
# Development of Neural Networks to Tag Emerging Jets in the ATLAS Experiment

Sophie Gélinas
Supervised by Matthias Danninger & Jackson Burzynski
Report submitted to the CERN Summer Student Programme
August 22, 2025

Performed at:
Simon Fraser University, Burnaby, Canada
and
CERN, Geneva, Switzerland

**Abstract**

A class of dark matter models explored by the ATLAS experiment features a dark sector containing dark quarks. Like Standard Model quarks, these dark quarks can shower and hadronize into dark mesons, which may travel significant distances before decaying into Standard Model particles. This produces topologically distinct emerging jets, which are well-suited to identification by neural networks. To increase the sensitivity of future analyses, improvements have been made to an existing supervised jet tagger. A new unsupervised algorithm, aimed at enabling model-independent searches, has also been developed.

# Contents



# 1 Introduction

Dark matter is currently one of the major unsolved problems in physics. Astronomical observations have provided convincing evidence for its existence [1], but its exact nature remains a mystery. Various theoretical frameworks for dark matter have been proposed, including a class of models known as hidden valley models [2]. Such models postulate the existence of a dark sector containing dark particles that interact with the Standard Model (SM) through a new mediator particle. Two potential mediator particles relevant to this report are a vector boson $Z'$ (corresponding to an $s$-channel process) and a scalar mediator $\Phi$ (corresponding to a $t$-channel process). When a proton-proton collision occurs, these mediator particles may be produced by quark-antiquark annihilation, and subsequently decay into dark quarks. Feynman diagrams for these processes are shown in Fig. 1. Similarly to SM quarks, dark quarks then undergo showering and hadronization, forming dark mesons. In some models, dark mesons are unstable and have lifetimes that depend on the strength of their couplings to the SM. When these couplings are relatively weak, dark mesons have longer lifetimes, and can travel macroscopic distances before decaying back to SM particles. Dark particles are not detected, so only these SM particles are observed. As seen in Fig. 2, this results in the production of topologically

distinct emerging jets, characterized by SM particles appearing at displaced vertices within the jet cone [3].

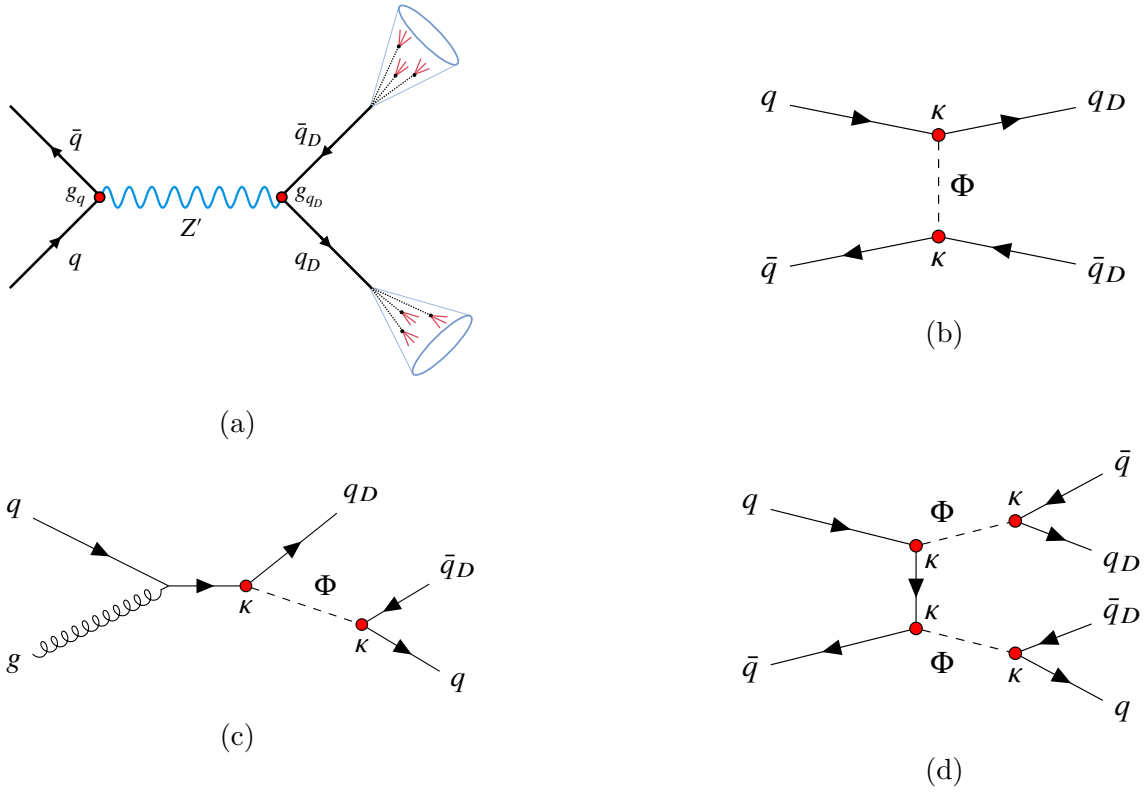



Figure 1: Feynman diagrams for dark quark production. (a) depicts production through an $s$-channel mediator $Z'$ and subsequent pair production of emerging jets (drawn as cones, with dashed lines for dark mesons and solid lines for SM particles). The coupling constants $g_q$ and $g_{q_D}$ govern the strengths of the interactions between $Z'$ and the SM quarks, and between $Z'$ and the dark quarks, respectively. (b), (c), and (d) depict production through a $t$-channel mediator $\Phi$, with interactions governed by a coupling matrix $\kappa$. Images from Ref. [4].

The ATLAS collaboration has searched for evidence of emerging jets in a previous analysis [4]. One strategy used to identify these jets was a specialized machine learning algorithm, which can discriminate between the distinct structures of emerging and SM jets. This approach allowed for a higher sensitivity than the complementary cut-based approach. This report describes improvements made to the existing supervised jet tagging algorithm, which will be used in future analyses, as well as the development of a new unsupervised algorithm. A supervised algorithm is trained on a mix of signal and background jets, and these must be labelled as such; the analysis must therefore make assumptions about a signal model and its parameters. In contrast, an unsupervised algorithm is trained on unlabelled data (in this case, background jets only). This could allow for a model-independent search, in which the algorithm identifies "anomalous" jets that differ in some way from the Standard Model background, but does not make any assumptions about what the signal may be. In comparison to a search with the supervised algorithm, this search would likely have reduced sensitivity to models the supervised

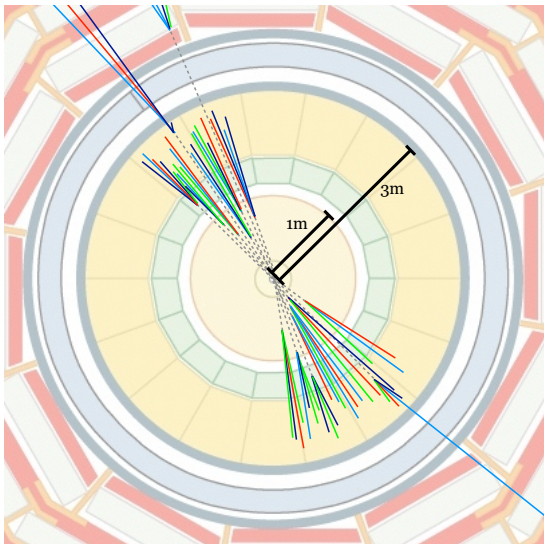



Figure 2: Schematic of emerging jet production in a detector cross section. Dashed lines represent dark mesons, which are not detected. These eventually decay to SM particles, represented by colored lines, which are observed at displaced vertices. Image from Ref. [3].

algorithm is trained on, but increased sensitivity to other models, making it a more general search with potential sensitivity to a far larger parameter space.

# 2 Supervised Jet Tagger

## 2.1 Model Architecture

The supervised jet tagger is a transformer-based neural network derived from the ATLAS flavour tagging algorithm [5]. The model architecture is depicted in Fig. 3. The input data, listed in Table 1, consists of jet pseudorapidity and the features of up to 200 tracks associated with the jet. In general, a neural network processes input data through several layers of nodes. Each layer applies mathematical operations (mainly matrix operations with associated parameters, known as weights and biases) to transform the input data. Training the neural network involves optimizing the matrix parameters to minimize a loss function, which measures how far the model's predictions are from the true values. The optimization is performed using an algorithm called gradient descent, which iteratively updates the parameters in the direction that reduces the loss.

In the supervised jet tagger, the input vectors are first processed by a track initialization network, which is a neural network made up of several layers that produces a higher-dimension representation of each track (which is meant to better capture the important characteristics of the track). The track representations are then encoded by a transformer, which outputs conditional track representations. These are track representations that consider the relationships and correlations between the different tracks in the jet. The conditional track representations are then combined using attention pooling, which weighs and aggregates the tracks (placing higher value on features it finds more relevant) to produce a single representation of the jet as a whole. This global jet representation is used to perform three classification tasks. The classification tasks are also neural networks which output classification scores corresponding to

the model's prediction for each task.

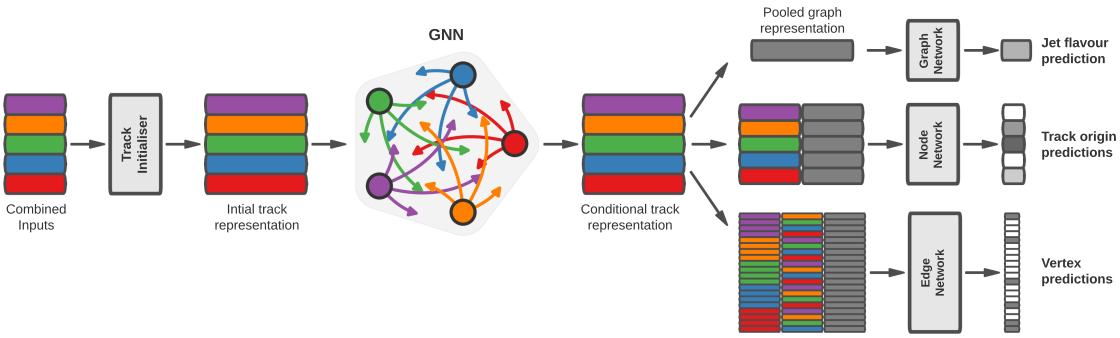



Figure 3: Schematic of the supervised tagger's architecture. Image from Ref. [5] (this is the ATLAS flavour tagging algorithm; "Jet flavour prediction" should be "Jet type prediction.")

The primary task is jet type classification, or to predict whether a jet is a signal (emerging/displaced) jet or a background (prompt) jet. The other two tasks are auxiliary tasks whose output is not directly used in the primary task, but which improve the model's performance at jet type classification. The first auxiliary task is track origin classification (predicting which type of particle produced each track in the jet), and the second is vertex prediction (grouping tracks within the jet). The loss function thus contains one term for each classification task:

$$L_{\text{total}} = L_{\text{jet}} + \alpha L_{\text{track}} + \beta L_{\text{vertex}} \tag{1}$$

The coefficients $\alpha$ and $\beta$ make each term in the loss roughly equal in magnitude to avoid optimizing one task over the others. The terms $L_{\text{jet}}$, $L_{\text{track}}$ and $L_{\text{vertex}}$ are cross-entropy losses for the jet classification, track origin prediction, and vertex prediction tasks respectively. Cross-entropy loss, commonly used for classification tasks, quantifies how well the model's predictions (probabilities between 0 and 1 for each class) match the true values.

## 2.2 Parameter Updates

To try to improve the supervised model, modifications to several training parameters were tested. These updates were chosen based on those made to the ATLAS flavour tagging algorithm. A separate model was trained for each modification in order to observe their effects independently. A first test was to increase the size of the track embedding (the dimensionality of the track representations after tracks are processed by the initialization network) from 256 to 512. Another was to switch the activation functions from ReLU (Rectified Linear Unit) to SiLU (Sigmoid Linear Unit). Activation functions are mathematical operations that are applied between the layers of a neural network to transform the output of each node in the layer. They introduce non-linearity to the network, allowing it to model more complex relationships and patterns in the data. ReLU takes the non-negative part of its input (i.e., $\text{ReLU}(x) = \max(0, x)$) and is commonly used in neural networks, in part because it is computationally efficient. SiLU, defined as $\text{SiLU}(x) = x \cdot \text{sigmoid}(x)$, is a smoother alternative to ReLU.

Changes to the learning rate scheduler were also tested. The learning rate scheduler modifies the model's learning rate (how much the weights are adjusted after each training step) during

| Input | Description |
|---|---|
| Jet $\eta$ | Jet pseudorapidity |
| $d_0$ | Track closest distance to PV in transverse plane |
| $z_0 \sin(\theta)$ | Track closest distance to PV in longitudinal plane |
| $\Delta\phi$ | Azimuthal angle of the track, relative to the jet $\phi$ |
| $\Delta\eta$ | Track pseudorapidity, relative to jet $\eta$ |
| $q/p$ | Track charge over momentum |
| $\sigma(\phi)$ | Uncertainty in track $\phi$ |
| $\sigma(\theta)$ | Uncertainty in track $\theta$ |
| $\sigma(q/p)$ | Uncertainty in track $q/p$ |
| $d_0/\sigma(d_0)$ | signed $d_0$ significance |
| $z_0/\sigma(z_0)$ | signed $z_0$ significance |
| $N_{\text{PIX hits}}$ | Number of Pixel hits per track |
| $N_{\text{SCT hits}}$ | Number of SCT hits per track |
| $N_{\text{IBL hits}}$ | Number of innermost pixel layer hits |
| $N_{\text{PIX shared}}$ | Number of Pixel shared hits |
| $N_{\text{SCT shared}}$ | Number of SCT shared hits |

Table 1: List of variables used to train the supervised tagger. The only jet-level variable is jet $\eta$ (jet $p_T$ is not used as it was previously found to increase momentum dependence without improving tagger performance); the rest are track-level features. PV is the primary vertex of the track (from the main proton-proton interaction), and SCT refers to the semiconductor tracker. Pseudorapidity ($\eta$) measures the angle of the particle with respect to the beam axis (which is the $z$-axis of the coordinate system), and can be related to the polar angle $\theta$ by $\eta = -\ln[\tan(\theta/2)]$. $\phi$ is the azimuthal angle around the beam axis. $q$ and $p$ are the particle's charge and momentum respectively. $d_0$ is the track's distance of closest approach to the PV in the plane transverse to the beam axis, and $z_0 \sin(\theta)$ is the distance of closest approach in the longitudinal plane ($z_0$ is the value of $z$ at the point that determines $d_0$).

training. The baseline model used in the previous analysis had an initial learning rate of $1 \times 10^{-7}$, a maximum rate of $5 \times 10^{-4}$, a final rate of $1 \times 10^{-5}$, and a `pct_start` (fraction of training steps to reach the maximum value) of 0.01. In one test, the initial, maximum, and final rates were decreased to $2 \times 10^{-8}$, $1 \times 10^{-4}$, and $2 \times 10^{-6}$ respectively. In a separate test, `pct_start` was increased to 0.1.

The final modification was to the loss calculation for the jet classification task. It was previously computed using the `CrossEntropyLoss` function provided by PyTorch. This function can compute cross entropy loss for an arbitrary number of classes, and it was set up with two classes, outputting two classification scores: one representing the probability a jet was an emerging jet, and the other that it was a prompt jet. However, jet type classification is in fact a binary task, and can be performed using a single score ranging between 0 and 1 (with 1 corresponding to an emerging jet, and 0 to a prompt jet). The model was therefore modified to output a single classification score, and to instead use Pytorch's `BCEWithLogitsLoss` function. This function computes a binary cross entropy loss (suited to binary classification tasks), and "with logits" means that it also applies a sigmoid function which normalizes the model's output

to be between 0 and 1.

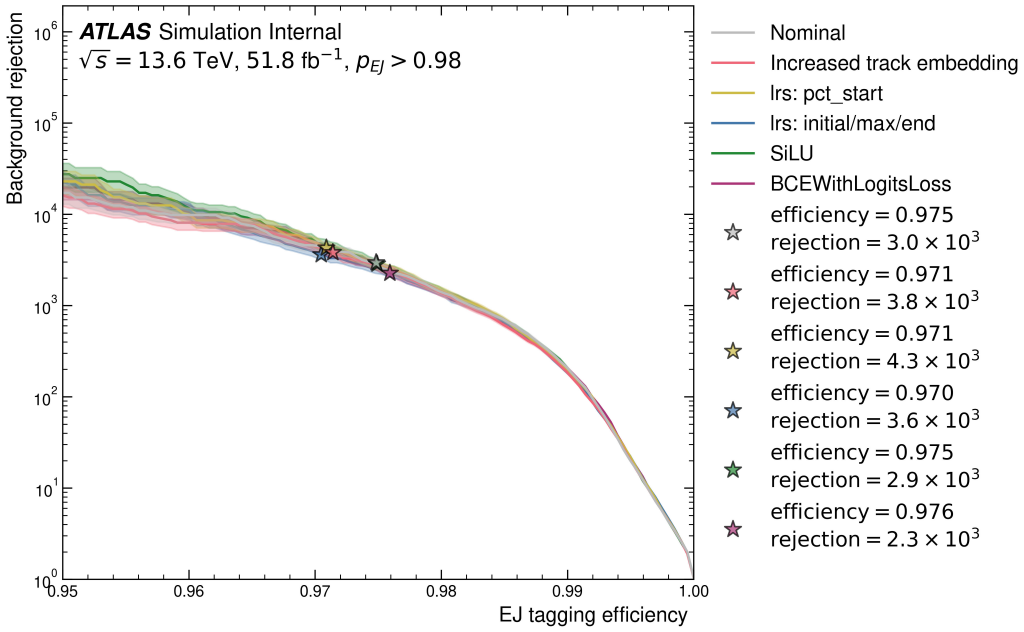



Figure 4: ROC curves for models with different modifications to training parameters. The "Nominal" curve is the model used in the previous analysis (without any changes). Signal efficiency and background rejection for a cut at $p_{\text{EJ}} > 0.98$ are shown. There is no significant difference in performance between models.

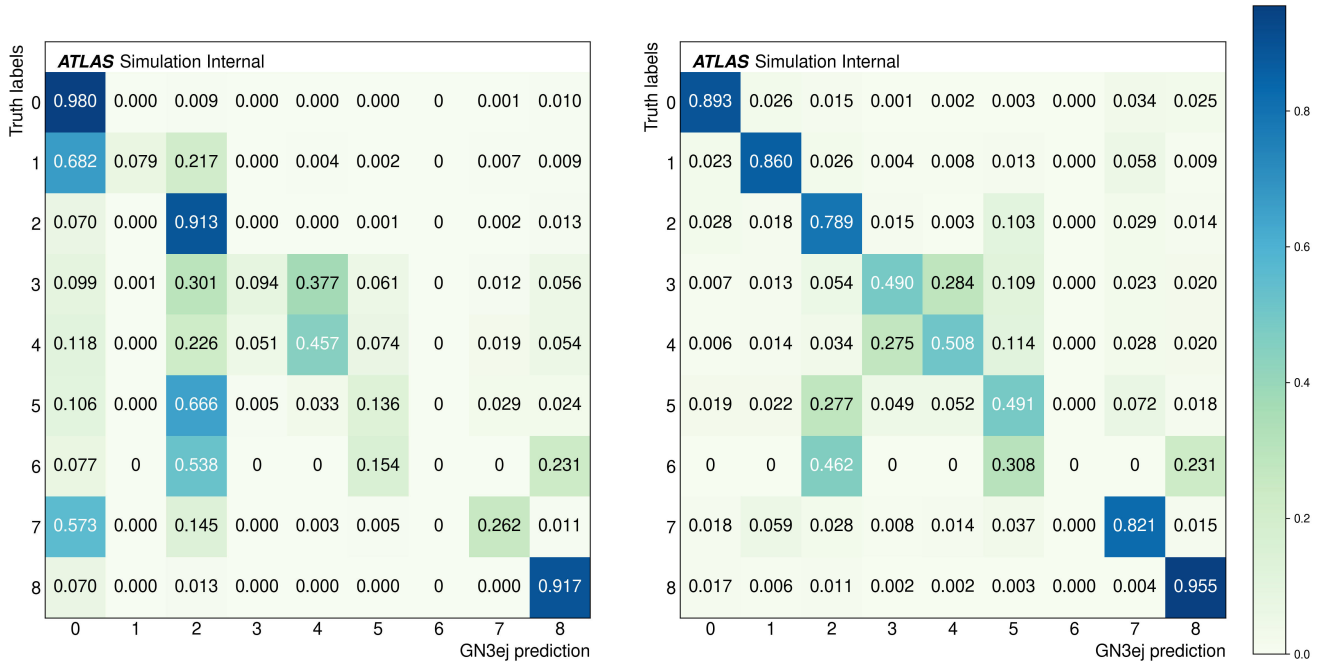
The algorithm's performance with all these changes is plotted in Fig. 4, which is a ROC (Receiver Operating Characteristic) plot of background rejection as a function of signal efficiency. Each point on the curve corresponds to a different choice of jet classification score $p_{EJ}$, which is used to select which jets are predicted to be signal ($p_{EJ}$ above a certain value) and which are background ($p_{EJ}$ below that value). Ideally, both signal efficiency and background rejection should be maximized, but there is in fact a trade-off between these two quantities: a lower $p_{EJ}$ will result in a higher signal efficiency but a lower background rejection, and vice-versa. Cuts at a $p_{EJ}$ of 0.98 (jets with $p_{EJ}$ greater than 0.98 tagged as signal) are shown, as this value was used in the previous analysis. As seen in Fig. 4, the curves are all essentially the same; no significant increase (or decrease) in performance was observed. The time to train the model also remained approximately the same, except for the case of the increased track embedding, which was significantly slower (the time to train one epoch increased from 2 h 47 min to 4 h 40 min). Given that none of these updates improved the performance, the model was reverted back to the baseline from the previous analysis, with one exception: the switch to `BCEWithLogitsLoss`. This change was implemented as it is the loss function and number of output scores that should generally be used for a binary classification task.

## 2.3 Extended Track Labelling Scheme

The next update to the model was to the auxiliary track origin classification task. The original model classified tracks into four categories: pile-up, fake, prompt, and displaced. Pile-up tracks come from other proton-proton collisions that occur at the same time (within the same bunch

crossing) as the hard-scatter collision of interest; these are typically less interesting, lower-energy interactions. As their name suggests, fake tracks are false tracks created by the track reconstruction algorithms that do not originate from an actual particle. Prompt tracks come from the collision of interest, but not the decay of a dark meson, whereas displaced tracks do come from the decay of a dark meson (and are also produced by the collision of interest).

To try to improve the model's performance at this task, the prompt track category was split into six more specific particle types, for a total of nine categories. The new classifications are primary, from B, from BC, from C, from tau, and other secondary, where B and C refer to $b$-hadrons and $c$-hadrons respectively. The prompt track category previously included some tracks that are in fact displaced from the primary interaction point (but originate from Standard Model particles, not dark mesons), which were easy to misclassify as displaced. The goal of this new track labelling scheme was to help the model correctly identify these tracks, and thus better discriminate between prompt and displaced jets. Once the extended labelling scheme was implemented, the model was tested with and without weights, which penalize the model more strongly for incorrectly classifying tracks which are more rare. The goal of the weights is to prevent the model from classifying all tracks as the most common origin, which might otherwise minimize the loss function.




(a) (b)

Figure 5: Confusion matrices for the track origin classification task. (a) shows the performance without weights, and (b) shows the performance with weights. In order from 0 to 8, indices correspond to pile-up, fake, primary, from B, from BC, from C, from tau, other secondary, and displaced.

The model's performance at the track classification task can be assessed using a confusion matrix, which compares the true origins of the tracks to the model's predictions; a perfectly efficient matrix would be diagonal. Confusion matrices with and without the weights are shown

in Fig. 5. The addition of the weights significantly improves the model's ability to correctly predict track origins. With the weights, the model's ability to identify pile-up, fake, primary, other secondary, and displaced tracks is quite good (it is best at displaced tracks, with an efficiency of 95.5%). There is some confusion between the different types of prompt tracks, and it cannot identify tracks from tau particles at all, but the performance at track origin classification is reasonably good overall. The effect of the extended labelling scheme on the model's performance at the primary jet classification task is discussed in Model Performance.

## 2.4 Updated Training Samples

The final update made to the supervised jet tagging algorithm was to the samples used to train it, which are selected from simulated Monte Carlo samples. There are an equal number of signal and background jets, and they are usually evenly split into two disjoint folds. In each fold, most of the jets are used for training, with a small subset reserved for validation. A model trained on jets from the first fold should be evaluated on jets from the second fold, and vice-versa.

The training dataset used in the previous analysis consisted of 12.1 million total jets, with 6.05 million in each fold. In each fold, 5.5 million jets were used for training, and 550 thousand for validation. To try to improve the model, a new dataset was created with 20 million training jets and 2 million validation jets in one fold, and 19.8 million training jets and 1.98 million validation jets in the other fold. The new dataset includes several updates to the signal model, with the goal of training the algorithm on a broader range of possible signal signatures in order to be sensitive to a larger parameter space. The old dataset only contained dark meson lifetimes of 5 and 50 mm, while the new dataset contains lifetimes of 1, 5, 10, 50, 100, 500, and 1000 mm. The old dataset used signal masses of 600, 1500, and 3000 GeV, to which the new dataset adds masses of 800, 1000, 1200, 1800, 2200, and 2500 GeV. Additionally, while the old dataset only included samples generated using the $s$-channel model, the new dataset includes the $s$-channel and $t$-channel models. Finally, a change was made to the transverse momentum ($p_T$) distribution of the background jets in order to try to reduce the dependence of the algorithm's efficiency on $p_T$. There were previously fewer background jets than signal jets at lower $p_T$ due to how the distributions were matched. In the new dataset, the background jets are resampled to exactly match the signal $p_T$ distribution.

## 2.5 Model Performance

The effect of the extended labelling scheme and of the updated dataset can be assessed using Fig. 6. The relevant curves to compare are the "Nominal" baseline curve from the previous analysis (trained using the old samples and the old track labelling scheme) and the "Trained on old, evaluated on old" curve, which is the same model, except with the new track labelling scheme. There is a clear improvement in performance, indicating that the extended track labelling scheme was a successful update to the model. The model's performance was also evaluated with and without the weights applied to the extended track labelling scheme. Relevant ROC curves are plotted in Fig. 7. There is a very marginal increase in performance with the addition of the weights, so the weights were kept primarily because they significantly improve the confusion matrix (as shown in Fig. 5).

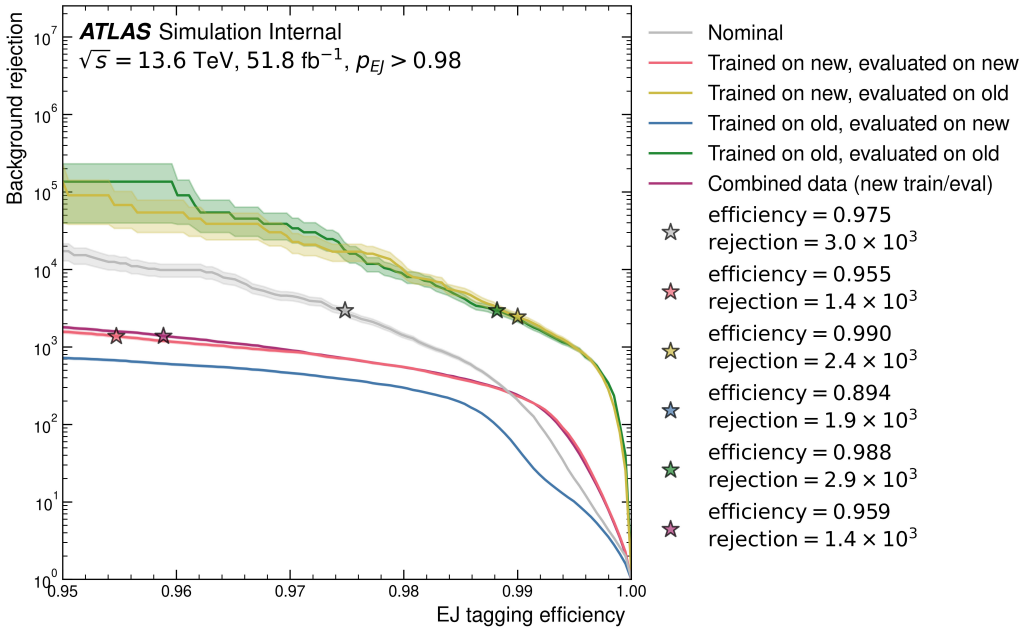



Figure 6: ROC curves for models trained and evaluated on different combinations of old and new samples. The "Nominal" curve (from the previous analysis) is trained and evaluated on the old samples, with the old track origin labelling scheme (without weights). All other curves have the extended labelling scheme, with weights.

Fig. 6 can also be used to assess the effect of the new training dataset on the model's performance. The only difference between the "Trained on old, evaluated on old" and "Trained on new, evaluated on old" curves is which dataset is used to train the model. The performance is the same, indicating that including new samples in the training dataset does not decrease the model's efficiency at tagging samples in the old dataset. However, comparing the "Trained on old, evaluated on new" and "Trained on new, evaluated on new" curves shows a clear improvement in the model's performance, indicating that including new samples in the training dataset does improve the model's ability to tag the new signal samples. The updates to the training dataset have thus enhanced the model's ability to tag the newly included signal samples, without losing any ability to tag the old ones, which is a clear improvement to the algorithm. It should be noted that both of the "evaluated on new" curves are below the baseline curve of the previous analysis, but this is because the new dataset includes jets that are much harder to tag (such as those with longer lifetimes). Although the overall performance has decreased, it is in fact a better jet tagging algorithm.

A model was also trained on the new data from both folds at once (for a combined dataset of 39.8 million training jets) to test whether a larger number of training jets improved the performance. The increase was marginal, as can be seen by comparing the "Trained on new, evaluated on new" and "Combined data (new train/eval)" curves in Fig. 6. These are the same model, but with the former trained on only the fold with 20 million jets, and the latter trained on the entire combined data set.

To examine in more detail where the model's performance has improved, its efficiency as a function of signal lifetime is plotted in Fig. 8. Models trained on the new samples show a significant improvement at higher lifetimes compared to those trained on the old samples. The performance is similar at intermediate lifetimes, and slightly improved at very low lifetimes.

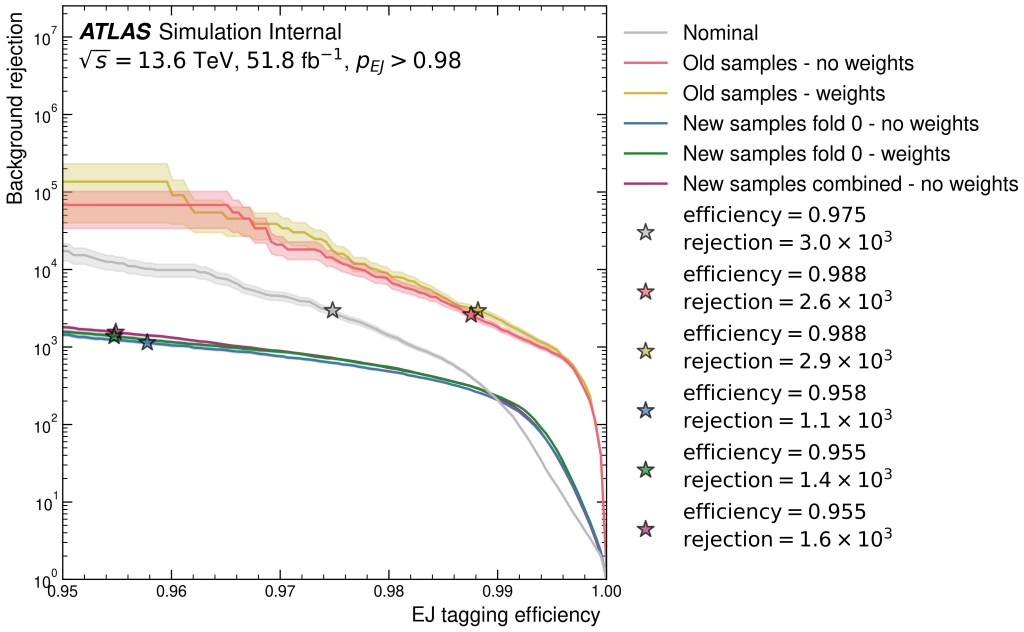



Figure 7: ROC curves for models trained with and without weights (which penalize the model more heavily for incorrectly classifying rarer tracks) used for track origin classification. All curves except for "Nominal" (which is the baseline model used in the previous analysis) are trained with the extended track labelling scheme.

This indicates that the inclusion of signal jets with much longer lifetimes in the training dataset has had the desired effect of improving the algorithm's efficiency at longer lifetimes, without losing any performance at other lifetimes. The slight increase at very low lifetimes could be from the inclusion of 1 mm lifetimes. Using weights for the track origin classification task does seem to slightly decrease the new model's efficiency at longer lifetimes, but the effect is much smaller than the overall increase in performance from the new samples.

Finally, the model's efficiency as a function of $p_T$ is plotted in Fig. 9. The main curves to compare are the baseline curve from the previous analysis, and the "Evaluated on new, trained on new" curve, which is the fully updated tagger. The new tagger's efficiency is lower overall for the same reason as for the ROC curve. The $p_T$ dependence of the old and new models seems to be approximately the same, indicating that the update to the background jet $p_T$ distribution has not had the desired effect of reducing $p_T$ dependence.

Overall, the updates to the supervised jet tagger have been successful. The extended track labelling scheme and updated dataset have strictly improved the model's performance, particularly at tagging jets with longer lifetimes, and the algorithm is essentially ready to be used in a future analysis.

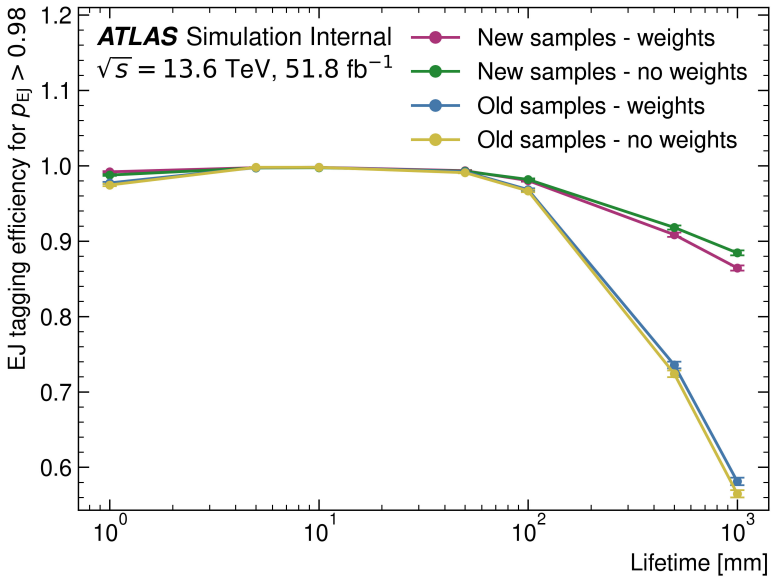



Figure 8: Emerging jet tagging efficiency as a function of signal lifetime. Training the model on new samples significantly improves performance at longer lifetimes, while keeping the same performance at intermediate lifetimes.

# 3 Unsupervised Jet Tagger

## 3.1 Initial Model Architecture

The unsupervised jet tagger is a variational autoencoder (VAE). The general structure of a VAE is that it first consists of an encoder, which is a neural network that compresses the input data into a latent space. The latent space is a lower-dimension representation of the data that is meant to capture its most important features and patterns. The encoder outputs parameters describing a probability distribution in the latent space (in this case, the mean and the logarithm of the variance of a normal distribution), and latent variables are randomly sampled from this distribution. Another neural network, the decoder, uses the latent variables to reconstruct an output as similar to the input as possible, while maintaining a regularized latent space (in which points that are close together correspond to similar inputs). The VAE is trained on background data only, and thus should only be able to effectively reconstruct background jets. Loss scores should then be low for background jets and high for signal jets, allowing for the identification of possible signal jets. Because the algorithm is not trained using a particular signal model, the approach is said to be model-independent: the neural network should tag anomalous jets (that differ from the Standard Model background), but unlike in the supervised approach, these need not correspond to a specific signal model.

Currently, the unsupervised jet tagger is trained on track features only. The variables are the same as those in Table 1, but without jet $\eta$ and with seven more track-level variables: chi squared, radius of first hit, $\theta$, number of degrees of freedom, track $p_T$, $z_0$ relative to beamspot, uncertainty in $z_0$ relative to beamspot, and uncertainty in $z_0 \sin(\theta)$. Jets are modelled as sets of tracks of variable length, as there are up to 200 tracks for each jet. To encode these variable-length sets, a Deep Set architecture [6] is used. Tracks are individually encoded by several neural network layers (with ReLU activation), and attention pooling is then used to combine the encoded tracks into a global jet representation of fixed dimension. Latent variables

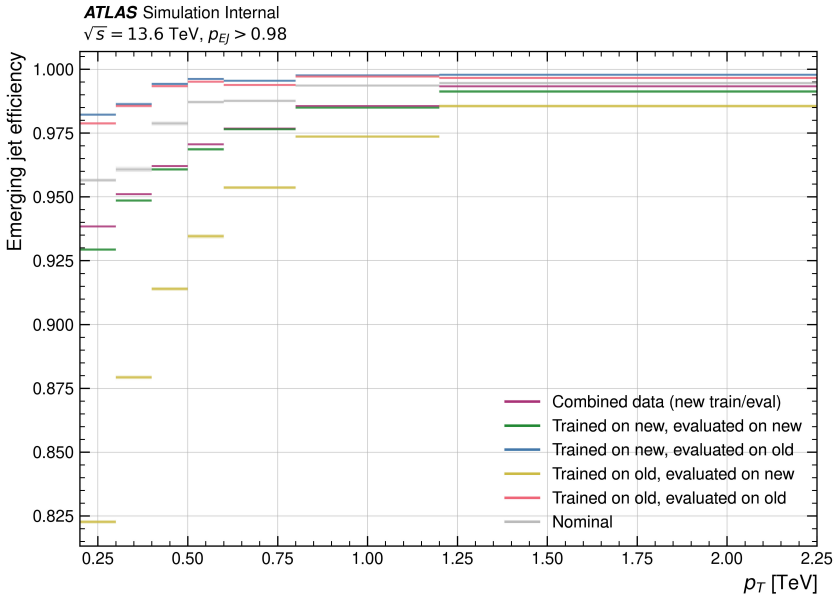



Figure 9: Emerging jet tagging efficiency as a function of $p_T$. The $p_T$ dependence of the new tagger ("Trained on new, evaluated on new") seems similar to that of the old one ("Nominal").

(the mean and the logarithm of the variance) are then sampled from the latent distribution. At this point, the number of tracks in the reconstructed jet must also be generated. The initial approach was to treat this similarly to a latent variable, and generate the mean and the logarithm of the variance of the number of tracks (these were usually generated from the encoded track representation rather than the global jet representation, but tests using the global jet representation and the unencoded track data had the same performance). The mean number of tracks was then scaled to be between 0 and 200 by taking the sigmoid (which normalizes the mean to between 0 and 1) and multiplying by 200. Once the number of tracks in the reconstructed jet is determined, the decoder generates each track in the jet, using the same layers as the encoder but with the sequence reversed.

The loss function contains three terms:

$$L_{\text{total}} = L_{\text{recon}} + L_{\text{KL}} + L_{\text{track number}} \tag{2}$$

The first term is meant to quantify the difference between the input and reconstructed jets. It is currently the Chamfer Distance, chosen because it has already been used for VAEs that operate on sets [7]. The Chamfer Distance measures the similarity between two sets of points (in this case, two sets of tracks). For each point in the first set, the nearest point in the second set is found, and the distance between them is computed. The same is done for the second set, and the Chamfer Distance is then the sum of the squares of all these minimum distances. The second term, $L_{\text{KL}}$, is the Kullback–Leibler (KL) divergence. It quantifies the difference between two probability distributions, and is used in VAEs to regularize the distributions of the latent variables by encouraging them to resemble a standard normal distribution. The third term in the loss, $L_{\text{track number}}$, is meant to push the input and reconstructed jets to have the same number of tracks. The initial approach was to also use the KL divergence for this term.

## 3.2 Performance and Modifications

The first step in creating the unsupervised algorithm was to implement the Deep Set architecture. To make sure it performed as expected, it was tested with a supervised jet classification task as the output. This was successful, so the classification task was replaced with a latent space and decoder to create a VAE. Several adjustments were then made to develop a functional algorithm. For numerical stability, the logarithm of the variance of the latent variables is clamped between $-5$ and 5. The Chamfer distance is significantly larger in magnitude than the other two terms in the loss, so it is scaled by $1 \times 10^{-6}$ or by taking the logarithm of the Chamfer distance (both of these approaches have similar results). Linear scalings other than $1 \times 10^{-6}$ were tested, but the performance was the same, albeit with different magnitudes of the loss. After testing several possible configurations of layers in the Deep Set, an input dimension of 24 (determined by the 24 track features), three hidden layers of dimensions 128, 64 and 32, and then a latent dimension of 16 were chosen. The decoder reverses this order. More layers or dimensions did not seem to improve the performance, whereas fewer did worsen it.

The $L_{\text{KL}}$ and $L_{\text{track number}}$ terms seem to drop in magnitude much faster than the $L_{\text{total}}$. To try to mitigate this, KL annealing was tested: this consists of multiplying $L_{\text{KL}} + L_{\text{track number}}$ by an annealing coefficient, which usually starts at 0 and later increases in value. Linear KL annealing, in which the coefficient linearly increases from 0 to 1, was implemented. The drop in KL divergences remained approximately the same, but the separation between background and signal loss scores did improve. Of the tested combinations of epochs at which to start and end the increase in the annealing coefficient, starting at the fifth epoch had the best results; the end epoch did not seem to matter much. Cyclical annealing with a cosine function was also tested, but performance was similar to linear annealing.

Another observed issue with the unsupervised tagger was the generation of the number of tracks in the reconstructed jet. The distributions of the mean and the logarithm of the variance of the track number were very tightly clustered around 0, resulting in almost all reconstructed jets having 100 tracks. This did not match the true distribution, which has a much broader spread. To try to correct this, a new version of the algorithm was tested, in which a supervised task was implemented only for the track number: the validity of each track is individually predicted by generating a score between 0 and 1 (tracks with scores above 0.5 are valid, otherwise they are invalid). $L_{\text{track number}}$ is now a binary cross entropy loss, similar to the losses for the classification tasks in the supervised jet tagger. With this approach, the reconstructed track number distribution is much closer to the true distribution, as shown in Fig. 10, and the separation between signal and background loss scores also improves. Two main variants of this model have been tested: one which generates the track validity scores from the encoded track representation, and one which uses the global jet representation. The former has a track number distribution which more closely resembles the true track number distribution, whereas the latter has a better separation between signal and background loss scores.

Overall, development of the unsupervised jet tagger is in progress, and the tests and changes that have been made are very much experimental. Its current state is shown in Fig. 11: there is a slight separation between the loss scores of background and signal jets, but there remains a lot of work to be done to refine and optimize the algorithm. Nevertheless, the current model is a good starting point for future work, and the unsupervised tagger is a promising approach to a more general search for dark matter within the framework of dark sector models.

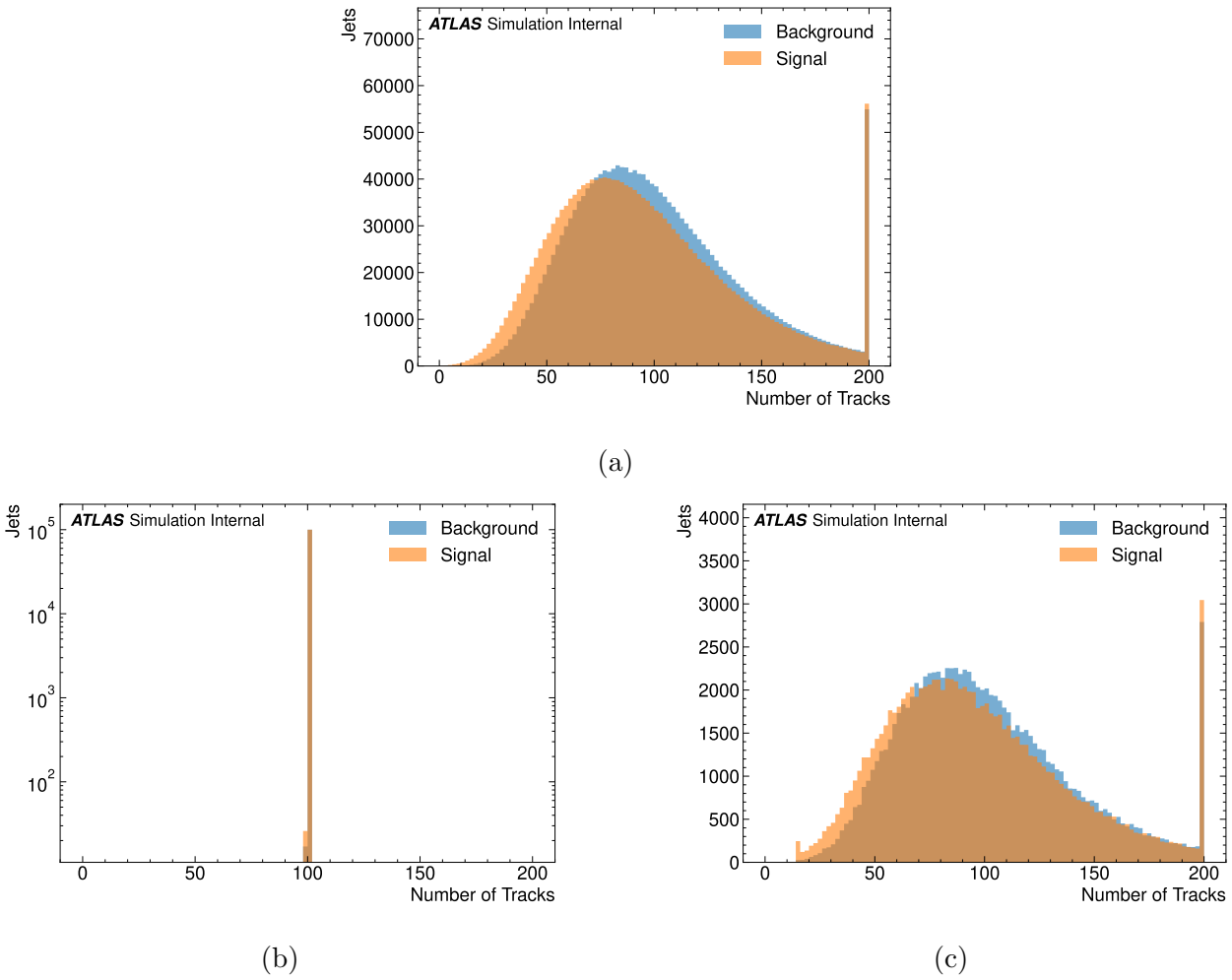
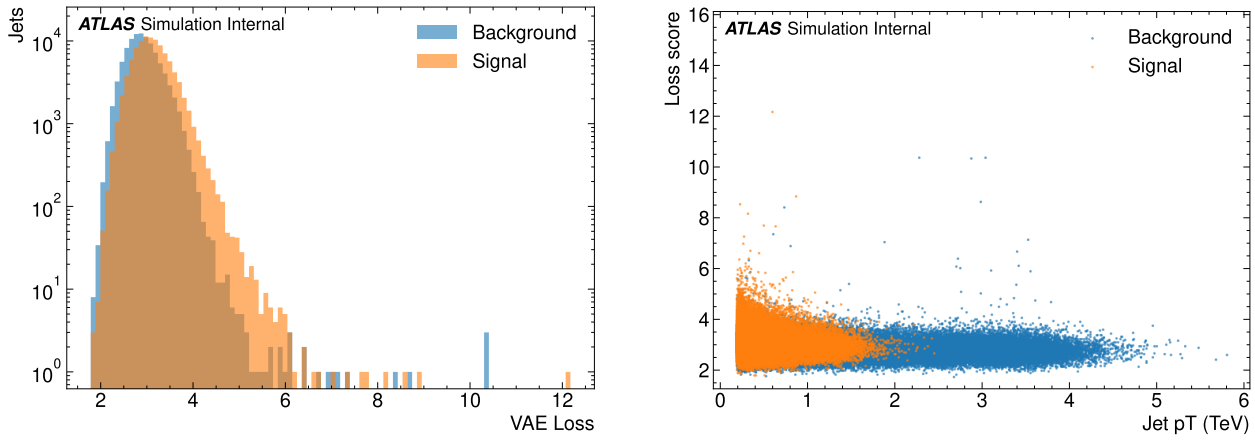



Figure 10: (a) shows the true track number distribution, (b) shows the reconstructed track number distribution of the fully unsupervised model (note the y-axis log scale), and (c) shows the reconstructed track number distribution of the model with a supervised task for track number generation. The small signal spike at low track numbers in (c) is likely because the model is only trained on background, and so doesn't learn that jets can have very few tracks.

# 4 Conclusions and Future Work

Efforts to search for evidence of dark matter in the form of emerging jets in the ATLAS detector are ongoing, and neural networks offer a powerful method to identify possible signal jets. A supervised jet tagging algorithm, which was used in a previous analysis, has been successfully improved. An extended track labelling scheme and new training dataset have enhanced the model's performance, with the most significant increases in efficiency at longer dark meson lifetimes. The model trained with the new samples is shown to perform better at tagging them without losing any performance at tagging the old samples, making it a strictly better algorithm. There is still room for improvement, for instance in reducing the signal efficiency's dependence on $p_T$, but the current changes should already significantly improve sensitivity in future analyses.

In parallel, an unsupervised jet tagger has been developed, although many refinements and




(a) (b)

Figure 11: (a) shows a histogram of the loss scores of background and signal jets, with a small amount of separation between the two. This iteration of the model was trained using the logarithm of the Chamfer loss, and with the supervised task for reconstructed track number generation. (b) shows the loss scores of signal and background plotted against $p_T$; there currently does not appear to be any obvious correlation between loss and $p_T$ (which is the desired outcome).

extensions to this algorithm must still be made to improve its performance. For instance, it could be possible to use the Earth Mover's Distance instead of the Chamfer Distance to quantify the similarity between input and reconstructed jets [7], or to test other ways to generate the number of tracks in the reconstructed jets. Further work may also involve training the model with reconstructed vertex information, which may help it identify displaced vertices and correlate them with anomalous jets. Additionally, much more thorough testing of this model still needs to be performed in order to gain a better understanding of its performance and any biases it might have. However, the current model already demonstrates some ability to discriminate between signal and background jets, and shows that it could be possible to conduct a model-independent search in the future.

# 5 Acknowledgements

I would like to extend my warmest thanks to Jackson Burzynski and Matthias Danninger for their support and guidance throughout this project. I am also grateful for the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through an Undergraduate Student Research Award (USRA) at Simon Fraser University, and for the support of the Institute of Particle Physics through the IPP Summer Student Fellowship Program. Finally, thank you to the CERN Summer Student Programme and to all the friends that I've made for an amazing summer.

# References

[1] Douglas Clowe et al. "A Direct Empirical Proof of the Existence of Dark Matter". In: *The Astrophysical Journal* 648.2 (Aug. 2006), p. L109. DOI: 10.1086/508162. URL: https://dx.doi.org/10.1086/508162.

[2] Matthew J. Strassler and Kathryn M. Zurek. "Echoes of a hidden valley at hadron colliders". In: *Physics Letters B* 651.5 (2007), pp. 374–379. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2007.06.055. URL: https://www.sciencedirect.com/science/article/pii/S0370269307007721.

[3] Pedro Schwaller, Daniel Stolarski, and Andreas Weiler. "Emerging jets". In: *Journal of High Energy Physics* 2015.5 (May 2015). ISSN: 1029-8479. DOI: 10.1007/jhep05(2015)059. URL: http://dx.doi.org/10.1007/JHEP05(2015)059.

[4] ATLAS Collaboration. *Search for emerging jets in pp collisions at* $\sqrt{s} = 13.6$ *TeV with the ATLAS experiment*. Tech. rep. Geneva: CERN, 2025. ARXIV: 2505.02429. URL: https://cds.cern.ch/record/2931240.

[5] ATLAS Collaboration. *Graph Neural Network Jet Flavour Tagging with the ATLAS Detector*. Tech. rep. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2811135.

[6] Manzil Zaheer et al. "Deep Sets". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

[7] Jinwoo Kim et al. "SetVAE: Learning Hierarchical Composition for Generative Modeling of Set-Structured Data". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 15054–15063. DOI: 10.1109/CVPR46437.2021.01481.