Standard Model of Particle Physics and Machine Learning

Oklahoma State Univeristy

Luke Martin Vaughan

Expected Completion: Fall 2025 or Spring 2026

Contents

1 Introduction **2**
1.1 The Standard Model of Particle Physics 2

A Pileup Studies **3**
A.1 Introduction 3
A.2 Related Work 5
A.3 Jet Corrections in High Pileup Conditions 6
A.4 Simulated Dataset 7
A.5 Improving JVT using Attention 8
A.5.1 Jet Features 8
A.5.2 Jet Labels 9
A.5.3 Classical JVT Architecture 10
A.5.4 AttnJVT Architecure 11
A.6 Hogwild Attention 11
A.6.1 Model Input 12
A.6.2 Encoders Architecture 12
A.6.3 Results 14
A.7 Analysis 15
A.8 Conclusion 16

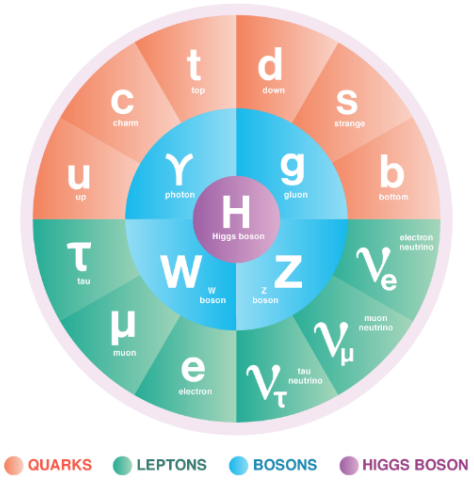
B More Appendix **18**
B.1 Appendix A 18
B.2 Appendix B 18

Chapter 1

Introduction

1.1 The Standard Model of Particle Physics

The standard model of particle physics explains the fundamental forces of nature.



Appendix A

Pileup Studies

A.1 Introduction

High Energy Particle physicists study proton-proton collisions at the Large Hadron Collider (LHC) [**?**] to better understand phenomena of the universe related to the fundamental forces of nature. These proton-proton collisions produce particles that are irreducible quanta of energy. Due to the subatomic size of protons, the mean number of interactions per bunch crossing in billions, $\langle\mu\rangle$, is very small. Only one of these collisions interacts via a strong head-on collision that produces deep inelastic scattering called *hard scatter* or *signal*, while other weaker interactions are called *pileup* or *background*. Each of these collisions, in turn, produces heavy particles which decay into lighter particles in a cascading effect. Streams of such decayed particles are called *jets*, which can be clustered to form a tightly knit cone containing charged particles, called *tracks*. All these collisions are recorded as *events*, usually at a rate of every 25 nanoseconds, containing a variable number of jets and a variable number of tracks in each jet. It is important to note that jets originating from hard scatter processes share underlying physics and thus can be correlated with other jets in the hard scatter process. Such a correlation is not feasible for pileup jets due to their stochastic nature, as given in Figure A.1.

Pileup interactions are considered as contamination because of their massive quantities compared to signal processes. It is important to identify and mitigate pileup from collision events to increase sensitivity of signal processes

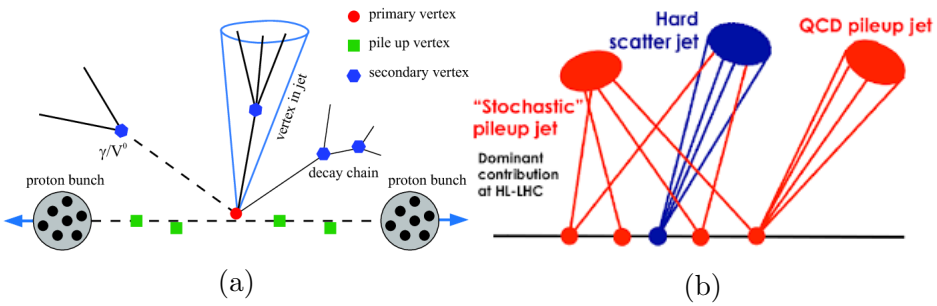


Figure A.1: As the proton bunches cross and interact, there typically exists one hard scatter interaction originating from the primary vertex while the other interactions originate from pileup vertices. The jets that originated from the HS vertex form a correlated set, while the jets from PU are stochastic in nature and do not have correlations. Remake this graphic and combine!

and assist physicists to discover new particles. Many data science and AI approaches have been introduced in recent years to mitigate pileup in variety of ways. All related work goes here.

Although there is growing interest in this domain, existing methods assume (i) only low pileup events, and (ii) pileup identification as a binary classification problem Find citations. As LHC prepares to upgrade to the High-Luminosity (HL)LHC [**?**], with $\langle\mu\rangle = 200$, it brings several billion collisions with exponentially higher number of pileup processes. Explain μ values used in existing works and why it will be a problem for $\mu = 200$. In low pileup conditions, the data is sparse such that jets can be easily binary classified as hard scatter or pileup jets. There could be a few pileup tracks entering a hard scatter jet, which can be corrected using Charged Hadron Subtraction [**?**], but the overall mass and energy of the hard scatter jet are seemingly unaffected. This becomes challenging with the upcoming upgrades to LHC which introduces several noisy pileup tracks into hard scatter jets and eventually altering core properties of hard scatter jets.

The existing algorithms developed for pileup mitigation using binary classification at low pileup conditions for the LHC will start to struggle as pileup is increased for the HL-LHC. Therefore at high pileup conditions, future algorithms must consider pileup as continuous regression problem to determine by what fraction the *hard scatter* jet's mass and energy has been affected by *pileup*. In this paper we propose a model that attemphts to solve the prob-

lem of pileup by directly predicting energy and mass fractions of each jet using transformer encoders using self-attention and cross-attention to learn enriched representations of jets using all possible correlations between jets and tracks within the context of an event.

In this work, we present the following contributions.

1. We propose a first-of-its-kind pileup prediction modeled as a regression problem.

2. We propose cross-attention based neural network architecture that utilizes jets and tracks information for pileup fraction detection.

3. We show with extensive analysis that the proposed method outperforms the baseline approaches.

4. We also show that the predictions from the proposed approach also assist with physics processes.

A.2 Related Work

Machine learning approaches for High Energy Physics problems are gaining traction from multiple perspectives with advancements in neural networks [**?**, **?**, **?**]. Although there are several sub-problems in High Energy Physics, such as jet flavor tagging [**?**], top tagging, generative event modeling [**?**], unfolding, anomaly detection, and calibration that have been explored with machine learning, methods to analyze *PileUp* have been mostly overlooked in the existing work. One of such interesting problems is mitigating pile-up particles and correcting jet mass and energy for events that occur at the Large-Hadron Collider [**?**].

Existing pileup mitigation techniques for the ATLAS and CMS experiments have focused on solving the binary classification problem for either jets or tracks. ATLAS uses algorithms to classify jets using classifiers such as kNN algorithms [**?**] which rely on constructing high level variables on a per jet basis. CMS uses an algorithm to classify tracks using a statistical χ^2 approach through the PUPPI [**?**] algorithm classifies pileup at the particle level. However, these methods fail to incorporate correlations between jets

at the Event level for Event-Wide context. Correlations exists between jet originating from Hard Scatter processes.

Graph Neural Networks have also been studied for for pileup mitigation, however, it is non-trivial how to form a graph in the context of an event. Its unfeasible to connect all particles in an event due to computation limitations, and connecting all tracks within a jet can confuse the model unless edge weights are assigned properly for HS and PU particles. Dynamic edge convolutions have been studied, but this can lead to long training times due to calculating kNN in latent space. Attention, on the other hand, gives a highly parallelizable algorithm to automatically determine weights between objects and update node representations accordingly.

A.3 Jet Corrections in High Pileup Conditions

As $\langle\mu\rangle$ increases from 60 to 200 for the HL-LHC upgrade, the pileup contamination of hard scatter jets will begin to dominate. As PU contamination degrades the quality of HS jets, the task shifts from a binary classification at low pileup conditions to a continuous regression at high pileup conditions. Instead of the traditional binary labels as HS or PU, we introduce a continuous labels for Energy and Mass Fraction, Efrac and Mfrac, which represents the fraction of the jets energy and mass originating from pileup.

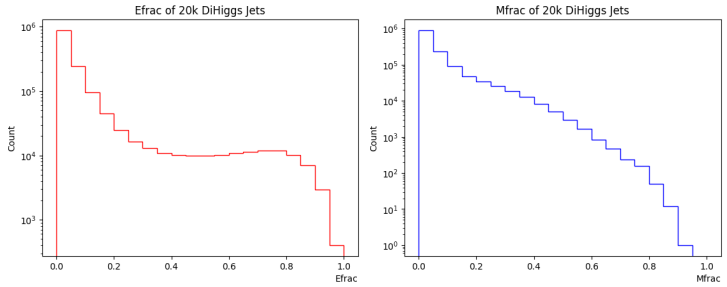
To construct these continuous labels, we sum over the Lorentz Four-Vector of the tracks, $\vec{T_i} = (E, p_x, p_y, p_z)_i = (E, \vec{p})_i = (T_0, T_1, T_2, T_3)_i$, which are truth-associated to each jet, $\vec{J}$:

$$\vec{J}_{HS} = \sum_{i \in \text{HS}} \vec{T_i} \qquad \vec{J}_{PU} = \sum_{i \in \text{PU}} \vec{T_i} \tag{A.1}$$

Now that we have the four vector HS and PU contributions of each jet, we can directly evaluate the Energy and Mass fractions on a per jet basis. The total Lorentz Four-Vector is simply the sum of the HS and PU contributions, $\vec{J}_{Total} = \vec{J}_{HS} + \vec{J}_{PU}$. Note that Energy is the first component of the jets Four-Vector, $J_0 = E$, and mass is found using the relativistic energy relations in natural units, $m^2 = E^2 - |\vec{p}|^2$. These expressions can be used to derive the

labels used for regression.

$$Efrac = \frac{E_{HS}}{E_{Total}} \qquad Mfrac = \frac{m_{HS}}{m_{Total}} \tag{A.2}$$



(a) (b)

Figure A.2: The truth energy and mass fractions used for the continuous regression task at $\langle\mu\rangle = 60$.

These continuous labels allow for corrections to be applied directly to the jet mass and energy. Of course a proper calibration will need to be applied when evaluated on data.

A.4 Simulated Dataset

A sample of 100k $pp \to HH$ events were generated using MADGRAPH_AMC@NLO [**?**] and showered through $H \to b\bar{b}$ channel using Pythia8 [**?**]. As each hard scatter process is showered in Pythia, pileup processes are overlaid using SoftQCD:inelastic generated with the A14 central tune with NNPDF2.3LO [**?**]. The average number of pileup processes are controlled by parameter $\langle\mu\rangle$ which follows a poisson distribution. Each pileup vertex undergoes gaussian smearing where the width in the x and y directions, representing the beam width, are 0.3mm and the spread in the z direction is 50mm. Stable final state particles are then passed to FastJet [**?**] to be cluster into jets using $anti-k_t$ algorithm [**?**] with R=0.4 and minimum p_T of 25 GeV.

To incorporate detector limitations, neutral particles and particles below 400 MeV were cut from the training dataset. After cuts, the dataset contains

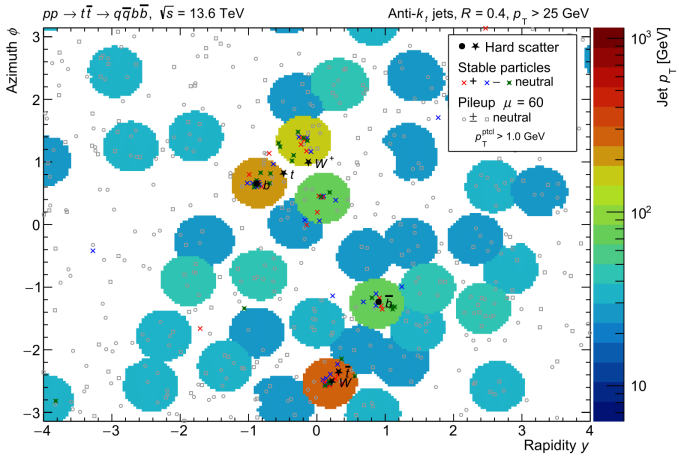


Figure A.3: An example of a simulated event of $pp \to t\bar{t}$ decaying hadronically at $\langle\mu\rangle = 60$.

500k jets and 10M particles. However, it is important to note that the jets studied in this paper are not calorimeter jets, but instead more idealistic track-jets. The results from this study offer the best-case scenario when tracks can be perfectly assigned the correct vertex, and the performance would be expected to degrade slightly in realistic detector conditions.

A.5 Improving JVT using Attention

The JVT model used by ATLAS is used to classify jets as HS or PU. However, at high pileup conditions JVT algorithm will need to be improved. Here we benchmark the JVT kNN algorithm [**?**] against AttnJVT.

A.5.1 Jet Features

Each jet has 6 features which includes the kinematic 4-vector of each jet, p_T, η, ϕ, m as well as track dependent features $corrJVF, Rp_T$ which were first proposed by ATLAS [**?**]. First, corrJVF can be understood as the fraction of the jet's momentum coming from hard scatter particles originating from the primary vertex with respect to the total momentum coming from all other

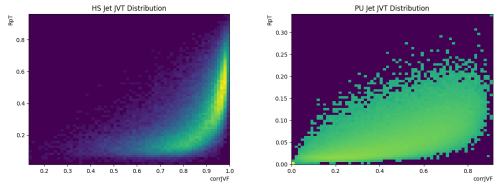
vertices. However, it has been shown that this variable has a dependence on $\langle\mu\rangle$, so to correct for this behavior the contribution from pileup vertices are normalized by the number of pileup particles in the event and a free parameter, k. This parameter has been shown to best remove dependence on $\langle\mu\rangle$ at k=0.01 [**?**].

$$corrJVF = \frac{\sum_k p_T^{trk_k}(PV_0)}{\sum_l p_T^{trk_l}(PV_0) + \frac{\sum_{n\geq 1}\sum_l p_T^{trk_l}(PV_n)}{(k \cdot n_{trk}^{PU})}} \tag{A.3}$$

Second, R_{pT} can be understood as the fraction of the jet's momentum originating from hard scatter primary vertex.

$$R_{pT} = \frac{\sum_k p_T^{trk_k}(PV_0)}{p_T^{jet}} \tag{A.4}$$

These features have unique distributions for HS and PU jets. HS jets tend to have higher corrJVF and RpT than pileup jets as shown in figure [*].



(a) (b)

Figure A.4: Distribution of jets in the R_{pT}-corrJVF plane for (a) HS and (b) PU jets.

A.5.2 Jet Labels

To fairly benchmark JVT and AttnJVT, a binary label was constructed by cutting on the squared sum of the constituent particle's p_T.

$$PU_{fr} = \frac{\sum_{PU} p_T^2}{\sum_{HS} p_T^2 + \sum_{PU} p_T^2} \tag{A.5}$$

An arbitrary cut is imposed on this distribution to convert the continuous distribution into a binary distribution. This cut allows for a consistent and fair benchmark against existing models. In this study, HS jets have $PU_{fr} < 0.7$ and PU jets have $PU_{fr} > 0.7$.

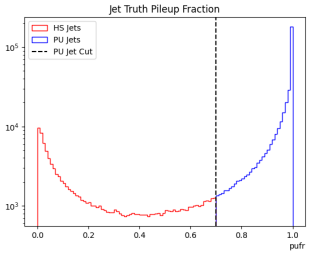
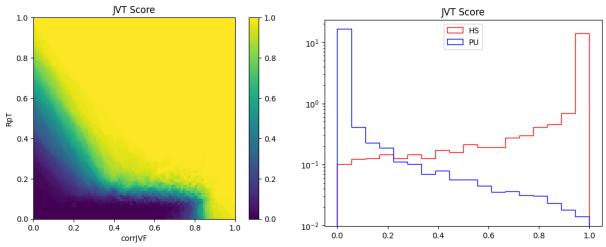


Figure A.5: An arbitrary cut on the continuous PU_{fr} to recover binary labels.

A.5.3 Classical JVT Architecture

The Jet Vertex Tagger, JVT, was orignally proposed by ATLAS [**?**]. The model uses R_{pT} and corrJVF as input and constructs a likelihood between zero and one which represents the probability in which the jet originated from hard scatter. The JVT model is based on a k-Nearest Neighbor, kNN, algorithm which uses a Euclidean metric in the R_{pT}-corrJVF plane and is fit using k=100.



(a) (b)

Figure A.6: Figure (a) shows JVT liklihood in the R_{pT}-corrJVF plane. Figure (b) shows the JVT liklihood for HS and PU jets.

A.5.4 AttnJVT Architecure

To perform a fair benchmark, the orignal JVT model was recreated and performance was evaluated between the original model and the attention model on the same dataset. The input to the model is a unordered set of jets from an event. Each event has a variable number of jets, N, so the input tensor will be of shape $\mathbb{J} \in \mathbb{R}^{N,F}$ where F is the six input features described in section 2.1. The input jets are fed through an initializer, ϕ, to transform them into the embedding space with dimension D:

$$\phi(\mathbb{J}) \rightarrow \mathbb{E} \in \mathbb{R}^{N,D} \tag{A.6}$$

The embedded jets are then passed through a multi-head attention layer to update their representations in the context of an entire event. First the embeddings $\mathbb{E}$ are passed through a multi-head attention layer using self attention to generate the context tensor $\mathbb{C}$:

$$\mathbb{C} = MHA(\mathbb{E}, \mathbb{E}, \mathbb{E}) \tag{A.7}$$

Then the context tensor, $\mathbb{C} \in \mathbb{R}^{N,D}$, is concatenated with the original embedding to update the original representation of each jet, $\mathbb{E} = \mathbb{E} + \mathbb{C}$. Lastly, the jet embedding is passed through a final classifier, Ψ, to perform binary classification:

$$\Psi(\mathbb{E}) \rightarrow \vec{y} \in \{0, 1\} \tag{A.8}$$

The model is trained using the Binary Cross Entropy loss function.

A.6 Hogwild Attention

This section describes how to predict the Efrac and Mfrac of a jet using a stack of Transformer Encoders using Multi-Head Attention layers in the context of an event using track features and jet features as input.

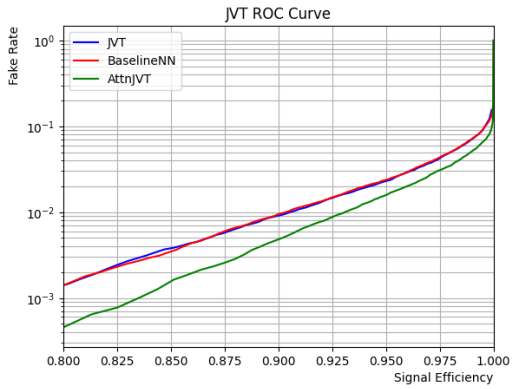


Figure A.7: Using a MHA layer, the jets have access to the context of the entire event which allows the model to capture correlations between HS jets. This effect reduces the fake rate of the model.

A.6.1 Model Input

The model takes an entire event as an input. An event can be described as three tensors: jet tensor, jet-track tensor, and all-track tensor. Jet tensor $\mathbb{J} \in \mathbb{R}^{N_{jets} \times F_{jets}}$ has shape N jets per event with F input features, p_T, η, ϕ, m. Jet-Track tensor $\mathbb{T}_{jet} \in \mathbb{R}^{N_{jets} \times N_{trks} \times F_{trks}}$ has shape number of jets per event, number of tracks per jet, and number of features per track, $p_T, \eta, \phi, q, d_0, z_0$. Lastly, All-Track tensor $\mathbb{T}_{Event} \in \mathbb{R}^{\times N_{trks} \times F_{trks}}$ has shape number of tracks per event and features per track. Each of these tensors if first passed through a linear preprocessing layer to transform them into the embedding space.

A.6.2 Encoders Architecture

Four transformer encoder stacks are used to enrich each tensor – $\mathbb{J}, \mathbb{T}_{jet}, \mathbb{T}_{jet}$ – with context from the event. Each encoder follows the NormFormer [*] architecture by (... et al). NormFormers conist of LayerNorms, LN(), multi-head attention, MHA(), skip connections, + operator, feed-forward network, FFN, which consists of a linear layer and a GELU activation function. These layers are the components of each encoder block.

First, each jet is enriched with associated tracks. Since each jet only carries four features, p_T, η, ϕ, m, tracks within a radius of $\Delta R = 0.4$ are used

in the first encoder stack to achieve a rich representation of each jet.

$$\mathbb{T}_{context} = \mathbb{T}_{jet} + LN(MHA(LN(\mathbb{T}_{jet}), LN(\mathbb{T}_{jet}), LN(\mathbb{T}_{jet}))) \tag{A.9}$$

$$\mathbb{T}_{embed} = \mathbb{T}_{context} + FFN(\mathbb{T}_{context}) \tag{A.10}$$

$$\mathbb{T}_{aggregated} = \sum_{dim=1} \mathbb{T}_{embed} \tag{A.11}$$

$$\mathbb{J}_{enriched} = FFN(\mathbb{J} \underset{dim=1}{\oplus} \mathbb{T}_{aggregated}) \tag{A.12}$$

$\mathbb{T}_{jet}$, $\mathbb{T}_{context}$, and $\mathbb{T}_{embed}$ all have shape $N_{jet} \times N_{trk} \times E_{dim}$. Then the summation operator reduces the N_{trk} dimension which will then match the dimension of $\mathbb{J}$ with shape $N_{jet} \times E_{dim}$. The jet and aggregated track tensors are concatenated along the embedding dimension, and the FFN has input $N_{jet} \times 2 \cdot E_{dim}$ and outputs $\mathbb{J}$ of shape $N_{jet} \times E_{dim}$. This encoder block can be interpreted physically as learning to enrich the jet in the context of associated particles. For example, if there is are particles that resemble b-hadron decay, this encoder block might enrich this jet as a b-jet in the latent space.

Second, $\mathbb{T}_{event}$ with shape $N_{trk} \times E_{dim}$ are enriched using an encoder block using self-attention:

$$\mathbb{T}_{context} = \mathbb{T}_{event} + LN(MHA(LN(\mathbb{T}_{event}), LN(\mathbb{T}_{event}), LN(\mathbb{T}_{event}))) \tag{A.13}$$

$$\mathbb{T}_{embed} = \mathbb{T}_{context} + FFN(\mathbb{T}_{context}) \tag{A.14}$$

$$\tag{A.15}$$

The purpose of this encoder is to update all tracks in the context of the event and initialize them for cross attention with jets.

Third, cross attention between $\mathbb{J}_{enriched}$ and $\mathbb{T}_{embed}$ is performed to update the jets in the context of all tracks of the event.

$$\mathbb{J}_{context} = \mathbb{J}_{enriched} + LN(MHA(LN(\mathbb{J}_{enriched}), LN(\mathbb{T}_{event}), LN(\mathbb{T}_{event}))) \tag{A.16}$$

$$\mathbb{J}_{embed} = \mathbb{J}_{context} + FFN(\mathbb{J}_{context}) \tag{A.17}$$

$$\tag{A.18}$$

This encoder allows tracks to update the jet embedding in the context of an event. For example, if we consier $t \to Wb \to l\nu b$, this encoder allows the high energy lepton, l, to update the context of the b-jet.

Fourth and finally, $\mathbb{J}_{embed}$ with shape $N_{jet} \times E_{dim}$ is enriched using a encoder block using self attention:

$$\mathbb{J}_{context} = \mathbb{J}_{embed} + LN(MHA(LN(\mathbb{J}_{embed}), LN(\mathbb{J}_{embed}), LN(\mathbb{J}_{embed}))) \tag{A.19}$$

$$\mathbb{J}_{embed} = \mathbb{J}_{context} + FFN(\mathbb{J}_{context}) \tag{A.20}$$

$$\tag{A.21}$$

This encoder is performed last because at this stage the jets have achieved a rich representation after passing through the previous encoders. This last encoder allows jets to update their representation in the context of an event. This encoder can be interpreted physically as allowing jets to update representations according to conservation of momentum or other properties that might be shared between jets.

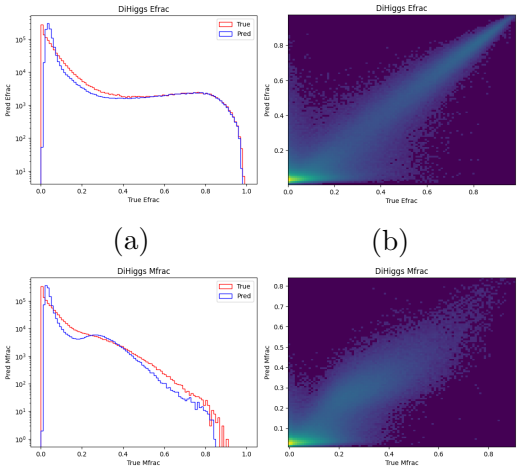
Lastly, the embedded jet vectors are passed through a final classification layer to predict the continuous pileup fraction label.

A.6.3 Results

The model was constructed using a stack of 3x jet associated-track encoders, 3x all track encoders, 3x jet all track encoders, and finally 3x jet encoders. In total the model has 4.1M parameters and took about 1 hour to converge on an NVIDIA 3090 using a sample size of 10k $t\bar{t}$ events with around 500k jets and 10M charged tracks greater than 400 MeV.

Metric	Value
Train Loss	0.0016
Val Loss	0.0018
Test Loss	0.0022
Test MAE	0.017
Test RMSE	0.046

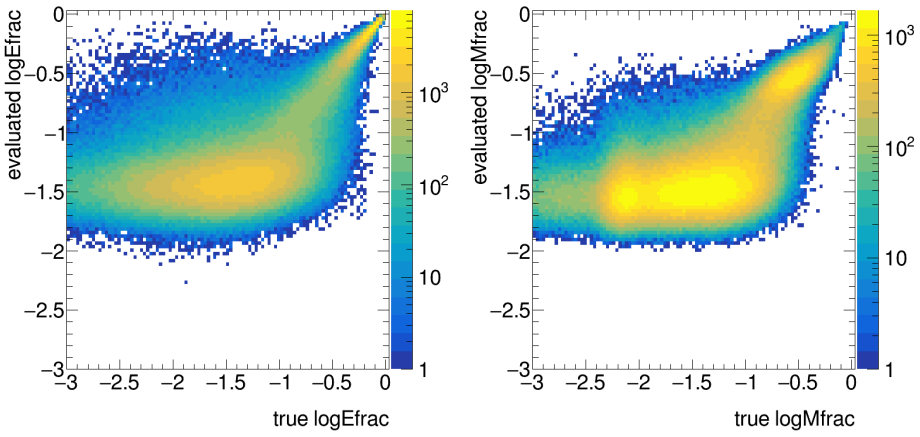
The model is trained using mean squared error loss. After converging, the train loss reaches around 0.00164, validation loss of about 0.0018, and train loss of 0.0022. The mean absolute error, MAE, of about 0.017, and the root mean squared error, RMSE, of about 0.046.



(a) (b)

(c) (d)

Figure A.8: The predicted Mass fraction of jets. Figure (a) 1-D Histogram. Figure (b) 2-D Historgram.



(a) (b)

Figure A.9: The predicted fractions vs the true fraction on a log scale. Figure (a) Efrac. Figure (b) Mfrac.

A.7 Analysis

In order to determine if the model is able to provide useful insight into a practical physics analysis, we attempt to reconstruct the Higgs mass from DiHiggs decaying to 4b events with a non-resonant 4b background. The non-resonant 4b sample has a 60GeV pTB filter applied at generation level to ensure the kinematics of each sample are indistinguishable. Therefore, the only difference between these samples is the the resonant mass peak that appears in the DiHiggs sample and the flat background from the non-resonant 4b sample.

Using the results of the Efrac and Mfrac model, we are able to apply jet corrections and see

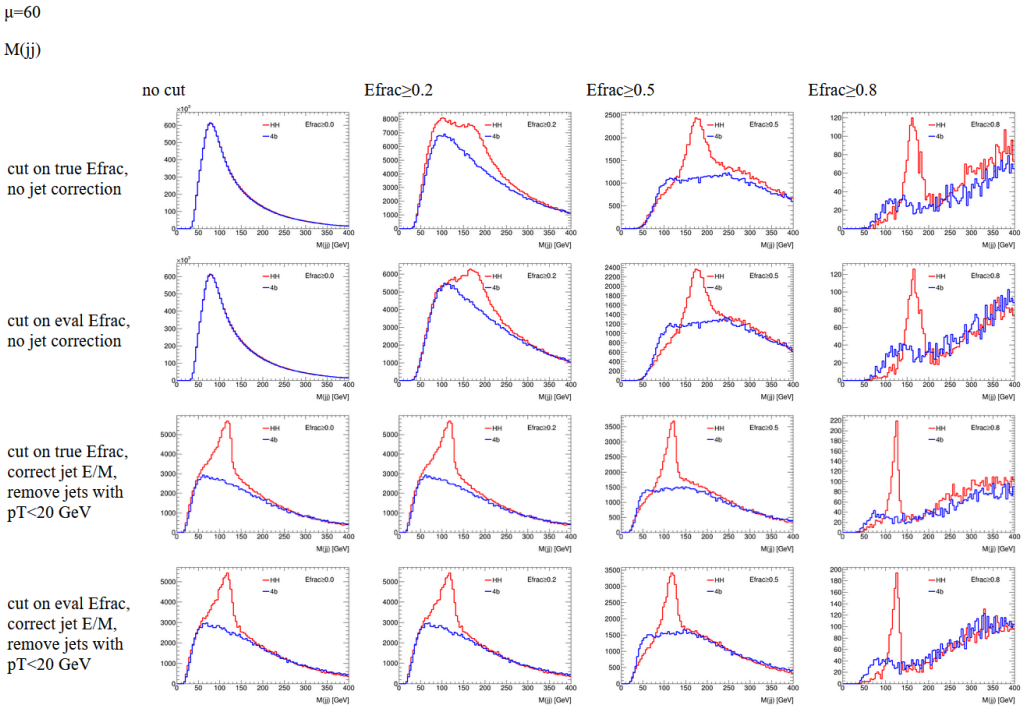


Figure A.10: Physics Analysis Results

A.8 Conclusion

In conclusion, we presented the following contributions.

1. We proposed a first-of-its-kind pileup prediction modeled as a regression problem.

2. We proposed a cross-attention based neural network architecture that utilizes jets and tracks information for pileup fraction detection.

3. We showed with extensive analysis that the proposed method outperforms the baseline approaches.

4. We also showed that the predictions from the proposed approach also assist with physics processes.

Appendix B

More Appendix

B.1 Appendix A

Here is A.

B.2 Appendix B

Here is B.