Self-supervised learning: machine can learn a lot by just observing

– an application to LHC jet data

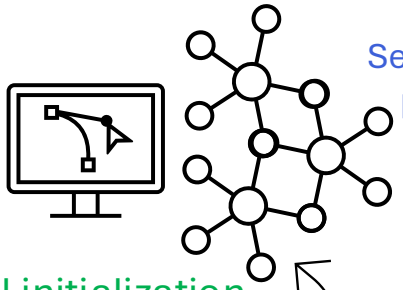
Ho Fung Tsoi, Dylan Rankin [UPenn]

TREASURE workshop at BNL

29 Apr 2026

Base on 2601.11719

Self-supervised learning paradigm



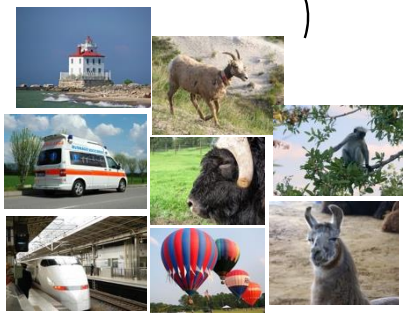
Model initialization

Self-supervised pre-training

General knowledge

Supervised fine-tuning

Object detection
Instance segmentation
Classification
...



Machine observes large amounts of unlabeled data to learn underlying context

Fine-tune on labeled data for specific tasks such as classification

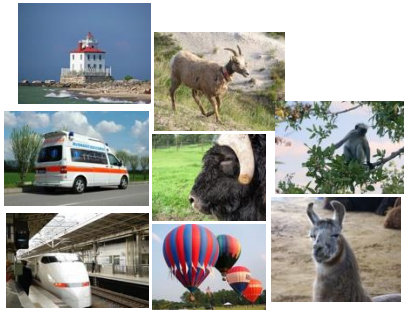
Self-supervised pre-training (upstream)

- Train a 'backbone' model on unlabeled data
- Let the model observes structure in data
- Learn the underlying structure and patterns
- Map raw inputs into a meaningful representation space

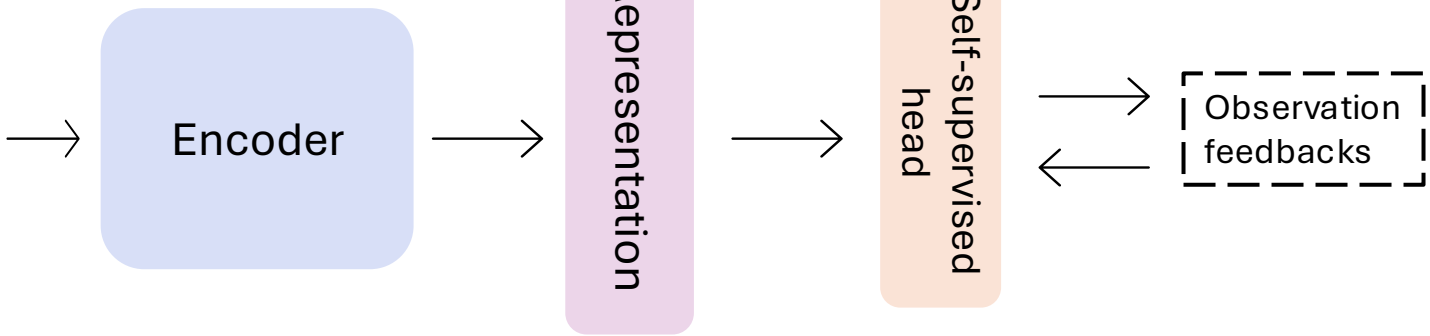
Supervised fine-tuning (downstream)

- Use the pre-trained model as the starting point
- Fine-tune on labeled data for specific downstream tasks

Upstream: pre-training



Unlabeled data
(often abundant)



Self-supervised learning (build ‘common sense’ upstream)

Downstream: fine-tuning

Unlabeled data
(often abundant)

Encoder

Representation

Self-supervised head

Observation feedbacks

Self-supervised learning (build 'common sense' upstream)

Labeled data
(often scarce)

Pre-trained encoder ('common sense')

Representation

Supervised head

Dog? Cat?

Supervised fine-tuning (acquire specific skills downstream)

Observation by self-prediction (e.g. language)

As a self-predictive task
- Mask part of the input
- Predict the missing part from the visible part

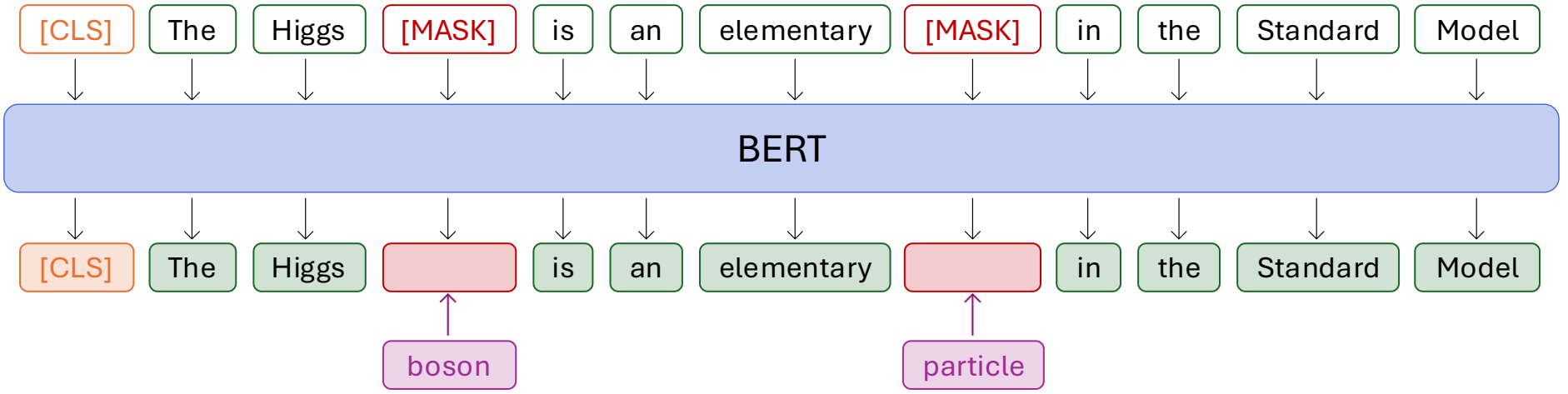
An example from LLMs
[mask language modeling]

BERT: Pre-training of deep bidirectional transformers for language understanding [1810.04805]

"The Higgs boson is an elementary particle in the Standard Model."

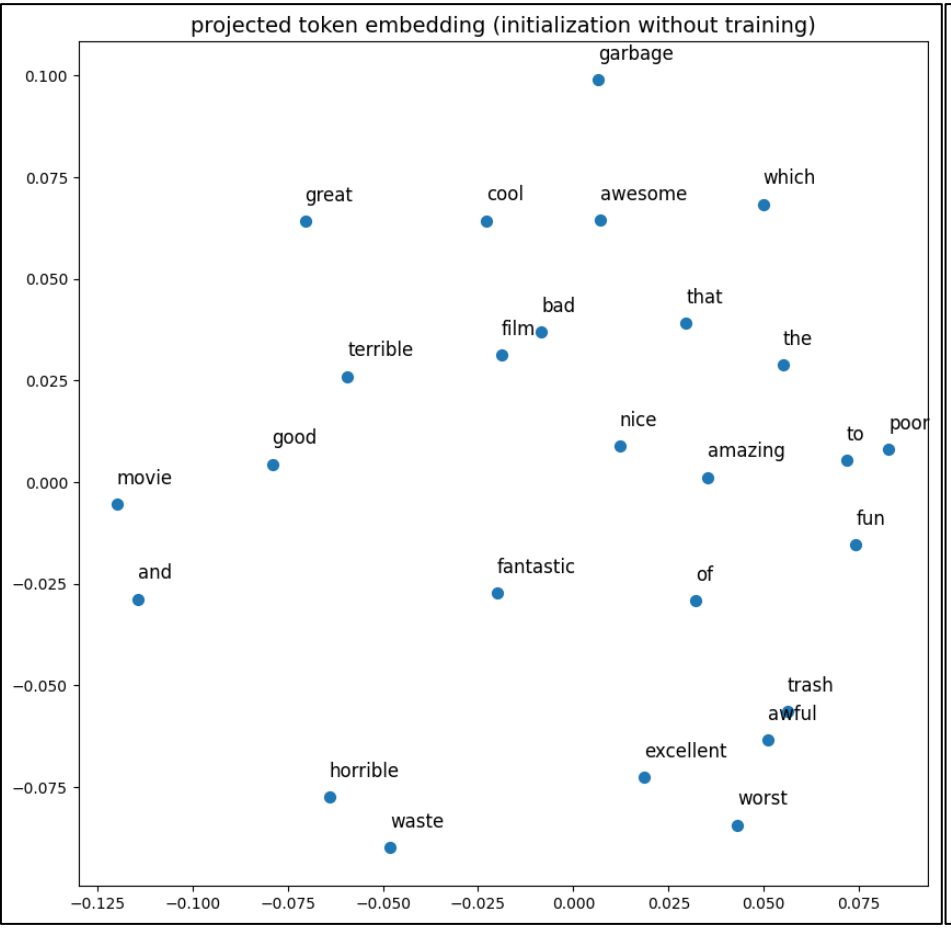
Mask words in a sentence, then train to fill in the blanks.

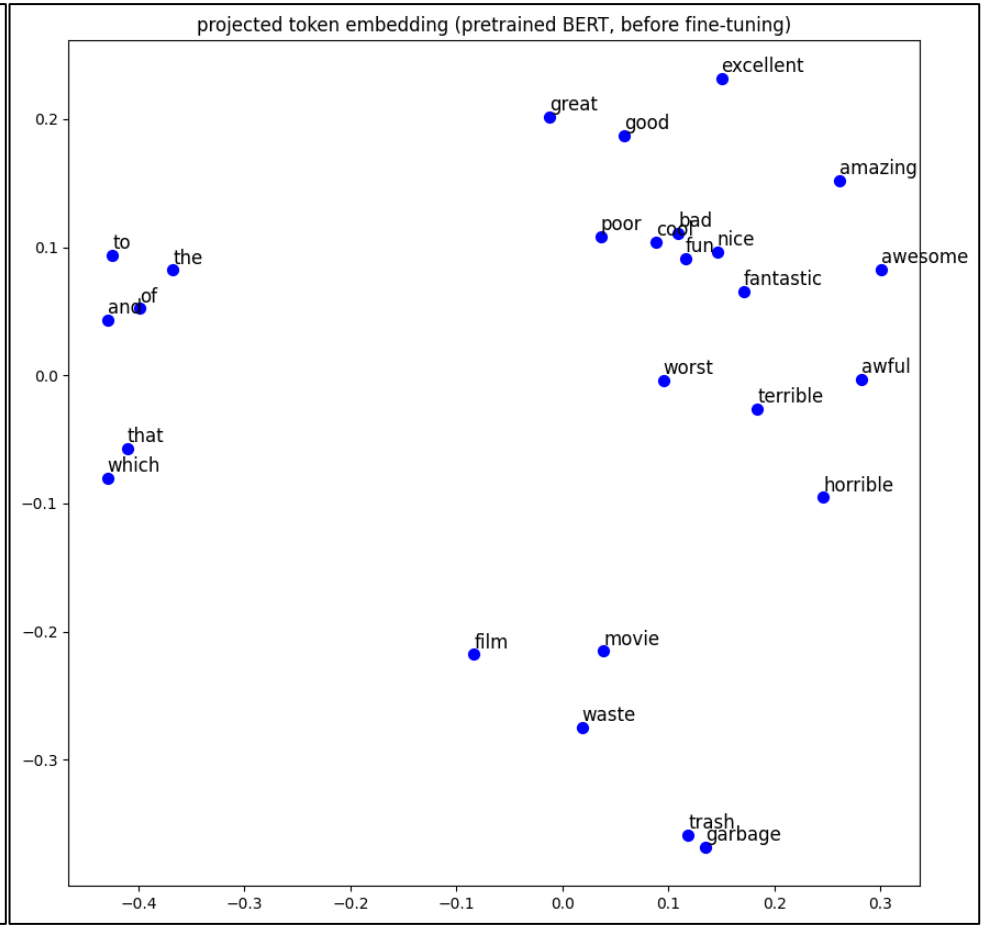
"The Higgs _____ is an elementary _____ in the Standard Model."



Self-generated labels from data itself

- Emergent word semantics visualized in projected 2D embedding space





Random initialization

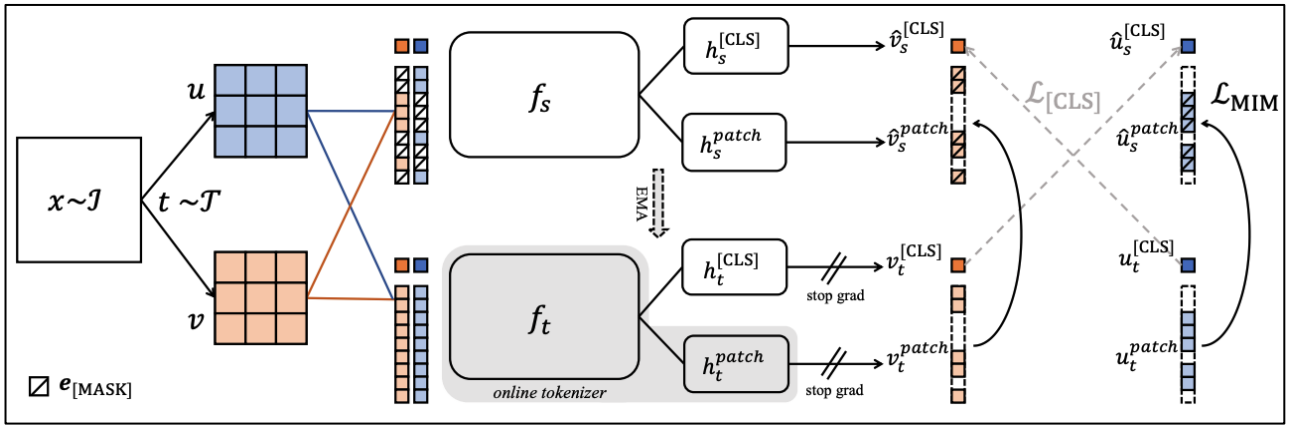
After self-supervised pre-training (fill-in-the-blanks task)

Observation by self-prediction (e.g. image)

As a self-predictive task
- Mask part of the input
- Predict the missing part from the visible part

An example from computer vision [mask image modeling]

iBOT: Image BERT pre-training with online tokenizer [2111.07832]



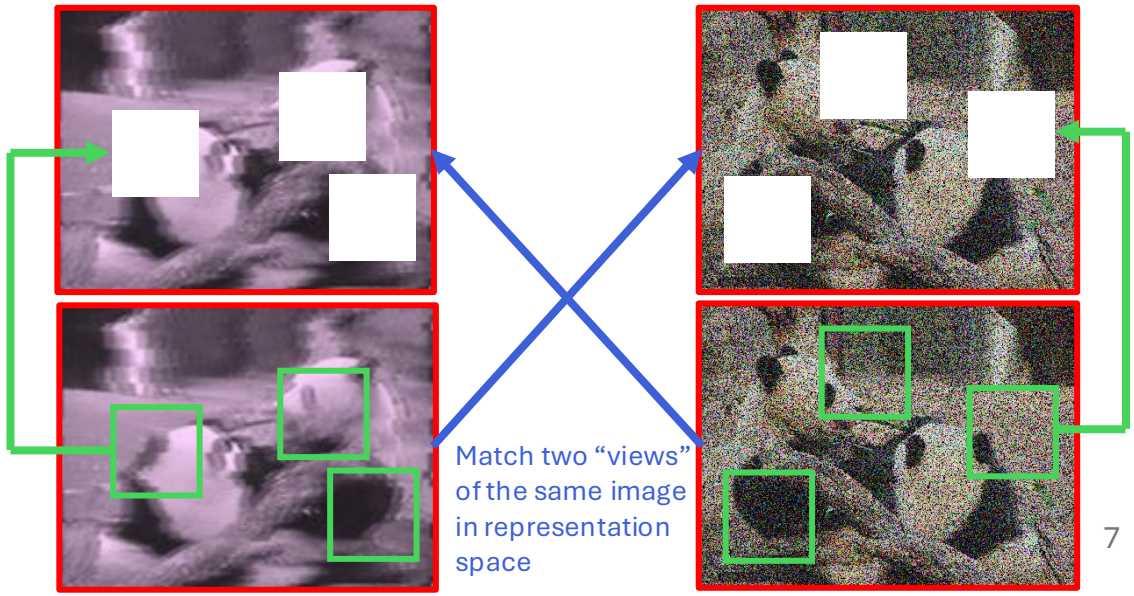
For each image, generate two "views" (glitch, blur, flip,...)

Self-distillation
- Teacher sees full image
- Student sees masked image
- Teacher provides target to student

Self-supervised objectives
- Predict masked image patches
- Learn invariance between the two 'views' (they are the same image)

Unlabeled data

Generate two distorted views through augmentation



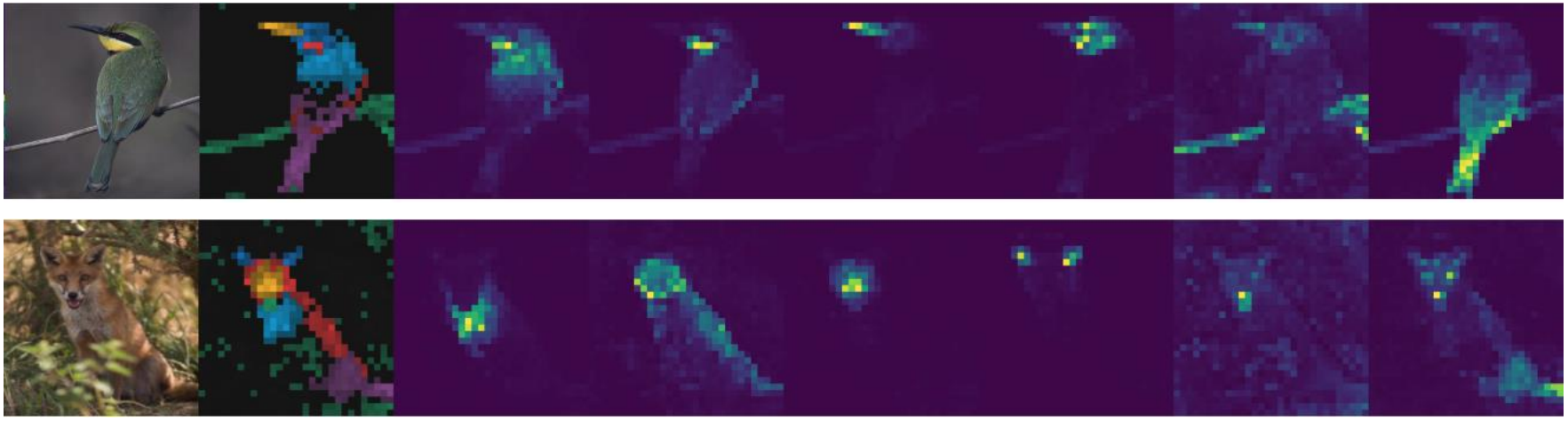
Match two "views" of the same image in representation space

Fill in the blanks

Figure 6: **Visualization for self-attention map.** Self-attention map from multiple heads are visualized with different color.

- The learned features often appear semantic, even no labels are used
- Different attention heads focus on different structures → semantic segmentation
- Pre-training significantly improves downstream performance compared to supervised model trained from scratch

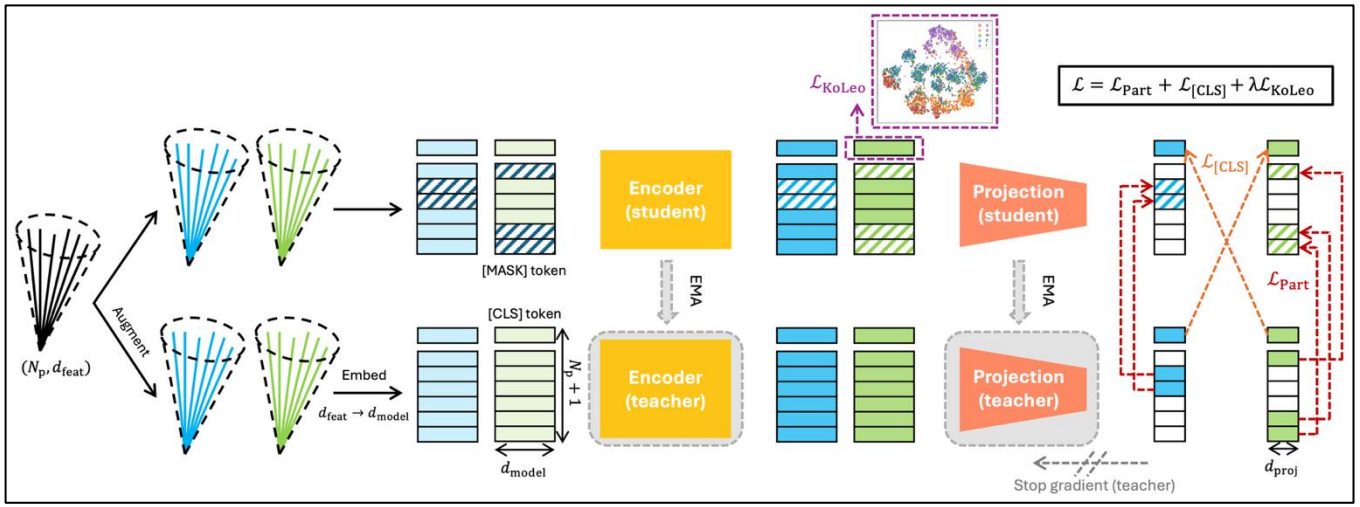
iBOT: Image BERT pre-training with online tokenizer [2111.07832]



Input image | Attention from multiple attention heads | Per attention head

jBOT (ours)

2601.11719



jBOT: adapting iBOT from computer vision to LHC jet data

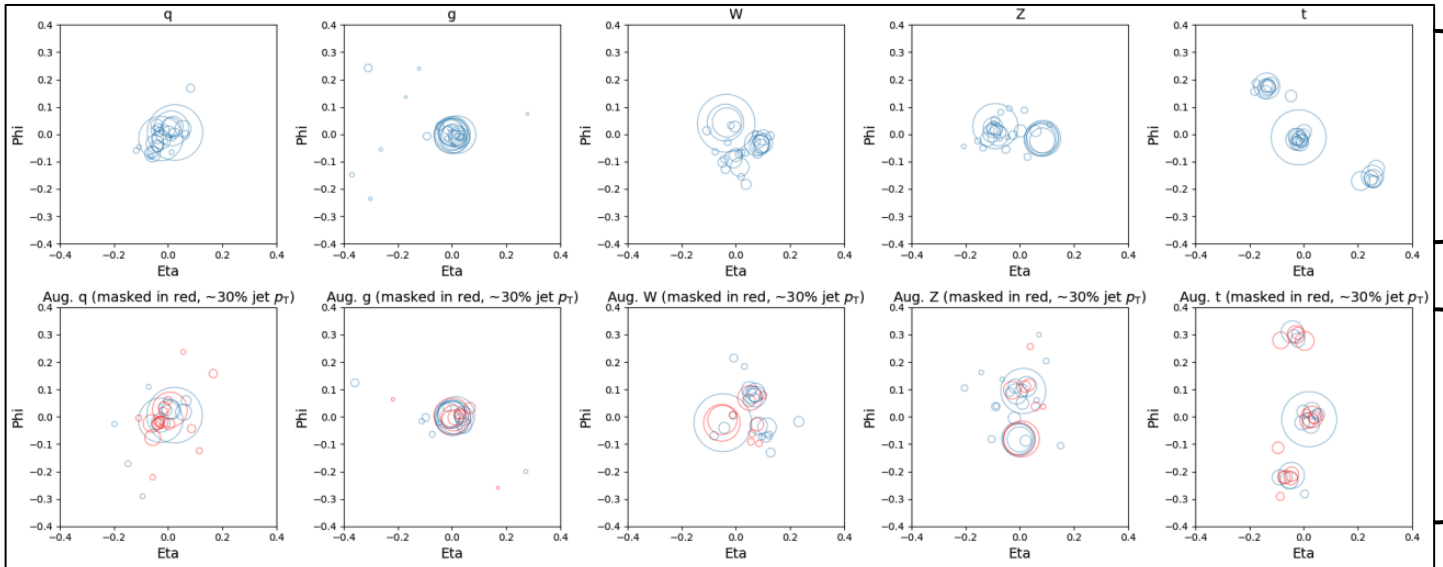


Figure 3: One example jet per class, where each circle represents a particle: the circle center is at the particle location and the radius is proportional to its p_T. Upper row: input jets. Lower row: augmented jets with masking shown in red (e.g., ~30% of the jet p_T).

Input jets
Simulated q/g/W/Z/t jets of ~1 TeV at the LHC [JetNet]

Augmented jets
- Rotation
- Position smearing
- Collinear splitting

Mask particles that add up to x% jet pT, where x is uniformly sampled from 0 to 50

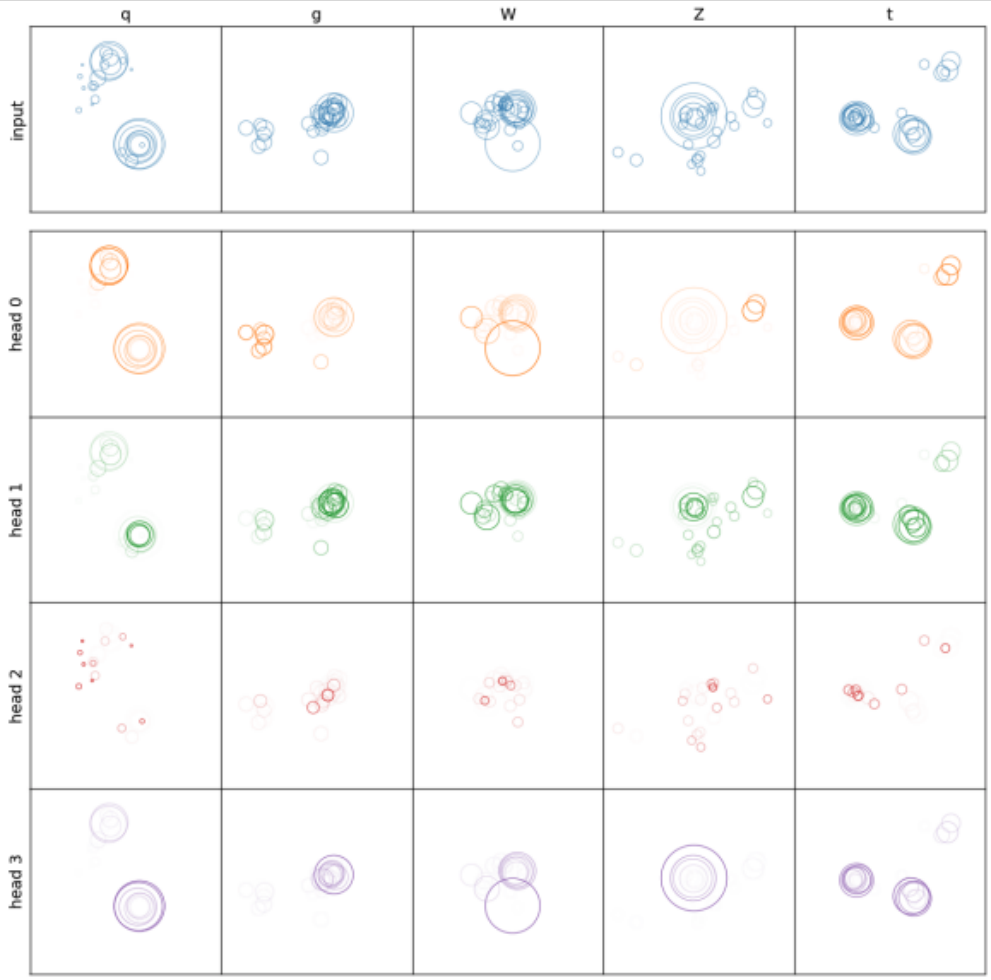


Figure 5: Attention weights from the last transformer block in jBOT-S pre-trained on all five classes, obtained using the [CLS] token as the query attending to the particle tokens. Top row: one example input jet per class (each circle represents a particle: the circle center is at the particle location, the radius is proportional to its p_T, and the edge alpha is uniform across all particles here). Other rows: attention weights per head for the same input jets, shown with the same drawing style as the input jets, but with the attention weight represented by the edge alpha (higher edge alpha indicates larger attention weight).

- The transformer encoder attends to different groups of particles across attention heads
- Learning high-level info and correlations within jet substructure

2601.11719

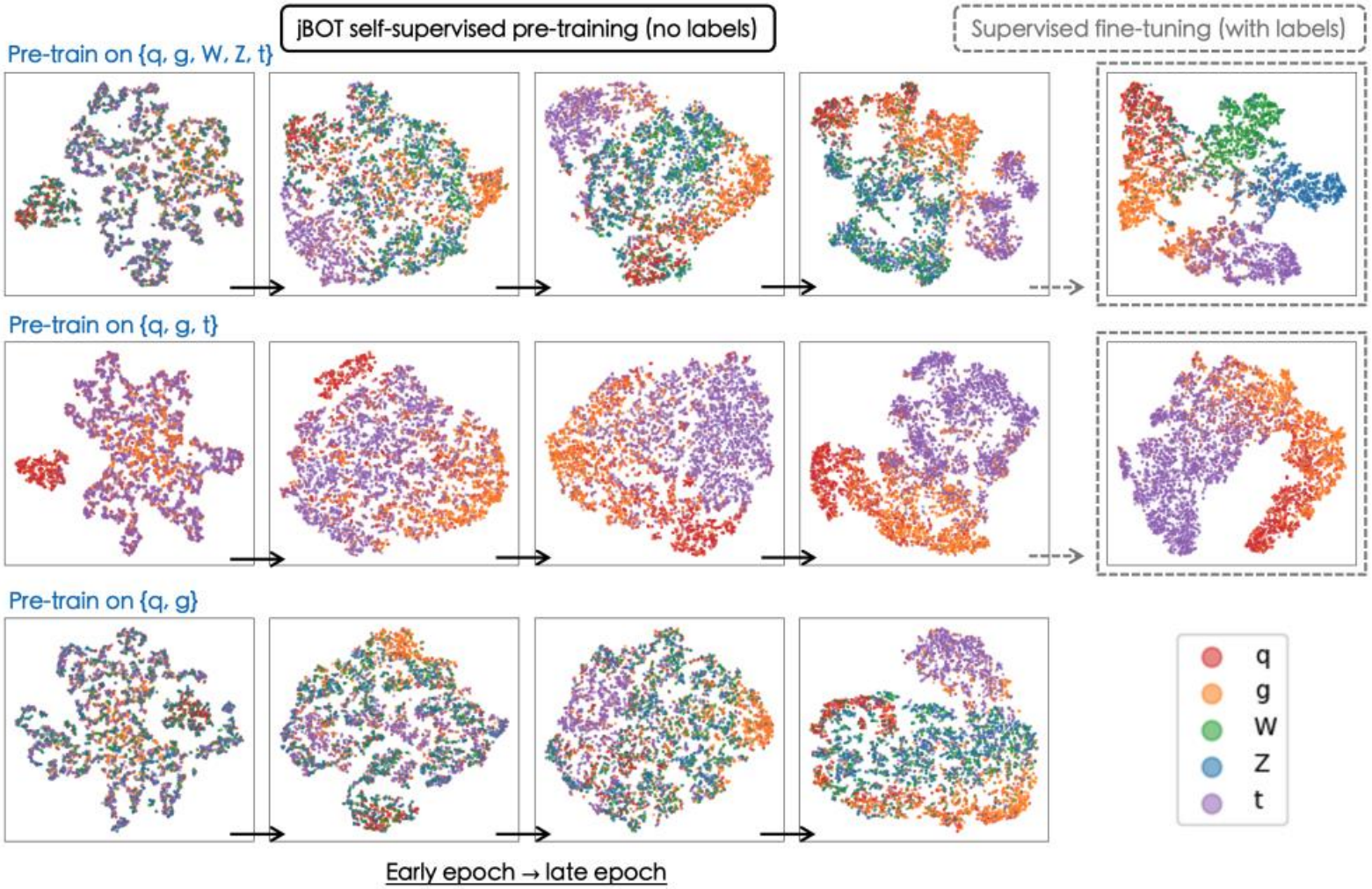
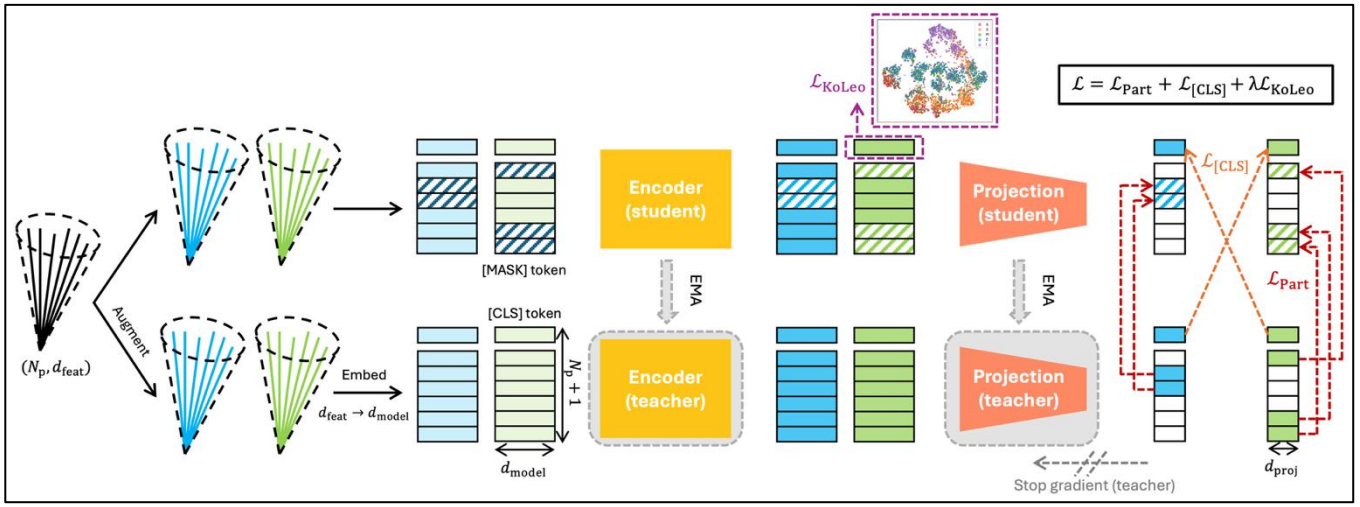


Figure 6: Example evolution of 2D t-SNE projections of the [CLS] token during pre-training (and after fine-tuning). Top row: pre-training on all five classes and then fine-tuning. Middle row: pre-training on the t, q, and g classes and then fine-tuning. Bottom row: pre-training on the q and g classes.

2601.11719



Self-supervised pre-training (upstream)

2601.11719

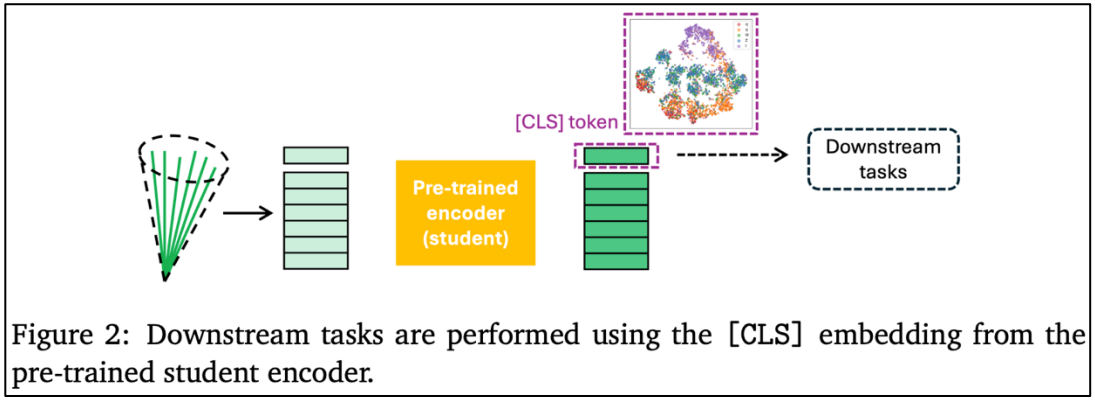
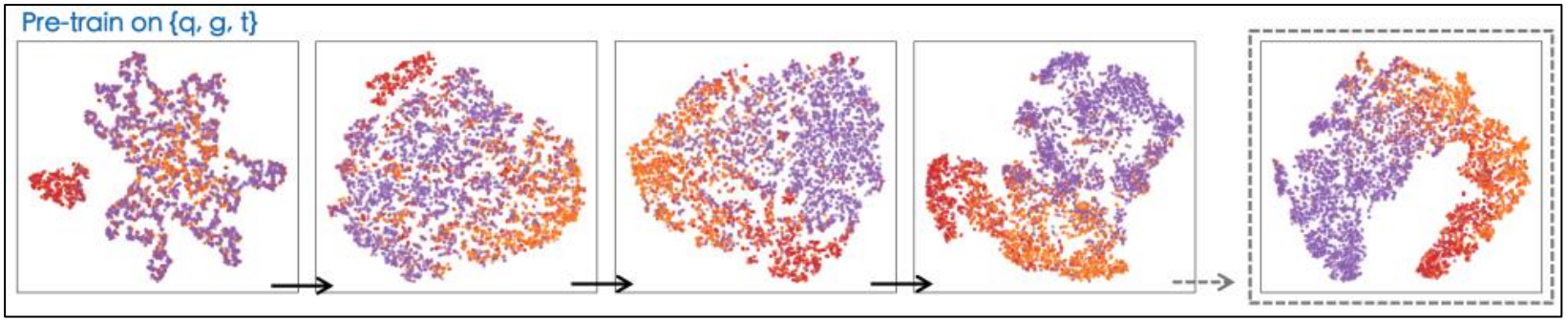


Figure 2: Downstream tasks are performed using the [CLS] embedding from the pre-trained student encoder.

Supervised fine-tuning (downstream)

- After self-supervised pre-training, one can probe the jet representation via the [CLS] token
- The learned representations provide a good starting point for downstream tasks (tagging, anomaly detection,...), instead of training directly from raw input features

E.g., top vs QCD jets



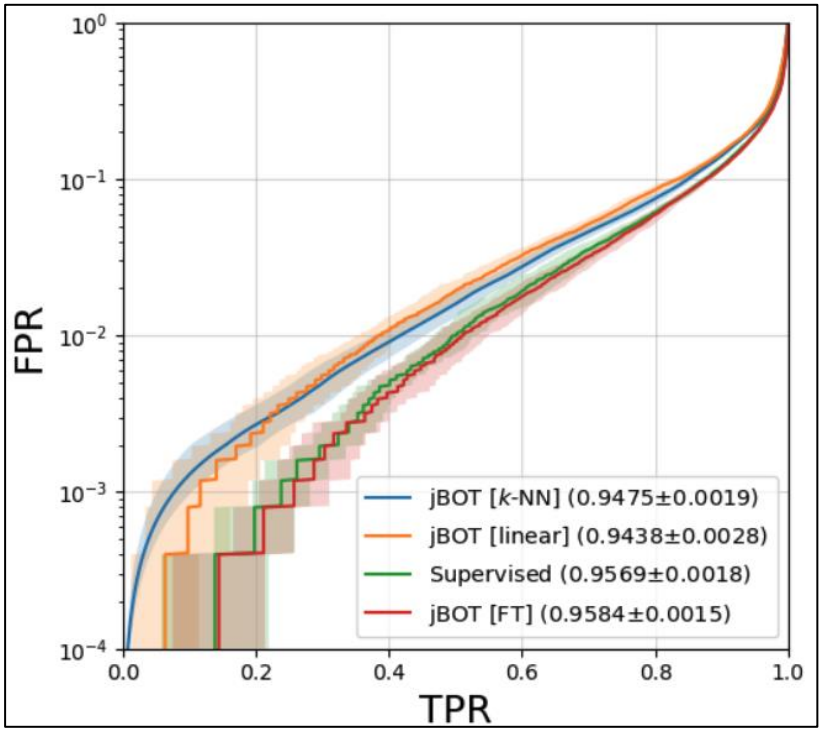


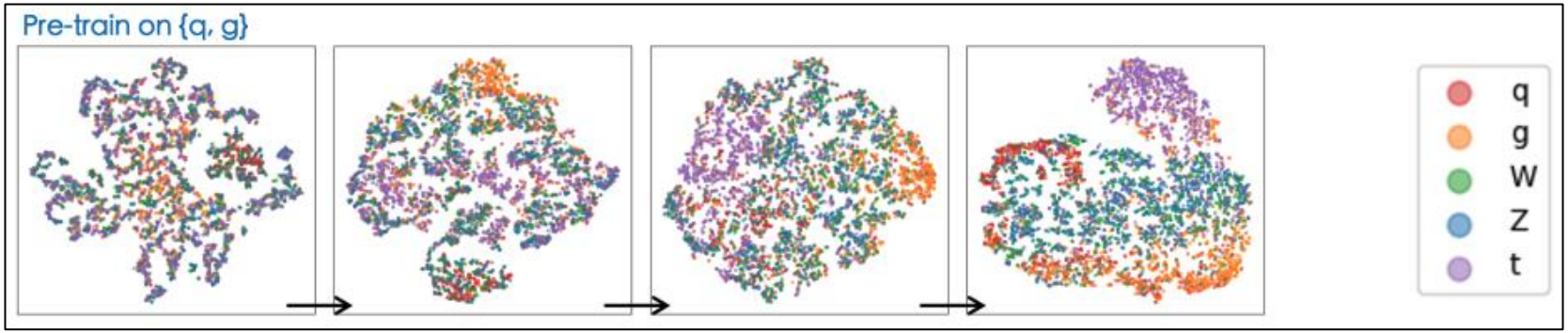
Table 3: Top tagging performance (accuracy, AUC, and signal efficiency ϵ_s at different background efficiencies) comparing jBOT with supervised models. Note that all models use particle features only and ignore jet-level features.

Model	Acc. [%]	AUC	$\epsilon_s(10^{-1})$	$\epsilon_s(10^{-2})$
Frozen embedding (no labels)				
k-NN (jBOT-S)	0.8777 ± 0.0037	0.9447 ± 0.0023	0.8352 ± 0.0125	0.3337 ± 0.0572
Linear (jBOT-S)	0.8709 ± 0.0039	0.9355 ± 0.0020	0.8115 ± 0.0138	0.2408 ± 0.0327
k-NN (jBOT-B)	0.8793 ± 0.0035	0.9475 ± 0.0019	0.8402 ± 0.0066	0.3494 ± 0.0699
Linear (jBOT-B)	0.8765 ± 0.0041	0.9438 ± 0.0028	0.8344 ± 0.0116	0.3892 ± 0.0315
Fine-tuning (with labels)				
Sup.-S (10%)	0.8723 ± 0.0040	0.9368 ± 0.0028	0.8172 ± 0.0158	0.2166 ± 0.0394
jBOT-S (10%)	0.8807 ± 0.0047	0.9499 ± 0.0019	0.8559 ± 0.0106	0.4306 ± 0.0317
Sup.-S (100%)	0.8852 ± 0.0047	0.9524 ± 0.0016	0.8659 ± 0.0080	0.4474 ± 0.0366
jBOT-S (100%)	0.8875 ± 0.0040	0.9554 ± 0.0015	0.8712 ± 0.0097	0.4814 ± 0.0328
Sup.-B (10%)	0.8784 ± 0.0051	0.9467 ± 0.0023	0.8429 ± 0.0132	0.4131 ± 0.0285
jBOT-B (10%)	0.8862 ± 0.0038	0.9542 ± 0.0017	0.8665 ± 0.0110	0.4843 ± 0.0306
Sup.-B (100%)	0.8899 ± 0.0035	0.9569 ± 0.0018	0.8756 ± 0.0072	0.5021 ± 0.0331
jBOT-B (100%)	**0.8911 ± 0.0029**	**0.9584 ± 0.0015**	**0.8771 ± 0.0079**	**0.5122 ± 0.0389**

2601.11719

- Model pre-trained without supervision can already perform competitively with a fully supervised model trained from scratch
- After supervised fine-tuning, it often outperforms, especially when labeled data is scarce

E.g., anomalous signal (W, Z, top) vs QCD jets



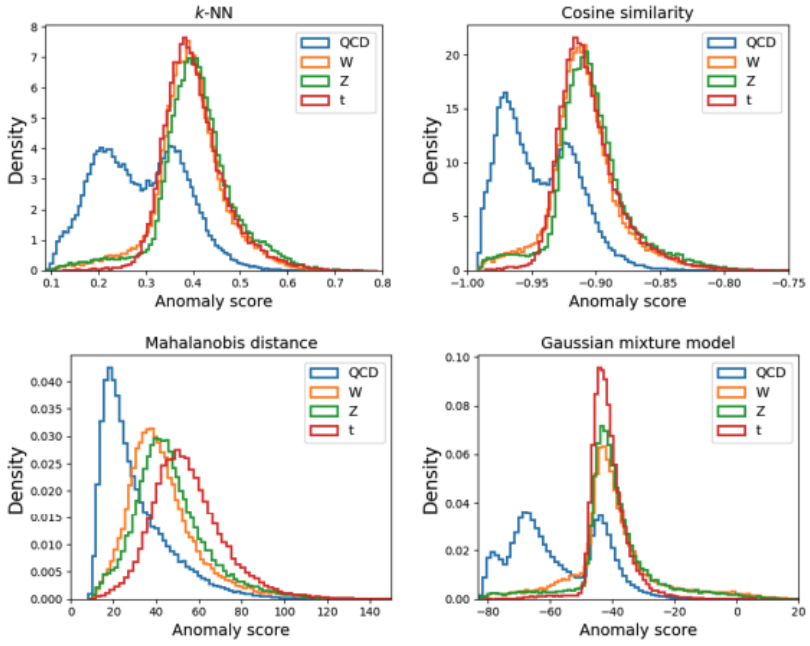


Figure 8: Anomaly score distributions: k-NN distance (upper left), cosine similarity (upper right), Mahalanobis distance (lower left), and GMM (lower right).

Table 4: Anomaly detection performance comparing jBOT (particle features only) and reconstruction-based autoencoder models from Ref. [43].

Model	AUC			
	W	Z	t	Combined
CNNAE [43]	0.6886	0.7247	**0.8962**	0.7700
GNNAE [43]	0.7558	0.7805	**0.8917**	0.8195
LGAE [43]	0.7489	0.7909	0.8669	**0.8313**
jBOT-S (k-NN)	**0.8072 ± 0.0028**	**0.8355 ± 0.0025**	0.8356 ± 0.0029	0.8261 ± 0.0028
jBOT-S (Cosine)	**0.8064 ± 0.0028**	**0.8355 ± 0.0027**	0.8388 ± 0.0028	**0.8269 ± 0.0027**
jBOT-S (Maha.)	0.7431 ± 0.0030	0.7821 ± 0.0029	0.8620 ± 0.0026	0.7957 ± 0.0037
jBOT-S (GMM)	0.8062 ± 0.0040	0.8204 ± 0.0040	0.8197 ± 0.0049	0.8155 ± 0.0038

2601.11719

- When trained on QCD jets only, anomaly detection can be performed by probing the learned representation space using simple distance/density-based metrics

Summary

- jBOT: a self-distillation-based pre-training method for jet data
- Results look encouraging and suggest further exploration and broader applications
- Provide a viable architectural option for foundation model training
- We are also actively working on various advances in SSL: improving augmentations, pre-training scheme, events vs jets etc.