Large-Scale Pretraining and Finetuning for Efficient Jet Classification in Particle Physics

Zihan Zhao[1], **Farouk Mokhtar**[1], **Raghav Kansal**[1], **Haoyang (Bily) Li**[1], **and Javier Duarte**[1]

[1]University of California San Diego, La Jolla, CA 92093, USA

E-mail: ziz078@ucsd.edu, fmokhtar@ucsd.edu, rkansal@ucsd.edu, hal113@ucsd.edu, jduarte@ucsd.edu

Abstract. This study introduces an innovative approach to analyzing unlabeled data in high-energy physics (HEP) through the application of self-supervised learning (SSL). Faced with the increasing computational cost of producing high-quality labeled simulation samples at the CERN LHC, we propose leveraging large volumes of unlabeled data to overcome the limitations of supervised learning methods, which heavily rely on detailed labeled simulations. By pretraining models on these vast, mostly untapped datasets, we aim to learn generic representations that can be finetuned with smaller quantities of labeled data. Our methodology employs contrastive learning with augmentations on jet datasets to teach the model to recognize common representations of jets, addressing the unique challenges of LHC physics. Building on the groundwork laid by previous studies, our work demonstrates the critical ability of SSL to utilize large-scale unlabeled data effectively. We showcase the scalability and effectiveness of our models by gradually increasing the size of the pretraining dataset and assessing the resultant performance enhancements. Our results, obtained from experiments on two datasets—JetClass, representing unlabeled data, and Top Tagging, serving as labeled simulation data—show significant improvements in data efficiency, computational efficiency, and overall performance. These findings suggest that SSL can greatly enhance the adaptability of ML models to the HEP domain. This work opens new avenues for the use of unlabeled data in HEP and contributes to a better understanding the potential of SSL for scientific discovery.

1 Introduction

To enable precision measurements of the standard model (SM) and searches for new physics at the CERN LHC, physicists train machine learning (ML) models using detailed, labeled simulations of proton-proton collisions for a variety of tasks including triggering [1, 2], charged particle tracking, calorimetry [3], particle-flow reconstruction [4], and jet tagging [5, 6, 7] and mass regression. These trained ML models are subsequently applied to real data. This paradigm is called *supervised learning* because it uses explicit labels derived from simulation settings, e.g., whether a signal or background event is simulated.

However, a significant issue is ensuring that the performance of these ML models trained in simulation translates to real data, especially when the conditions of the two settings are different. For example, data often has a larger number of average simultaneous proton-proton interactions, known as *pileup* (PU), run-dependent detector noise, and different trigger efficiencies than simulation. This general problem of different conditions between the training setting and the inference setting has been studied in ML and particle physics and is known as *domain adaption* [8, 9]. Typically, these differences are corrected by applying data-simulation scale factors parameterized as a function of a small number of features. Nonetheless, in recent searches for high-momentum Higgs boson production [10, 11], this scale factor measurement and application can be a dominant source of systematic uncertainty.

In this paper, we propose to apply a generalized ML approach, in which models are first pretrained on large quantities of unlabeled data and subsequently *adapted* or *finetuned* using smaller quantities of labeled data for a specific downstream task. In the pretraining stage, models are trained to learn generic representations of the input features. These representations can then be used to train a smaller, simpler model in smaller quantities of simulation. Generally, pretraining is achieved through self-supervised learning (SSL), in which portions of input data are masked, paired together, or augmented and the model is tasked to reconstruct the missing data, find matching pairs, or identify data augmentations, respectively. This forces the model to learn the context of and correlations among elements in the data. The specific approach we take in this paper is contrastive learning, in which multiple views of the same data are input to the model, and the model learns common representation. In our case, the algorithm's inputs are the original jets and their augmented versions.

There are two primary objectives of this approach:

1. To demonstrate gains in performance in the smaller labeled dataset by pretraining on larger unlabeled dataset.

2. To demonstrate the effect of scaling up the sizes of the large unlabeled dataset on the performance of SSL models.

By applying this strategy to particle physics, we can simultaneously (1) leverage copious amounts of unlabeled real data for training, (2) ensure that model performance transfers between the two domains of data and simulation seamlessly, and (3) potentially build more generalizable, transferable, and powerful ML models that accelerate scientific discovery.

This framework, sometimes referred to as the foundation model (FM) [12] approach, has powered major advances in natural language processing, computer vision, audio/video processing, and learning across these different modalities of data. These FMs, e.g., GPT-3 [13], are often large with up to $\mathcal{O}(10^9)$ parameters, trained on massive unlabeled data sets, serve as powerful backbones that extract key information from data, and can be adapted for different downstream tasks.

Related work includes using self-supervision for jet tagging [14], resimulation based SSL [15], masked particle modeling [16], and generative pretraining [17]. These studies have laid the groundwork for developing foundation models tailored to the unique challenges of LHC physics, highlighting the potential of various pretraining techniques. However, to fully harness the capabilities of a foundation model, it is crucial to demonstrate its ability to utilize the vast, mostly untapped amounts of unlabeled data produced by the LHC. Our study focuses on this critical aspect by progressively increasing the size of the pretraining dataset and assessing the resultant performance enhancements, thereby underscoring the model's scalability and effectiveness in leveraging large-scale unlabeled data. Our software is available at `https://github.com/JavierZhao/JetCLR/tree/JetClass`.

This paper is organized as follows. Section 2 describes the chosen datasets and how we use them to represent real data and simulation, respectively. Section 3 describes the pretraining objective and training process we employ. Section 4 discusses our finetuning stage. The results are presented in Section 5. Finally, Section 6 provides a summary and outlook.

2 Datasets and experimental setup

2.1 Datasets

In this study, we used two datasets: JetClass [7, 18], a large dataset with 100 million jets that stands in for unlabeled data, and Top Tagging [19, 20], a smaller dataset with 1.2 million jets that stands in for labeled simulation.

The JetClass dataset contains ten distinct classes of jets, each corresponding to a unique particle decay channel, making it suitable to serve as a pretraining dataset. The background jets are initiated by gluons and quarks (q/g). The signal classes include five decay channels of the Higgs boson ($H \to b\bar{b}$, $H \to c\bar{c}$, $H \to gg$, $H \to 4q$, and $H \to \ell\nu qq'$); two decay channels of the top quark ($t \to bqq'$ and $t \to b\ell\nu$); and the W and Z bosons decaying into a pair of quarks ($W \to qq'$ and $Z \to q\bar{q}$).

The Top Tagging dataset only contains two classes of jets: top quark jets and QCD background jets, making it suitable for a downstream classification task.

2.2 Experiment setup

For both the JetClass dataset and the Top Tagging dataset, we used six particle-level kinematic features as inputs, described in Table 1. We use the "raw" η and ϕ instead of the relative $\Delta\eta$ and $\Delta\phi$. The reason is that as part of our contrastive learning approach, we augment the jets by translating the η and ϕ coordinates of the individual particles. The relative $\Delta\eta$ and $\Delta\phi$ would be invariant under this augmentation because all particles would have been shifted by the same amount.

Variable	Definition
η	pseudorapidity
ϕ	azimuthal angle
$\log p_\mathrm{T}$	logarithm of the particle's transverse momentum p_T
$\log E$	logarithm of the particle's energy
$\log(p_\mathrm{T}/p_\mathrm{T}^\mathrm{jet})$	logarithm of the particle's p_T relative to the jet p_T
$\log(E/E^\mathrm{jet})$	logarithm of the particle's energy relative to the jet energy

Table 1: Particle input kinematic features used in training [7]

3 Self-supervised pretraining

Our pretraining strategy draws on the innovative framework of JetCLR [14], which introduces a contrastive learning paradigm specifically tailored for jet tagging. Instead of using only 3 particle input features $(p_\mathrm{T}, \eta, \phi)$ as in JetCLR, we took advantage of more features, as described in Section 2.2.

Central to the JetCLR approach, and by extension to our methodology, are the concepts of alignment and uniformity: bringing representations of similar jets—both augmented and original—closer together while distancing those of dissimilar jets.

3.1 Augmentations

To generate the positive pairs, we employed the same augmentation techniques as those detailed in the JetCLR framework [14]. These augmentations are visualized in Fig. 1, which shows the angular distribution of jet particles, with the size of each marker being indicative of the particle's transverse momentum.

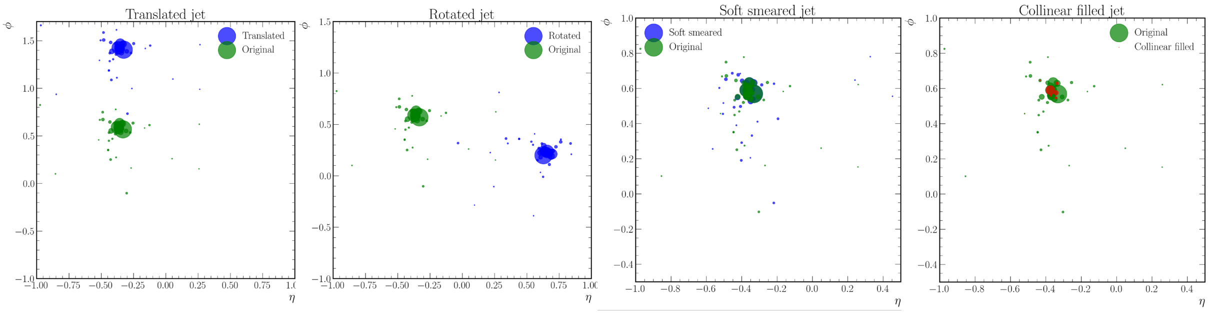

Figure 1: Illustration of the augmentations

3.2 Training details

To encode the input jets into representations, we used a transformer encoder [21] with hyperparameters specified in Table 2.

In order to efficiently pretrain on a large number of jets, we made several optimizations to the existing code base from [14] to speed up the pretraining. First, we removed unnecessary CPU-GPU synchronizations, especially read-out from GPU for the purpose of recording losses. In our optimized code base, this is done asynchronously to avoid synchronization barriers. We also modified the default model dimensions to be multiples of 8 in order to make use of CUDA matrix multiplication kernels more efficiently. In addition, we fused point-wise operations into a single CUDA kernel when computing the contrastive loss.

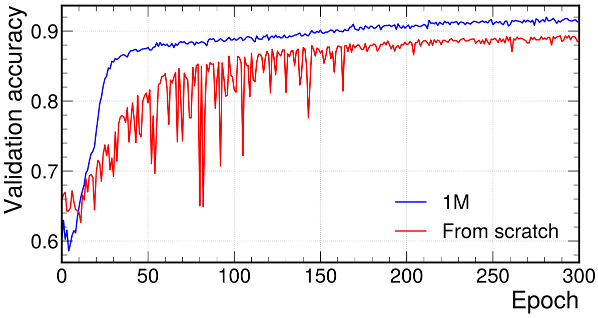

Figure 2: Comparison of the training performance of the pretrained model and the model trained from scratch

Last but not least, we utilized the Automatic Mixed Precision (AMP) package from PyTorch [22] to run certain parts of pretraining with lower precision, reducing memory usage and improving computational speed without significantly impacting the accuracy of results. We used gradient clipping with a maximum norm of 0.1 and set the ϵ parameter to 10^{-4} in the Adam optimizer, in order to mitigate the numerical instability issues caused by the use of AMP. These settings help control the scale of gradients, preventing them from becoming too large, which can lead to numerical instability during training. These optimizations have improved our training speed significantly, by a factor of 3, compared with the original JetCLR implementation, with little impact on performance.

Hyperparameter	Value
model (embedding) dimension	1024
feed-forward hidden dimension	1024
output dimension	1024
Self-attention heads	4
Transformer layers	4
Layers	2
dropout rate	0.1

Table 2: Hyperparameters of the transformer encoder.

4 Supervised finetuning

In the finetuning phase, we aim to preserve the rich, pretrained representations by appending only a single linear layer to the encoder, as opposed to a more elaborate multi-layer perceptron (MLP). The rationale behind this decision was to prevent a sophisticated classifier from diluting the robust features that had been learned during the pretraining stage.

During finetuning, we do not freeze the transformer encoder: both the encoder and the newly added linear layer are subject to training. This allows for the fine adjustment of the features, ensuring that the encoder representations can adapt to the nuances of the task at hand while leveraging the generalizable knowledge acquired during pretraining. This guides the model to maintain its powerful representational ability while also becoming more specialized for the targeted application.

5 Results

5.1 Transfer learning: From JetClass to Top Tagging

To demonstrate the generalizability and transferability of the features learned during the pretraining state, we pretrained a model on 1 million unlabeled jets from the JetClass dataset and then finetuned it on a certain number of labeled jets from the Top Tagging dataset. Fig 2 shows that despite the limited data, the pretrained model (blue curve) achieves higher accuracy and converges faster than the one trained from scratch (red curve).

To further quantify the advantages of pretraining, we compared the performances of the pretrained model against that of a counterpart initialized without pretraining. Figures 3 and 4 show the accuracy and rejection rates plotted against varying quantities of labeled training data, which show that the pretrained model consistently outperforms its from-scratch-trained counterpart. This is apparent not only in terms of accuracy but also in the efficiency of rejecting false positives. As the volume of finetuning data increases, an improvement is noted for both models. However, the pretrained model achieves superior performance with a markedly smaller dataset—a clear testament to the pretraining phase's value.

Additionally, the pretrained model demonstrates faster convergence, which is highlighted in Fig. 5 comparing the number of epochs required to approach the final accuracy within a 1% margin. Illustrated by the blue line in the graph, the pretrained model reaches the convergence threshold in fewer epochs over the entire span of labeled data presented, suggesting an optimized training that capitalizes on the learned features during pretraining to accelerate overall convergence.

5.2 Demonstrating the power of dataset scaling through pretraining on different amounts of jets

To demonstrate the effectiveness of dataset scaling, we progressively increased the size of our pretraining dataset from 1 million jets to 5 million and finally to 10 million jets. We then analyzed the outcomes of models pretrained on these varying dataset sizes. As illustrated in Figs. 3 and 4, the results show a clear trend: larger pretraining datasets significantly enhance model performance in terms of both rejection power and accuracy. Here, 'rejection power' refers to the model's ability to accurately reject background samples while correctly identifying signal samples. In addition to enhancing model performance, larger pretraining datasets also allow the model to converge more quickly during finetuning. As depicted in Fig. 5, as the sizes of the pretraining dataset grows, the number of epochs required to reach the final accuracy with a 1% margin decreases across the range of labeled training samples used for finetuning. These results demonstrate the feasibility of a foundation model that leverages the vast amounts of unlabeled data produced by the LHC. This approach highlights the potential of using extensive, unlabeled datasets to enhance model capabilities significantly. By training on increasingly larger datasets, we can tap into deeper insights and detect subtle patterns that are not apparent in smaller datasets.

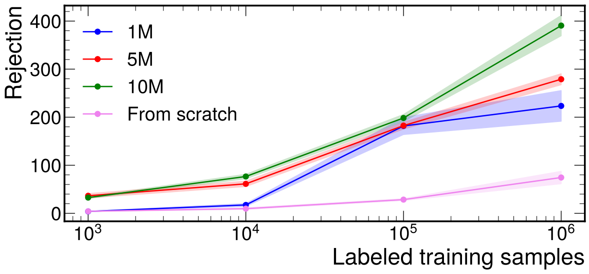

Figure 3: Background rejection at 50% signal efficiency of different training strategies as a function of labeled training samples

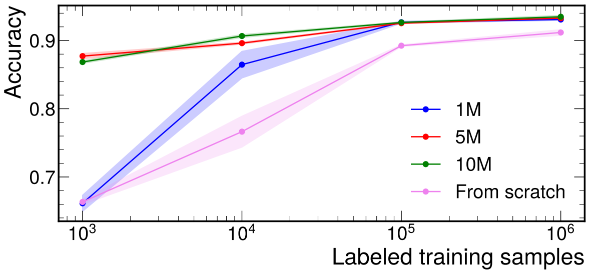

Figure 4: Accuracy of different training strategies as a function of labeled training samples

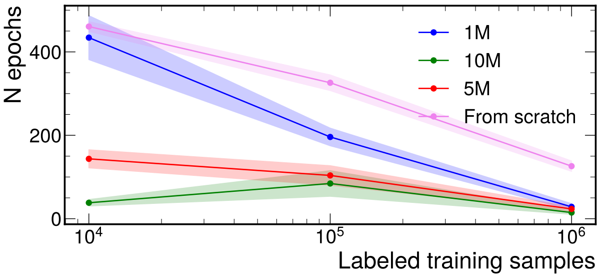

Figure 5: Number of epochs required to reach within 1% of the final accuracy of different training strategies as a function of labeled training samples

6 Summary and outlook

In our study, we were able to show that

1. Through contrastive learning, a vanilla transformer encoder was able to learn useful representations of jets from unlabeled data.
2. By pretraining on unlabeled data, the transformer encoder was able to learn the downstream task **faster** and with **fewer labeled training samples**, compared with one we trained from scratch.
3. By scaling up the pretraining dataset, the model demonstrated enhanced performance and faster convergence, achieving higher accuracy and rejection rates with increased volumes of unlabeled data.

Through large-scale pretraining followed by finetuning, our SSL approach has demonstrated **enhanced data efficiency**—requiring fewer labeled training samples to achieve superior performance compared to the fully supervised approach. Additionally, it offers **greater computational efficiency** by enabling the model to converge significantly faster than its fully supervised counterpart. This paves the way for the use of unlabeled data in HEP and contributes to a better understanding the potential of SSL for scientific discovery.

7 Acknowledgements

This work was supported by the U.S. Department of Energy (DOE), Office of Science, Office of High Energy Physics Early Career Research program under Award No. DE-SC0021187, the DOE, Office of Advanced Scientific Computing Research under Award No. DE-SC0021396 (FAIR4HEP), and the U.S. National Science Foundation (NSF) Harnessing the Data Revolution (HDR) Institute for Accelerating AI Algorithms for Data Driven Discovery (A3D3) under Cooperative Agreement OAC-2117997. This work was performed using the Pacific Research Platform Nautilus HyperCluster supported by NSF awards CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the University of California San Diego's California Institute for Telecommunications and Information Technology/Qualcomm Institute. Thanks to CENIC for the 100 Gpbs networks.

References

[1] J. Duarte et al., "Fast inference of deep neural networks in FPGAs for particle physics", *J. Instrum.* **13** (2018) P07027, `doi:10.1088/1748-0221/13/07/P07027`, `arXiv:1804.06913`.

[2] CMS Collaboration, "The Phase-2 upgrade of the CMS Level-1 trigger", CMS Technical Design Report CERN-LHCC-2020-004. CMS-TDR-021, CERN, 2020. https://cds.cern.ch/record/2714892.

[3] S. Bhattacharya et al., "GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter", in *20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI Decoded - Towards Sustainable, Diverse, Performant and Effective Scientific Computing.* 3, 2022. `arXiv:2203.01189`.

[4] J. Pata et al., "Machine Learning for Particle Flow Reconstruction at CMS", in *20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI Decoded - Towards Sustainable, Diverse, Performant and Effective Scientific Computing.* 2022. `arXiv:2203.00330`.

[5] E. A. Moreno et al., "Interaction networks for the identification of boosted $H \to b\bar{b}$ decays", *Phys. Rev. D* **102** (2020) 012010, `doi:10.1103/PhysRevD.102.012010`, `arXiv:1909.12285`.

[6] H. Qu and L. Gouskos, "ParticleNet: Jet tagging via particle clouds", *Phys. Rev. D* **101** (2020), no. 5, 056019, `doi:10.1103/PhysRevD.101.056019`, `arXiv:1902.08570`.

[7] H. Qu, C. Li, and S. Qian, "Particle Transformer for Jet Tagging", in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri et al., eds., volume 162 of *Proceedings of Machine Learning Research*, p. 18281. 2022. `arXiv:2202.03772`.

[8] J. M. Clavijo, P. Glaysher, J. Jitsev, and J. M. Katzy, "Adversarial domain adaptation to reduce sample bias of a high energy physics event classifier *", *Mach. Learn. Sci. Tech.* **3** (2022), no. 1, 015014, `doi:10.1088/2632-2153/ac3dde`, `arXiv:2005.00568`.

[9] M. Baalouch, M. Defurne, J.-P. Poli, and N. Cherrier, “Sim-to-Real Domain Adaptation For High Energy Physics”, in *33rd Annual Conference on Neural Information Processing Systems.* 12, 2019. arXiv:1912.08001.

[10] CMS Collaboration, “A portrait of the Higgs boson by the CMS experiment ten years after the discovery”, *Nature* **607** (2022) 60, doi:10.1038/s41586-022-04892-x, arXiv:2207.00043.

[11] CMS Collaboration, “Search for nonresonant pair production of highly energetic Higgs bosons decaying to bottom quarks”, arXiv:2205.06667. Accepted by *Phys. Rev. Lett.*

[12] R. Bommasani et al., “On the opportunities and risks of foundation models”, arXiv:2108.07258.

[13] T. Brown et al., “Language models are few-shot learners”, in *Advances in Neural Information Processing Systems*, H. Larochelle et al., eds., volume 33, p. 1877. Curran Associates, Inc., 2020. arXiv:2005.14165.

[14] B. M. Dillon et al., “Symmetries, safety, and self-supervision”, *SciPost Phys.* **12** (2022), no. 6, 188, doi:10.21468/SciPostPhys.12.6.188, arXiv:2108.04253.

[15] P. Harris et al., “Re-Simulation-based Self-Supervised Learning for Pre-Training Foundation Models”, 3, 2024. arXiv:2403.07066.

[16] T. Golling et al., “Masked Particle Modeling on Sets: Towards Self-Supervised High Energy Physics Foundation Models”, 1, 2024. arXiv:2401.13537.

[17] J. Birk, A. Hallin, and G. Kasieczka, “OmniJet-α: The first cross-task foundation model for particle physics”, 3, 2024. arXiv:2403.05618.

[18] H. Qu, C. Li, and S. Qian, “JetClass: A Large-Scale Dataset for Deep Learning in Jet Physics”, 2022. doi:10.5281/zenodo.6619768.

[19] G. Kasieczka et al., “The LHC Olympics 2020: A community challenge for anomaly detection in high energy physics”, *Rep. Prog. Phys.* **84** (2021) 124201, doi:10.1088/1361-6633/ac36b9, arXiv:2101.08320.

[20] G. Kasieczka, T. Plehn, J. Thompson, and M. Russel, “Top Quark Tagging Reference Dataset (v0 (2018_03_27))”. [Data set]. Zenodo, 2019. doi:10.5281/zenodo.2603256.

[21] A. Vaswani et al., “Attention is all you need”, arXiv:1706.03762.

[22] A. Paszke et al., “Pytorch: An imperative style, high-performance deep learning library”, in *Advances in Neural Information Processing Systems 32*, H. Wallach et al., eds., pp. 8024–8035. Curran Associates, Inc., 2019.