectories, and their physical properties, which can then be used for later analysis.

The reconstruction process starts with signals collected by the individual subdetectors. These signals consist of hits in the silicon tracker and muon system, and energy deposits in the ECAL and HCAL. Primitive objects can then be built locally within the subdetector by grouping the hits together to create track segments and clustering the energy deposits together to form calorimeter clusters. These primitive objects are then used to create more sophisticated local objects which represent the fundamental elements from each subdetector. The fundamental elements are then

combined across all subdetectors with the Particle Flow algorithm [47] to form particle candidates complete with direction, energy, and particle-type information. Sections 3.3.1 and 3.3.2 will describe this process in more detail.

Particle candidates are often combined to form higher-level physics objects useful in the analysis of many physics signals. For example, the emerging jets analysis in Chapter 4 searches for clusters of particles that originate from the same vertex to form a “jet” which has a cone-like structure. Section 3.3.3 will describe how jets and other high-level physics objects used by the emerging jet analysis are reconstructed from particle candidates.

3.3.1 Fundamental Elements

Before reconstruction can happen globally to create particle candidates, it must first be done locally within each CMS subsystem. When particles travel outward from the collision point, all charged particles will leave hits in the silicon tracker that can be combined to create charged particle tracks. Electrons, photons, and hadrons will shower and produce multiple energy deposits in the calorimeters which can be combined to create a supercluster relating back to a single particle. Muons are expected to be the only particles which will pass through and interact with the muon system to create hits which can be combined to create a muon track. In this section, the reconstruction of all of these fundamental elements will be described.

3.3.1.1 Charged Particle and Muon Tracks

Charged particle and muon tracks reconstructed in the silicon tracker and muon system, respectively, are done so using a similar method. Charged particles will have curved trajectories due to the force applied by the solenoid magnetic field, where the radius of curvature is proportional to a particle’s transverse momentum (see Eqn. 3.5 and Figure 3.5). As these particles pass through the detector, hits are left in the tracker and muon system. These hits are then combined to form a track by fitting a sequence of hits to potential helix paths using a Kalman filter [48].

First, track seeds are created from a small number of hits in different tracker or muon system

layers. The hits in the seeds and their location relative to the center of the detector can then be used to fit a helix. The Kalman filter will then use the helix and seed to project where hits in additional layers would be found. If a hit is found near the projection, then it is added to the potential track. Once all the layers have been tested, the potential track is then checked to see if there are enough hits associated with it, not too many consecutive layers were missed, and the associated hits are a good enough match to the helix. If the track passes these checks, then the track's p_T , direction, and closest approach to the beamline is calculated.

For muons, tracking efficiency is essentially 100% for $p_T > 1$ GeV. For charged pions (and many other charged hadrons), which are subject to nuclear interactions unlike muons, the efficiency is $> 85\%$ for p_T between 1 and 20 GeV. For electrons, which lose a lot of their energy due to Bremsstrahlung radiation before they reach the end of the silicon tracker, the efficiency is $> 80\%$ for $p_T > 2$ GeV. Reconstruction in the barrel is better than the endcap for all particles.³ Comparisons between all three of these particle types can be seen in Figure 3.11.

Vertices are created from a collection of tracks and are paramount in removing pileup to study the hard interaction. They are also important in determining secondary vertices in long-lived particle and dark matter analyses like emerging jets. Tracks are first selected to find those consistent with originating from the primary interaction point. These tracks are then clustered along point of closest approach z coordinate using a deterministic annealing algorithm [49] to determine candidate vertices. These candidate vertices are then re-fit using associated tracks to determine the final position. The resolution of the primary vertex z position is generally $\sim 10 - 20 \mu\text{m}$.

3.3.1.2 Calorimeter Clusters

Energy deposits in the ECAL and HCAL are evaluated in a similar way. Local energy maxima within the collection of energy deposits are labeled “cluster seeds”. These seeds will then grow into “topological clusters” by aggregating the energy of the neighboring cells until all clusters are isolated.

³Performance degradation in the endcap is due to many complex effects, such as higher radiation, larger density of particles passing through, and less bend in particle trajectories in this detector region.

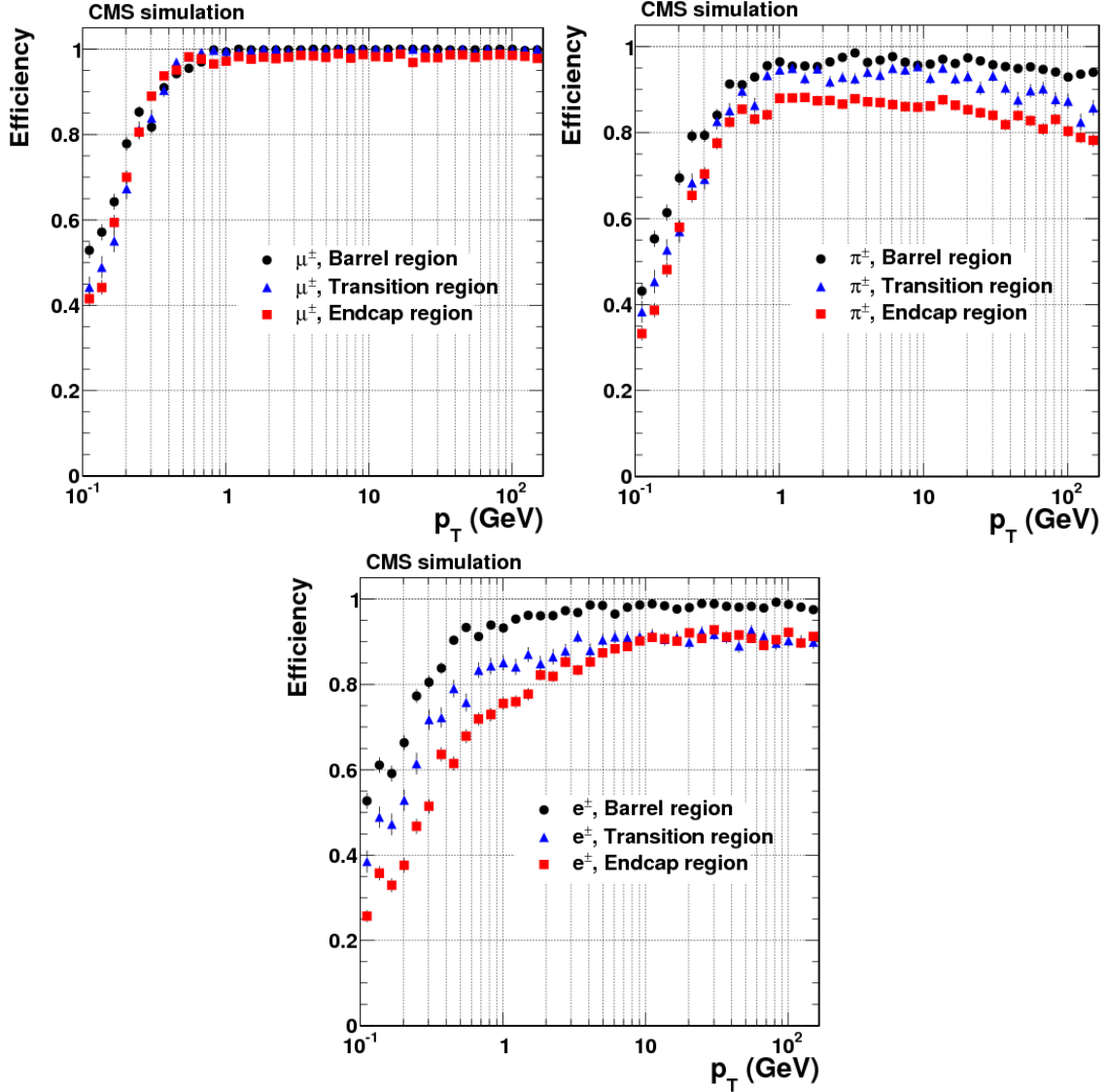


Figure 3.11: Track reconstruction efficiency of muons (top left), charged pions (top right), and electrons (bottom) for the different detector regions defined by the $|\eta|$ intervals of 0–0.9 (barrel), 0.9–1.4 (transition) and 1.4–2.5 (endcap) [49].

Only deposits with energy above a noise threshold are used in the algorithm. Once the topological clusters are determined, the position can be calculated using an energy-weighted average position of all original energy deposits added to the cluster. In the ECAL, the energy resolution for electrons is between 2 – 5%, and for photons is between 1 – 5%. In the HCAL, the energy resolution of a 300 GeV pion is $< 10\%$.

3.3.2 Particle Flow Candidates

The Particle Flow algorithm [47] takes in all of the fundamental elements reconstructed locally in the subdetectors and combines them to create final particle candidates. The fundamental elements are first linked across different subsystems to create element blocks. Each block has the potential to carry one charged particle track, several calorimeter clusters, and one muon track, dependent on the physics object the block represents. For example, a block representing an electron will probably have a charged particle track and ECAL cluster. Once the blocks are formed, they are classified by particle type and reconstructed using the block elements.

Links between the fundamental elements are all based on position. The charged particle tracks are extrapolated along the full range of the detector and linked to calorimeter clusters if a cluster is found in the expected calorimeter location. Similarly, charged particle tracks are linked to muon tracks if the propagation of both track trajectories match well. ECAL and HCAL clusters are linked if they are within the same (η, ϕ) space. The links found are used to help determine the type of particle the block of elements represents.

The blocks are identified by particle type as follows:

- **Muons** are the only particles that will leave a signature in both the silicon tracker and muon system. Any calorimeter cluster also found within a muon block will account for some of the muons energy, expected to be ~ 3 GeV for an ECAL cluster and ~ 0.5 GeV for the HCAL.
- **Electrons** are expected to leave a signature in the silicon tracker and ECAL. Tracks formed from electrons may be low in quality since their low masses make them prone to Bremsstrahlung radiation and energy losses in the tracker. Therefore, these candidates are refit using both the track and cluster using a Gaussian-sum filter [50] which is a modified Kalman filter which accounts for energy losses within each material layer.
- **Photons** are only expected to interact with ECAL. Therefore, an ECAL cluster is all that is needed to determine the energy and position.

- **Neutral hadrons** will mainly interact with the HCAL, while also leaving small energy deposits in the ECAL.
- **Charged hadrons** will look like neutral hadrons, but will also have a signature in the tracker.

Once the blocks are identified and checks are made to ensure that the block elements correspond well with the same particle, the final particle candidates are created.

3.3.3 High-Level Objects

Any physics objects reconstructed from the Particle Flow candidates are considered high-level physics objects. These objects are useful to analyse interesting physical topologies produced from proton-proton collision events. This section will describe the high-level objects studied within the next few chapters of this thesis: jets, jet p_T sum, and missing transverse momentum.

3.3.3.1 Jets

When quarks or gluons hadronize, they produce a collection of particles that spread outward in a cone-like structure in the same direction as the original quark/gluon. This cone of particles is called a “jet”, and a visual representation can be found in Figure 3.12. Particle Flow will reconstruct the individual particles that fall within the jet cone, but an additional reconstruction step must be made to cluster the particles together and calculate the properties of the jet.

Jet reconstruction is done using the anti- k_t algorithm [51], which generally clusters jets around the highest p_T particles spaced a certain distance apart in the event. This algorithm revolves around two distance metrics, given by

$$d_{ij} = \min \left(\frac{1}{p_{T,i}^2}, \frac{1}{p_{T,j}^2} \right) \left(\frac{\Delta \bar{R}_{i,j}^2}{R^2} \right) \text{ and } d_i = \frac{1}{p_{T,i}^2}, \quad (3.6)$$

where d_{ij} represents the distance between the i th particle candidate and a pseudo-jet j , $\Delta \bar{R}_{i,j}^2 = \Delta p_{T,\eta}^2 + \Delta p_{T,\phi}^2$ is the angular separation between the candidate and pseudo-jet p_T , R is a chosen

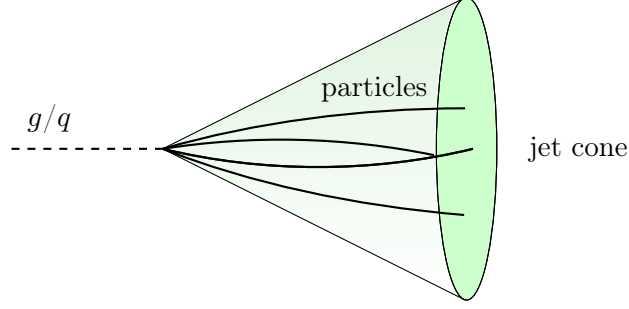


Figure 3.12: An example of a jet where the solid lines represent the particles produced by the hadronization of the gluon or quark (dashed line) falling within a jet cone.

parameter representing the width of the jet cone being built, and d_i represents the distance between the i th particle candidate and the collision point. The pseudo-jets start with high p_T particles spaced far away from one another, and particle candidate i will be added to pseudo-jet j if $d_{i,j} < d_i$. The new pseudo-jet p_T will be the vector sum of all particles assigned to the jet. If instead $d_{i,j} \geq d_i$, then candidate i will be tested against the other pseudo-jets. If candidate i does not pass the distance measure to join any pseudo-jet, then it will start its own pseudo-jet for other particles to be tested against. This process will continue for all particle candidates in the Particle Flow candidate collection, until they are all assigned to a pseudo-jet. These final reconstructed jets are what are used in the emerging jet search in Chapter 4 with R set to 0.4 (producing “AK4” jets).

It should be noted that the distance depends largely on the inverse p_T^2 of the particles. This causes lower energy particles to cluster with higher energy particles long before they will cluster with themselves. A high energy particle with no other high energy particle neighbor within $2R$ will collect all low energy particles around it to create a jet with a circle of radius R . If two high energy particles forming their own jets are within $(R, 2R)$ of each other, then neither jet will be conical as there will be some overlapping of the jets. Overall, a key feature of the anti- k_t algorithm is that low energy particles do not change the shape of the jet, therefore making this algorithm resilient to soft radiation.

3.3.3.2 Jet p_T Scalar Sum

H_T is an event-level variable built from jet properties. It is the scalar p_T sum of all jets in an event, or

$$H_T = \sum_{jets} |p_T|, \quad (3.7)$$

and is used as a scale of the total energy within an event. Although H_T is more of a variable than an object, it is very useful in triggering certain physics signals which are expected to produce a large amount of hadronic energy. For example, an event with emerging jets is expected to produce high H_T (generally > 1200 GeV in the search presented in this thesis), and therefore analysers can remove all events with low H_T , greatly reducing the search space. This variable will also be utilized in the track quality development presented in Chapter 6.

3.3.3.3 Missing Transverse Momentum

The only particles that are not expected to interact with the CMS detector are those which are only weakly interacting, such as neutrinos, or those that are not currently described by the Standard Model. Because of the conservation of momentum, the total p_T of an event is expected to be 0. Therefore, any excess p_T alludes to the presence of particles not captured by the detector. This excess is called missing transverse momentum, and is defined as

$$\vec{p}_T^{miss} = - \sum_i \vec{p}_{T,i}, \quad (3.8)$$

where i sums over all Particle Flow candidates. This object will often be seen labeled as E_T^{miss} .

Although this object is useful for dark matter studies as dark matter candidates are generally theorized to not interact directly with the detector, the emerging jets analysis did not find that this object helped with search sensitivity. Instead, this object will be referenced in the track quality development chapter when benchmarking a new algorithm.

Chapter 4

Emerging Jets Analysis

4.1 Introduction

Although dark matter is widely accepted to exist based on observations of how celestial objects move, behave, and interact [52], not much else is known beyond its existence. Many dark matter theories have been developed and tested, but none have resulted in experimental observation to date. A few compelling theories draw inspiration from the complex structure of the current SM and postulate a dark sector that is QCD-like with strongly self-interacting dark matter candidates charged under a new force in the dark sector. In one particular model, the dark sector can interact with the visible (SM) sector through a mediator particle to produce exotic signatures which can be found at the LHC, called “emerging jets” [53, 54].

In the emerging jets theory, the dark sector has $SU(N_{C_{dark}})$ gauge symmetry, where $N_{C_{dark}}$ is the number of dark quark (Q_{dark}) colors, and a confinement energy scale of Λ_{dark} . A TeV-scale dark mediator (X_{dark}) [54] is charged under both QCD and dark QCD and the dark confinement scale Λ_{dark} is close to that of visible QCD at $\mathcal{O}(\text{GeV})$. X_{dark} therefore acts as a portal between the visible and dark sectors, coupling their respective quarks via Yukawa interactions (mentioned in Section 2.3.4) described by the coupling matrix κ . In the dark sector, dark hadrons are created by the dark force binding dark quarks together. Dark pions have a mass $m_{\pi_{dark}} < \Lambda_{dark}$ and are unstable, causing them to decay into SM particles through X_{dark} interaction.

These dark mediators can be pair produced at the LHC through either gluon fusion or quark-antiquark annihilation, as seen in Figure 4.1. The mediators will each carry an electric charge of $1/3$

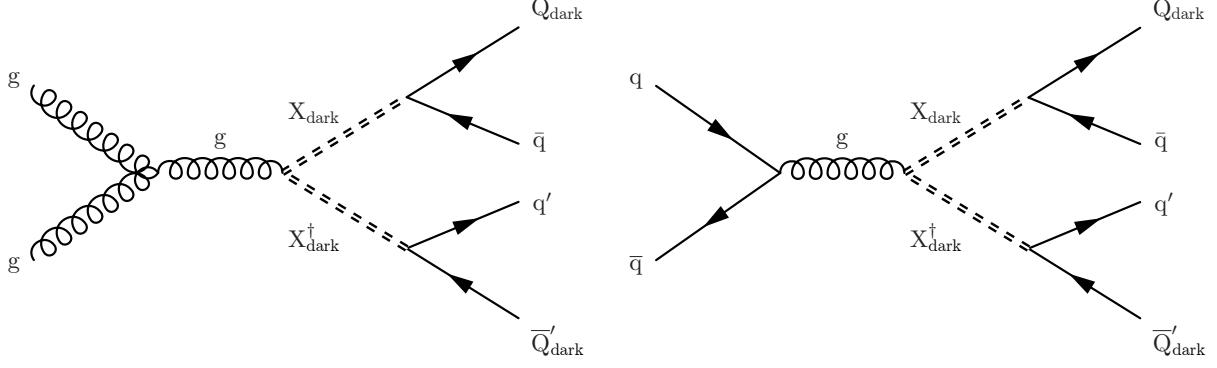


Figure 4.1: Feynman diagrams¹ of dark mediator (X_{dark}) pair production through gluon fusion (left) and quark-antiquark annihilation (right) [55].

or $2/3$ and can decay into a visible quark with the same charge and a neutral dark quark through Yukawa coupling. The dark quarks will then hadronize in the dark sector to produce multiple dark pions which travel some distance before decaying back into SM particles. The collection of SM particles that “emerge” some measurable distance from the proton-proton collision point due to the non-negligible dark pion lifetimes create an emerging jet.

Figure 4.2 shows the full process of the emerging jet production at the LHC. Each emerging jet event contains a pair of SM jets created through hadronization of the visible quarks, and a pair of emerging jets. Due to the large mass of the dark mediators, each of the four jets are also expected to have high energy as well. Figure 4.3 shows what an emerging jet would look like in the cross section of the CMS detector. This analysis searches through proton-proton collision physics events collected by the CMS detector to see if an emerging jet signature was produced within a significant portion of the events. This analysis is published on arXiv [55] and has been submitted to the Journal of High Energy Physics.

4.2 Signal Model

This analysis focuses on two separate emerging jet scenarios which are defined by the coupling allowed between the dark and SM quarks. The scenarios are as follows:

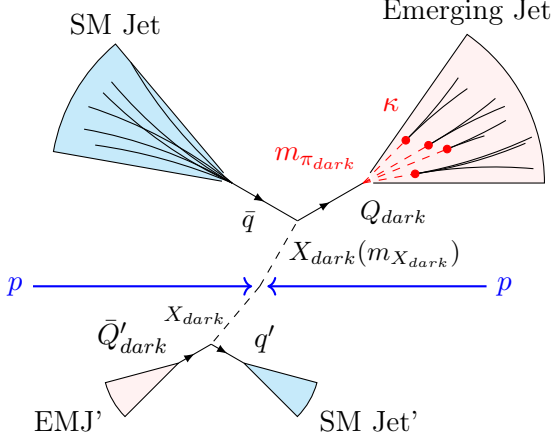


Figure 4.2: A schematic of the emerging jets signal produced through proton-proton collisions. The four jets represent the final state of the particles produced by this signature. The blue jets are SM jets, pink jets are emerging jets, and red dashed lines are dark pions within the emerging jets parametrized by their mass $m_{\pi_{dark}}$ and coupling strength κ .

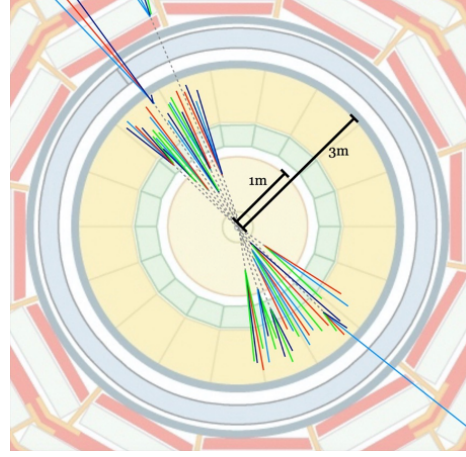


Figure 4.3: A cross section view of the CMS detector with emerging jets present [53]. The black dashed lines indicate the dark pions that do not interact with the detector, and the colored solid lines represent SM particles produced by the dark pion decays that interact and are seen by the detector.

4.2.1 Unflavored Down Scenario

The unflavored down scenario, which will often be referred to as “unflavored”, assumes degenerate dark pion masses and couplings for all dark pions to a down SM quark only [54]. With these constraints applied, the decay width (probability of a certain decay process per unit time) of the dark pions into pairs of down quarks is given by [53]

$$\Gamma(\pi_{dark} \rightarrow d\bar{d}) = \frac{\kappa^4 N_{C_{dark}} f_{\pi_{dark}}^2 m_{down}^2 m_{\pi_{dark}}}{32\pi m_X^2}, \quad (4.1)$$

where κ is the Yukawa coupling matrix, $N_{C_{dark}}$ is the number of dark colors, $f_{\pi_{dark}}$ is the dark pion decay constant (rate at which the dark pions decay), m_{down} is the mass of the down quark, $m_{\pi_{dark}}$ is the mass of the dark pion, and m_X is the mass of the dark mediator.

Since the dark pions in this scenario only decay to down quarks, all dark pions will have the

same lifetime distribution. The lifetime of the long-lived dark pions is approximated as [53]

$$c\tau_{\pi_{dark}} = \frac{c\hbar}{\Gamma} \approx 80 \text{ mm} \left(\frac{1}{\kappa} \right)^2 \left(\frac{2 \text{ GeV}}{f_{\pi_{dark}}} \right)^2 \left(\frac{100 \text{ MeV}}{m_{down}} \right)^2 \left(\frac{2 \text{ GeV}}{m_{\pi_{dark}}} \right) \left(\frac{m_X}{1 \text{ TeV}} \right)^4. \quad (4.2)$$

It can be seen here that centimeter to meter dark pion decay lengths will occur for GeV scale dark pions and TeV scale dark mediators, allowing an emerging jet signature to be contained within the CMS tracker volume.

The unflavored down scenario has been studied previously with CMS proton-proton collision data from 2016 corresponding to an integrated luminosity of 16.1 fb^{-1} , setting exclusion limits on the mediator mass up to 1250 GeV. This study will extend the search of emerging jets to higher mediator masses, as well as include nine times more data.

4.2.2 Flavor-Aligned Down Scenario

In the flavor-aligned scenario, which will often be referred to as “flavored” or “aligned”, the coupling is extended to allow dark quarks to couple to down, strange, and bottom SM quarks [56]. The decay width of the dark pions of composite flavor $\bar{Q}_\alpha Q_\beta$ decaying to a pair of SM quarks $\bar{q}_i q_j$ is given by [56]

$$\Gamma_{\alpha\beta ij} = \frac{N_{C_{dark}} m_{\pi_{dark}} f_{\pi_{dark}}^2}{8\pi m_X^4} |\kappa_{\alpha i} \kappa_{\beta j}^*|^2 (m_i^2 + m_j^2) \sqrt{\left(1 - \frac{(m_i + m_j)^2}{m_{\pi_{dark}}^2}\right) \left(1 - \frac{(m_i - m_j)^2}{m_{\pi_{dark}}^2}\right)}. \quad (4.3)$$

Under the assumption that there are 3 different dark quark flavors which each couple to a unique SM down-type quark, the Yukawa coupling matrix becomes $\kappa_{\alpha i} = \kappa_0 \delta_{\alpha i}$ where $\delta_{\alpha i}$ is the Kronecker-delta matrix. The factors within the decay width which contain final-state SM quark masses indicate that the dark pions favor decays to $\bar{q}q$ pairs containing the heaviest accessible quark.

This scenario provides four distinct flavors of dark pions: $\bar{Q}_1 Q_2 \rightarrow \bar{d}s$, $\bar{Q}_1 Q_3 \rightarrow \bar{d}b$, $\bar{Q}_2 Q_3 \rightarrow \bar{s}b$, and $\bar{Q}_i Q_i \rightarrow \bar{q}q$, as well as their complex conjugates. Figure 4.4 shows different lifetimes that are possible for a given mass of dark mediator and dark pion based on the π_{dark} quark composition.

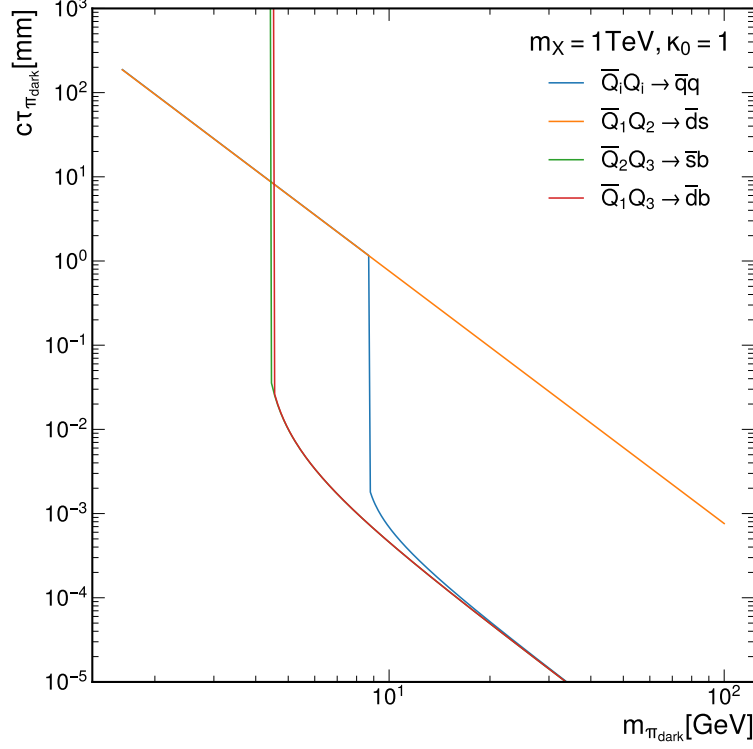


Figure 4.4: Dark pion lifetime as a function of dark pion mass for each dark pion quark composition in the flavor-aligned scenario. The mediator mass m_X is set to 1 TeV, and the Yukawa coupling scale κ_0 is set to 1.

In general, the lifetimes decrease as the dark pion mass increases, leading emerging jet production closer to the proton-proton collision point. Additional lifetimes become accessible when enough energy is present to produce heavier SM quark pairs, shown by the jumps in the figure. The b quark, with a mass of 4.18 GeV, is the heaviest down-type quark and can not be produced until $m_{\pi_{dark}}$ has enough energy. The first jump around $m_{\pi_{dark}} \approx 4.5$ GeV is when the dark pion has enough energy to produce one b quark alongside another quark, and the second jump around $m_{\pi_{dark}} \approx 9.5$ GeV is when there is enough energy to produce pairs of b quarks. Overall, the dark pion lifetime distributions will differ based on the pion quark composition, making this emerging jet signal more complicated than the signal produced by the unflavored coupling scenario.

4.3 Signal and Background Data Samples

This analysis searches for emerging jet physics events in the CMS Run 2 dataset collected in the years 2016, 2017, and 2018 (corresponding to an integrated luminosity of 137.6 fb^{-1}). Given that the four jets expected in the signal (two SM and two emerging jets) are theorized to have high energy, the scalar sum of the jet transverse momenta (p_T) within the event, called H_T , can be used as a trigger to select a subset of the full data where we may expect signal. This signal region of data is often referred to as “JetHT” data, and all events are required to pass the lowest- H_T un-prescaled² triggers per data collection era,³ achieving $> 98\%$ signal efficiency after making this selection.

There is another subset of data that is used for background estimation (see Section 4.5 for more information on estimation techniques) which is deemed signal-free, meaning that emerging jets have a highly suppressed probability of being produced in these events. This subset of data is called “ γ -triggered” or “single photon” data, and requires a single, isolated, high p_T photon to be present in the event. Since the emerging jet theory is not electromagnetically charged, a photon can not be produced from the emerging jet interaction shown in Figure 4.2. Therefore, the probability of having an event with both a high p_T photon and emerging jets is essentially zero.

In order to develop the selection criteria for events that potentially contain signal, data was simulated using the existing Hidden Valley model framework [57] in PYTHIA 8 [58] by specifying the flavor indices for the dark mesons in the Hidden Valley to match certain dark quark compositions [59]. For both the flavored and unflavored coupling scenarios, $N_{C_{dark}}$ was set to 3, the dark pions were mass degenerate and the dark confinement scale Λ_{dark} was set to double the mass of the dark pions. The unflavored scenario then has 3 free and independent parameters: the dark mediator mass m_X , the dark pion mass $m_{\pi_{dark}}$, and the dark pion lifetime $c\tau_{\pi_{dark}}$. The aligned scenario also has 3 free and independent parameters: the dark mediator mass m_X , the dark pion mass $m_{\pi_{dark}}$, and the Yukawa coupling constant κ_0 since the lifetime $c\tau_{\pi_{dark}}$ depends on the dark pion quark composition, as shown in Figure 4.4.

²An un-prescaled trigger means that every event which passes the selection is recorded and studied.

³ H_T must pass 900 GeV for 2016, and 1050 GeV for 2017 and 2018 data.

Model Parameter	List of values
$m_{X_{dark}}$ [GeV]	1000, 1200, 1400, 1500, 1600, 1800, 2000, 2200, 2400, 2500
$m_{\pi_{dark}}$ [GeV]	10, 20
$c\tau_{\pi_{dark}}$ [mm]	1, 2, 5, 25, 45, 60, 100, 150, 225, 300, 500, 1000

Table 4.1: Mass of dark mediator m_X , mass of dark pion $m_{\pi_{dark}}$, and lifetime of dark pion $c\tau_{\pi_{dark}}$ being tested in the unflavored coupling scenario.

Table 4.1 shows all free parameters that are scanned for this analysis in the unflavored scenario. The mass of the dark mediator ranges between 1000 and 2500 GeV to extend the search region from the previous analysis which excluded the mass up to 1250 GeV. The lifetime of the dark pions was chosen to be between 1 to 1000 mm to ensure that most emerging jets would be fully contained within the CMS tracker volume. The emerging jet signal does not change much as a function of the dark pion mass, and so two benchmarks were tested at 10 and 20 GeV.

For the flavor-aligned scenario, Table 4.2 shows all free parameters studied in this analysis. The mass of the dark mediator is the same range as for the unflavored scenario. An additional dark pion mass at 6 GeV is studied as this mass does not allow a decay to pairs of bottom quarks, and so the signal will be slightly different. The coupling κ_0 is related to the lifetime $c\tau_{\pi_{dark}}$ of the dark pions given the dark quark composition. This study will scan the maximum lifetime of the event ($c\tau_{\pi_{dark}}^{\max}$) between 5 and 500 mm so that the emerging jets are again contained within the CMS tracker volume.

Simulated background data representing the JetHT and single photon data streams was also created to help develop the analysis strategy, explained further in Section 4.5. The simulated JetHT data will sometimes be labeled as “SM Multijet MC” or “QCD MC” and the simulated single photon data will sometimes be labeled as “GJets MC” or “ γ +jets”.

$m_{X_{dark}}$ [GeV]	$m_{\pi_{dark}}$ [GeV]	κ_0 value ($c\tau_{\pi_{dark}}^{\max}$ [mm])					
		(–	5	25	45	100	500)
1000	6	1	0.92	0.61	0.53	0.43	0.29
	10	1	0.62	0.42	0.36	0.30	0.20
	20	1	0.37	0.25	0.21	0.18	0.12
1200	6	1	1.10	0.73	0.63	0.52	0.35
	10	1	0.75	0.50	0.43	0.35	0.24
	20	1	0.45	0.30	0.26	0.21	0.14
1400	6	1	1.28	0.86	0.74	0.61	0.41
	10	1	0.87	0.58	0.50	0.41	0.28
	20	1	0.52	0.35	0.30	0.25	0.16
1600	6	1	1.47	0.98	0.85	0.69	0.46
	10	1	1.00	0.67	0.58	0.47	0.32
	20	1	0.59	0.40	0.34	0.28	0.19
1800	6	1	1.65	1.10	0.95	0.78	0.52
	10	1	1.12	0.75	0.65	0.53	0.36
	20	1	0.67	0.45	0.39	0.32	0.21
2000	6	1	1.83	1.23	1.06	0.87	0.58
	10	1	1.25	0.84	0.72	0.59	0.40
	20	1	0.74	0.50	0.43	0.35	0.23
2200	6	1	2.02	1.35	1.16	0.95	0.64
	10	1	1.37	0.92	0.79	0.65	0.43
	20	1	0.82	0.55	0.47	0.39	0.26
2400	6	1	2.20	1.47	1.27	1.04	0.70
	10	1	1.50	1.00	0.87	0.71	0.47
	20	1	0.89	0.60	0.51	0.42	0.28
2500	6	1	2.29	1.53	1.32	1.08	0.72
	10	1	1.56	1.04	0.90	0.74	0.49
	20	1	0.93	0.62	0.54	0.44	0.29

Table 4.2: Mass of dark mediator m_X , mass of dark pion $m_{\pi_{dark}}$, and Yukawa coupling constant κ_0 tested in the flavor-aligned coupling scenario. Each κ_0 corresponds to a specific $c\tau_{\pi_{dark}}^{\max}$ of 5, 25, 45, 100, or 500 mm.

4.4 Signal Event Selection

This section will outline the different criteria that are used to determine whether a given proton-proton collision physics event contains an emerging jet signal or not.

4.4.1 Physics Object Reconstruction

There are a couple of physics objects reconstructed from the data collected that will be used throughout this analysis. These objects will then be used to determine key properties in physics events that help distinguish signal emerging jet events from background events. All of these objects are reconstructed by the ParticleFlow algorithm described in Section 3.3.2.

Primary Vertex The primary vertex is the position of the proton-proton collision in the detector. This object is used to determine the displacement of SM particles in the CMS tracker. It is important that the vertices are reconstructed correctly as the displacement of particles in emerging jets cause by the non-negligible distance travelled by the dark pions is a key property of emerging jets. Therefore, the primary vertices are required to pass a *Good* and *Not fake* flag, and must be reconstructed within 15 cm of the center of the CMS detector along the z -axis.

Tracks Tracks are physics objects that represent charged particles interacting with the detector. They hold particle properties such as location in the detector, momentum, and trajectory. In order to ensure that the tracks are well reconstructed, this analysis discards any tracks that have transverse momentum $p_T \leq 1$ GeV and which do not pass the *high purity* quality flag. Tracks become associated to jets if they are within a $\Delta R < 0.8$ of angular separation ($\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$) between the track and central location of the jet. Tracks that are associated to the top four leading jets in p_T are used to compute the prompt track fraction, given by

$$PTF = \frac{N_{trk}(|z_{trk} - z_{PV}| < 0.01\text{cm})}{N_{trk}} \quad (4.4)$$

where N_{trk} is the number of tracks associated to the jets that may or may not pass additional criteria, and z is the position of the track or primary vertex PV along the z -axis. Figure 4.5 shows

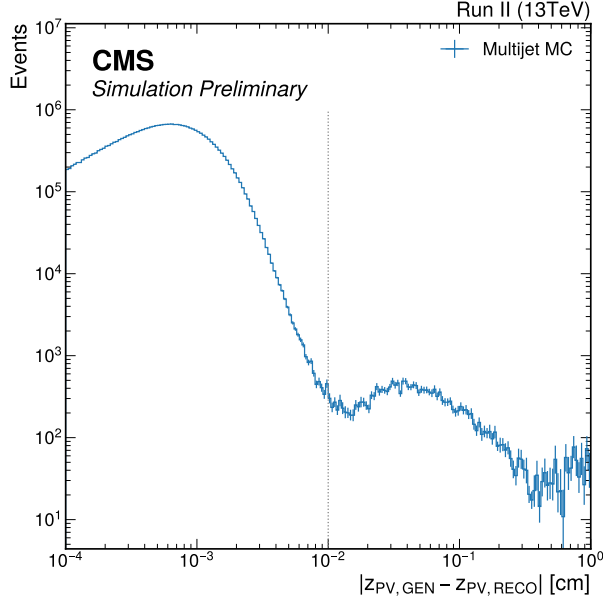


Figure 4.5: z residual between the true and reconstructed primary vertex for the simulated JetHT dataset. The black dashed line at 0.01 cm represents a cut off where vertices are deemed poorly reconstructed.

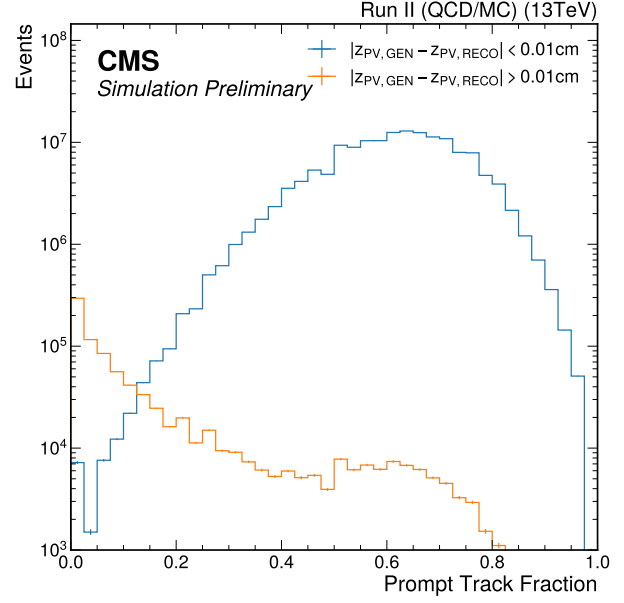


Figure 4.6: The prompt track fraction for vertices that are deemed poorly reconstructed (orange) versus all other vertices (blue). The blue and orange cross close to 0.1, making this a good threshold for removing events with poorly reconstructed vertices.

the difference of the true and reconstructed z value of the primary vertex, indicating errors in reconstruction, and Figure 4.6 shows the PTF distribution for the different regions of Figure 4.5. It can be seen that cutting events with $PTF > 0.1$ does a good job of removing events with poorly reconstructed vertices while maintaining a high efficiency for events with well reconstructed primary vertices.

Jets Jets represent a cone of particles within a certain angular separation. This analysis uses jets with a cone size of $R = 0.4$ and with jet energy scale corrections [60] applied. All jets studied are required to have large momentum $p_T > 100$ GeV, be fully contained within the tracker volume with $|\eta| < 2.0$, and pass a jet ID requirement recommended by the CMS collaboration [61]. The jets must also have no one track associated to them that makes up greater than 60% of the full jet p_T to help remove jets created from secondary interactions with few tracks.

4.4.2 Emerging Jet Tagger

Each emerging jet signal event is expected to have two emerging jets, as shown in Figure 4.2. In order to determine whether a jet is an emerging jet or not, a jet tagger is developed which takes in jet and associated track properties and assigns (or “tags”) the jet as signal or background. Two separate taggers are created for each coupling scenario, described in Section 4.2, as the signature for the flavor-aligned coupling is more complicated than for the unflavored coupling.

4.4.2.1 Cut-Based Tagger Overview

In the previous emerging jets analysis [62], emerging jet tagging was done by selecting subsets of jets using 1-dimensional jet-level variable cuts. This “cut-based” method was redone in this analysis and will be used to benchmark a new tagger using machine learning techniques.

For the unflavored scenario, the tracks associated to the jets⁴ are used to compute the following jet-level variables [62]:

- $\langle |d_{xy}| \rangle$: Tracks produced through dark pion decay tend to have larger displacement (in the xy -plane) with respect to the primary vertex compared to tracks from SM processes, as shown in Figure 4.7.
- α_{3D} : This is the p_T -weighted prompt track fraction, defined as

$$\alpha_{3D} = \frac{\sum p_T (D_N < D_{N,cut})}{\sum p_T}, \quad (4.5)$$

and measures the fraction of tracks in the jet originating from the primary vertex. D_N is a measure of the displacement significance and is defined as

$$D_N = \sqrt{\left(\frac{d_z}{0.01\text{cm}} \right)^2 + \left(\frac{d_{xy}}{\sigma(d_{xy})} \right)^2} \quad (4.6)$$

⁴For the cut-based unflavored tagger only, the tracks are associated to the jets within $\Delta R < 0.4$ instead of 0.8. This follows from the previous analysis [62].

where d_z is the displacement measure along the z -axis and σ is the uncertainty given by the track fitting algorithm. Figure 4.8 shows how emerging jets do not have many tracks which originate from the primary vertex.

For the flavor-aligned scenario, the tracks associated to the jets are used to compute the following jet-level variables [62]:

- $n_{trk}^{d_{xy} > d_{xy, cut}}$: The number of tracks within the jet that are displaced at least $d_{xy, cut}$ is larger for emerging jets than SM jets, as shown in Figure 4.9.
- **Track girth**: The “track girth”, defined as

$$g = \frac{\sum (p_T^{trk} \times \Delta R_{trk})}{\sum p_T^{trk}} \quad (4.7)$$

where ΔR_{trk} is the angular separation between the track and center of the jet, exploits the

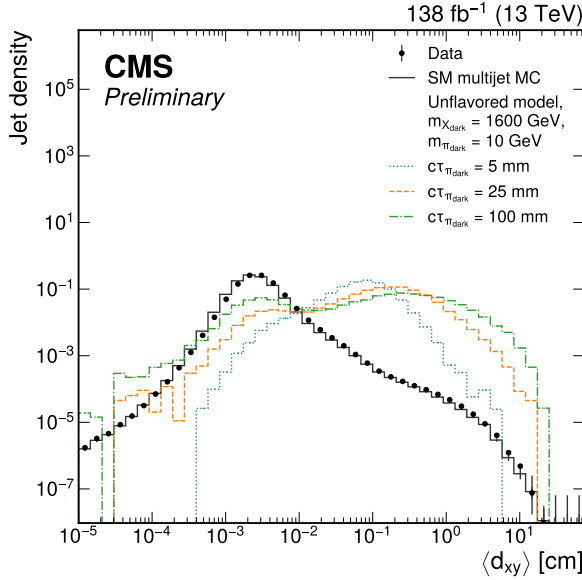


Figure 4.7: Median displacement in the xy -plane for different jet collections. The colored and dashed lines represent different emerging jet samples, black line is the simulated background, and black dots are the jets in JetHT data.

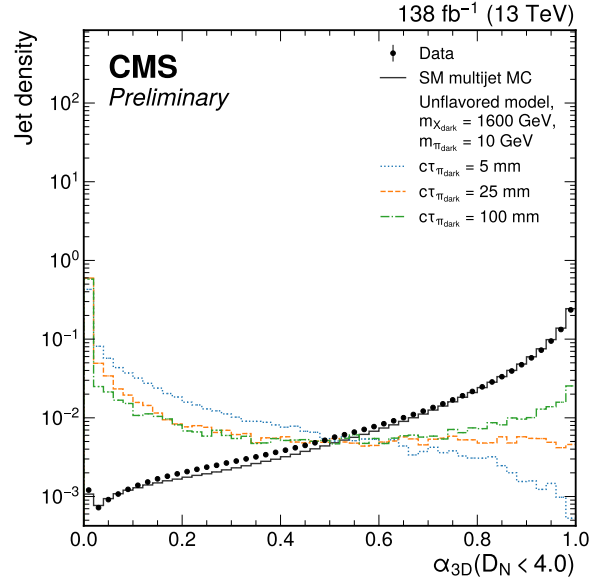


Figure 4.8: α_{3D} for different jet collections. The colored and dashed lines represent different emerging jet samples, black line is the simulated background, and black dots are the jets in JetHT data.

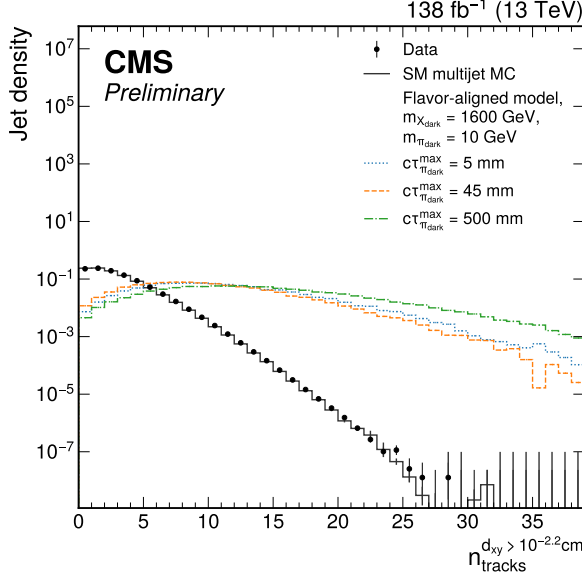


Figure 4.9: Number of tracks displaced in the xy -plane for different jet collections. The colored and dashed lines represent different emerging jet samples, black line is the simulated background, and black dots are the jets in JetHT data.

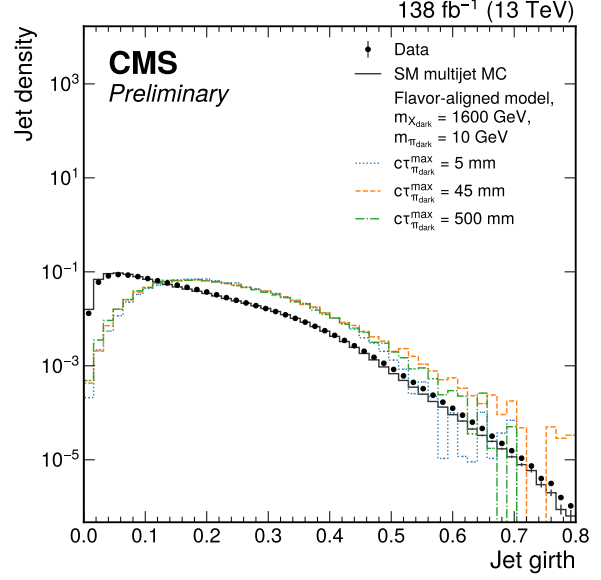


Figure 4.10: Track girth for different jet collections. The colored and dashed lines represent different emerging jet samples, black line is the simulated background, and black dots are the jets in JetHT data.

fact that emerging jets are more spread out than background jets due to the travel of the dark pions before decay. This distribution can be seen in Figure 4.10.

- τ_2/τ_1 : This variable is primarily used for pile-up jet rejection as it can reject jets with a softer (smaller p_T) jet within the angular vicinity of the main jet. τ_x is defined as

$$\tau_x = \frac{\sum p_{T,i} \times \min(\Delta R_{i,j})}{\sum 0.8 p_{T,i}}. \quad (4.8)$$

All of the above variable distributions may change based on the emerging jet parameters being looked at, so the cuts made on each variable is optimized per signal sample (see Section 4.4.4).

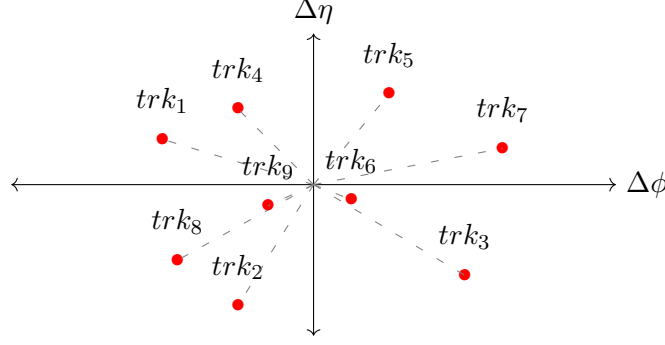


Figure 4.11: Graph representation of a jet object where each red point represents a unique track associated with the jet. The coordinates of each track are spatial coordinates $(\Delta\phi, \Delta\eta)$ with respect to the center (or vertex) of the jet.

4.4.2.2 Graph Neural Networks for Jets

Jets are built from a cluster of reconstructed particles spaced closely together within the CMS detector. The tracks associated to the jets hold properties that can be used to calculate jet attributes, like some of those used by the cut-based tagger in the previous section. Therefore, a jet can be represented as a point-cloud of tracks where the tracks are placed in the jet-space with respect to the center of the jet, and each point carries the properties that the track it represents holds. Ultimately, a graph representation of each jet can be built.

Figure 4.11 shows an example of what a jet graph representation looks like. Each track associated to the jet is placed in $(\Delta\phi, \Delta\eta)$ -space with respect to the center of the jet. The center of the jet is defined by where the jet vertex is reconstructed. Each track also carries information distinct to the track, such as its momentum and energy. The number of tracks within a jet is not fixed, so the graph can be populated with as many points as needed.

Graph neural networks (GNNs) are machine learning models that work well on graph-like data and can leverage information about each data point (node) and its relationship with the points next to it (edges) to learn about how the points collectively affect one another and tie together. GNNs are capable of making predictions on the node, edge, and graph level. These neural networks becoming more widely used within particle physics as much of the data collected by detectors in

this field are sparse and heterogenous and therefore can be difficult to represent in a structured and ordered format [63]. In this analysis, a graph neural network is used to learn about the track level relationship within jets to determine whether the graph represents an emerging jet or not.

4.4.2.3 GNN Tagger Development

This analysis draws its GNN structure inspiration from ParticleNet, a GNN built to classify jets represented by a collection of reconstructed particles [64]. In this model, the following track-level variables are used as inputs to the GNN:

Track coordinates There are two coordinates associated with each track that indicate where the track is relative to the center of the jet. These coordinates give the GNN taggers information on the spacing of the tracks within the jet and allows them to learn about the relationships between the neighboring tracks. The definition of each coordinate is given below with the distributions of the track coordinate shown found in Figure 4.12.

- $\Delta\phi$ is the angular separation between the track and the associated jet center represented along the ϕ axis of the CMS detector.
- $\Delta\eta$ is the angular separation between the track and the associated jet center represented along the η axis of the CMS detector.

Track features Each track in the jet is represented by a feature vector that includes 5 track-level variables. All of the features with unconfined ranges are transformed to a similar reduced range so that the learned GNN weights stay small and are therefore faster and more efficient to train. The definition of each feature is given below with the distributions of the track features shown found in Figure 4.13.

- ΔR . The scalar value of the angular separation of the track within the jet can be calculated with the relation

$$\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2} \quad (4.9)$$

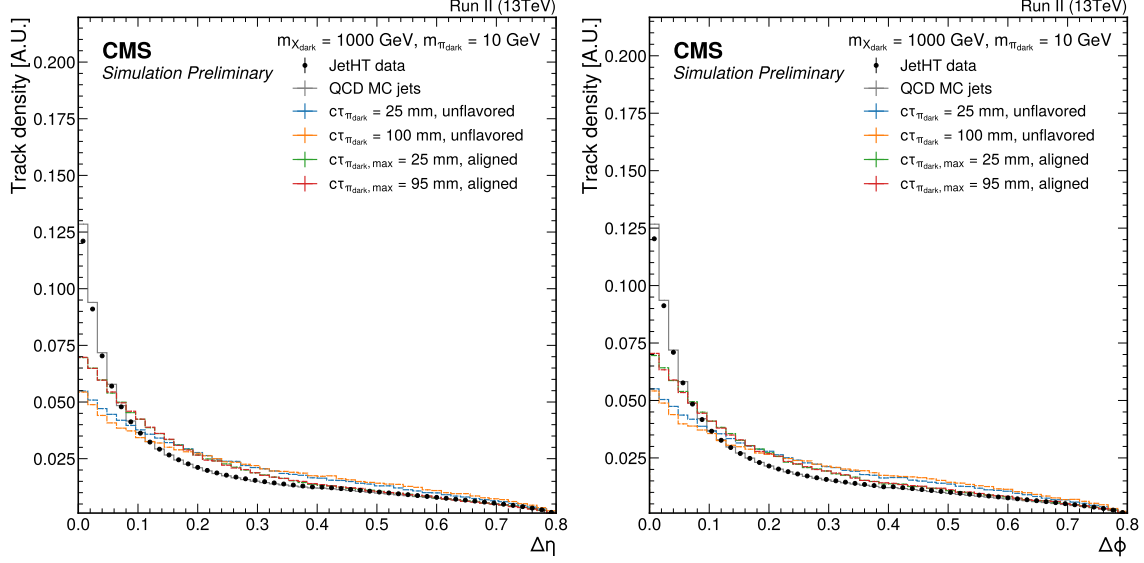


Figure 4.12: The $\Delta\eta$ and $\Delta\phi$ between the track and the associated jet center along two different detector axes for emerging jets (colored lines) and SM (QCD) jets (gray line). These variables make up the GNN track coordinates. All distributions are normalized to an area of 1.

where $\Delta\phi$ and $\Delta\eta$ are the coordinates mentioned above. Because of the nature of the hadronization and decay of the dark quark back into standard model particles, the tracks in an emerging jet are expected to be spread further from the center of the jet than for a SM jet.

- $\ln(p_T^{trk})$. Emerging jets are formed from the reconstructed tracks produced by two consecutive processes: the first is the hadronization in the dark sector from a dark mediator to dark pions, and the second is the decay dark pions to SM particles. Each process splits the full momentum of the jet into smaller pieces, therefore the p_T^{trk} of tracks in an emerging jet is smaller on average than the p_T^{trk} in a regular SM jet which only has the hadronization into SM particles process.
- $\ln\left(\frac{p_T^{trk}}{\sum_i p_T^{trk_i}}\right)$. Following the double process logic above, any one track from the second process is less likely to make up a high percentage of the full emerging jet p_T compared to a SM jet which has a single process. This variable is sometime called the “ p_T importance” as it describes what percentage of the jet p_T each track takes.

- **Transformed d_{xy} and d_z .** For the same reason the cut-based emerging jet taggers uses track displacement signatures, the GNN tagger uses the impact parameters of the tracks. The transformation applied to the variable displacement variables is defined as

$$T(x) = \text{sign}(x) \ln(x + 1) \quad (4.10)$$

to preserve the sign and continuity of the variable while reducing the variable input range.

More on this choice of transformation can be found in Appendix A.1.

Figure 4.14 shows the Pearson correlation coefficients [65] between each track feature. In general, p_T and p_T importance are positively correlated since a higher momentum track will also take up a larger percentage of the full jet momentum, and p_T and p_T importance are both negatively correlated with ΔR as higher momentum tracks will stay central in the jet while lower momentum tracks will be more spread within the jet. These correlations, however, vary between the different jet types. For example, the correlation between p_T and p_T importance is less strong in unflavored emerging jets than aligned emerging jets, which are both less strong than in SM jets. Since the GNN can pick up on variable correlations, this can be an important quality used to distinguish between jet types.

Since the signals in the unflavored and flavored emerging jet samples look different, as demonstrated by the correlation matrices, two different GNN taggers are trained for each coupling scenario. Both taggers have the same model and input structure. For each GNN, the dataset used in model development is composed of all emerging jets from the signal samples, and an equal number of randomly selected background SM jets from the QCD MC samples. The unflavoured emerging jet samples are used to train the unflavoured model, where all emerging jets from each sample are weighted equally. Similarly, the aligned emerging jets samples are used to train the aligned model where all emerging jets from each sample are weighted equally. 60% of the full dataset is used for training, 15% is used for validation throughout the training process, and the remaining 25% of the dataset is used to test the model performance. Each step will be outlined below.

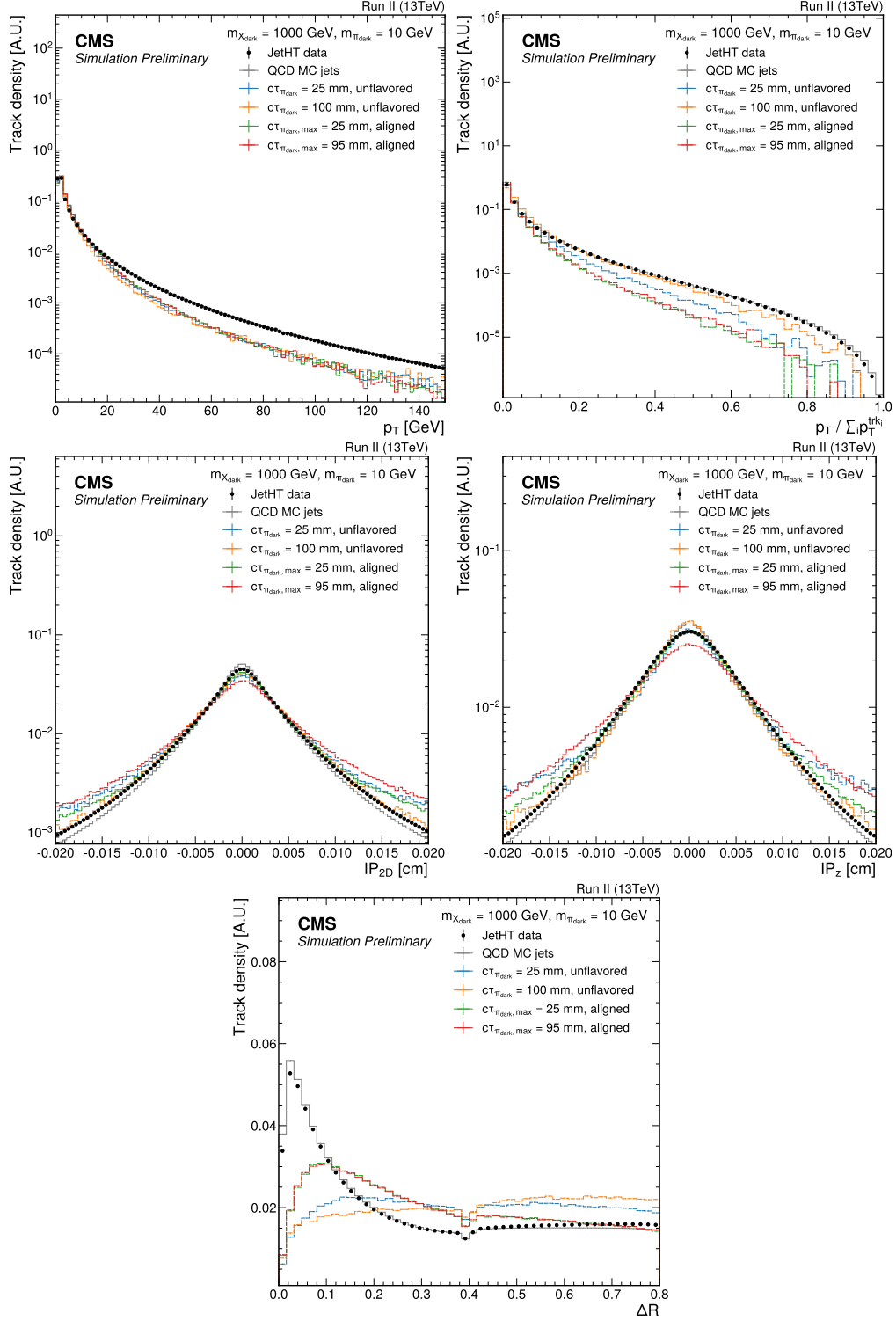


Figure 4.13: The track p_T , p_T importance, impact parameters IP_{2D} (d_{xy}) and IP_z (d_z), and ΔR for emerging jets (colored lines) and SM (QCD) jets (gray line). These variables are used for the GNN track features. All distributions are normalized to an area of 1.

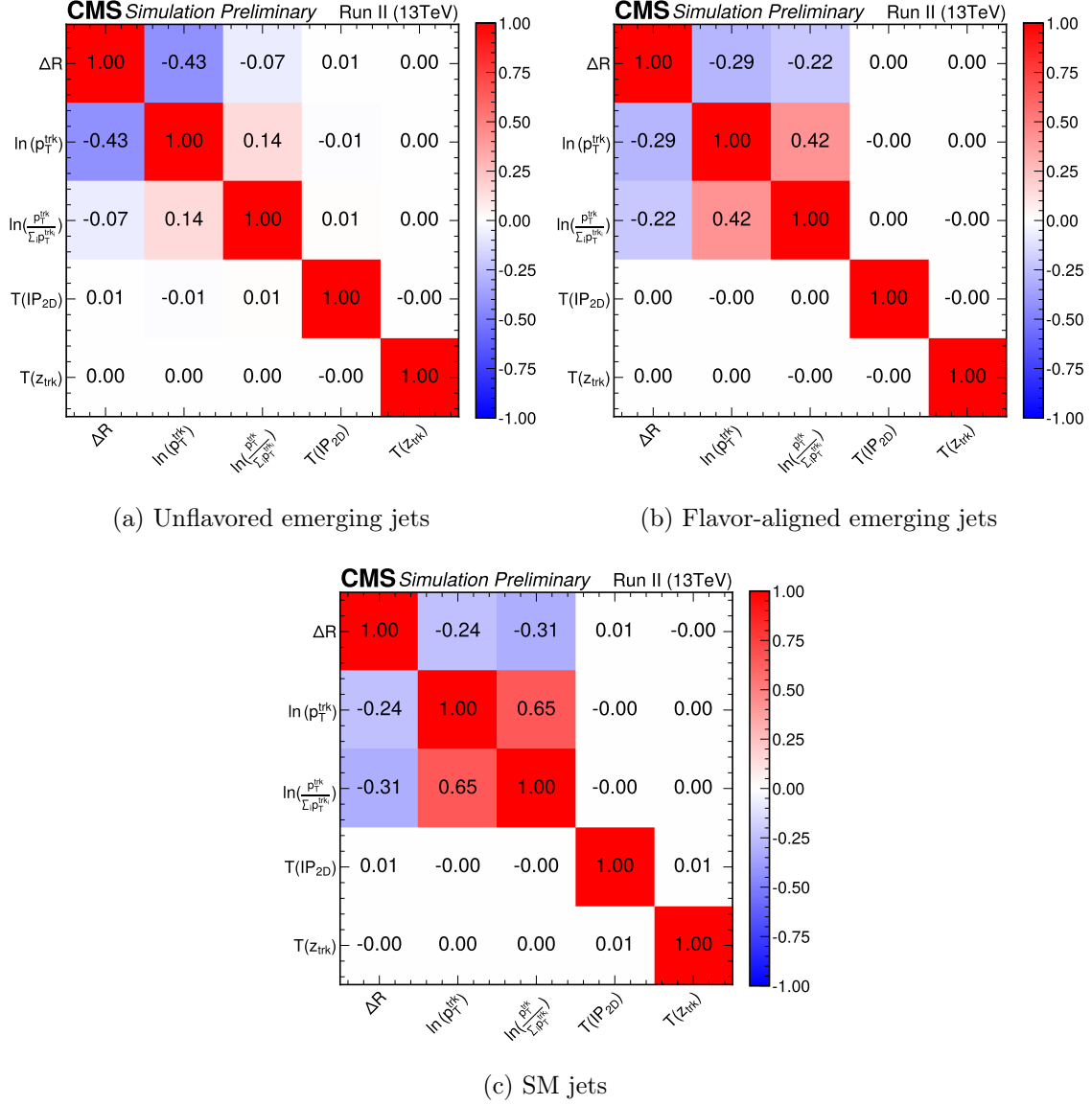


Figure 4.14: Pearson correlation coefficients between the GNN input features for unflavored emerging jets (a), flavor-aligned emerging jets (b), and SM jets (c). $T(IP_{2D})$ is the transform of d_{xy} , and $T(z_{\text{trk}})$ is the transform of d_z .

Parameter	Value	Description
<code>num_workers</code>	10	Number of threads used to load and process the dataset
<code>num_epochs</code>	20	Number of training epochs
<code>samples_per_epoch</code>	500000	Number of samples processed per epoch
<code>in_memory</code>	True	Keep loaded and preprocessed data in memory during training
<code>stop_tol</code>	5	Number of epochs to tolerate overtraining before early stopping
<code>optimizer</code>	ranger	Optimizer used for training
<code>start_lr</code>	5×10^{-3}	Start learning rate
<code>batch_size</code>	128	Number of samples in each batch

Table 4.3: Parameters used in training each GNN emerging jet tagger. All other parameters found at <https://github.com/cgsavard/weaver> are left at their default values.

During the training of a model, jet graphs are input into the algorithm and transformed via weights to output **score** variables, which measure the likelihood that a jet is an emerging jet. The score is a value between 0 and 1, where 0 represents a SM jet, and 1 represents an emerging jet with the highest probability. These GNN outputs are compared against the **truth tag** of each jet, which represents whether the reconstructed jet is truly an emerging jet or not. The truth tag of a jet is assigned a 1 (emerging jet) if at least 60% of the jet p_T comes from simulated dark mesons within the jet cone (which are produced in emerging jets from the hadronization of the dark quark), or a 0 (SM jet) otherwise. Since the truth tag uses information about dark mesons which will not interact with the detector, this is not a variable that can be reconstructed and can only be used to evaluate GNN performance in simulated data. The goal of training is to minimize the difference between the jet score and truth tag by updating the model weights to make the classifier more accurate. This is achieved when emerging jets produce scores closer to 1 and SM jets produce scores closer to 0. Each GNN is trained using the **Weaver** [66] ParticleNet training package.⁵ The training parameters used can be found in Table 4.3. Training was done on an NVIDIA P100 GPU and took roughly 2-3 hours per GNN.

⁵The forked repository used for training can be found at <https://github.com/cgsavard/weaver>

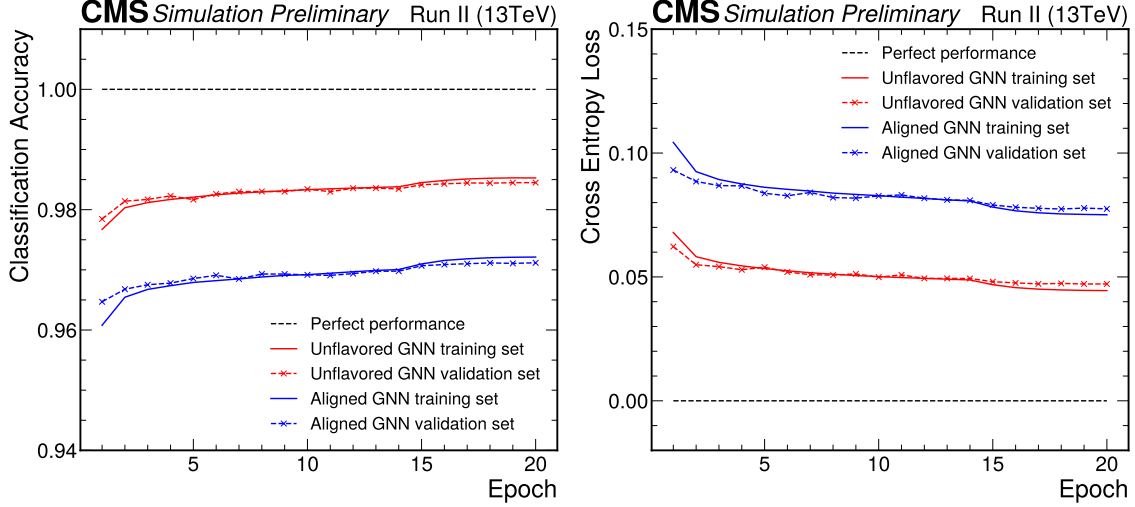


Figure 4.15: The classification accuracy and cross entropy loss of the unflavored and aligned GNNs for the training and validation sets after each training epoch.

When machine learning models perform well on training samples but cannot generalize to new data, the model is said to be “overtrained”. To avoid overtraining, the performance of a validation sample is compared to that of the training sample after each training epoch. An epoch is a training cycle consisting of calculating jet scores, comparing the score against the truth tags, and updating the model weights accordingly. Divergence in performance between the two samples indicates overtraining. Figure 4.15 shows comparisons between the training and validation samples for two different performance metrics: classification accuracy and cross entropy loss. The classification accuracy is the percent of jets that are correctly classified (where the classification is 1 if the score \geq is 0.5, or 0 otherwise) as emerging or SM jets. The cross entropy loss is defined as

$$\text{cross entropy loss} = -\frac{1}{N} \sum_{\text{jet } i=1}^N ((\text{truth tag}_i) \log(\text{score}_i) + (1 - \text{truth tag}_i) \log(1 - \text{score}_i)), \quad (4.11)$$

where N is the total number of jets tested. This is a measure of the GNN performance error. Neither of these measures show any indication of overtraining on either the unflavored or flavor-aligned GNNs.

The testing sample is used after training to measure the unbiased final GNN performance.

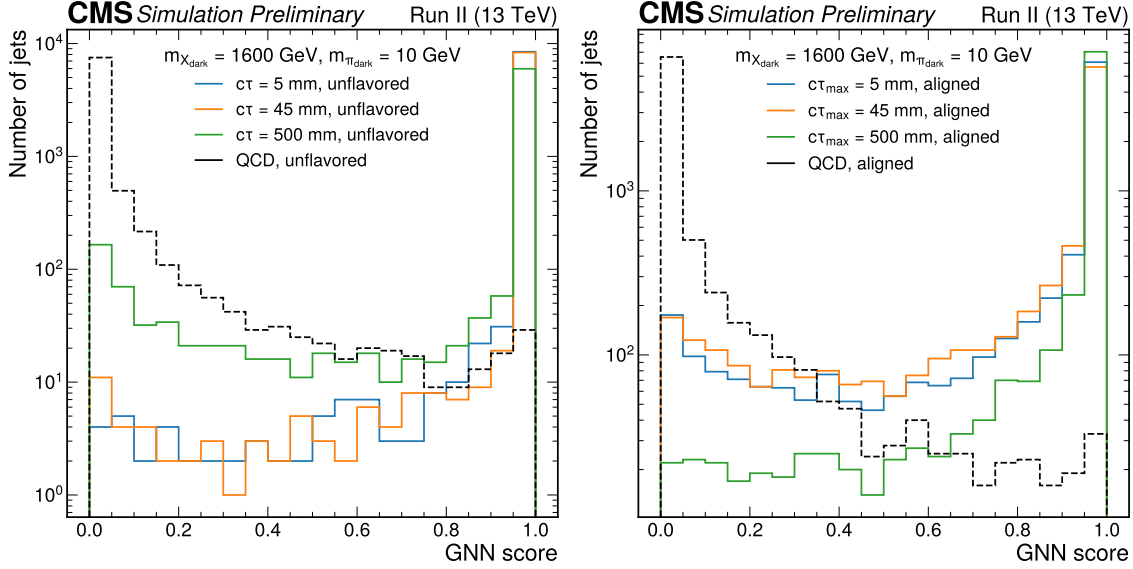


Figure 4.16: The output scores of the unflavored GNN tagger (left) and aligned GNN tagger (right) tested on 3 different unflavored and aligned emerging jet samples as well as QCD MC background.

The score distribution in the testing sample can be found in Figure 4.16 and shows a peak at 0 for background jets and a peak at 1 for emerging jets, as expected. Although the output of each GNN is a score, a threshold is used to determine the actual classification of each jet. The threshold is used as follows:

$$\text{classification} = \begin{cases} 1, & \text{if score} \geq \text{threshold} \\ 0, & \text{otherwise.} \end{cases} \quad (4.12)$$

The classification of each jet can then be compared against the truth tag to determine how accurate the GNN is.

A look at how performance of the tagger varies when selecting different thresholds is shown in Figure 4.17. The mistag rate is defined as the fraction of incorrectly classified SM jets, while the signal acceptance rate is defined as the fraction of correctly classified emerging jets. In general, the unflavored GNN tagger performs better on the unflavored samples than the aligned GNN tagger performs on the aligned samples. This is because there is much more variety in the emerging jets present in the aligned models due to the coupling of the dark quark to any down, strange, or bottom

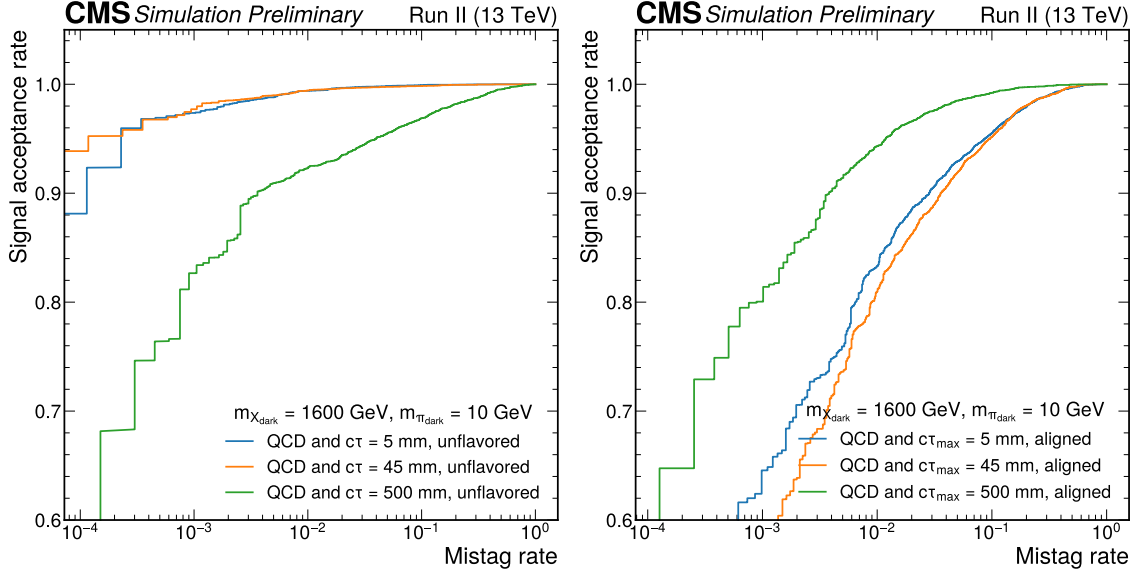


Figure 4.17: The signal acceptance (fraction of correctly tagged emerging jets) vs the mistag rate (fraction of incorrectly tagged background jets) of the unflavored GNN tagger (left) and aligned GNN tagger (right) tested on 3 different unflavored and aligned emerging jet samples as well as QCD MC background. Each point on the curve is a result of a different cut applied to the GNN output score.

quarks. We also see a decrease in performance at the more extreme $c\tau_{\pi_{\text{dark}}}$ values as the jets begin to decay closer to the limits of the CMS silicon tracker. At smaller $c\tau_{\pi_{\text{dark}}}$, emerging jets look more like SM jets and so they are harder to distinguish. At larger $c\tau_{\pi_{\text{dark}}}$, some jet information is lost as the dark pions decay back into SM particles at the edge of or beyond the tracker.

Feature importance is a key part of understanding what input features have a significant effect on model performance. Permutation feature importance manipulates the inputs by shuffling the values of one feature among all inputs and seeing how that shuffling affects the output. The output affects are measured by the change in the AUC metric (ΔAUC), defined as the area under the performance curves found in Figure 4.17, before and after feature manipulation. Figure 4.18 shows the ΔAUC after permuting each feature individually. For both emerging jet models, the track d_{xy} feature has the largest effect on the AUC metric, and is therefore deemed the most important feature. The next two features of importance are the track p_T and ΔR which have a strong correlation as shown in Figure 4.14. This can be indicative that this correlation is important, as when either of

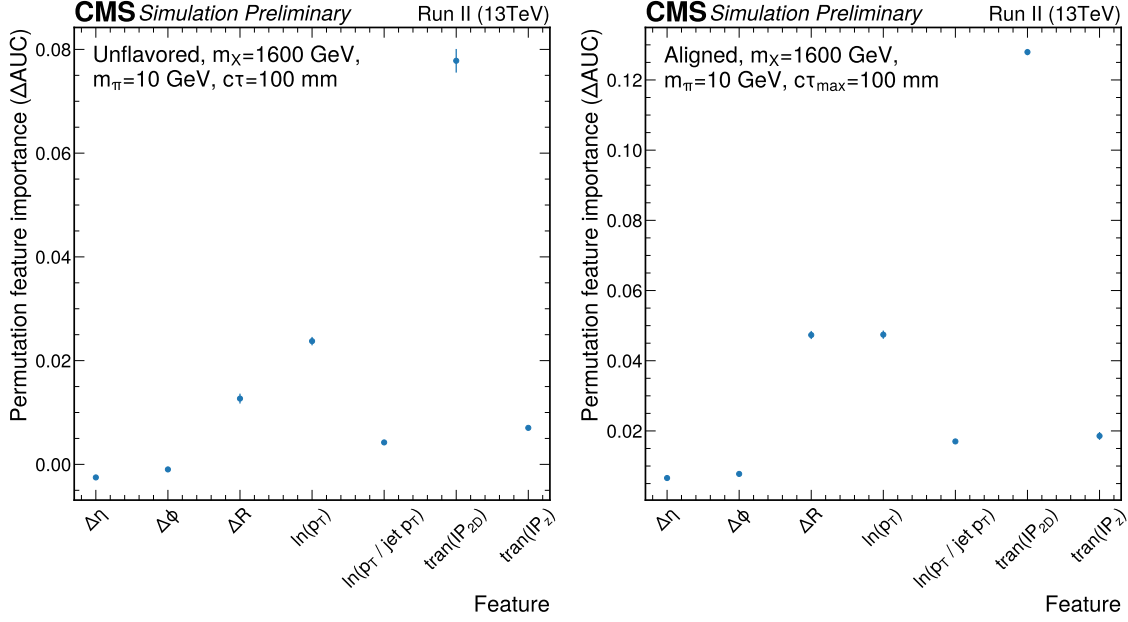


Figure 4.18: The permutation feature importance, measured as the difference in AUC when each feature is manipulated, of the unflavored (left) and aligned (right) GNN taggers. A larger importance value indicates a higher importance.

these features changes, the correlation is broken and the performance of the classifier degrades.

There are pros and cons to using machine learning techniques applied to a dark matter search. The indisputable gain in tagging performance, as shown in Figure 4.19, makes a great case for replacing all cut-based methods to a machine learning method. In general, an increase in signal sensitivity is expected when networks are trained for specific tasks. However, because machine learning models are trained on specific signal samples, they may not generalize as well to other similar problems when compared to a cut-based method which makes selection motivated by the general physical processes. In a dark matter search, for example, it is not known whether the theory being tested is true. The true theory could potentially be emerging jets with free parameters that are not being tested in this analysis, or another complementary theory that may share signal similarities. More information about the generalizability of the GNN tagger can be found in Appendix A.2

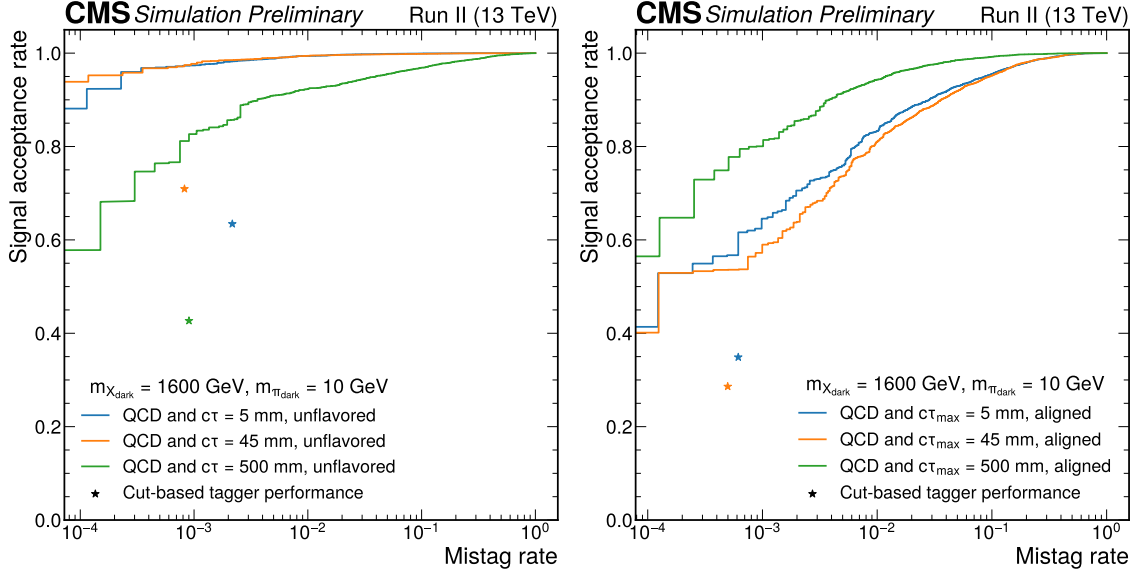


Figure 4.19: The signal acceptance (percent of correctly tagged emerging jets) vs the mistag rate (percent of incorrectly tagged background jets) of the unflavored GNN tagger (left) and aligned GNN tagger (right) tested on 3 different unflavored and aligned emerging jet models as well as QCD MC background. Each point on the curve is a result of a different cut applied to the GNN output score. The stars represent the cut-based tagging method performance. The green star on the right plot has a mistag rate of $< 10^{-4}$ and so it is out of range for the plot.

4.4.3 Event-Level Variables

On an event level, each emerging jet signal is expected to produce four high p_T jets, leading to a high H_T , with at least two emerging jets tagged using the taggers described in Section 4.4.2. Figure 4.20 shows the difference between emerging jet events and SM background events for H_T and p_T of the two highest (leading and subleading) p_T jets in the event. Given the difference in distributions between the signal and background shown, these variables are useful for selecting signal events. Since at least four jets are present in potential signal events, the top four leading jet p_T spectrums will be used for selection.

4.4.4 Selection Criteria Optimization

The actual cuts that are made on each variable described in the sections above to classify an event as signal or background may differ based on the emerging jet sample being tested. In order

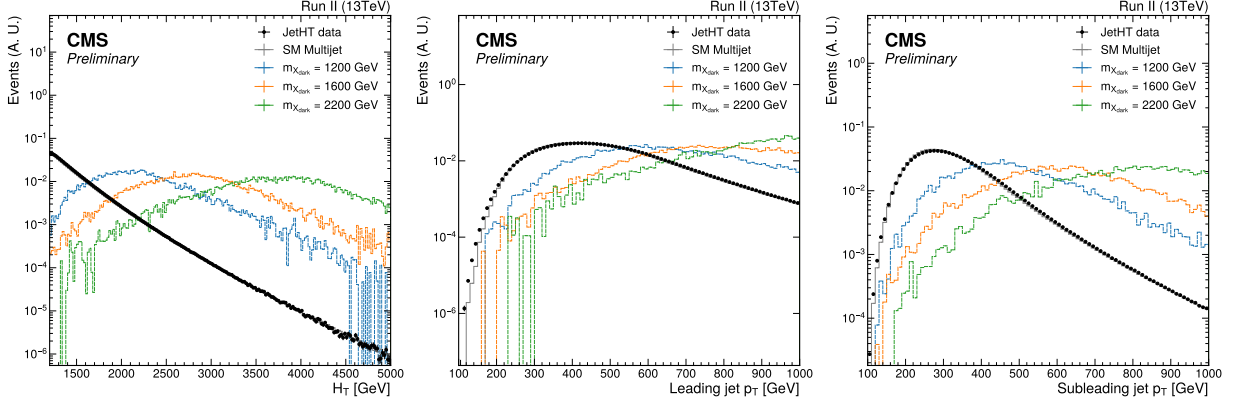


Figure 4.20: The distribution of H_T (left), leading jet p_T (middle) and subleading jet p_T (right) for SM multi-jet events and signal events of the unflavored model with $m_{\pi_{dark}} = 10$ GeV, $c\tau_{\pi_{dark}} = 25$ cm and varying dark mediator masses.

to determine where the cuts for each sample should be made, a scan over multiple cut options is performed and a significance measure which optimizes signal sensitivity and suppresses background, defined as

$$\sigma = \frac{S}{\sqrt{S + B + (0.1)^2 B^2}}, \quad (4.13)$$

is calculated. S and B are the number of signal and background events, respectively, that pass the cuts being tested, and 0.1 is an estimated background event uncertainty. This metric favors background rejection as it is deemed better to reject a real signal event than to accept a fake background event. Whichever set of cuts, also called “cutset”, that achieves the highest σ value is deemed the optimal cutset. There is one optimal cutset per signal sample.

Many of the optimal cutsets are either the same or very similar for signal samples that are close together in the $(m_X, m_{\pi_{dark}}, c\tau_{\pi_{dark}})$ space. Therefore, the cutsets are then clustered together into a final, smaller set of cutsets. Since the background estimation methods described in Section 4.5 have to be done on each cutset used by the analysis, it is beneficial to minimize the number of cutsets being used in order to minimize compute time and analysis complexity. Of course, this must be balanced with a loss in performance if the cutset assigned to a signal sample is no longer the optimal one. This is why this analysis ensures that the final cutsets still achieve $> 90\%$ of the maximal σ

value from the optimal cutsets.

In the cut-based method, again following what was done by the previous version of this analysis [62], the optimal cutsets were clustered manually. The GNN method, however, clustered the optimal cutsets using another machine learning technique, k -means clustering. k -means clustering, as opposed to clustering manually, helps minimize human bias present when selecting cutsets simply by looking at the collection of optimal cutsets and making human judgment on what k cutsets should be chosen to best represent the optimal cutset space.

k -means clustering is a machine learning algorithm that assigns all points in an N dimensional space to one of k clusters determined by which cluster centroid each point is closest to. The cluster centroids are chosen to minimize the distance between each point and respective cluster center [67]. In the case of clustering the optimal cutsets, each cutset variable $\{H_T, p_T^{jet1}, p_T^{jet2}, p_T^{jet3}, p_T^{jet4}, N_{EMJ}, \text{score}\}$ is a feature which the cutsets will be clustered along. Since the distances used to determine the clusters are highly dependent on the distribution of the data along a given feature axis, min-max scaling is used to reduce each axis distribution into the range of 0 to 1 and therefore weight the importance of each feature similarly. Min-max scaling is defined as

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}, \quad (4.14)$$

where x is the old value, x' is the updated value, and X represents the full x distribution.

The optimal cutsets associated with the unflavored and aligned signal samples were clustered separately. To determine the optimal k value to use, the k -means algorithm was run multiple times scanning a k of 1 to 8 clusters and the sum of the squared distanced between the points and cluster centroids was calculated to determine where the metric stopped dropping quickly. This is sometimes called the “elbow method”, and the results of scanning each k can be found in Figure 4.21 [68]. It was determined that $k=3$ clusters effectively represented the data points for both the unflavored and aligned emerging jet models. A visual representation of the clustering of the optimal cutsets for the unflavored and flavour-aligned models is shown in Figure 4.22.

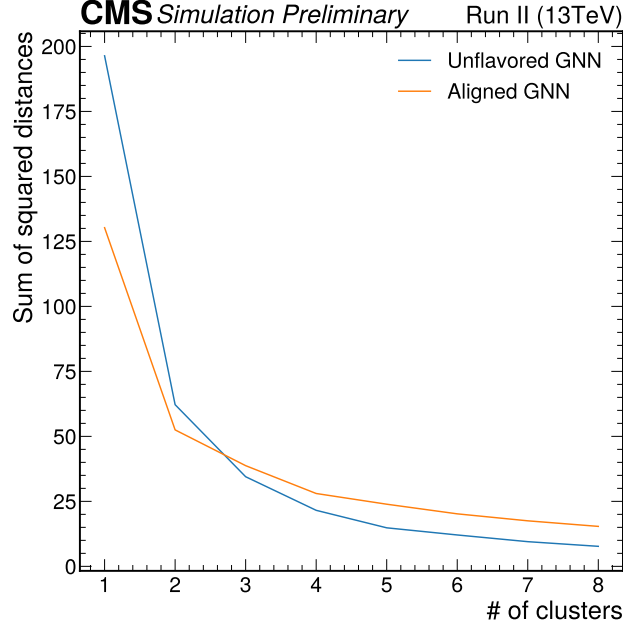


Figure 4.21: The sum of squared distance for varying k (number of clusters) when clustering the unflavored and aligned emerging jet models. k was determined using the “elbow method” to be 3 for both emerging jet model types.

The final cutsets for the unflavored and flavored cut-based methods can be found in Tables 4.4 and 4.5, respectively. The unflavored cutsets are all named “dr4 set $\{i\}$ ”, and the aligned cutsets are all named “dr8 set $\{i\}$ ”. The final cutsets for the GNN methods can be found in Table 4.6. The unflavored cutsets are all named “GNN uset $\{i\}$ ”, and the flavored cutsets are all named “GNN aset $\{i\}$ ”.

Figures 4.23 and 4.24 show the signal acceptance (percent of signal events that are selected by the final cutsets) that the different GNN cutsets achieve for the different emerging jet models. In general, the efficiency is higher for the unflavored models than for the aligned models, which is as expected given the flavor-aligned model is a more complex signal than the unflavored model. For the unflavored models, the efficiency drops significantly at low $c\tau_{\pi_{dark}}$, where emerging jets are more prompt in the detector and therefore more like SM jets, and high $c\tau_{\pi_{dark}}$, where jet information becomes partially lost due to some tracks leaving the tracker volume without leaving any trace. As expected, there is little difference in efficiency for $m_{\pi_{dark}} = 10$ or 20 GeV. For the aligned

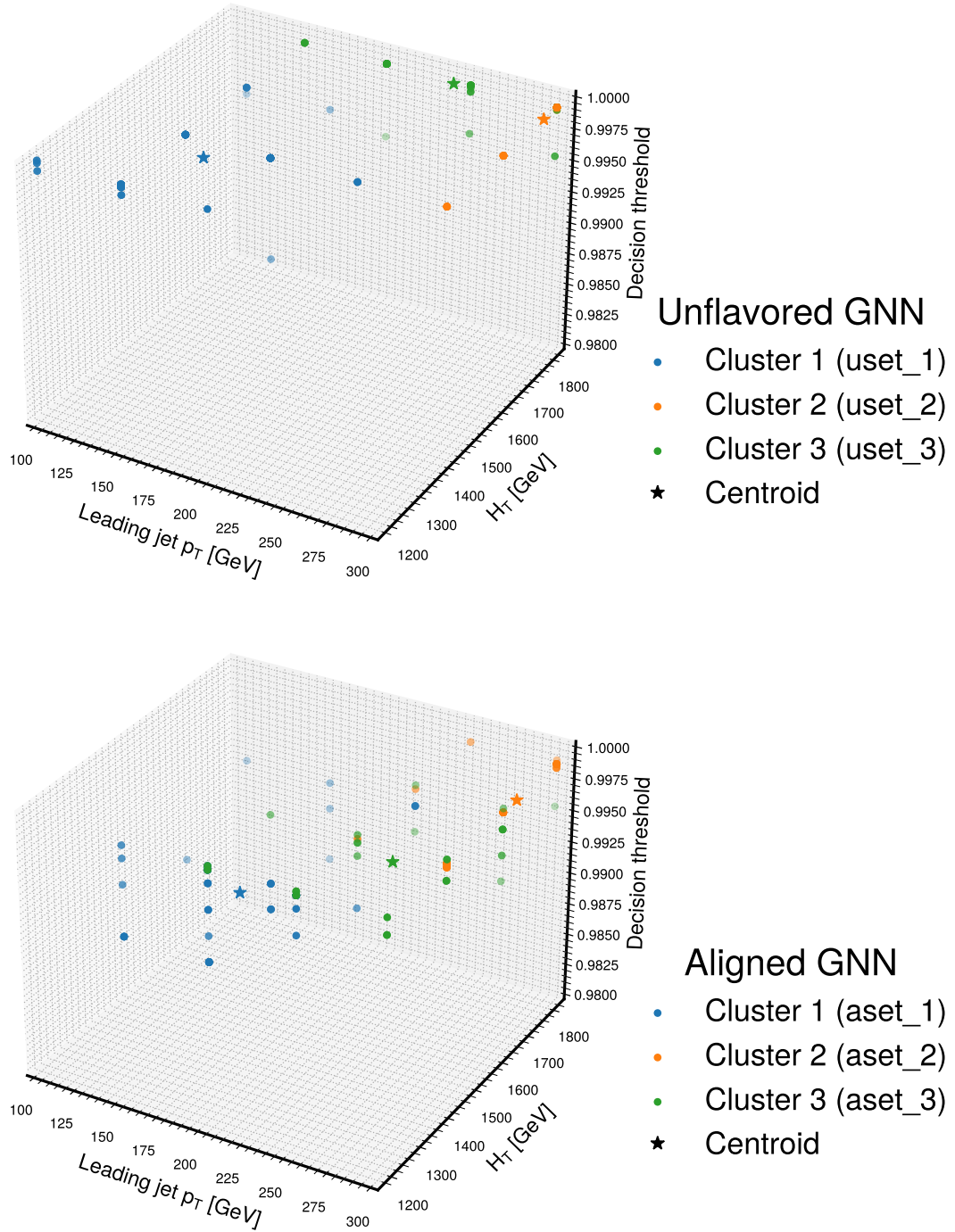


Figure 4.22: The resulting clusters for the unflavored (top) and aligned (bottom) emerging jet models using $k=3$ k -means clustering. Each round marker is an optimal significance cutset that is clustered with other markers of the same color. The cluster centroids are determined using the points within each cluster and are indicated with a star, along with a label by the color like “uset 1” which indicates which final cutset each signal model is assigned to. The decision threshold represents the cut made on the GNN output score.

Name	H_T [GeV] ($>$)	Jet p_T [GeV] ($>$)	$\langle d_{xy} \rangle$ [cm] ($>$)	D_N ($<$)	α_{3D} ($<$)
dr4 set 1	1600	275, 250, 250, 150	$10^{-1.6}$	4	0.25
dr4 set 2	1600	200, 200, 150, 150	$10^{-1.4}$	8	0.25
dr4 set 3	1600	200, 150, 100, 100	$10^{-1.2}$	8	0.25
dr4 set 4	1500	200, 150, 100, 100	$10^{-1.2}$	12	0.15
dr4 set 5	1200	200, 150, 100, 100	$10^{-1.0}$	12	0.15

Table 4.4: Final cutsets selected for the unflavored cut-based method. The second and third columns are event-level variables, and the last three columns are jet-level variables used to tag at least two emerging jets within the event.

Name	H_T [GeV] ($>$)	Jet p_T [GeV] ($>$)	g ($>$)	$d_{xy, cut}$ [cm]	$n_{track}^{ d_{xy} > d_{xy, cut}} (>)$	$\tau_2/\tau_1 (>)$
dr8 set 1	1500	200, 150, 100, 100	0.5	$10^{-2.2}$	12	0.5
dr8 set 2	1800	250, 250, 200, 200	0.5	$10^{-2.2}$	12	0.5
dr8 set 3	1200	275, 250, 250, 200	0.5	$10^{-2.2}$	12	0.5
dr8 set 4	1500	275, 250, 250, 100	0.5	$10^{-2.3}$	14	0.5
dr8 set 5	1800	200, 150, 100, 100	0.5	$10^{-2.4}$	14	0.5

Table 4.5: Final cutsets selected for the flavor-aligned cut-based method. The second and third columns are event-level variables, and the last four columns are jet-level variables used to tag at least two emerging jets within the event.

Name	H_T [GeV] ($>$)	Jet p_T [GeV] ($>$)	GNN score ($>$)
GNN uset 1	1350	170, 120, 120, 100	0.9997
GNN uset 2	1750	300, 260, 250, 250	0.9998
GNN uset 3	1800	240, 180, 180, 100	0.9996
GNN aset 1	1300	200, 140, 120, 100	0.9953
GNN aset 2	1650	300, 250, 200, 200	0.9993
GNN aset 3	1400	270, 220, 220, 120	0.9983

Table 4.6: Final cutsets selected for the unflavored GNN method (top 3) and flavor-aligned GNN method (bottom 3). The second and third columns are event-level variables, and the last column is a jet-level variables used to tag at least two emerging jets within the event.

models, the $c\tau_{\pi_{dark}}$ range is smaller and therefore there are not large drops in efficiency seen at the edges of the $c\tau_{\pi_{dark}}$ range. Instead, the efficiency when $m_{\pi_{dark}} = 6$ GeV is a bit higher than when $m_{\pi_{dark}} = 10$ or 20 GeV as dark pion decays to b-quarks, which produce more prompt emerging jets that are therefore harder to distinguish from SM jets, are not yet accessible in this mass (seen in Figure 4.4).

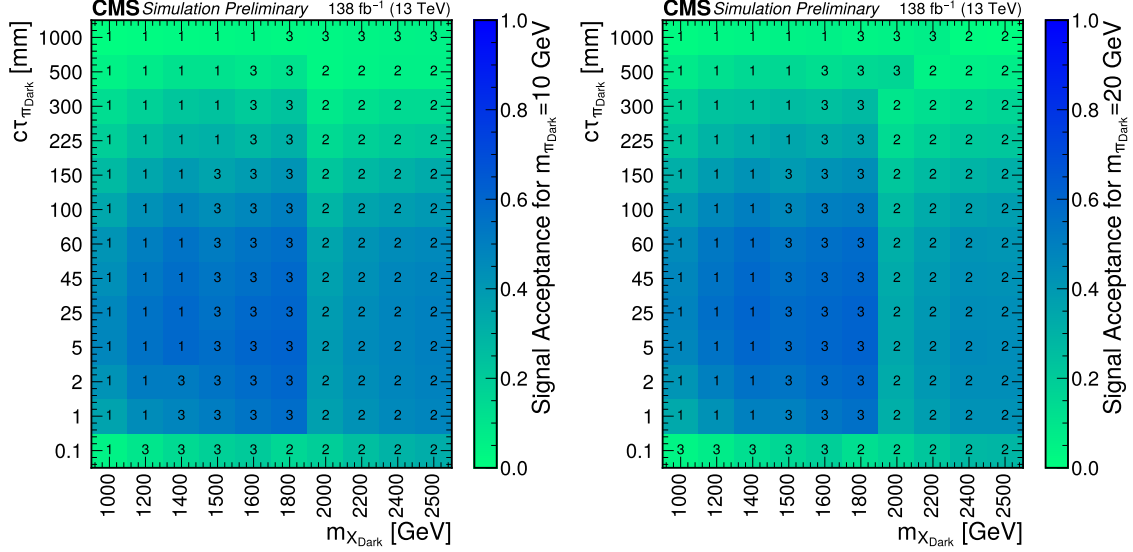


Figure 4.23: The signal selection efficiency for the GNN unflavored model with $m_{\pi_{dark}} = 10$ GeV (left) and $m_{\pi_{dark}} = 20$ GeV (right). The color of each cell is the selection efficiency, while the number in each cell indicates the corresponding cut set used for that signal model.

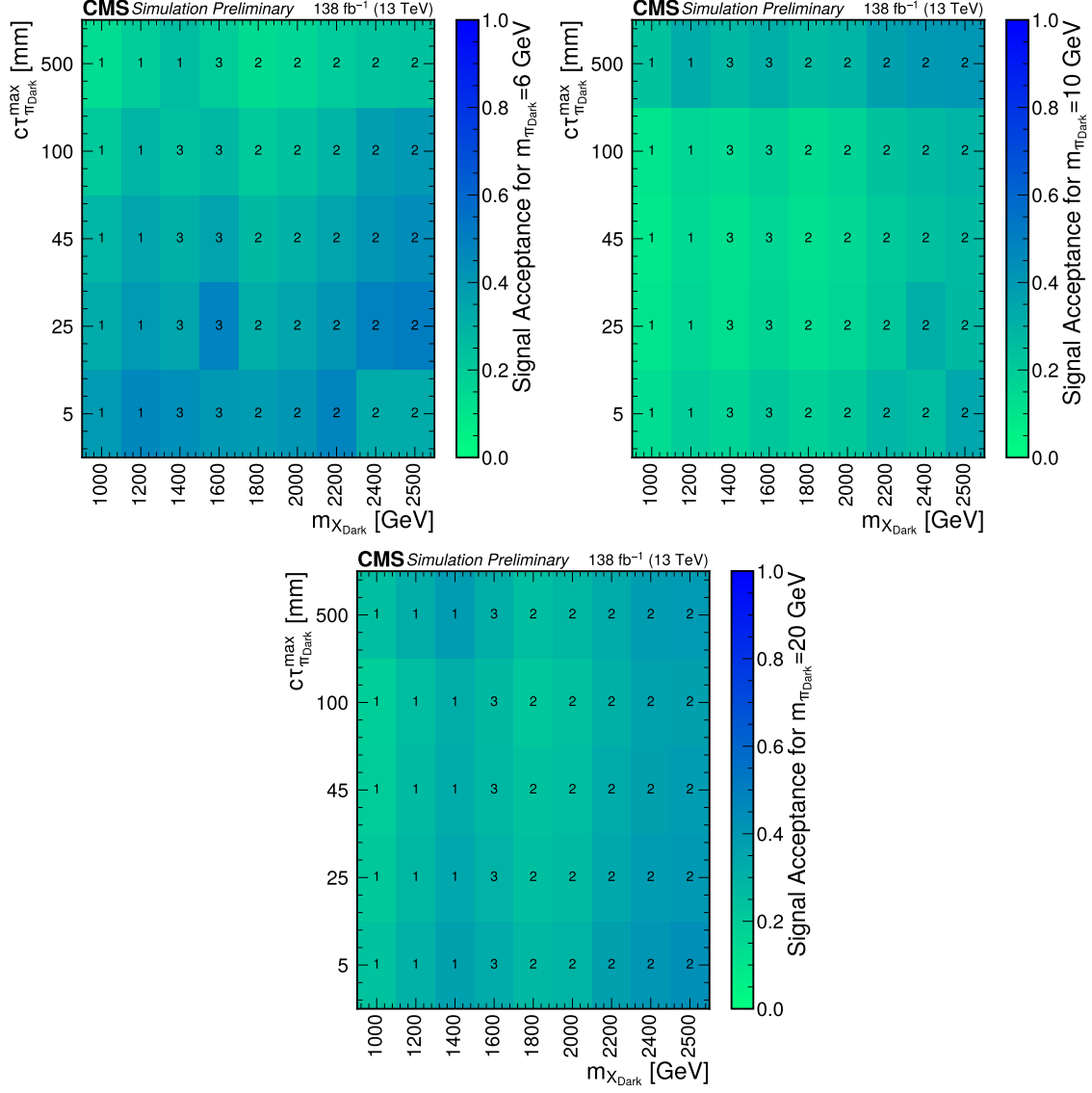


Figure 4.24: The signal selection efficiency for the GNN flavor-aligned model with $m_{\pi_{\text{dark}}} = 60 \text{ GeV}$ (top left), $m_{\pi_{\text{dark}}} = 10 \text{ GeV}$ (top right), and $m_{\pi_{\text{dark}}} = 20 \text{ GeV}$ (bottom). The color of each cell is the selection efficiency, while the number in each cell indicates the corresponding cut set used for that signal model.

4.4.5 Uncertainties

Each cutset defined in Tables 4.4, 4.5, and 4.6 is used to count the number of potential signal events that enter the signal region (SR, the region defined by the cutsets shown in Tables 4.4, 4.5, and 4.6). As the cutsets are sensitive to reconstruction and measurement error, systematic uncertainties due to these effects are included. These errors are used to vary the simulated signal samples to calculate the change in signal acceptance. Many of these systematic uncertainties have already been studied by simulation experts in detail and the uncertainty values are applied directly without additional study.

Each systematic uncertainty is listed briefly below.

- **Simulated track modeling** Track-level variables in simulation have varying distributions from those in data collected from 2017 and 2018, the largest difference due to mis-modeling the impact parameters d_{xy} and d_z where the distributions in simulation are narrower than in data. To mitigate this, the d_{xy} and d_z distributions are smeared using a Gaussian function centered about 0 such that the smeared simulated distributions match those seen in data. This smearing is then applied to each impact parameter to assess the signal acceptance affects.
- **Luminosity** The uncertainty on the luminosity measurement for each data collection year is between 1.2 – 2.5 % [34–36].
- **Pileup re-weighting** The uncertainty on the number of proton-proton interactions per bunch crossing is 4.6%.
- **Trigger efficiency** The H_T trigger efficiency between data and simulation does not match perfectly. The ratio of these different efficiencies are therefore used as scale factors applied to the simulated H_T to see how signal acceptance is affected.
- **Jet energy correction and resolutions** The jet energy and p_T is varied by the recommended jet energy scales [69], which also propagates to changes in event H_T .

- **PDF variation** The parton distribution function (PDF) is the momentum probability density of the partons (gluons and quarks) in the protons, and therefore affect what occurs in a collision. This distribution is calculated in the PYTHIA [58] event generator and compared to the distribution set by NNPDF [70] to determine an uncertainty. The uncertainty is applied on an event-level using weights.
- **QCD matrix element scales** The QCD matrix describes the QCD interactions between the different gluons and quarks. The matrix elements are dependent on a renormalization scale μ_r , which is related to the strong interaction coupling strength, and factorization scale μ_f , which separates the short- and long-distance QCD behaviors. The uncertainty on these scales is determined in PYTHIA by multiplying each scale by a factor of 0.5 and 2 and calculating the relative probability of the generated events with each change. The uncertainty is applied on an event-level using weights.

Table 4.7 shows a summary of each of these uncertainties for both the cut-based and GNN signal selection methods. Each uncertainty is generally around 1%.

4.5 Background Estimation

The goal of background estimation is to estimate the number of background events (non-emerging jet events) expected to pass into the SR (N_{SR}) given some error in the event selection methods. For example, the GNN does not classify emerging jets and background jets correctly 100% of the time; this mistag, or incorrect classification, can lead to background events containing two SM jets tagged as emerging jets which may cause the event selection method to label a SM event as an emerging jet event. If N_{SR} is well estimated, then a comparison can be made with the true number of events that pass into the SR to determine whether the difference between these values is statistically significant enough to conclude emerging jet events were actually present in the data. In order to forgo simulation uncertainties as the SM is challenging to simulate accurately, the background estimation method only uses data after its development and validation in simulation.

Uncertainty source (%)	Cut-based				GNN			
	Unflavored		Flavor-aligned		Unflavored		Flavor-aligned	
	mean	std.	mean	std.	mean	std.	mean	std.
MC track modeling	0.2	0.3	1.4	1.8	0.3	0.8	0.5	0.6
Luminosity	1.8	0.6	1.8	0.6	1.8	0.6	1.8	0.6
Pileup reweighting	1.6	1.4	1.4	1.2	0.9	0.8	1.0	0.9
Trigger efficiency	0.3	0.1	0.3	0.1	0.3	0.1	0.3	0.1
JEC	1.0	1.3	0.8	0.7	1.3	0.9	0.7	0.4
JER	0.3	0.4	0.3	0.3	0.2	0.3	0.2	0.1
PDF	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1
μ_R, μ_F	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1	< 0.1

Table 4.7: Signal systematic uncertainties summarized for the cut-based (left) and GNN (right) selection methods. Each value is either a mean or standard deviation (std.) of the set of cutsets using the same tagger type.

4.5.1 Estimation Calculation

The first step in estimating N_{SR} is choosing a control region (CR) that is orthogonal to the SR but has events with the same physical properties as those expected in the SR. The CR is also chosen such that the ratio of signal to background events present in the CR is small. Thus, the CR events can be scaled to determine the number of background events expected in the SR using scale factors SF . For each CR event, a corresponding SF estimates the ratio of how often a similar event would be selected into the SR as opposed to the CR. The N_{SR} is then defined as

$$N_{\text{SR}} = \sum_{a \in \text{CR events}} SF_a. \quad (4.15)$$

This analysis has multiple SRs, and the background estimation will be run on each one separately. Each cutset presented in Tables 4.4, 4.5, and 4.6 defines a SR as these regions were found to be optimal for background rejection and signal sensitivity. One characteristic shared between each of the SRs is that at least two jets must be tagged as an emerging jet, as is expected in the theory.

Their respective CRs share all of the same cuts found in the tables but will have exactly 1 tagged emerging jet. These requirement ensure that the events present in the CR have similar kinematics, while restricting to only one tagged jet makes the CR much more likely to contain background than signal events and makes the CR and SR independent from one another.

4.5.1.1 Scale Factors

For a SR and CR which differ only by the number of tagged jets, the scale factors representing the difference in the number of events that enter each region will be a function of a mistag rate ϵ , or the rate at which a background jet is tagged as an emerging jet.

Take an event which passes the shared cuts between the SR and CR criteria and has at least one jet tagged. Let $Q(n, \alpha)$ be the probability that the event would have n -tagged jets (out of the four highest p_T jets) where at least jet α is tagged. Every jet that is tagged an emerging jet happens with a probability ϵ , and every jet that is not tagged happens with a probability $(1-\epsilon)$. Then,

- $Q(0, \alpha) = 0$ by construction, since jet α is tagged
- $Q(1, \alpha) = \epsilon_\alpha(1 - \epsilon_1)(1 - \epsilon_2)(1 - \epsilon_3)$ where jets 1, 2, and 3 are the other 3 non-tagged jets
- $Q(2, \alpha) = \frac{1}{2!} \sum_{i \neq \alpha} \epsilon_\alpha \epsilon_i \prod_{j \neq \alpha, i} (1 - \epsilon_j)$
- $Q(3, \alpha) = \frac{1}{3!} \sum_{i \neq j \neq k \neq \alpha} \epsilon_\alpha \epsilon_i \epsilon_j (1 - \epsilon_k)$
- $Q(4, \alpha) = \frac{1}{4!} \sum_{i \neq j \neq k \neq \alpha} \epsilon_\alpha \epsilon_i \epsilon_j \epsilon_k$.

The probability that such an event would fall into the CR is $Q(1, \alpha)$, and the probability that the event would fall into the SR is $Q(2, \alpha) + Q(3, \alpha) + Q(4, \alpha)$.

Given an event falling in the CR and $\epsilon_i \ll 1$ (which can be seen in the next section), the scale factor, calculated as

$$SF_{\text{CR event}} = \frac{Q(2, \alpha) + Q(3, \alpha) + Q(4, \alpha)}{Q(1, \alpha)} \sim \frac{1}{2} \sum_{i \neq \alpha} \epsilon_i, \quad (4.16)$$

is the number of similar events expected to be in the SR. Thus, the estimated number of background events in the SR is

$$N_{\text{SR}} = \sum_{\text{CR events}} \frac{1}{2} \sum_{i \neq \alpha} \epsilon_i. \quad (4.17)$$

4.5.1.2 Mistag Rates

Since the background estimation is fully data driven, the mistag rates must be calculated from data. It can not be ruled out that the CR does not contain any emerging jets, and therefore mistag rates cannot be confidently calculated in this region since they must be evaluated on *only* background jets. Therefore, a signal-free region (FR) is used. The FR is a subset of the data which contains at least one isolated, high p_T photon in the physics event. Since the dark sector in the emerging jets theory is not charged under the electromagnetic force, it is impossible for a photon to be produced within this signal. It is also highly unlikely for such a high p_T photon to be produced from a secondary interaction within an event already containing emerging jets.

The exact FR event criteria are as follows:

- Pass the lowest un-prescaled photon p_T trigger.⁶ This γ trigger is a different trigger used for the CR and SR since it looks for the indication of a photon present.
- Have at least one well-reconstructed photon (is contained in the ECAL minus the endcap-barrel transition and does not have an associated pixel seed) with $p_T > 200$ GeV and which passes an isolation requirement [71].
- Has at least one jet with $p_T > 100$ GeV fully contained within the tracker volume with at least one associated track. These jets must also be angularly separated by the photon with $\Delta R(\text{jet}, \gamma) > 0.4$.

The top one or two highest p_T jets (if a second passes the aforementioned criteria) are the background jets used to calculate the mistag rates. The mistag rates are defined as the fraction of background

⁶The lowest photon p_T trigger in 2016 was 165 GeV, and for 2017 and 2018 it was 200 GeV

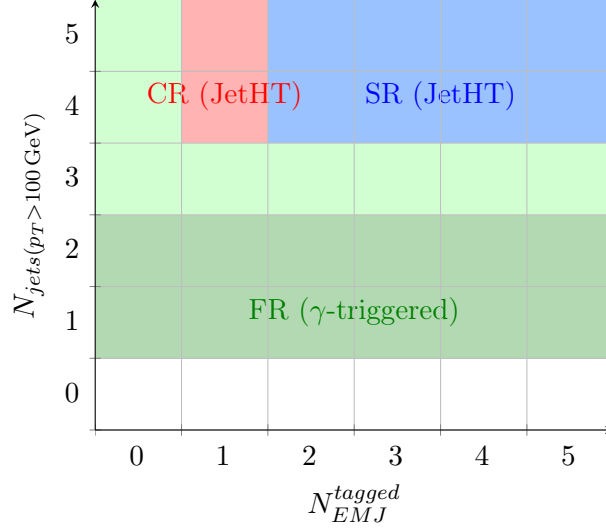


Figure 4.25: A comparison of the 3 different regions used in the background estimation: SR (blue), CR (red), and FR (green). The parenthesis indicates the trigger used in the region, and the darker green section of the FR indicates the jets used to mistag rate calculations.

jets that pass the cutset being tested. Figure 4.25 shows how the three different regions (SF, CR, and FR) compare to one another.

The mistag rates ϵ_i are calculated on a per-jet basis as they differ greatly dependent on certain jet properties. Figure 4.26 shows how the mistag rates vary in a simulated background sample as a function of jet p_T and underlying parton flavor. The shape of the distributions along p_T are a result of the composition of the training sample while developing the GNN. In the training sample, the p_T distributions of the background jets peak at around 300 GeV and falls slowly as the p_T increases, as is shown in Figure 4.20. Since the jets are weighted equally in training, the GNN learns to put more emphasis on correctly classifying the jets around 300 GeV since that is where the majority of jets live, and therefore the mistag rate is smaller in that region. Similarly, there are less jets at high p_T and so the mistag rate is larger. The b quark jets are missclassified more often than the other jet types because b hadrons are heavier than other hadrons, resulting in a wider jet, and b hadrons tend to travel a significant distance before decaying. Both of these features are shared with emerging jets and therefore make b quark and emerging jets harder to distinguish. Thus, this analysis chose to

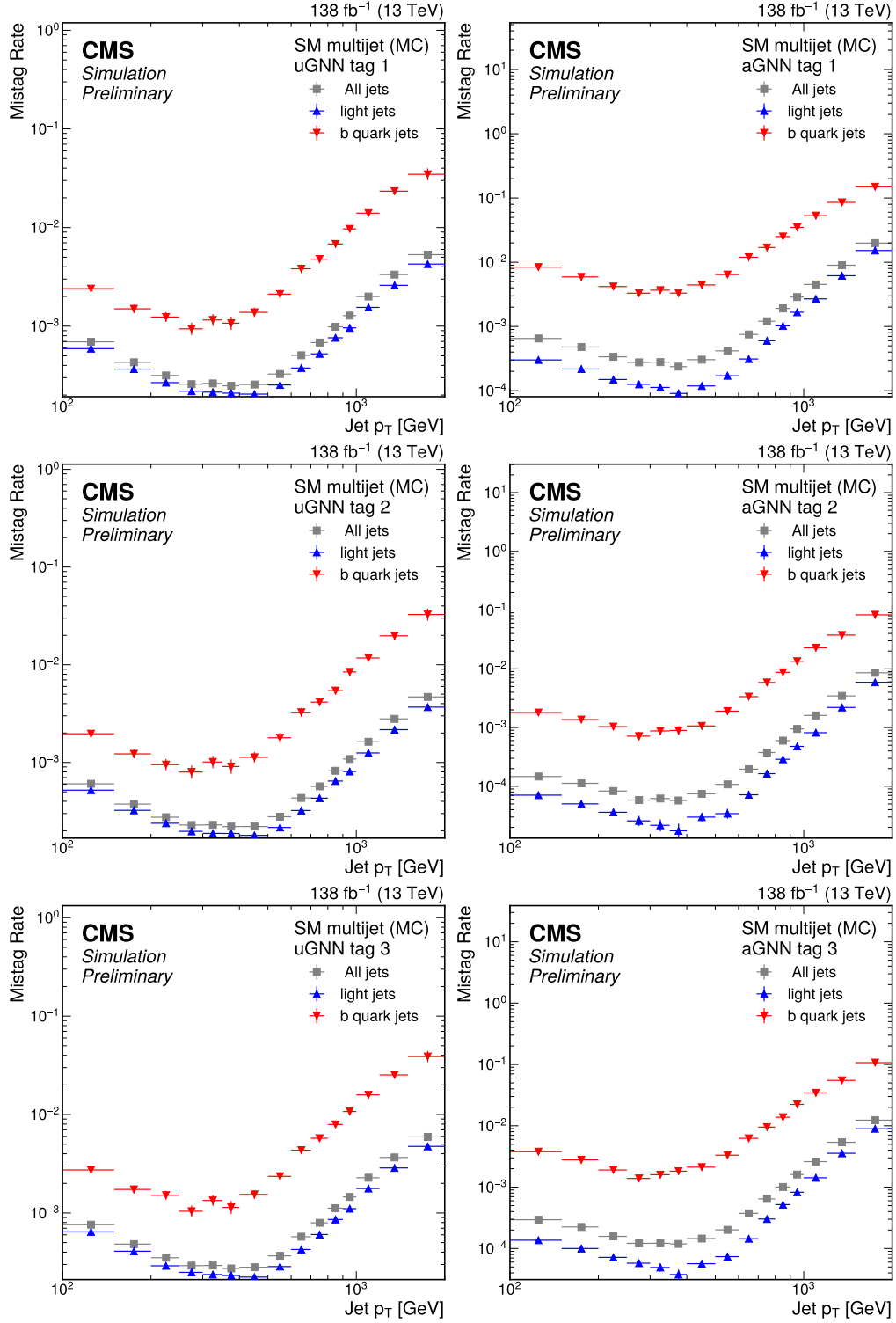


Figure 4.26: Mistag rate of simulated background jets along jet p_T and jet flavor for the GNN unflavored (left) and flavored (right) cutsets. The b quark jets are SM jets created through the hadronization of a b quark, and light jets indicates all other SM jets.

parametrize the mistag rate in p_T and jet flavor (f),

$$\epsilon_i = \epsilon(p_{T,i}, f_i). \quad (4.18)$$

In data, the p_T of the jet can be reconstructed but the flavor of the jet is unknown. Instead, the mistag rate of the different flavors of the jets can be estimated using a DeepJet b discriminator [72] and separating the FR into two orthogonal regions. An “extra” jet is defined as the highest p_T jet (> 50 GeV) in the FR *not* used to calculate the mistag rates. The b-jet discriminator will be used on this extra jet to split into “b-enhanced” and “b-suppressed” regions, FR_E and FR_S , dependent on if the jet passes the medium working point of the DeepJet score or not. Since b quark jets are often produced by gluon splitting, if the extra jet is a true b quark jet, then the selected jets will have an enhanced probability of containing a b quark jet.

The mistag rates of the different regions can then be written as

$$\begin{cases} \epsilon^E(p_T) &= B^E(p_T)\epsilon(p_T, b) + (1 - B^E(p_T))\epsilon(p_T, l), \\ \epsilon^S(p_T) &= B^S(p_T)\epsilon(p_T, b) + (1 - B^S(p_T))\epsilon(p_T, l), \end{cases} \quad (4.19)$$

where $\epsilon^X(p_T)$ is the mistag rate of FR_X , and $B^X(p_T)$ is the fraction of b-jets in FR_X .

$B^X(p_T)$ can be estimated by using the DeepJet b discrimination score to make template distributions obtained using truth b quark jets and light jets in the simulated signal free sample. $\epsilon^X(p_T)$ can be calculated directly in data as it no longer relies on the jet flavor. Therefore, everything is known in Eqn. 4.19 except for the individual b and light quark mistag rates, and the equations can be inverted to calculate for them:

$$\begin{cases} \epsilon(p_T, b) &= \frac{1-B^S}{B^E-B^S}\epsilon^E - \frac{1-B^E}{B^E-B^S}\epsilon^S \\ \epsilon(p_T, l) &= -\frac{B^S}{B^E-B^S}\epsilon^E + \frac{B^E}{B^E-B^S}\epsilon^S. \end{cases} \quad (4.20)$$

Figure 4.27 shows examples of how well this estimation of the b and light jet mistag rates does in data

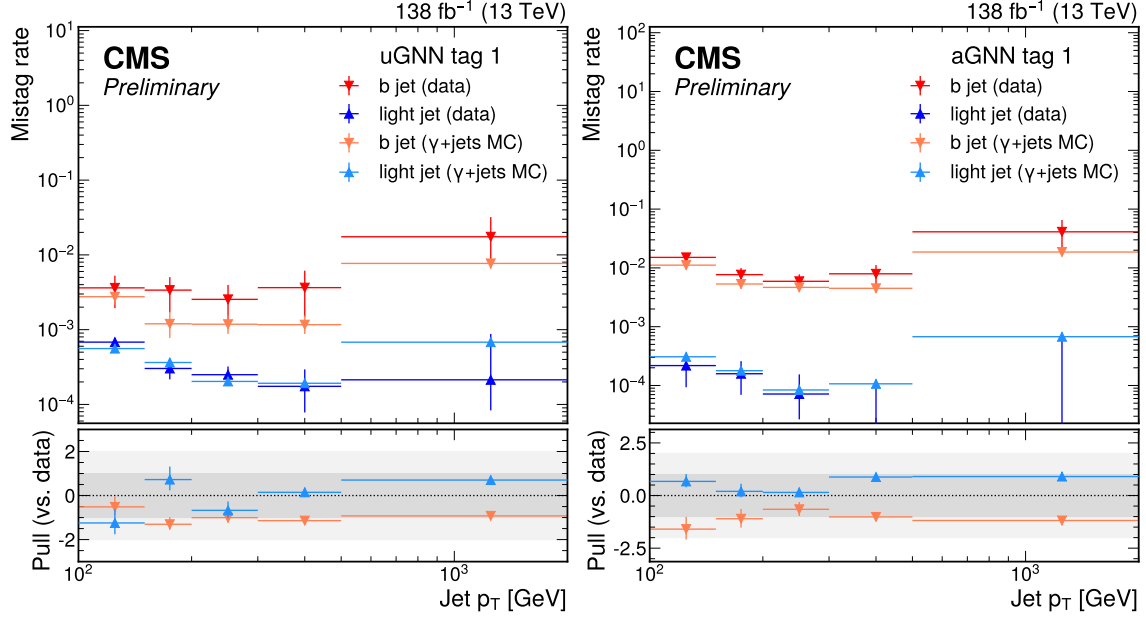


Figure 4.27: The mistag rates of the b and light jets in the data using the flavor estimation scheme versus the FR simulation truth information. An example cutset for the unflavored GNN model is shown on the left and aligned GNN model is shown on the right. The pull on the lower panel is the difference in mistag rates from the data and γ +jets MC sample, divided by the uncertainty in data.

compared with the true mistag rates in the FR simulation (γ +jets MC). In general, the estimated data and true FR values are all within reasonable error, indicating that the flavor estimating scheme is sound and the jets found in FR closely represent what is found in the data.

The mistag rates have now successfully been separated into individual mistag rates for b and other light quark jets. Referencing back to the SF , Equation 4.16, each jet in the CR event will have its own $\epsilon(p_T, f)$ depending on its properties. However, the flavor of the jets in the CR is still unknown. Following what was done in the previous emerging jets analysis [62], the mistag rate for each jet is assigned a weighted average of the mistag rates of the b and light quark jets based on the b quark jet fraction in the CR. This is given as

$$\epsilon_i = B^{\text{CR}}(p_T)\epsilon(p_{T_i}, b) + (1 - B^{\text{CR}}(p_T))\epsilon(p_{T_i}, l) \quad (4.21)$$

where B^{CR} is the b jet fraction of the CR calculated again using the DeepJet b discriminator. The

final estimation of the number of background events that is expected to appear in the SR is

$$N_{\text{SR}} = \sum_{\text{CR events}} \frac{1}{2} \sum_{\text{jet } i \neq \alpha} B^{\text{CR}}(p_{T,i}) \epsilon(p_{T,i}, b) + (1 - B^{\text{CR}}(p_T)) \epsilon(p_{T,i}, l). \quad (4.22)$$

4.5.2 Uncertainties

There are many estimations made in the evaluation of N_{SR} which can lead to uncertainties on the final value. In order to break down each uncertainty, a background estimation scheme will be defined as

$$Est_{\alpha}^A(\epsilon_{\beta}^B(\lambda)) \quad (4.23)$$

where A is the set of events used to estimate the SR events, B is the set of events used to evaluate the mistag rate ϵ , α is the method used to assign the jet flavors in region A , β is the method used to assign the jet flavors in region B , and λ is the variable that ϵ is parametrized along. The final background estimation N_{SR} in Equation 4.22 has $A = \text{CR}$, $B = \text{FR}$, $\alpha = \text{avg}$ and $\beta = \text{inv}$ where *avg* is the flavor averaging shown in Equation 4.21 and *inv* is the mistag rate estimation using the inverse method shown in Equation 4.20 ($Est_{\text{avg}}^{\text{CR}}(\epsilon_{\text{inv}}^{\text{FR}}(p_T))$).

There are four main approximations made in calculating the final background estimation N_{SR} . Each one of these approximations leads to an uncertainty in the final calculation which is assumed to contribute independently to the total uncertainty. The labeling scheme defined in Eqn. 4.23 can be used to highlight the changes made to the background estimation scheme before and after each approximation. Starting with the final estimation $N_{\text{SR}} = Est_{\text{avg}}^{\text{CR}}(\epsilon_{\text{inv}}^{\text{FR}}(p_T))$, each change to the estimation scheme will introduce a new uncertainty, shown below:

- 1) **Jet flavor uncertainty**, $Est_{\text{avg}}^{\text{CR}}(\epsilon_{\text{inv}}^{\text{FR}}(p_T)) \leftrightarrow Est_{\text{true}}^{\text{CR}}(\epsilon_{\text{true}}^{\text{FR}}(p_T))$: The flavor averaging and inversion methods are both used to estimate the mistag rates of b and light jets separately. If no estimation is made, then the true mistag rate of b and light jets are used.
- 2) **Phase space uncertainty**, $Est_{\text{true}}^{\text{CR}}(\epsilon_{\text{true}}^{\text{FR}}(p_T)) \leftrightarrow Est_{\text{true}}^{\text{CR}}(\epsilon_{\text{true}}^{\text{CR}}(p_T))$: Although the FR is assumed to contain background events that are representative of what is in the SR, this may

not be entirely true. Removing this assumption means the CR events are used, which are the same events in the SR but differ by the number of jets tagged.

- 3) **Bin width uncertainty**, $Est_{true}^{CR}(\epsilon_{true}^{CR}(\mathbf{p}_T)) \leftrightarrow Est_{true}^{CR}(\epsilon_{true}^{CR}(\mathbf{p}_T^{fine}))$: The widths of the p_T bins may not be fine enough to fully encompass the rapid change of ϵ along p_T . If statistics were unlimited, the mistag rate would be binned into infinite bins. Since this is not possible, the next best thing is to do a very fine binning.
- 4) **Parameter uncertainty**, $Est_{true}^{CR}(\epsilon_{true}^{CR}(\mathbf{p}_T^{fine})) \leftrightarrow Est_{true}^{CR}(\epsilon_{true}^{CR}(\mathbf{n}_{trk}))$: The choice to use p_T was made due to the large changes of ϵ along this variable. This variable may not be the best choice, however, and so the estimation can be tested against another variable with large variation in ϵ , namely the jet multiplicity (or number of tracks) n_{trk} .

Since most of these approximations cannot be evaluated in data as the true jet properties are unknown and the SR can not be presumed to only contain background, the uncertainties due to each of the small changes listed above are evaluated in simulation. The uncertainty is calculated as

$$U(Est_1, Est_2) = \frac{2|Est_1 - Est_2|}{Est_1 + Est_2} \quad (4.24)$$

where each Est_1 and Est_2 are slightly altered estimations represented by $Est_1 \leftrightarrow Est_2$ in one of the above uncertainties. Table 4.8 compares each uncertainty between the cut-based and GNN cutsets for both the unflavored and flavored scenarios.

In general, the cut-based unflavored method has a higher uncertainty for the mistag parameter choice than the GNN since the variables used in the cut-based method all strongly rely on p_T as opposed to n_{trk} . The GNN, however, has a larger binning error due to the effects that the p_T distribution has on the training of the classifier where the regions with less jets (high p_T) are not deemed as important and therefore the mistag rate increases quickly but is harder to capture in small bins, as was shown in Figure 4.26. Assuming these uncertainties are independent, they are summed in quadrature to produce the total uncertainty on N_{SR} for each cutset.

Tagger	Source	Unflavored [%]		Flavored [%]	
		Avg.	Max	Avg.	Max
Cut-based	ϵ param.	22.5	33.1	7.9	14.9
	ϵ bin.	0.2	0.5	2.3	5.5
	Phase space	8.1	15.7	15.0	33.6
	Jet flavor	24.0	70.4	10.2	14.2
GNN	ϵ param.	4.0	6.2	7.9	17.9
	ϵ bin.	10.9	15.6	41.4	59.2
	Phase space	11.0	22.9	12.1	18.1
	Jet flavor	23.7	34.3	14.6	24.8

Table 4.8: Background uncertainties for both the cut-based and GNN event selection methods. The values shown are the average and maximum uncertainties for the set of cutsets using a specific tagger and coupling scenario.

There are also statistical uncertainties which will need to be considered. One of these uncertainties is a Poisson uncertainty of the finite number of events in the CR. The other is an uncertainty on the mistag rate, which is calculated assuming the rates are perfectly correlated across all events while being uncorrelated across the different p_T bins. This uncertainty is calculated by varying the input jets in simulation and propagating these changes through to the SF computation to determine the differences these variations make.

4.5.3 Simulation Closure Tests

In order to ensure that the estimation methods above are sound, the estimation versus the actual number of background in the SR can be tested in simulation (labeled “MC” for Monte Carlo). The QCD MC sample simulates the physics processes expected in the JetHT data and can be further split into a SR and CR. The GJets MC sample simulated the physics processes expected in the γ +jets data and can be used for the FR. Using these samples to compare the estimated value directly with the QCD MC SR counts, no significant deviations are seen, indicating that the estimation method is working as expected.

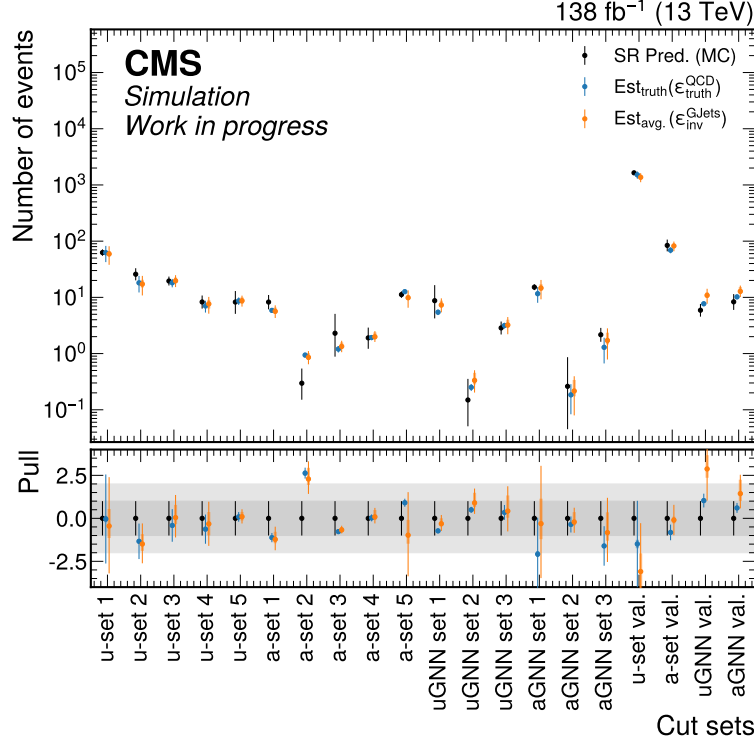


Figure 4.28: Background estimation closure test in simulation, where the full estimation (orange) closes within two σ of the actual SR counts (black). The pull on the lower panel is the difference between an estimation scheme and the SR prediction, divided by the uncertainty in the SR prediction.

Figure 4.28 shows how the full estimations (in orange) all fall within 2σ of the actual SR counts (in black, labeled SR Pred), representing a “closure” in the calculation. The uncertainty on SR Pred is statistical. The estimation in blue represents the estimation removing approximations on jet flavor and on jet composition similarities between the SR and FR. Large uncertainties in the full estimation, shown with cutset aGNN set 1 for example, are due to statistics as these cutsets have the tightest restrictions and therefore have the fewest number of events that pass all cuts.

4.5.4 Data Closure Tests

The estimation method can also be tested in data, but must be done so in a manner that greatly reduces the potential signal in the JetHT data as this estimation is meant to represent a background only physics scenario. To do this, four new cutsets were created which are designed for signal suppression or dilution. For the cut-based methods, each cut value was loosened to the

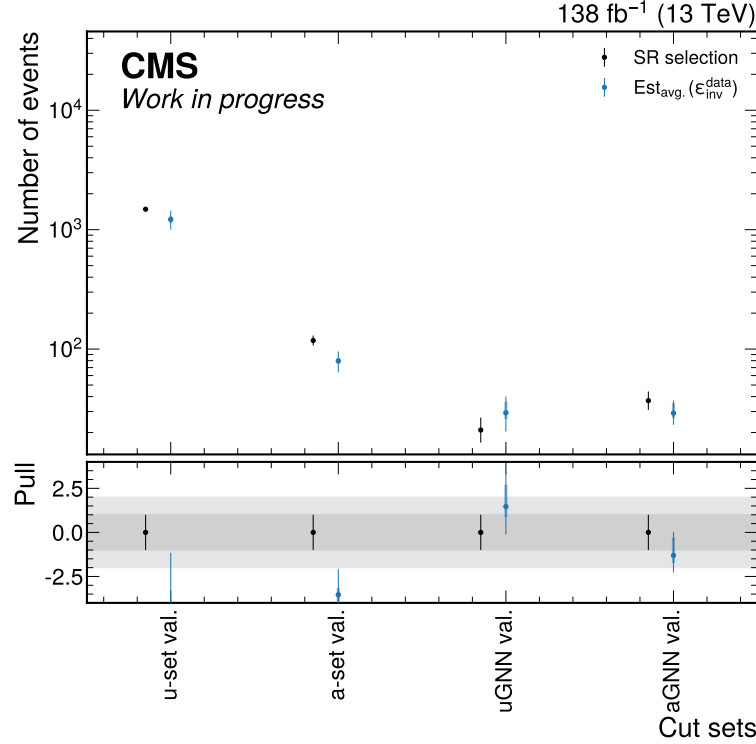


Figure 4.29: Background estimation closure test in data using signal suppressed or diluted cutsets, where the full estimation (blue) shows no large deviation from the actual SR counts (black). The pull on the lower panel is the difference between an estimation scheme and the SR prediction, divided by the uncertainty in the SR prediction.

minimum value of the collection of similar cutsets in order to ensure decent statistics, but the H_T was given a lower range (1000 – 1200 GeV) which suppresses the signal potentially present with large amount of background. This signal suppression relies on results from the previous emerging jets analysis [62] which excludes models with $m_{X_{dark}}$ up to 1200 GeV with 95% confidence. For the GNN methods, also loosening the cuts to the minimum values and choosing a “side bin” of the GNN score variable (0.998 – 0.9995 for the unflavored model and 0.99 – 0.995 for the flavored model) suppresses the signal present in the chosen region. To further dilute signal events, the H_T minimum cut is reduced to 800 GeV. Based on tests in simulated data, the expected amount of signal events that would pass these cuts is $< 5\%$.

Figure 4.29 shows the results of running the background estimation scheme on these new cutsets, labeled ‘u-set val.’ and ‘a-set val.’ for the cut-based unflavored and aligned methods and

‘uGNN val.’ and ‘aGNN val.’ for the GNN unflavored and aligned methods. No significant excess is seen in the number of events passing into the SR when comparing with the estimation in data using a signal suppressed or diluted region (black). This indicates that the calculation is doing as is expected and can now be used for signal extraction in the true SR cutsets applied to data.

4.6 Results

The final results for the background estimation on the full 138 fb^{-1} of data for all cutsets can be seen in Table 4.9. These value comparisons are displayed visually in Figure 4.30. No significant excess of events passing the SR criteria was observed when compared with the estimated background. These observed and estimated values are then used to compute the 95% confidence level (CL) upper limits for each emerging jet model parameter point with a simple cut-and-count method using the Higgs Combine Tool [73] CL_s criterion [74]. This tool will determine whether, given the background estimation and observation in SR and signal sensitivity for a specific emerging jet model, each hypothetical model can be rejected with $>95\%$ confidence.

The CL_s upper limit results⁷ are presented in terms of signal cross sections (σ), which is the rate at which the signal process occurs, and an example of these results can be seen in Figure 4.31. The median expected upper limit (dashed blue line) represents the lowest signal cross section which can be rejected with at least 95% confidence given the search methods’ signal sensitivity and a background-only hypothesis. The green and yellow shaded areas around the dashed blue line represent the spread of the expected median value within one and two standard deviations, respectively. The observed upper limit (solid black line) represents the lowest signal cross section which can be rejected with at least 95% confidence given the observed number of events in the SR. If the observed and expected upper limits match well with one another, then the observed values agree well with a background-only hypothesis. The red dashed line represents the theoretical cross section of the signal process. If the observed upper limit is below the theoretical cross section for a given

⁷The CL_s upper limit method is widely used in high energy physics analyses and can be explained in more detail at [75] and [76].

Cut set	Estimated background			Observed yield
u-set 1	56.2	$^{+9.0}_{-5.2}$	± 19.5	67
u-set 2	20.0	$^{+4.3}_{-2.5}$	± 7.0	21
u-set 3	22.9	$^{+7.3}_{-2.1}$	± 4.9	24
u-set 4	7.9	$^{+2.0}_{-1.6}$	± 2.2	10
u-set 5	11.3	$^{+2.7}_{-1.9}$	± 2.0	13
a-set 1	8.8	$^{+2.4}_{-1.0}$	± 2.0	16
a-set 2	1.67	$^{+0.49}_{-0.23}$	± 0.38	3
a-set 3	1.97	$^{+0.47}_{-0.22}$	± 0.37	2
a-set 4	2.30	$^{+0.81}_{-0.30}$	± 0.39	3
a-set 5	10.2	$^{+2.3}_{-1.1}$	± 3.4	16
uGNN set 1	15.6	$^{+5.4}_{-1.9}$	± 3.8	18
uGNN set 2	0.73	$^{+0.44}_{-0.16}$	± 0.27	0
uGNN set 3	7.6	$^{+3.5}_{-1.3}$	± 2.3	9
aGNN set 1	44.9	$^{+17.8}_{-7.6}$	± 15.9	59
aGNN set 2	0.30	$^{+0.23}_{-0.07}$	± 0.18	1
aGNN set 3	3.8	$^{+2.2}_{-0.7}$	± 2.0	5

Table 4.9: Final background estimation results for the cut-based method (top) and GNN method (bottom). The first, asymmetrical errors in the estimation are statistical, and the second errors are systematic.

signal model, then the hypothetical emerging jet model can be excluded by experimental observation at CL_s of $>95\%$. The example in Figure 4.31 shows an exclusion of unflavored emerging jet models with $m_{X_{dark}} = 1600$ GeV, $m_{\pi_{dark}} = 10$ GeV, and $c\tau_{\pi_{dark}} < \sim 400$ mm for the GNN selection method.

Since hundreds of different emerging jet models have been tested, it is useful to aggregate each model's CL_s test results to determine the region of models which are excluded. An example of the aggregation of multiple CL_s test results can be seen in Figure 4.32 for the GNN tagging method. The red and black “exclusion” lines represent where the expected and observed upper limit cross sections intersect with the theoretical signal cross section, as this intersection is the threshold on the exclusion region. The x and y axes scan through the different $m_{X_{dark}}$ and $c\tau_{\pi_{dark}}$, respectively,

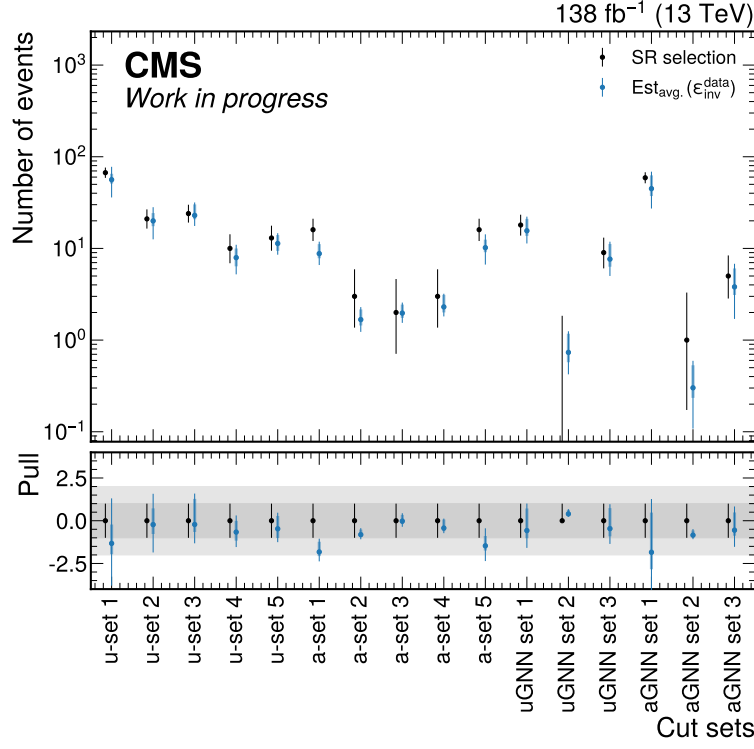


Figure 4.30: Visual depiction of the final background estimation results shown in Table 4.9. The pull on the lower panel is the difference in mistag rates from the data and γ +jets MC sample, divided by the uncertainty in data.

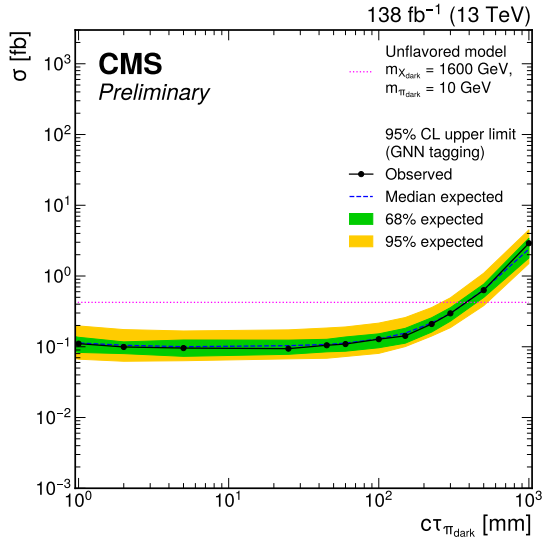


Figure 4.31: CL_s upper limit cross sections for the $m_{X_{\text{dark}}} = 1600$ GeV and $m_{\pi_{\text{dark}}} = 10$ GeV unflavored emerging jet models.

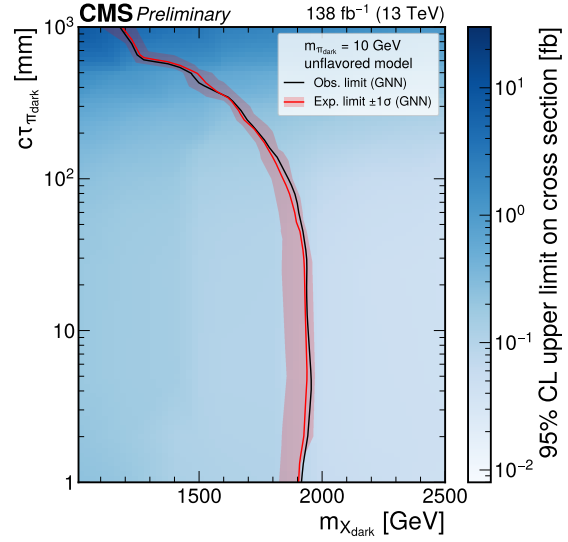


Figure 4.32: Exclusion results for all $m_{\pi_{\text{dark}}} = 10$ GeV unflavored emerging jet models.

so that the region of exclusion is to the left of the exclusion lines. The z-axis is the median value of the expected upper limit cross section. Figure 4.31 represents a vertical slice of Figure 4.32 at $m_{X_{dark}} = 1600$ GeV.

4.6.1 GNN Exclusion Results

Since the GNN has higher signal sensitivity and lower mistag rates than the cut-based method, it is capable of excluding a larger region of the emerging jet parameter space. Therefore, these exclusion results are deemed the final results from this analysis.

The exclusion range of the full emerging jet model parameter scan can be found in Figure 4.33 for the unflavored model, and Figure 4.34 for the flavor-aligned model. This analysis excludes unflavored models with $m_{X_{dark}}$ up to about 1950 GeV for $c\tau_{\pi_{dark}} < 100$ mm and flavored models with $m_{X_{dark}}$ up to about 1850 GeV for an average $c\tau_{\pi_{dark}}$. These results sets the most stringent exclusion limits to date, as well as providing the very first exclusions for a flavor-aligned emerging jet coupling scenario.

In the unflavored exclusion limits, the GNN method performs worse at higher $c\tau_{\pi_{dark}}$ due to the decrease in signal acceptance since the tracks produced from an emerging jet reaches the outer limits of the CMS tracker and therefore track information is lost. The exclusion limits set in the previous analysis [62] are seen on the left plot in Figure 4.33 in dashed lines. This analysis pushes back the original exclusion limits by about 500 GeV in $m_{X_{dark}}$. The flavored exclusion limits do not show any large degradation in performance at higher $c\tau_{\pi_{dark}}$ as the $c\tau_{\pi_{dark}}$ range studied for this model is smaller. The exclusion limit is therefore more stable along $c\tau_{\pi_{dark}}$ than what is seen in the unflavored model results. In all model variations, the observed exclusion limit falls within 1σ of error of the expected limit determined by the background estimation method.

The uncertainty on the flavor-aligned limits are very small due to the GNN aligned cutset two being used in this model region, as shown in Figure 4.24. This cutset has the tightest constraints and therefore almost no events are expected to pass into the SR (shown in Table 4.9). Thus, many iterations of this same analysis will have 0 background events pass into the SR, leading to a Poisson

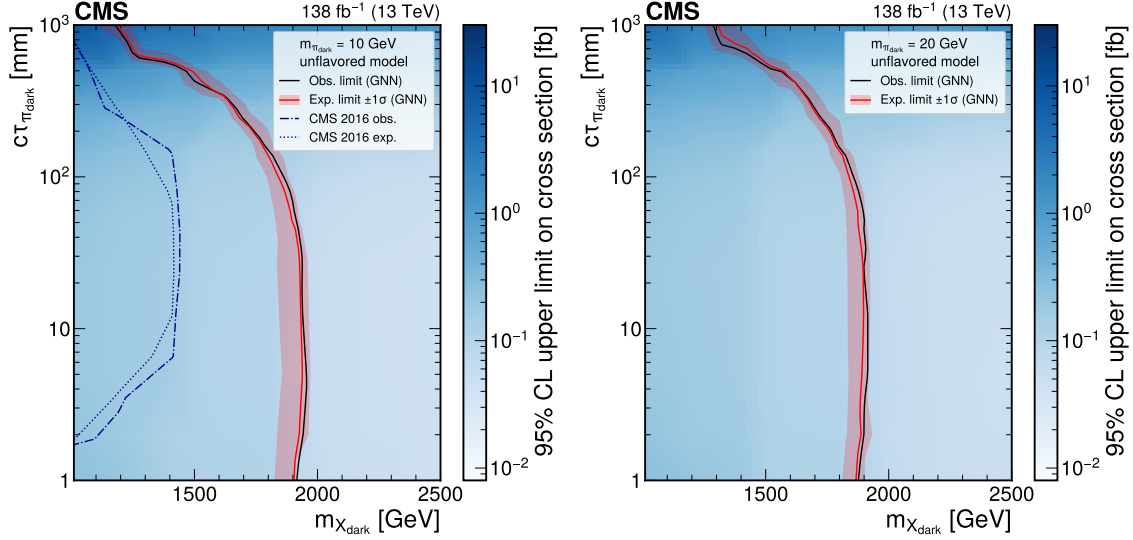


Figure 4.33: Final exclusion limits for the unflavored GNN method with $m_{\pi_{dark}} = 10$ GeV (left) and $m_{\pi_{dark}} = 20$ GeV (right) [55]. The $m_{\pi_{dark}} = 10$ GeV plot also includes the previous analysis [62] results (dashed lines) for comparison.

test statistic having discrete and concentrated peaks and the background and background+signal hypotheses distributions looking very similar. Consequently, only a small change in signal strength is needed to require a 95% CL, leading to small uncertainty on the estimated limits.

4.6.2 Comparisons to Cut-Based Exclusion Results

Overall, the GNN tagging method has a larger exclusion region than the cut-based tagging method, as is expected given the improvement in event selection performance the GNN achieves. The largest difference is seen in the unflavored emerging jet models at small $c\tau_{\pi_{dark}}$ where the cut-based method sees a large degradation in sensitivity and therefore the exclusion limit is much worse, as shown in Figure 4.35. The low $c\tau_{\pi_{dark}}$ region represents emerging jets that are more prompt in the detector which is where the majority of the background jets are as well. Furthermore, the cut-based method uses jet-level variables that rely strongly on the displacement of an emerging jet, whereas the GNN can capture track-level relationships within a jet. This leads to the GNN being better at distinguishing prompt emerging jets from background jets. The flavor-aligned results for the cut-based method can be found in Appendix A.3.

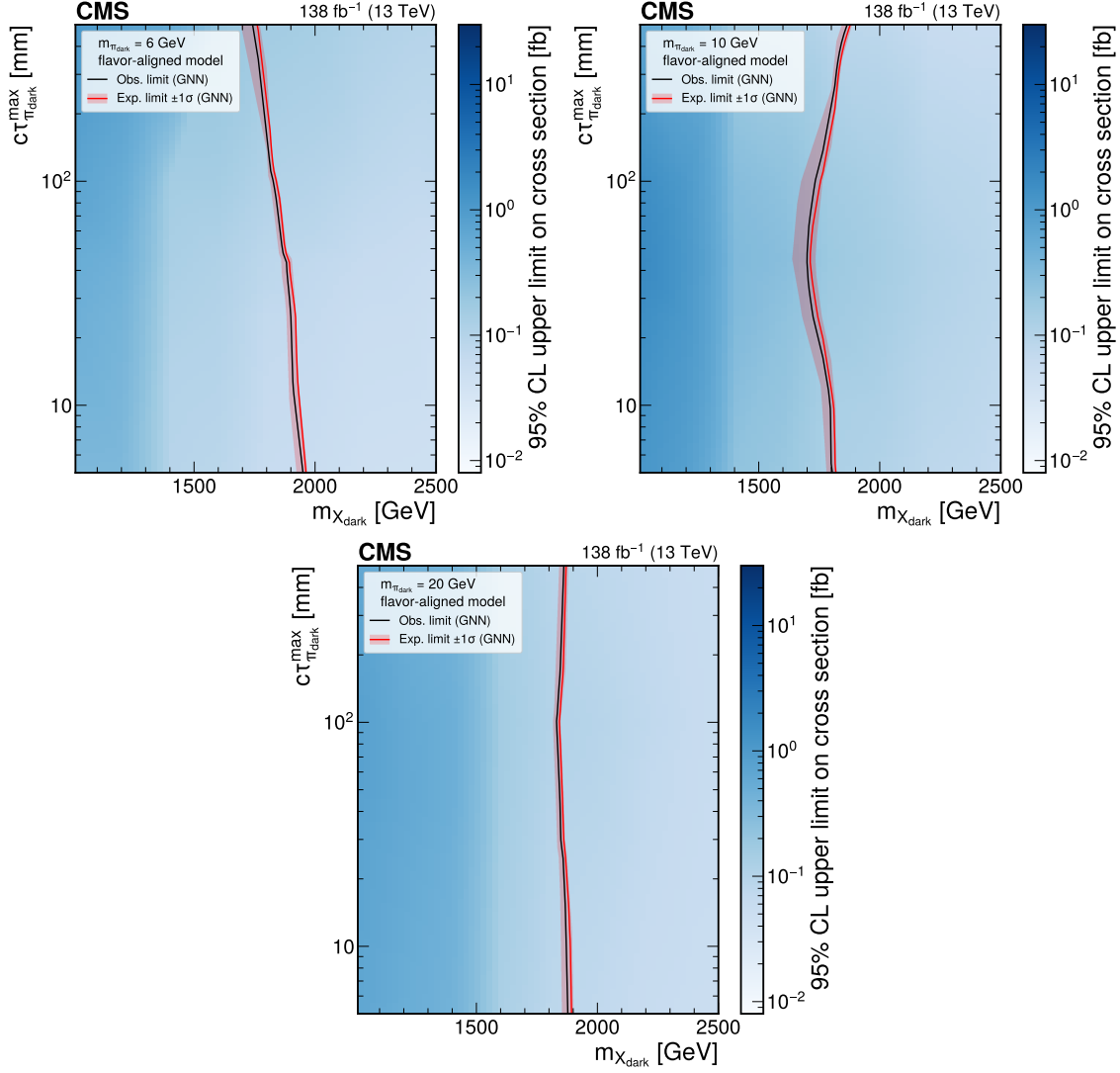


Figure 4.34: Final exclusion limits for the flavor-aligned GNN method with $m_{\pi_{\text{dark}}} = 6$ GeV (top left), $m_{\pi_{\text{dark}}} = 10$ GeV (top right), and $m_{\pi_{\text{dark}}} = 20$ GeV (bottom) [55].

One interesting question that can be answered in the context of this analysis is “how much does machine learning help speed up physics discovery?”. This analysis has optimized both a traditional physics event selection algorithm (the cut-based method) and a machine learning event selection algorithm (the GNN method) on the same physics signal using the same analysis methods. The outcome shows that the machine learning method is capable of excluding more theoretical dark matter models, and therefore physics discovery is achieved quicker, in a sense, as scientists can move on to studying other theories.

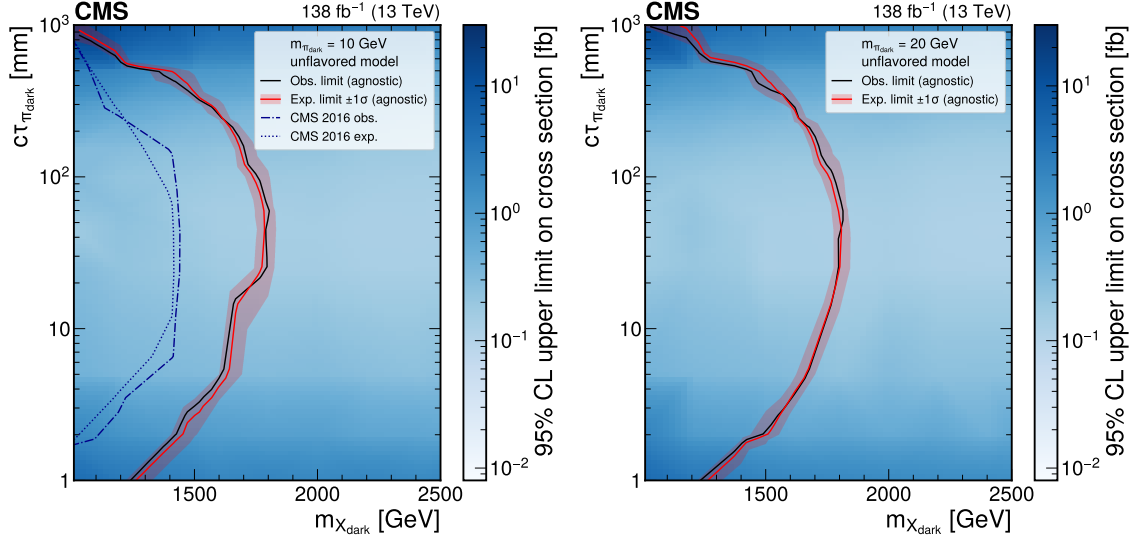


Figure 4.35: Final exclusion limits for the unflavored cut-based (or “agnostic”) method with $m_{\pi_{dark}} = 10$ GeV (left) and $m_{\pi_{dark}} = 20$ GeV (right) [55]. The $m_{\pi_{dark}} = 10$ GeV plot also includes the previous analysis [62] results (dashed lines) for comparison.

Assuming the event selection methods are optimal and the data produced by the CMS detector is stable, one can estimate how much more data it would have taken for the cut-based method to achieve approximately the same exclusion limit as the GNN method. Figures 4.36 and 4.37 show an example for the $m_{\pi_{dark}} = 10$ GeV unflavored and flavor-aligned scenarios of how the exclusions limits would change if the amount of data collected and studied was increased by a scale. These simple tests show that the unflavored cut-based method would require approximately 10 times more data to get a maximum exclusion limit similar to that of the GNN method (up to ~ 1950 GeV), while the flavored method would require approximately 9 times more data (up to ~ 1850 GeV).

Data collection, of course, also corresponds to time and money. In order to collect 9-10 times more data with the same experimental configurations as what was used in the 2018 collection era,⁸ it would take roughly 20 – 23 more years of data collection. When the energy consumption of the LHC and the CMS detector is also taken into consideration, it is clear that the use of machine learning techniques in this analysis has done a great deal in helping the advancement of physics discovery

⁸The instantaneous luminosity of the detector was continually increasing between 2016 – 2018 so more data was collected in 2018 (59.8 fb^{-1}) than the previous years.

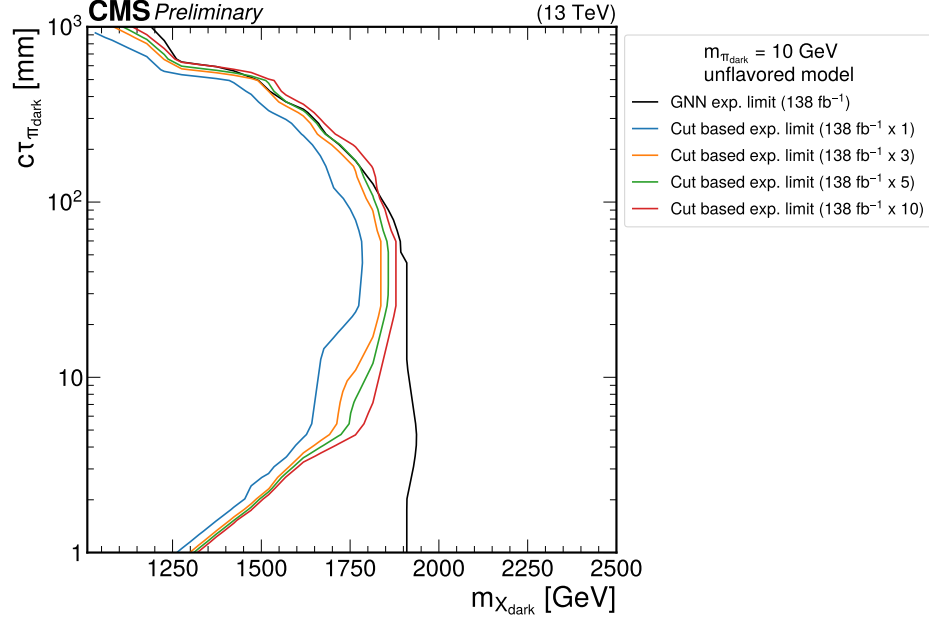


Figure 4.36: Example of how increasing the data collected by a scale can affect the cut-based unflavored $m_{\pi_{dark}} = 10$ GeV exclusion limits to be more similar to the GNN limits. The black and blue lines are the original limits for the GNN and cut-based methods (using 138 fb^{-1} of data) while the other colors represent a data scaling with the cut-based method.

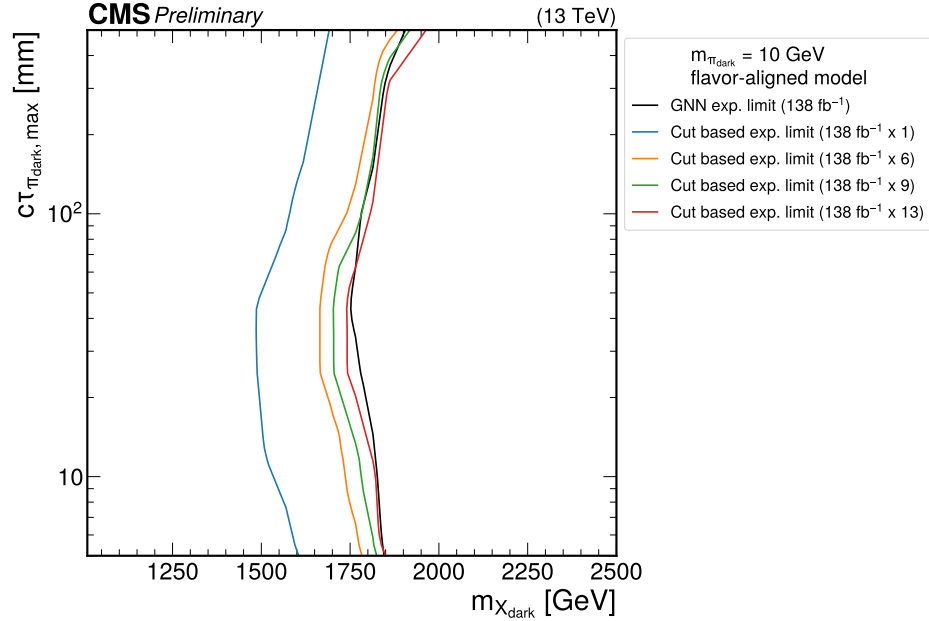


Figure 4.37: Example of how increasing the data collected by a scale can affect the cut-based flavor-aligned $m_{\pi_{dark}} = 10$ GeV exclusion limits to be more similar to the GNN limits. The black and blue lines are the original limits for the GNN and cut-based methods (using 138 fb^{-1} of data) while the other colors represent a data scaling with the cut-based method.

when compared to traditional physics algorithms.

Chapter 5

Triton Server Deployment at Fermilab Computing Facilities

5.1 Preface

In the dark matter search for emerging jets performed in Chapter 4, a GNN was developed to help analyze the data collected. This GNN was developed to replace a more traditional physics algorithm which makes simple selections on different physics object variables. The advantage of using machine learning (ML) techniques in physics analyses is the gain in algorithm performance, where an example of this gain can be found in Section 4.4.2.3. A drawback of switching physics analyses to use ML is the increase in computational complexity of the analysis code.

In the case of the emerging jets search which used computing centers at Fermi National Accelerator Laboratory (Fermilab) to evaluate analysis code, it became clear during evaluation that the current facility was not efficiently set up to run ML inference of the emerging jet GNN. By swapping only the emerging jet tagger from a traditional algorithm to the GNN, the run time of the analysis code went from $\mathcal{O}(\text{hours})$ to $\mathcal{O}(\text{days})$. As physics analyses using complex ML tools are still pretty new, no effort had yet been made to reconfigure the Fermilab computing centers to better accommodate for an increase in ML usage by its users. This motivated the work of this chapter, which has been presented at *Fast Machine Learning for Science Workshop* [77], published by *arXiv* [78], submitted to the *Computing and Software for Big Science* Journal, and is reprinted below.¹ The goal of this publication was to bring awareness to a problem and provide a solution

¹The author of this thesis is first author of the paper reprinted below, had a major part in all aspects of the work, and is responsible for >85% of the text.

for many current physics analysis computing facilities, and this work has since influenced a similar configuration at the Coffea-casa Analysis Facility [79] and will hopefully help more in the future.

5.2 Introduction

ML is a continually growing field, gaining traction across disciplines as new applications are found and tested. In high energy physics (HEP), for example, ML frequently outperforms traditional algorithms, leading to adoption for a wide variety of tasks, now encompassing the reconstruction and classification of physics objects and events recorded by particle detectors such as those at the Large Hadron Collider (LHC) [80, 81]. The most powerful ML techniques, such as GNNs, are more complex and correspondingly require more computing power and time [82, 83].

Computing power can be expensive and is not readily available to everyone. Therefore, many turn towards shared computing facilities that give users access to otherwise unaffordable computational resources [84]. In general, these facilities provide a variety of different platforms and processors to users, such as CPUs and GPUs, but tend to be optimized for conventional tasks requiring minimal computational power per user. Facilities like the LHC Physics Center [85] at Fermilab, which serves Large Hadron Collider (LHC) physicists from the CMS experiment, provide resources to hundreds of HEP researchers per year, but now struggle to meet computational demands efficiently because of growing machine learning enthusiasm.

This work aims to reconfigure shared computing facilities to allow for more efficient machine learning inference from their numerous users. In Section 5.4, we use the Fermilab shared facilities to show how an NVIDIA Triton Inference Server can be deployed and used to optimize machine learning inference when scaled to multiple users running parallel computing jobs. Section 5.5 then shows the computational gain and the effect of optimizing such a configuration at Fermilab. All results are specific to the Fermilab facility, but the tests and trends are reproducible by all similar multi-user facilities and are anticipated to show similar results.

5.3 Background

In this section, we define shared computing facilities and distinguish the different machine learning processors that are typically made available to users. We then briefly discuss the NVIDIA Triton Inference Server and how it interacts with the different processors.

5.3.1 Shared computing facilities

Computing facilities are widely used around the world to share computing resources among users [84]. As computational tasks become more complex and computationally expensive, shared facilities hold great value by allowing users to access powerful machines that are expensive to own individually. A few companies offer services that give the public access to their computing clusters for a fee, such as Microsoft Azure, Amazon Web Services, Google Cloud Platform, and IBM Cloud. Other companies, universities, research collaborations, and federal laboratories maintain private computing facilities to enable their researchers and employees to perform cutting-edge computations with a scope far outstripping the resources that can be dedicated to typical individuals.

Within HEP, researchers need the capability to process data in the terabyte (TB) to petabyte (PB) range, which may represent the sum of collected information for billions of particle physics collisions or years of continuous data collection. Subsets and variations of the data analysis processing may be repeated thousands of times each year. For the LHC experiments [30, 31], data processing is typically facilitated by large CPU-centric computing clusters like the LHC Physics Center (LPC) [85].

5.3.2 Common machine learning processors

Revolutionary advancements in the past decade have enabled machine learning to become a ubiquitous feature in modern research and commercial environments. As the field continues to develop, many of the resulting algorithms take increasingly larger proportions of the available computing power and runtime. GNNs, notable for their ability to process irregularly structured graph-like data, are an example of an ML model that can be rather complex and consequently poses

a computational burden when processing large sets of data. GNNs also represent a transformative paradigm shift for HEP, which naturally deals with events containing diverse and irregularly shaped inputs, often without an intrinsic ordering. Until their advent, HEP data needed to be heavily pre-processed for ML models having regular input shapes, with significant feature engineering involved, to attain high performance; GNNs have enabled similar or better performance with fewer input features, which is very desirable for HEP data. It is imperative for facilities like those employed in HEP to evolve and adapt to accommodate multiple users running complex machine learning algorithms, in order to avoid decreased computational efficiency and increased costs to researchers both in terms of money and time.

The two most common processing hardware classes seen at shared computing facilities, the CPU and GPU, have different trade-offs for running machine learning algorithms. CPUs can be faster in data transfer and storage, with better branch prediction and shorter pipelines, all of which are suited to general-purpose workflows. However, they are limited in parallelism and therefore computational throughput. Commercial GPUs, being designed for highly parallel paradigms like single instruction multiple data (SIMD) workloads [86], are particularly well-suited to accelerating ML training and inference [82]. By trading the more complex branch-prediction hardware and low pipeline latencies of CPUs for more vectorized compute capability, these devices gain considerable advantage in total FLOPS and compute/watt. GPUs are more expensive than CPUs (an individual NVIDIA H100 costs around \$35,000, whereas a 32-core AMD EPYC 7543 is approximately \$2,350 in 2023), but have $\mathcal{O}(10)$ better performance per watt, which closes the cost gap. In combination with lower general-purpose utility and need for specialized programming paradigms and code, GPUs are less frequently employed in HEP computing centers. Multi-user computing facilities are obliged to allocate such expensive resources efficiently for the increasing fraction of researchers using ML techniques.

A concept frequently considered in HEP is the time-to-insight, which is the amount of time it takes for a new idea to be proposed, implemented, validated, and analyzed on TB to PB data quantities. Being able to provide a short, large burst of resources to an analysis has significant

benefits to users. However, while minimizing analysis latency is paramount, it must be balanced with achieving high computational efficiency in shared facilities. The NVIDIA Triton Inference Server [87] supports both of these goals when paired with GPUs to augment multi-user computing facilities.

5.3.3 NVIDIA Triton Inference Server

One way to minimize cost while providing high burst capability is to provide GPUs as centralized resources for offloading ML computations, while general-purpose calculations are distributed across CPU-only servers. The GPUs are then accessed on-demand, with usage requests satisfied on the order of seconds, rather than minutes or hours, as is typical when requesting dedicated GPUs at HEP computing clusters. This paradigm, known as Inference-as-a-Service (IaaS), can be accomplished using the NVIDIA Triton Inference Server [87], which is open-source software that allows users to send inference requests from any framework to any CPU- or GPU-based platform. With this tool, shared computing facility users can run all of their code on CPUs except for the ML inference, which will take place on a GPU. A Triton server can simultaneously handle ML inference requests from multiple users, for multiple models, using multiple ML frameworks such as PyTorch and TensorFlow [88, 89].

With the Triton server set up on a cluster of GPUs, multiple models can be accessed in a device-agnostic way. All server instances connect to an object store where ML models are uploaded, and any server can dynamically load any model that a client requests. Additionally, dynamic batching can concatenate inference requests with sub-optimal batch sizes, perform the inference with near-peak efficiency by filling the GPU registers, then split and return the results to separate clients [90]. An individual client is not constrained by how many models can fit into device memory locally, and so may address dozens of models in fast succession, taking advantage of a one-to-many client-to-server connection via one unified interface [91].

5.4 Fermilab Triton Server Implementation

In this section, we discuss examples of shared computing facilities at Fermilab and how the NVIDIA Triton Inference Server is deployed.

5.4.1 Computing facility statistics

Fermilab is a national laboratory in the United States which specializes in particle physics research. It is the host laboratory of the US CMS Collaboration, which studies the fundamental particles of the universe using the CMS detector located at CERN in Geneva, Switzerland [31]. As such, Fermilab has several computing clusters accessible to US CMS researchers for all their computing needs.

Two shared computing facilities at Fermilab used in this work are the LHC Physics Center (LPC) and the Elastic Analysis Facility (EAF). The LPC is reserved for US CMS-affiliated researchers and has 240 cores available for interactive use (via 60 virtual machines) and another 4500 cores for batch submission. Each LPC batch node has a 10 Gb/s ethernet connection. The LPC currently has hundreds of users and is predominantly used for data analysis. The EAF is also designed for physics analysis, but is accessible to any Fermilab affiliate, intended to provide industry-standard data science frameworks and toolkits for low-latency analyses. It is built on the OKD [92] framework (the community-supported distribution of Red Hat OpenShift [93]), which provides scalable, reliable, multi-tenant Kubernetes [94]. The EAF consists of 12 machines with 286 CPU cores and 1643 GiB of memory, along with 8 NVIDIA A100 80 GB GPUs. It can also submit large workloads to the LPC batch system.

5.4.2 Typical user workflow

Users at the LPC and EAF typically use these computing facilities for data analysis. Upon connecting to one of these facilities, the user will be assigned to a node with access to communal software and storage areas. The collaborator then processes things in two ways: either locally on the

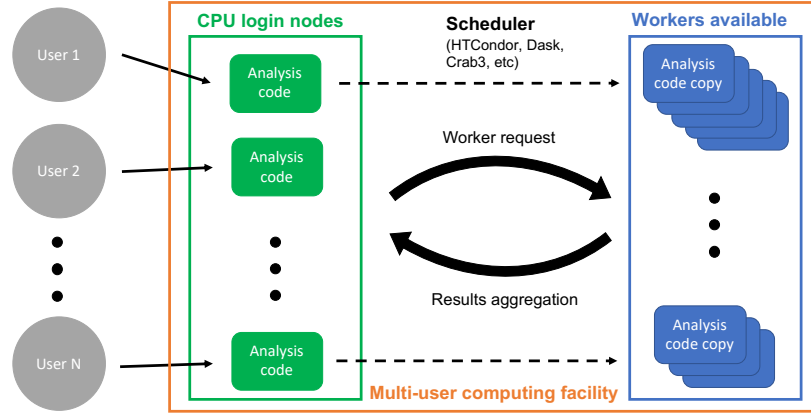


Figure 5.1: A schematic of a typical user workflow at a shared multi-user computing facility [78]. This example is based on the LPC facility.

login CPU node, or by distributing units of work to multiple CPUs/GPUs through a job scheduler. Figure 5.1 shows this typical user workflow as a schematic.

Physics analyses generally entail running the same code over billions of physics events. The analysis code is structured for immense parallel processing over the many data files storing all of these events. Therefore, users generally package a copy of their code to send to each CPU/GPU along with different subsets of the data to analyze so that total processing time is minimized. There are a number of tools that are used to scale the code out within large computing clusters, such as HTCondor [95], CRAB3 [96], and Dask [97].

Machine learning algorithms are becoming more commonly used by physics data analysts for a variety of tasks, such as event reconstruction and object classification [80]. When running analysis code on CPUs, machine learning inference generally takes up a significant amount of the full processing time, depending on the model. Utilizing GPUs to process the entire analysis would speed up the inference time, but is not efficient as significant portions of analysis code are not adapted for GPU usage. Optimizing this efficiency is imperative when the demand for GPUs exceeds what is available, as is the case in many computing facilities.

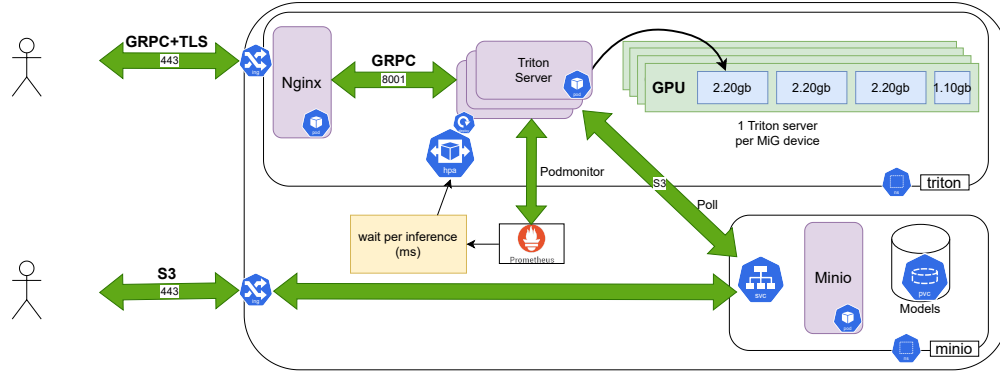


Figure 5.2: The Triton server implementation at the EAF showing the path of an ML inference request as it is created by the user and processed by the servers [78].

5.4.3 Triton server implementation

Instead of running uniquely on a CPU or GPU, the Triton server allows these two processors to work together. GPU resources are allocated to the server, which uniquely identifies each available model and dynamically loads needed models so that CPU clients can communicate inference requests. The researchers then send copies of their code to CPUs that compute everything locally except for the ML inference, which is processed by the GPUs on the Triton server. This implementation allows for fast GPU ML inference shared among multiple users.

A diagram of the current implementation is shown in Fig. 5.2. It includes two inference machines, each with 4 NVIDIA A100 80 GB GPUs and 2 AMD Epyc 7543 32-core CPUs. The Ampere architecture’s Multi-Instance GPU (MIG) capability is utilized to partition the GPU resources into multiple virtualized resources, and Triton Inference Server instances are deployed on MIG slices with 20 GB of RAM and 14 Streaming Multiprocessor (SM) cores. In Section 5.5.3, we also deploy MIG slices with 40 GB of RAM and 28 SM cores.

The A100 architecture has 1935 GB/s of bandwidth to the High Bandwidth Memory attached to the die, and 6912 CUDA cores, providing up to 19.5 TFLOPS of compute on FP32 data and 9.7 TFLOPS for double precision FP64. Each MIG slice has dedicated L2 caches, DRAM bandwidth, and memory controller allocations, helping ensure consistent performance regardless of the usage of neighboring MIG slices. Each Triton server periodically polls a MinIO [98] object store where all the

models are stored.

Inference requests originate from worker nodes on the LPC batch system. Users send requests via TLS-wrapped gRPC [99] to a haproxy [100] service built into OKD (not pictured), which are then immediately passed through to an nginx [101] service. The nginx service unwraps the gRPC request and sends it to a Triton Inference Server, using Kubernetes load-balancing. The Triton Inference service is configured to automatically scale up and down the number of server instances based on the average queue time for an inference request (called “auto-scaling”). Each inference machine is connected via 100 Gbps ethernet; however, the nginx and haproxy servers are only connected to the fabric of the LPC batch system at 100 Gbps. This connection could be a bottleneck when numerous LPC batch workers are making inference requests.

The Prometheus open-source monitoring system [102] built into OKD is used to collect inference metrics every 15 seconds from the Triton application via Kubernetes podmonitor objects, as well as machine characteristics such as core/memory utilization. The metrics are written to a Grafana Mimir [103] server for long-term storage, accessed via the REST API, and displayed via Grafana monitoring. The metrics collected by Prometheus are used to analyze the performance of the system in Section 5.5.

5.4.3.1 Parameter optimization

Multiple free parameters must be chosen when deploying the Triton server, which affects how quickly and efficiently models can be processed given the resources allocated. The parameters associated with the EAF Triton implementation mentioned above are all based on a standard GNN model used frequently for HEP applications, ParticleNet [104]. The ParticleNet GNN applies dynamic graph convolutional neural networks and edge convolution techniques to variable-dimensioned, unordered “point cloud” data. This model (exact model parameters given in App. B.2) will be used as the demonstration model in Section 5.5 and is a fair representation of the ML models being used in HEP today.

The size of the MIG slice (20 GB) for a server instance was chosen based on the RAM required

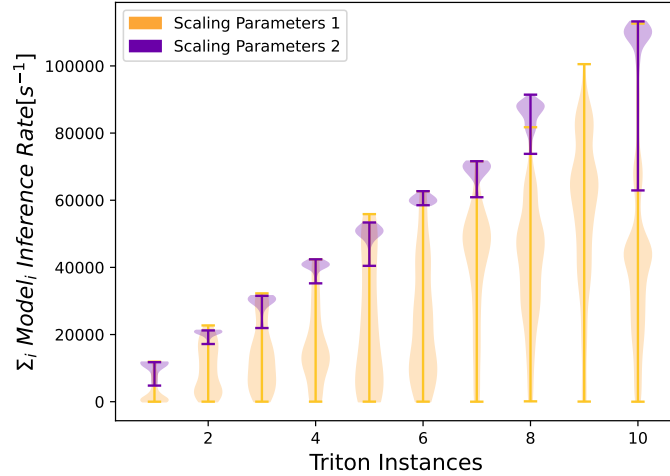


Figure 5.3: The violin plots show the net inference rate (inferences/s) as a function of the active Triton instances for two different sets of scaling parameters [78]. Each violin shows the minimum, maximum, and (through the width of the shaded band) the frequency of time samples (120 s long). For SP2, the server scaling skips from 8 to 10 instances as additional GPU resources became free.

to execute inference requests on the ParticleNet model. Section 5.5.3 will discuss how performance changes as this parameter varies.

The queue time per inference request is sampled every 15 seconds. If the average queue time exceeds 400 ms for four consecutive samples and it has been at least 3 minutes since the last scale up, an additional server is deployed. Conversely, if the average queue time is less than 400 ms for 40 consecutive samples and it has been at least 1 minute since the last scale down, a server is shut down. These settings are collectively referred to as Scaling Parameters 2 (SP2). The pre-optimized scaling parameters (Scaling Parameters 1, SP1) used a 100 ms threshold on the average over all models and different windows for scaling up and down. See App. B.1 for more scaling parameter information.

Figure 5.3 depicts the throughput of the ParticleNet model at the EAF for SP1 and SP2. The naive expectation is linear scaling of the maxima as a function of instances. With pre-optimized parameters SP1, the servers are under-utilized, with unused inference capacity the majority of time. Post-optimization gives the performance seen by SP2, demonstrating larger throughput and more consistent scaling with respect to Triton instances, which better maximizes the per-GPU throughput with the ParticleNet model. This indicates the importance of proper parameter selection.

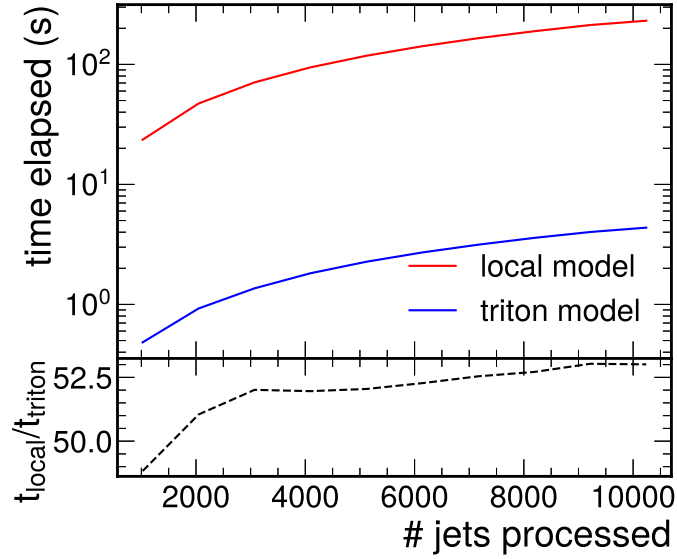


Figure 5.4: Comparison of the time it takes to process batches of data using a local CPU model vs. a Triton model instance on a GPU for ParticleNet [78].

5.5 Benchmarking Tests

In order to understand the benefits of setting up an NVIDIA Triton Inference Server at a shared computing facility, a few metrics are computed and analyzed. The timing and computational efficiency for the setup described in Section 5.4 are assessed. While all of the results shown in the subsections below are specific to the LPC and EAF Triton server setup at Fermilab, these tests can also be used as benchmarks for other Triton server deployments (code is publicly available at https://github.com/cgsavard/triton_multiuser_benchmarks).

5.5.1 Timing comparison

At the LPC, users typically run their ML models on the CPU nodes readily available to everyone. As discussed previously in Section 5.3.2, CPUs are not as efficient for machine learning inference as GPUs. The Triton server setup, which allows users to execute their inference on a GPU, therefore greatly reduces the overall computing time. For this test, we compared the processing time for inference on a local CPU instance of the ParticleNet model to a Triton instance of the model hosted on GPUs.

Figure 5.4 shows a significant speed-up of $\mathcal{O}(50)$ when processing 10,000 inputs (called “jets” for the ParticleNet model), motivating the use of the Triton server. The time elapsed starts when the full dataset is passed to the model and ends when all of the inference results are available, including data batching and pre-processing into the proper format for the selected model. Each data point on the plot represents the time elapsed (cumulative) after processing the indicated number of jets, with the batch size set to 1024. To minimize noise, which causes small timing fluctuations, the time elapsed is averaged over 10 trials for the local model and 100 trials for the Triton model. The fluctuations for the Triton model are larger than for the local model because of the network connection between the LPC CPUs and the EAF GPUs, which acts as an additional source of noise.

It is important to note that different machine learning models will achieve different speed-ups, or even slow-downs, when using a Triton server for GPU inference. In Fig. 5.5, we can see a speed-up of $\mathcal{O}(6)$ for a ResNet50 model [105, 106] when using the same Triton set up described in Section 5.4. ResNet50 has approximately 12 times more parameters and 7 times more FLOPS than ParticleNet

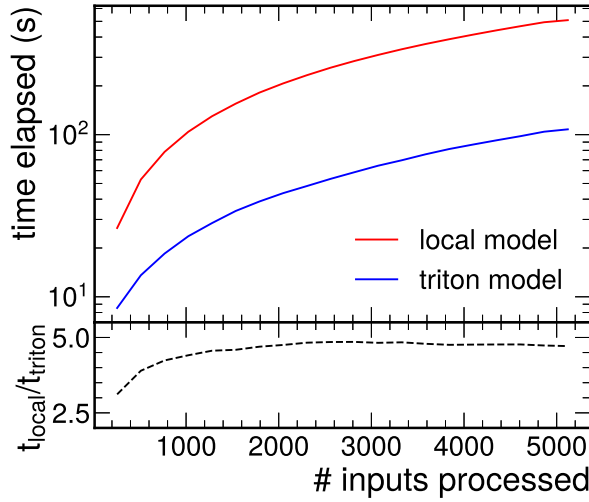


Figure 5.5: Comparison of the time it takes to process batches of data using a local CPU model vs. a Triton model instance on a GPU for ResNet50 [78]. 5000 inputs were processed in batches of 256. Results are averaged over 5 trials.

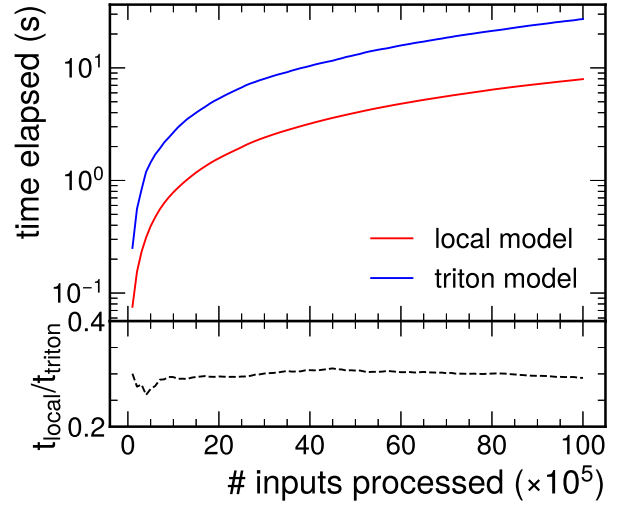


Figure 5.6: Comparison of the time it takes to process batches of data using a local CPU model vs. a Triton model instance on a GPU for the BDT [78]. 10M inputs were processed in batches of 10000. Results are averaged over 5 trials.

[104], as well as approximately 47 times larger inputs. Thus, the inputs of ResNet50 are a lot larger relative to the size of the neural network in comparison with ParticleNet. This causes the input processing step of the Triton inference to be a much larger fraction of the total inference time, about 10% compared to <1%. Therefore, the speed-up for GPU inference is smaller, as the input processing is less efficient than inference computation on the GPU.

Figure 5.6 shows an example of a model that takes more time for inference on the Triton server GPUs than on the local CPUs. This model is a small boosted decision tree (BDT) with 20 input features and 100 trees, trained using XGBoost [107]. BDTs are machine learning models that already run very efficiently on CPUs because the inference computation is dominated by simple logical operations. When using the Triton model, there is overhead that stems from data transfer and the packaging/unpackaging of the data. In this case, we see that the overhead from the Triton server masks any speed-up from accelerated GPU computing. Therefore, it is a bad choice to implement the BDT on the server, as it wastes the valuable GPU resources. Users of the server should always test their models to make sure that it is actually beneficial to use the Triton server.

5.5.2 Increasing workers

Now, we examine how the Triton server performs as a user runs inference in parallel on multiple workers to speed up the total inference time. For this test, we spawn varying numbers of workers that make parallel inference requests and see how this affects the inference time with the Triton server auto-scaling (as described in Section 5.4.3).

The Triton instances as a function of the workers can be seen in Fig. 5.7. The increase in instances is steady, determined by the server scale-out rate and queue time threshold, which then remains constant at 8 servers at around 28 workers. As the GPUs on the EAF are a shared resource, no additional MIG slices were available to expand further. Additional MIG slices were freed by other users around the time the benchmark reached 70 workers, and two additional servers were spawned. Fig. 5.7 shows how the resources can be reallocated for the Triton server efficiently as more GPUs become available.

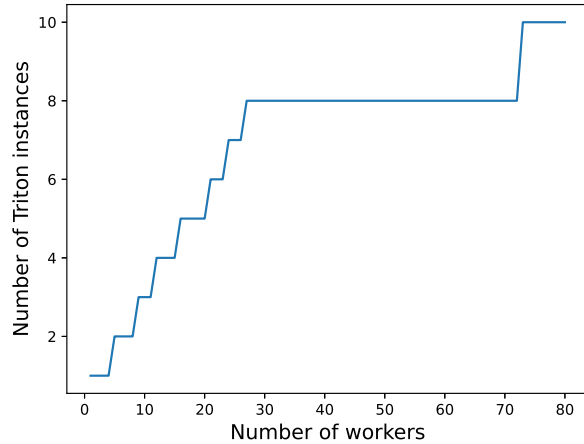


Figure 5.7: As the number of workers which make parallel requests to the Triton server increases, the number of Triton instances increases to parallelize the request processing [78].

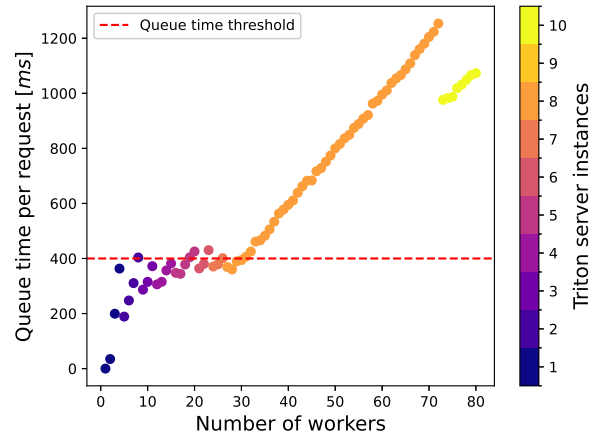


Figure 5.8: The relationship between the number of workers and the queue time per inference requests, showing the affects of the Triton instance auto-scaling [78].

Figure 5.8 shows how the auto-scaling affects the queue time of the requests as a function of the number of workers. A new instance is spawned when the queue time per inference request surpasses the thresholds described in Section 5.4.3.1. If the number of instances increases, there are more servers capable of processing requests and therefore the queue time decreases. When the maximum number of instances is reached, the queue will continually increase as more workers send requests and can only decrease when more resources become available to share the load. If the queue time becomes unmanageable because of GPU resource limitations, it may no longer be beneficial to spawn up more workers from the client side.

The throughput of the Triton server is defined as the rate at which inference requests are processed. As the number of Triton instances increases, more inference requests can be processed in parallel and therefore the throughput increases, as can be seen in Fig. 5.9. We may expect the throughput to remain constant so long as the number of servers stays the same, but we actually see a slight increase as more requests fill the queue. The throughput increases as a function of the number of workers because the queuing and processing pipeline becomes more efficient. As the number of instances increases, the processing pipeline stabilizes and the throughput grows more steadily with

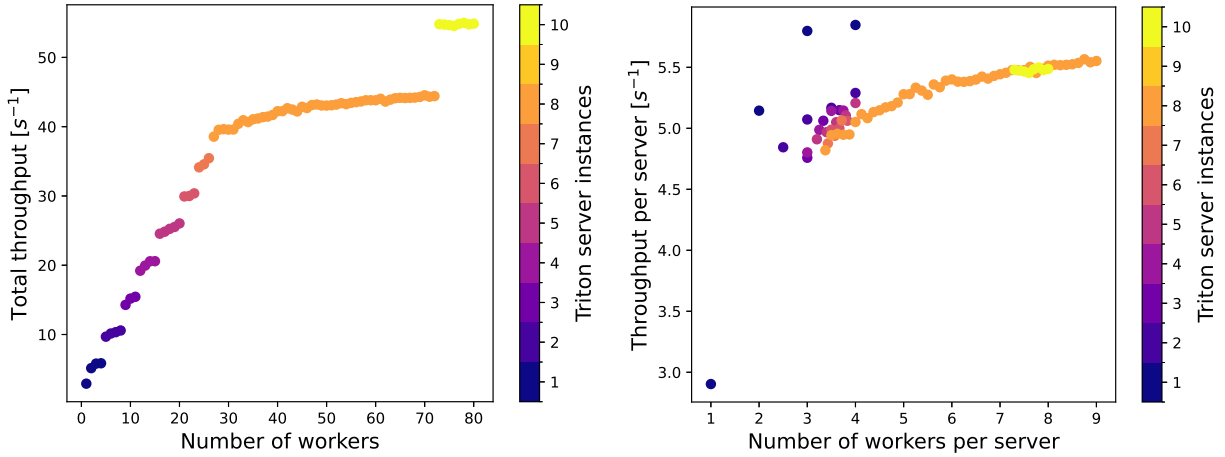


Figure 5.9: The throughput as a function of the number of workers in the full Triton server system (left) and a single server instance on average (right) [78].

increases in workers.

5.5.3 Multi-model scaling

In the previous subsection, we looked at the performance of an individual machine learning model using the Triton server for inference. In shared multi-user computing facilities, we expect to have multiple models running inference concurrently. When this occurs, the performance of a single model (“demo model”) can change due to the additional stress put on the Triton server.

The Triton server loads every model on every server instance running by default. This means that the 20 GB MIG slice hosting an instance is split among the different models and therefore the throughput for a single model decreases. In order to test performance when inference occurs for different models at the same time, we created “background models”: copies of the demo ParticleNet model, but labeled in such a way that the server would treat them distinctly. Figure 5.10 shows the relationship between the throughput of all models and throughput of a single model as a function of the number of background models for 20 and 40 GB slices.

The throughput of the individual models scale as $1/n$ when n models are perfectly sharing the GPU slice, as long as there is enough memory for each model to run in parallel. As the number of

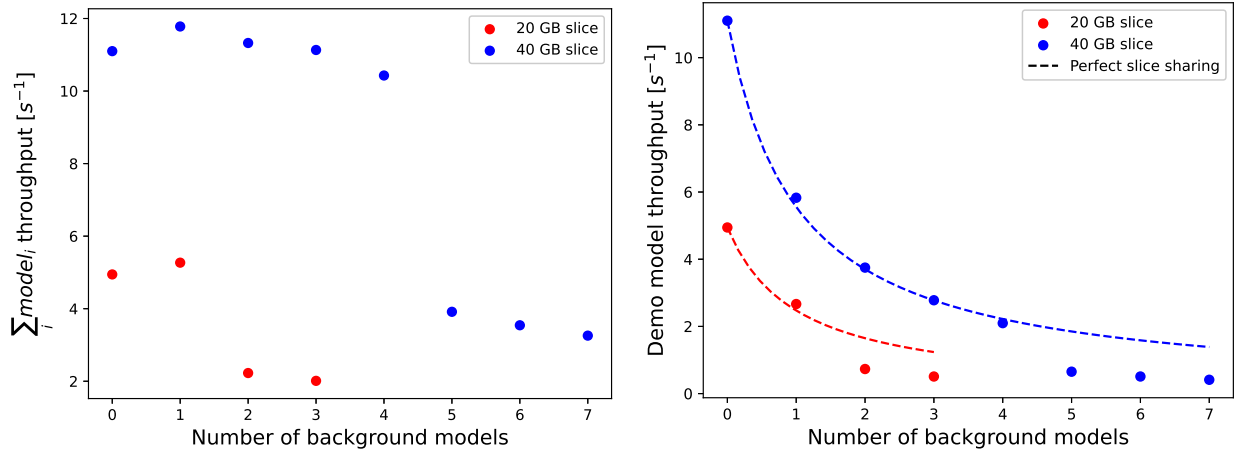


Figure 5.10: The throughput for all models aggregated (left) and the demo model (right) as a function of the number of additional background models running in parallel with the demo model on a single Triton server [78]. Perfect slice sharing leads to a $1/n$ decrease in throughput with the number of background models n . Each model has four workers sending inference requests in parallel.

background models increases, however, the models begin to compete for the instance resources and the throughput decreases faster than $1/n$. This degradation of performance can be due to models loading and unloading on the server or models remaining idle until memory for inference is made available (called “thrashing”). Figure 5.10 shows that this thrashing occurs after 2 models on a 20 GB slice and 5 models on a 40 GB, indicating that the demo model requires around 7 to 8 GB minimum in order to run inference efficiently.

Since all models are loaded onto each Triton instance by default, adding more instances does not fix the thrashing that occurs on a single instance. Instead, it is more efficient to make use of the multiple GPU slices available to process each model on a unique instance. Figure 5.11 shows the difference in throughput when all models are sharing each instance versus each instance holding only one model.

The constant throughput of the demo model for uniquely-assigned instances when other models are running in the background shows that processing performance of one model will not affect any other model at a multi-user facility as long as enough GPU resources are available. When GPU resources are constrained, the instances will have to begin splitting among the models carefully to

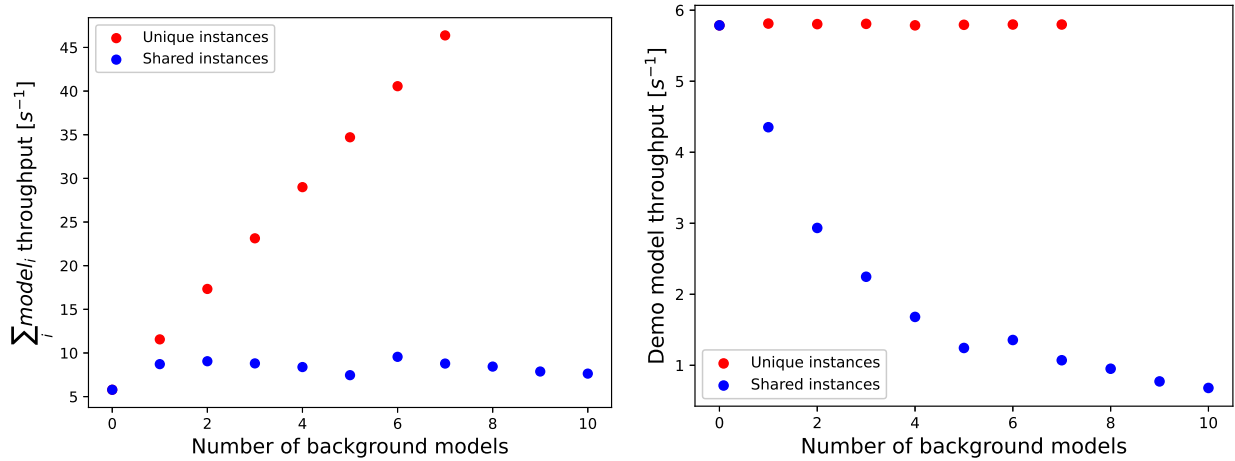


Figure 5.11: The throughput for all models aggregated (left) and the demo model (right) as a function of the number of background models running in parallel with the demo model with Triton instance auto-scaling [78]. The distributions show differences in performance when each model is hosted on a unique instance versus the models sharing each instance. Each model has four workers sending inference requests in parallel to it.

avoid the thrashing seen in Fig. 5.10. Such model orchestration is not a feature of the current default Triton deployment and therefore must be implemented manually. NVIDIA's Triton Management Service (TMS), part of their AI Enterprise product, advertises that it "allocates models to individual GPUs/CPU, and efficiently collocates models by frameworks". Unfortunately, TMS is not currently available to test at the EAF.

5.6 Limitations

All results mentioned in Section 5.5 are specific to the Triton server implementation at Fermilab described in Section 5.4. Other facilities may require a different configuration based on the resources available and design of the facility. Similarly, the Triton server parameters optimized in Section 5.4.3.1 are tuned on a model architecture frequently used in HEP research at the Fermilab facility. These parameters were only optimized on one variant of the model architecture and may need re-tuning as the collection of models used in the multi-user computing facility change or when the Triton server is implemented at a new facility with different network and compute resources.

The benefits of this work, mainly the exploitation of GPUs for quick bursts of resources resulting

in high inference throughput and fast turnaround, may be less obvious at a shared computing facility with fewer users or more GPUs. This work also does not explore implementations of inference-as-a-service on different coprocessor architectures such as Tensor Processing Units and FPGAs. These may provide complementary benefits through their performance [83, 108], and are an interesting area for additional study and comparison.

These results do not study the potential impact of insufficient GPU resources in detail, leading to over-subscription and untenable latency for the pool of users, nor potential fallbacks in the event that the GPU resources become unavailable for long periods of time.

5.7 Conclusion

In this work, we explore the usage and optimization of NVIDIA Triton Inference Servers at a shared multi-user facility aimed at maximizing throughput when scaling computational resources out to hundreds of users each parallelizing computing jobs. The Fermilab computing facilities have these large-scale computing requirements and are used to demonstrate the performance of model inference-as-a-service under such intense conditions.

The timing comparisons shown in Section 5.5.1 motivate using the Triton server to process inference requests on GPUs with a speed-up of ~ 50 compared to CPU-only processing for the ParticleNet model. Sections 5.5.2 and 5.5.3 show how machine learning inference performance on the Triton server changes as parallel requests and active background models increase. Both of these results show that high throughput can be maintained as more stress is placed on the Triton service when the server GPU resources are divided efficiently among the models to maintain a reasonable queue time and minimize competition for resources.

As machine learning becomes more established and ubiquitous in a variety of fields, it is more and more important to ensure that computing centers are capable of handling the increased load from machine learning inference. At shared computing facilities, resources must be allocated to users efficiently, and high throughput is important so that allocated resources can be freed up quickly for use by other users, and the time to insight can be minimized. Triton servers have been shown

to efficiently allocate GPU resources for high throughput computing, making this work a leading example of how other multi-user computing facilities can alter their systems to optimize efficiency for new machine learning demands.

Chapter 6

Level-1 Track Quality Development

6.1 Introduction

In the CMS experiment, data is often classified as “offline” or “online”. Offline data is data that was collected by the experiment, selected for study by the trigger (explained in Section 3.2.5), and saved to computing clusters to be analyzed by physicists at any point in time. An example of offline data is the data that was analysed for the emerging jet search in Chapter 4. Online data is the data that is collected by the detector, but not yet selected and sent off to be saved for later processing. This data is therefore being analyzed by the trigger system. Online data computation must satisfy additional algorithmic constraints in order to keep up with the rate of incoming data flow. Each individual algorithm must operate at the highest efficiency with regards to processing time and resources to ensure that there is room for each step in the processing chain.

This section focuses on the development of a new online data processing algorithm which computes a “track quality” variable for implementation in the next upgrade of the CMS Level-1 Trigger.¹ Machine learning tools are used to develop this algorithm which lead to great performance improvement over current track quality algorithms, and precautions are taken to ensure that the algorithm uses minimal resources and meets the timing constraint budgeted for the computation. This work has been presented at the *Machine Learning and the Physical Sciences NeurIPS Workshop* [110], the *Fast Machine Learning for Science Workshop* [111], and the *International Conference on High*

¹An overview of the Level-1 Trigger upgrade was presented by the author of this thesis at the *International Conference on Computing in High Energy and Nuclear Physics* and much of Sections 6.2 and 6.3 draw from those proceedings [109].

Energy Physics [112].

6.2 High Luminosity LHC

In 2029, the LHC will be upgraded to the High-Luminosity LHC (HL-LHC) [113] in order to increase the chance of potential physics discovery. The goal of this upgrade is to achieve a proton-proton collision instantaneous luminosity of $7.5 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$. This luminosity increase from the current LHC will allow for more data ($\sim 4000 \text{ fb}^{-1}$) which will in turn help in searches for new and rare physics. For example, the HL-LHC is expected to produce > 15 million Higgs bosons per year which will allow for more thorough analyses of Higgs properties and mechanics. As another example, Section 4.6.2 shows how an increase in the amount of data collected can affect the emerging jet search positively by pushing the exclusion limits further back.

A consequence of increasing the LHC luminosity is an increase in additional proton-proton interactions per LHC bunch crossing (called pileup), making each collision event more complicated than ever before; the HL-LHC is expected to produce an average of 200 pileup interactions. The algorithms and hardware used for careful selection of interesting physics events therefore need to be updated to accommodate the harsher conditions present in each event. Consequently, the CMS detector’s 2-step event selection chain—the Level-1 Trigger (L1T) followed by the High-Level Trigger (HLT)—is undergoing major upgrades in order to ensure the physics reach of the detector can be maintained, as well as expanded into new domains of physics previously made inaccessible by the trigger system.

6.3 Level-1 Trigger Upgrade

The CMS detector consists of many individual subdetectors with unique technologies for probing specific elementary particles. In brief, the silicon tracker is the innermost component which reconstructs charged particle paths passing through, the calorimeters measure the energies of electrons, photons, and hadrons, and the muon system provides more precise muon measurements. A more detailed description can be found in Section 3.2. Each of these subdetectors feed information

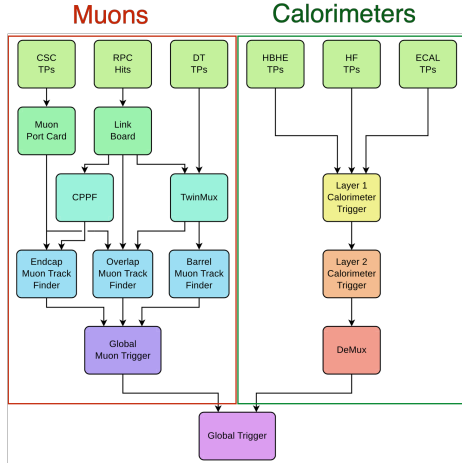


Figure 6.1: A schematic of the current Level-1 Trigger system [115].

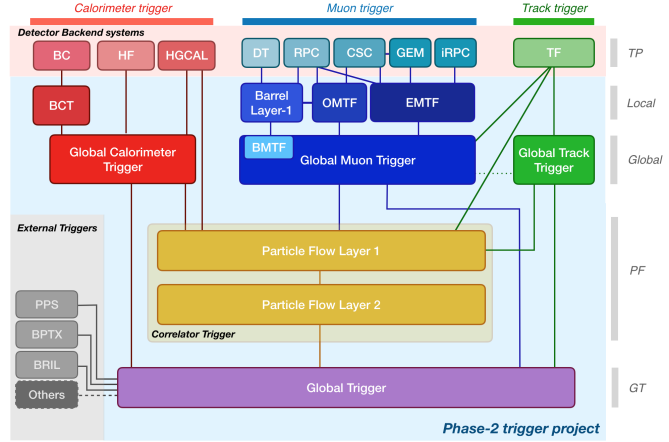


Figure 6.2: A schematic of the Phase-2 Level-1 Trigger system [114].

into the L1T system for informative event selection. The current L1T structure can be seen in Figure 6.1, and the proposed upgrade, also called the “Phase-2 L1T” [114], is shown in Figure 6.2. Both versions of the L1T collect information from the muon system and calorimeters, but Phase-2 will also gather information from the silicon tracker which is represented in green in Figure 6.2. Also in the L1T upgrade, the different components of the calorimeter, muon, and tracker trigger systems will feed into an intermediate correlator trigger before sending all information collected and processed to the global trigger for final selection.

6.3.1 Subsystem upgrades

There are 4 independent data processing paths in the upgraded L1T, as seen in Figure 6.2: the calorimeter trigger, muon trigger, global track trigger, and correlator trigger. As mentioned earlier, we can see that two of the subsystems, namely the correlator and track triggers, are completely new. The calorimeter and muon triggers have also been given major upgrades to provide more accurate information of the physics within the corresponding sub-detector.

The most important subsystem components are summarized below:

- **Calorimeter trigger** The calorimeter trigger receives information from the electromagnetic

and hadronic calorimeters. The main calorimeter trigger objects built are: electrons, photons, jets, hadronically decaying taus, and various energy sums. In comparison to the current calorimeter trigger, the updated calorimeters, including the new High Granularity Calorimeter in the endcap, will provide higher granularity, which will allow for high-resolution clusters and identification variables that increase the accuracy of the trigger objects.

- **Muon trigger** The muon trigger aims to identify and reconstruct muon tracks passing through the CMS detector. This trigger system will take in information from both the muon system and tracking system to create the muon trigger objects. In comparison to the current muon trigger, the upgrade will extend its reconstruction coverage from $|\eta| < 2.4$ to $|\eta| < 2.8$ and add in new primitive information from the CMS silicon tracker to those from the separate muon track finders in different regions of the muon system.
- **Global track trigger** The Global Track Trigger (GTT) is a new trigger system being introduced in the upgrade that takes in information from the track finder module which reconstructs charged particle tracker tracks. This trigger system will then use these tracks to build high-level track objects, such as jets, vertices, and jet H_T . These trigger objects can help significantly in tasks like pileup mitigation..
- **Correlator trigger** The correlator trigger is another new trigger system. Its function is to aggregate all information processed from the previous three upstream trigger systems (calorimeter, muon, and global track triggers) to achieve the best possible trigger performance on challenging physics topologies. There are two essential algorithms on the correlator trigger: Particle Flow identifies and reconstructs all particles given all sub-detector information, and Pileup Per Particle Identification (PUPPI) mitigates pileup effects. The correlator trigger ultimately creates more accurate trigger objects like hadronic taus, jets, missing transverse energy, and H_T .

A summary of where the trigger objects are created can be found in Figure 6.3.

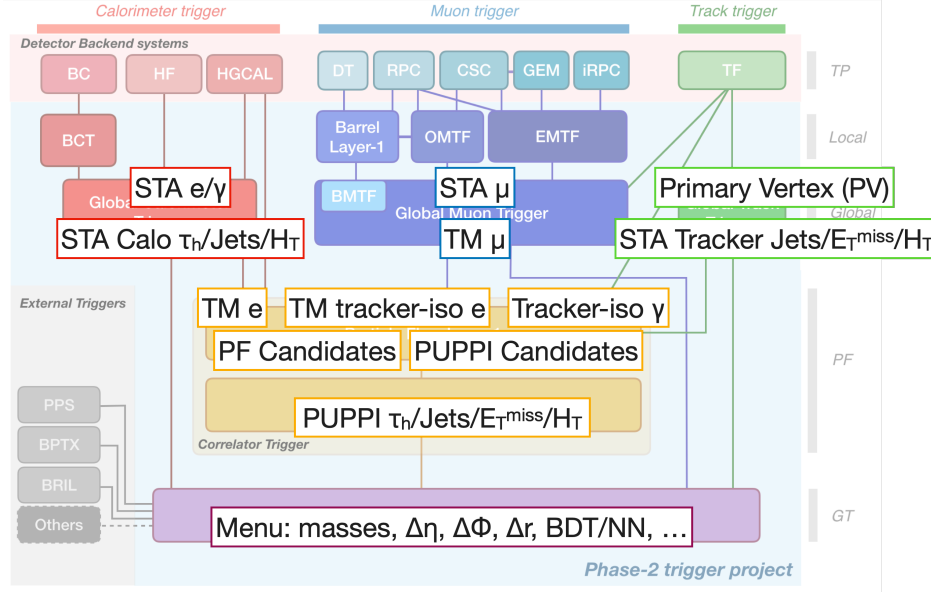


Figure 6.3: Schematic showing where on the trigger system certain trigger objects are reconstructed [115]. STA is stand-alone, meaning it only take information from one sub-system. TM is track-matched, meaning it takes in information from the tracker and another system. PF and PUPPI combine information from all sub-systems.

6.3.2 Architecture

During the HL-LHC era, the L1T will receive data at a rate of 40 MHz and will need to make selections within a $12.5 \mu\text{s}$ latency to output at a rate of 750 kHz. Within these tight constraints, the L1T will need to run sophisticated algorithms which reconstruct the physics objects shown in Figure 6.3. All firmware for the L1T system will be run on Field Programmable Gate Arrays (FPGAs) [116], which are configurable integrated circuits containing arrays of programmable logic blocks. This hardware was chosen due to its computing speed and re-programmable nature. All FPGAs used will be the same chip, currently expected to be a Xilinx Ultrascale+ VU13P. The FPGAs will be placed onto printed circuit boards (PCBs), and the functions performed by the different board families are different and therefore have unique firmware. Information is transferred between boards at 25 Gb/s through optical fibers.

Figure 6.4 shows a map of the proposed architecture for the upgraded CMS L1T system. The different highlighted sections indicate the separate trigger subsystems, such as the section labeled

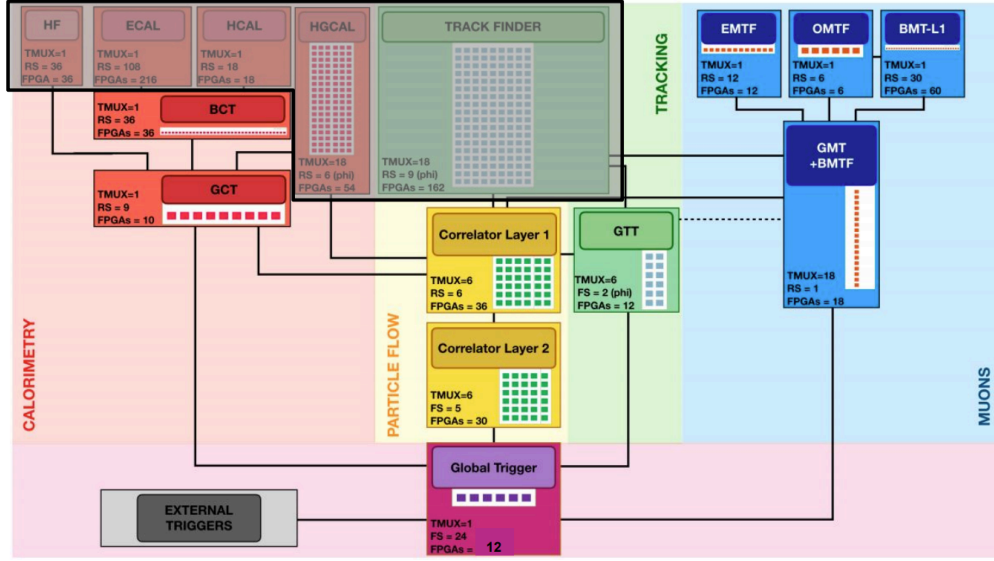


Figure 6.4: The proposed L1T architecture mapping indicating the full layout of the PCBs, FPGAs that they contain, and optical links between the boards. The gray shaded section outlined in black at the top left indicates the trigger boards associated with the detector backend system which creates trigger primitive objects used further down the trigger pipeline [109, 114].

“muons”. With each highlighted section, the dark boxes indicate PCBs, such as the box labeled “EMTF”. Within each PCB box, smaller boxes laid out like a grid indicate the FPGAs assigned to that PCB. The optical links are represented by a black line which shows the flow of information downward.

6.4 Level-1 Tracker Tracks

Trigger primitive objects, shown in the top row of Figure 6.2 labeled “TP”, are the basic physics objects generated in the backend electronic system of each sub-detector. These primitives are then sent into the trigger system to help reconstruct higher-level objects. With the addition of tracker information in the Phase-2 L1T, charged particle trajectories passing through the tracker can be reconstructed to create L1 tracker “tracks” [117]. These tracks hold information about the particles they represent through reconstructed track properties, and will be sent into the Global Muon Trigger, Global Track Trigger (GTT), and Correlator Trigger for later analysis.

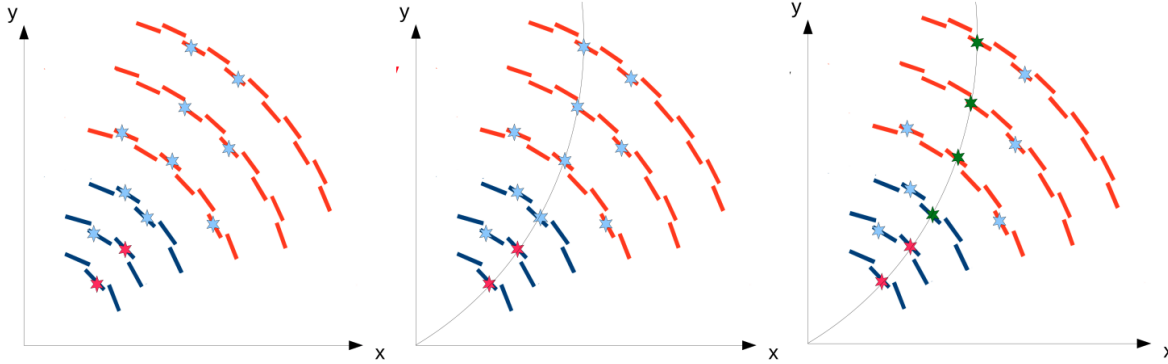


Figure 6.5: A visual of how the seeds are projected to form a track [117]. The blue and red curved line segments represent the different layers of the detectors, and the stars represent stubs. The seed (red stars in layers 1 and 2 of left plot) is projected out (middle plot) to match other stubs (dark green in right plot) and form a track.

6.4.1 Reconstruction

The reconstruction of tracks will occur at 40 MHz using information from the Phase-2 Outer Tracker [118, 119]. The outer tracker will consist of pairs of silicon modules with strips and macro-pixel sensors. In brief, the particles produced from the proton-proton collisions will travel outwards in a 3.8 T uniform solenoid magnetic field which causes the charged particles to curve in the detector $r\phi$ -plane via the magnetic force proportionally to the particle's transverse momentum. When the charged particles pass through the tracker modules, they will interact and leave energy deposits in their wake, also called “hits”. The hits will then be used to construct a track accordingly:

- (1) If hits are detected close enough together in both modules in the pair, then they are combined to form a “stub”. Due to the sensor spacing of the pairs between 1-4 mm, only stubs that come from particles with a transverse momentum $p_T > 2$ GeV. This is because charged particles with smaller p_T will bend more in the magnetic field and their resulting module hits will be too far apart to form a stub.
- (2) Stubs in neighboring outer tracker layers are paired up to form track “seeds”. These seeds are used to get an estimate of the particle's momentum and position along the detector's z -axis, and seeds are rejected if they are inconsistent with a track that has $p_T > 2$ GeV and

$$|z_0| < 15 \text{ cm.}$$

- (3) All seeds are projected to the other layers of the outer tracker to see if other stubs within those layers match the track. The projection happens both outward and inward from the seed to the center of the tracker. If multiple stubs within the same layer match the seed, then the one with the smallest difference with respect to the projected track position (also called the “residual”) is chosen. If there are less than four matched stubs (n_{stub}) to the seed, then the potential track is removed. Figure 6.5 shows an example of how the seeds project to include more stubs and create a track.
- (4) Particles can be reconstructed multiple times as multiple different seeds can project to match the same stubs. These redundant tracks are called “duplicates” and are merged together if they share too many of the same stubs.
- (5) A Kalman filter is applied to the remaining tracks to calculate the track parameters more precisely with a constraint that the tracks originate from the collision region (are prompt). This will output a collection of track objects from the event where each track has several features associated with it. The track features represent different properties of the track, and can be found in Table 6.1.

Due to some reconstruction limitations mentioned above as well as detector limitations, all tracks created are expected to be within the criteria mentioned in Table 6.2, but these cuts are applied to the collection to ensure this is the case.

6.4.2 Use-Cases

L1 tracker tracks are used by the GTT to build various track-level objects. A brief description of how these objects are created is listed below:

- Primary vertex (PV) - The track z_0 values are histogrammed along the z -axis with a p_T^2 weighting. The bin with the largest weight is assigned the PV z value. This algorithm is

Track feature	Description
ϕ	direction of track in ϕ
$\tan(\lambda)$	related to the direction of track in η
z_0	location of track along the z direction
$\frac{q}{r}$	charge \div radius of curvature, related to p_T by the magnetic field strength
hit pattern	indicates which layers the hits associated with the track were in
χ^2_{rz}/dof	the quality of fit of the stubs to the track projection in the rz -plane
$\chi^2_{r\phi}/\text{dof}$	the quality of fit of the stubs to the track projection in the $r\phi$ -plane
χ^2_{bend}	the consistency of the stub bend to the track bend in the rz -plane

Table 6.1: List of track features that are output from the Kalman filter. “dof” is degrees of freedom. The definition of χ^2_{bend} can be found at [114].

Track feature	Cut applied
p_T	$> 2 \text{ GeV}$
$ \eta $	< 2.4
n_{stub}	≥ 4
$ z_0 $	$\leq 15 \text{ cm}$

Table 6.2: Standard L1 track cuts applied to all tracks due to limitations in the detector geometry and reconstruction.

called *FastHisto* [114] and is the currently accepted algorithm, but other PV algorithms are under development.

- **Jets** - The tracks are binned in the $\eta\phi$ -plane, weighted by their p_T . The bin values are then aggregated with neighboring bins twice (called “two-layer clustering”), consecutively, and the final high p_T bins are used to define the center of the jet, the tracks associated with each jet, and the jet p_T .
- **H_T** - The jet p_T calculated within the event are scalar summed to create the event H_T . This is the same definition as what is described in Section 3.3.3.2.
- **E_T^{miss}** - All track \vec{p}_T in the event is vector summed to determine the excess transverse momenta in the event. This is the same definition as what is described in Section 3.3.3.3.

These objects will then be passed on to the muon, correlator, and global triggers to aid in the reconstruction of their trigger-level objects. The tracks themselves are also sent to the muon and correlator triggers to provide more information for their physics object reconstruction.

6.4.3 Fake Tracks

If the tracker tracks are poorly reconstructed, then the physics objects will not be a proper representation of the physical process occurring in the event and event selection is not well informed. There are two ways in which a track can be poorly reconstructed: the track properties do not match the true particle properties very well, or the track does not represent a true particle at all. The latter is called a “fake” track, as opposed to a “real” track which is a track that represents an actual particle interacting with the detector. Figure 6.6 shows a simple example of how stubs can be combined to form fake vs. real tracks. Here, combinations of stubs from two different particles are creating a new track that is fake.

Fake tracks stem from errors in the reconstruction process and mask the real physics occurring within an event. These tracks are especially problematic for any analysis that uses the physics objects which rely on the combination of reconstructed tracks to be built, like those mentioned in

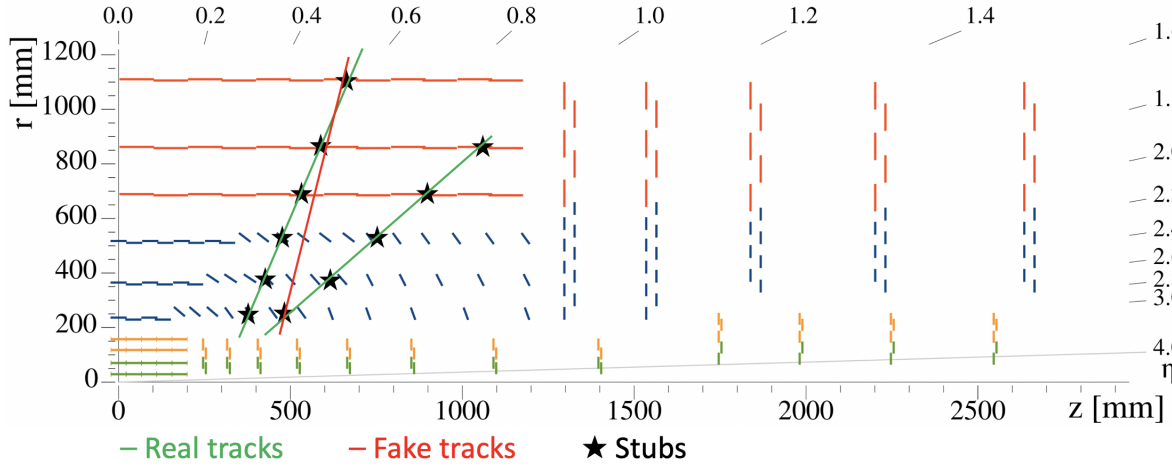


Figure 6.6: A simplistic example of fake vs. real tracks in the outer tracker [112] The green and yellow modules are part of the inner tracker, while the blue and red are part of the outer tracker. The black stars indicate stubs, the green line is a real track, and the red line is a fake track.

Section 6.4.2. The addition of fake tracks in jet reconstruction, for example, can change the central location of the jet and increase its p_T from the true jet value. This consequently increases the event H_T as well. In order to mitigate these effects, the quality of the track can be measured to determine whether each track is fake and should therefore be removed from the track collection.

6.5 Track Quality Classification

“Track quality” (which may be referred to as TQ) is the measure of how real or fake a reconstructed track is. This track property is calculated from the other track features mentioned in Table 6.1 and added to the list of features that other algorithms will receive downstream. Thus, each algorithm using the track collection can decide what quality threshold the input tracks must pass, depending on how much fake tracks affect the algorithm performance.

6.5.1 Current Method

The current method of removing fake tracks from the full collection is to make selection requirements on the track χ^2 variables which measure how close the stubs are to the track projection. If the stubs are farther from the projected track path along a particular direction, then the corresponding

Track feature	Cut applied
$\chi^2_{r\phi}/\text{dof}$	< 5
χ^2_{rz}/dof	< 20
χ^2_{bend}	< 2.2

Table 6.3: Track quality cuts applied to the track collection to remove fake tracks before calculating E_T^{miss} .

χ^2 will be larger. Since fake tracks tend to have worse stub-track matching, tracks with too large of a χ^2 can be removed. Table 6.3 shows an example of the selection criteria made on the track collection before calculating track E_T^{miss} . The selection is not perfect and so some fake tracks will remain and some real tracks will be removed after the selection. These track quality cuts will be used as a baseline for the development of a new track quality algorithm.

At the moment, each physics object algorithm must do their own optimization of the χ^2 cuts based on the amount of fake tracks the algorithm can safely allow. This requires scanning three χ^2 variables and rerunning the physics object reconstruction to see which set of cuts achieves the best performance. Sometimes, additional variables are scanned over as well such as z_0 and n_{stub} , which exponentially grows this computational problem. As the track reconstruction and physics object algorithms are continually being developed, this optimization might have to be done a couple of times before anything is finalized. Therefore, the aggregation of all variables useful for fake track rejection into one track quality value can make this optimization much more efficient, as well as simplify the fake rejection process.

6.5.2 Machine Learning Development

The binary classification of a tracker track as real or fake given the track properties is a straightforward machine learning problem. It was decided to develop both a neural network (NN) and a boosted decision tree (BDT) in parallel to see if either model would produce better performance or use fewer FPGA resources. The input features used were η , z_0 , n_{stub} , $n_{laymiss}^{interior}$, χ^2_{bend} , χ^2_{rz}/dof , and

$\chi^2_{r\phi}/\text{dof}$ where all variables can be calculated from the track properties shown in Table 6.1.² p_T was intentionally not used as an input because track p_T is highly influenced by the physics processes occurring in the physics event, and the real or fake-ness of a track should be independent of this physical process. ϕ was initially included as a feature, but was removed during feature selection as it was not seen to be useful for classification.

Both models were pruned in order to minimize the number of resources they would consume on an FPGA. Pruning a machine learning model is the process of compressing the model by removing parts (either BDT trees or NN nodes) that are not critical to classification. Since the task is quite simple, each model can be quite small. The NN contains one input layer, eight hidden layers with eight and four nodes consecutively, then one output node with a softmax activation function. The BDT is composed of 60 trees, each with a max depth of three, and the trees are gradient boosted. The scikit-learn Python package [120] was used to develop both models.

In order to use a wide variety of tracks and avoid bias from the underlying physics process in a proton-proton collision event, tracks from three different simulated processes were used. The first is from top-quark pair production, the second from the decay of a Z boson to two electrons, and the third from the decay of a Z boson to two muons. These samples were also chosen in order to collect numerous amounts of hadrons, muons, and electrons which are three particle types that are important in the analysis of different physics signals. To emphasize equal importance of muons, hadrons, electrons, and fake tracks (the first three are all real tracks), the training set consisted of 5000 tracks from each for a total of 20,000 tracks. Since the models that were being trained were very small, it was not beneficial to increase the size of the training sample. All track types were chosen randomly from a combination of the three samples. Tracks with $p_T < 20$ GeV were not included in the training samples as this range in p_T is overwhelmingly dominated by pileup tracks. Pileup tracks are not as important for physics analyses as the tracks originating from the hard interaction, which generally have higher p_T .

² $n_{laymiss}^{interior}$ is the number of layers in the detector that the track missed within the sequence of those where track stubs were found. For example, if the hit pattern of a track was 1110110, then there was one interior layer missed.

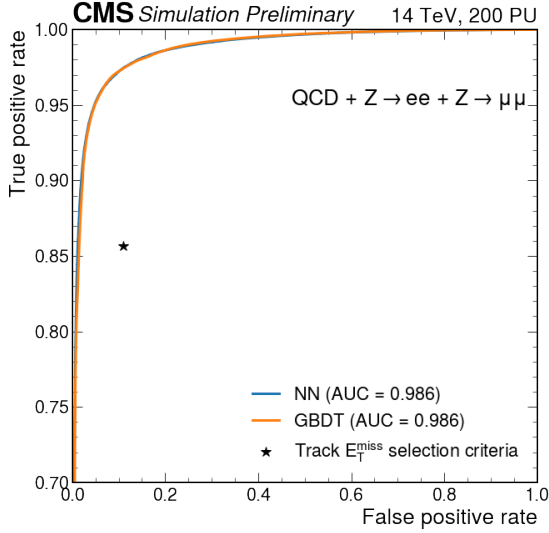


Figure 6.7: A performance comparison between the TQ NN, (G)BDT, and E_T^{miss} cuts shown in Table 6.2. AUC is “area under the curve”.

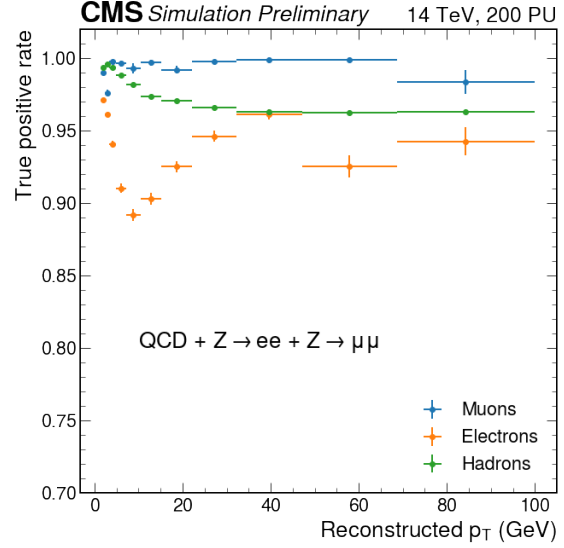


Figure 6.8: BDT performance of each real particle type as a function of track p_T . The decision threshold used in this plot gives a false positive rate of 0.3.

The performance of both models can be found in Figure 6.7. The NN and BDT perform almost exactly the same, and both ML models greatly outperform the cut based E_T^{miss} cuts shown in Table 6.2. For the same false positive rate (fraction of fake tracks classified as real) of the E_T^{miss} cuts, the true positive rate (fraction of real tracks classified as real) for the ML models is ~ 0.12 higher. The performance can be further split by the particle types, as shown in Figure 6.8 for the BDT model. Muons are generally the cleanest tracks, meaning that they pass through the detector in a neat path as they are fairly unbothered by the things around them due to their heavy mass. Electrons, in contrast, are light and leave messy tracks as their paths are easily interrupted through interactions with the detector and other particles. Therefore, the BDT performs the best on muons and the worst on electrons, with hadron performance somewhere in between.

Since performance looked the same in software between the NN and BDT, both models were then tested in simulated hardware to check their resource usage. The HLS4ML package [121] and Conifer package [122] were used to convert the NN and BDT to FPGA-readable language and

Model	Python AUC	HLS AUC	Latency (clk)	LUT%	FF%	DSP%
NN	0.985	0.982	8	0.104	0.029	0.292
BDT	0.986	0.981	3	0.140	0.027	0.0

Table 6.4: Resource usage of both ML TQ models on a simulated Xilinx VU9P FPGA. The HLS [123] tool is used to convert the model code to and FPGA-readable language, LUT stands for look-up-table, FF stands for flip flops, and DSP stands for digital signal processing (where the latter three are the main units present in an FPGA).

simulate their resources on a Xilinx VU9P FPGA with a 240 MHz clock cycle (related to processing speed), initiation interval of 1 (number of clock cycles before accepting new data), and input variables set to an `ap_fixed<10,5>`³ type. The results from the test are found in Table 6.4. From the AUC evaluation, the classification performance of both models are equivalent in both software (python) and simulated hardware (HLS). The slight decrease in performance from python to HLS is due to the switch from floating to fixed point precision. The latency and DSP usage of the BDT is smaller than for the NN, which is a great advantage of the BDT as the TQ algorithm should take up minimal time and computing resources since track reconstruction is a hefty algorithm which will sit on the same boards. DSPs, which are mainly used for mathematical functions, are especially in-demand by track reconstruction and so minimal use of these are desirable. Hence, the BDT was chosen as the TQ ML algorithm.

6.5.3 Hardware Implementation

In hardware, all computation is done in fixed precision and each variable has a set precision based on the accuracy required for the algorithms using them. A 96-bit track “word” is associated with each track coming out of the Track Finder board which holds the information of all of the track features shown in Table 6.1 in addition to the TQ variable. Some of these variables have an extra transformation applied to them before being digitized. The χ^2 variables are transformed into irregular bin values to achieve higher precision at values closer to zero. χ_{bend}^2 , for example, is

³`ap_fixed<A,B>` is defined such that A bits are used in total for precision and B bits are used for the integer portion.

assigned three bits and a value between $[0, 0.5)$ is assigned bin value 0 (000 in three bits) while a value between $[50, \infty)$ is assigned bin value 7 (111 in three bits). z_0 is scaled by a non-base 2 number due to the internal computation of this value in the track reconstruction step. Hence, the TQ BDT must either transform the variables to the floating-point precision format that it expects them in, or take in the variables as they are given in the track word.

In order to avoid extra computation which takes more time and resources, the BDT was retrained with the features formatted in the way that they will be represented in the track word. Therefore, the χ^2 variables were binned, z_0 was scaled by the proper value (20.46912512), and η was converted back to $\tan(\lambda)$. After retraining the BDT, no significant difference was seen in the performance of the classification on all tracks. Therefore, the finalized TQ BDT takes in the track word features and computes a track quality variable which is then added to the track word.

Three bits were assigned to the output TQ variable (sometimes labeled “MVA”) which are used to bin the variable, such as what is done to the χ^2 variables in the track word. Each of the eight bins will provide different levels of performance, and so the physics object reconstruction algorithms using this variable will need to decide what true positive and false positive rates they would like to use. Figure 6.9 shows the performance differences within each bin for the different particle track types when the bins edges are chosen as $[0, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 1]$. This binning scheme provides a wide range of efficiencies and false positive rates for a physics algorithm to choose from, depending on what the algorithm is most sensitive to.

Continued Development It should be noted that the Phase-2 L1T is a current work in progress and therefore the TQ BDT must be continually updated to account for any future changes made that might affect the BDT performance. For example, if the track reconstruction is updated such that the distribution of the variables in the track word are different, the BDT might need to be retrained. Or if the physics object reconstruction algorithms using the TQ variable would like different TQ binning, then that will need to be updated. The current status of the BDT, along with information on how to retrain the classifier, is held at <https://twiki.cern.ch/twiki/bin/view/CMS/L1TrackQuality> (CMS account needed to access link).

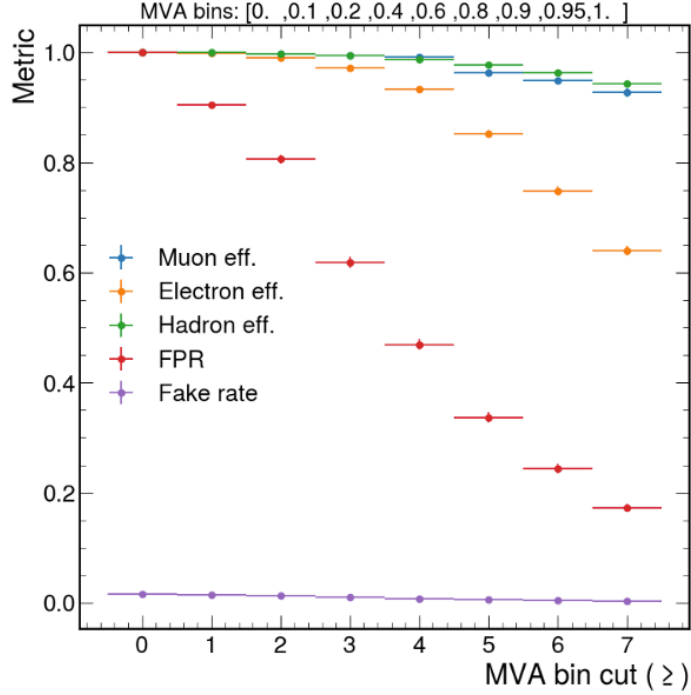


Figure 6.9: Performance of each bin representing the TQ BDT variable in the track word split by particle track type. “eff.” is the true positive rate, FPR stands for false positive rate, fake rate is the percent of fake tracks out of all tracks after selection, and MVA represents the TQ value.

6.6 Applications

Fake tracks can be problematic in the creation of physics objects that rely on track variables as input to the reconstruction algorithms. Inaccuracies due to improper removal of fake tracks can be diminished with proper track quality selection. This section will describe three different physics objects – primary vertex, track jets, and track H_T – and show how the track quality variable can be used to improve their accuracy in reconstruction when compared to using more traditional χ^2 -based fake rejection methods.

6.6.1 Primary Vertex

Proper identification of the primary vertex in a event is essential for the removal of pileup tracks and studying the physics process in the hard interaction. In the Phase-2 L1T, this object will be reconstructed using tracker tracks in the GTT with the *FastHisto* algorithm [114], as mentioned

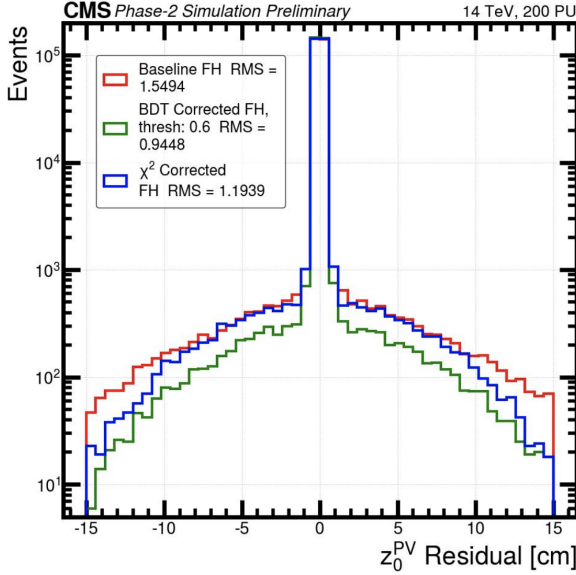


Figure 6.10: z_0^{PV} residual when no fake rejection is done (“baseline” in red), the track quality variable is used for fake rejection (“BDT corrected” in green), and the χ^2 variables are used for fake rejection (“ χ^2 corrected” in blue).

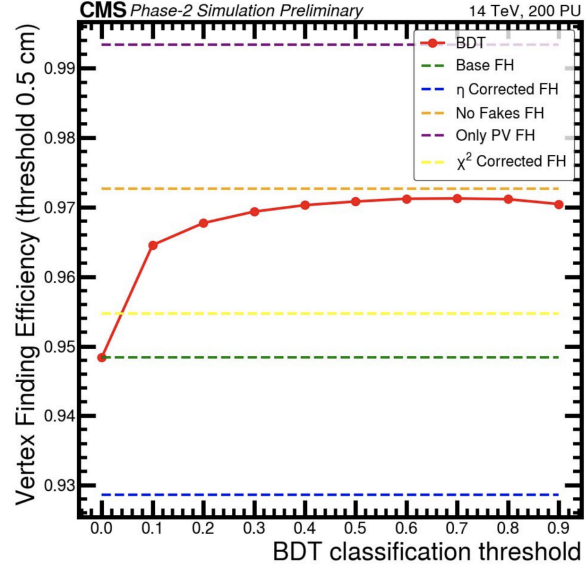


Figure 6.11: z_0^{PV} efficiency versus track quality BDT threshold when no fake rejection is done (“base” in green), the track quality variable (“BDT” in red) or χ^2 variables are used (“ χ^2 corrected” in yellow), all fakes are removed (“no fakes” in orange), and only hard interaction tracks are used (“only PV” in purple). “ η corrected” in blue is another fake rejection method using different η regions in the detector.

in Section 6.4.2. The accuracy of the PV reconstruction can be measured by calculating the z_0 residual of the PV, or the difference between the reconstructed and true z_0 value of the PV using simulated data. If the input track collection contains many fake tracks, the z_0^{PV} residual is expected to be larger on average as these fake tracks created error in the reconstruction. Alternatively, if track quality selection does a good job of removing fake tracks from the collection before PV reconstruction, the z_0^{PV} residual will be more concentrated around zero.

Figure 6.10 shows how the z_0^{PV} residual changes when no track quality cut is made, a track quality cut on the BDT output of ≥ 0.6 is made, and cuts on the χ^2 variables described in Table 6.3 are made. Each of these three fake rejection methods were run in sequence with *FastHisto* over the same events. It can be seen that the residual distribution has the largest tails when no fake rejection

is attempted (red line). Between the χ^2 corrections and track quality BDT corrections, the PV z_0 is most accurately reconstructed when using the newly developed track quality variable.

Another way to measure the performance of the PV reconstruction is to determine how often a true PV is reconstructed well. A measure of the total reconstruction efficiency can be calculated by determining the fraction of true PVs that had a reconstructed PV within $z_0 = 0.5$ cm of the z_0^{true} value (0.5 was chosen as the threshold for a “good” reconstruction). Figure 6.11 shows how the efficiency changes as different fake rejection algorithms are applied. Notably, using the track quality BDT (in red) achieves a higher efficiency than the χ^2 cuts, and also is only about 0.01 in efficiency away from the the efficiency achieved with perfect fake rejection applied. This means that the track quality variable is incredibly successful at reducing all PV reconstruction error associated with fake tracks. The efficiency can be further increased to about 0.993 from 0.972 if all pileup tracks are removed from the input track collection, but that is not a task targeted by the track quality variable.

6.6.2 Track Jets

The reconstruction of track jets in the GTT is through the clustering of tracker tracks, as mentioned in Section 6.4.2. These objects are sensitive to both fake tracks and pileup tracks, and therefore the PV is used in addition to track quality for pileup and fake rejection before the track collection is input into the reconstruction algorithm. The optimal χ^2 cuts implemented before upgrading to the track quality ML variable is shown in Table 6.5, where χ^2/dof is a combination of χ_{rz}^2 and $\chi_{r\phi}^2$, and dz is the z distance between the z_0^{PV} and z_0^{trk} . After switching the χ^2 cuts to using

Track feature	Cut applied
χ^2/dof	< 10
χ_{bend}^2	< 2.2
dz	≤ 1 cm

Table 6.5: Outdated fake and pileup rejection cuts used before reconstructing L1 track jets.

Track feature	Cut applied
MVA	≥ 0.1
dz	≤ 0.55 cm

Table 6.6: Updated fake and pileup rejection cuts used before reconstructing L1 track jets.

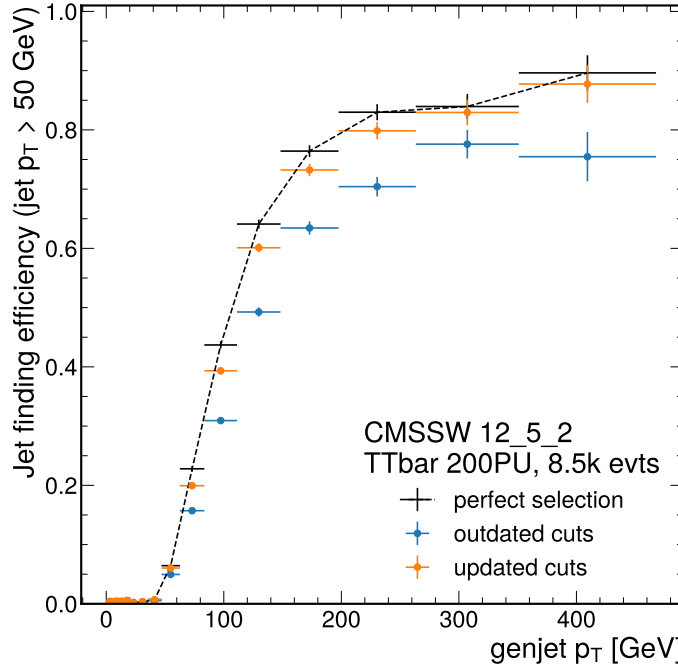


Figure 6.12: Track jet finding efficiency as a function of true (gen)jet p_T for the outdated (blue) and updated (orange) track selection cuts in comparison with perfect track selection (black dashed line). The track jets have an additional $p_T > 50$ GeV constraint.

the track quality ML variable (MVA), the updated optimal cuts became those seen in Table 6.6.

The jet finding efficiency, which is a measure of the fraction of truth-level jets (labeled “genjet”) that have a reconstructed jet within an angular separation of $\Delta R < 0.4$ in simulated data, is used to evaluate the jet reconstruction algorithm. Truth-level jets are reconstructed from simulated particles as opposed to reconstructed particles, and so they are unaffected by track reconstruction inefficiencies. A higher efficiency means that more true jets were reconstructed which is indicative of better reconstruction performance. Figure 6.12 shows the difference in jet finding efficiency for both the outdated and updated track selections. The jet finding efficiency is only looking at track jets with $p_T > 50$ GeV to remove the large collection of jets from pileup collisions that have a low momentum and are not of great interest. With this p_T cut implemented, a large increase in efficiency occurs at around genjet p_T of 50 GeV, which is expected as ideally the genjet p_T is reconstructed with good accuracy and therefore should match a track jet with around the same p_T .

It can be seen that the updated track selection cuts using the track quality ML variable

achieves higher efficiency than the outdated ones. Furthermore, the updated cuts are only about 0.08 away in efficiency from the best scenario shown in black which is when perfect track selection is attained, meaning all fake tracks and pileup tracks are removed. Perfect selection does not have an efficiency of higher than 0.9 due to track reconstruction errors, such as track-level variables being poorly reconstructed or useful tracks being discarded due to bandwidth constraints. These track reconstruction issues are known and are currently being worked on.

6.6.3 Track H_T

As a reminder, H_T is the scalar sum of all jet p_T within an event. An additional cut on jet $p_T > 3$ GeV is made as it was seen to increase the H_T trigger efficiency, which will be described further below. Due to bandwidth constraints, not all events can be stored and sent further down the L1T pipeline. The rate of events budgeted for selection based on the track H_T physics object is around 25 kHz. Since high H_T events are of interest, a selection on all events with H_T greater than some threshold can be used to maintain the 25 kHz rate. That threshold can be determined in Figure 6.13 by looking at the rate of events for different H_T thresholds. For both outdated and updated track selection cuts mentioned in Tables 6.5 and 6.6, the H_T threshold that maintains a 25 kHz rate is 207 GeV, which can be seen in Figure 6.13.

With this H_T threshold determined, an H_T trigger efficiency can be evaluated. The trigger efficiency is the fraction of events that have an H_T above the threshold set by the 25 kHz rate. Figure 6.14 shows the difference in the efficiency when the track selection cuts are the outdated or updated set. In general, the updated cuts – which use the track quality variable – have an increase in efficiency of about 0.1 over the outdated cuts. The efficiency curve for the updated cuts also rises at a quicker rate, which is desired so that a selection on the offline event H_T , like what is done in the emerging jets analysis in Section 4.4.3, is more efficient.

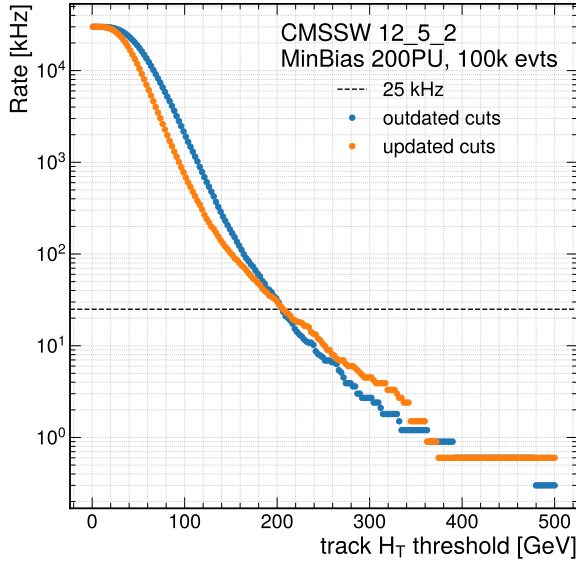


Figure 6.13: The track H_T rate as a function of different H_T thresholds with the outdated and updated track selection cuts. The black dashed line indicates the 25 kHz rate budgeted for the track H_T object.

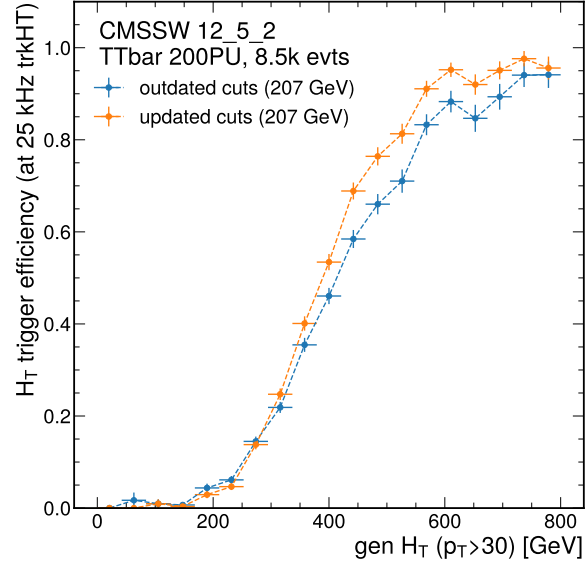


Figure 6.14: The track H_T trigger efficiency as a function of the true (gen) H_T for the outdated and updated track selection cuts. Only true (gen)jets with $p_T > 30$ GeV were included in the gen H_T calculation.

6.7 Displaced Track Quality

In the track reconstruction summary in Section 6.4.1, it was mentioned that the Kalman filter fits the tracks with the constraint that the tracks originate from the proton-proton collision point. For the overwhelming majority of particles that are created from the collisions, this is true and is therefore useful to apply as a constraint as it uses one less free parameter in fitting. If this constraint is dropped, however, then tracks which are displaced from the collision point are also capable of being reconstructed.

6.7.1 Extended Tracking

Many physics signals of interest produce displaced particles, such as long-lived particles and theoretical dark matter signals like the emerging jets signal described in Chapter 4. This motivated the development of the extended tracking algorithm which reconstructs tracks in the same way as

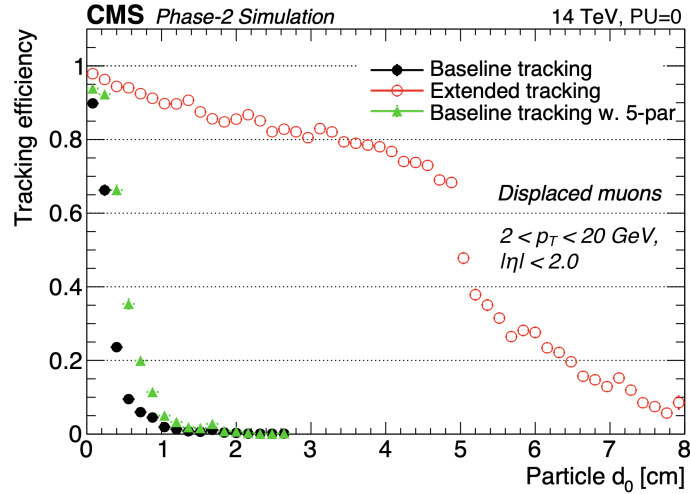


Figure 6.15: Tracking efficiency as a function of true particle d_0 on a displaced muon simulated sample for the standard “baseline” tracking (black), baseline tracking with an additional d_0 parameter allowed to be fit by the Kalman filter (green), and extended tracking (red) [114].

the prompt tracking (Section 6.4.1) aside from three major changes:

- (1) track seeds can also be formed from three consecutive layers of stubs, instead of just two,
- (2) an additional displacement parameter, called the transverse impact parameter d_0 , is also fit by the Kalman filter and included in the output track features, and
- (3) the z_0 position of the track can now extend as far as $|z_0| < 30$ cm, instead of 15 cm.

Figure 6.15 shows how much the efficiency of tracking on a displaced muon simulated sample increases when using extended tracking as opposed to the standard prompt (labeled “baseline”) tracking.

Although extended tracking is widely accepted as a useful addition to the current prompt tracking algorithm, there is still much work to do in order to solidify its place in the Phase-2 L1T. Ongoing work includes optimizing the algorithm for displaced tracks in all regions of the detector, developing efficient truncation techniques to ensure the output tracks meet the bandwidth required (maximum of 185 tracks per track finder board), and understanding the FPGA resources needed to run the algorithm.

6.7.2 Preliminary Displaced Track Quality Classifier

The tracks that are output from the extended tracking algorithm have all of the same variables as those mentioned in Table 6.1, in addition to the track displacement d_0 . The TQ BDT which was optimized for prompt tracks performs well on the extended tracking tracks with very small displacement ($|d_0| < 1$ cm), but the performance diminishes quickly for tracks with larger displacement. This motivated the creation of a displaced TQ classifier which focused on proper fake track rejection for displaced tracks.

Much of the development of the ML track quality classifier for displaced tracks follows what was done with the prompt TQ BDT. The input features are η , z_0 , n_{stub} , $n_{laymiss}^{interior}$, χ_{bend}^2 , χ_{rz}^2/dof , $\chi_{r\phi}^2/\text{dof}$, and d_0 , and the BDT is gradient boosted with 150 trees and a maximum depth of 4. The training and testing samples come from a displaced muon, exotic Higgs, and minimum bias simulated sample,⁴ all of which produce many displaced tracks. The training set consisted of 10,000 tracks, where half were real and half were fake tracks.

Since the purpose of this displaced TQ BDT was to focus on displaced tracks, all non-displaced tracks, defined as those with $|d_0| < 1$ cm, were removed from the training sample. These tracks can use the prompt TQ BDT for fake rejection. The distribution of track d_0 is highly skewed towards zero, so even with the removal of the non-displaced tracks, the BDT will naively learn to put more emphasis on correctly classifying tracks with smaller d_0 . To mitigate this effect, a weight was applied to each track, given by

$$w = |d_0| \times 0.1, \quad (6.1)$$

which puts approximately equal importance on all tracks along d_0 in the training sample, as seen in Figure 6.16.

The overall performance of the displaced and prompt TQ BDT on displaced tracks can be seen in Figure 6.17. The true positive rate as a function of track d_0 for both classifier can be seen

⁴The displaced muons are simulated directly and not through a physical process, the exotic Higgs process decays into a pair of light spin-0 ϕ particles [124], and the minimum bias sample contains only pileup interactions and no hard interaction.

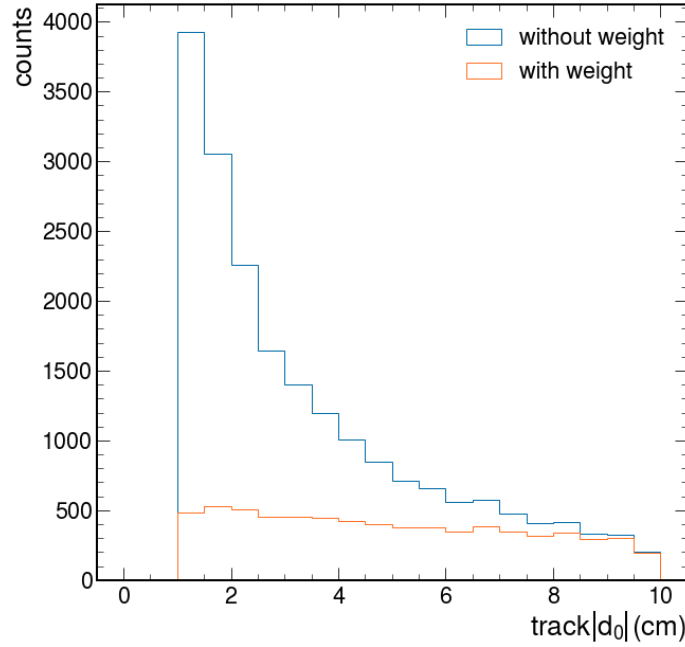


Figure 6.16: How the displaced TQ BDT training sample d_0 distribution (blue) is affected by the weighting (orange).

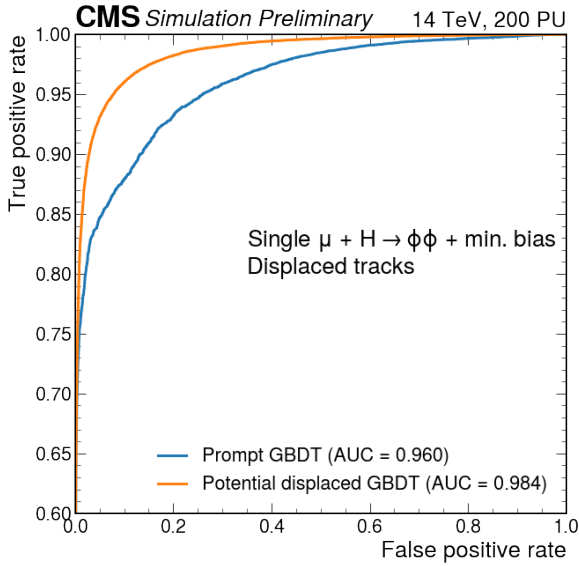


Figure 6.17: A performance comparison between the prompt (blue) and displaced (orange) TQ BDT on displaced tracks with $|d_0| > 1$ cm.

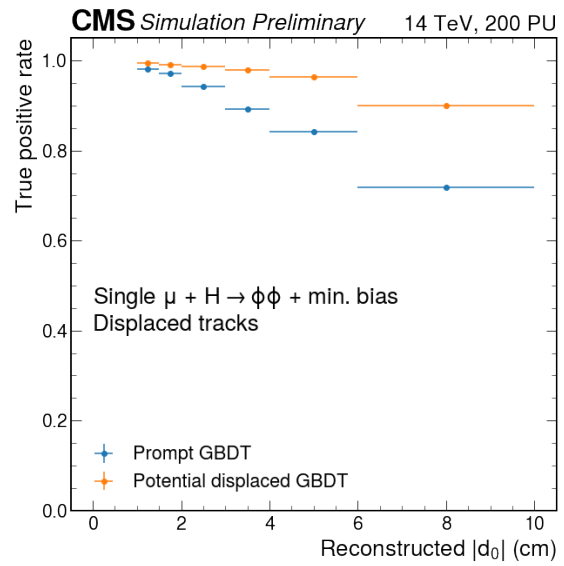


Figure 6.18: Displaced (orange) and prompt (blue) TQ BDT performance on real tracks as a function of track d_0 . The decision threshold used in this plot gives a false positive rate of 0.15.

in Figure 6.18. In general, the displaced TQ BDT achieves a true positive rate of > 0.9 along the full d_0 range while the prompt TQ BDT drops down to ~ 0.73 for the tracks with a displacement between 6 and 10 cm. The displaced BDT is labelled as “potential” as this classifier has not been finalized and still requires more studies to see if it is the most efficient classifier for displaced tracks. What can be seen in these two figures, however, is that a fake rejection on displaced tracks can be improved a great amount if a displaced-specific track quality algorithm is used. This displaced TQ BDT has not yet been tested in the reconstruction of any physics object.

6.8 Final remarks

It is clear from the applications of the TQ BDT seen in Section 6.6 that the use of ML to create a track quality variable focused on fake rejection helps with the reconstruction of many physics objects on the Phase-2 L1T. This method of track quality greatly improves on the χ^2 track quality selections done previously. In addition, the ML TQ variable takes a minimal amount of resources and time to compute and is therefore an overall positive addition to the Phase-2 L1T. The future performance of the TQ BDT will not remain stable as L1 tracking is continually under-development, so it is important that this affect is closely monitored and changes to the TQ BDT are made accordingly.

If the extended tracking algorithm becomes an official part of the Phase-2 L1T, then fake track rejection will also be a crucial component to physics objects reconstructed from displaced tracks. The TQ BDT, which is optimized for performance on prompt tracks, does not perform as well on displaced tracks. It therefore may be beneficial to create a displaced-specific TQ BDT which can greatly increase the performance of track quality classification on displaced tracks, as seen in Section 6.7. The displaced TQ classifier described here is preliminary, but shows how displaced fake rejection can be improved using this technique.

Chapter 7

Conclusion

The objective of upgrading the LHC to the HL-LHC is to increase the integrated luminosity, thereby increasing the potential for physics discovery by experiments located along the accelerator ring, such as the CMS detector. Physics discovery, however, is limited in part by data reconstruction and analysis efficiencies. In order to take advantage of the substantial, yet complicated, data the HL-LHC will provide, high energy physicists must strive for algorithms which will provide the best chance of detecting new physics. In this thesis, I showed how machine learning algorithms can be used to advance high energy physics data reconstruction and analysis, ultimately hastening physics discovery.

Chapter 4 demonstrated how machine learning can be used to substantially improve a beyond-standard-model physics search. The emerging jets analysis looked for dark matter signatures with an unflavored and flavor-aligned quark coupling scenario in CMS Run 2 data. I showed how the use of a graph neural network applied to jet classification can significantly increase sensitivity to signal events compared to a traditional cut-based method. As a consequence of switching the analysis method from using a cut-based method to machine learning techniques, the exclusion limits were pushed back 150 - 600 GeV in the dark mediator particle mass. The results of this search set the most stringent exclusion limits to date for the unflavored coupling scenario, and they provide the very first exclusion limits for the flavor-aligned coupling scenario.

Motivated by the computing inefficiencies of the emerging jets analysis, Chapter 5 showed how I re-optimized the Fermilab computing facilities with a Triton Inference server for high throughput

machine learning inference. This change led to a speed up of the emerging jets graph neural network inference time by a factor of ~ 50 , reducing data processing times from days to hours. I benchmarked the behavior of the system to exhibit the properties that the Fermilab setup has when the number of inference requests and models being called for inference increases, and these benchmarking tests are publicly available for use in testing future Triton systems. This work has facilitated access to powerful machine learning processors within the typical physics analysis framework, and has already inspired similar computing facilities to reconfigure their setup in a more machine-learning-friendly way.

Chapter 6 showed a machine learning application in physics object reconstruction on the CMS Phase 2 Level-1 Trigger, leading to more refined selection of new and interesting physics data. The Level-1 Track Finder outputs a collection of charged particle tracks which are contaminated with low-quality tracks due to algorithm error. These bad tracks augment physics object reconstruction errors and should therefore be removed. I developed a boosted decision tree to replace a simpler χ^2 selection method tasked with removing these low-quality tracks, resulting in a $\sim 15\%$ increase in jet and H_T reconstruction efficiency. Furthermore, I created a preliminary track quality classifier focused on displaced tracks, which ultimately benefits long-lived and beyond-standard-model physics searches, like emerging jets.

These applications offer only a glimpse of the benefits machine learning can bring to high energy physics. Many other great applications can be found before these, and are in part what inspired this work, and there are certainly many more to come. As the field of machine learning continues to develop quickly, the scope of applications will only widen. The extent of machine learning's utility in high energy physics is not yet known, but rest assured, we are far from running into the limit. The only way to know the benefits that lie ahead is to try.

References

- [1] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, 3rd ed. (Cambridge University Press, 2020) (Cited on p. 3).
- [2] W. Commons, *Standard Model of Elementary Particles*, (2019) https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg (Cited on p. 4).
- [3] K. Collaboration, “Direct neutrino-mass measurement with sub-electronvolt sensitivity”, *Nature Phys.* **18**, 10.1038/s41567-021-01463-1 (2022) 10.1038/s41567-021-01463-1 (Cited on p. 6).
- [4] M. D. Schwartz, *Quantum Field Theory and the Standard Model* (Cambridge University Press, 2013) (Cited on p. 7).
- [5] H. J. W. George B. Arfken and F. E. Harris, *Mathematical Methods for Physicists* (Academic Press, Orlando, FL, 2012) Chap. 4, pp. 211–217 (Cited on p. 8).
- [6] M. Gell-Mann, “Symmetries of Baryons and Mesons”, *Phys. Rev.* **125**, 1067 (1962) 10.1103/PhysRev.125.1067 (Cited on p. 9).
- [7] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman lectures on physics; New millennium ed.* Originally published 1963-1965 (Basic Books, New York, NY, 2010) Chap. The Pauli spin matrices (Cited on p. 11).
- [8] S. Weinberg, “A Model of Leptons”, *Phys. Rev. Lett.* **19**, 1264 (1967) 10.1103/PhysRevLett.19.1264 (Cited on p. 12).
- [9] M. Persic, P. Salucci, and F. Stel, “The universal rotation curve of spiral galaxies — I. The dark matter connection”, *Monthly Notices of the Royal Astronomical Society* **281**, 27 (1996) 10.1093/mnras/278.1.27 (Cited on p. 12).
- [10] D. Clowe et al., “A Direct Empirical Proof of the Existence of Dark Matter”, *The Astrophysical Journal* **648**, 10.1086/508162 (2006) 10.1086/508162 (Cited on p. 12).
- [11] P. Collaboration, “Planck2018 results: VI. Cosmological parameters”, *Astronomy & Astrophysics* **641**, 10.1051/0004-6361/201833910 (2020) 10.1051/0004-6361/201833910 (Cited on p. 12).
- [12] A. I. Lonappan et al., “Bayesian evidences for dark energy models in light of current observational data”, *Phys. Rev. D* **97**, 10.1103/PhysRevD.97.043524 (2018) 10.1103/PhysRevD.97.043524 (Cited on p. 12).
- [13] P. J. E. Peebles and B. Ratra, “The cosmological constant and dark energy”, *Rev. Mod. Phys.* **75**, 10.1103/RevModPhys.75.559 (2003) 10.1103/RevModPhys.75.559 (Cited on p. 12).
- [14] T. Tait, “Dark Matter: Theoretical Overview”, 2016 Aspen Winter Conference on Particle Physics, 2016 (Cited on p. 13).

- [15] V. Collaboration, “VERITAS limits on dark matter annihilation from dwarf galaxies”, [AIP Conference Proceedings](#) **1505**, [10.1063/1.4772353](#) (2012) [10.1063/1.4772353](#) (Cited on p. 13).
- [16] P.-4. Collaboration, “Dark Matter Search Results from the PandaX-4T Commissioning Run”, [Phys. Rev. Lett.](#) **127**, [10.1103/PhysRevLett.127.261802](#) (2021) [10.1103/PhysRevLett.127.261802](#) (Cited on p. 13).
- [17] G. 't Hooft, “Symmetry Breaking through Bell-Jackiw Anomalies”, [Phys. Rev. Lett.](#) **37**, [10.1103/PhysRevLett.37.8](#) (1976) [10.1103/PhysRevLett.37.8](#) (Cited on p. 14).
- [18] A. Collaboration, “Search for Invisible Axion Dark Matter in the 3.3–4.2 μeV Mass Range”, [Phys. Rev. Lett.](#) **127**, [10.1103/PhysRevLett.127.261803](#) (2021) [10.1103/PhysRevLett.127.261803](#) (Cited on p. 14).
- [19] A. De Angelis et al., “Photon propagation and the very high energy γ -ray spectra of blazars: how transparent is the Universe?”, [Monthly Notices of the Royal Astronomical Society: Letters](#) **394**, [10.1111/j.1745-3933.2008.00602.x](#) (2009) [10.1111/j.1745-3933.2008.00602.x](#) (Cited on p. 14).
- [20] D. Gorbunov and A. Panin, “Minimal active-sterile neutrino mixing in seesaw type I mechanism with sterile neutrinos at GeV scale”, [Phys. Rev. D](#) **89**, [10.1103/PhysRevD.89.017302](#) (2014) [10.1103/PhysRevD.89.017302](#) (Cited on p. 14).
- [21] A. Aguilar-Arevalo et al., “Significant Excess of Electronlike Events in the MiniBooNE Short-Baseline Neutrino Experiment”, [Physical Review Letters](#) **121**, [10.1103/physrevlett.121.221801](#) (2018) [10.1103/physrevlett.121.221801](#) (Cited on p. 14).
- [22] A. H. G. Peter et al., “Cosmological simulations with self-interacting dark matter – II. Halo shapes versus observations”, [Monthly Notices of the Royal Astronomical Society](#) **430**, [10.1093/mnras/sts535](#) (2013) [10.1093/mnras/sts535](#) (Cited on p. 14).
- [23] M. Vogelsberger, J. Zavala, and A. Loeb, “Subhaloes in self-interacting galactic dark matter haloes: Self-interacting galactic dark matter haloes”, [Monthly Notices of the Royal Astronomical Society](#) **423**, [10.1111/j.1365-2966.2012.21182.x](#) (2012) [10.1111/j.1365-2966.2012.21182.x](#) (Cited on p. 14).
- [24] J. Zavala, M. Vogelsberger, and M. G. Walker, “Constraining self-interacting dark matter with the Milky Way’s dwarf spheroidals”, [Monthly Notices of the Royal Astronomical Society: Letters](#) **431**, [10.1093/mnrasl/sls053](#) (2013) [10.1093/mnrasl/sls053](#) (Cited on p. 14).
- [25] M. Fabbrichesi, E. Gabrielli, and G. Lanfranchi, *The Physics of the Dark Photon: A Primer* (Springer International Publishing, 2021), [10.1007/978-3-030-62519-1](#) (Cited on p. 14).
- [26] D. Curtin et al., “Exotic decays of the 125 GeV Higgs boson”, [Physical Review D](#) **90**, [10.1103/physrevd.90.075004](#) (2014) [10.1103/physrevd.90.075004](#) (Cited on p. 14).
- [27] M. J. Strassler and K. M. Zurek, “Echoes of a hidden valley at hadron colliders”, [Physics Letters B](#) **651**, [10.1016/j.physletb.2007.06.055](#) (2007) [10.1016/j.physletb.2007.06.055](#) (Cited on p. 15).
- [28] O. Baker, A. Afanasev, T. Lagouri, J. Pan, and C. Weber, “Particle Physics of the Dark Sector”, [Symmetry](#) **14**, [10.3390/sym14112238](#) (2022) [10.3390/sym14112238](#) (Cited on p. 15).
- [29] E. Mobs, “The CERN accelerator complex in 2019. Complexe des accélérateurs du CERN en 2019”, [General Photo](#) (2019) (Cited on p. 17).

- [30] A. Collaboration, “The ATLAS Experiment at the CERN Large Hadron Collider”, *JINST* **3**, Also published by CERN Geneva in 2010, S08003 (2008) [10.1088/1748-0221/3/08/S08003](https://arxiv.org/abs/10.1088/1748-0221/3/08/S08003) (Cited on pp. 17, 96).
- [31] CMS Collaboration, “The CMS Experiment at the CERN LHC”, *JINST* **3**, S08004 (2008) [10.1088/1748-0221/3/08/S08004](https://arxiv.org/abs/10.1088/1748-0221/3/08/S08004) (Cited on pp. 17, 20, 29, 96, 99).
- [32] L. Collaboration, “The LHCb Detector at the LHC”, *JINST* **3**, Also published by CERN Geneva in 2010, S08005 (2008) [10.1088/1748-0221/3/08/S08005](https://arxiv.org/abs/10.1088/1748-0221/3/08/S08005) (Cited on p. 17).
- [33] A. Collaboration, “The ALICE experiment at the CERN LHC. A Large Ion Collider Experiment”, *JINST* **3**, Also published by CERN Geneva in 2010, S08002 (2008) [10.1088/1748-0221/3/08/S08002](https://arxiv.org/abs/10.1088/1748-0221/3/08/S08002) (Cited on p. 17).
- [34] A. M. Sirunyan et al., “Precision luminosity measurement in proton-proton collisions at $\sqrt{s} = 13$ TeV in 2015 and 2016 at CMS”, *Eur. Phys. J. C* **81**, 800 (2021) [10.1140/epjc/s10052-021-09538-2](https://arxiv.org/abs/10.1140/epjc/s10052-021-09538-2) (Cited on pp. 18, 71).
- [35] CMS Collaboration, *CMS luminosity measurement for the 2017 data-taking period at $\sqrt{s} = 13$ TeV*, CMS Physics Analysis Summary CMS-PAS-LUM-17-004 (2018) (Cited on pp. 18, 71).
- [36] CMS Collaboration, *CMS luminosity measurement for the 2018 data-taking period at $\sqrt{s} = 13$ TeV*, CMS Physics Analysis Summary CMS-PAS-LUM-18-002 (2019) (Cited on pp. 18, 71).
- [37] C. Collaboration, *Public CMS Luminosity Information*, <https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults> (Cited on pp. 18, 19).
- [38] T. Sakuma (CMS), “Cutaway diagrams of CMS detector”, (2019) (Cited on p. 20).
- [39] D. Barney, “CMS Detector Slice”, CMS Collection., 2016 (Cited on p. 21).
- [40] I. Neutelings, “CMS coordinate system” (Cited on p. 22).
- [41] C. Collaboration, *CMS Tracker Detector Performance Results*, <https://twiki.cern.ch/twiki/bin/view/CMSPublic/DPGResultsTRK> (Cited on p. 24).
- [42] C. Collaboration, “The CMS Phase-1 Pixel Detector Upgrade”, *JINST* **16**, P02027 (2021) [10.1088/1748-0221/16/02/P02027](https://arxiv.org/abs/10.1088/1748-0221/16/02/P02027) (Cited on p. 24).
- [43] B. Marzocchi (CMS), “Simulation of the MCS electromagnetic calorimeter response at the energy and intensity frontier”, *J. Phys.: Conf. Ser.*, [10.1088/1742-6596/1162/1/012007](https://arxiv.org/abs/10.1088/1742-6596/1162/1/012007) (2019) [10.1088/1742-6596/1162/1/012007](https://arxiv.org/abs/10.1088/1742-6596/1162/1/012007) (Cited on p. 26).
- [44] C. Collaboration, “Calibration of the CMS hadron calorimeters using proton-proton collision data at $\sqrt{s} = 13$ TeV”, *JINST* **15**, P05002 (2020) [10.1088/1748-0221/15/05/P05002](https://arxiv.org/abs/10.1088/1748-0221/15/05/P05002) (Cited on p. 27).
- [45] *Resistive plate chambers*, <https://cms.cern/detector/detecting-muons/resistive-plate-chambers> (Cited on p. 29).
- [46] T. Virdee, A. Petrilli, and A. Ball, *CMS High Level Trigger*, tech. rep. (CERN, Geneva, 2007) (Cited on p. 31).
- [47] *Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and MET*, tech. rep. (CERN, Geneva, 2009) (Cited on pp. 32, 35).
- [48] W. Adam et al., *Track Reconstruction in the CMS tracker*, tech. rep. (CERN, Geneva, 2006) (Cited on p. 32).

- [49] C. Collaboration, “Description and performance of track and primary-vertex reconstruction with the CMS tracker”, *JINST* **9**, P10009 (2014) [10.1088/1748-0221/9/10/P10009](#) (Cited on pp. 33, 34).
- [50] W. Adam et al., *Reconstruction of Electrons with the Gaussian-Sum Filter in the CMS Tracker at the LHC*, tech. rep. (CERN, Geneva, 2005) (Cited on p. 35).
- [51] G. P. S. Matteo Cacciari and G. Soyez, “The anti- k_t jet clustering algorithm”, *J. High Energ. Phys.* **2008**, [10.1088/1126-6708/2008/04/063](#) (2008) [10.1088/1126-6708/2008/04/063](#) (Cited on p. 36).
- [52] G. Bertone, D. Hooper, and J. Silk, “Particle dark matter: evidence, candidates and constraints”, *Physics Reports* **405**, 279 (2005) [10.1016/j.physrep.2004.08.031](#) (Cited on p. 39).
- [53] P. Schwaller, D. Stolarski, and A. Weiler, “Emerging Jets”, *J. High Energ. Phys.* **5**, [10.1007/JHEP05\(2015\)059](#) (2015) [10.1007/JHEP05\(2015\)059](#) (Cited on pp. 39, 41, 42).
- [54] Y. Bai and P. Schwaller, “Scale of dark QCD”, *Phys. Rev. D* **89**, [10.1103/PhysRevD.89.063522](#) (2014) [10.1103/PhysRevD.89.063522](#) (Cited on pp. 39, 41).
- [55] C. Collaboration, “Search for new physics with emerging jets in proton-proton collisions at $\sqrt{s} = 13$ TeV”, (2024) (Cited on pp. 40, 89–91, 154).
- [56] S. Renner and P. Schwaller, “A flavoured dark sector”, *J. High Energ. Phys.* **8**, [10.1007/JHEP08\(2018\)052](#) (2018) [10.1007/JHEP08\(2018\)052](#) (Cited on p. 42).
- [57] L. Carloni and T. Sjostrand, “Visible Effects of Invisible Hidden Valley Radiation”, *J. High Energ. Phys.* **9**, [10.1007/JHEP09\(2010\)105](#) (2010) [10.1007/JHEP09\(2010\)105](#) (Cited on p. 44).
- [58] C. Bierlich et al., “A comprehensive guide to the physics and usage of PYTHIA 8.3”, [10.48550/arXiv.2203.11601](#) [10.48550/arXiv.2203.11601](#) (Cited on pp. 44, 72).
- [59] *Modifications of existing Hidden Valley model in pythia8*, <https://github.com/kpedro88/pythia8/blob/emg/230/src/HiddenValleyFragmentation.cc> (Cited on p. 44).
- [60] C. Collaboration, “Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV. Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV”, *JINST* **12**, P02014 (2017) [10.1088/1748-0221/12/02/P02014](#) (Cited on p. 48).
- [61] *Jet identification for 13 TeV UL*, <https://twiki.cern.ch/twiki/bin/view/CMS/JetID13TeVUL> (Cited on p. 48).
- [62] C. Collaboration, “Search for new particles decaying to a jet and an emerging jet”, *J. High Energ. Phys.* **2019**, [10.1007/JHEP02\(2019\)179](#) (2019) [10.1007/JHEP02\(2019\)179](#) (Cited on pp. 49, 50, 65, 79, 84, 88, 89, 91).
- [63] S. Thais et al., “Graph Neural Networks in Particle Physics: Implementations, Innovations, and Challenges”, [arXiv:2203.12852](#) [arXiv:2203.12852](#) (Cited on p. 53).
- [64] H. Qu and L. Gouskos, “Jet tagging via particle clouds”, *Phys. Rev. D* **101**, 056019 (2020) [10.1103/PhysRevD.101.056019](#) (Cited on p. 53).
- [65] I. Cohen et al., “Pearson correlation coefficient”, Noise reduction in speech processing, 1 (2009) (Cited on p. 55).
- [66] *Weaver*, <https://github.com/hqucms/weaver-core> (Cited on pp. 58, 158).

- [67] J. MacQueen, “Some methods for classification and analysis of multivariate observations.”, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **1**, 281 (1967) [10.1007/BF02289263](#) (Cited on p. 65).
- [68] R. L. Thorndike, “Who belongs in the family?”, *Psychometrika* **18**, 267 (1953) [10.1007/BF02289263](#) (Cited on p. 65).
- [69] A. M. Sirunyan et al. (CMS), “Measurement of the inelastic proton-proton cross section at $\sqrt{s} = 13$ TeV”, *JHEP* **07**, 161 (2018) [10.1007/JHEP07\(2018\)161](#) (Cited on p. 71).
- [70] R. D. Ball et al. (NNPDF), “Parton distributions with QED corrections”, *Nucl. Phys. B* **877**, 290 (2013) [10.1016/j.nuclphysb.2013.10.010](#) (Cited on p. 72).
- [71] C. Collaboration, “Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC”, *JINST* **16**, P05014 (2021) [10.1088/1748-0221/16/05/P05014](#) (Cited on p. 75).
- [72] E. Bols et al., “Jet Flavour Classification Using DeepJet”, *JINST* **15**, P12012 (2020) [10.1088/1748-0221/15/12/P12012](#) (Cited on p. 78).
- [73] *CMS Higgs Combination Toolkit*, <https://github.com/cms-analysis/HiggsAnalysis-CombinedLimit> (version 8.0.1) (Cited on p. 85).
- [74] A. L. Read, “Presentation of search results: the CL_s technique”, *J. Phys. G* **28**, 2693 (2002) [10.1088/0954-3899/28/10/313](#) (Cited on p. 85).
- [75] G. Cowan et al., “Asymptotic formulae for likelihood-based tests of new physics”, *The European Physical Journal C* **71**, [10.1140/epjc/s10052-011-1554-0](#) (2011) [10.1140/epjc/s10052-011-1554-0](#) (Cited on p. 85).
- [76] A. L. Read, “Presentation of search results: the CLs technique”, *Journal of Physics G: Nuclear and Particle Physics* **28**, [10.1088/0954-3899/28/10/313](#) (2002) [10.1088/0954-3899/28/10/313](#) (Cited on p. 85).
- [77] C. Savard, “Using NVIDIA Triton Server for Inference-as-a-Service at Fermilab”, *Fast Machine Learning for Science Workshop*, 2023 (Cited on p. 94).
- [78] C. Savard et al., “Optimizing High Throughput Inference on Graph Neural Networks at Shared Computing Facilities with the NVIDIA Triton Inference Server”, [10.48550/arXiv.2312.06838](#) [10.48550/arXiv.2312.06838](#) (Cited on pp. 94, 100, 101, 103–105, 107–110, 158).
- [79] M. Adamec et al., “Coffea-casa: an analysis facility prototype”, *EPJ Web Conf.* **251**, 02061 (2021) [10.1051/epjconf/202125102061](#) (Cited on p. 95).
- [80] K. Albertsson et al., “Machine Learning in High Energy Physics Community White Paper”, *J. Phys. Conf. Ser.* **1085**, 022008 (2018) [10.1088/1742-6596/1085/2/022008](#) (Cited on pp. 95, 100).
- [81] D. Guest, K. Cranmer, and D. Whiteson, “Deep Learning and Its Application to LHC Physics”, *Annual Review of Nuclear and Particle Science* **68**, 161 (2018) [10.1146/annurev-nucl-101917-021019](#) (Cited on p. 95).
- [82] E. Buber and B. Diri, “Performance Analysis and CPU vs GPU Comparison for Deep Learning”, *International Conference on Control Engineering & Information Technology Proceedings*, 1 (2018) [10.1109/CEIT.2018.8751930](#) (Cited on pp. 95, 97).
- [83] Y. Wang, G.-Y. Wei, and D. Brooks, “Benchmarking TPU, GPU, and CPU Platforms for Deep Learning”, (2019) (Cited on pp. 95, 111).

- [84] M. Baker, G. C. Fox, and H. W. Yau, “Cluster Computing Review”, Northeast Parallel Architecture Center (1995) (Cited on pp. 95, 96).
- [85] I. Bloch, “The LHC physics center”, *Nucl. Phys. B Proc. Suppl.* **177-178**, 261 (2008) 10.1016/j.nuclphysbps.2007.11.121 (Cited on pp. 95, 96, 159).
- [86] D. C. Marinescu, “Chapter 3 - Parallel processing and distributed computing”, in *Cloud Computing (Third Edition)* (Morgan Kaufmann, Burlington, MA, USA, 2023), pp. 41–94, 10.1016/B978-0-32-385277-7.00010-5 (Cited on p. 97).
- [87] NVIDIA Corporation, *Triton Inference Server: An Optimized Cloud and Edge Inferencing Solution*. <https://github.com/triton-inference-server> (Cited on p. 98).
- [88] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., Red Hook, NY, USA, 2019), pp. 8024–8035 (Cited on pp. 98, 159).
- [89] Martín Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015 (Cited on p. 98).
- [90] Y. Inoue, “Queueing analysis of GPU-based inference servers with dynamic batching: A closed-form characterization”, *Performance Evaluation* **147**, 102183 (2021) 10.1016/j.peva.2020.102183 (Cited on p. 98).
- [91] NVIDIA Corporation, *Triton Architecture*, https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/user_guide/architecture.html (Cited on p. 98).
- [92] *okd: the Community Distribution of Kubernetes that powers Red Hat OpenShift*, <https://www.okd.io> (Cited on p. 99).
- [93] *Red Hat OpenShift*, <https://www.redhat.com/en/technologies/cloud-computing/openshift> (Cited on p. 99).
- [94] *kubernetes*, <https://kubernetes.io> (Cited on p. 99).
- [95] HTCondor Team, “Distributed Computing in Practice: The Condor Experience”, *Concurrency and Computation: Practice and Experience* **17**, <https://github.com/htcondor/htcondor>, 323 (Cited on p. 100).
- [96] *CRAB Server*, <https://github.com/dmwm/CRABServer> (Cited on p. 100).
- [97] *Dask*, <https://github.com/dask/dask> (Cited on p. 100).
- [98] *MinIO, High Performance Object Storage for AI*, <https://min.io> (Cited on p. 101).
- [99] *gRPC, a high performance, open source universal RPC framework*, <https://grpc.io> (Cited on p. 102).
- [100] *HAProxy, the Reliable High Performance TCP/HTTP Load Balancer*, <https://www.haproxy.org> (Cited on p. 102).
- [101] *nginx*, <https://nginx.org/en> (Cited on p. 102).
- [102] *Prometheus monitoring system & time series database*, <https://prometheus.io> (Cited on p. 102).
- [103] *Grafana Labs*, <https://grafana.com> (Cited on p. 102).
- [104] H. Qu and L. Gouskos, “ParticleNet: Jet Tagging via Particle Clouds”, *Phys. Rev. D* **101**, 056019 (2020) 10.1103/PhysRevD.101.056019 (Cited on pp. 102, 106, 158).

- [105] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, in [2016 IEEE Conference on Computer Vision and Pattern Recognition \(CVPR\)](#) (2016), pp. 770–778, [10.1109/CVPR.2016.90](#) (Cited on p. 105).
- [106] A. Canziani, A. Paszke, and E. Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications”, (2017) (Cited on p. 105).
- [107] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, in [Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining](#), KDD ’16 (2016), [10.1145/2939672.2939785](#) (Cited on p. 106).
- [108] D. S. Rankin et al., “FPGAs-as-a-Service Toolkit (FaaSST)”, [2020 IEEE/ACM International Workshop on Heterogeneous High-performance Reconfigurable Computing \(H2RC\)](#), 38 (2020) [10.1109/H2RC51942.2020.00010](#) (Cited on p. 111).
- [109] C. Savard, “Overview of the HL-LHC Upgrade for the CMS Level-1 Trigger”, [EPJ Web Conf. 251](#), 02054 (2021) [10.1051/epjconf/202125102054](#) (Cited on pp. 113, 118).
- [110] C. Savard, “Applying Machine Learning to Particle Track Identification in the L1-Trigger of the CMS Detector”, [Machine Learning and the Physical Sciences \(ML4PS\) NeurIPS Workshop](#), 2019 (Cited on p. 113).
- [111] C. Savard, “Level 1 trigger track quality machine learning models on FPGAs for the Phase 2 upgrade of the CMS experiment”, [Fast Machine Learning for Science Workshop](#), 2020 (Cited on p. 113).
- [112] C. Savard, “Level-1 Track Quality Evaluation at CMS for the HL-LHC”, in [41st International Conference on High Energy Physics \(ICHEP 2022\)](#) (2022) (Cited on pp. 114, 123).
- [113] O. Aberla et al. (CERN LHC), *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*, CERN Yellow Reports: Monographs CERN-2020-010 (2020) (Cited on p. 114).
- [114] C. Collaboration, *The Phase-2 Upgrade of the CMS Level-1 Trigger*, Detectors and Experimental Techniques CERN-LHCC-2020-004 ; CMS-TDR-021 (2020) (Cited on pp. 115, 118, 121, 122, 129, 135).
- [115] B. Radburn-Smith, “Overview of the HL-LHC Upgrade for the CMS Level-1 Trigger”, [41st International Conference on High Energy Physics](#), 2022 (Cited on pp. 115, 117).
- [116] I. Kuon, R. Tessier, and J. Rose, “FPGA Architecture: Survey and Challenges”, [Foundations and Trends® in Electronic Design Automation](#) **2**, 135 (2008) [10.1561/10000000005](#) (Cited on p. 117).
- [117] R. Aggleton et al., “An FPGA based track finder for the L1 trigger of the CMS experiment at the High Luminosity LHC”, [Journal of Instrumentation](#) **12**, P12019 (2017) [10.1088/1748-0221/12/12/P12019](#) (Cited on pp. 118, 119).
- [118] C. Collaboration, *The Phase-2 Upgrade of the CMS Tracker*, Detectors and Experimental Techniques CERN-LHCC-2017-009 ; CMS-TDR-014 (2017) (Cited on p. 119).
- [119] C. Collaboration, *The Level-1 Track Finder for the CMS High-Luminosity LHC Upgrade*, Detectors and Experimental Techniques CMS-CR-2022-278 (2022) (Cited on p. 119).
- [120] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python”, [Journal of Machine Learning Research](#) **12**, 2825 (2011) (Cited on p. 125).
- [121] FastML Team, *fastmachinelearning/hls4ml*, version v0.8.1, 2023, [10.5281/zenodo.1201549](#) (Cited on p. 126).

- [122] S. Summers et al., “Fast inference of Boosted Decision Trees in FPGAs for particle physics”, *Journal of Instrumentation* **15**, P05026 (2020) [10.1088/1748-0221/15/05/P05026](https://doi.org/10.1088/1748-0221/15/05/P05026) (Cited on p. 126).
- [123] P. Coussy et al., “An Introduction to High-Level Synthesis”, *IEEE Design & Test of Computers* **26**, 8 (2009) [10.1109/MDT.2009.69](https://doi.org/10.1109/MDT.2009.69) (Cited on p. 127).
- [124] F. Kling, J. M. No, and S. Su, “Anatomy of exotic Higgs decays in 2HDM”, *Journal of High Energy Physics* **2016**, [10.1007/jhep09\(2016\)093](https://doi.org/10.1007/jhep09(2016)093) (2016) [10.1007/jhep09\(2016\)093](https://doi.org/10.1007/jhep09(2016)093) (Cited on p. 136).
- [125] *Kubernetes: Horizontal Pod Autoscaling*, <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/> (Cited on p. 156).

Appendix A

Emerging Jet Specifics

A.1 Impact Parameter Transformation

A motivation behind transforming machine learning inputs is to constrain the input ranges within a similar small range to help with training. In training a neural network, weights applied to input values and between network layers are constantly being updated to optimize the performance of the network. If inputs have a large range, then the weights may need to change drastically from their initial values in order to arrive at the optimal values. The amount that weights can change each training epoch is constrained (decided upon by the user, also called the “learning rate”), and so a drastic change in the weight required combined with minimal updates allowed results in a very long training time. Increasing the amount in which a weight can change per epoch can result in shorter training times, but may consequently achieve poorer performance as there is not as much granularity in the weight values. It is instead more popular to transform all inputs to a smaller, similar range, and to use a small learning rate.

In this analysis, both the p_T and impact parameters d_{xy} and d_z are unconstrained values with the range $(2, \infty)$ and $(-\infty, \infty)$, respectively. Although the transformation chosen for an input might not make a large difference, this analysis decided to use a natural log on each of these features. The natural log is a standard transformation used in machine learning to reduce the range of the inputs which can make training more efficient. Since the natural logarithm $\ln(x)$ is undefined at $x \in (-\infty, 0]$ and positive definite, the absolute value of the impact parameter (which is symmetric about 0) is

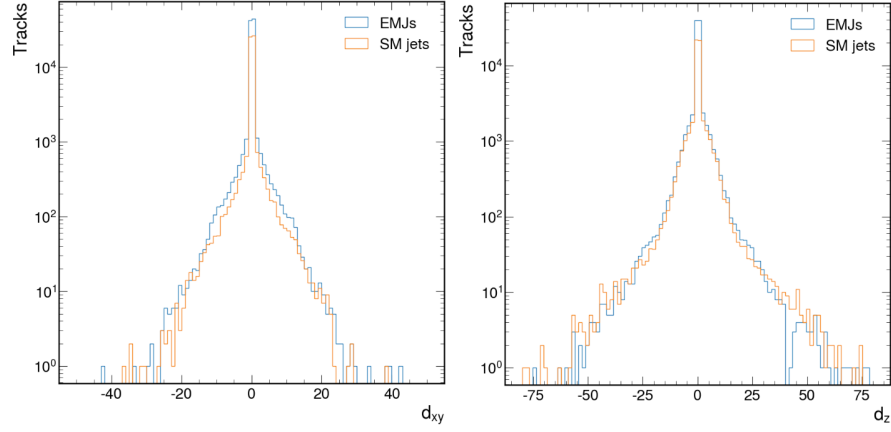


Figure A.1: Distribution of the track impact parameters along the xy -plane and z -axis for emerging jets (blue) and SM jets (orange).

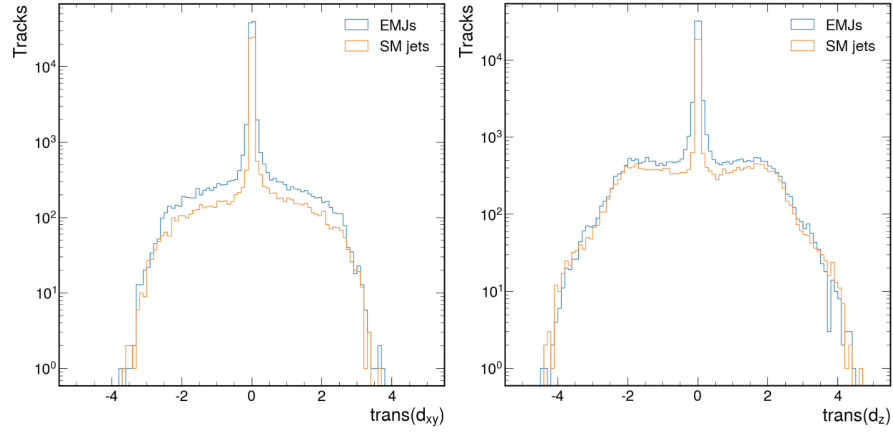


Figure A.2: Distribution of the transformed track impact parameters along the xy -plane and z -axis for emerging jets (blue) and SM jets (orange).

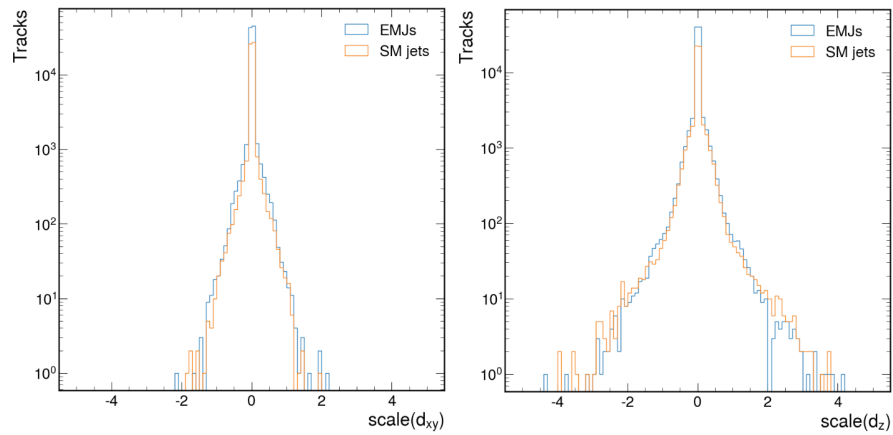


Figure A.3: Distribution of the scaled track impact parameters along the xy -plane and z -axis for emerging jets (blue) and SM jets (orange).

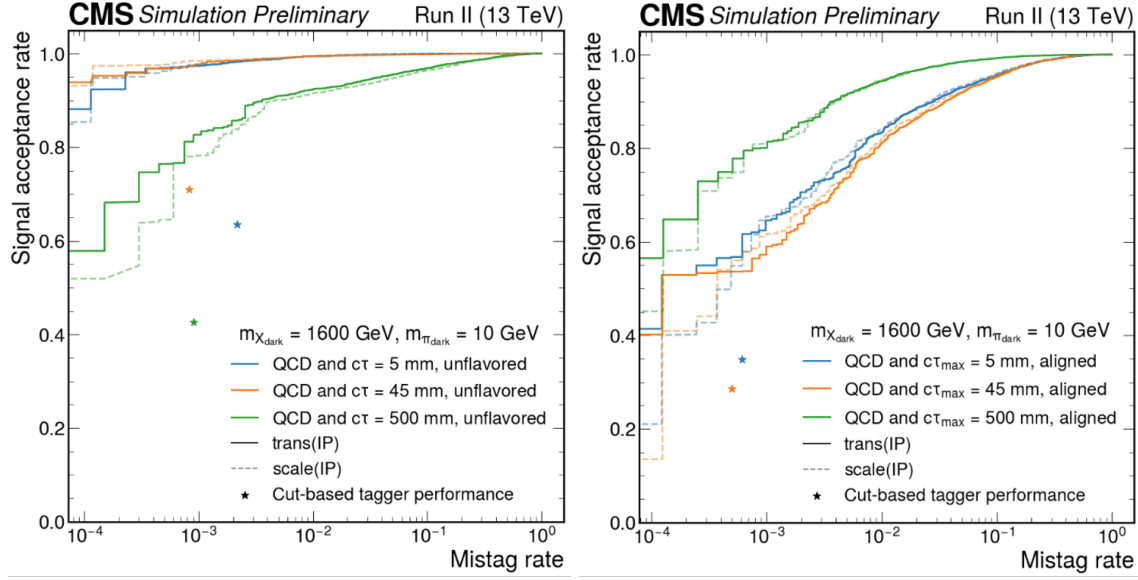


Figure A.4: Performance comparison of the GNN when trained on transformed impact parameters (solid lines) versus scaled impact parameters (dashed lines).

shifted by 1 and the sign is preserved outside of the logarithm. This transformation is given as:

$$\text{trans}(IP) = \text{sign}(IP) \ln(|IP| + 1), \quad (\text{A.1})$$

where IP is the impact parameter and $\text{sign}(IP)$ preserves the sign of the impact parameter.

Given the complexity of the transformation on the impact parameters above, a simpler scale was also tested to see if this had an effect on the performance of the model. This scale is defined as

$$\text{scale}(IP) = \left(\frac{IP}{20} \right). \quad (\text{A.2})$$

Figure A.1 shows what the impact parameters look like before any transformation, Figure A.2 shows the $\text{trans}(IP)$ distributions, and Figure A.3 shows the $\text{scale}(IP)$ distributions. It can be seen both variable alterations help constrain the impact parameter values to within $(-5, 5)$, but the distributions themselves do differ, as expected. When training the GNN with the scale, however, no significant change in performance is seen, shown by Figure A.4. Therefore, the decision was made to use the

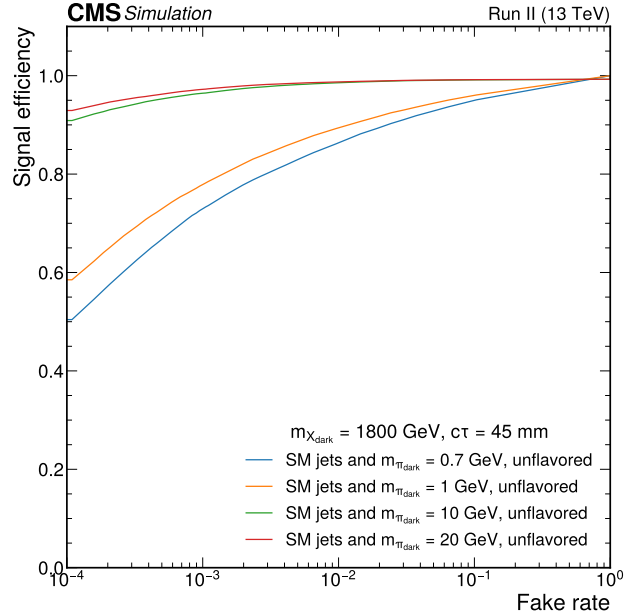


Figure A.5: Performance of the GNN tagger on different emerging jet unflavored samples. The GNN was trained on the $m_{\pi_{dark}} = 10$ and 20 GeV (green and red curves), but not 0.7 or 1 GeV (blue and orange curves), showing how the performance degrades when the tagger is applied to a sample outside what it is presented with at training.

logarithm transformation.

A.2 GNN Tagger Generalizability

In order to test the generalizability of the GNN tagger performance, some more simulated emerging jet samples with free parameters outside of the main samples described in Section 4.3 are tested on the already-trained taggers. Figure A.5 presents results for the unflavored GNN tagger, showing the large degradation in performance that occurs when testing the GNN on a sample that it was not trained on. This sensitivity to the specific samples trained on and not to more general physical processes should be kept in mind when developing machine learning models for similar physics tasks.

A.3 Cut Based Flavor-Aligned Results

The final results for the flavor-aligned cut based results can be found in Figure A.6. The shape of the exclusion limits matches what is seen for the GNN method, indicating that the same underlying physics processes drive the sensitivity of both tagging methods. In general, the exclusion limits are worse in the cut based flavor-aligned method by about 1000 GeV, in comparison to about 500 GeV for the unflavored method, indicating that the more complicated flavored signal gains more of a performance boost when using a tagging method that is sensitive to track-level relationships.

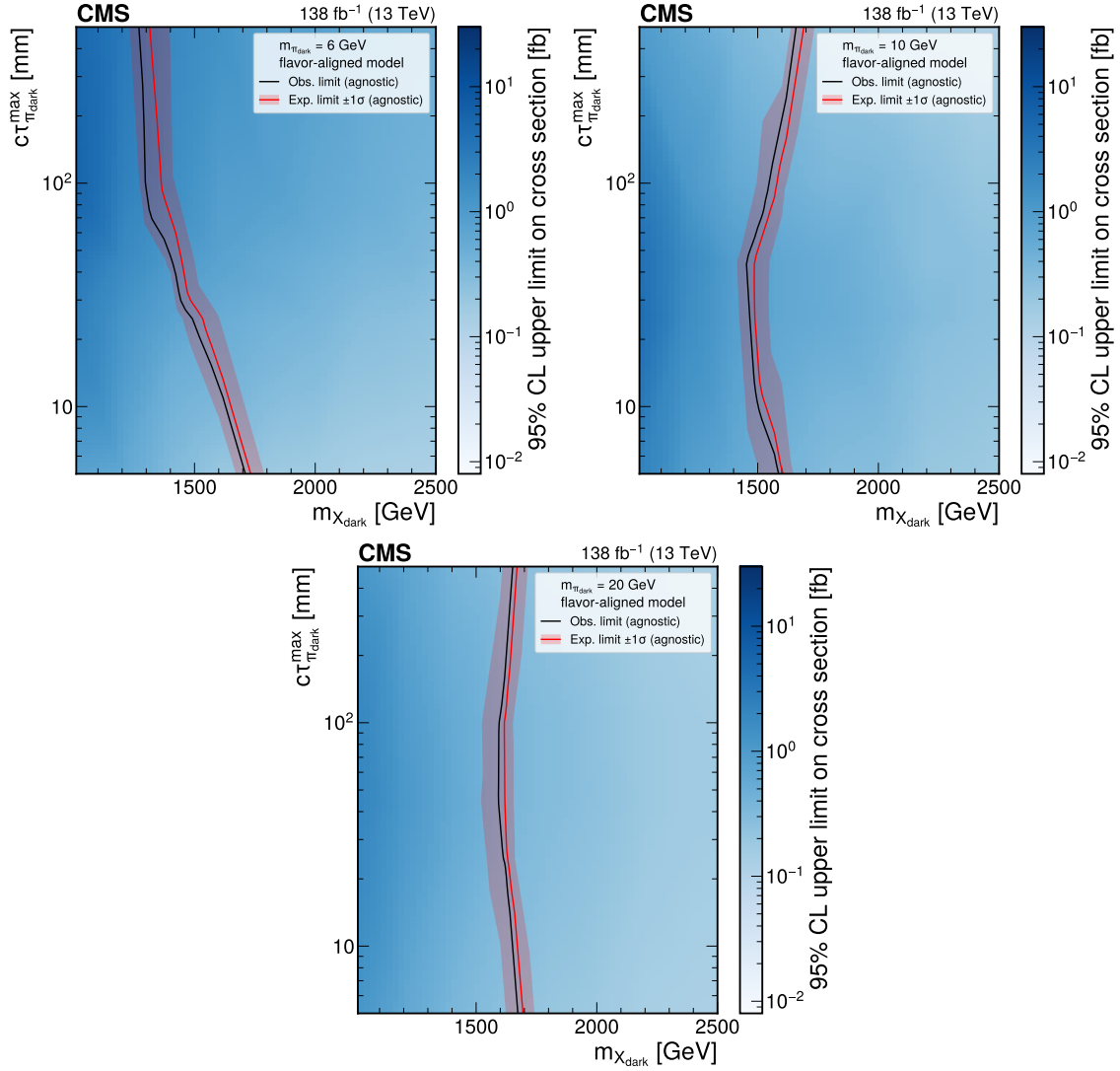


Figure A.6: Final exclusion limits for the flavor-aligned cut based method with $m_{\pi_{dark}} = 6$ GeV (top left), $m_{\pi_{dark}} = 10$ GeV (top right), and $m_{\pi_{dark}} = 20$ GeV (bottom) [55].

Appendix B

Triton Server Specifics

B.1 Triton server parameters

Several scaling parameters must be set to determine how the Triton server will create new instances, as discussed in Section 5.4.3.1. The parameters are carefully tuned to ensure that the instances are scaling out in a stable and efficient manner, as shown in Fig. 5.3. These parameters will be described below, along with a brief explanation of how we chose the parameters for the FNAL Triton server implementation. Each parameter is set uniformly for all models running on the server. These are not configurable on a per-user basis, as any change will affect all users and models using the same Triton server deployment.

Metric collection and analysis parameters

There are several time-related parameters for the inference server metrics.

- Metric collection interval: the default Triton server settings are used, such that statistics for model inference are collected every 15 s
- Analysis time step: sets the interval between analyzed data points. For this analysis, the time step is set to the same value as the collection interval, 15 s.
- Data collection window: determines the typical number of metrics used as input for the calculation of rates, deltas, and averages. By selecting an interval of 30 s, 2 consecutive measurements are used to compute an analysis data point. When used in conjunction with

Parameter	Value
scaleUp	
stabilizationWindowSeconds	60
selectPolicy	Max
policies.periodSeconds	180
policies.type	Pods
policies.value	1
scaleDown	
stabilizationWindowSeconds	600
selectPolicy	Max
policies.periodSeconds	60
policies.type	Pods
policies.value	1

Table B.1: Scaling behavior parameters of Triton HPA

a smaller analysis time step, the result is a sliding-window algorithm. This is well suited to averages and queue times.

Some metrics, such as the integrated number of requests, must be computed on unique values, and in such a case, the analysis time step and data collection window should be set to the same value to avoid double-counting. Inference metrics, such as inference request rate and queue time, are calculated and used to determine the performance of the server and whether more instances should be launched or shut down.

Horizontal Pod Autoscaling Parameters

The Triton server is configured as a Horizontal Pod Autoscaler (HPA) in Kubernetes [125]. It is configured to scale based on an external metric, referred to as the “queue time”, which is the maximum of the approximate queue time per inference, averaged per model. This metric gives a measure of the latency for a single request to be processed in the inference queue. Our implementation chose a threshold of 400 ms, which achieved a smooth scaling of MIG instances

while maintaining a reasonable throughput of approximately 5 inference requests per second per instance for the ParticleNet demo model. Note that the throughput of a model is model-dependent and the threshold may need to be adjusted to achieve reasonable throughput depending on the models being served.

The HPA scaling behavior parameters are summarized in Table B.1. Given the relatively small amount of MIG instances available (10), `policies.type` and `policies.value` were set to "Pods" and "1", respectively, to ensure that we would only start or stop a single server at a time.

The stabilization window for scale-up (`scaleUp.stabilizationWindowSeconds`) was chosen to be one minute, or four measurements (15 second interval) collected by the server. This was found to be a long enough time to determine whether the queue time continuously passes the threshold, but short enough to scale up quickly if the number of inference requests increases suddenly.

There is a delay before the queue time responds to a new inference instance being spawned, as seen in Figure B.1. For this reason, the `scaleUp.policies.periodSeconds` should be larger than the stabilization window in order to allow the queue time to decrease and stabilize. We chose 180 seconds, allowing the service two minutes for the queue time to stabilize and an additional minute to evaluate if an another instance should be spawned.

To avoid “flapping” — constantly starting and stopping instances as the queue time oscillates around the threshold — we choose a longer stabilization window for scaling down (`scaleDown.stabilizationWindowSeconds`): 600 seconds, or 40 consecutive measurements. This allows for long, uninterrupted processing time. Since HEP analyses tend to process millions of data events, the data is split into chunks, each generating batches of inference requests directed towards the Triton server. When processing begins for a chunk, a number of synchronous tasks unrelated to inference are performed which causes the number of inference requests to drop to zero until inference for that chunk finally begins, as shown in Fig. B.2. We want to ensure that the stabilization window is long enough to avoid scaling down during this downtime, trading off a small inefficiency for a decrease in overall latency.

We set `scaleDown.policies.periodSeconds` to 60, ensuring that the servers can scale down

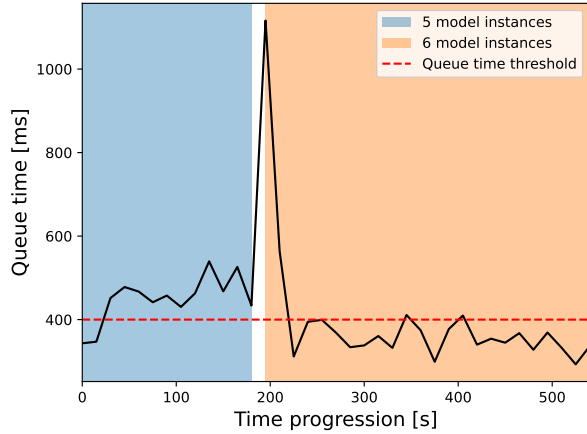


Figure B.1: When the Triton server starts up a new model instance, the queue time becomes noisy for a couple of collection intervals until stabilizing again [78].

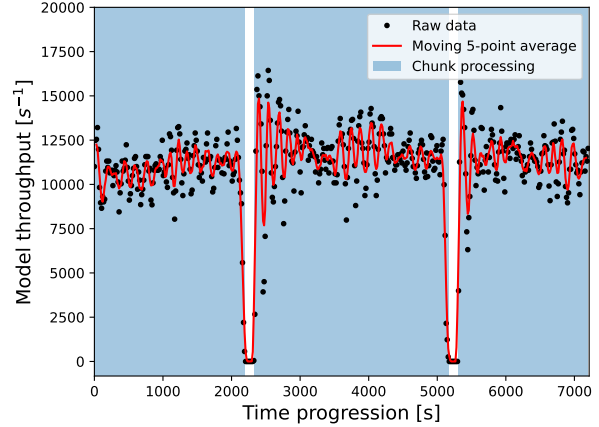


Figure B.2: Once a chunk of data is finished processing and while the next chunk is being pre-processed and undergoing non-inference computations, the server throughput drops [78].

rapidly once all inference requests have been processed, releasing the allocated resource so they can be used elsewhere. It is beneficial for the `scaleDown.policies.periodSeconds` to be smaller than the stabilization window as we want stable, long processing times but should be quick to free resources once processing has finished.

B.2 ParticleNet demo model parameters

The ParticleNet model used in this work to optimize the Triton server parameters and for benchmark testing is an exact replica of the model described in Ref. [104]. There are five input features, two coordinates, three EdgeConv Blocks using $k = 16$ nearest neighbors and $C = (64, 64, 64)$, $(128, 128, 128)$, and $(256, 256, 256)$ channels, and two fully connected layers with 256 nodes and 0.1 dropout rate to two nodes. A schematic of the exact structure can be found in Fig. 2a of Ref. [104].

This demo model was developed using the “Weaver” package [66] and was left untrained with randomized weights as the application and performance of the model is unrelated to the performance of the Triton server implementation which we are studying. Similarly, the input data are pseudo-randomized and arranged in the proper format required for inference. The structure of

the demo model is based on a ParticleNet model being used in an ongoing physics analysis at the LHC physics center at Fermilab [85].

The demo model was created using the PyTorch package [88] and converted using TorchScript to a version that can run on the Triton server. The Triton server reads in the converted file along with configuration files that tell the server how the model inputs and outputs are structured and how to partition the model on the server. In the configuration file, the following selections were made:

- *Dynamic batching*: The preferred batch size was set to 1024, determined by testing different batch sizes and balancing the inference speed with memory required to run.
- *Inputs and outputs*: There are 3 different inputs for the ParticleNet: the features, coordinates, and mask. Each of these three, along with the output, was assigned FP32 datatypes.
- *Inference mode*: This is set to **True**, letting the server know that the model is being used to run inference.
- *Instance group*: By default, a single model instance is created for each MIG spawned. This default is kept, as the demo model already requires 7–8 GB for inference and it is easier to test the Triton server implementation with a single model on each MIG. No specific GPU is targeted for the model as the GPUs available to the server are never fixed.

All other configurations are left to the default settings.