Transfer Learning for Jet Tagging in Particle Physics

Jade Ducharme* and Egor Serebriakov†
Department of Physics, Brown University, Providence, RI 02912 USA
(Dated: May 10, 2024)

Data volumes collected during particle collision events at CERN require vast amounts of computational resources to store and process. Traditional methods, which have proven proficient at differentiating interesting top quark collisions from less interesting background events (a binary classification task known as jet tagging), struggle to keep up with the ever increasing data rates. Even newly researched machine learning approaches, often featuring complex architectures such as graph neural networks in order to achieve high accuracy, fail to reach sufficiently low inference times for real-time processing. In this work, we explore the two ways in which Transfer Learning can help with reaching real-time jet tagging at CERN. First, we explore how knowledge distillation from a Teacher model trained on high-resolution jets to an identically constructed Student model trained on lower-resolution jets can help by enabling us to save a smaller volume of data to disk. We then explore how knowledge distillation from a large, robust Teacher to a smaller and more compact Student, both trained on high-resolution data, can help by decreasing the inference time. We consider both the fully-connected neural network and the graph neural network architectures for both experiments.

Keywords: CERN, machine learning, deep learning, jet tagging, transfer learning, graph neural networks

I. INTRODUCTION

High-energy collisions between massive particles lead to the formation of a collimated spray of less energetic secondary particles; this phenomenon is known as a jet. Jets are particularly useful as probes to help us investigate potential physics beyond the Standard Model (SM), which aims to describe every particle known in the Universe. Currently, the SM features six different flavors of quarks, six leptons, and a handful of mediators responsible for particle interactions via three out of the four fundamental forces – strong nuclear, weak nuclear, and electromagnetic (gravity not being included in the SM). For a more in depth introduction to jets and particle physics as a whole, the interested reader is referred to Griffiths 2010 [1].

Experiments at CERN aim to push these boundaries and hope to uncover new particles not yet described by the SM. For example, through ambitious experiments and careful data analysis, the ATLAS and CMS collaborations at CERN were able to prove the existence of the Higgs boson in 2012, a groundbreaking discovery that forced us to reconsider the SM [2], [3].

Similar experiments are constantly being conducted at CERN, with perpetually increasing data volume rates as new technological advances are being made. We are currently seeing collision rates of 40 million events per second at CMS [4]; this number is expected to be increased significantly with the upcoming High-Luminosity upgrade to the Large Hadron Collider [5].

* Correspondence email address: jade_ducharme@brown.edu
† Correspondence email address: egor_serebriakov@brown.edu

II. OBJECTIVE

Our goal is to build upon the work of Qu and Gouskos [6] by implementing transfer learning for jet tagging. The current state-of-the-art jet taggers achieve remarkable accuracy on simulated top, bottom, and gluon tagging datasets [6], [7], [8]. However, in order to achieve such high accuracy, their architectures are highly advanced and complex, which leads to a very slow inference time. In order to be able to process data in real time at CERN, we need to be able to do one of two things:

- First, we could significantly decrease the volume of data we aim to process.
- Second, we could substantially reduce the size (i.e. number of trainable parameters) of our tagging model.

In this work, we explore both approaches, individually detailed in Subsections II B and II C, in the context of transfer learning.

A. Transfer Learning

Transfer learning is a way of distilling knowledge from a so-called Teacher model, exhibiting excellent performance, to a so-called Student model with poorer individual performance.

Following the approach presented in [9], this is done by first defining a classification loss, obtained using the Student's predictions, as shown below:

$$L_{CE} = -\sum y_i \log P_s(i), \tag{1}$$

where y_i corresponds to the true label, and $P_s(i)$ to the Student's prediction. Next, we also define a distillation loss which quantifies how well the Student's predictions match the Teacher's:

$$L_{KD} = T^2 \sum P_t(i) \log \frac{P_t(i)}{P_s(i)}, \tag{2}$$

where $P_t(i)$ is the Teacher's prediction and T is an adjustable hyperparameter called the temperature. Its purpose it to diffuse the Teacher's loss in order to reduce the confidence and encourage the Student to learn by itself.

Finally, in order to obtain the total loss, these two terms are combined together using the following equation:

$$L = \alpha L_{CE} + (1 - \alpha) L_{KD}, \tag{3}$$

where α is another adjustable hyperparameter with values between zero and one. Pushing α closer to zero places more weight on the distillation loss whereas pushing it closer to one places more weight on the classification loss.

B. Data Quality Reduction

Our first experiment involves reducing the data volume we need for analysis. In order to do so, we start by considering how the quality of our data is related to the data volume.

The LHC is composed of several interacting components that enable us to obtain the precise features (momentum, energy, azimuthal angle, pseudo-rapidity) of all constituent particles that make up a jet. A simplified diagram of the LHC is shown in Figure 1. Modeling all of these components is beyond the scope of this work, so we will focus chiefly on the tracker.

The tracker is composed of several layers or "shells" which are plastered with tiny detection cells [4]. When a particle hits a detection cell, its position and momentum are recorded. Due to the large number of events happening every second, and the large number of particles in a single event, the number of detector hits that must be recorded and saved to disk is nearly unfathomable.

In order to mitigate this, some of the tracking shells can be turned off. This introduces some uncertainty in the particles' energies and momenta, since there could now be several different paths the particle could have taken through the tracker. However, it reduces the overall amount of data needed to process. This process of turning off some tracking shells is shown in Figure 2

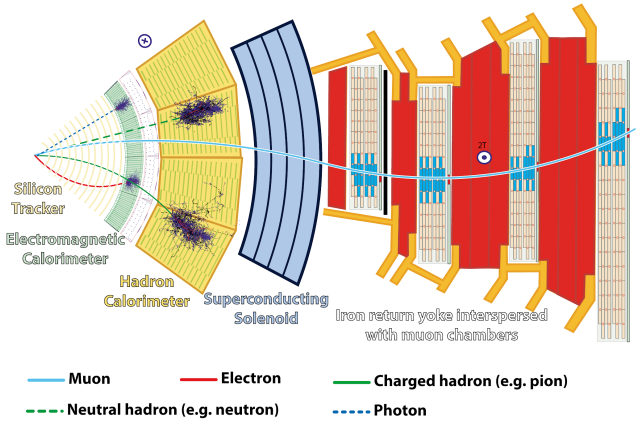

Figure 1: Simplified cross-sectional diagram of the LHC. The collision event occurs on the left. Emitted particles first travel through the tracker (where their precise trajectory is recorded), the calorimeters (where they deposit energy and lose momentum), the superconducting solenoid (where their trajectories are bent by the strong magnetic field), and finally the muon chambers (which only muons are able to reach, as all other types of particles have lost all energy and momentum by then). Image source (clickable): *Reconstructing thousands of particles in one go at the CERN LHC with TensorFlow*.

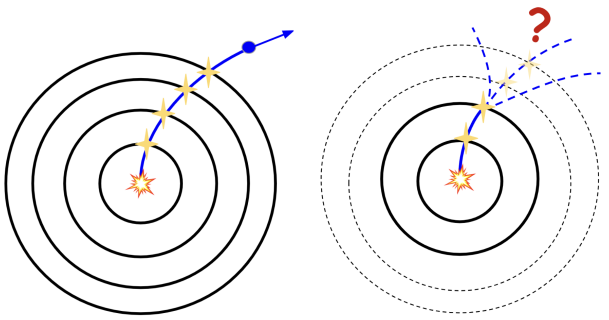

Figure 2: Simplified cross-sectional view of the tracking portion of the LHC. (Left) All tracking shells are in use, and precise particle tracks are reconstructed. (Right) Some tracking shells are inactive (dashed shells), introducing uncertainty in the particles' exact trajectories.

A detailed description of how we simulate low-resolution data in this work is provided in Section III C.

C. Model Compression

Our second approach to reducing the computational costs associated with data analysis at CERN consists in compressing existing state-of-the-art taggers. Specifically, we build a smaller model (i.e. a model with less layers and trainable parameters) which will act as our Student model for this experiment. This Student model, on its own, is not expected to perform comparably to the state-of-the-art model (the Teacher). However, using knowledge distillation, we can transfer some of the Teacher's knowledge to the Student in order to encourage it to learn a more meaningful representation of the data, leading to increased accuracy with a smaller inference time. Specific architectures considered in this work are detailed in Section III.

III. METHODS

In this section, we first introduce our training data as well as all adopted preprocessing steps in Subsections III A and III B. We then present the methods used to implement knowledge transfer for our two explored contexts: data quality reduction, detailed in Subsection III C, and model compression, detailed in Subsection III E. We also discuss the different architectures considered in this work in Subsection III D.

A. Training Data

All our models are trained on the publicly available ATLAS Top Tagging Open Data Set. This dataset contains over 44 million randomly distributed simulated jet events and occupies over 120 Gb of disk space. Every jet event contains information about as many as 200 constituent particles, and each of these particles possesses four features: energy (E), transverse momentum (p_t), azimuthal angle (ϕ), and pseudo-rapidity (η). Together, these four features serve to completely characterize the constituent particle.

Since we have neither the RAM nor the patience to train on 44 million samples, we select the first 500,000 as our Teacher set, and the following 500,000 as our Student set. We verified that the binary label distribution is about 50/50 for both of these sets. We split the Teacher and Student sets into training, validation, and test sets using a 70-15-15 split. The training sets therefore each have the following shape: `[350000, 200, 4]`, corresponding to `[INPUT_SZ, NUM_CONSTITUENTS, NUM_FEATURES]`.

B. Preprocessing

Following the approach presented by the ATLAS Collaboration [10], we first shift each jet so it is centered at the origin of the η-ϕ plane, since the jet features translational invariance in this plane. Next, we preprocess the energy and transverse momentum by taking their logarithms in order to limit their dynamic ranges (this places them on a $\mathcal{O}(1)$ scale).

Next, still according to the ATLAS guidelines, we calculate three additional features for each constituent. The first one is the distance from the jet axis, calculated as follows:

$$R = \sqrt{\eta^2 + \phi^2} \tag{4}$$

The next two are found by normalizing the constituent transverse momentum (energy) by the total transverse momentum (energy) in the jet and taking the logarithm of the resulting fraction.

The ATLAS collaboration has found that these specific preprocessing steps and the inclusion of these additional features provide a significant boost in top tagger performance. After implementing the above steps, our training data now has shape `[350000, 200, 7]`.

Finally, we apply basic standardization in order to center the mean of each feature on zero with a standard deviation of one (of course, using only the training set and recording this mean and standard deviation for standardization of the validation and test sets).

C. Data Quality Reduction

In this work, we simulate the reduction in data quality due to having less operational tracking shells in two different ways. First, we can diffuse the constituent particles' energy and momenta via the addition of Gaussian-distributed random noise. This is shown in Figure 3.

Secondly, given that our input data is made up of N constituents per jet, we can sample $n < N$. For every experiment considered in this work, the undersampled data retained 5 out of the total 200 available constituents.

D. Architectures Considered

For this project, we consider two model architectures. First, we implement a fully-connected (dense) neural network (FCNN) to serve as a benchmark. We present the architecture details in Figure 4.

The second model we consider is a graph neural network (GNN) whose architecture closely mimics the one

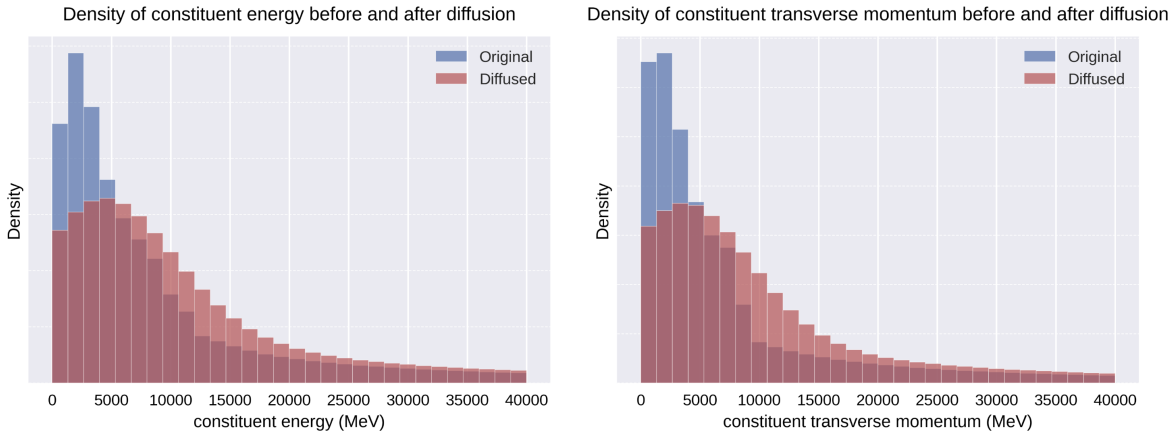

Figure 3: Energy (left) and momentum (right) diffusion via the addition of Gaussian-distributed random noise.

Hidden layers:	*5*
Nodes per layer:	*400*
Activ. function:	*ReLU*
Learning rate:	*1.2e-5*
Input size:	*500,000*
Batch size:	*250*
Optimizer:	*Adam*

Figure 4: Architecture details for the Teacher fully-connected (dense) neural network. The hyperparameters are set according to the ATLAS recommendations [10].

presented in Qu and Gouskos [6]. We present its details in Figure 5.

It features three so-called "edge convolutional" layers that extract features along the graph's edges. After passing through the edge convolutional layers, the data are then passed through a max-pooling layer, followed by a fully-connected layer to return the binary predictions.

The edges are constructed using the k-nearest-neighbors algorithms using $k = 16$.

k (for KNN):	*16*
Edgeconv layers:	*3*
Activ. function:	*ReLU*
Learning rate:	*3e-4*
Input size:	*250,000*
Batch size:	*384*
Optimizer:	*AdamW*

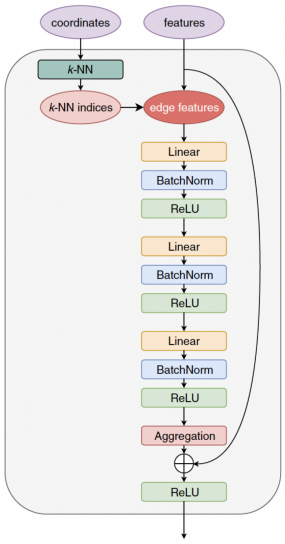

Figure 5: Architecture details for the Teacher graph neural network. The hyperparameters are set according to Qu and Gouskos [6].

E. Model Compression

We implement "compressed" Student models for both our FCNN and GNN experiments. For the FCNN, the compressed Student features one single hidden layer with 64 nodes (all other hyperparameters remain the same as in Figure 4). Meanwhile, for the GNN, the compressed Student features one edge convolutional layer with a single hidden linear layer inside (all other hyperparameters remain the same as in Figure 5).

IV. RESULTS

A. Phase I: FCNN

The first stage of this project was focused on the simple FCNN to serve as a benchmark. We performed two experiments: first, we implemented transfer learning between a Teacher FCNN trained on the original high-quality data and a Student FCNN trained on the degraded (both via diffusion and undersampling) data. The goal of this experiment is to verify the potential to turn off some detector tracking shells in order to save

computational resources at the LHC. For this experiment, both the Teacher and Student were constructed with the exact same architecture shown in Figure 4. We tested out the transfer learning for the diffused and undersampled data separately, and present the resulting ROC curves in Figure 6.

In both cases, we surprisingly note that the Student trained only on the degraded data performs almost as well as the Teacher trained on the original data. However, we also regrettably note that the knowledge distillation wasn't effective and that the Student's performance actually goes down after implementing transfer learning.

Our current hypothesis, although we did not have enough time to properly verify it, is that the Teacher learns a representation of the original data that doesn't generalize to the degraded data. As a result, it gives the Student "bad advice", leading to an overall poorer performance.

The second experiment we performed using the FCNN is the implementation of transfer learning between a Teacher FCNN featuring the more complex architecture shown in Figure 4, and a Student FCNN featuring the simpler architecture discussed in Subsection III E, both trained on the original data without degradation. Since the goal of this specific experiment is to verify whether a compressed model can achieve similar accuracy with a lower inference time, we also keep track of this latter quantity. The ROC curves for this experiment are presented in Figure 7.

This experiment proved successful. The Teacher achieves the overall best performance, followed by the Student after distillation, and finally the Student on its own. Additionally, we were able to reduce the inference time by approximately 15% (although more iterations are required to get a more robust figure, since the inference time is relatively short and is subject to considerable statistical fluctuations).

B. Stage II: GNN

The second stage of this project consisted in repeating all the experiments we performed using the FCNN, this time using a much more powerful and data-appropriate GNN.

For the knowledge distillation between a Teacher GNN trained on the original data and a Student GNN trained on the degraded (both via diffusion and undersampling) data, where the Teacher and Student are both designed with the architecture presented in Figure 5, we present the resulting ROC curves in Figure 6.

As with the FCNN, the knowledge distillation for this experiment was not effective. The Teacher achieves a good performance on its own using the original data, but doesn't seem to learn a representation that generalizes to either the diffused or undersampled data.

Finally, our last experiment consisted in implementing transfer learning between a complex Teacher GNN, built according to Figure 5, and a more compact Student GNN, built as discussed in Subsection III E, both trained on the original data. The resulting ROC curves for this experiment are shown in Figure 7.

This experiment also shows considerable success, as we are able to recover 0.03 AUC points between the Student on its own and the Student with knowledge transfer. Additionally, the inference time for the Student is approximately three times smaller than for the Teacher, which is a considerable advantage considering the volume of data processed by the LHC.

All experiments were run using the Ocean State Center for Advanced Resources (OSCAR), which is Brown University's high performance computing cluster. Training was conducted using Nvidia-compatible GPUs, mainly `quadrortx` or `titanrtx`, although the exact GPU used for any given job was left to the OSCAR scheduler.

V. CONCLUSIONS

The first category of experiments we conducted, wherein an identically constructed Teacher and Student are respectively trained on the original and degraded data, did not produce significant results. Both for the FCNN and GNN architectures we considered, the Student performed less well after we implemented knowledge distillation. Our current hypothesis is that the Teacher, being trained on completely different dataset, is unable to learn a representation that generalizes to the degraded data, and provides the Student with useless advice that reduces its performance. However, more research is needed in order to confirm this. Additionally, we believe that this experiment could succeed given a more thought-out approach. For example, perhaps the Teacher could be trained on a combination of the original and degraded data in order to force a more generalizable representation? Or perhaps there are specific preprocessing steps we could apply to either the original or degraded data sets to ensure mutual generalizability? Moreover, there is room for improvement in the specific implementation of knowledge distillation. Our approach follows the work presented in Hinton et al. 2015 [9], but newer and potentially more appropriate approaches have since been published. These implementations are beyond the scope of this work but represent potential future directions for this research.

The second category of experiments we conducted, wherein a larger, more complex Teacher model and a smaller, more compact Student are both trained on the original data, showed encouraging results for both the

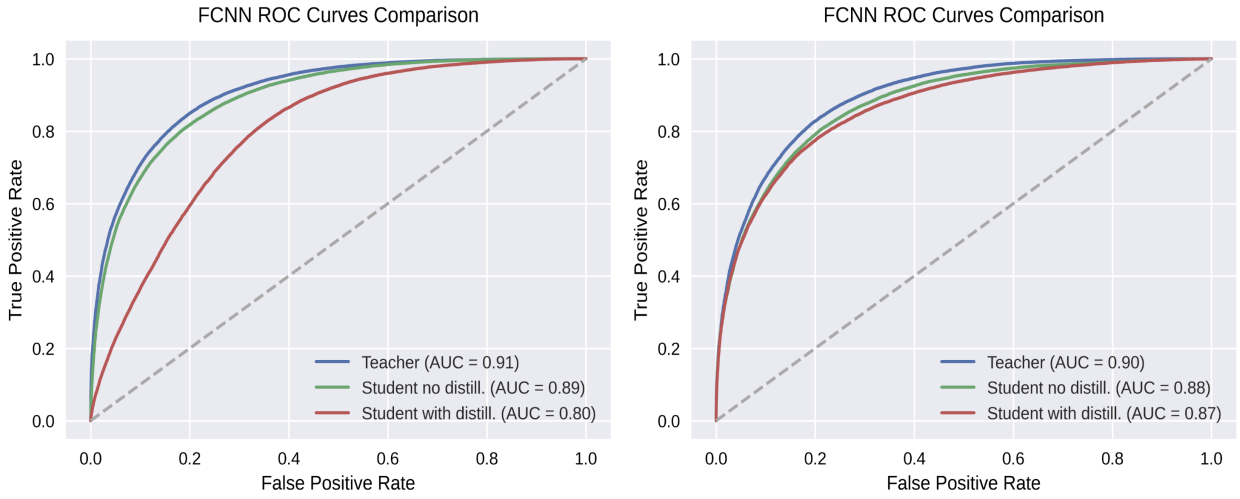

Figure 6: ROC Curve comparison for the data degradation experiment using the FCNN. (Left) Data degraded via diffusion along the energy and momentum axes. (Right) Data degraded via undersampling by selecting only 5 out of the 200 total constituents.

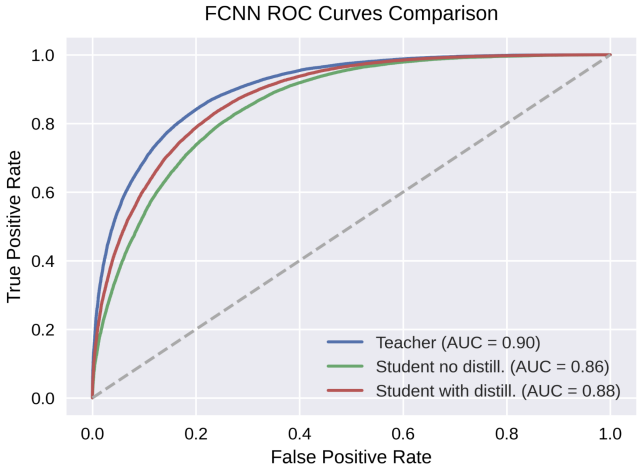

Figure 7: ROC curves for knowledge distillation between a large Teacher FCNN and a more compact Student FCNN, both trained on the original, undegraded data.

FCNN and GNN architectures. Additionally, the inference time for both compressed Students was reduced compared to their paired Teachers. Given the huge volume of data constantly streaming out of LHC experiments, reducing the inference time as much as possible will prove invaluable if we ever hope to achieve real-time data processing.

ACKNOWLEDGEMENTS

We thank Professor Singh and all the CSCI 1470/2470 TA staff for the constant help and support during the semester. We are also thankful that the ATLAS collaboration at CERN made the dataset used in this work publicly accessible. Special thanks to Professor Gouskos from the Department of Physics for the detailed introduction into the current "state-of-art" DL techniques for jet tagging as well as explaining physics of particle collisions at LHC.

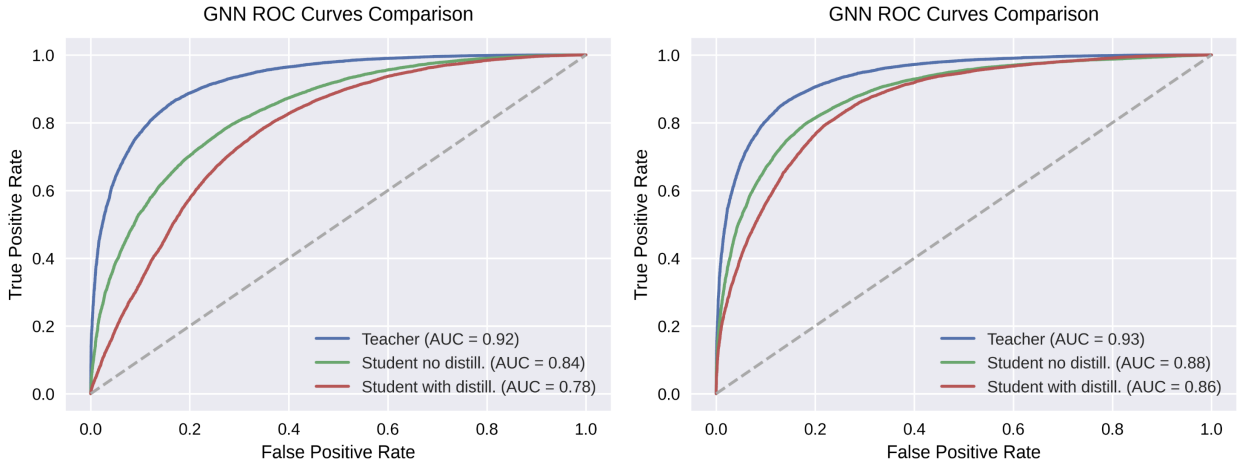

Figure 8: ROC Curve comparison for the data degradation experiment using the GNN. (Left) Data degraded via diffusion along the energy and momentum axes. (Right) Data degraded via undersampling by selecting only 5 out of the 200 total constituents.

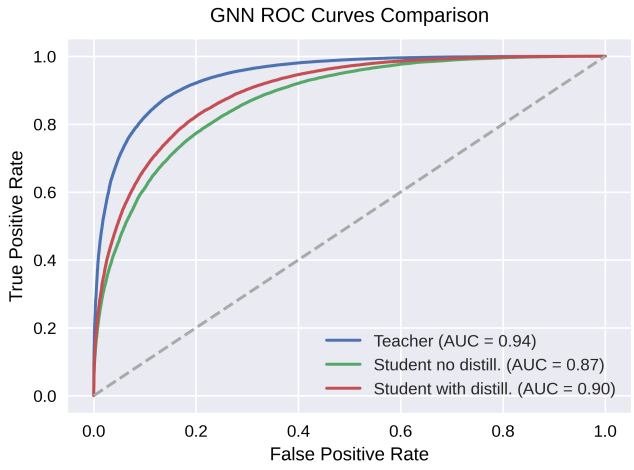

Figure 9: ROC curves for knowledge distillation between a large Teacher GNN and a more compact Student GNN, both trained on the original, undegraded data.

[1] David Griffiths. *Introduction to Elementary Particles*. Wiley-VCH, Weinheim, 2nd edition, 2010.

[2] The ATLAS Collaboration. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. 7 2012.

[3] Serguei Chatrchyan and et al. CMS Collaboration Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716:30–61, 2012.

[4] The CMS Collaboration and et al. The CMS experiment at the CERN LHC. *INSTITUTE OF PHYSICS PUBLISHING AND SISSA*, 8 2008.

[5] Oliver Brüning and Lucio Rossi. The High-Luminosity Large Hadron Collider. *Nature Reviews Physics*,

1(4):241–243, 3 2019.

[6] Huilin Qu and Loukas Gouskos. ParticleNet: Jet Tagging via Particle Clouds. 2019.

[7] Huilin Qu, Congqiao Li, and Sitian Qian. Particle Transformer for Jet Tagging. 2 2022.

[8] Taoli Cheng, Jean-François Arguin, Julien Leissner-Martin, Jacinthe Pilette, and Tobias Golling. Variational autoencoders for anomalous jet tagging. *Physical Review D*, 107(1):016002, 1 2023.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. 3 2015.

[10] ATLAS Collaboration. Constituent-Based Top-Quark Tagging with the ATLAS Detector. Technical report, CERN, 8 2022.