Learning Symmetry-Independent Jet Representation via Jet Joint Embedding Predictive Architecture

Haoyang "Billy" Li†, Subash Katel†, **Zihan Zhao**†, Farouk Mokhtar, Javier Duarte (UCSD)
Raghav Kansal (Caltech)

Larger than Larger Ep1 2025
Jan 7

†: equal contribution

https://arxiv.org/abs/2412.05333

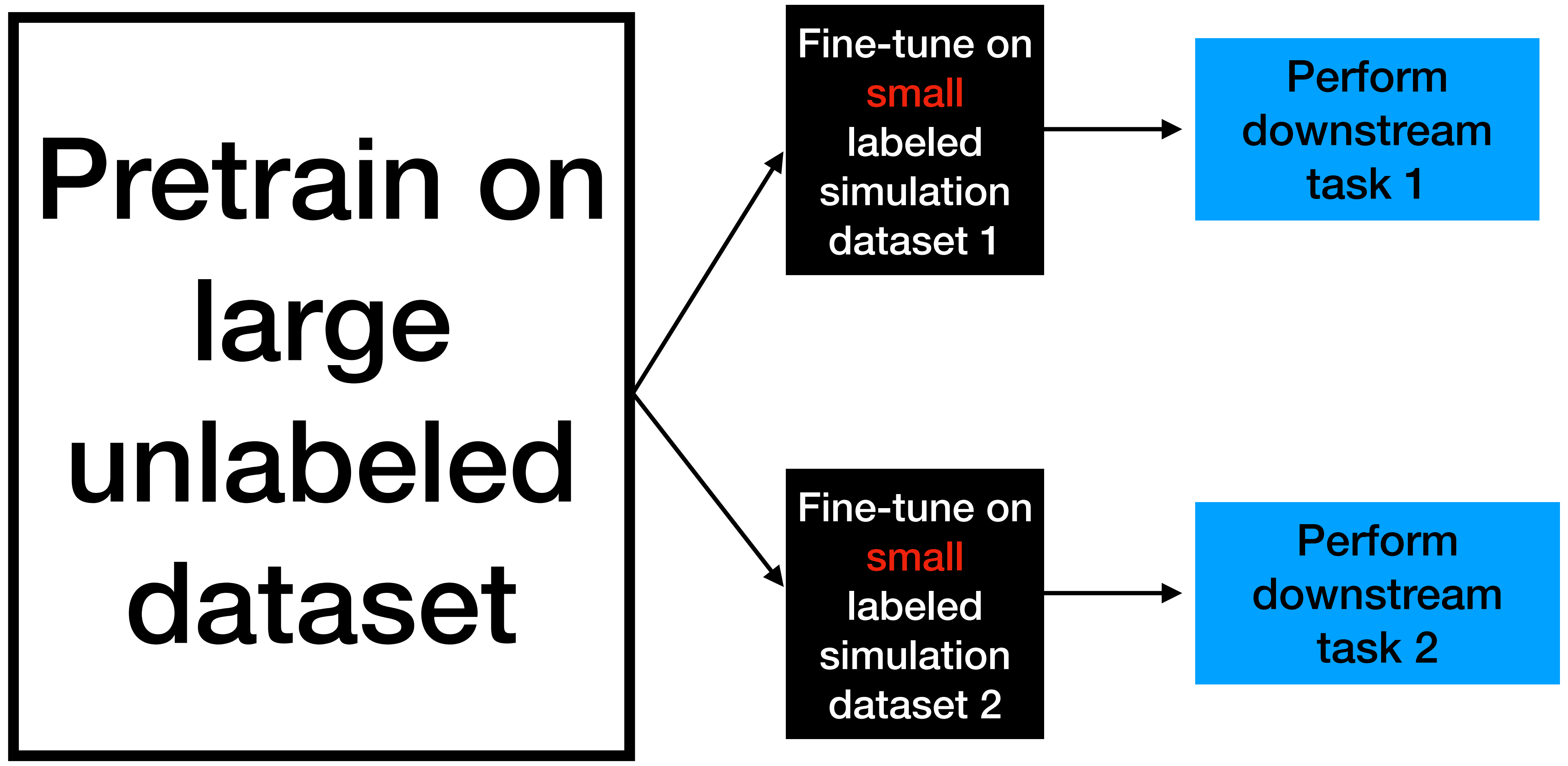
Outline

- Motivation
- Introduction to JEPA
- **Our J-JEPA approach**
- Dataset
- Pretraining + fintuning setup
- Pretraining result
- Pretraining + fine-tuning result
- Ongoing and Future work

Motivations for Self-Supervised Learning (SSL)

Learning without labels

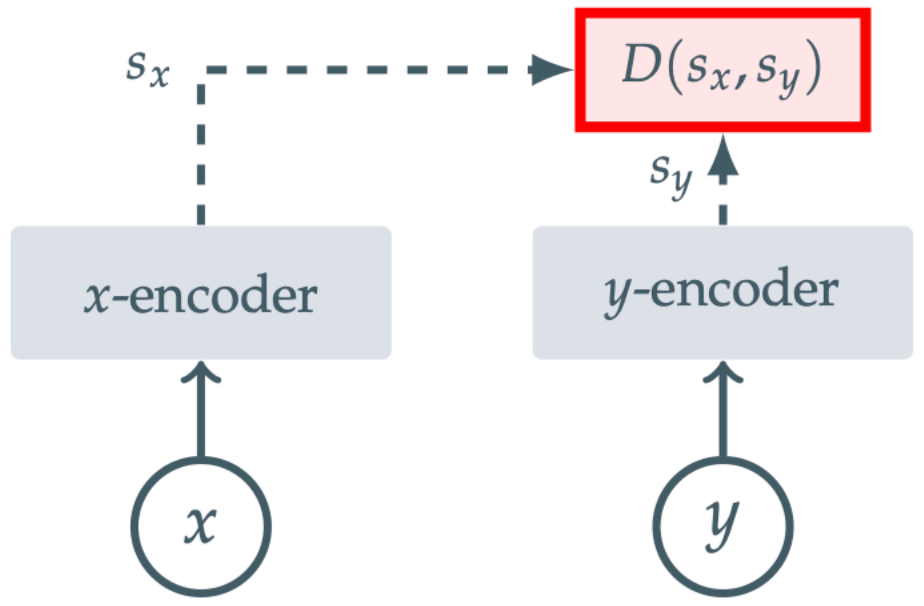
- Self-Supervised Learning: A type of machine learning where models learn useful features and representations from unlabeled data
- To learn effectively (like human), system must learn these representations directly from unlabeled data such as images or sounds, rather than from manually assembled labeled datasets.
- With the HL-LHC upgrade [1] in the near future, we will need to simulate an order of magnitude more events with a more complicated detector geometry to keep up with the recorded data [2].



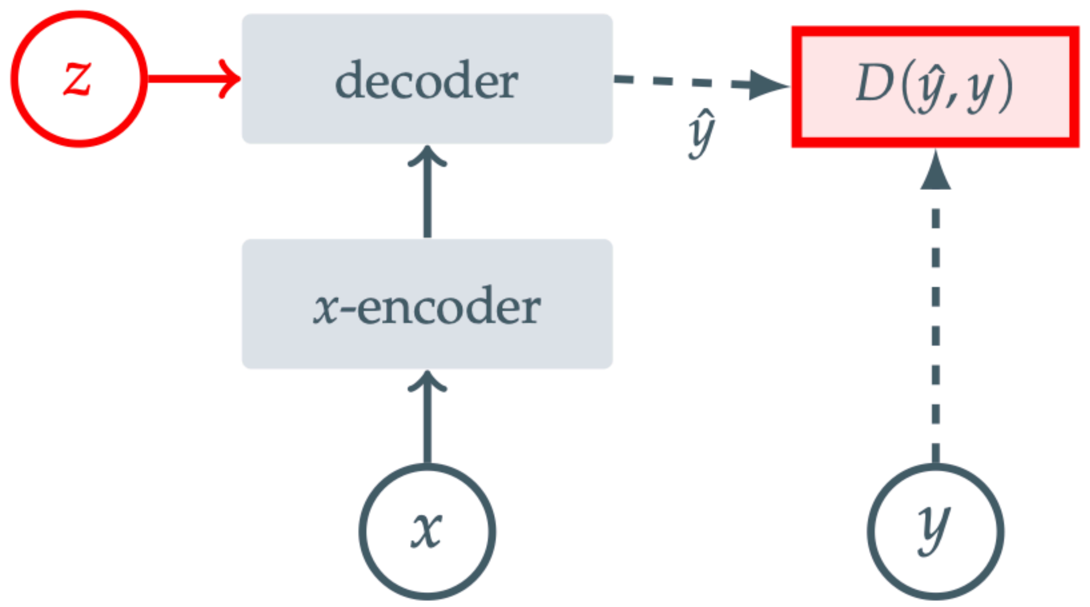
SSL for foundation model

1. [HL-LHC] https://arxiv.org/abs/1705.08830
2. [Computing for HL LHC] https://doi.org/10.1051/epjconf/201921402036

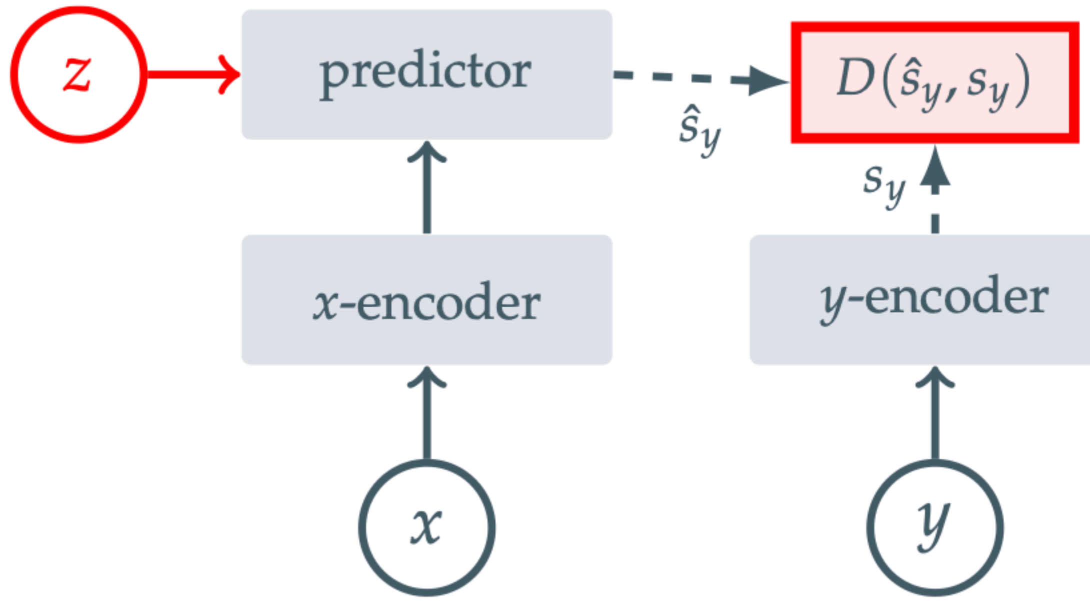
JEPA: Different SSL Architectures



(a) **Joint-Embedding Architecture**

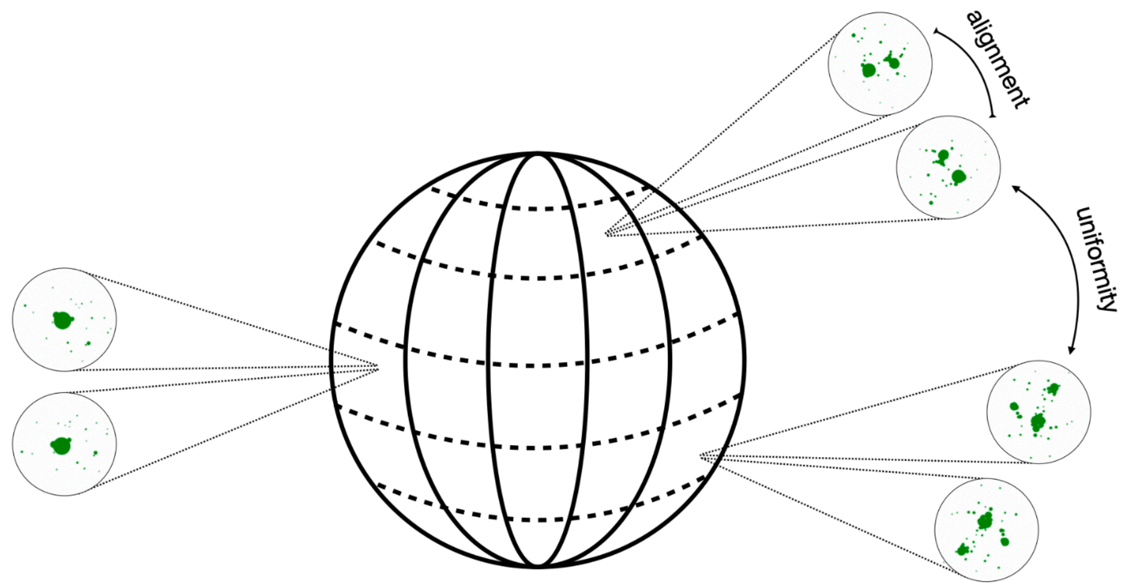


(b) **Generative Architecture**



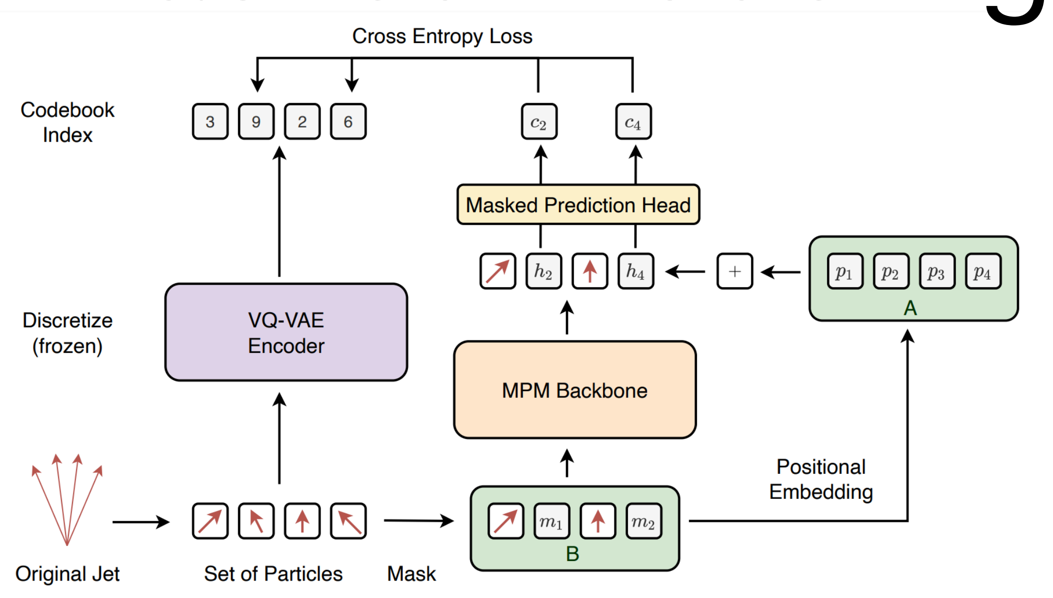
(c) **Joint-Embedding Predictive Architecture**

Contrastive Learning



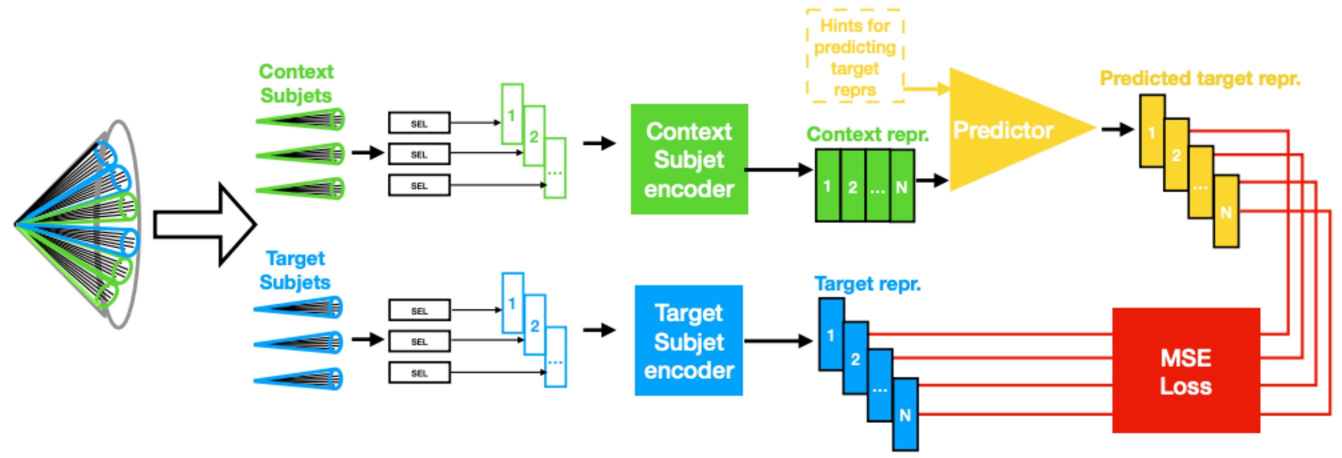
https://arxiv.org/abs/2108.04253

Masked Modeling



https://arxiv.org/abs/2401.13537

Our Work

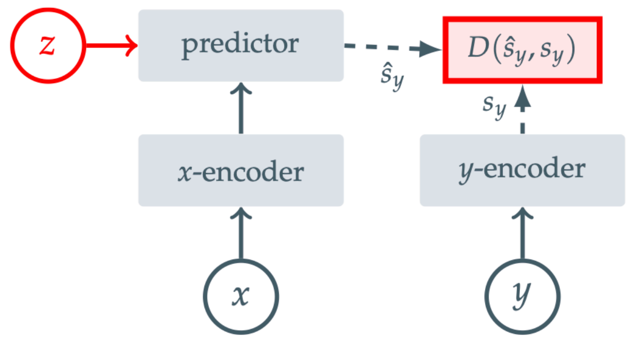


https://arxiv.org/abs/2412.05333

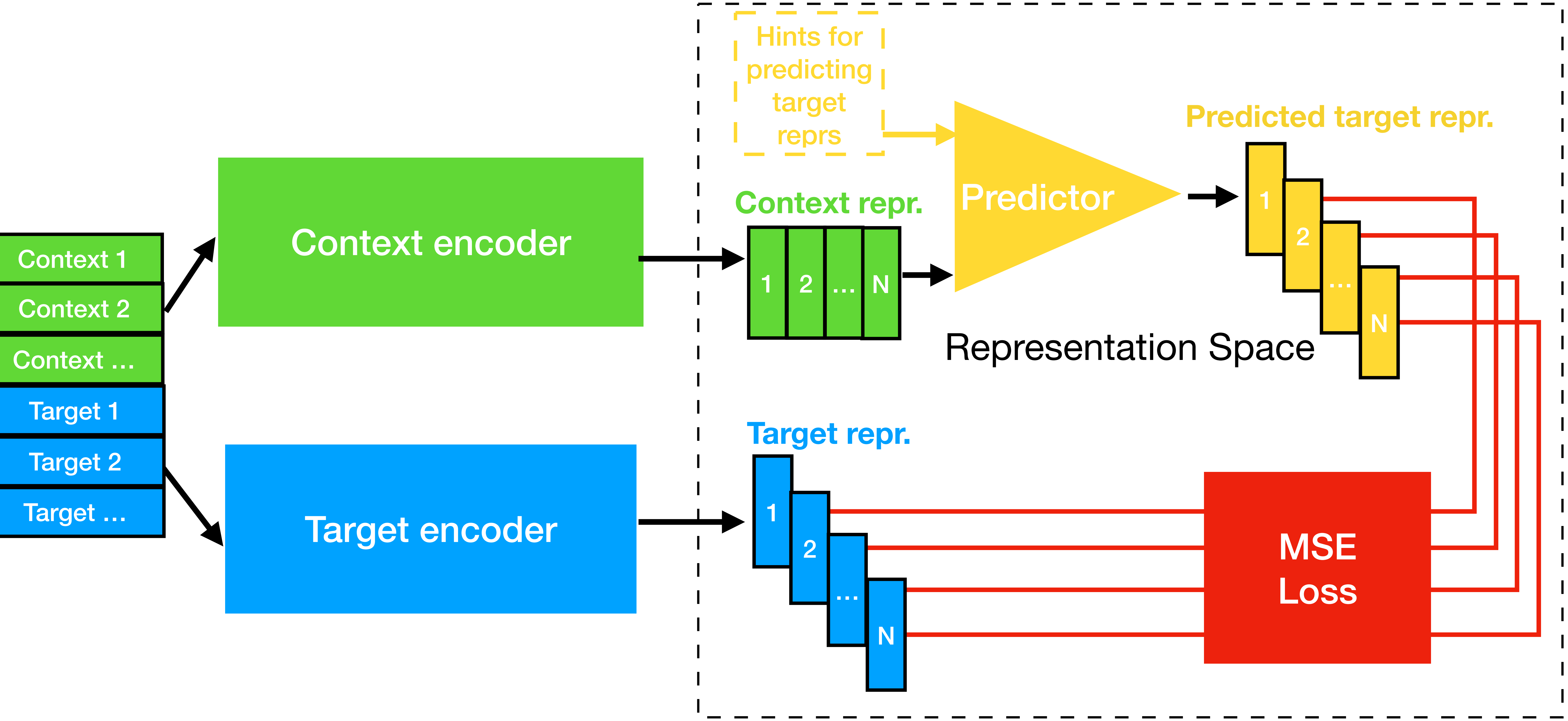
- Difference between JEPA and (a): JEPA is augmentation free and predictive
- Difference between JEPA and (b): JEPA predicts in the latent space and does not mask the input

Assran et al., “Self-supervised learning from images with a joint-embedding predictive architecture”, 2023.

JEPA: Joint Embedding Predictive Architecture



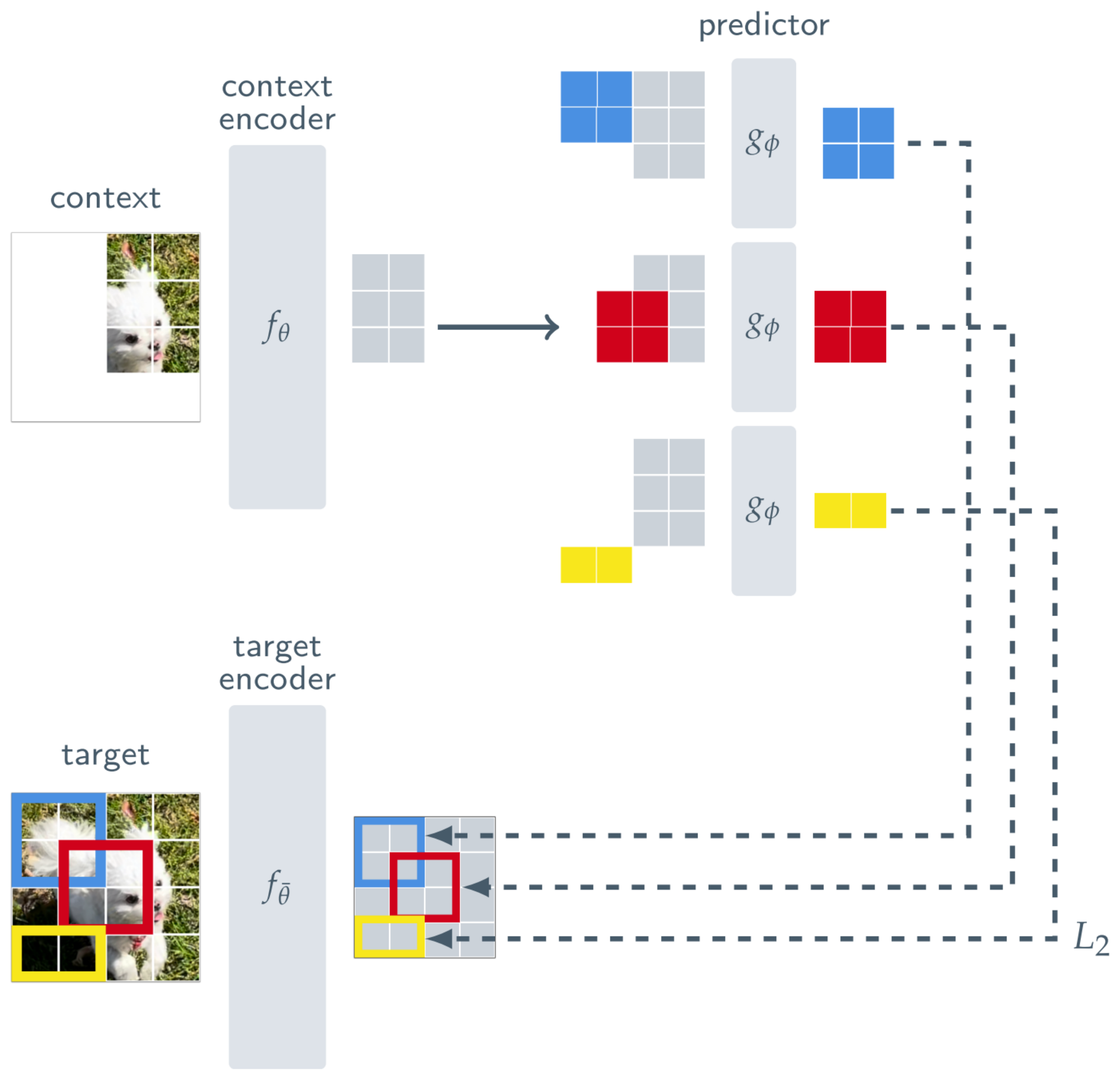
(c) Joint-Embedding Predictive Architecture



- Predict the masked parts in the representation space
- Augmentation free to minimize bias

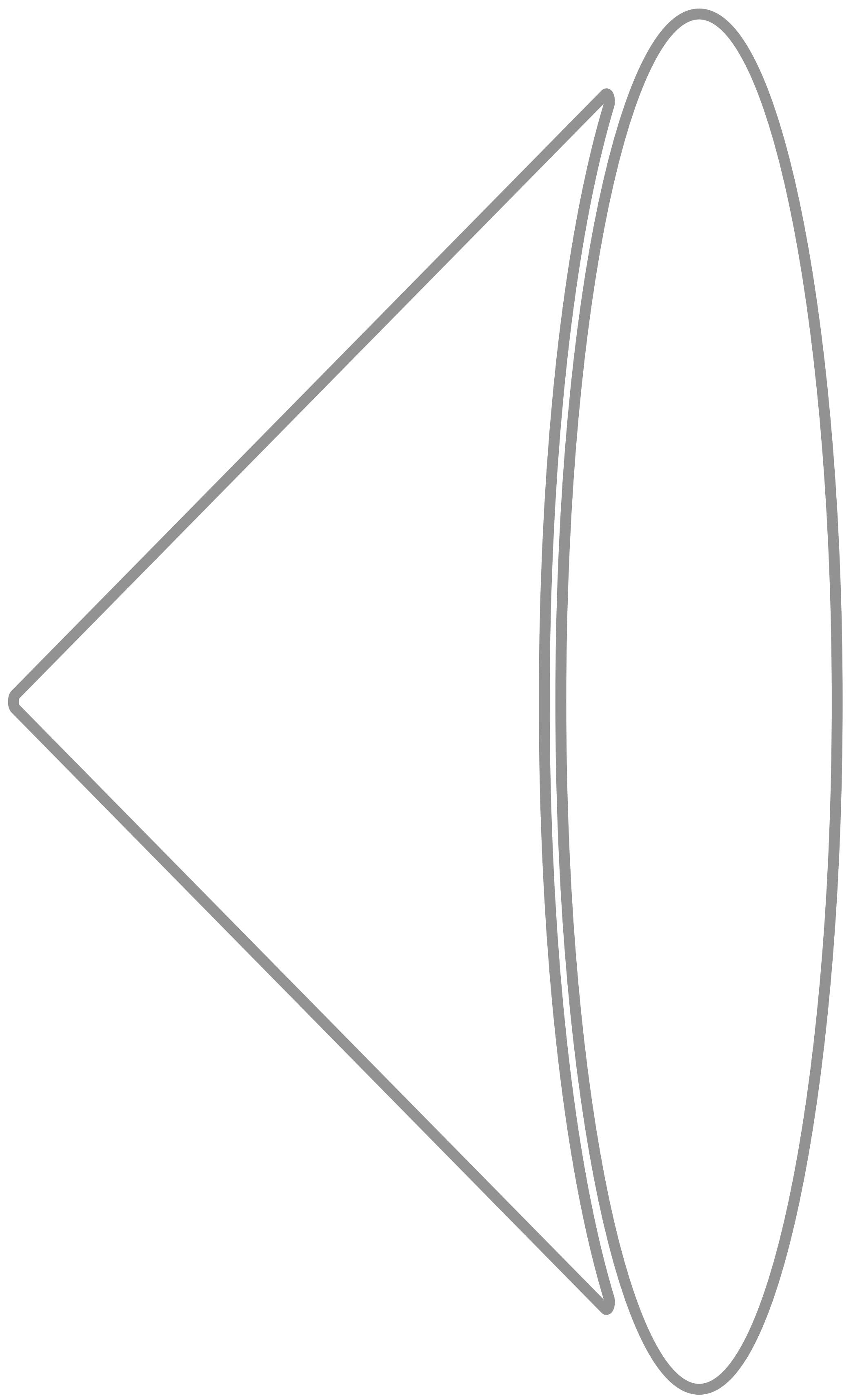
Example: The I-JEPA Architecture

I: Image

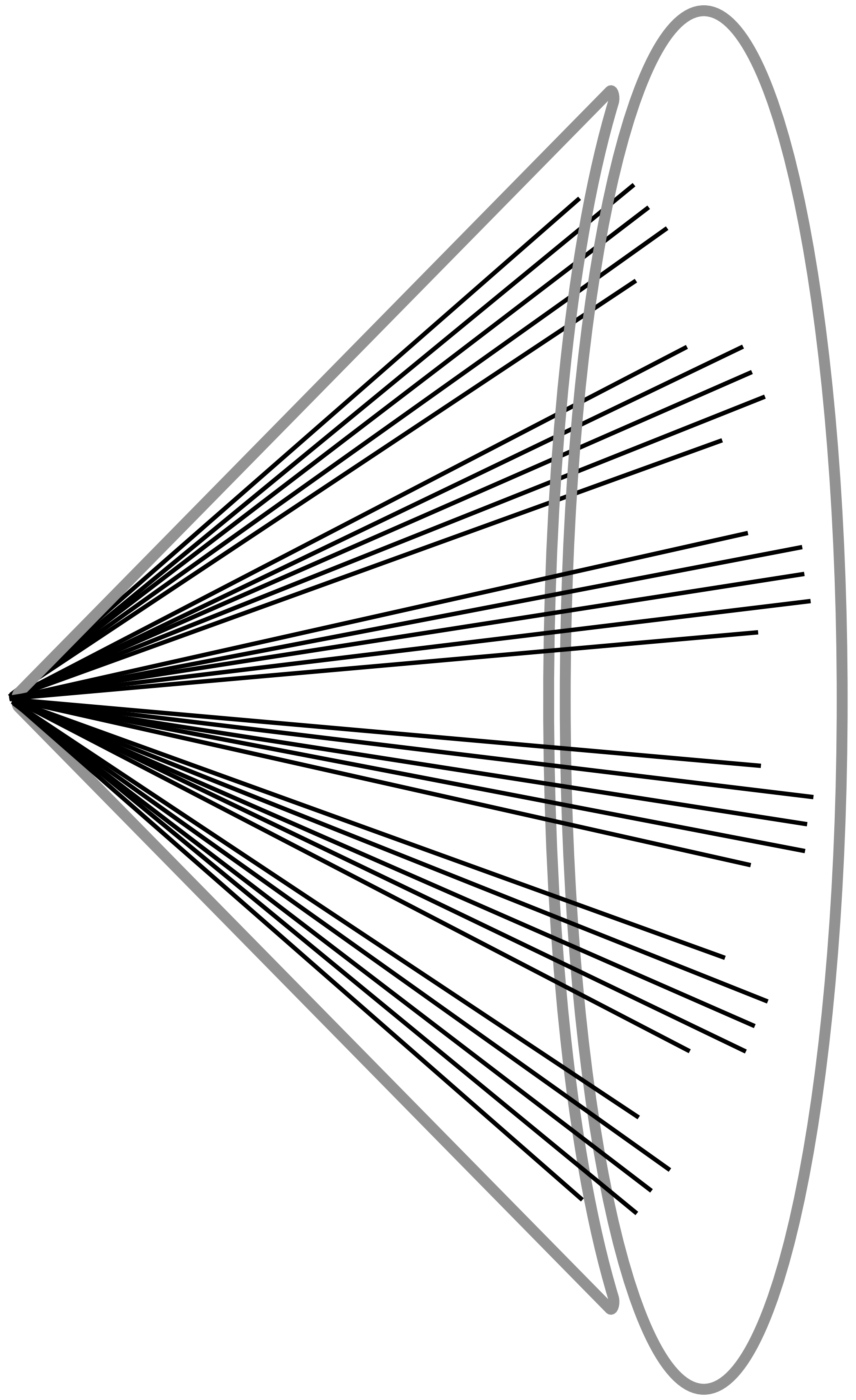


J (Jet) - JEPA

An AK8 Jet



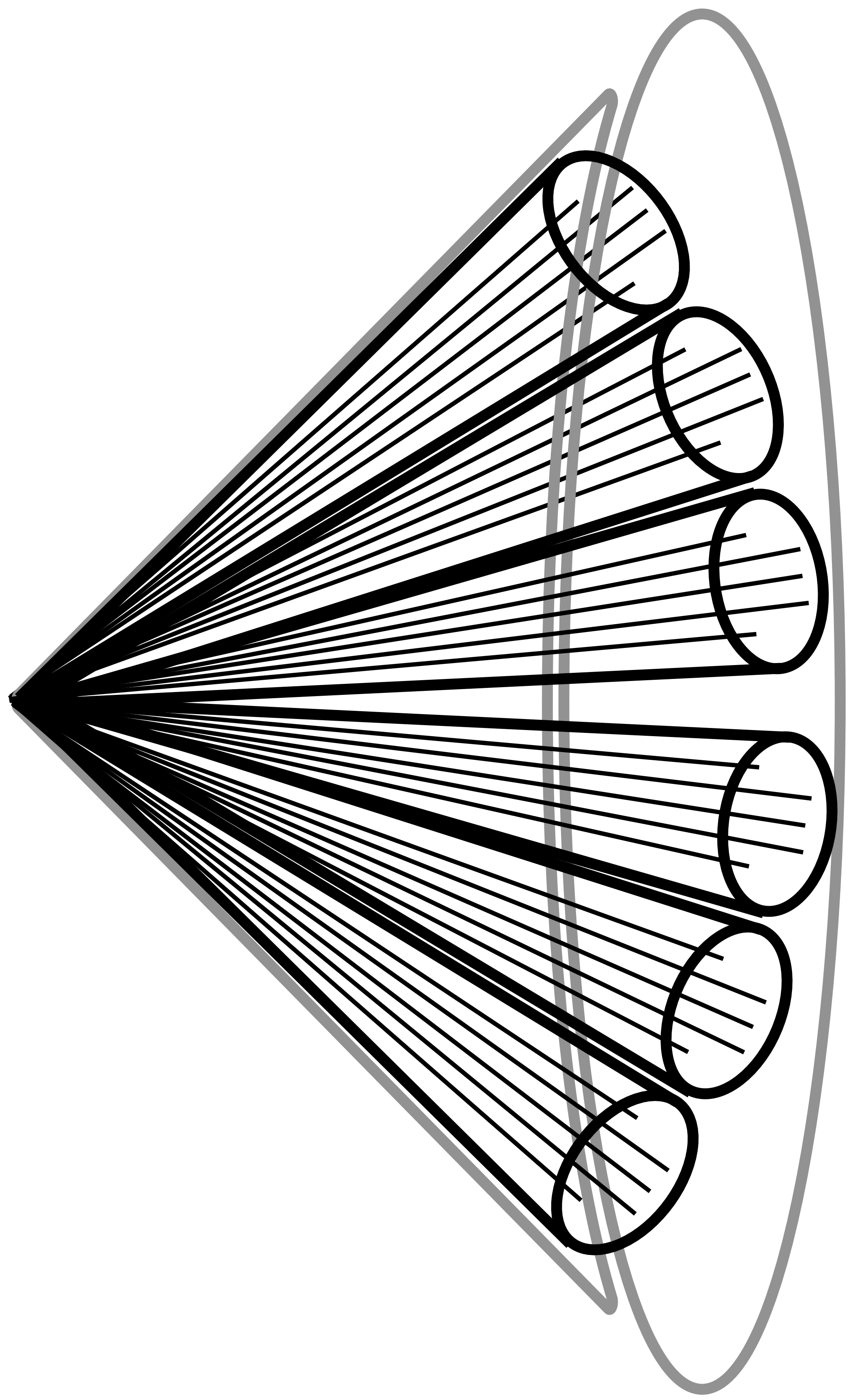
An AK8 Jet



J-JEPA

Cluster subjets with radius 0.2

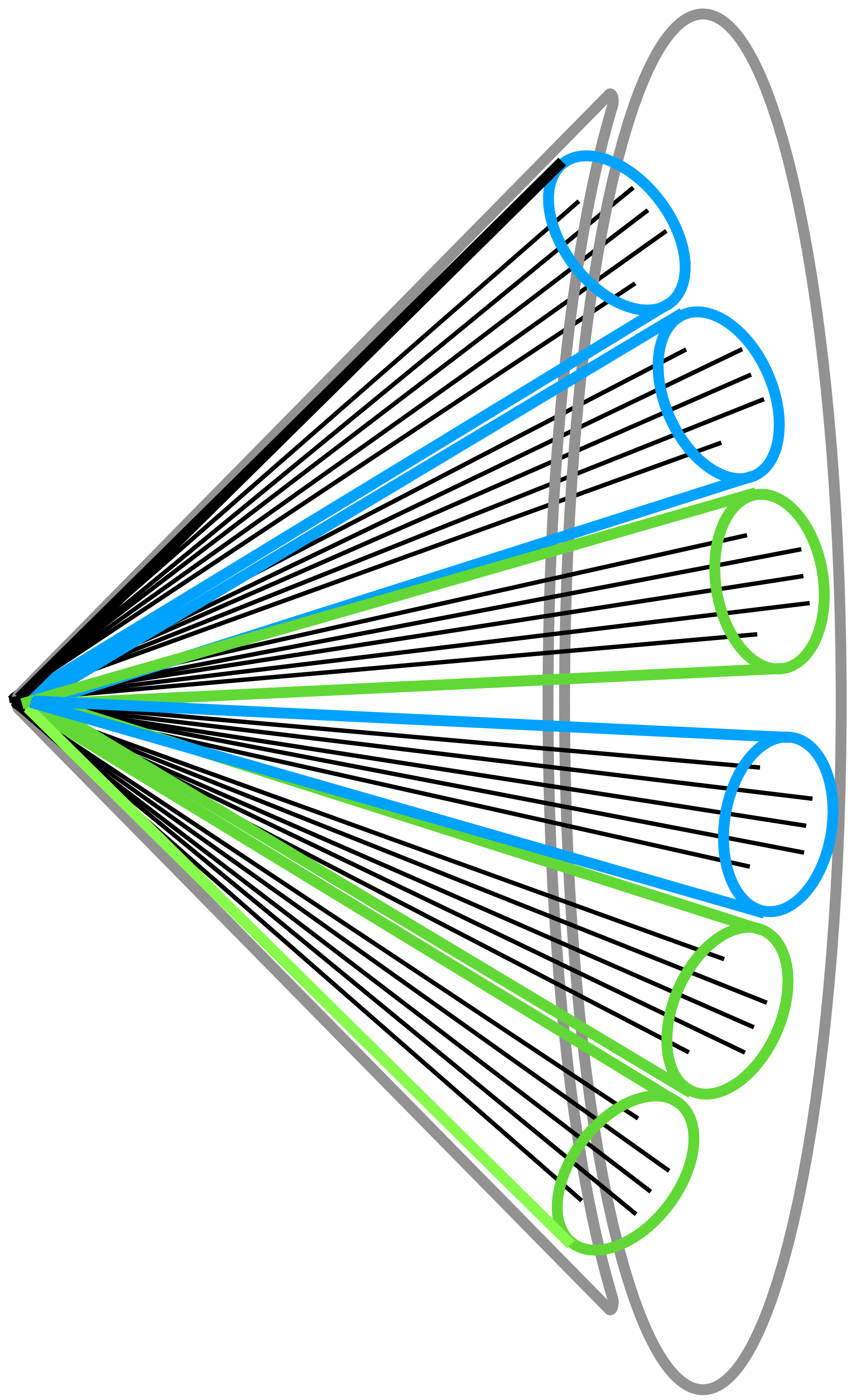
An AK8 Jet



J-JEPA: Define Target and Context Subjets

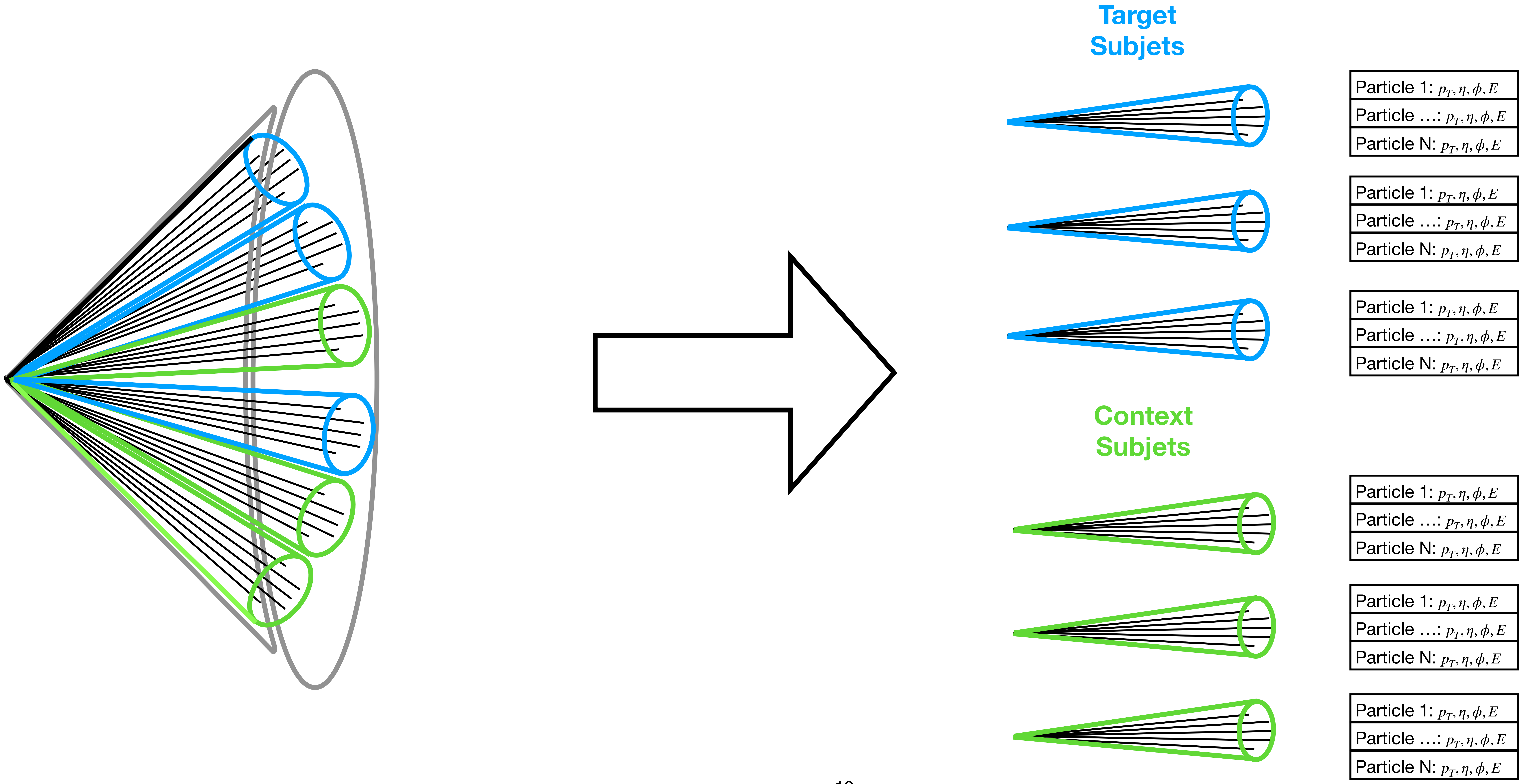
Randomly divide subjets into target/context categories

An AK8 Jet



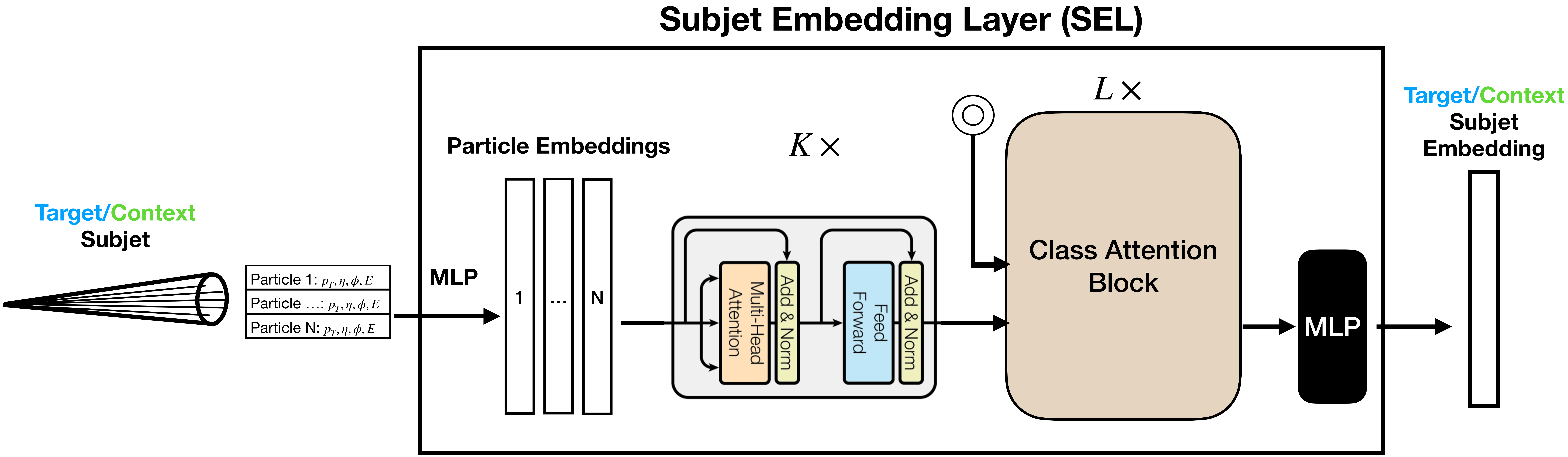
J-JEPA: Define Target and Context Subjets

Randomly divide subjets into target/context categories

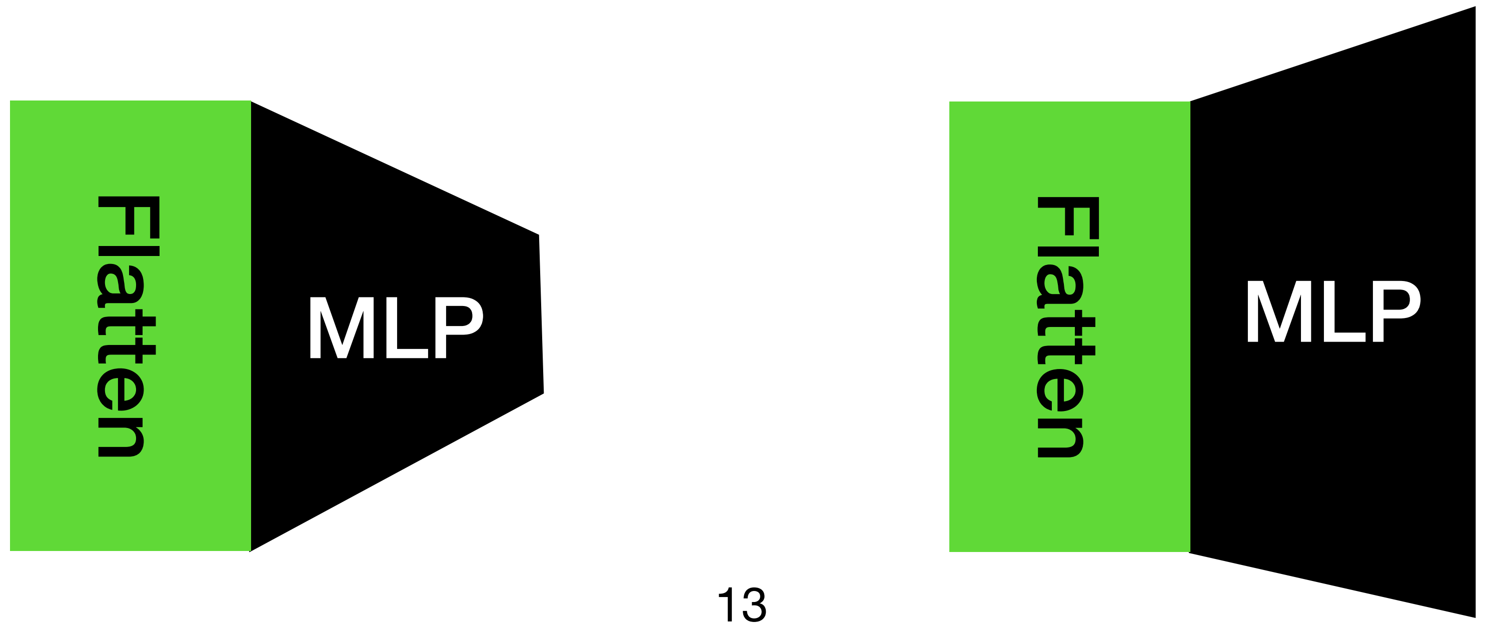


J-JEPA: Subjet Embedding Layer (SEL)

Each subjet creates its embedding independently

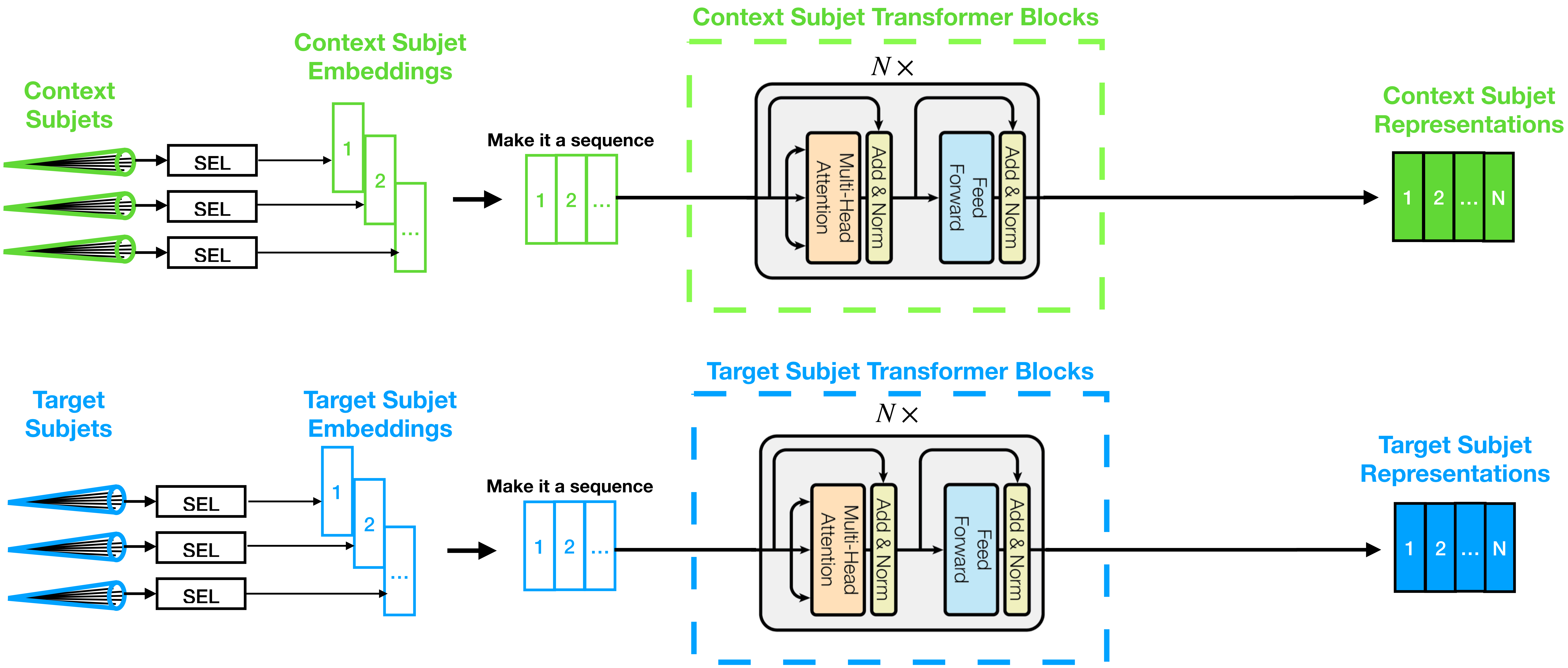


Other options:



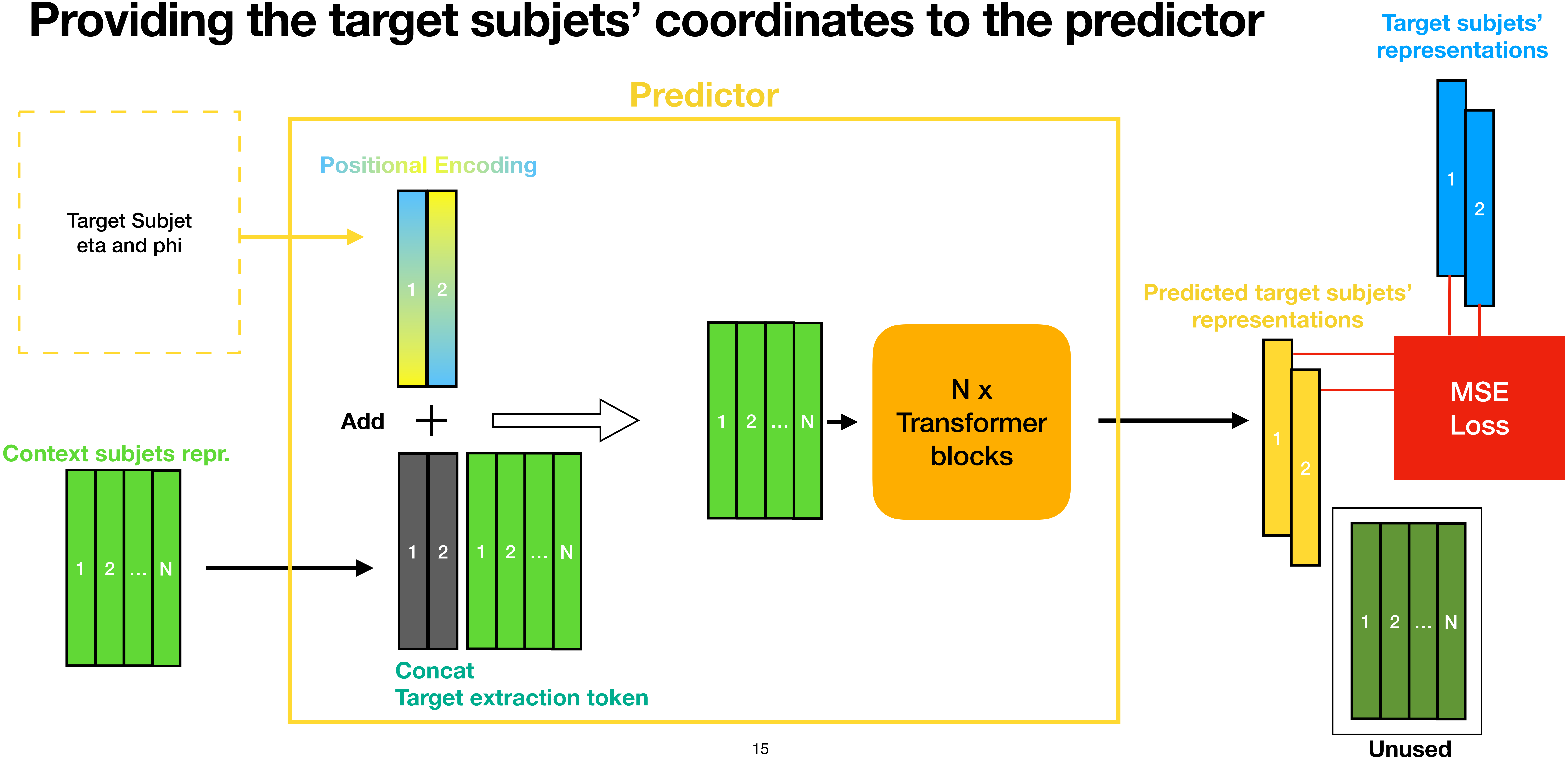
J-JEPA: Calculate Subjet Representations

Using Transformer Encoder Blocks

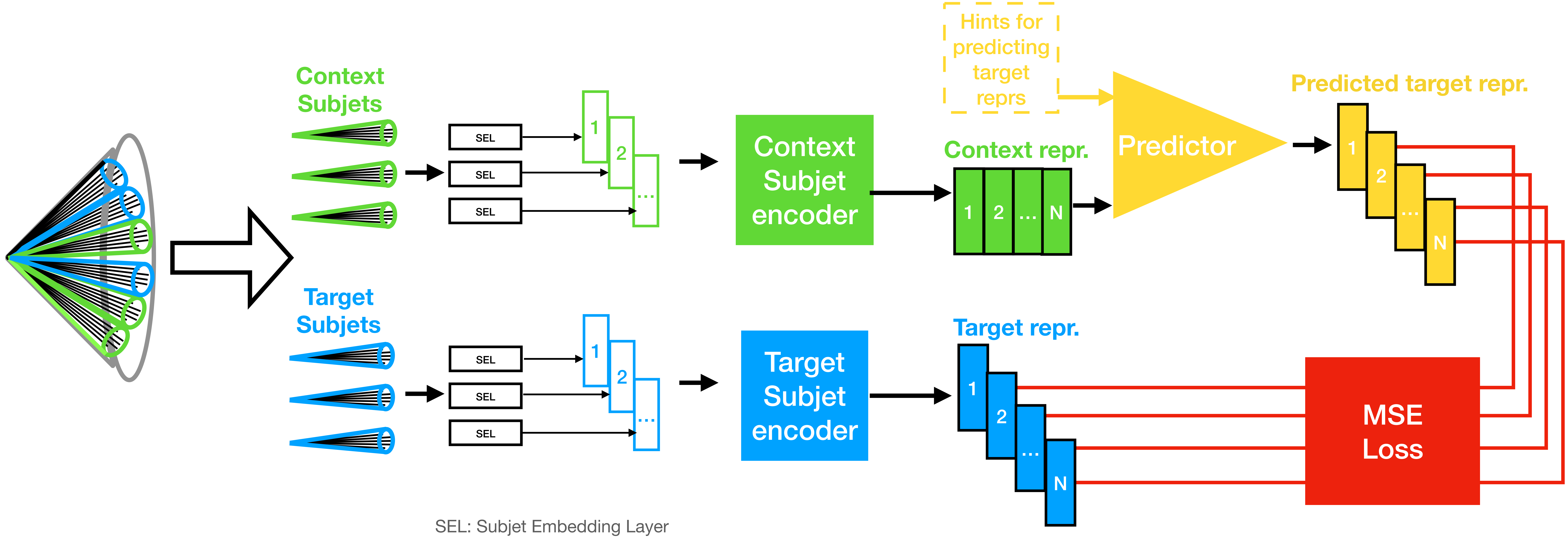


J-JEPA: Predict in the Representation Space

Providing the target subjets' coordinates to the predictor



J-JEPA: Pretraining



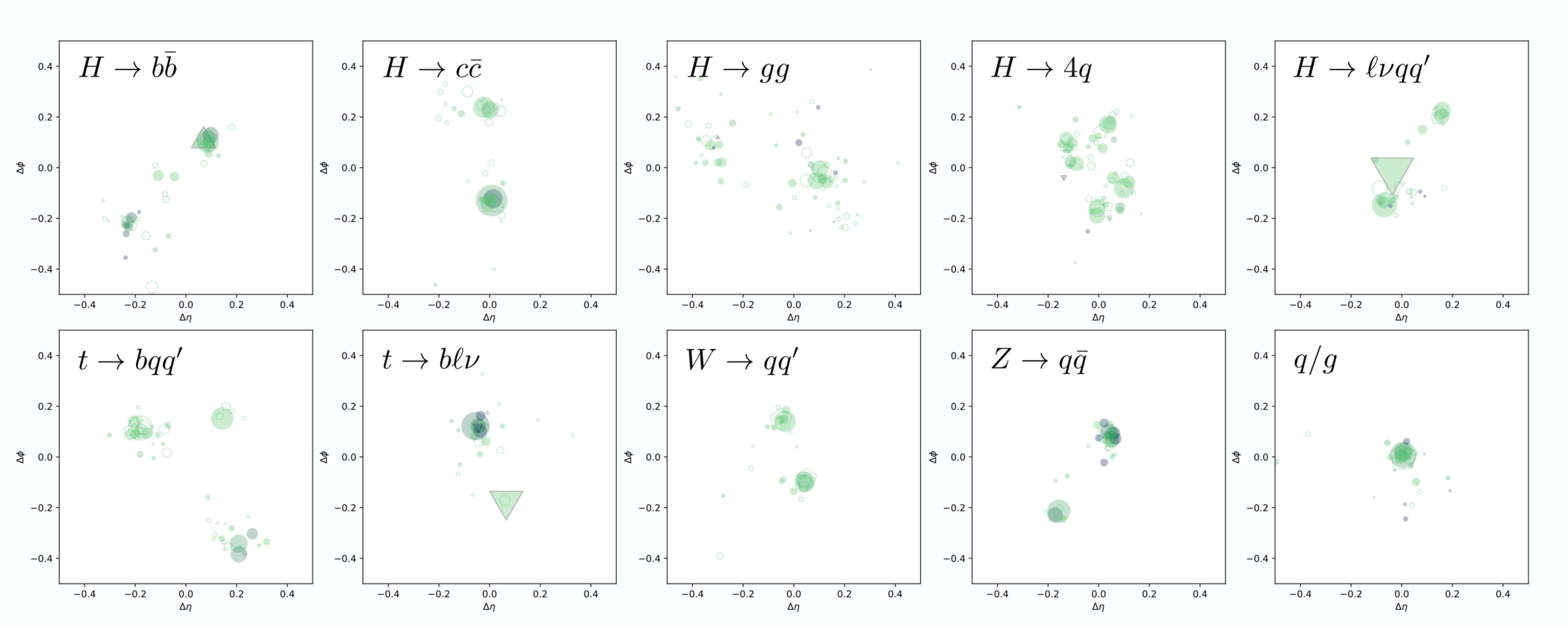
Questions?

Datasets

We use JetClass for pretraining and TopTagging for finetuning

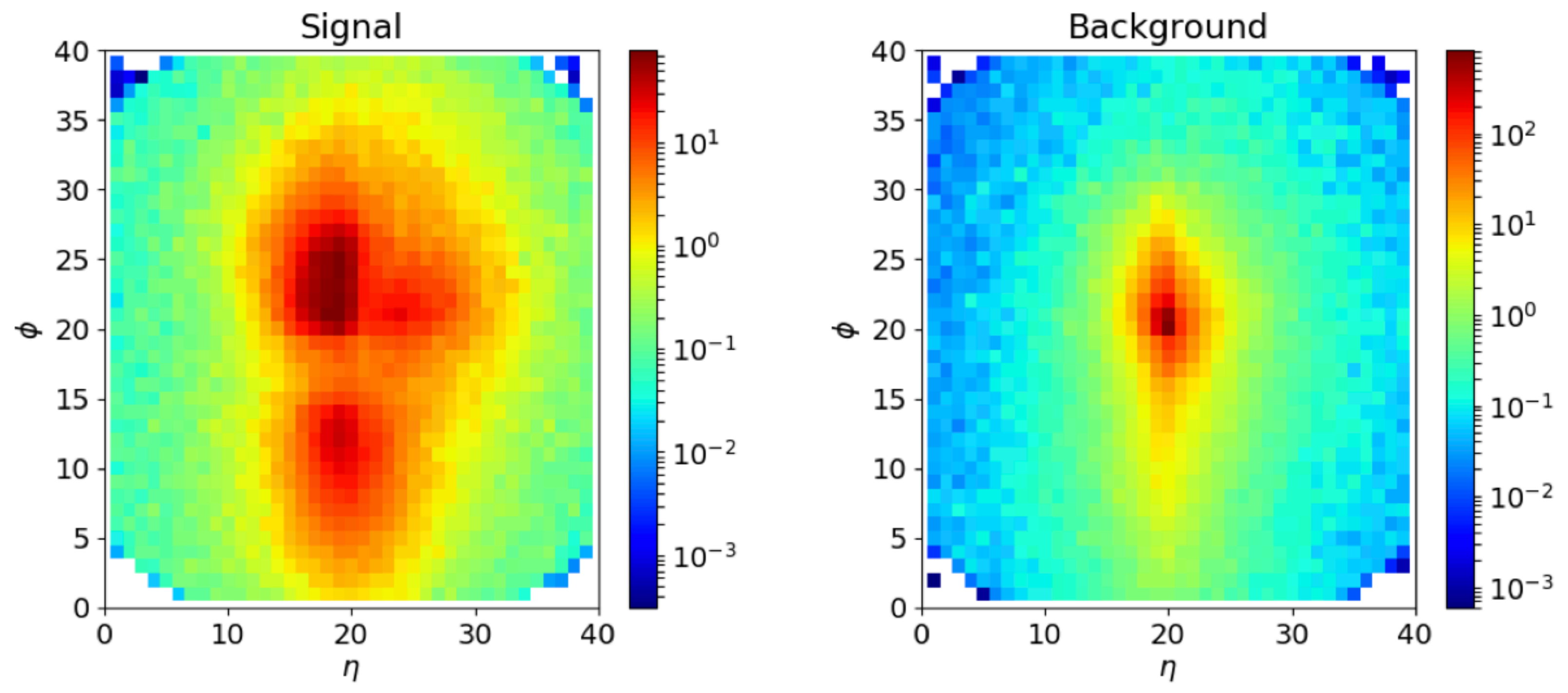
Dataset name	Size	Description	Portions we used	Role in transfer learning
JetClass	100 Million AK8 Jets	Contains 10 classes of jets	500K Top jets 500k q/g jets	Stand in for the large pretaining unlabeled dataset
Top Tagging	1.2 Million AK8 Jets	Only Top and QCD jets	760K mixed jets*	Stand in for the small fine-tuning dataset

* We only used jets with more than 10 subjets



JetClass Dataset

2202.03772



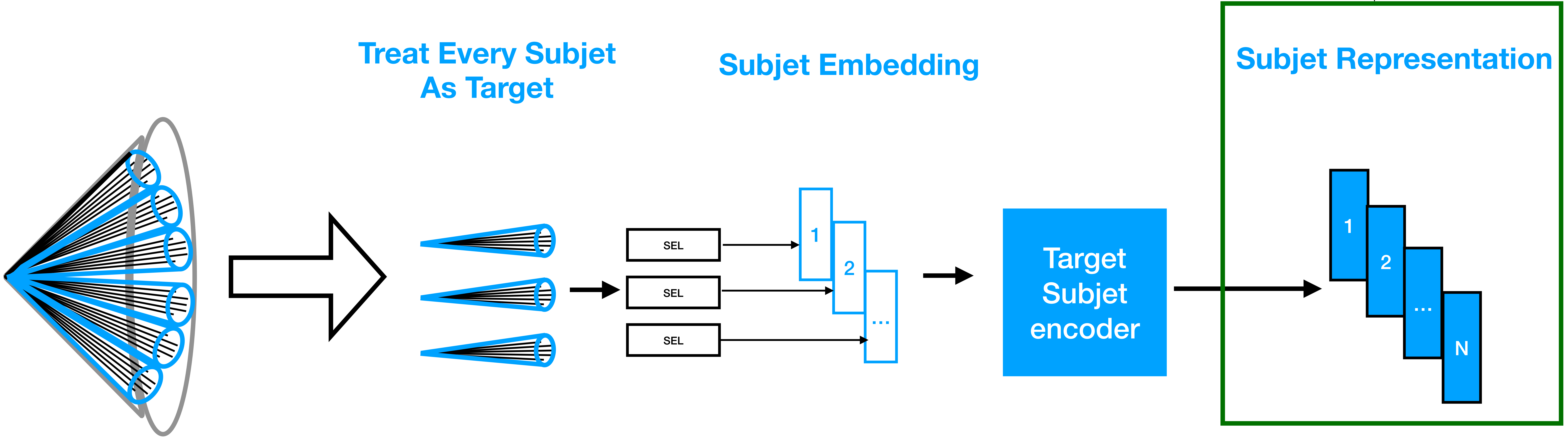
Top Tagging Dataset

1902.09914

J-JEPA: Pretraining Goals

Before we finetune the model with labels

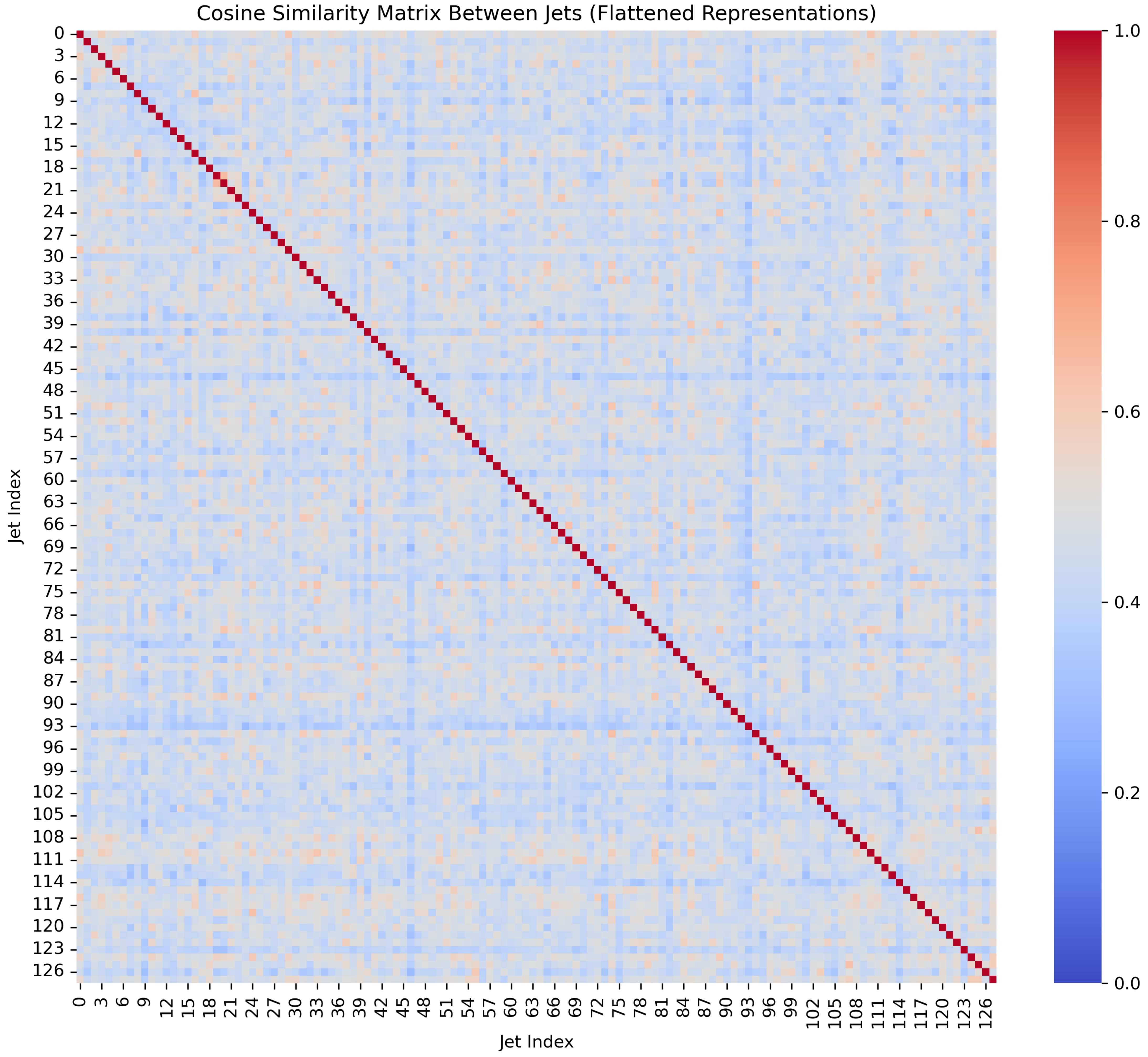
Goal: Jet representation space does not collapse as this will be the latent space connected to the down stream heads



Information collapse: The model fails to capture the meaningful variations in the data, leading to poor performance in tasks like classification or regression.

Latent after Pre-training: Not Collapsing

J-JEPA model learned a diverse latent space



Let A be the features of Jet 1, and B be the features of Jet 2, then the cosine similarity is defined as

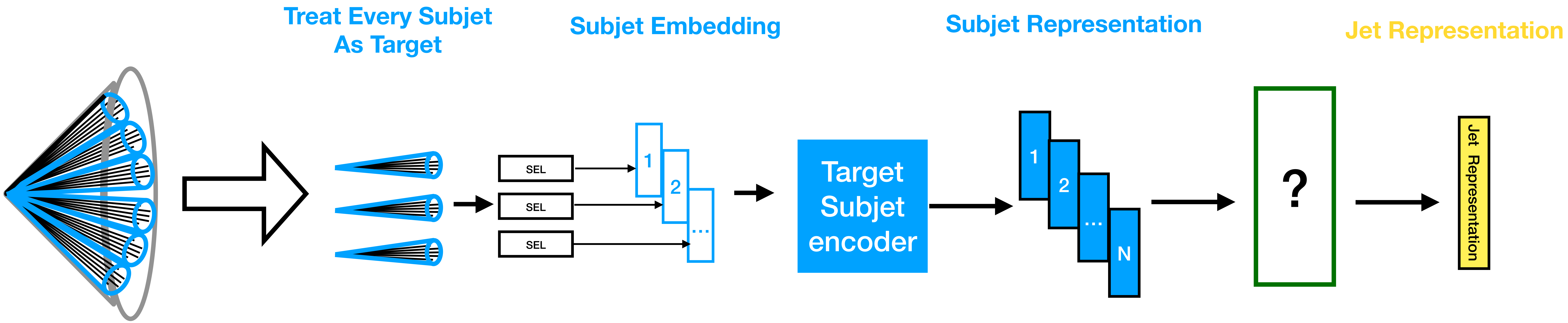
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

1. Randomly select 128 Jets.
2. Represent each jet by their flattened subjet representations
3. Calculate cosine similarity between each pair of jets

Average Cosine Similarity: 0.457

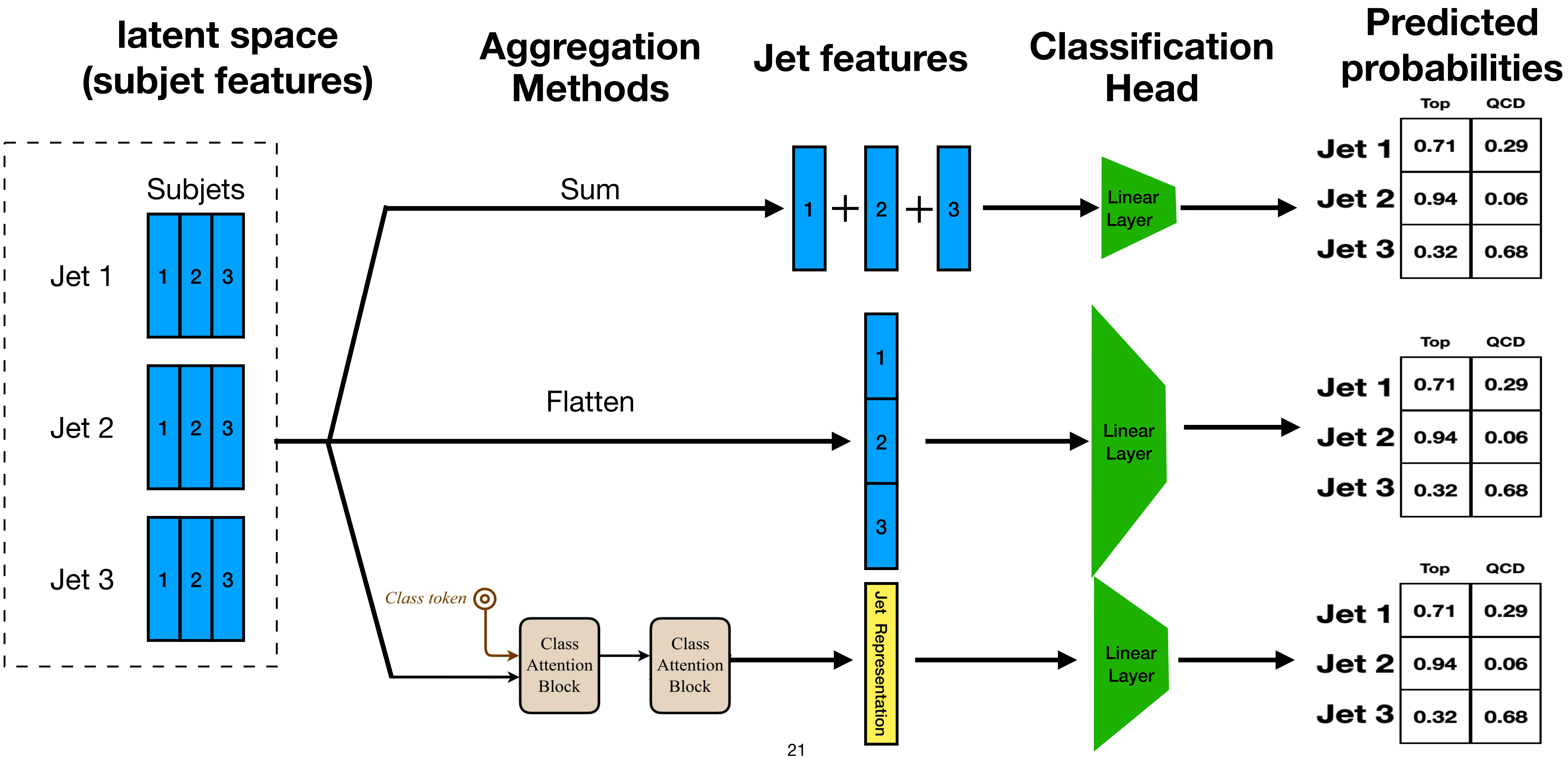
J-JEPA: Finetuning Setup

From subjet representation to jet representation



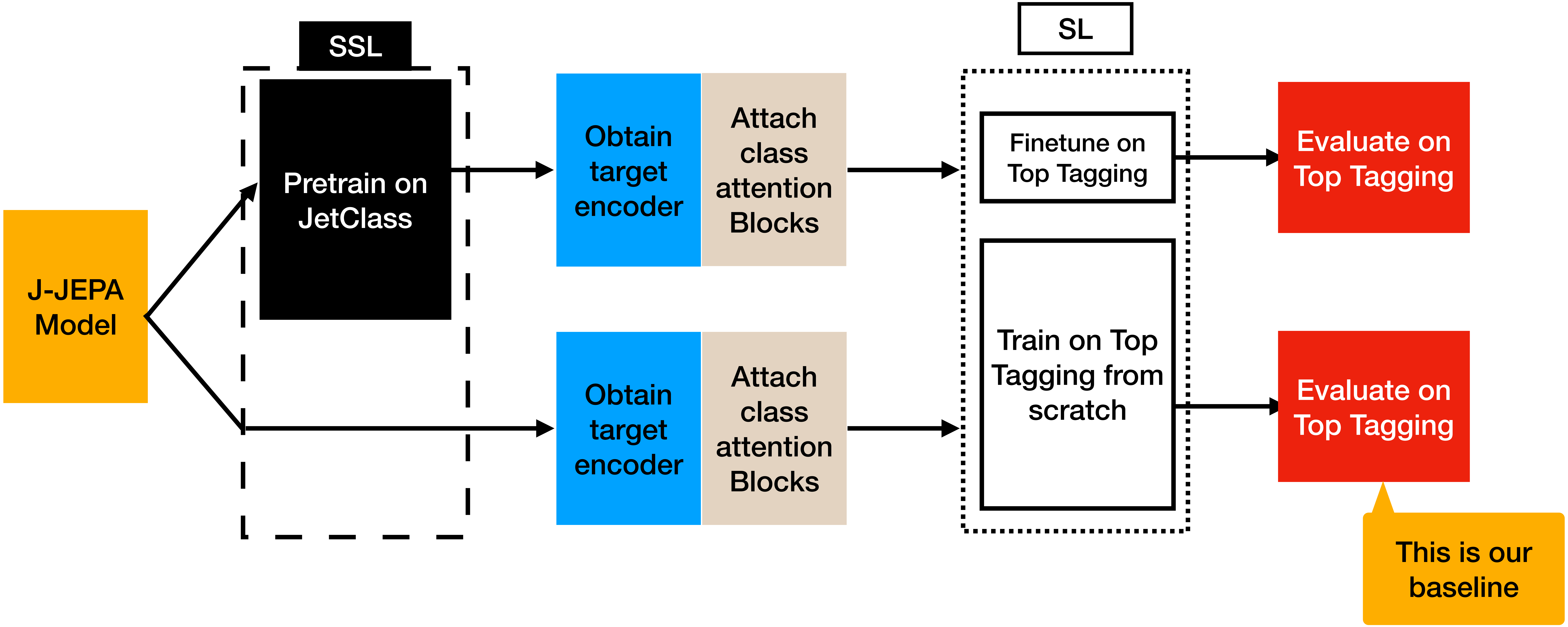
Aggregation Methods for Fine-tuning

3 Different methods of attaching the latent space to a classification head



Our training and evaluation setup

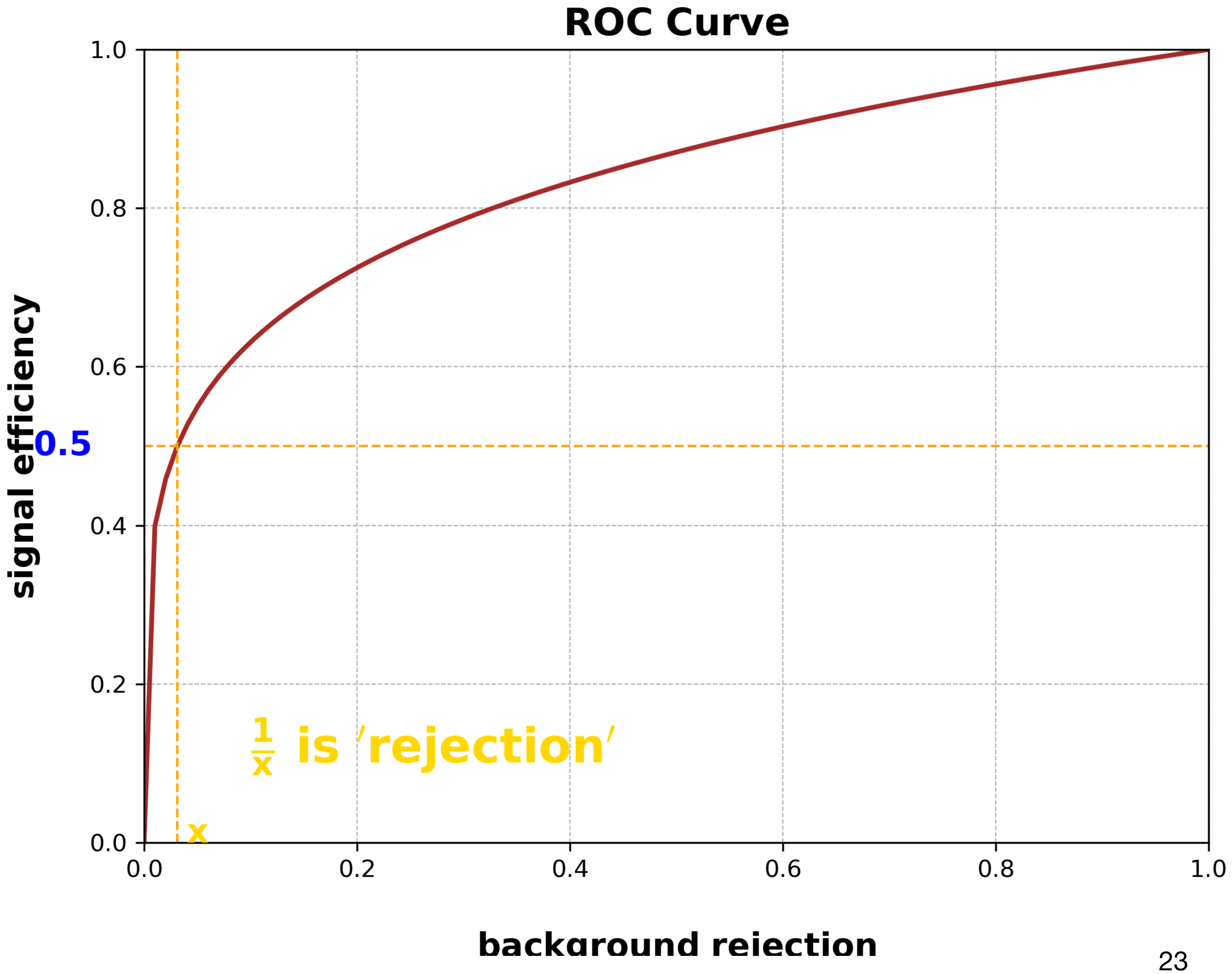
Baseline refers to the same model directly trained on the finetuning dataset without pretraining



Metrics

Accuracy: correctly predicted / total number of samples

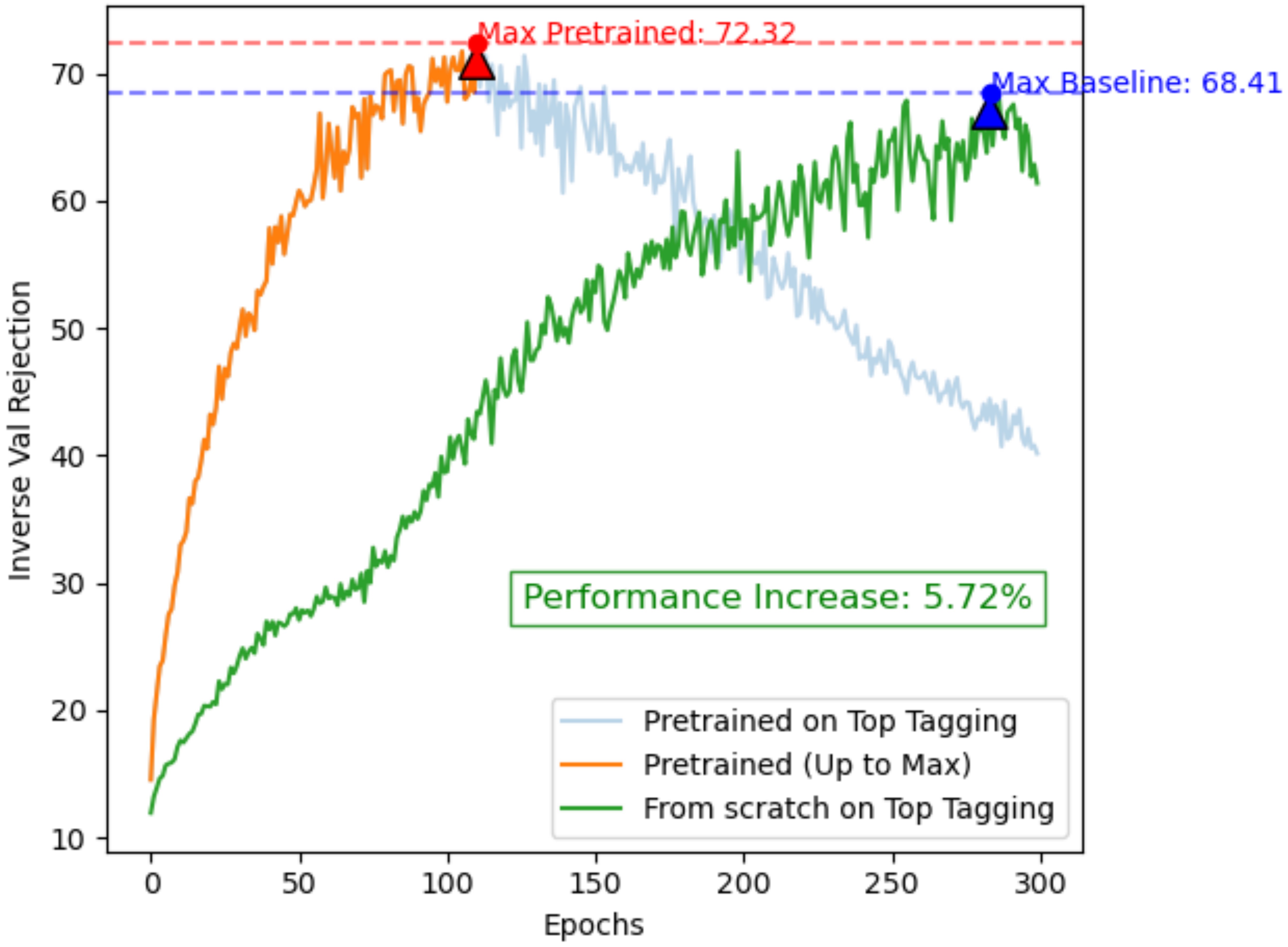
Rejection: inverse of background rejection (FPR) at 50% signal efficiency (TPR)



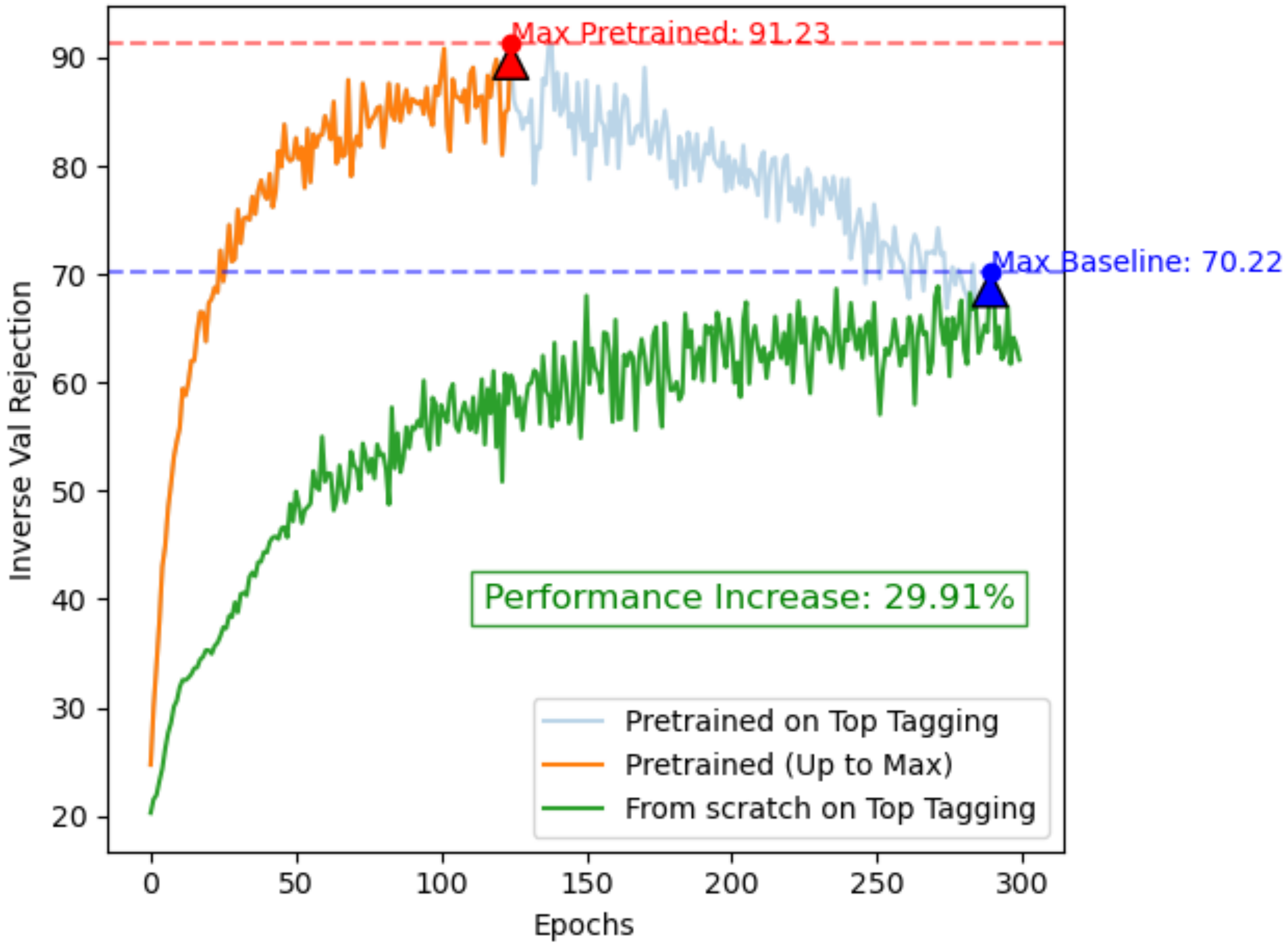
Significance: In a background dominant dataset, how much background can you reject while letting in a certain number of signal samples (the more the better)

J-JEPA Performance

Pretrain on JetClass and finetune on Top Tagging



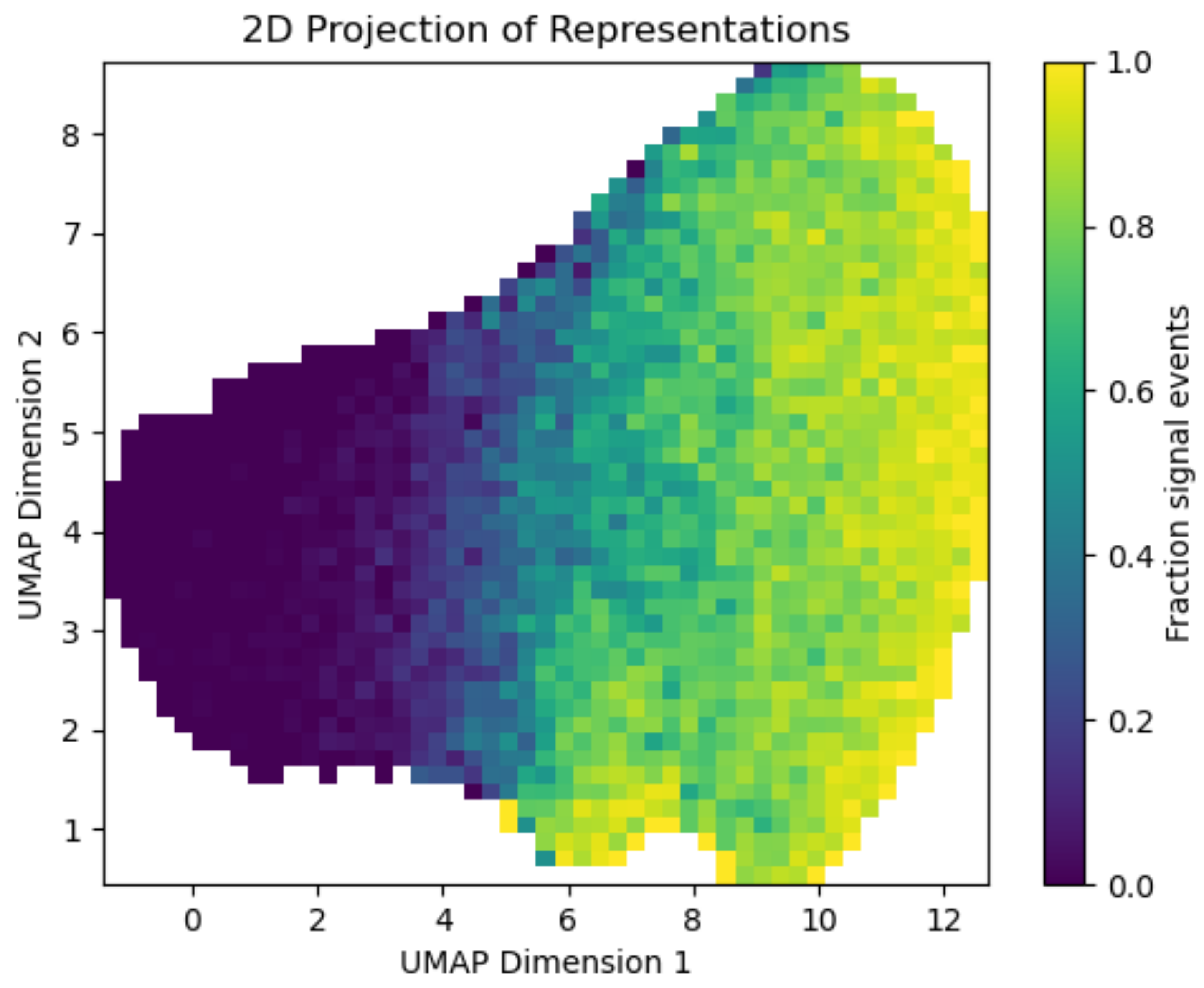
Attention-based SEL

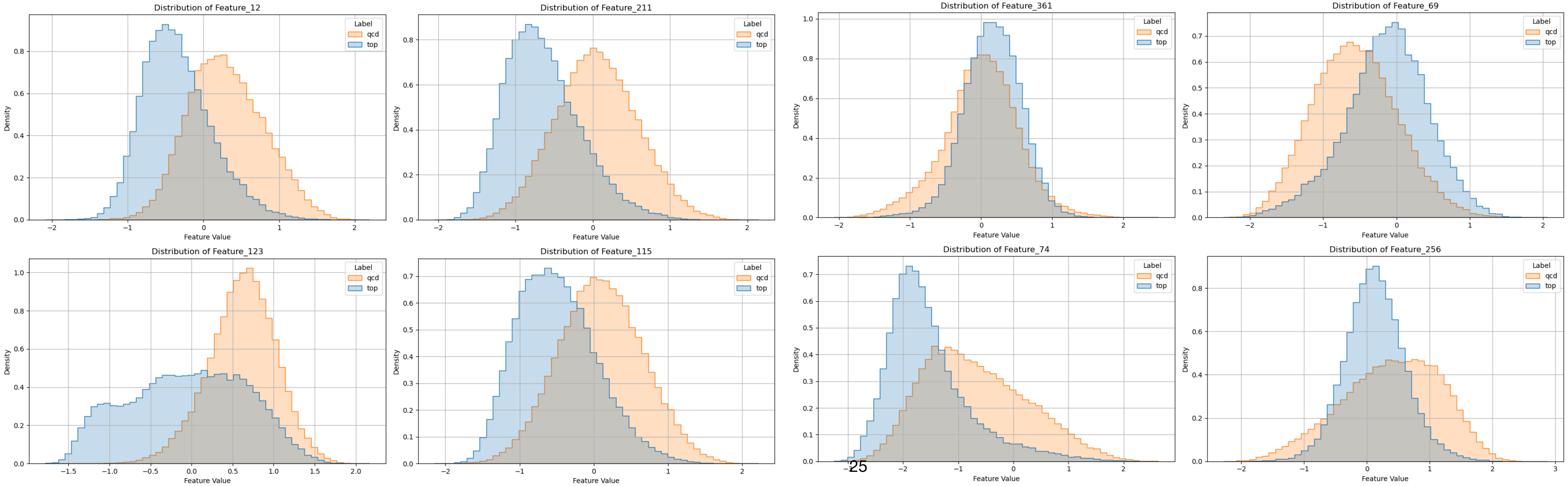


MLP-based SEL

Visualizing learned features

UMAP and direct comparison show that the features have good separation power

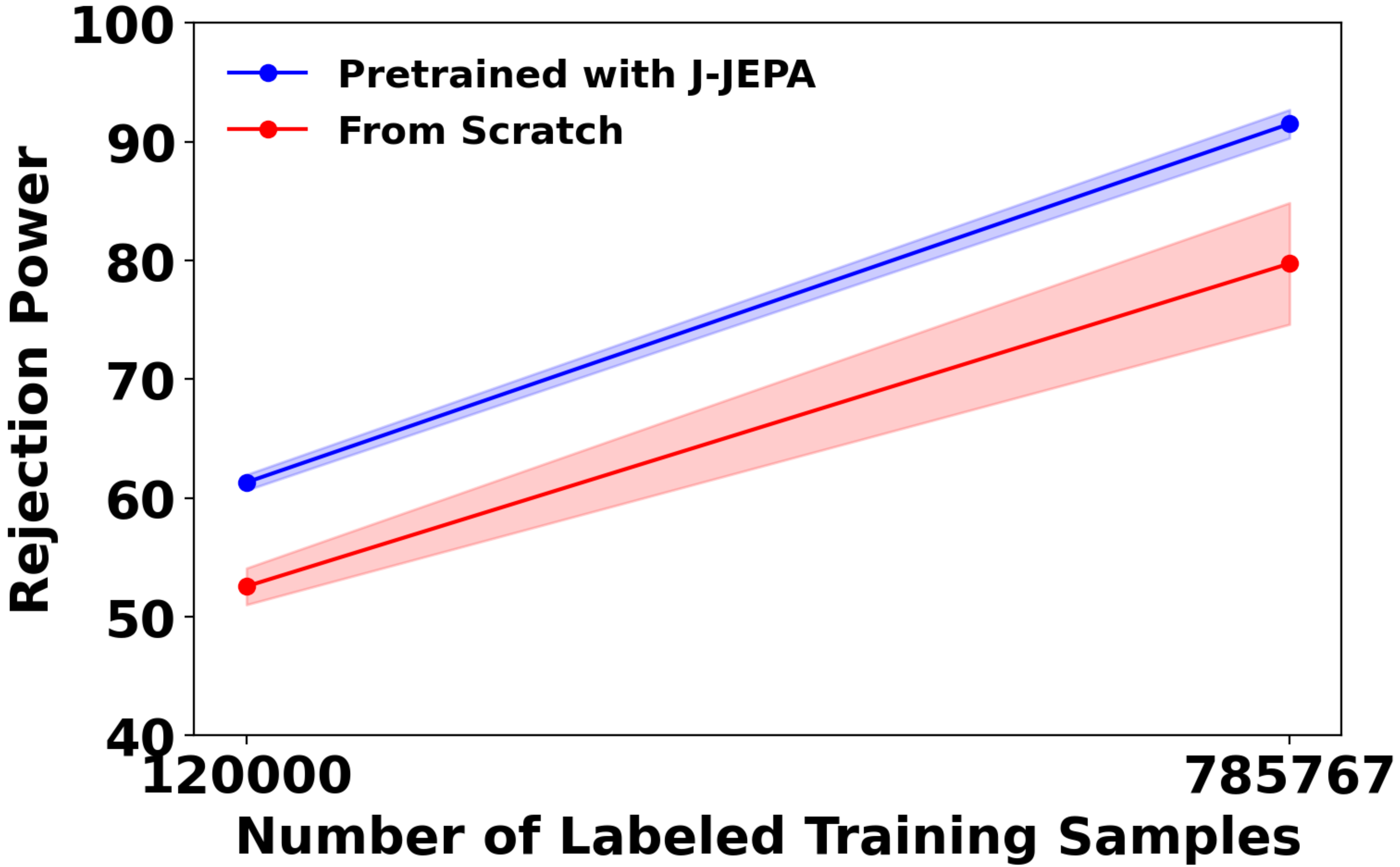




Our results

1. J-JEPA improves the downstream performance compared with from scratch (for most models)

Model	Aggregation	Baseline 10%	Baseline Full	Finetuned 10%	Finetuned Full
		Accuracy [%]			
SjT-T	Flatten	87.52 ± 0.16	89.13 ± 0.10	88.21 ± 0.55	89.95 ± 0.13
SjT-T	Cls Attn	88.30 ± 0.18	89.67 ± 0.13	88.67 ± 0.02	90.00 ± 0.07
AE-SjT-T	Flatten	88.92 ± 0.15	90.01 ± 0.08	$\mathbf{88.94 \pm 0.13}$	$\mathbf{90.03 \pm 0.07}$
AE-SjT-T	Cls Attn	88.84 ± 0.21	$\mathbf{90.03 \pm 0.05}$	88.82 ± 0.11	90.00 ± 0.12
		$1/\varepsilon_B(\varepsilon_S = 0.5)$			
SjT-T	Flatten	40.50 ± 1.26	70.70 ± 1.46	53.67 ± 9.97	90.06 ± 3.80
SjT-T	Cls Attn	52.56 ± 1.54	79.75 ± 5.12	61.32 ± 0.66	91.51 ± 1.20
AE-SjT-T	Flatten	67.34 ± 1.40	97.79 ± 3.90	$\mathbf{70.47 \pm 1.09}$	97.52 ± 1.71
AE-SjT-T	Cls Attn	67.19 ± 1.54	$\mathbf{99.38 \pm 2.80}$	68.25 ± 1.64	95.47 ± 1.83

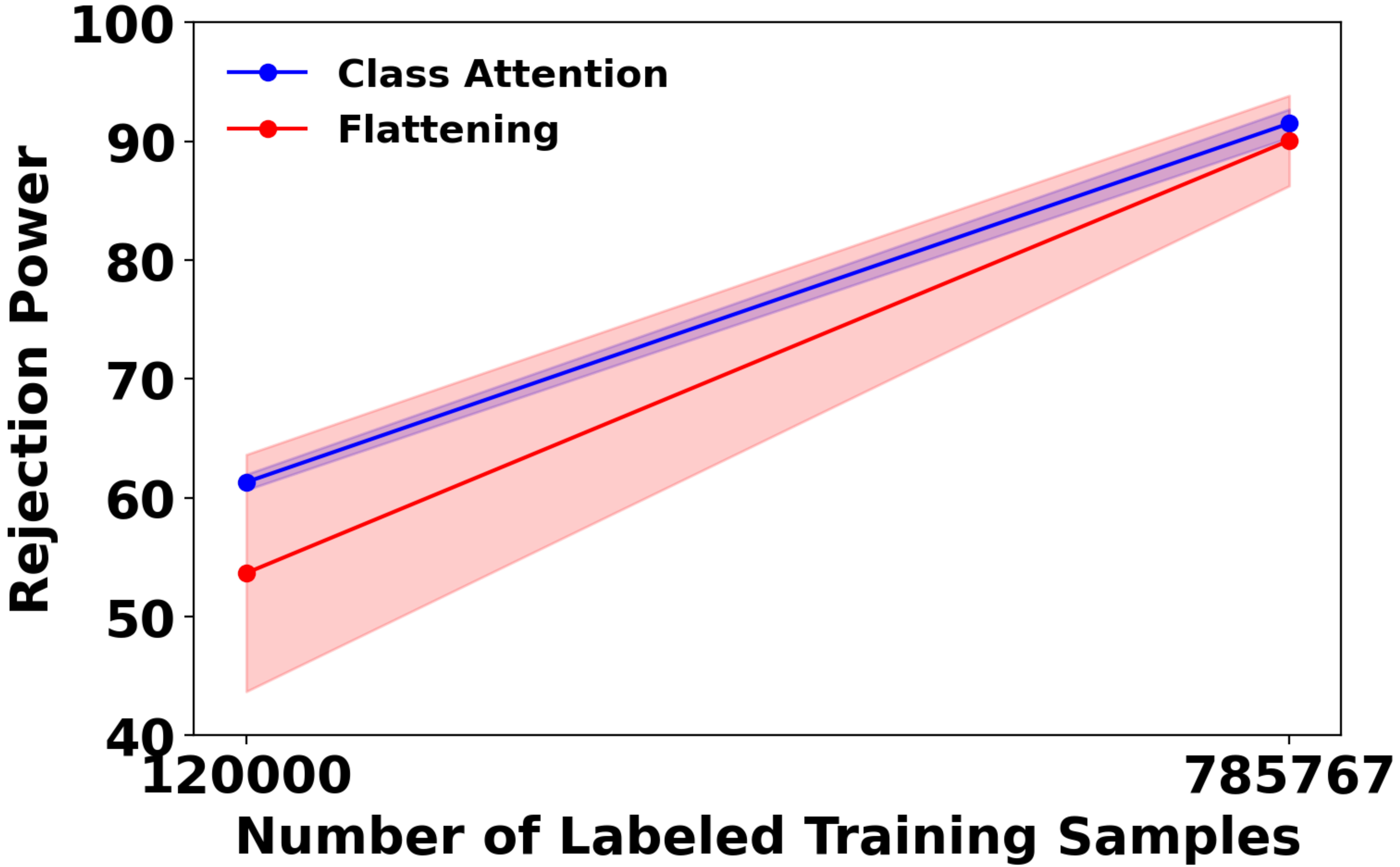


Pretrained with J-JEPA
From Scratch
Rejection Power
100
90
80
70
60
50
40
120000
785767
Number of Labeled Training Samples

Our results

2. Class attention blocks are more effective than simply flattening (for most models)

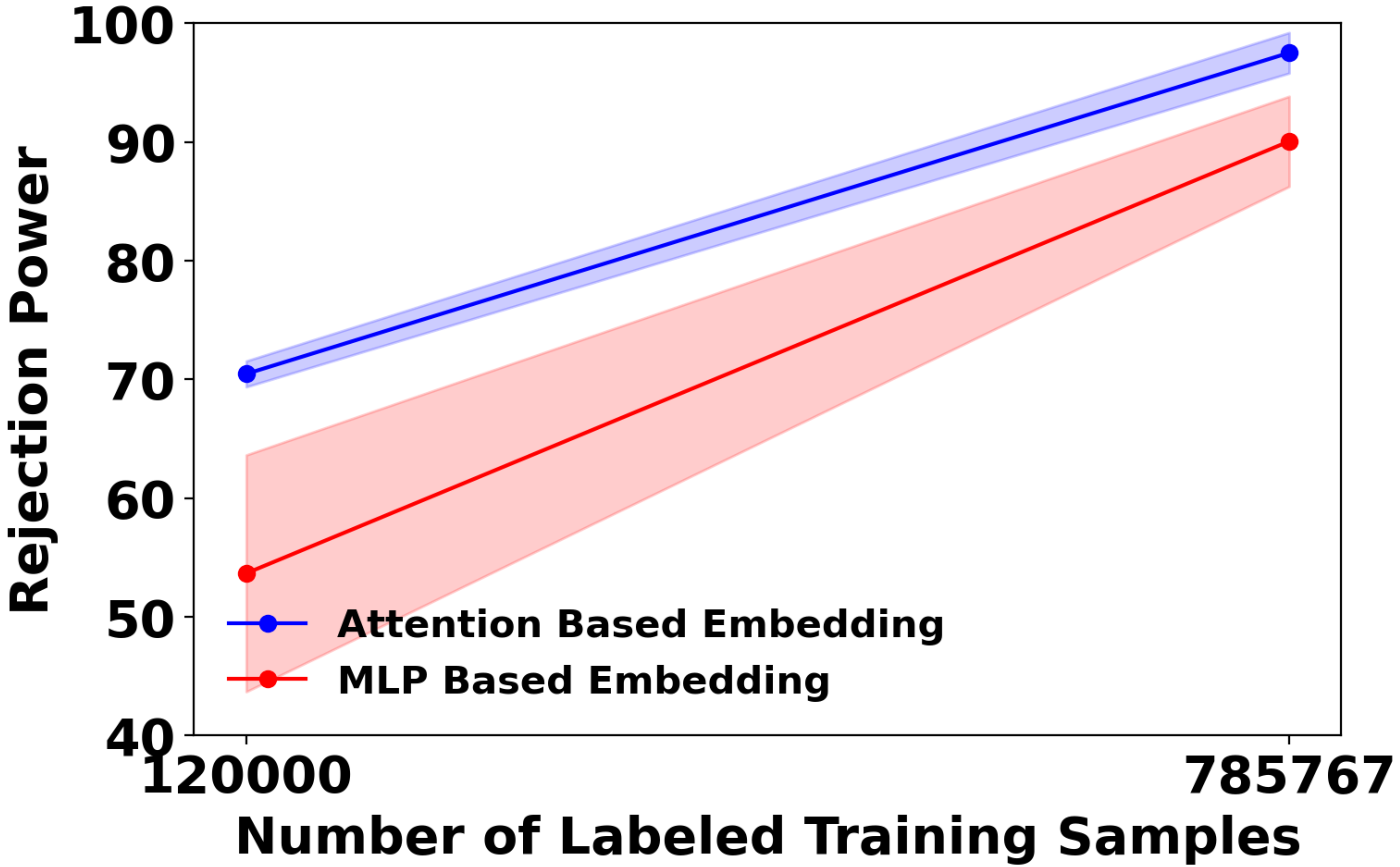
Model	Aggregation	Baseline 10%	Baseline Full	Finetuned 10%	Finetuned Full
		Accuracy [%]			
SjT-T	Flatten	87.52 ± 0.16	89.13 ± 0.10	88.21 ± 0.55	89.95 ± 0.13
SjT-T	Cls Attn	88.30 ± 0.18	89.67 ± 0.13	88.67 ± 0.02	90.00 ± 0.07
AE-SjT-T	Flatten	88.92 ± 0.15	90.01 ± 0.08	$\mathbf{88.94 \pm 0.13}$	$\mathbf{90.03 \pm 0.07}$
AE-SjT-T	Cls Attn	88.84 ± 0.21	$\mathbf{90.03 \pm 0.05}$	88.82 ± 0.11	90.00 ± 0.12
		$1/\varepsilon_B(\varepsilon_S = 0.5)$			
SjT-T	Flatten	40.50 ± 1.26	70.70 ± 1.46	53.67 ± 9.97	90.06 ± 3.80
SjT-T	Cls Attn	52.56 ± 1.54	79.75 ± 5.12	61.32 ± 0.66	91.51 ± 1.20
AE-SjT-T	Flatten	67.34 ± 1.40	97.79 ± 3.90	$\mathbf{70.47 \pm 1.09}$	97.52 ± 1.71
AE-SjT-T	Cls Attn	67.19 ± 1.54	$\mathbf{99.38 \pm 2.80}$	68.25 ± 1.64	95.47 ± 1.83


Class Attention
Flattening
Rejection Power
100
90
80
70
60
50
40
120000
785767
Number of Labeled Training Samples

Our results

3. Our custom attention-based embeddings offer a significant improvement in downstream performance compared with the traditional MLP-based embeddings

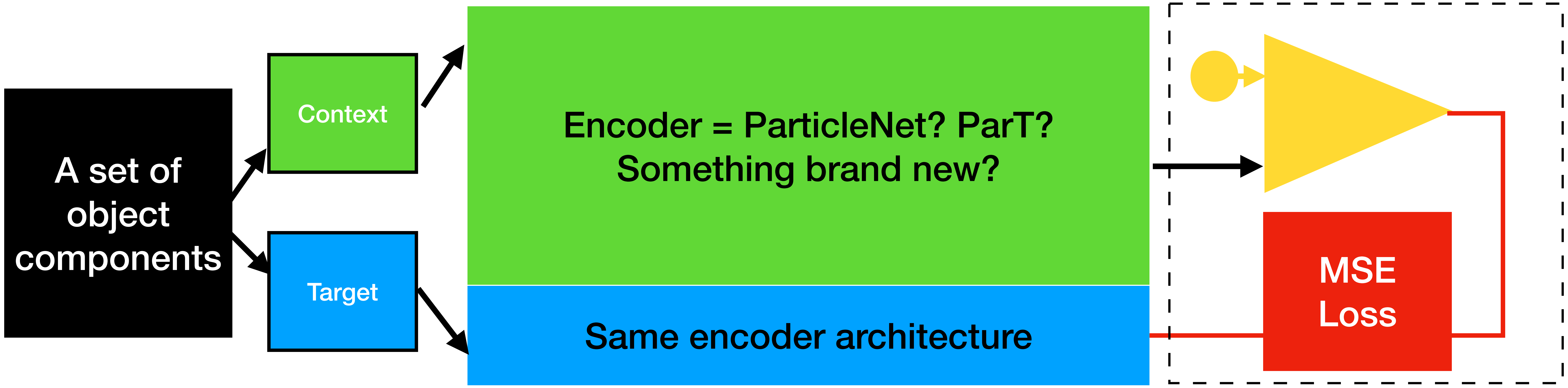
Model	Aggregation	Baseline 10%	Baseline Full	Finetuned 10%	Finetuned Full
		Accuracy [%]			
SjT-T	Flatten	87.52 ± 0.16	89.13 ± 0.10	88.21 ± 0.55	89.95 ± 0.13
SjT-T	Cls Attn	88.30 ± 0.18	89.67 ± 0.13	88.67 ± 0.02	90.00 ± 0.07
AE-SjT-T	Flatten	88.92 ± 0.15	90.01 ± 0.08	$\mathbf{88.94 \pm 0.13}$	$\mathbf{90.03 \pm 0.07}$
AE-SjT-T	Cls Attn	88.84 ± 0.21	$\mathbf{90.03 \pm 0.05}$	88.82 ± 0.11	90.00 ± 0.12
		$1/\varepsilon_B(\varepsilon_S = 0.5)$			
SjT-T	Flatten	40.50 ± 1.26	70.70 ± 1.46	53.67 ± 9.97	90.06 ± 3.80
SjT-T	Cls Attn	52.56 ± 1.54	79.75 ± 5.12	61.32 ± 0.66	91.51 ± 1.20
AE-SjT-T	Flatten	67.34 ± 1.40	97.79 ± 3.90	$\mathbf{70.47 \pm 1.09}$	97.52 ± 1.71
AE-SjT-T	Cls Attn	67.19 ± 1.54	$\mathbf{99.38 \pm 2.80}$	68.25 ± 1.64	95.47 ± 1.83



100
90
80
70
60
50
40
Rejection Power
120000
785767
Number of Labeled Training Samples
Attention Based Embedding
MLP Based Embedding

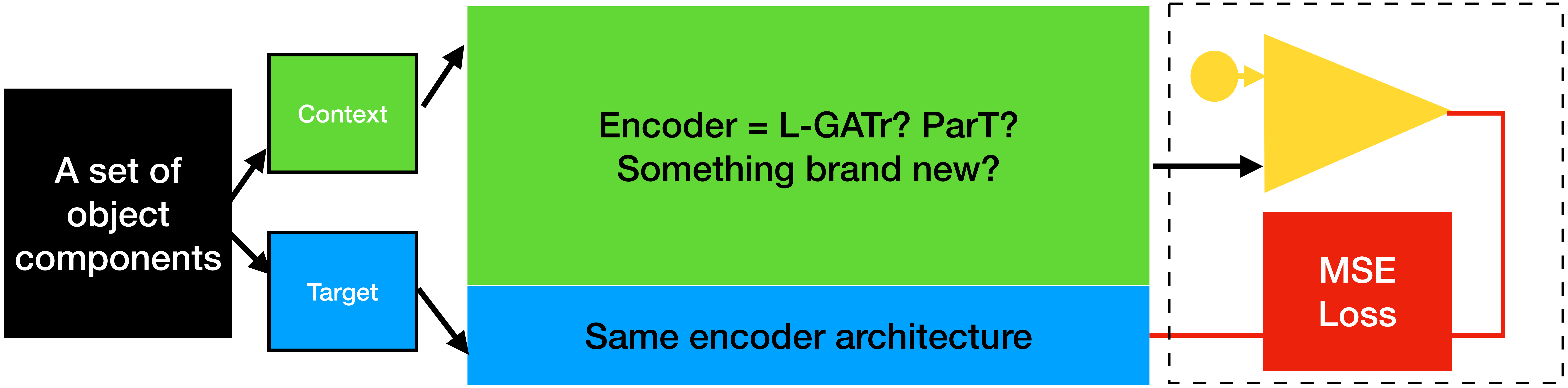
Summary

- J-JEPA: A subject-based Joint-Embedding Predictive Architecture
- Pre-train J-JEPA on a large dataset and finetune the target encoder on a small dataset achieves better performance than training the encoder from scratch,
- Different encoder architectures has different response to the J-JEPA pre-training, but overall positive.



Ongoing Work

- Implementing a particle-based JEPA
- Training shorter models to reduce overfitting
- Experiment different ways to provide information to the predictor
- Generalize the JEPA scheme to different physics objects: particles, events, detector readout, etc.



Large-Scale Pretraining and Finetuning for Efficient Jet Classification

Zihan Zhao, Farouk Mokhtar, Raghav Kansal, Billy Li, Javier Duarte

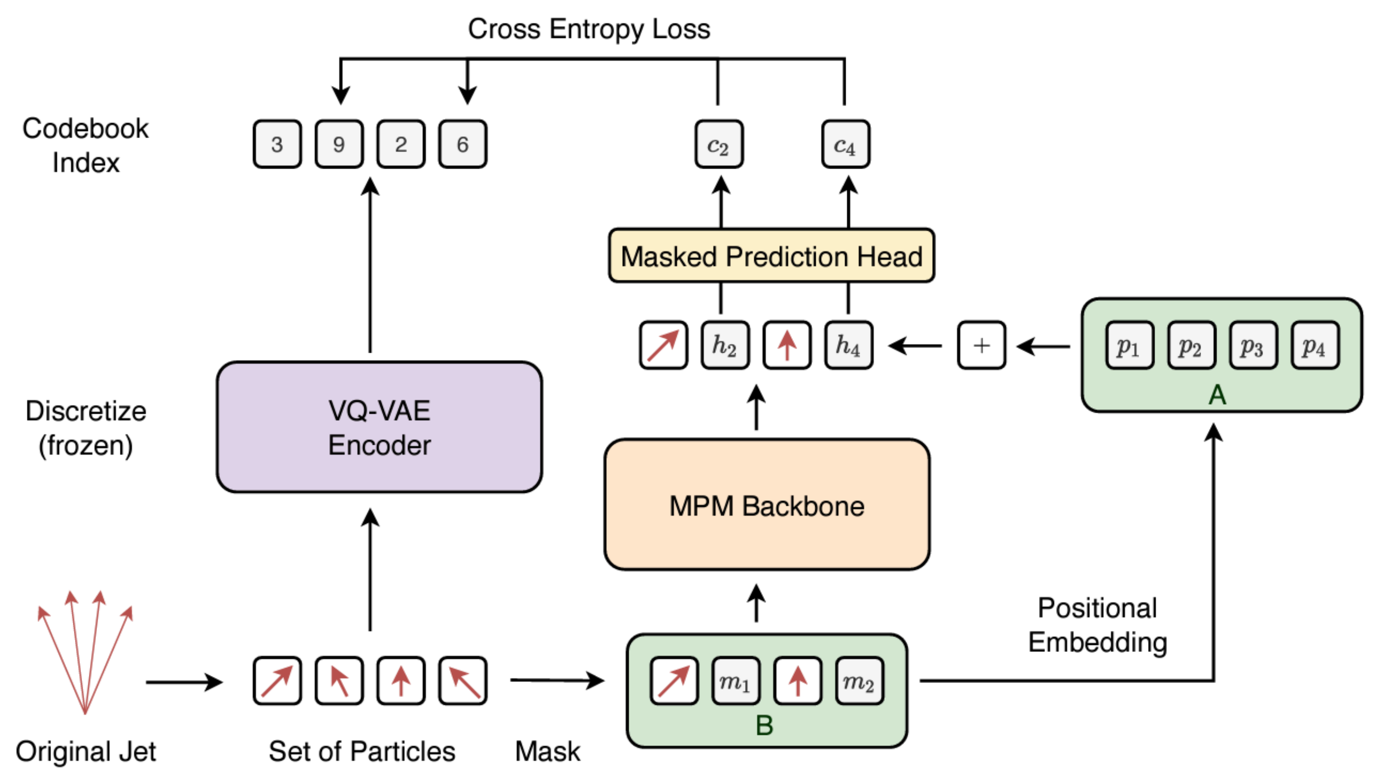
Larger than Larger Ep1 2025
Jan 7

https://arxiv.org/abs/2408.09343

Intro to SSL strategies

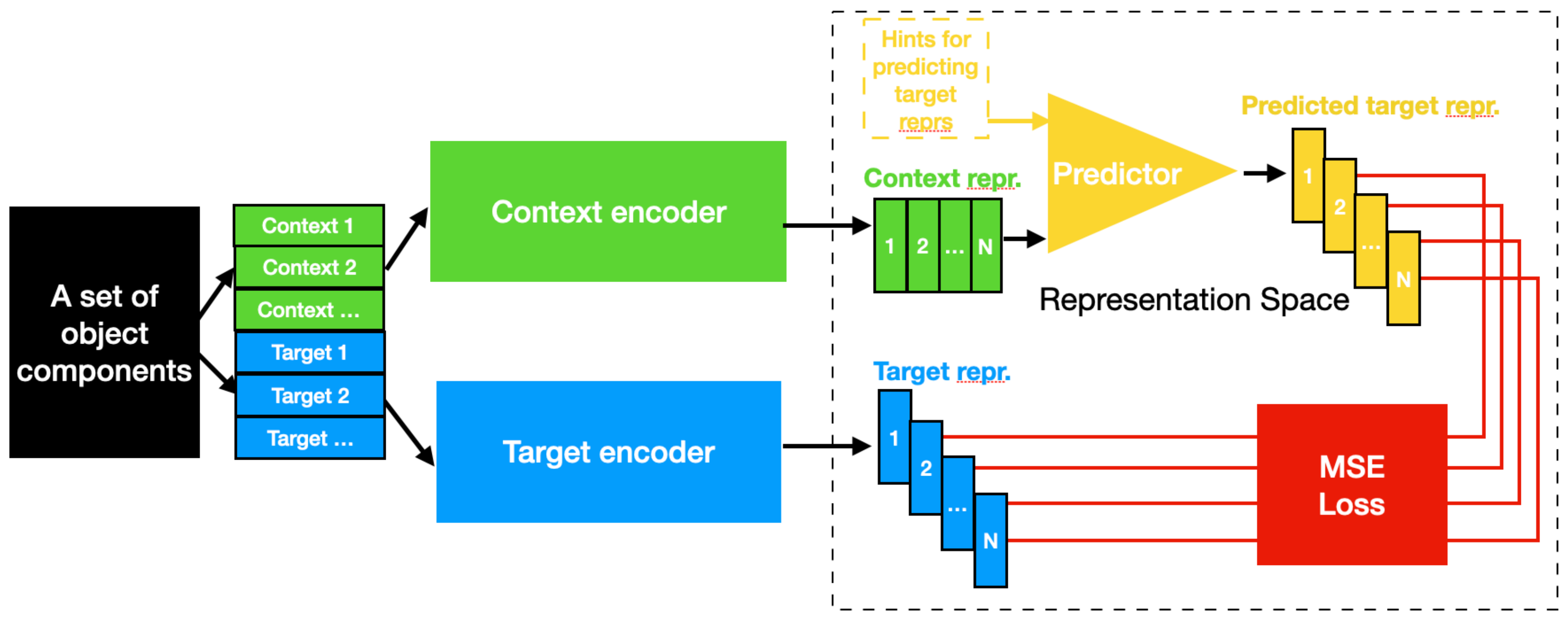
As opposed to supervised learning, which is limited by the availability of labeled data, self-supervised approaches can learn from vast unlabeled data (2304.12210)

To learn useful features from the data itself without using labels



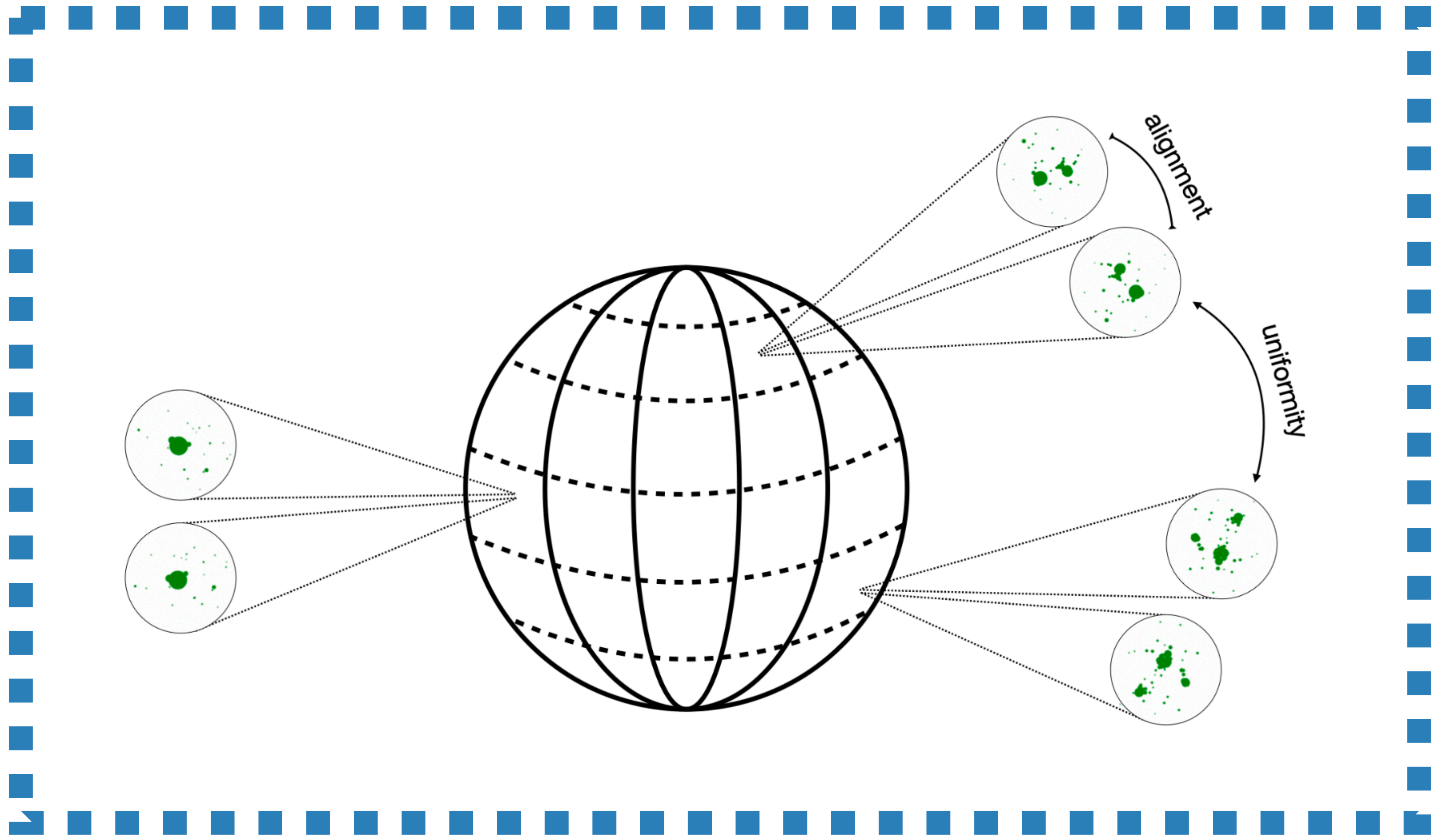
Masked Modeling

2401.13537



JEPA

2412.05333

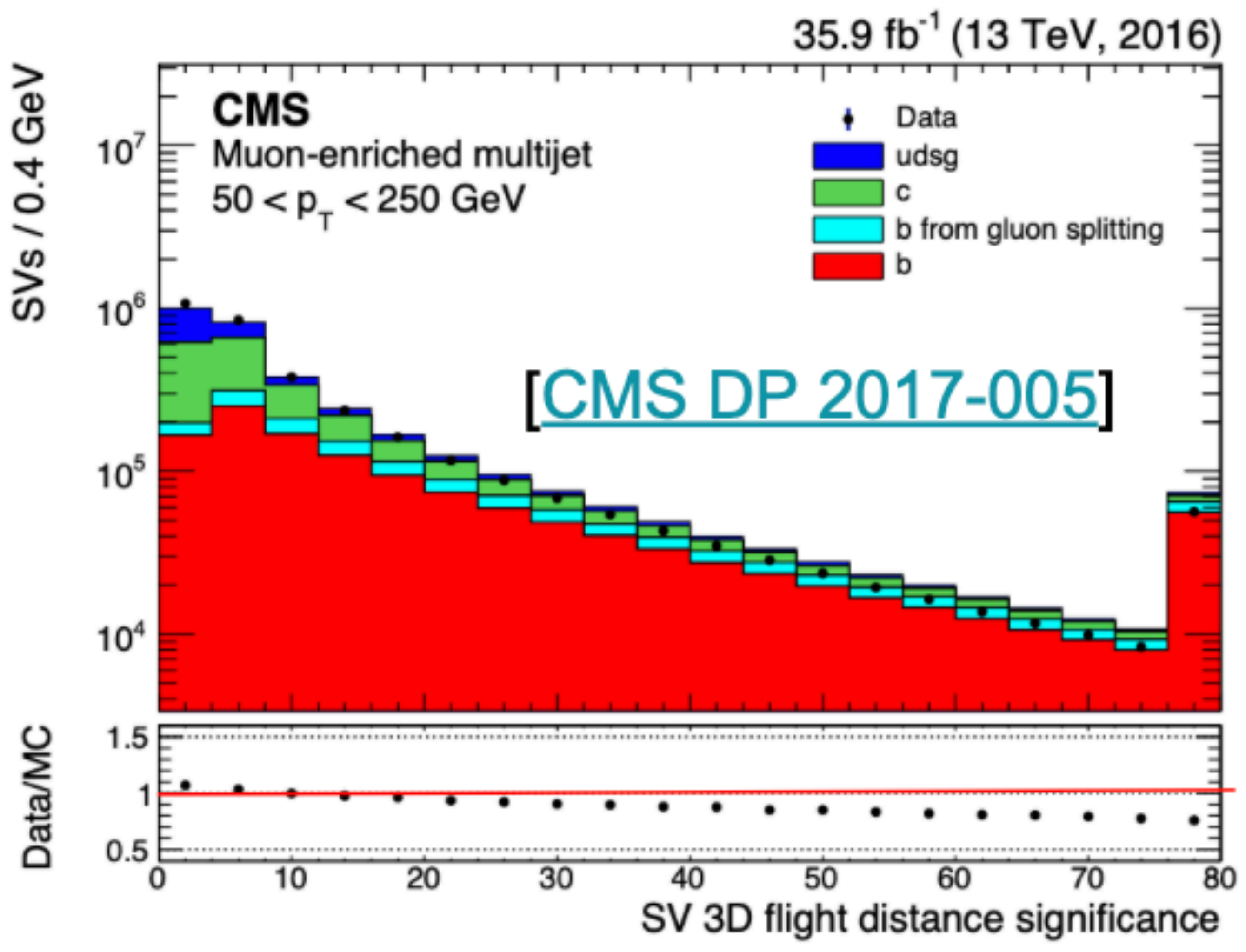


Contrastive Learning

2108.04253

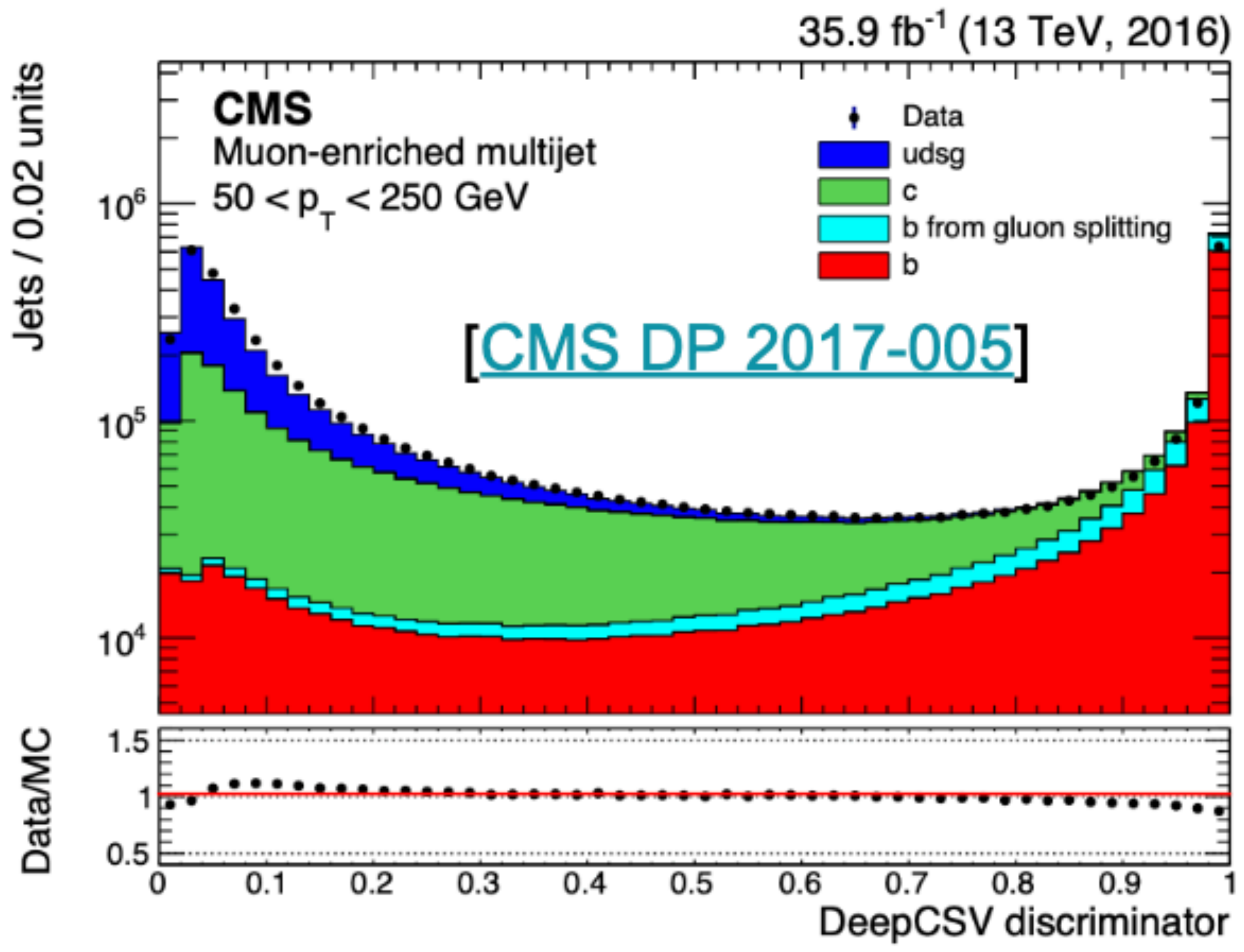
Necessity of SSL in LHC Physics

- Simulations don’t model the data perfectly: need a way to directly train on data
- It will be even harder and more computationally expensive to produce high-quality simulations for High Luminosity LHC (1803.04165)



Input variable

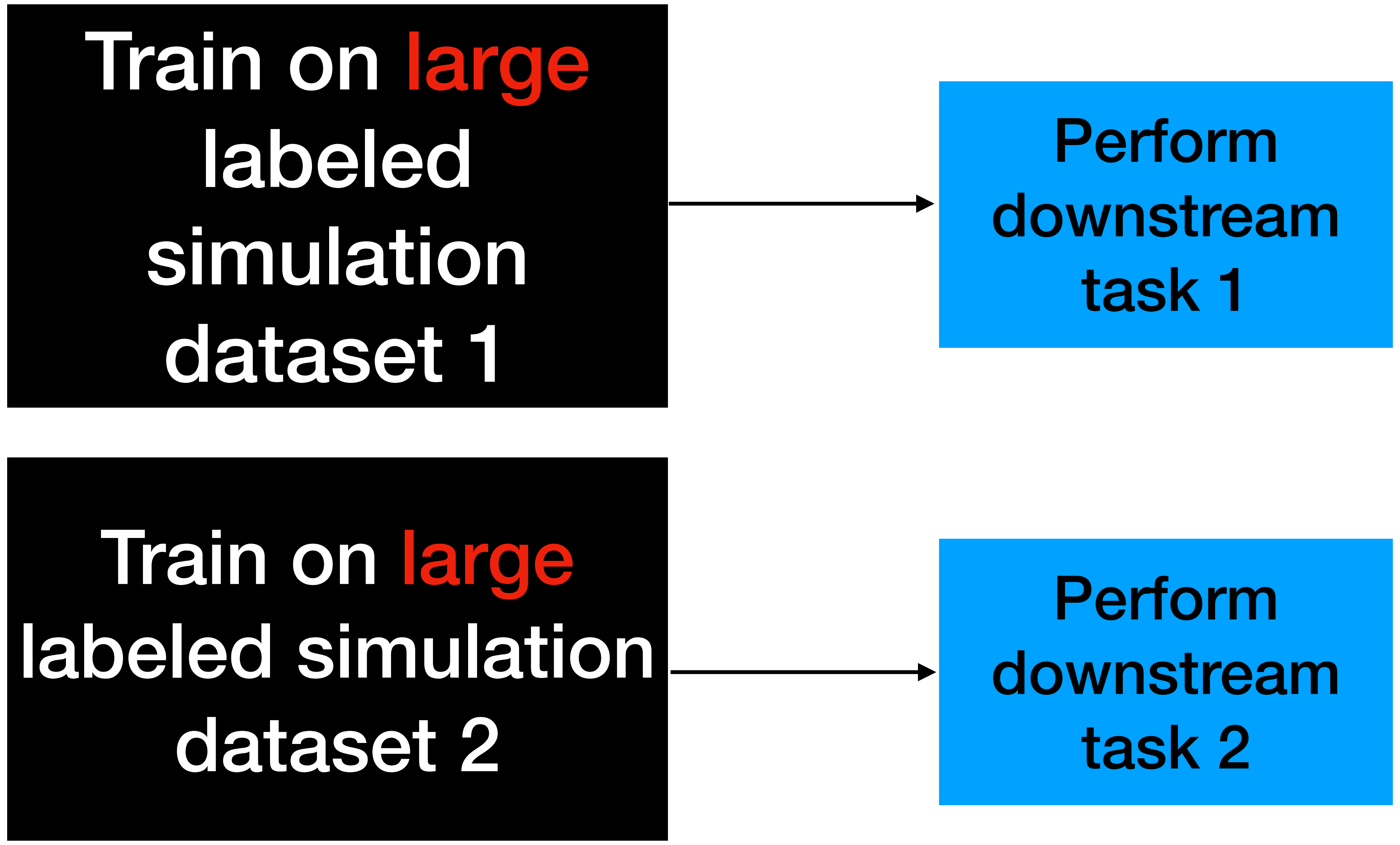
Mismodelling propagates to discriminant



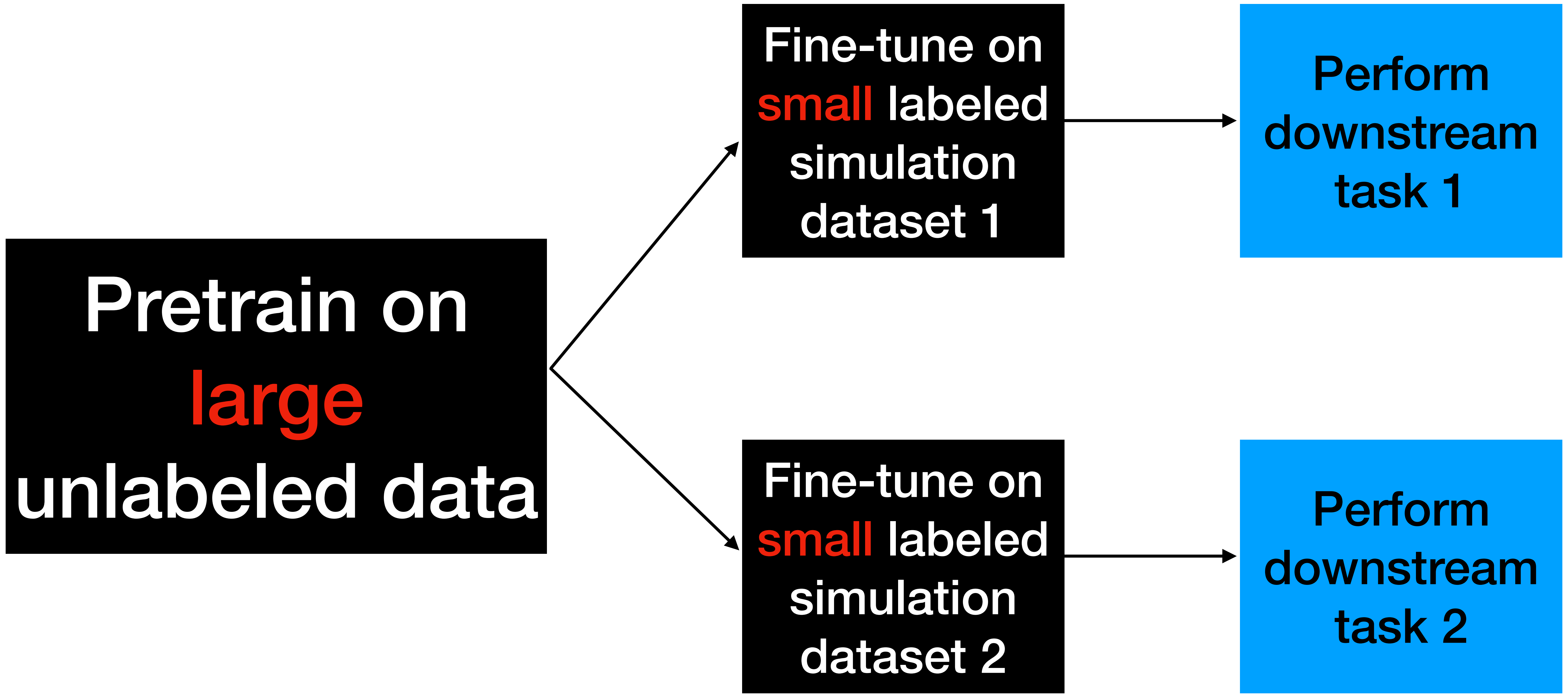
Output discriminant

First Goal of the Project

- To show that we can leverage SSL to learn powerful, generic, and transferable features directly from vast unlabeled data.

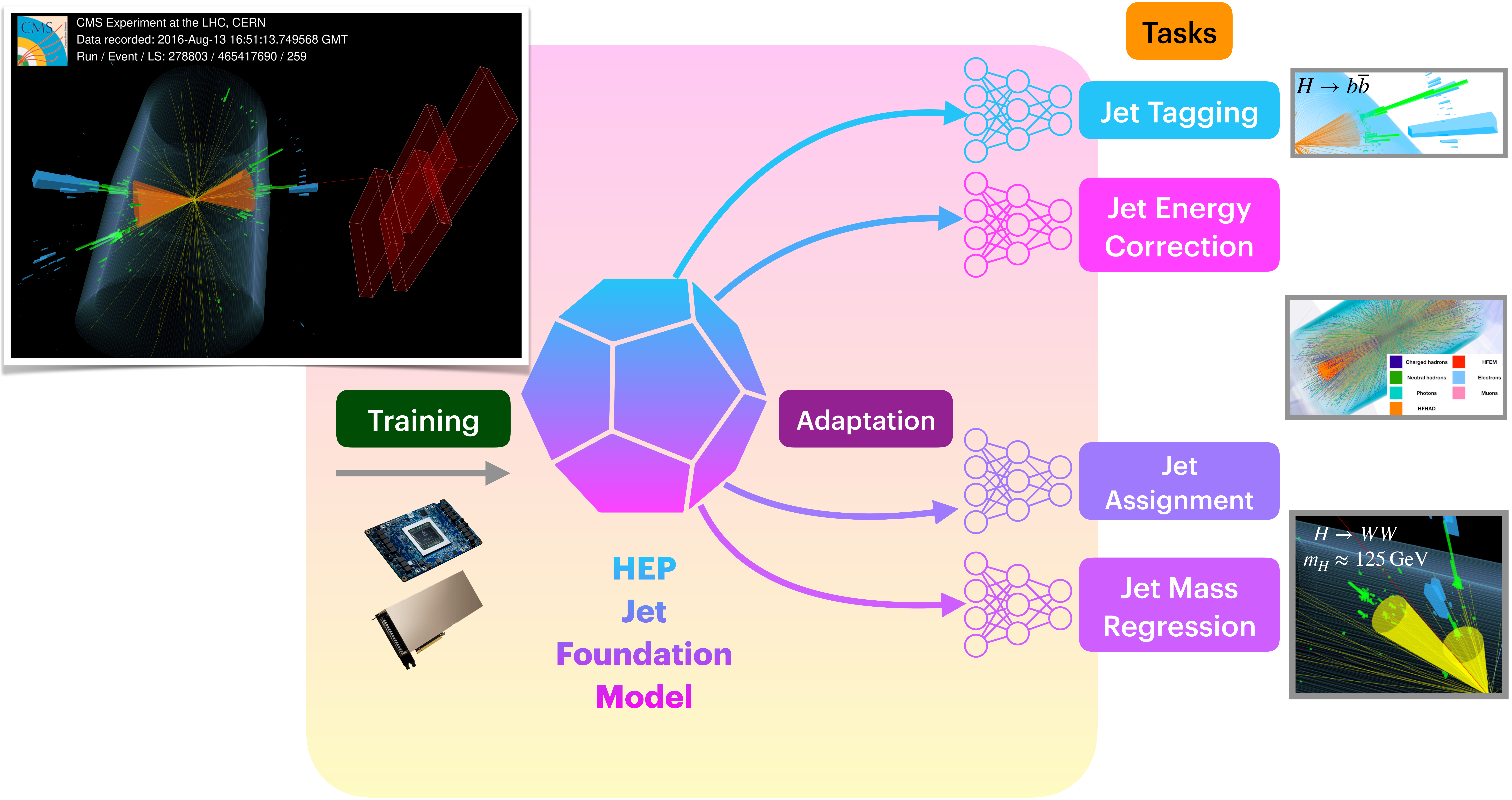


Current workflow using only Supervised Learning



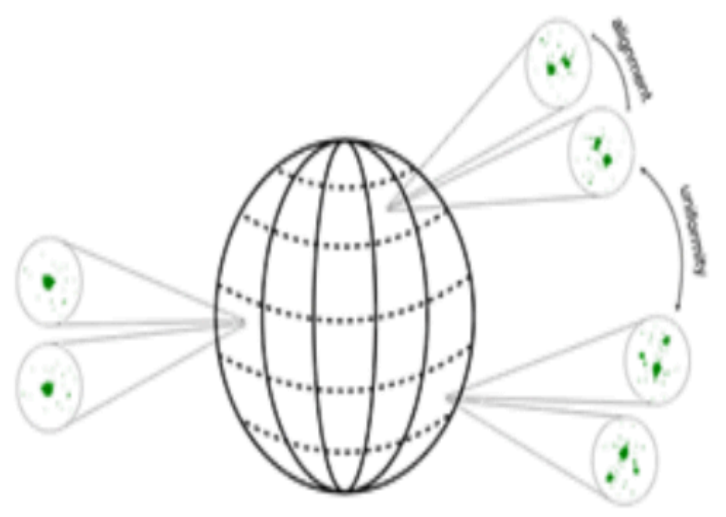
Workflow incorporating SSL

Toward Foundation Model



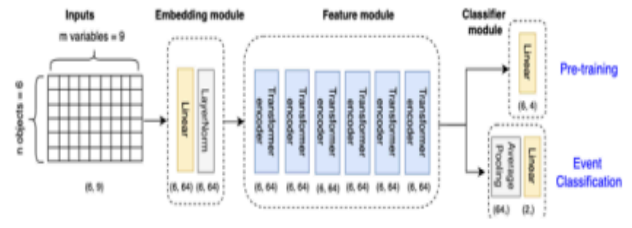
Towards Foundation Model in HEP

Contrastive Learning: Symmetry Augmentation



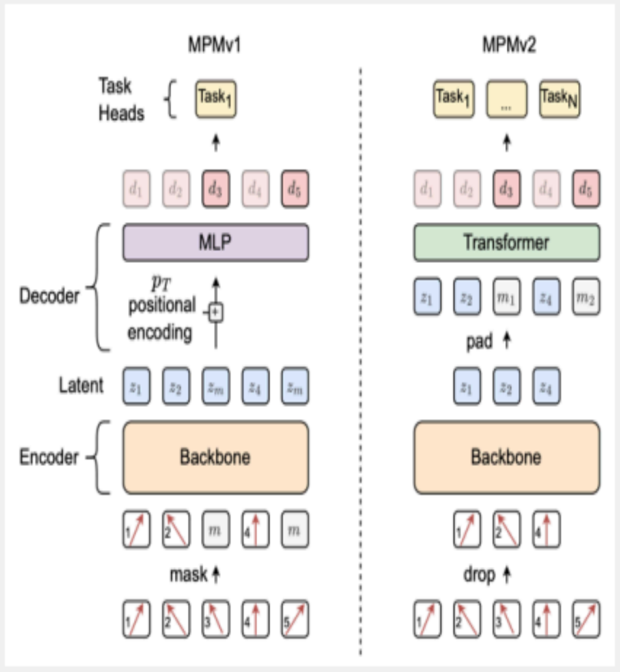
Dillon, Kasieczka, Olischlager
Plehn, Sorrenson, Vogel, 2108.04253

Masked Particle Type Prediction



Kishimoto, Morinaga, Saito
Tanaka, 2312.06909

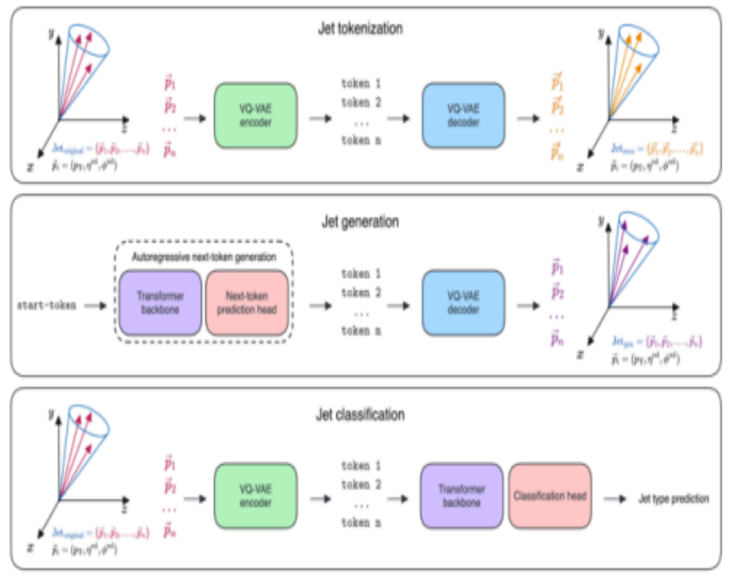
Masked Particle Modeling



Golling, Heinrich, **MK**, Klein, Leigh, Osadchy, Raine, 2401.13537

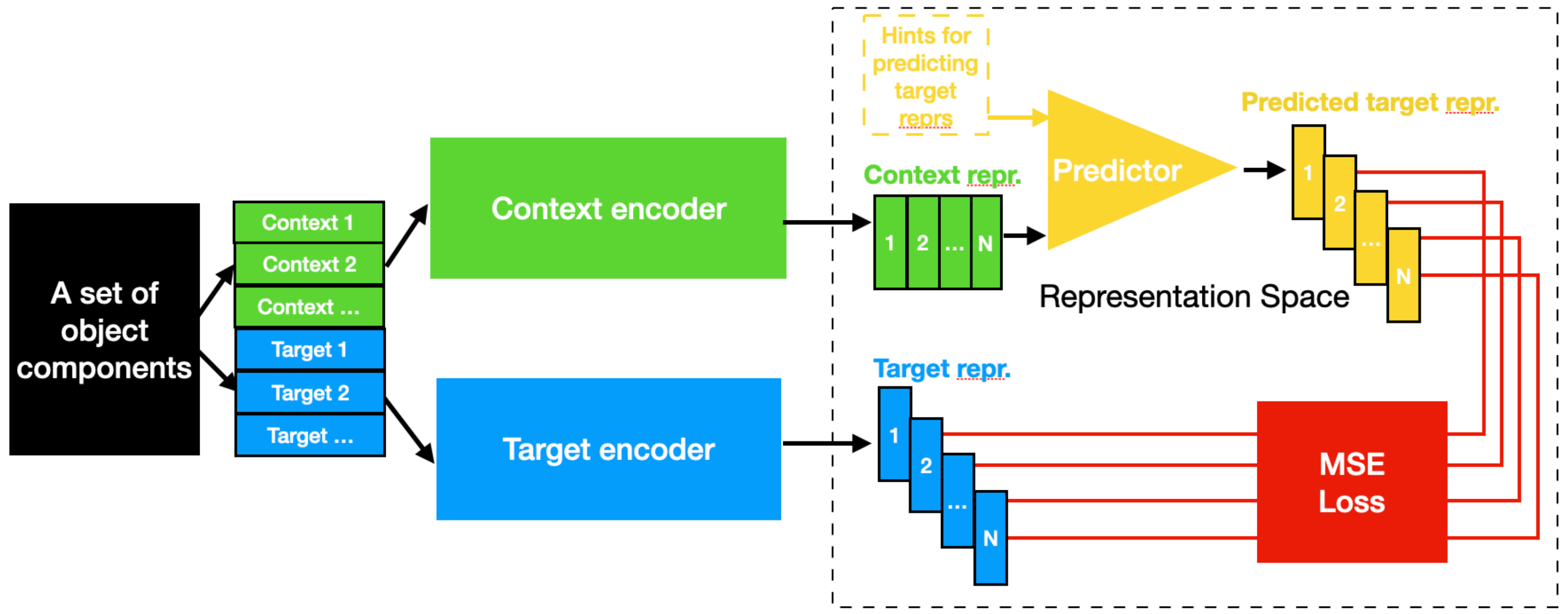
Leigh, Klein, Charton, Golling, Heinrich, **MK**, Ochoa, Osadchy, 2409.12589

Next Token Predictoin



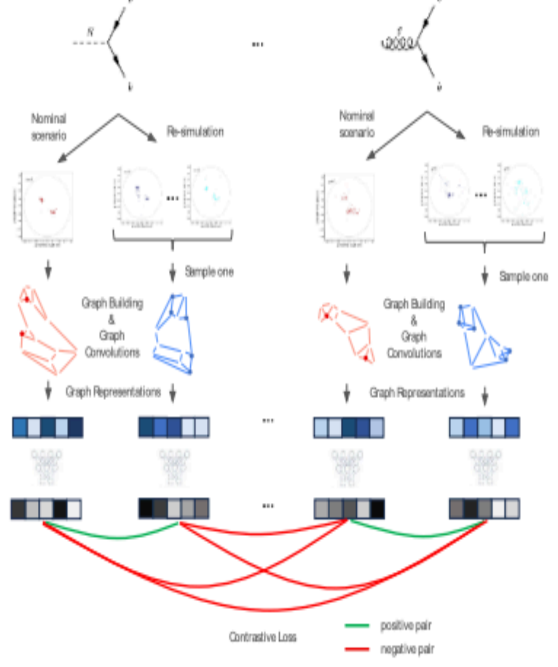
Birk, Hallin, Kasieczka, 2403.05618

J-JEPA



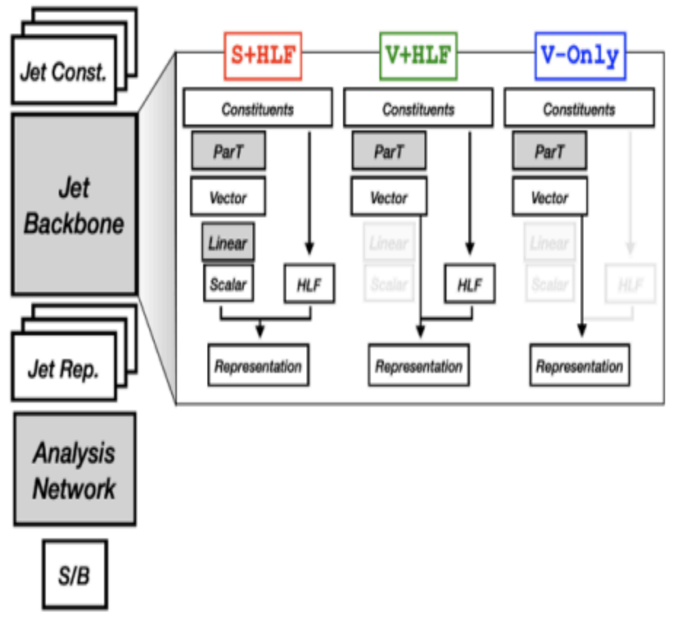
Katel, Li, Zhao, et al.
https://arxiv.org/abs/2412.05333

Contrastive Learning: Re-Simulation



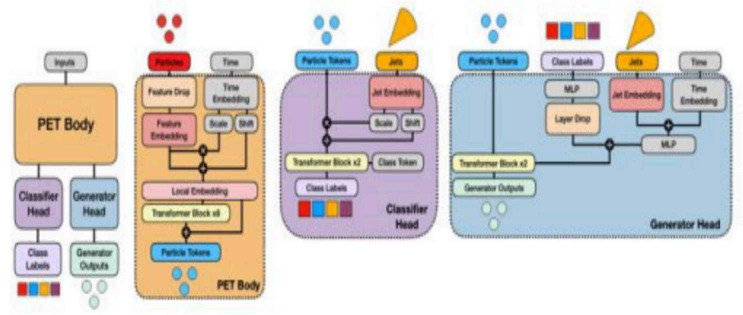
Harris, **MK**, Krupa, Maier, Woodward, 2403.07066

Supervised Pre-training and Joint Optimization



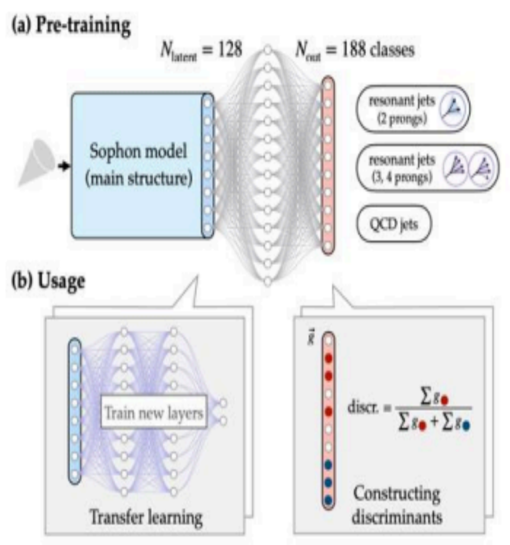
Vigl, Hartman, Heinrich, 2401.13536

Supervised Classification and Generation



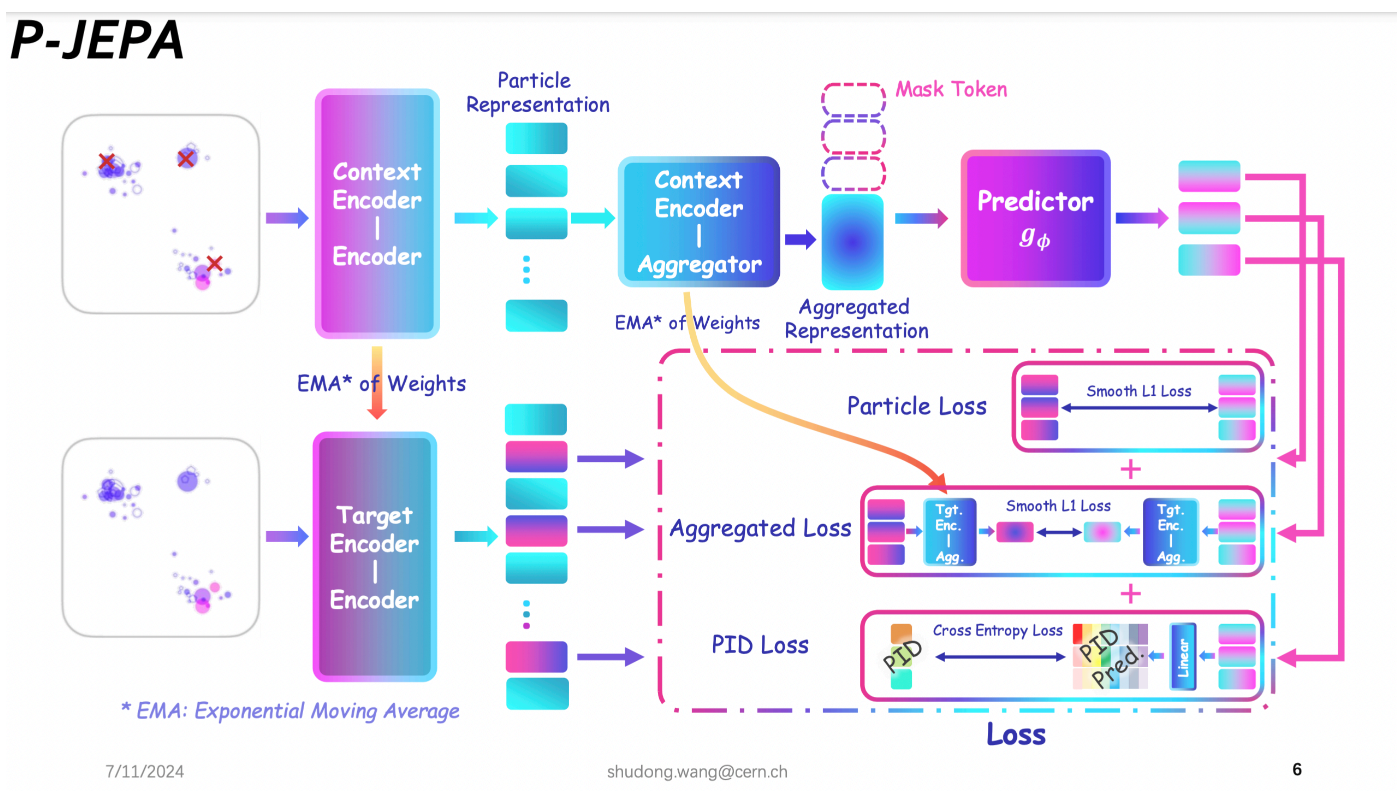
Mikuni, Nachman 2404.16091

Large-Scale Fine-Grained Classification



Li, Li, et al. 2405.12972

...

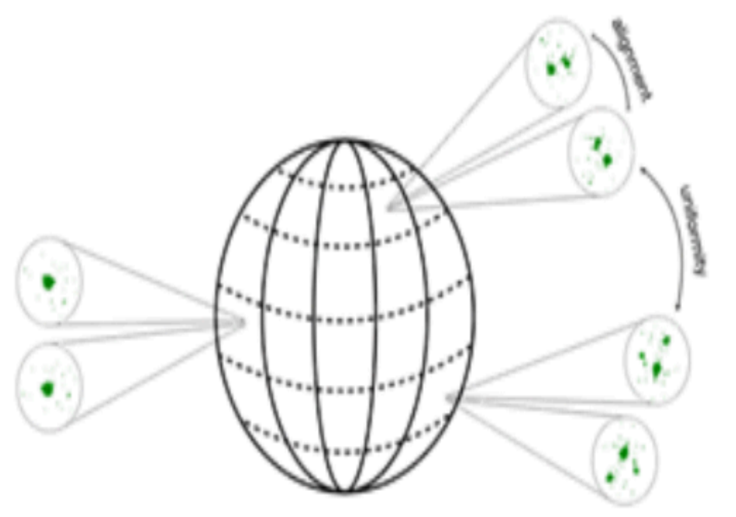


https://indico.cern.ch/event/1386125/contributions/6139666/

Credit: This slide is copied from Michael Kagan's talk in the FM Mini Workshop in October 2024

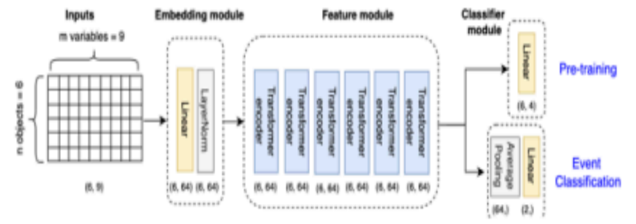
Towards Foundation Model in HEP

Contrastive Learning: Symmetry Augmentation



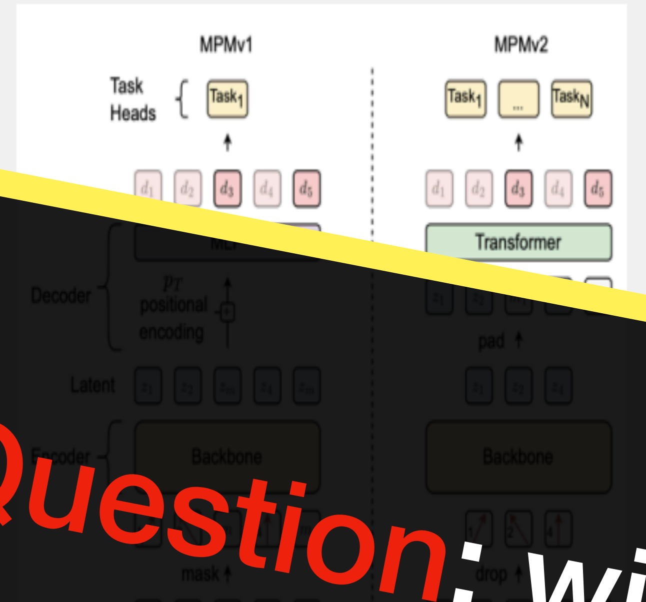
Dillon, Kasieczka, Olischlager
Plehn, Sorrenson, Vogel, 2108.04253

Masked Particle Type Prediction



Kishimoto, Morinaga, Saito
Tanaka, 2312.06909

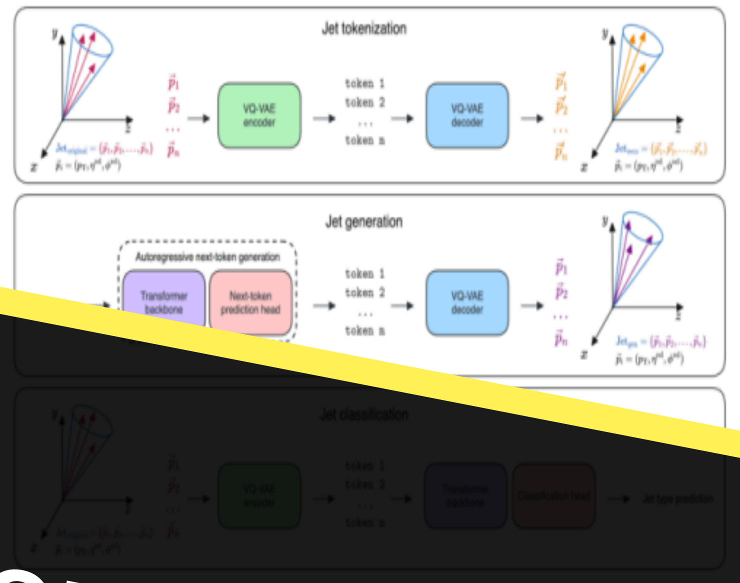
Masked Particle Modeling



Golling, Heinrich, ...
Leigh, Osadchy, Raine,
2401.13537

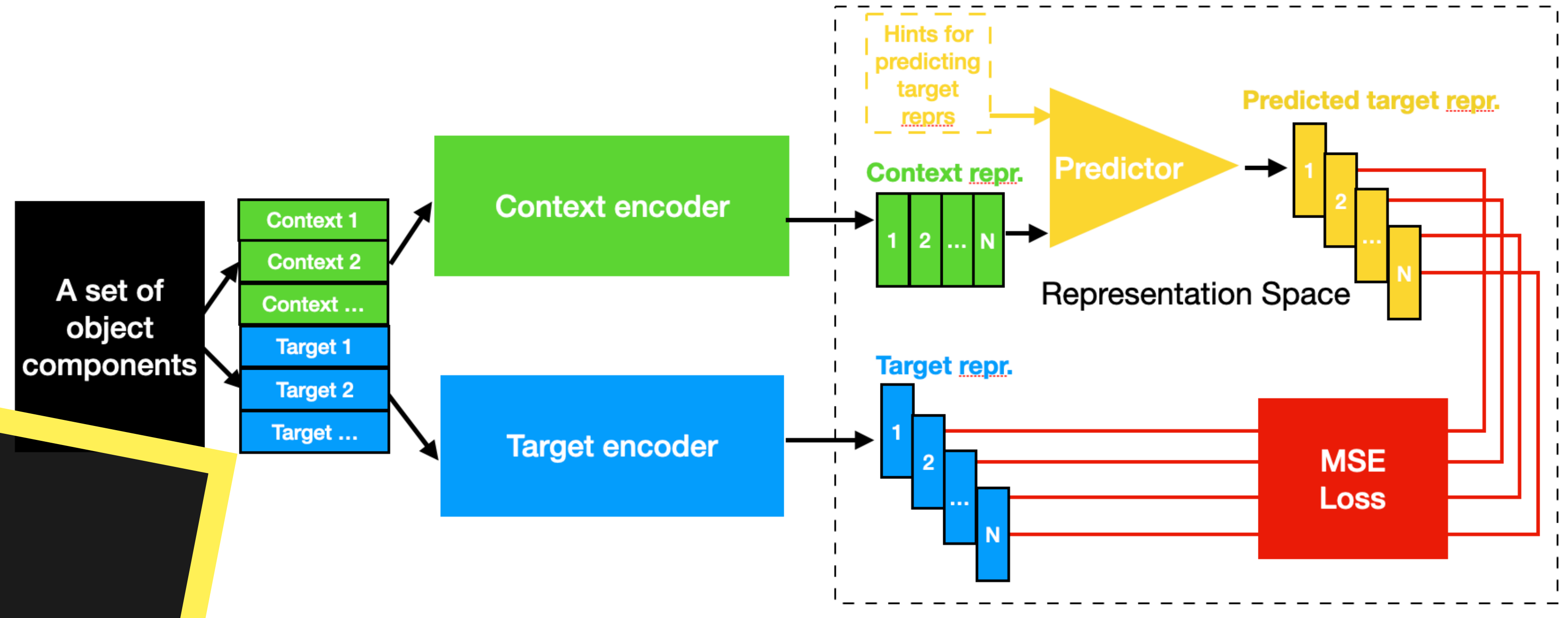
2409.12589

Next Token Predictoin



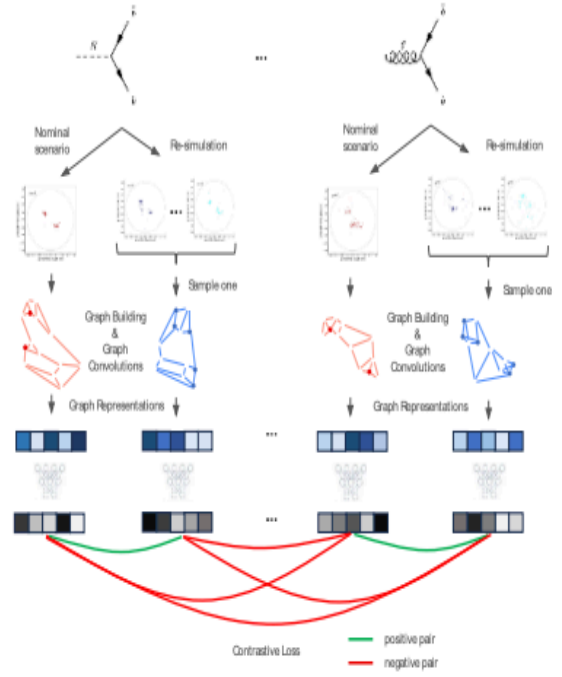
Question: will foundation models in HEP benefit from large scale pretraining?

J-JEPA



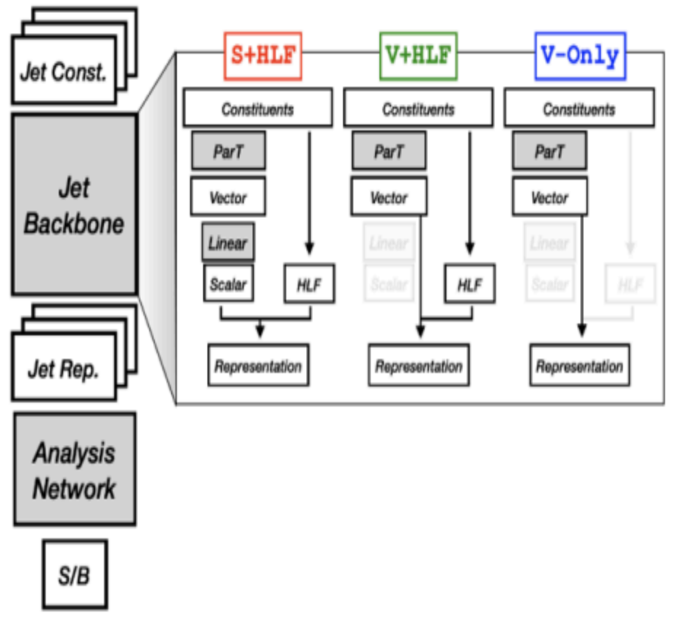
Katel, Li, Zhao, et al.
https://indico.cern.ch/event/1386125/contributions/6083379/

Contrastive Learning: Re-Simulation



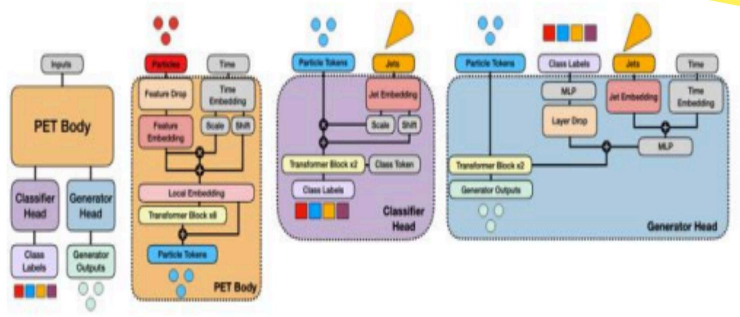
Harris, **MK**, Krupa, Maier, Woodward, 2403.07066

Supervised Pre-training and Joint Optimization



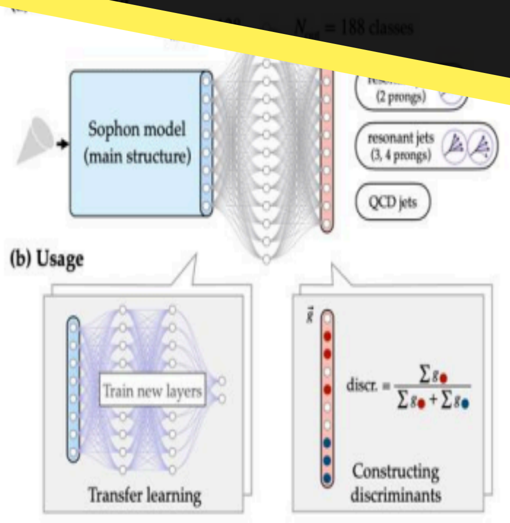
Vigl, Hartman, Heinrich, 2401.13536

Supervised Classification and Generation



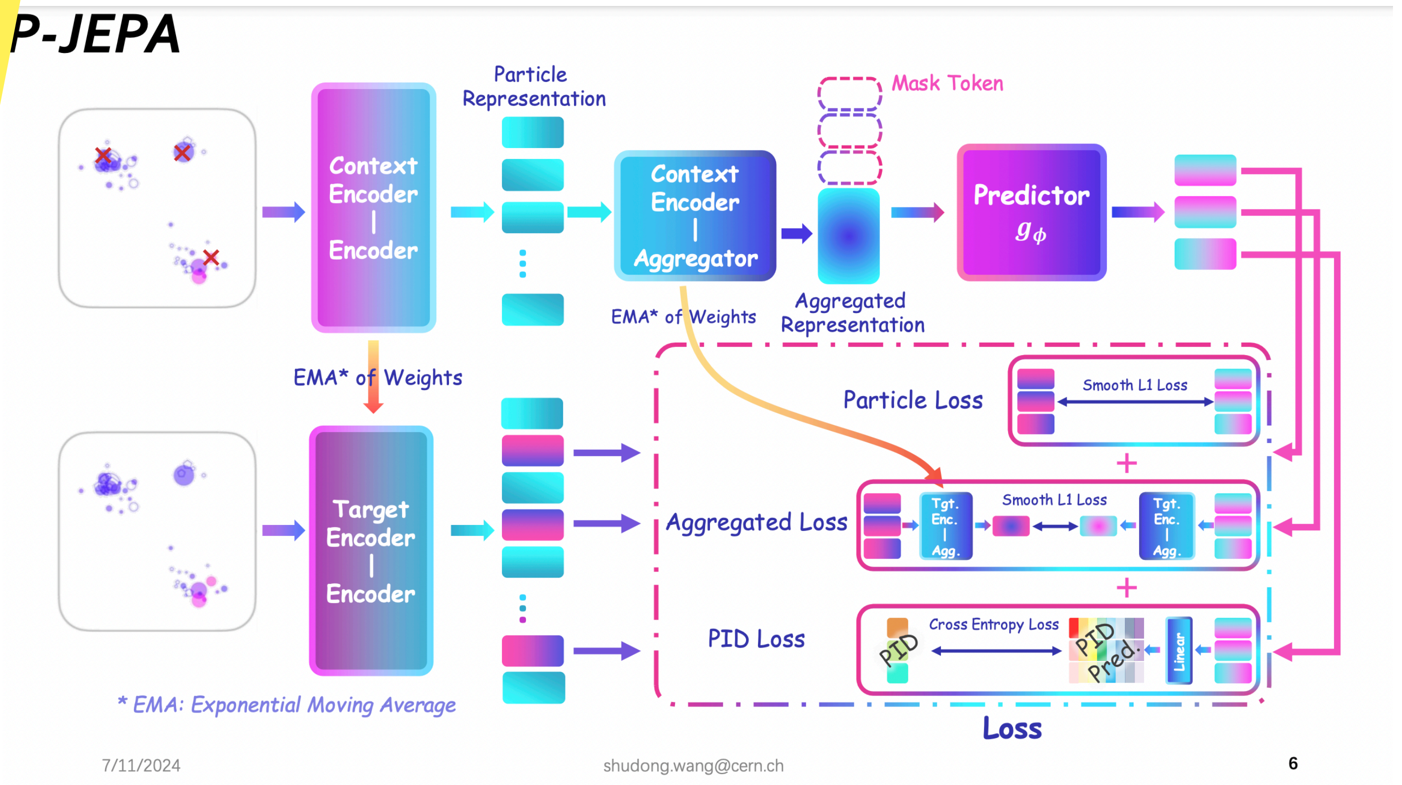
Mikuni, Nachman 2404.16091

Large Scale Fine-Grained Classification



Li, Li, et al. 2405.12972

P-JEPA

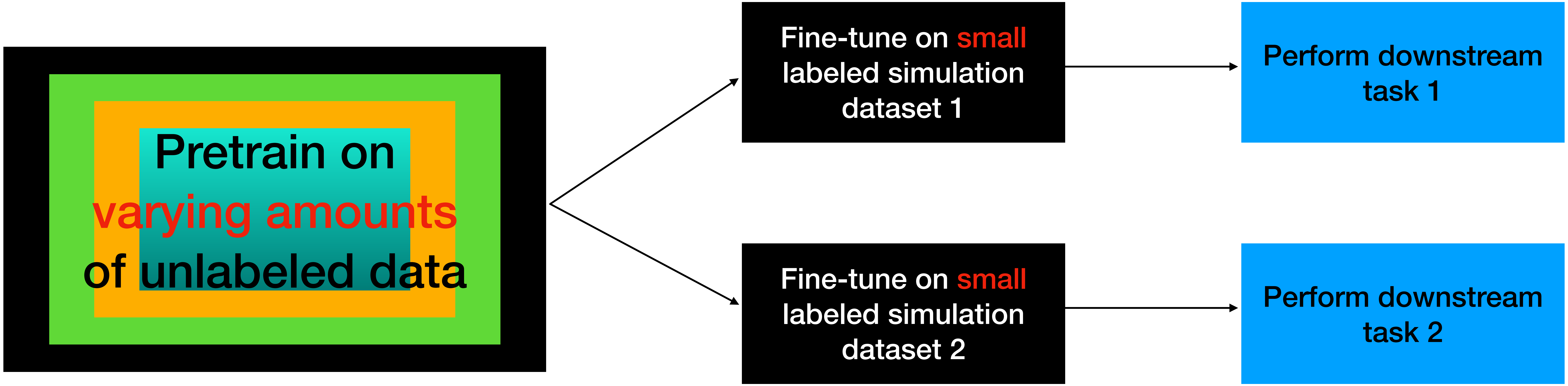


https://indico.cern.ch/event/1386125/contributions/6139666/

Credit: This slide is copied from Michael Kagan's talk in the FM Mini Workshop in October 2024

Primary Goal of the Project

- Focus on studying the effect of **scaling up** the sizes of pretraining datasets on the performance of foundation model.

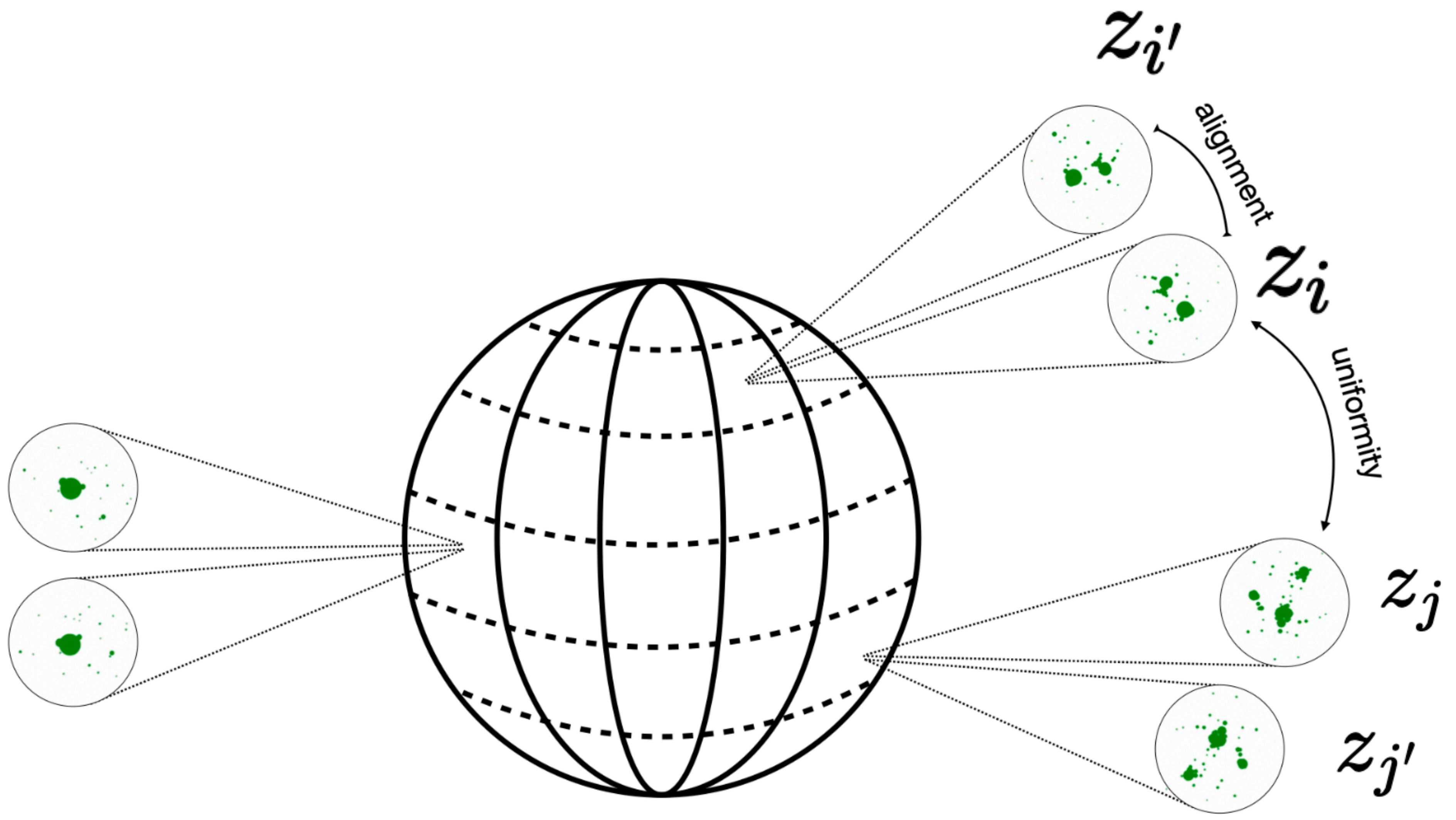


Outline

- Toward Foundation Model in HEP
- Goals of the Project
- Intro to JetCLR
- Transfer Learning: from JetClass to Top Tagging
- Scaling up pretraining dataset size
- Some technical details
    - Classification head for finetuning: MLP vs Linear Projection
    - Techniques to speed up training
- Ongoing and Future work

Intro to JetCLR

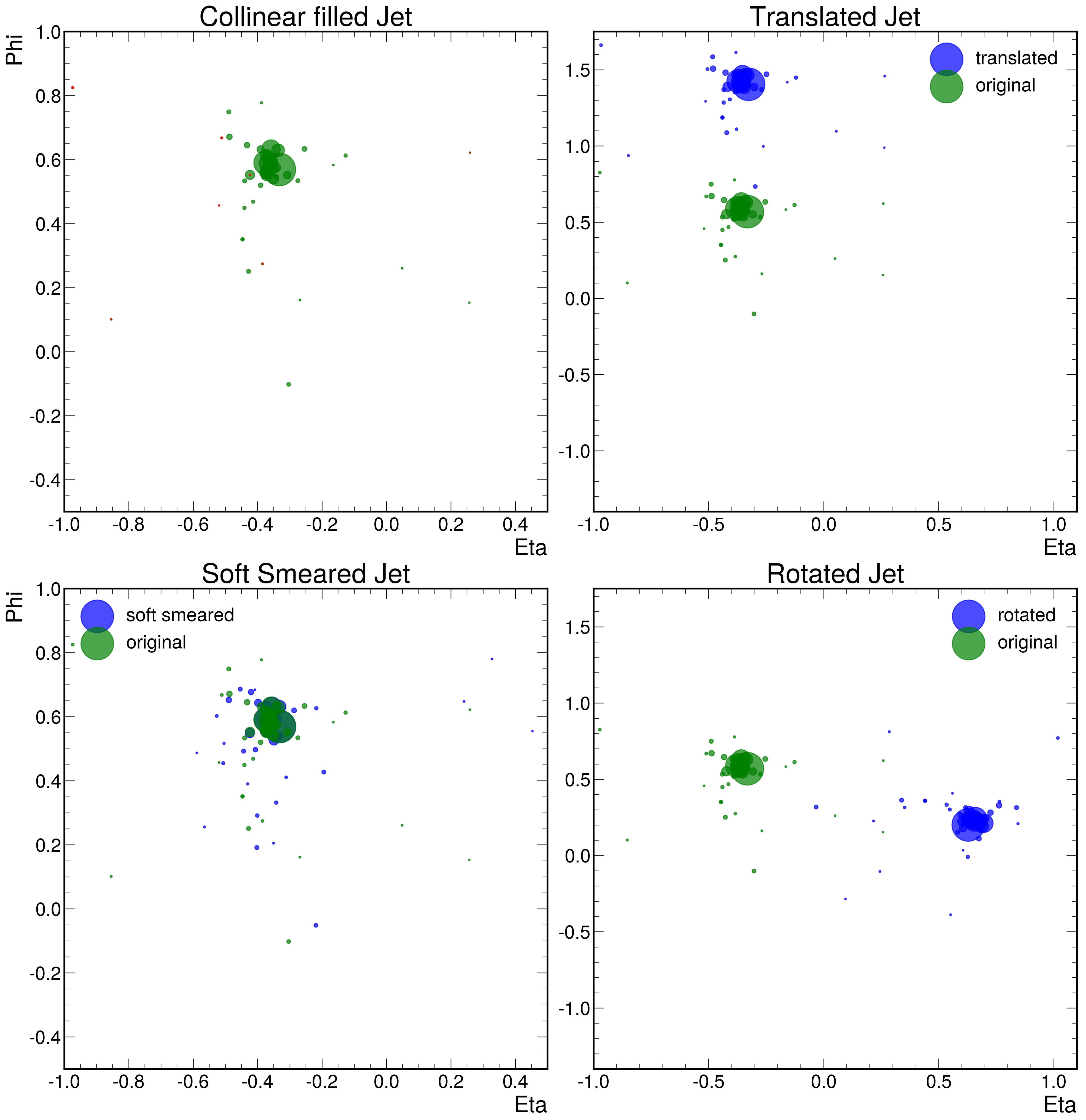
SimCLR loss



$$\mathcal{L}_i = -\log \frac{e^{s(z_i, z_i')/\tau}}{\sum_{j \neq i \in \text{batch}} \left[e^{s(z_i, z_j)/\tau} + e^{s(z_i, z_j')/\tau} \right]}$$

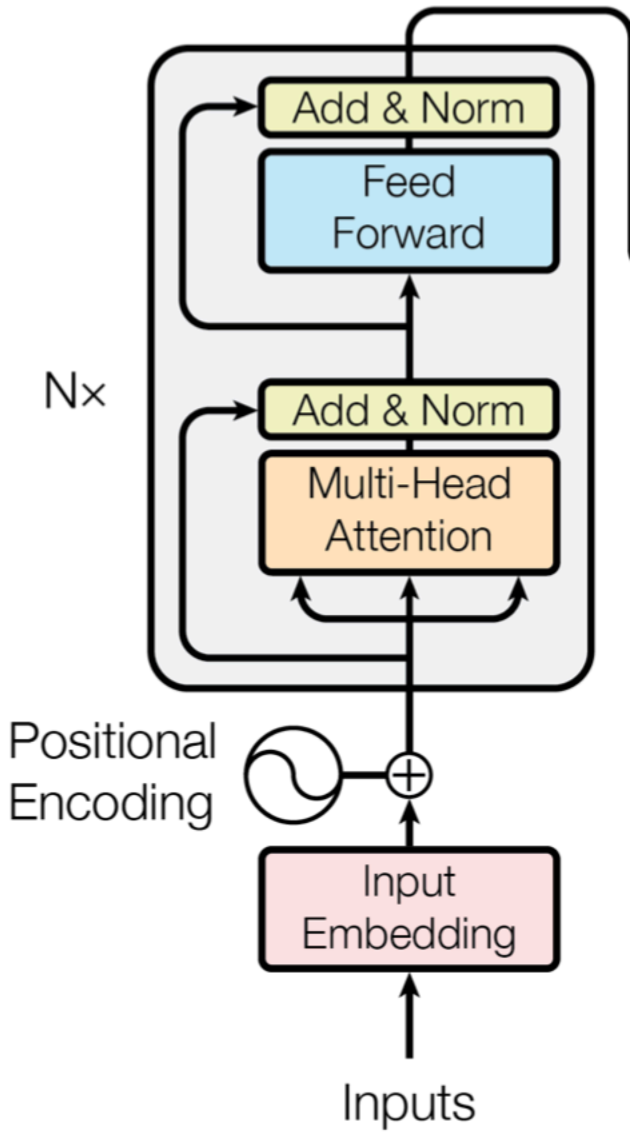
2108.04253

Augmentations



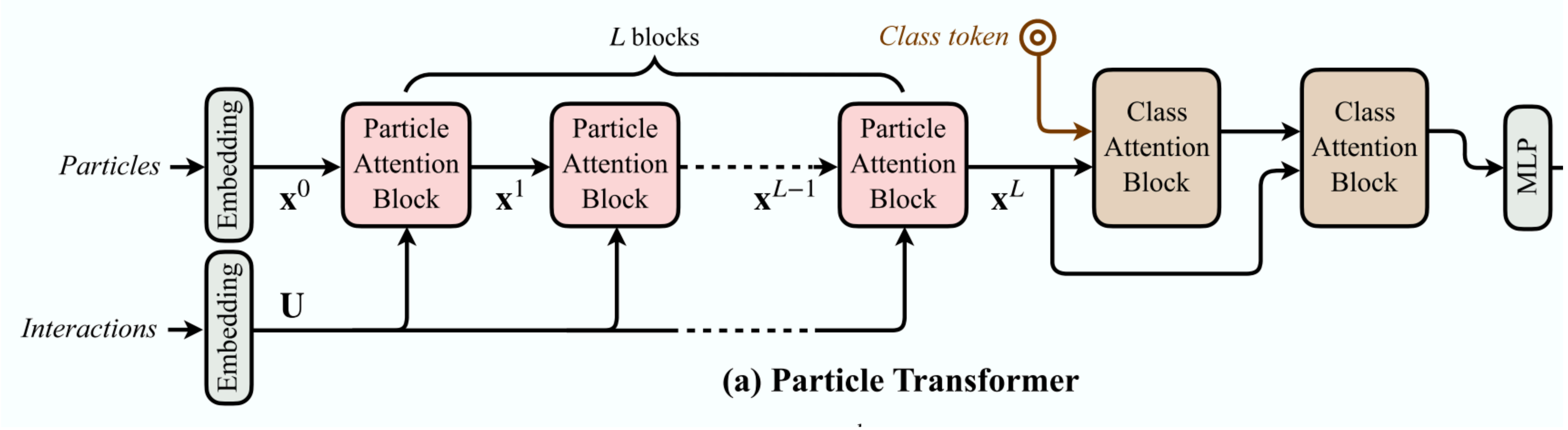
Model Architecture for encoder

- Started with a simple Transformer encoder
- Working on switching to more advanced architectures such as Particle Transformer



Transformer Encoder

1706.03762



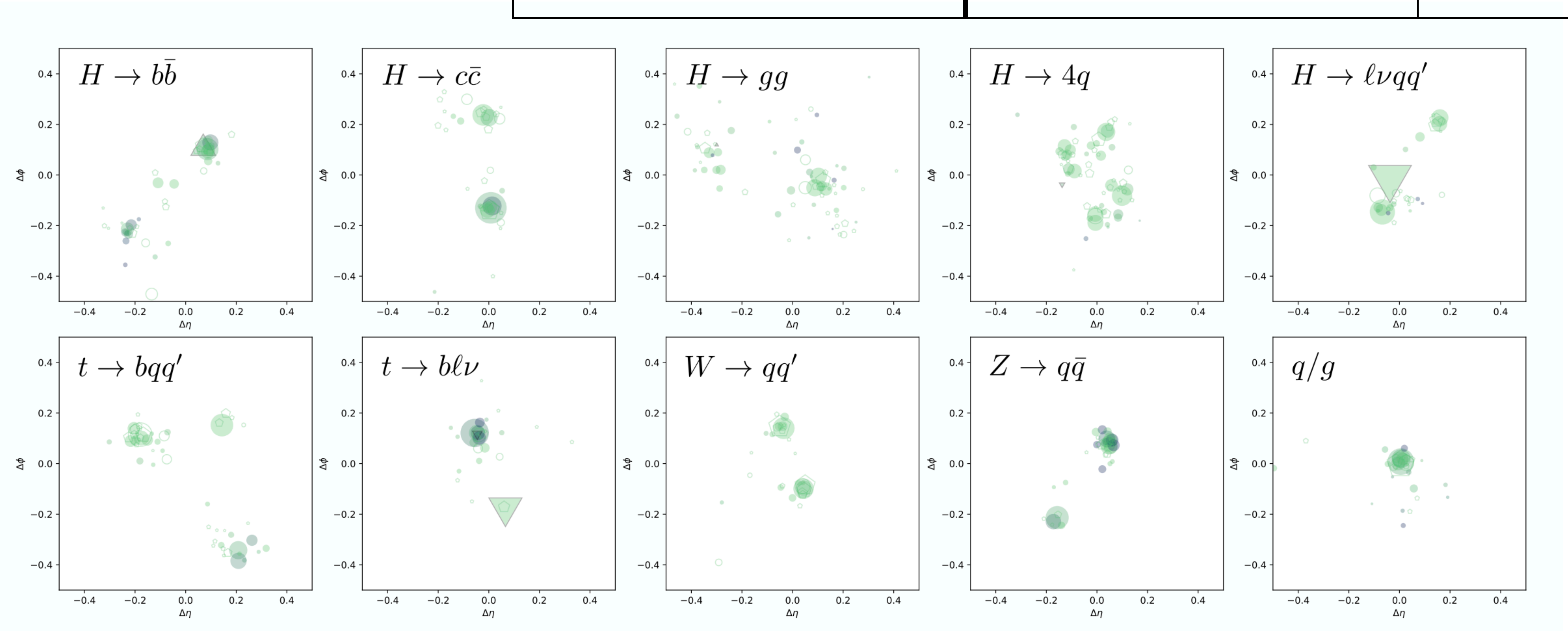
Particle Transformer

2202.03772

Datasets

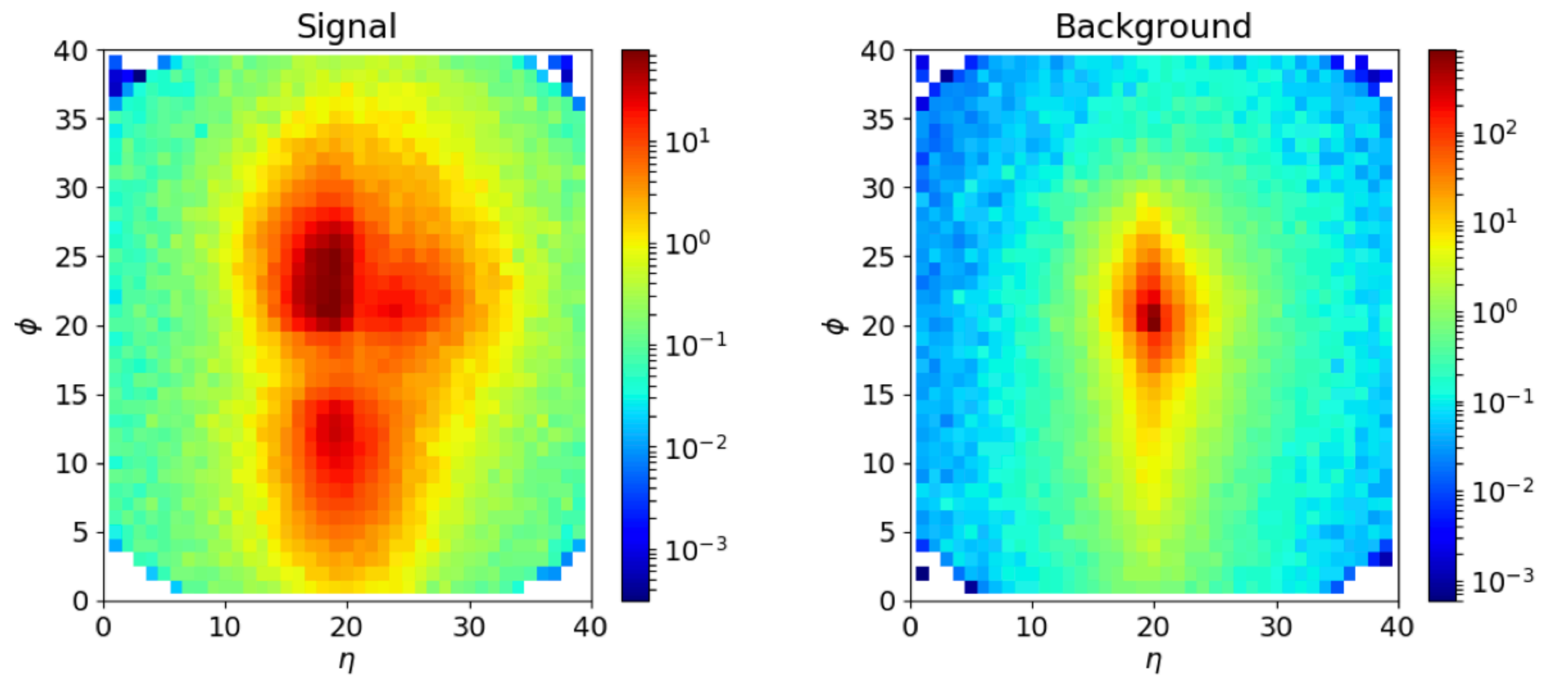
JetClass for unlabeled pretraining, Top Tagging for labeled finetuning

Dataset name	Size	Description	Role in transfer learning
JetClass Dataset	100 Million Jets	Contains 10 classes of jets	Stand in for unlabeled “data”, use for pretraining
Top Tagging Dataset	1.2 Million Jets	Only Top and QCD jets	Stand in for labeled “simulation”, use for fine-tuning



JetClass Dataset

2202.03772



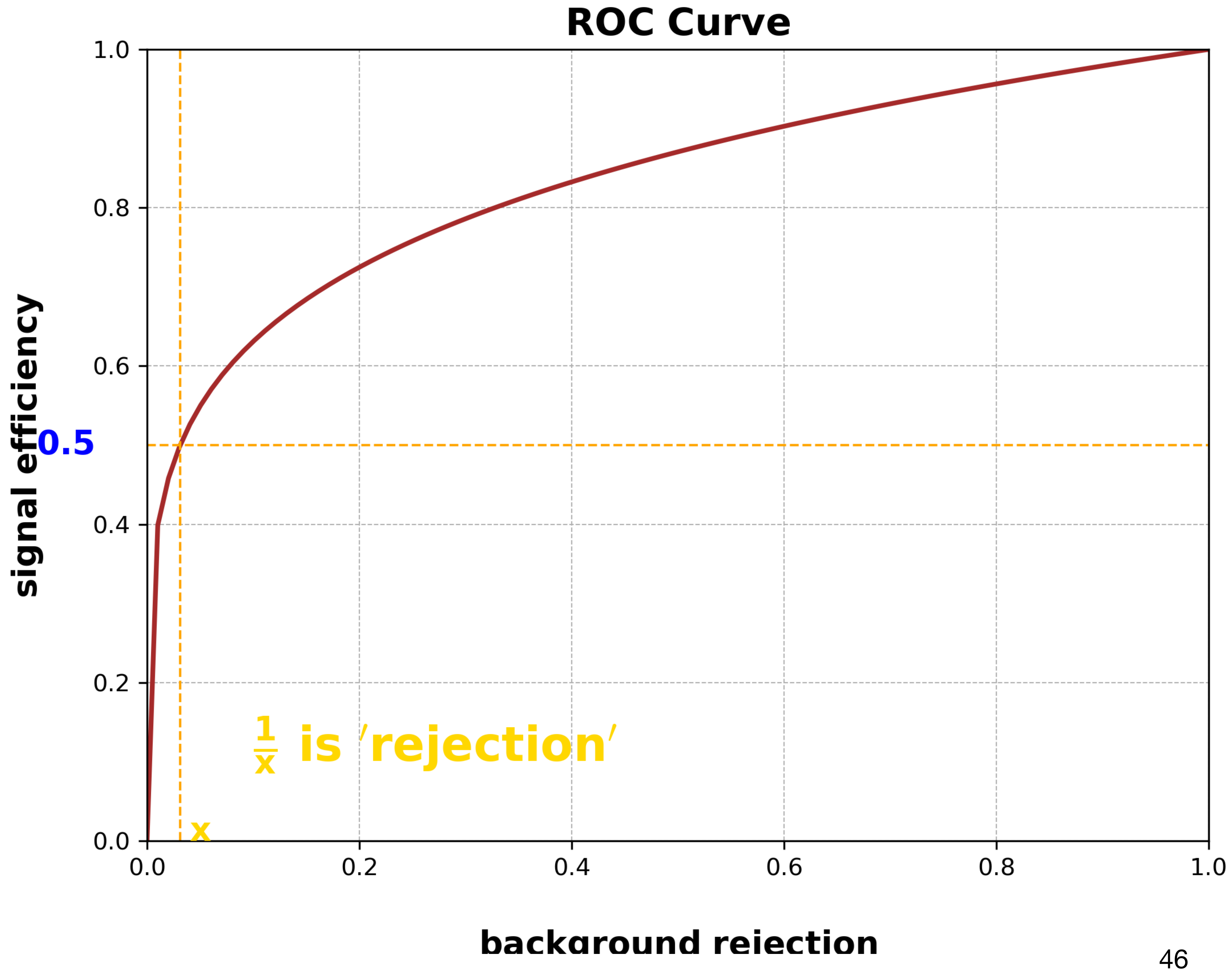
Top Tagging Dataset

1902.09914

Metrics

Accuracy: correctly predicted / total number of samples

Rejection: inverse of background rejection (FPR) at 50% signal efficiency (TPR)

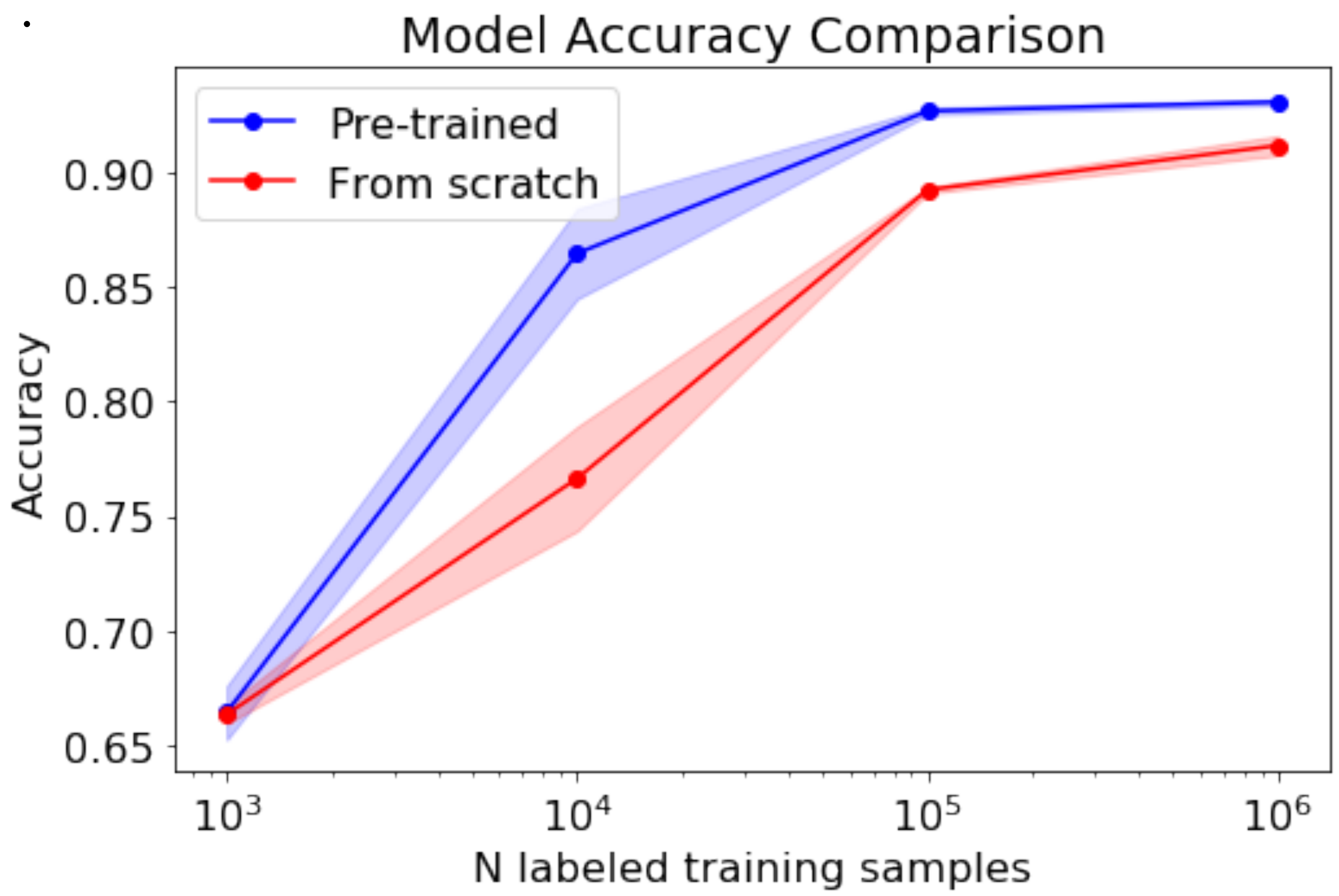


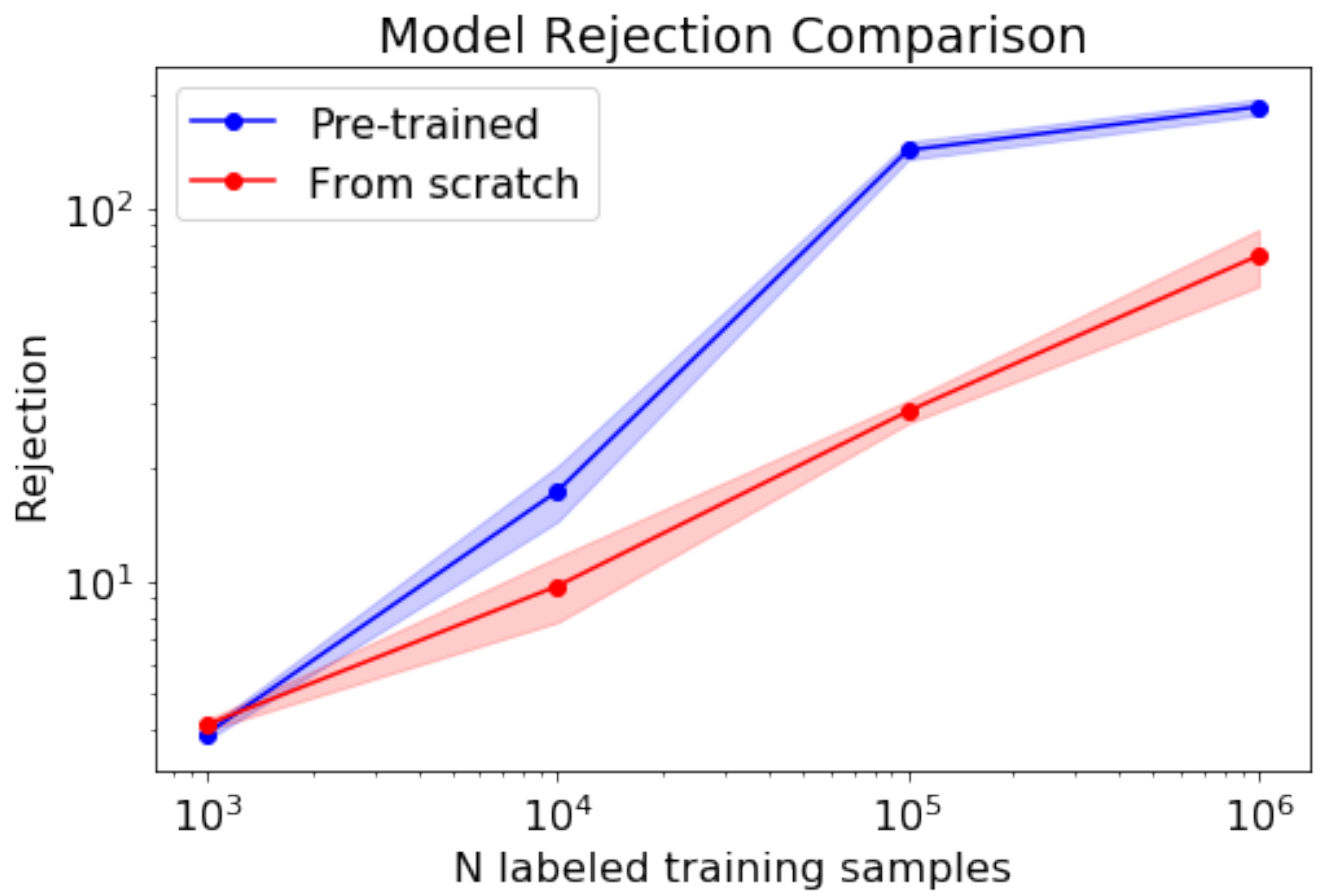
Significance: In a background dominant dataset, how much background you can reject while letting in a certain fraction of signal samples (the more the better)

Pretraining on JetClass and fine-tuning on Top Tagging

The pre-trained model requires significantly fewer samples to achieve high accuracy and rejection rate: higher data efficiency

- The averages and standard deviations over 5 trainings are shown in solid lines and uncertainty bands, respectively





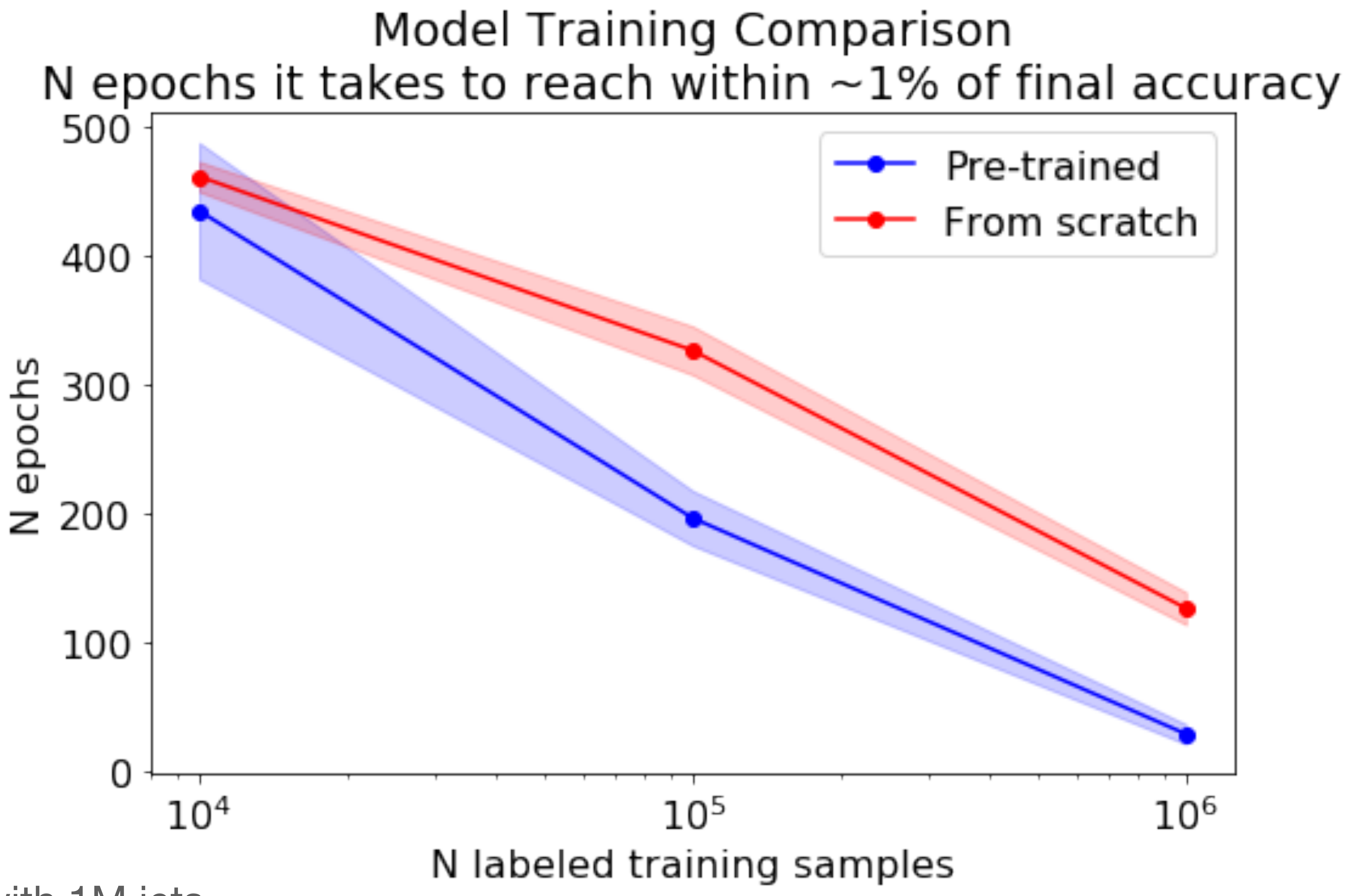
Pretrained: pretrained with 1M jets

Rejection: inverse of background rejection at 50% signal efficiency

Pretraining on JetClass and fine-tuning on Top Tagging

The pre-trained model converges much faster: higher computational efficiency

- The averages and standard deviations over 5 trainings are shown in solid lines and uncertainty bands, respectively

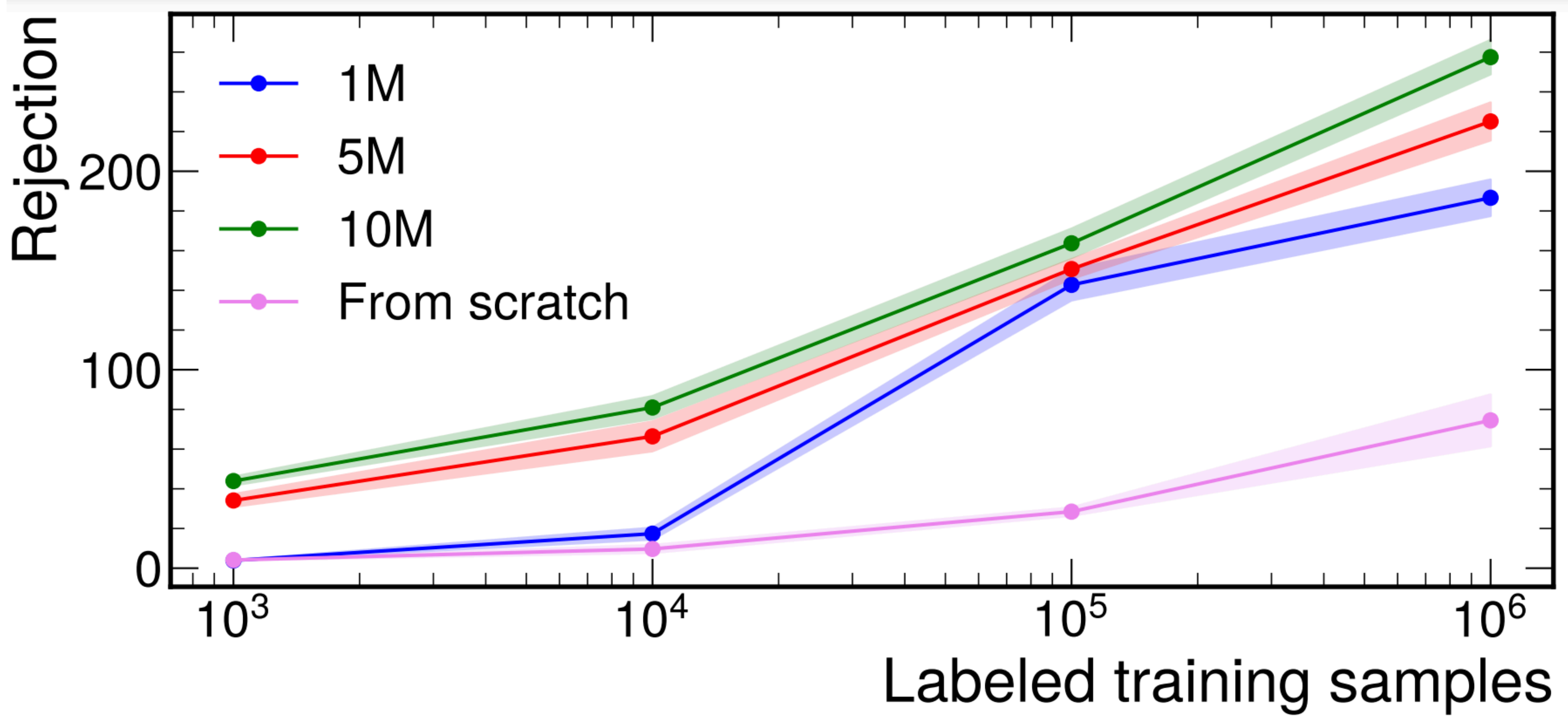


Pretrained: pretrained with 1M jets

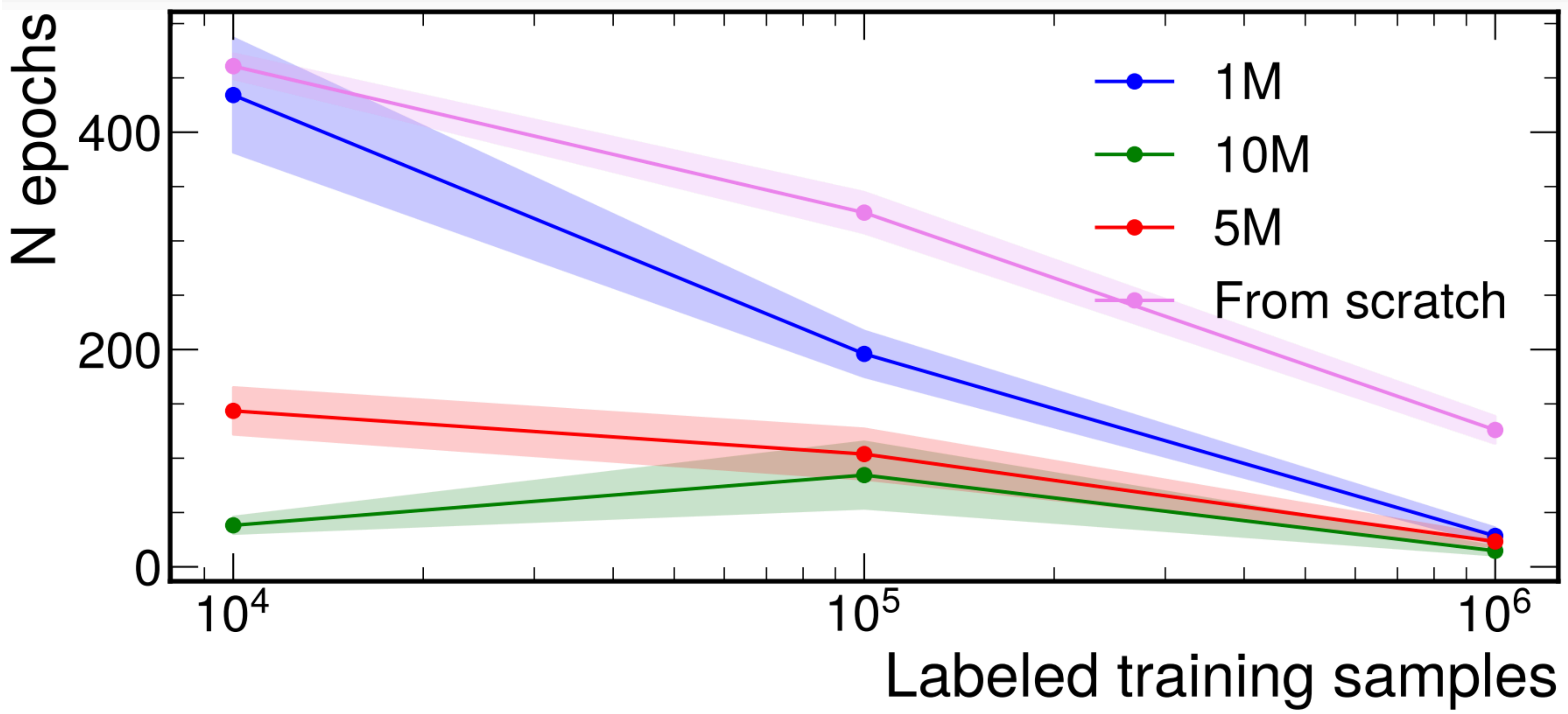
Scaling up pretraining dataset size

By scaling up the pretraining dataset, the model demonstrated enhanced performance and faster convergence: both data and computational efficiency improve as we use larger datasets for pretraining

Background rejection at 50% signal efficiency



Number of epochs required to reach within 1% of the final accuracy



Rejection: inverse of background rejection at 50% signal efficiency

Conclusion

- 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.
  - **Greater computational efficiency—**enabling the model to converge significantly faster than its fully supervised counterpart.
  - **Both efficiencies increase as the pretraining dataset size increases.**
- This paves the way for the use of unlabeled data in HEP and contributes to a better understanding of the potential of SSL for scientific discovery.

Ongoing and Future work

- Ongoing work
  - Study the effectiveness of more advanced architectures like the ParticleTransformer as the backbone encoder
- Pretrain on JetClass v2, an even larger dataset, or the Aspen Open Jets dataset, a real CMS dataset
- Evaluate on different SSL strategies beyond JetCLR
- Explore other physically motivated augmentations
  - Pairing the two jets from dijet events
  - Using two subjets clustered with smaller radii
  - Using tracks and clusters as two views of the same jet
  - …

Support

Thank you for listening!

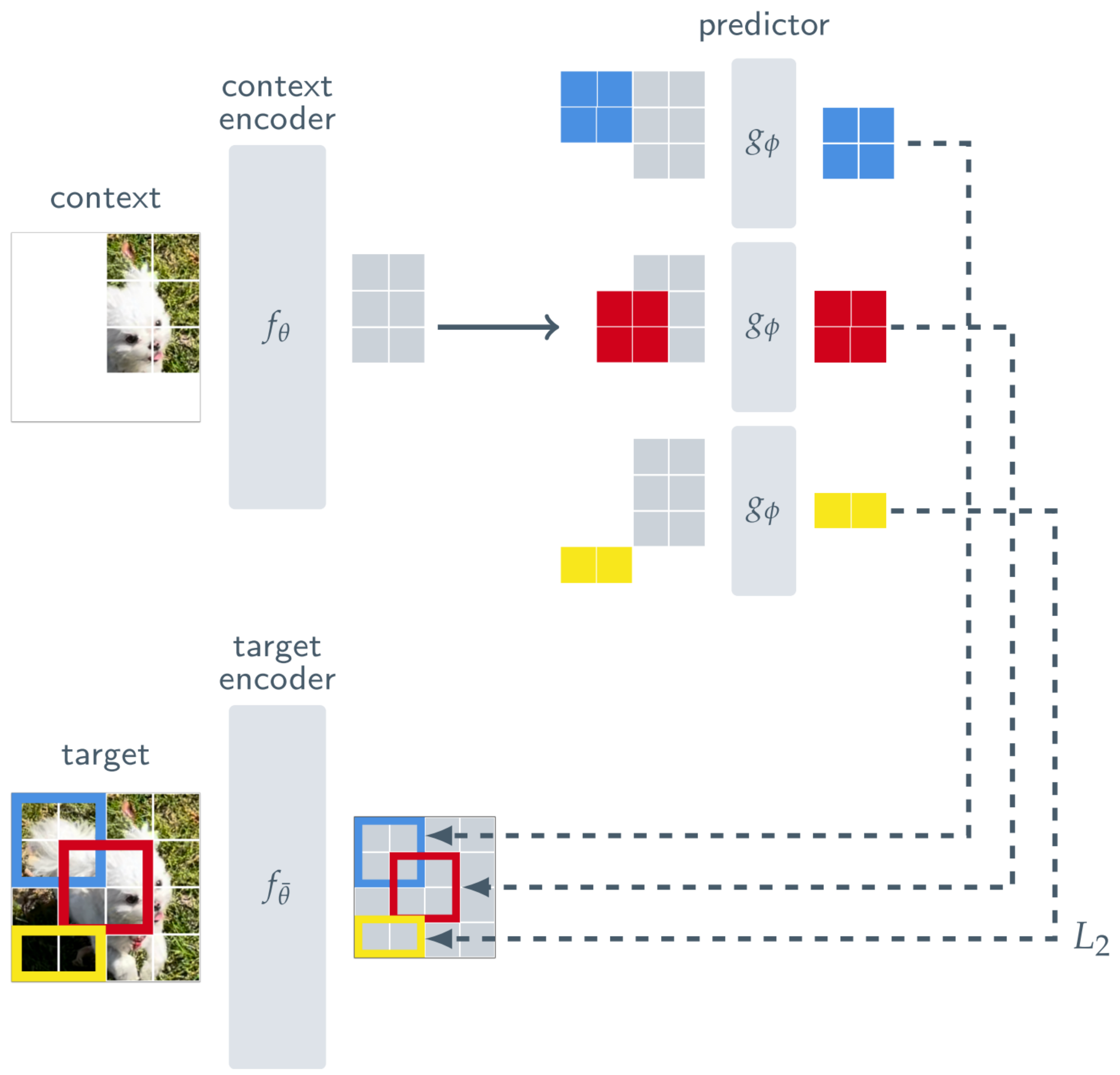
- This work is supported by the National Science Foundation under award number 2117997 (A3D3 Institute), Research Corporation For Science Advancement, the Alfred P. Sloan Foundation, and the U.S. Department of Energy

- This work was performed using the National Research Platform Nautilus HyperCluster supported by NSF

Back Up (Part 1)

Example: The I-JEPA Architecture

I: Image



Details of the Top Tagging Dataset

The top signal and mixed quark-gluon background jets are produced with using Pythia8 [25] with its default tune for a center-of-mass energy of 14 TeV and ignoring multiple interactions and pile-up. For a simplified detector simulation we use Delphes [26] with the default ATLAS detector card. This accounts for the curved trajectory of the charged particles, assuming a magnetic field of 2 T and a radius of 1.15 m as well as how the tracking efficiency and momentum smearing changes with η. The fat jet is then defined through the anti-k_T algorithm [27] in FastJet [28] with $R = 0.8$. We only consider the leading jet in each event and require

$$p_{T,j} = 550 \text{ .... } 650 \text{ GeV} . \tag{1}$$

For the signal only, we further require a matched parton-level top to be within $\Delta R = 0.8$, and all top decay partons to be within $\Delta R = 0.8$ of the jet axis as well. No matching is performed for the QCD jets. We also require the jet to have $|\eta_j| < 2$. The constituents are extracted through the Delphes energy-flow algorithm, and the 4-momenta of the leading 200 constituents are stored. For jets with less than 200 constituents we simply add zero-vectors.

1902.09914

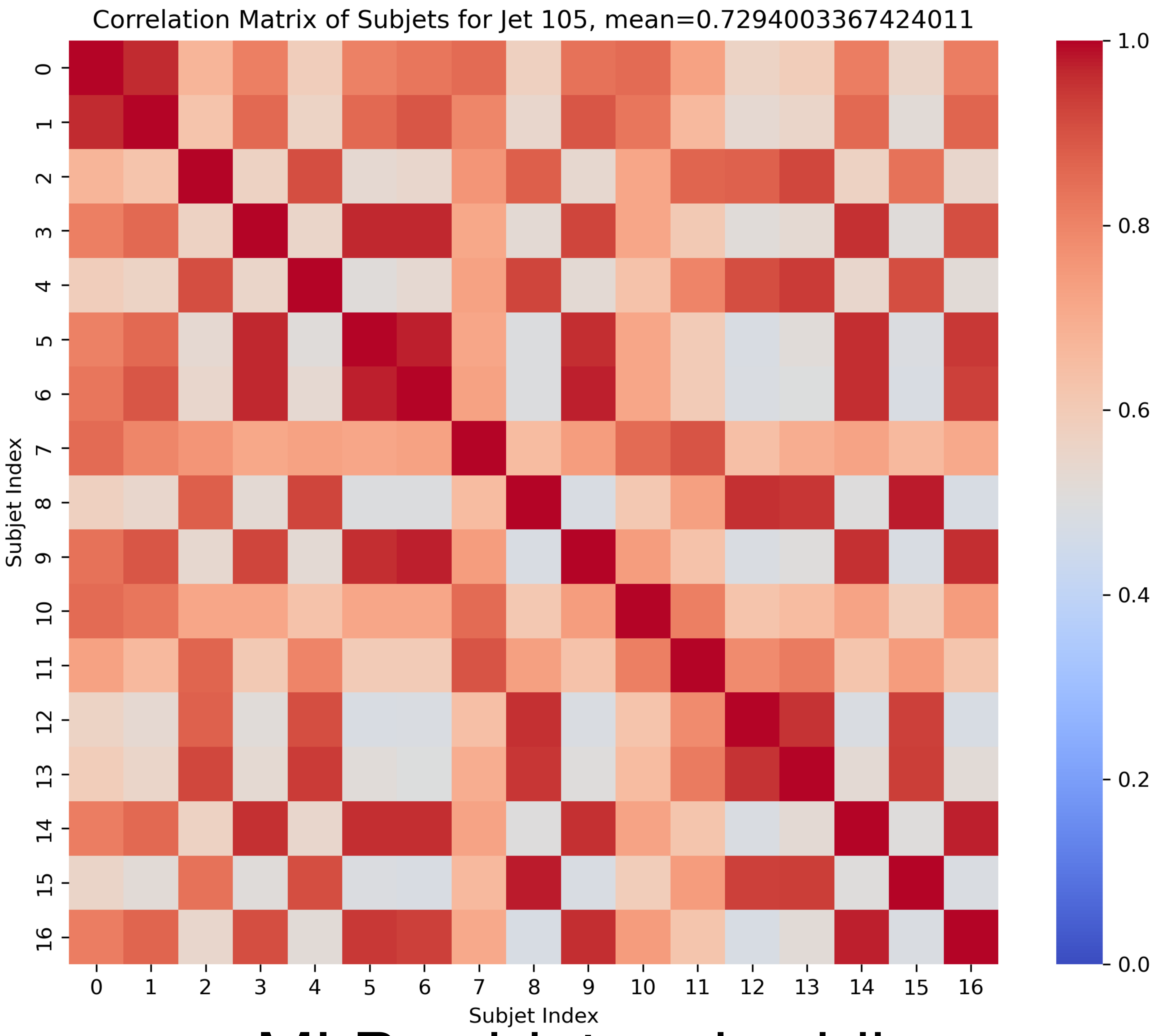
Details of the JetClass Dataset

Simulation setup. Jets in this dataset are simulated with standard Monte Carlo event generators used by LHC experiments. The production and decay of the top quarks and the W, Z and Higgs bosons are generated with MADGRAPH5_aMC@NLO (Alwall et al., 2014). We use PYTHIA (Sjöstrand et al., 2015) to evolve the produced particles, i.e., performing parton showering and hadronization, and produce the final outgoing particles[1]. To be close to realistic jets reconstructed at the ATLAS or CMS experiment, detector effects are simulated with DELPHES (de Favereau et al., 2014) using the CMS detector configuration provided in DELPHES. In addition, the impact parameters of electrically charged particles are smeared to match the resolution of the CMS tracking detector (CMS Collaboration, 2014). Jets are clustered from DELPHES E-Flow objects with the anti-k_{T} algorithm (Cacciari et al., 2008; 2012) using a distance parameter $R = 0.8$. Only jets with transverse momentum in 500–1000 GeV and pseudorapidity $|\eta| < 2$ are considered. For signal jets, only the "high-quality" ones that fully contain the decay products of initial particles are included[2].

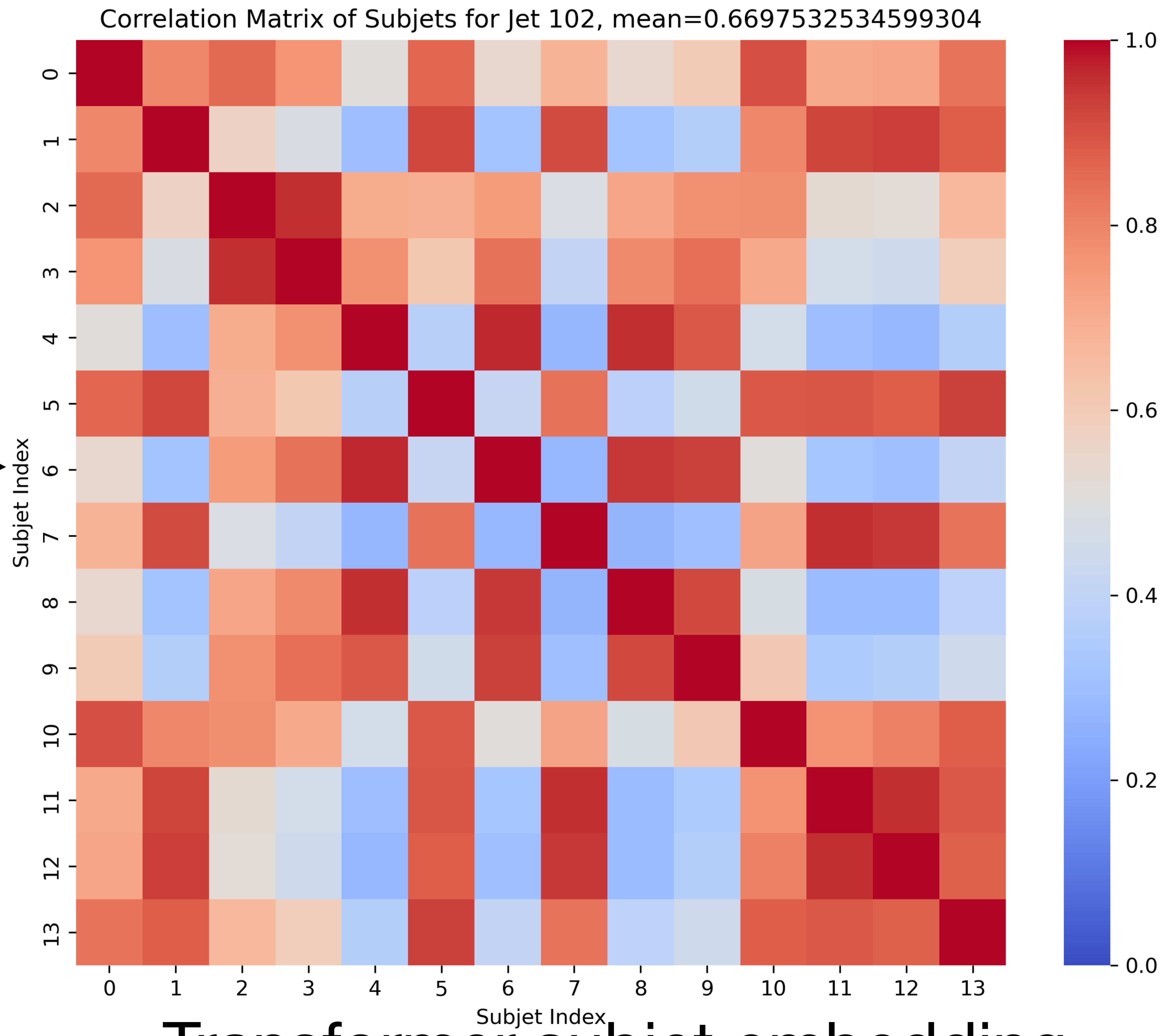
2202.03772

Transformer Embedding Layer Effects

Correlation between subjets is reduced



MLP subjet embedding



Transformer subjet embedding

WIP: Study of how to provide the additional info

Pre-train and fine-tune on Top Tagging

Experiments	Encode subjet coordinates at both (encoder and predictor)	Encode coordinates only at predictor	Encode pT ranking at both	Use a MLP to encode subjet coordinates
Inverse Rejection Power	63.99	45.33	45.02	Converging…

Study of subjet embedding

Pre-training and fine-tuning on Toptagging dataset

Inverse Rejection Power	Dimension Reduction	Dimension Expansion
Attention	**86.42**	73.81
MLP	73.55	63.99
Linear	44.31	

Strategies to prevent collapse

- Targets being padded subjets
- Most particles are padded so all subjets look the same to the model
- Information bottleneck in the predictor is too big
- Dataset was not normalized

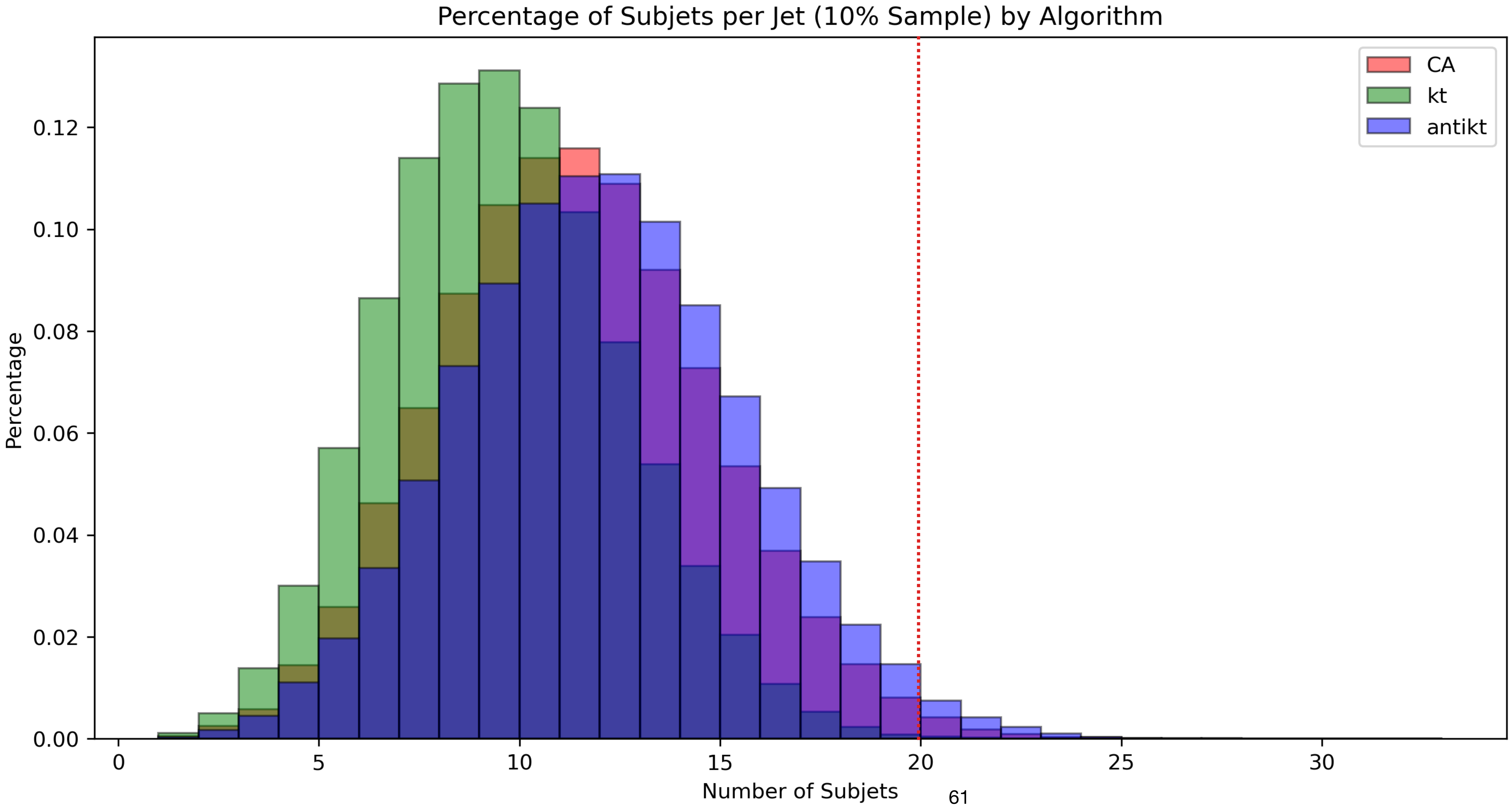
→

- We only select targets from non-empty subjets
- We implemented Attention-based embedding
- We decreased the size of the predictor dimension
- We normalized the dataset

Plus: EMA updating the Target Encoder

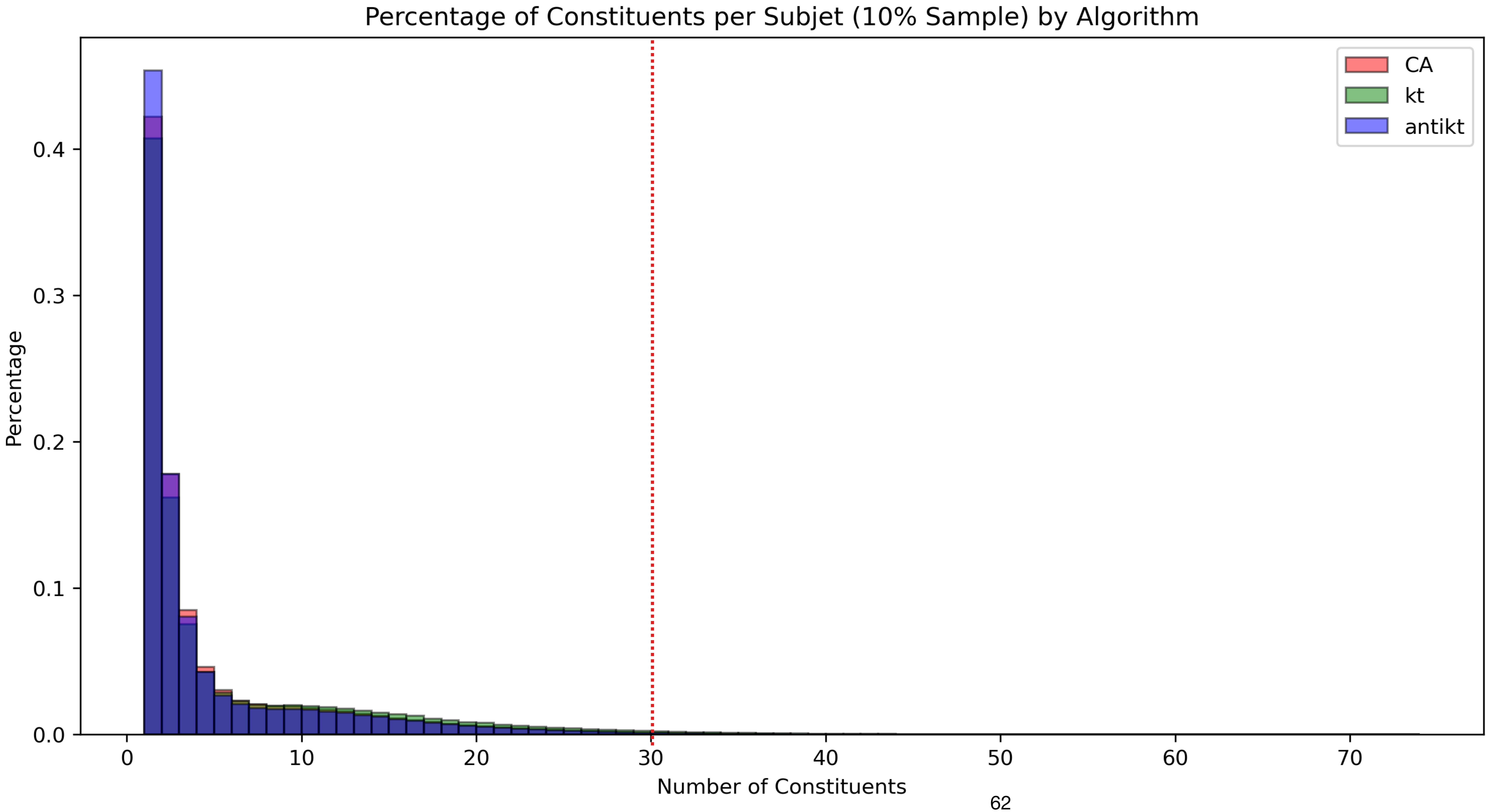
J-JEPA: Splitting jets into subjets

number of subjets per jet



J-JEPA: Splitting jets into subjets

number of particles per subjet

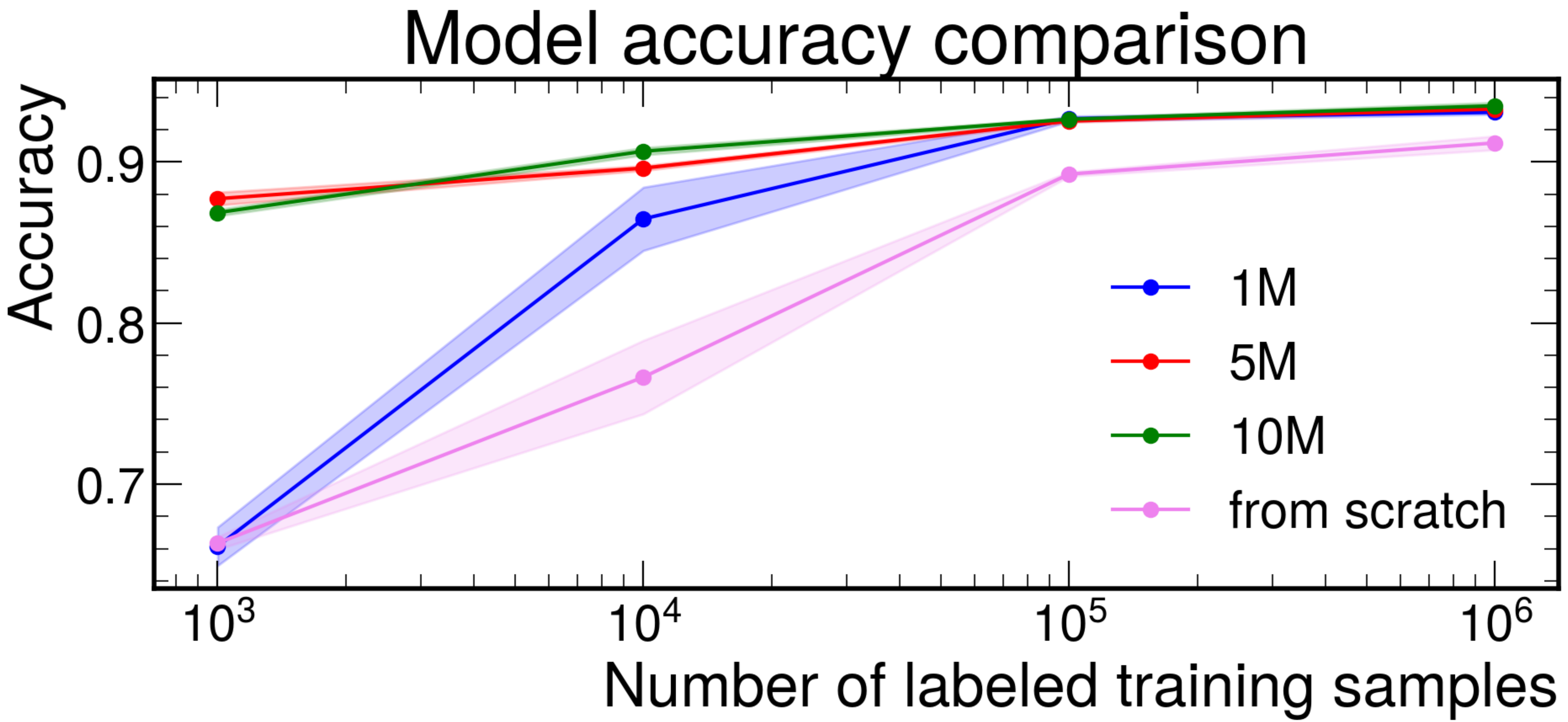


Back Up (Part 2)

Techniques to speed up training

Steps we took to ensure the model finished pretraining within a reasonable amount of time

- Removed unnecessary CPU-GPU synchronizations, especially read-out from GPU for recording losses
- Modified the default model dimensions to be multiples of 8 to make use of CUDA matrix multiplication kernels more efficiently
- Fused point-wise operations into a single CUDA kernel when computing the contrastive loss.
- Utilized the Automatic Mixed Precision (AMP) package
    - Measures to mitigate the numerical instability caused by using AMP in backup.



Model accuracy comparison
Accuracy
0.9
0.8
0.7
10^3
10^4
10^5
10^6
Number of labeled training samples
1M
5M
10M
from scratch

LHC and Jet Tagging

Collision event

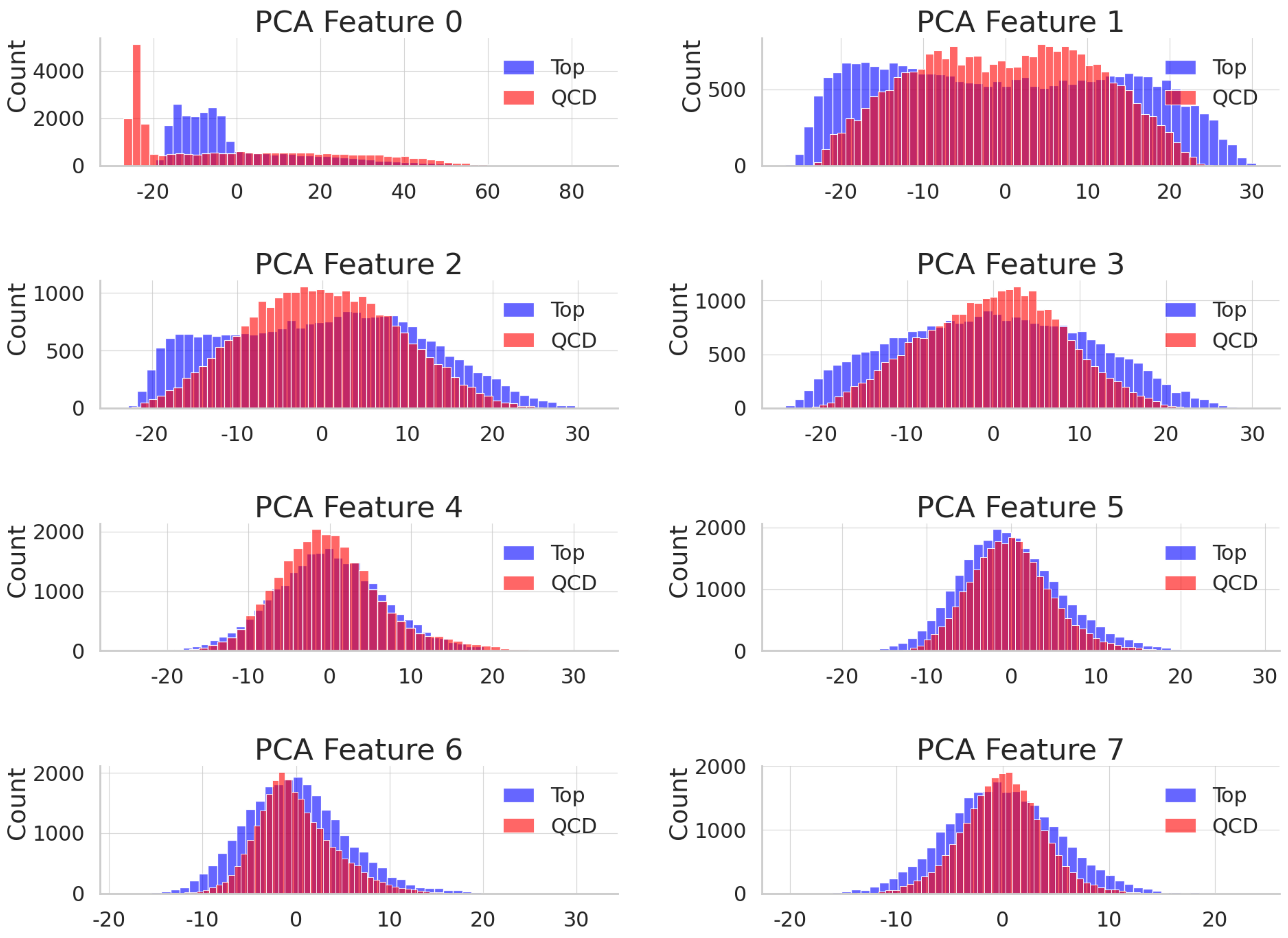
1709.04464

Measures to mitigate the numerical instability caused by using AMP

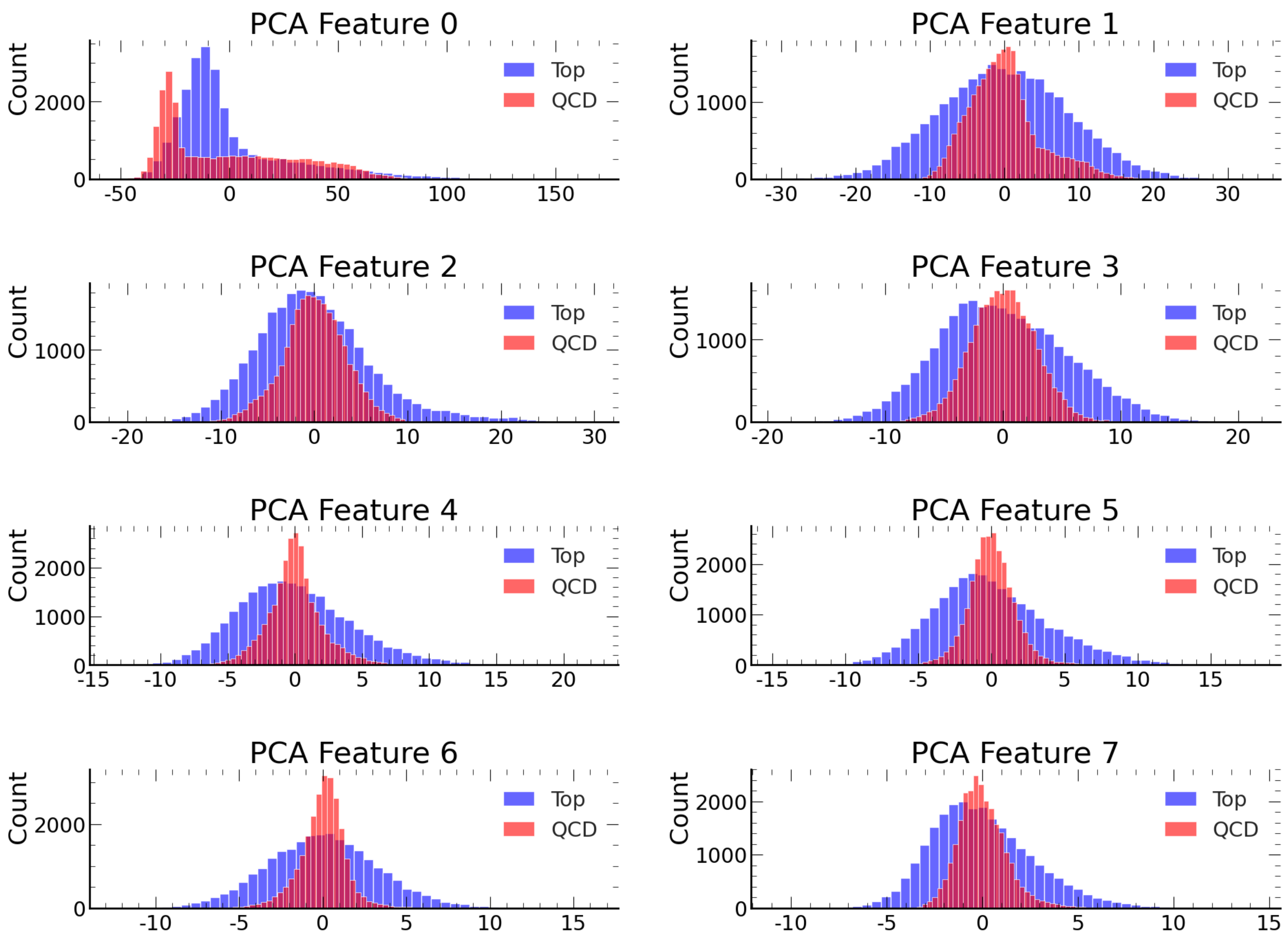
- Monitor loss and gradient values regularly with tensorboard
- Gradient clipping with a maximum norm of 0.1
- Set the ϵ parameter to 10^(−4) in the Adam optimizer.
- Manually run certain parts of the code in full precision

Pretraining on JetClass and fine-tuning on Top Tagging

The pre-trained model shows a much clearer separation between signal and background



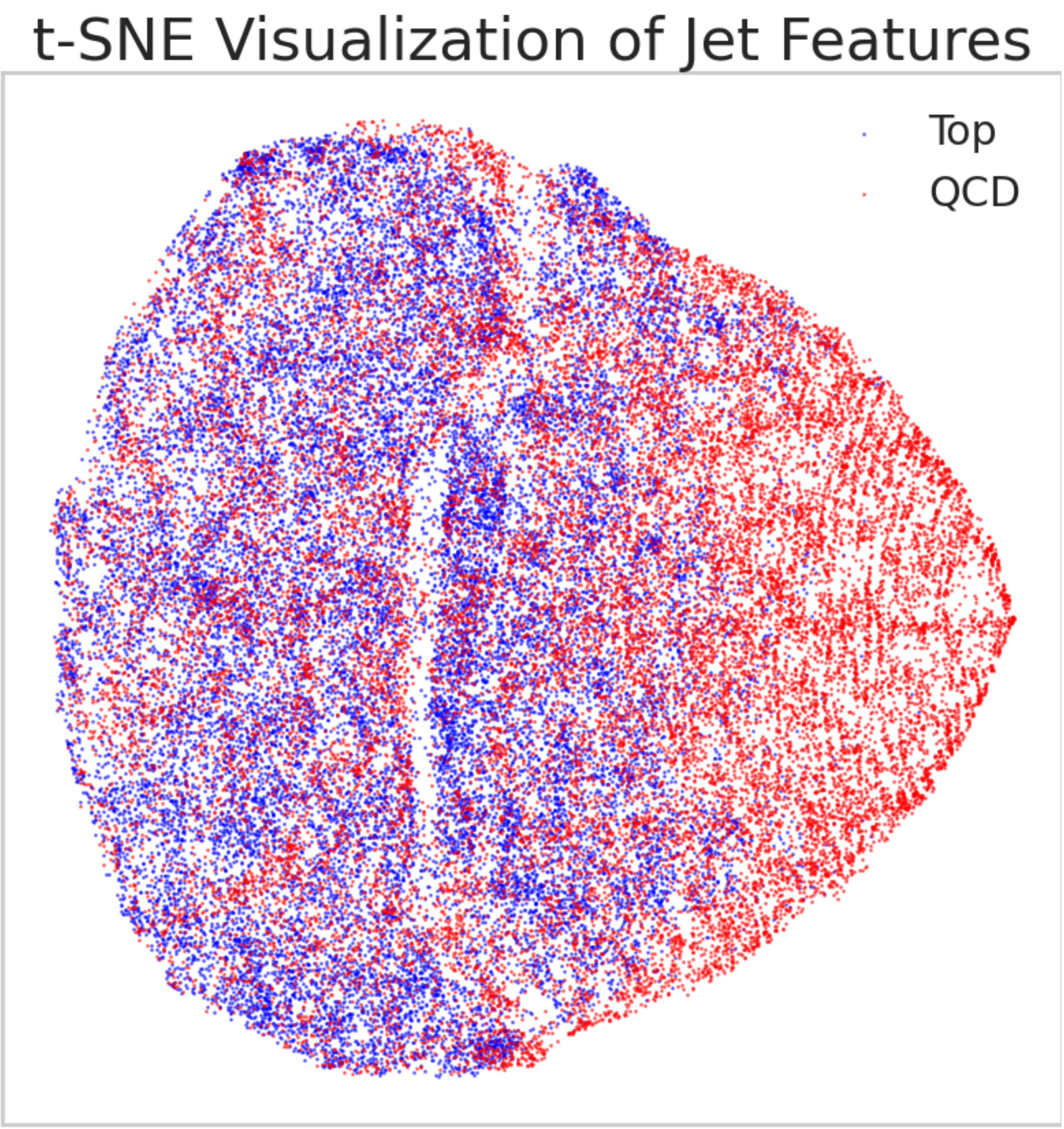
Trained from scratch



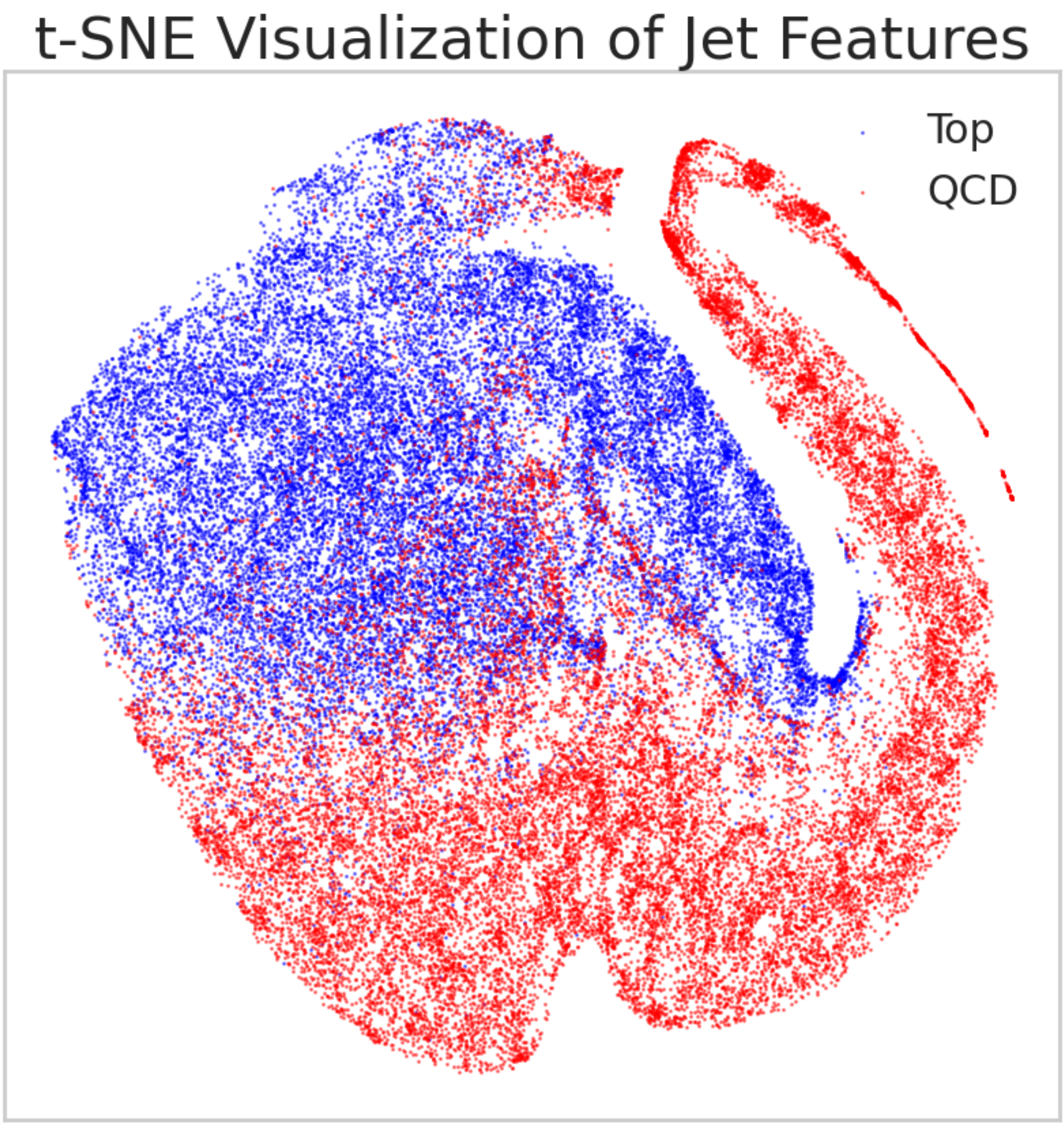
Pre-trained

Pretraining on JetClass and fine-tuning on Top Tagging

The pre-trained model shows a much clearer separation between signal and background



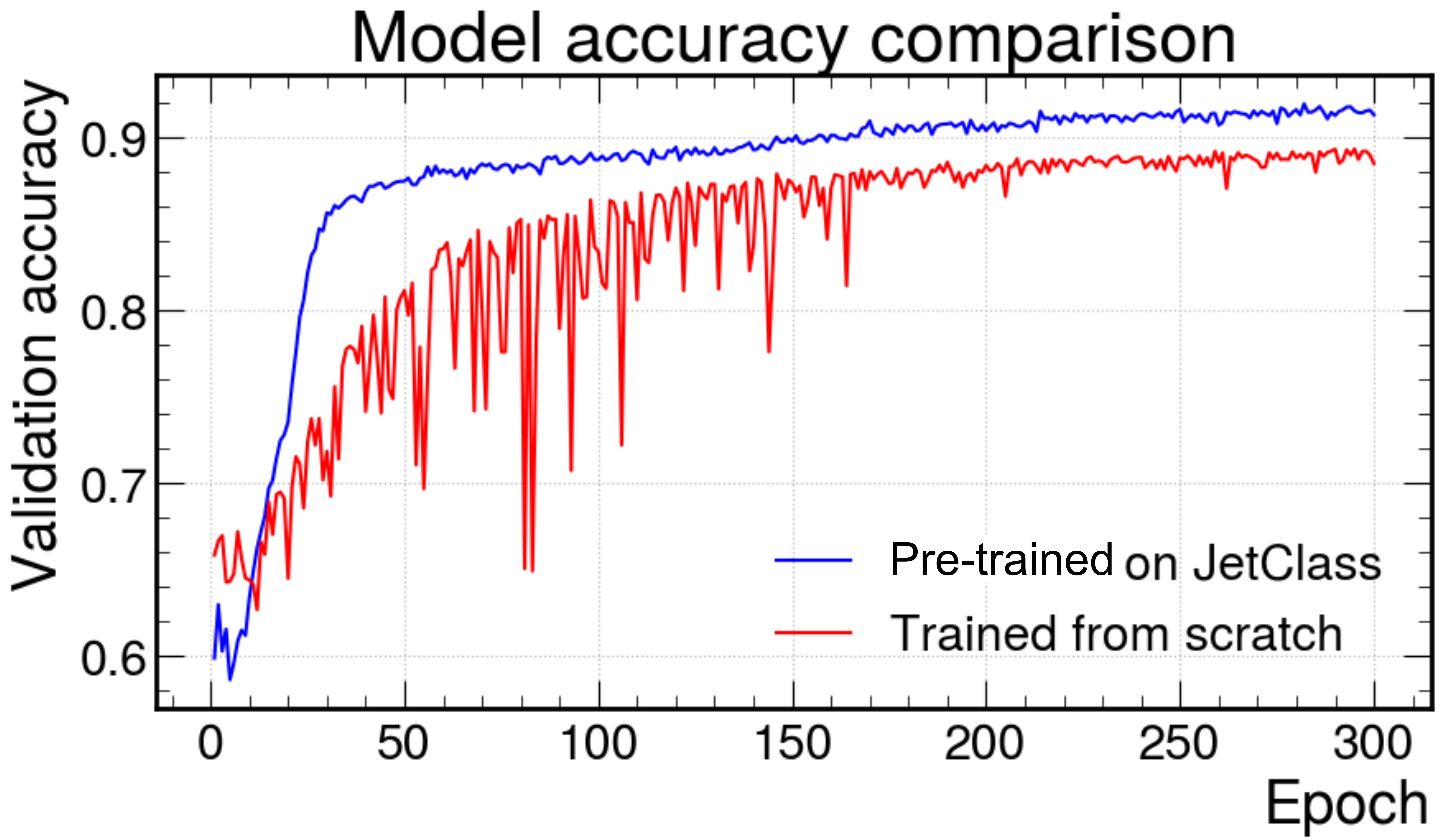
Trained from scratch



Pre-trained

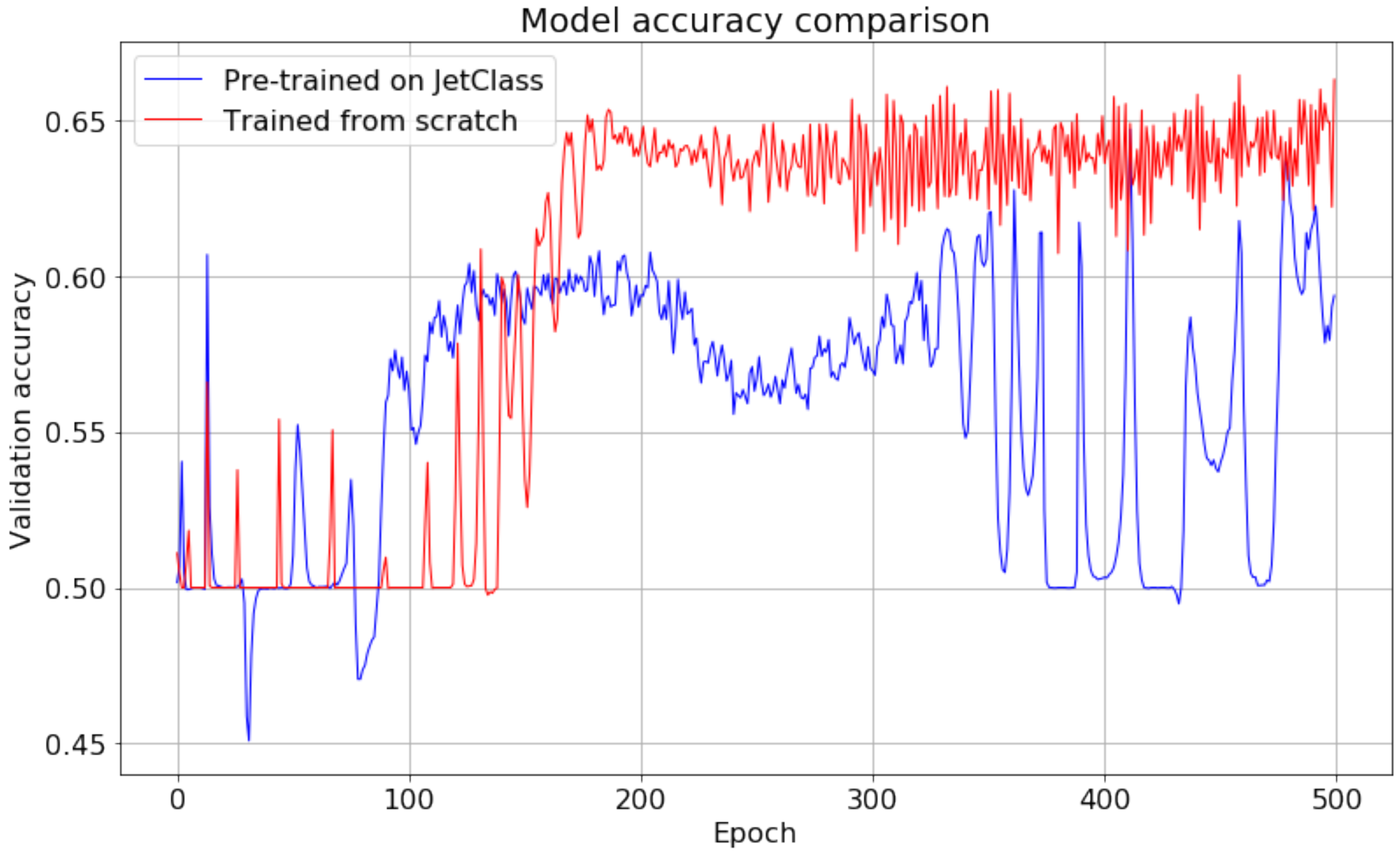
Pretraining on JetClass and fine-tuning on Top Tagging

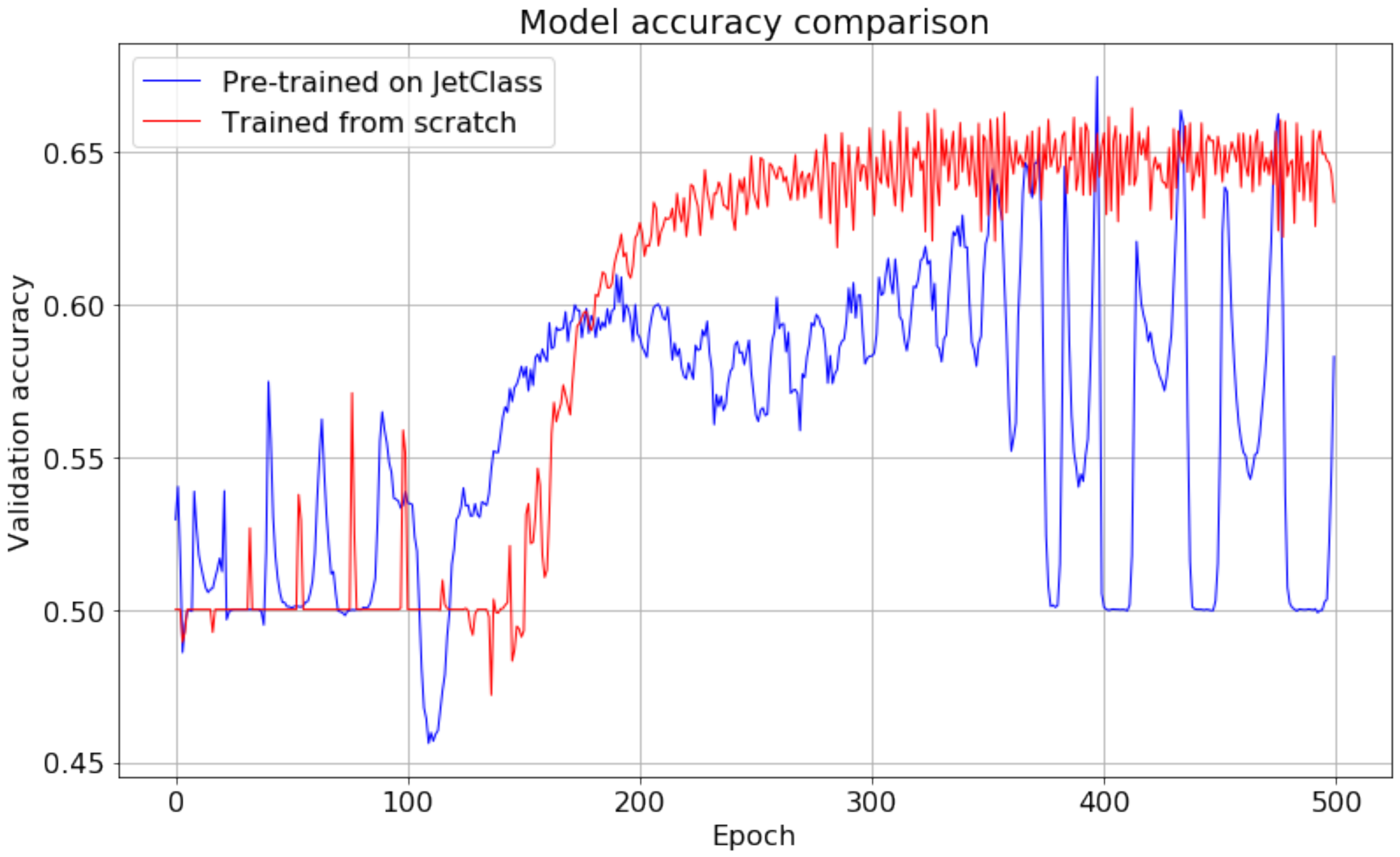
Despite limited data, the pre-trained model achieves higher accuracy and converges faster



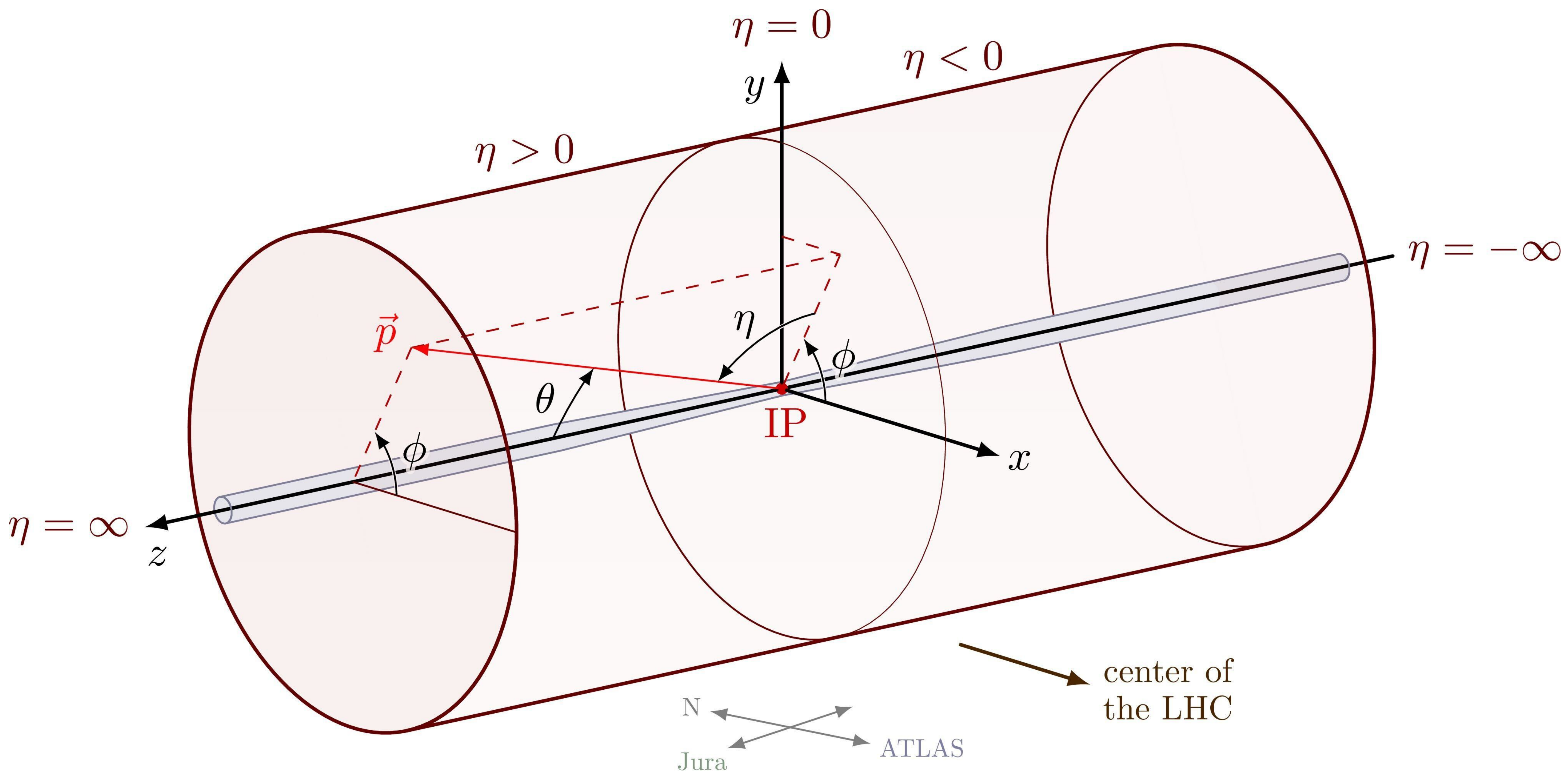
- A linear layer was added to the encoder for fine-tuning.
- Blue curve was pre-trained on 1% of the JetClass dataset (1 Million jets) with SimCLR
- Red curve was trained from scratch
- Both models share the same hyperparameters
- Both models are trained with 100k jets (1/12 of the Top Tagging Dataset)

Accuracies of two trials trained with 1000 labeled samples





The CMS detector coordinate system



$$\eta \equiv -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$

https://tikz.net/axis3d_cms/

Details of the Top Tagging Dataset

The top signal and mixed quark-gluon background jets are produced with using Pythia8 [25] with its default tune for a center-of-mass energy of 14 TeV and ignoring multiple interactions and pile-up. For a simplified detector simulation we use Delphes [26] with the default ATLAS detector card. This accounts for the curved trajectory of the charged particles, assuming a magnetic field of 2 T and a radius of 1.15 m as well as how the tracking efficiency and momentum smearing changes with η. The fat jet is then defined through the anti-k_T algorithm [27] in FastJet [28] with $R = 0.8$. We only consider the leading jet in each event and require

$$p_{T,j} = 550 \;....\; 650 \text{ GeV} . \tag{1}$$

For the signal only, we further require a matched parton-level top to be within $\Delta R = 0.8$, and all top decay partons to be within $\Delta R = 0.8$ of the jet axis as well. No matching is performed for the QCD jets. We also require the jet to have $|\eta_j| < 2$. The constituents are extracted through the Delphes energy-flow algorithm, and the 4-momenta of the leading 200 constituents are stored. For jets with less than 200 constituents we simply add zero-vectors.

1902.09914

Details of the JetClass Dataset

Simulation setup. Jets in this dataset are simulated with standard Monte Carlo event generators used by LHC experiments. The production and decay of the top quarks and the W, Z and Higgs bosons are generated with MADGRAPH5_aMC@NLO (Alwall et al., 2014). We use PYTHIA (Sjöstrand et al., 2015) to evolve the produced particles, i.e., performing parton showering and hadronization, and produce the final outgoing particles[1]. To be close to realistic jets reconstructed at the ATLAS or CMS experiment, detector effects are simulated with DELPHES (de Favereau et al., 2014) using the CMS detector configuration provided in DELPHES. In addition, the impact parameters of electrically charged particles are smeared to match the resolution of the CMS tracking detector (CMS Collaboration, 2014). Jets are clustered from DELPHES E-Flow objects with the anti-k_{T} algorithm (Cacciari et al., 2008; 2012) using a distance parameter $R = 0.8$. Only jets with transverse momentum in 500–1000 GeV and pseudorapidity $|\eta| < 2$ are considered. For signal jets, only the "high-quality" ones that fully contain the decay products of initial particles are included[2].

2202.03772

Training on Top Tagging

Are the features correlated?