RINO: RENORMALIZATION GROUP DISTILLATION WITH NO LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised machine learning in high-energy physics (HEP) requires training on labeled simulations of particle collisions, which are known to be unfaithful to the underlying physics. This leads to significant inaccuracies and biases in the transfer of trained models to real data, limiting the power of deep learning methods. Self-supervised learning (SSL) is a promising alternative for pretraining directly on real, unlabeled data. However, existing techniques largely operate with inductive biases inspired by domains such as images and natural language. We propose RINO (**R**enormalization Group D**I**stillation with **NO** Labels), which, for the first time, exploits the fundamentally multi-scale renormalization group structure of high-energy *jet* formation as a physically grounded source of SSL views. We define teacher-student representations across energy scales using different levels of hierarchical clustering, aligned via DINO-style self-distillation. To mirror the background-dominated nature of real collider data, RINO pretrains exclusively on jets from quantum chromodynamics (QCD) processes and is evaluated on out-of-distribution (OOD) transfer to rare signal classes. In our controlled data-efficiency study, RINO matches fully supervised training on JETNET after finetuning on only 1% of the labeled data. With the full labeled dataset, RINO improves OOD classification accuracy on top-quark jets by 5.9% and, without retraining, improves AUC on the unseen Higgs boson decay channel $H \to 4q$ by 4.1% over the strongest SSL baseline. Scale-structure diagnostics reveal that DINO-style self-distillation permits our backbone to retain scale-dependent structure. Ablation studies show that this accounts for an 8.6% OOD accuracy gap with respect to alternative SSL paradigms such as SimCLR. Code is available at `https://anonymous.4open.science/r/RINO-95DC`.

1 INTRODUCTION

The discrepancy between simulated training data and real measurements is a fundamental bottleneck for machine learning (ML) in the physical sciences. In high-energy physics (HEP), particles produced in real collisions cannot be labeled with absolute certainty, so supervised classifiers are trained using Monte Carlo (MC) simulations with well-defined collision contents (*i.e.*, labels). However, due to the resulting domain shift, trained models can lose 20% to 40% of their performance when applied to real data (CMS Collaboration, 2025a;b). This gap stems from imperfect modeling of effects such as nonperturbative quantum chromodynamics (QCD) and detector response. Self-supervised learning (SSL) offers a way to reduce the reliance on simulated data by pretraining directly on unlabeled real data and finetuning on labeled samples (Hendrycks et al., 2019; Asano et al., 2020; Gui et al., 2024). To do so, most SSL methods construct augmented *views* of the same input and train the model to treat them as equivalent, with the choice of views implicitly determining the inductive bias of the learned representation. A number of physics-motivated SSL methods have been developed for HEP (Dillon et al., 2022; Zhao et al., 2024; Golling et al., 2024; Leigh et al., 2025; Birk et al., 2024; Katel et al., 2024; Bardhan et al., 2025), utilizing pretext tasks such as learning to align augmented views based on symmetries (Dillon et al., 2022), predicting masked particle constituents (Golling et al., 2024; Leigh et al., 2025), and joint-embedding predictions of held-out subjets or constituents (Katel et al., 2024; Bardhan et al., 2025). These designs encode valuable symmetries; however, they all operate on a single physical scale, ignoring an important structural property of the input.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

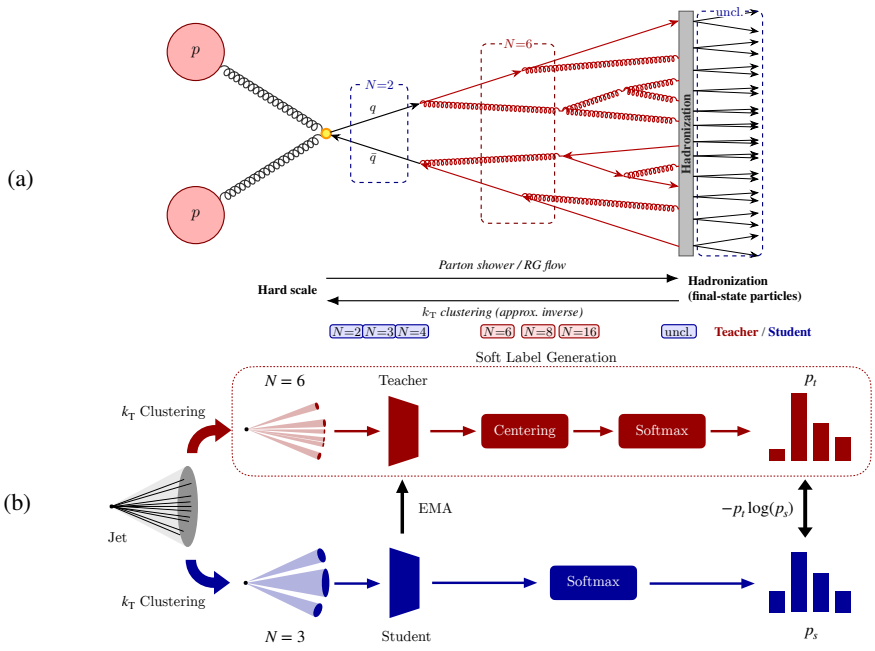

Figure 1: **(a)** A QCD parton shower (here, a $q\bar{q}$ pair radiating softer gluons and quarks) is analogous to a renormalization group (RG) flow down to the hadronization scale. The k_{T} clustering approximately inverts this process, so stopping at different numbers of subjets N generates the multi-scale views used for RINO self-distillation. **(b)** RINO training pipeline (DINO term of Eq. 2). The same jet is clustered at two k_{T} stopping points (here teacher $N{=}6$, student $N{=}3$). The teacher produces soft labels for the student to match via cross-entropy. Teacher parameters are updated from the student via an exponential moving average (EMA). This figure shows one representative student-teacher pair; in practice the loss aligns each student view to every teacher view at a different scale (Sec. 3.2).

A central object of study in HEP is the *jet*: a collimated spray of particles produced by quark and gluon emission at high-energy colliders such as the CERN Large Hadron Collider (LHC) (Andersson, 1987). From an ML perspective, a jet is a variable-length point cloud, while from a physics perspective, it is the observable end of QCD radiation, a stochastic cascade whose kinematic structure is *multi-scale* by construction. Identifying the source of a jet, called "jet tagging", is itself a well-studied ML task with a rich body of supervised methods (Komiske et al., 2019; Battaglia et al., 2016; Moreno et al., 2020; Qu & Gouskos, 2020; Qu et al., 2022b; Gong et al., 2022; Bogatskiy et al., 2024; Spinner et al., 2024; Brehmer et al., 2025; Krause et al., 2026). Concretely, the same jet can be described at different resolutions: as a fine-grained cloud of $\mathcal{O}(30\text{–}100)$ particles; a coarse-grained set of 2–4 subjets that capture only the high-level substructure; or anywhere in between. These descriptions correspond to physically meaningful coarse-graining of the same underlying object, effectively the renormalization group (RG) flow from the high-energy collision scale down to the hadronization scale (Nagy & Soper, 2022). RG structure has been widely studied across the physical sciences, such as cosmology (Pietroni et al., 2012; Ivanov, 2024; Calzetta et al., 1992), materials science (Kmiecik et al., 2016; Wang et al., 2019; Ingólfsson et al., 2014), and fluid dynamics (Frisch, 1995; Grinstein, 2024; Berendsen, 2007), suggesting that methods designed around physical coarse-graining may generalize beyond HEP.

To exploit this fundamental multi-scale property of jets, we propose RINO (**R**enormalization Group D**I**stillation with **NO** Labels), an SSL framework for learning rich and robust jet representations. RINO uses teacher-student self-distillation (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025; Zhang et al., 2019) over jet views generated at different energy scales by a hierarchical clustering algorithm called k_{T} *clustering* (Catani et al., 1993). Training across these scales encourages representations aligned across resolutions, while retaining scale-structured geometry in the backbone.

Our contributions are the following:

- **A physics-informed multi-scale SSL framework.** To the best of our knowledge, RINO is the first SSL framework to align representations of a jet across multiple k_{T} clustering scales.
- **A realistic controlled comparison.** RINO is pretrained exclusively on QCD jets, the dominant background process at hadron colliders, in contrast to prior HEP SSL work that pretrains on simulated mixtures with signal classes overrepresented relative to real data. To probe transfer to deployment-like conditions, we finetune on the smaller JETNET dataset and evaluate out-of-distribution (OOD) on the larger JETCLASS, where JETNET stands in for limited labeled simulation data and JETCLASS for the larger target distribution, with a distribution shift between the two. This protocol finetunes without target-domain labels, matching the realistic downstream task. Under this protocol, RINO outperforms baselines on OOD top tagging by 5.9% over the strongest baseline with lower seed-to-seed variance, and exhibits $\sim 3\times$ smaller 1-Wasserstein distance (W_1) and 6–7$\times$ smaller squared maximum mean discrepancy (MMD^2) between the finetuning and evaluation domains.
- **Ablation studies identifying both DINO and scale-based distillation as core improvements.** We observe that both DINO and SimCLR (Chen et al., 2020) achieve comparable cross-scale invariance as a whole system (*i.e.*, backbone + projection head), but differ in where. DINO delegates it to the projection head while retaining scale-dependent structure in the backbone, whereas the backbone of our JetCLR-scale model is itself nearly scale-invariant. This difference is associated with an 8.6% OOD accuracy gap. We further find that the multi-scale view construction contributes $+11.4\%$ to $+14.0\%$ over single-scale alternatives (Fig. 3).

2 RELATED WORK

Jet Tagging. Jet tagging has been the focus of numerous ML advances using deep sets, graph networks, transformers, and equivariant architectures (Komiske et al., 2019; Battaglia et al., 2016; Qu & Gouskos, 2020; Qu et al., 2022b; Gong et al., 2022; Bogatskiy et al., 2024; Spinner et al., 2024; Brehmer et al., 2025; Krause et al., 2026). These advances are largely supervised and architecture-driven, and therefore depend on labeled MC simulations whose imperfect modeling of detector and physics effects creates a persistent gap when the trained models are deployed on real collision data. RINO instead focuses on the pretraining signal that produces transferable jet representations, learning directly from unlabeled data and using jet tagging only as its primary downstream benchmark.

Learning from Augmented Views. Several SSL methods learn representations by aligning differently augmented views of the same input (Chen et al., 2020; He et al., 2020; Caron et al., 2020; Bardes et al., 2022; Zbontar et al., 2021). DINO (Caron et al., 2021) advances this via teacher-student self-distillation (Zhang et al., 2019), and DINOv2 (Oquab et al., 2024) adds iBOT-style (Zhou et al., 2022) masked prediction on patch tokens. In computer vision, views come from random cropping and photometric augmentation of dense pixel grids. However, these augmentations do not transfer directly to jets, which are variable-length point clouds with no fixed spatial topology, so the choice of view generation shapes the inductive bias.

SSL in HEP. There has been growing interest in applying SSL techniques to HEP, spanning contrastive (Dillon et al., 2022), masked (Golling et al., 2024; Leigh et al., 2025), generative (Birk et al., 2024), and joint-embedding paradigms (Katel et al., 2024; Bardhan et al., 2025). JetCLR (Dillon et al., 2022; Zhao et al., 2024) uses SimCLR-style contrastive learning with physics-motivated augmentations such as rotations, translations, and soft and collinear augmentations. MPMv1 (Golling et al., 2024) tokenizes jet constituents via a VQ-VAE (van den Oord et al., 2017) and predicts discrete token IDs, in analogy with masked language modeling (Devlin et al., 2019), while MPMv2 (Leigh et al., 2025) reconstructs masked constituent features directly. OmniJet-α (Birk et al., 2024) trains an autoregressive decoder on VQ-VAE tokens of jet constituents. J-JEPA (Katel et al., 2024) and HEP-JEPA (Bardhan et al., 2025) adapt the JEPA framework (Assran et al., 2023) to jets, predicting the embeddings of held-out subjets or constituents from the remaining ones. In parallel, jBOT (Tsoi & Rankin, 2026) adapts iBOT to jets and evaluates in-distribution on JETNET. Other approaches include supervised pretraining on auxiliary physics tasks (Li et al., 2024; Mikuni & Nachman, 2025a;b; Mokhtar et al., 2025) and learning from simulation resampling (Harris et al., 2025). While these approaches introduce valuable domain-specific adaptations, they generate views at a single physical

resolution. RINO constructs views from the unclustered particle cloud down to a few subjets via k_{T} clustering and evaluates transfer from labeled simulation to a distinct target, mirroring on-data use.

Multi-Scale and Hierarchical SSL. Multi-scale view construction is a recurring theme in vision SSL. DINO (Caron et al., 2021) combines large global crops with small local crops to encourage local-global consistency, and hierarchical vision transformers such as Swin (Liu et al., 2021) build features at multiple resolutions. These approaches define scale spatially (through pixel support) or architecturally, relying on artificial augmentations as a proxy for the underlying generative process. By contrast, RINO defines scale through the QCD-motivated k_{T} clustering sequence: the number of remaining subjets indexes a physically ordered sequence of resolution scales. The scale axis is therefore inherited from the physics that produces the data, and we analyze the resulting scale structure in Sec. 5.1.

RG Flow and Jet Clustering. A parton shower is a stochastic realization of a QCD renormalization-group flow, in which partons from the hard scattering successively branch into lower-scale partons until the evolution reaches the hadronization scale (Nagy & Soper, 2018). Jet clustering algorithms approximately invert this process by sequentially combining nearby constituents. At each step, the k_{T} algorithm (Catani et al., 1993) merges the pair that minimizes

$$d_{ij} = \min\left(p_{\mathrm{T}i}^2,\, p_{\mathrm{T}j}^2\right) \frac{\Delta R_{ij}^2}{R^2}, \tag{1}$$

or removes constituent i if its beam distance $d_{iB} = p_{\mathrm{T}i}^2$ is smallest, where $\Delta R_{ij}^2 = (\Delta\eta_{ij})^2 + (\Delta\phi_{ij})^2$ and R is the jet radius parameter, so soft and collinear pairs are merged first to mimic the reverse of QCD splitting (Salam, 2010; Thaler & Van Tilburg, 2011; CMS Collaboration, 2025c; Kogler et al., 2019; Bonilla et al., 2022). Stopping at N subjets yields a coarse-grained jet at a characteristic scale (App. A.5; Nagy & Soper, 2022), and varying N traces a physically ordered multi-scale axis for the same jet. We use "RG flow" as a heuristic analogy: k_{T} clustering provides a sequence of coarse-grained jet descriptions ordered by an emission scale. RINO converts this coarse-graining sequence into an SSL strategy by aligning representations *across* clustering scales.

3 Method: The RINO Framework

3.1 Pretraining and Finetuning Data

We pretrain RINO exclusively on the QCD-jet subset of JetClass (Qu et al., 2022a), excluding the simulated signal classes (*e.g.*, top quark, W boson, Z boson, and Higgs boson jets). While prior HEP SSL work pretrains on simulated mixtures of signal classes, real collider data is overwhelmingly composed of QCD jets. Hence, restricting pretraining to QCD jets is more realistic for real-data SSL in HEP. This also allows downstream evaluation to measure generalization to rare signal classes unseen during pretraining. Meanwhile, JetClass provides ground-truth labels for the held-out signal classes used in downstream evaluation, which would otherwise require signature-dependent triggers and physics cuts on real data that could themselves introduce biases.

For finetuning, we use the smaller JetNet (Kansal et al., 2021) as a proxy for limited labeled simulation data and evaluate OOD on the held-out signal classes of JetClass, the larger target distribution. The two datasets differ in p_{T} range, constituent count, and detector simulation, inducing a distribution shift between finetuning and evaluation (App. A.4).

3.2 Jet Clustering as Multi-Scale View Construction

RINO follows DINO's self-distillation framework (Caron et al., 2021; Oquab et al., 2024; Siméoni et al., 2025), but generates physics-informed views, each of which is a kinematic description of the same jet at a different scale. We construct these views by stopping the exclusive k_{T} clustering sequence at varying numbers of subjets N, which integrates out soft and collinear radiation below a corresponding resolution scale (Sec. 2, Fig. 1a). Varying N therefore produces a sequence of views of the same jet at physically ordered resolution scales. We use the discrete ladder $N \in \{2, 3, 4, 6, 8, 16\}$ in addition to the fully unclustered particle cloud (denoted $\mathrm{uncl.}$), with $N \in \{2, 3, 4\}$ covering the typical hard prong multiplicities at the LHC. Clustering histories are precomputed with FastJet (Cacciari et al., 2012) and cached.

During training, the teacher and student receive asymmetric view sets. In the DINO term, the teacher sees only the intermediate-scale views ($N \in \{6, 8, 16\}$), while the student additionally sees the views at the two extremes of the ladder: coarse-grained ($N \in \{2, 3, 4\}$) and unclustered (uncl.). The DINO loss aligns each student view's representation with the teacher's intermediate-scale representations, encouraging multi-scale substructure encoding rather than single-scale specialization. We ablate this choice in App. A.8.

3.3 Setup: Training Objective and Architecture

We adopt the composite pretraining objective of DINOv2 (Oquab et al., 2024) (Fig. 1b), with the teacher parameters tracking the student via an exponential moving average (EMA) (Holt, 2004; Tarvainen & Valpola, 2017; He et al., 2020). The RINO objective is

$$\mathcal{L} = \lambda_{\text{DINO}}\mathcal{L}_{\text{DINO}} + \lambda_{\text{iBOT}}\mathcal{L}_{\text{iBOT}} + \lambda_{\text{KoLeo}}\mathcal{L}_{\text{KoLeo}}. \tag{2}$$

The first term, $\mathcal{L}_{\text{DINO}}$, aligns representations across the view sets (Sec. 3.2) along the parton shower hierarchy, with Sinkhorn–Knopp centering (Sinkhorn & Knopp, 1967; Caron et al., 2020) to prevent collapse. The second term, $\mathcal{L}_{\text{iBOT}}$, provides local supervision via masked-particle prediction: the student predicts the teacher's representation of each particle at masked positions. This serves a dual role: a single-scale masked-prediction baseline analogous to MPM-style methods (Golling et al., 2024; Leigh et al., 2025) that we ablate against in Sec. 4.5, and an auxiliary co-training objective that sharpens representation quality alongside DINO (App. A.8). The third term, $\mathcal{L}_{\text{KoLeo}}$, is the Kozachenko–Leonenko (KoLeo) differential entropy estimator (Beirlant et al., 1997; Delattre & Fournier, 2017; Sablayrolles et al., 2019), which regularizes the embeddings for diversity on the hypersphere. Appendices A.1 and A.2 give the hyperparameters and the explicit loss terms.

Each jet is a point cloud of particle momentum vectors (an $N_{\text{p}} \times 3$ tensor) preprocessed into 7 kinematic features following the Particle Transformer (ParT) (Qu et al., 2022b), fed to a shared student-teacher transformer encoder (Vaswani et al., 2017). A polar-binned positional encoding (PE) indexed by each constituent's $(\Delta\eta, \Delta\phi)$ position breaks the degeneracy among masked iBOT tokens while preserving permutation equivariance. Architecture details are provided in App. A.3.

4 Experiments

4.1 Setup and Controlled Comparison Protocol

All comparisons follow a controlled protocol, a simulated proxy for the intended workflow on experimental data (App. A.13), designed to isolate the contribution of the pretraining strategy. Methods share the same jet transformer encoder backbone, except for the decoder-based OmniJet-α baseline, where we add causal masking to support its GPT-style task (App. A.17). SSL methods are finetuned with the LP-FT (Kumar et al., 2022) strategy, while the supervised baseline is trained end-to-end from random initialization. All methods use the same pretraining data (namely the JetClass QCD class, a proxy for QCD-dominated real data), downstream data splits, finetuning hyperparameters (optimizer, learning-rate scheduler, weight decay; App. A.6), and number of finetuning seeds. We evaluate top tagging (hadronic $t \to bqq$ vs. QCD jets) under the JetNet→JetClass OOD protocol of Sec. 3.1, with binary classification accuracy on the JetClass test set as the primary metric. For each pretrained model, we report two downstream-evaluation levels, each with two metrics. On the one hand, we probe the embedding quality by directly running a k-nearest-neighbors (k-NN; $k = 20$) classifier and a linear probe on the frozen pretrained backbone after re-introducing JetClass top jets and labels.[1] On the other hand, a linear classification head and a two-layer MLP head are finetuned together with the backbone on JetNet labels and evaluated on JetClass's top-tagging task to measure OOD generalization.[2] This hierarchy disentangles representation quality from OOD generalization. Unless otherwise noted, numbers are reported as mean ± standard deviation over 10 finetuning seeds; for pretrained backbones, the k-NN and linear-probe uncertainties are over probe fits on disjoint data subsets.

[1] We would like to point out that such labels are unavailable in deployment on real, unlabeled data, so these two metrics serve only to demonstrate and diagnose embedding quality.

[2] In this case, JetClass top jets and all JetClass labels (QCD and top) are not used in pretraining or finetuning.

4.2 OOD TOP TAGGING

Table 1 summarizes top-tagging accuracy on JETCLASS for RINO, four recent SSL baselines (JetCLR, MPMv1, MPMv2, and OmniJet-α), and supervised training from scratch on JETNET. To further isolate the contribution of multi-scale view construction, we include JetCLR-scale, a controlled variant that replaces JetCLR's augmentations with the same clustering-scale views used by RINO while keeping the contrastive objective (see App. A.17 for details).

RINO is the only method whose MLP head significantly exceeds its linear head ($t = +3.0$, $p \approx 0.01$); no baseline shows the same improvement, and JetCLR and MPMv2 in particular show significant degradation ($t < -3$; App. A.15). This indicates that RINO's representations are robust enough that the additional capacity of the MLP head improves, rather than degrades, OOD accuracy. With the MLP head, RINO demonstrates a 5.9% relative improvement over the strongest baseline (MPMv1) and 7.8% over the supervised model trained on JETNET. We also find that replacing JetCLR's augmentations with k_T views (JetCLR-scale) improves accuracy on the two OOD metrics, demonstrating the benefits of our scale-based augmentations under domain shift. RINO also has the lowest seed-to-seed variance with the MLP head (± 0.003, versus at least ± 0.016 for every other method), and its worst accuracy over five weight-decay and dropout settings (0.843) exceeds the best of supervised training, MPMv1, and MPMv2 (App. A.7), demonstrating robustness to common regularization hyperparameters (weight decay and dropout). When the domain shift is smaller (TOP QUARK TAGGING→JETCLASS), the leading methods saturate near 0.93 in accuracy (App. A.11), yet RINO's proxy scale factors stay closest to unity (App. A.12).

Table 1: Top-tagging accuracy on JETCLASS (top vs. QCD). The linear and MLP heads are finetuned on JETNET only (OOD); k-NN and the linear probe are fit on JETCLASS labels and measure frozen-embedding quality. The bottom block groups all methods using k_T-clustering views (ours), differing only in the training loss. Each entry is $\mu \pm \sigma$; bold marks the best per column; underline marks entries within 1σ of the best.

Method	k-NN	Linear probe	Linear head	MLP head
Supervised	0.722 ± 0.022	0.714 ± 0.030	0.784 ± 0.033	0.786 ± 0.023
SSL baselines				
JetCLR (Dillon et al., 2022)	0.780 ± 0.002	0.793 ± 0.001	0.791 ± 0.048	0.700 ± 0.031
OmniJet-α (Birk et al., 2024)	0.859 ± 0.001	0.869 ± 0.002	0.749 ± 0.080	0.751 ± 0.050
MPMv1 (Golling et al., 2024)	0.882 ± 0.002	0.907 ± 0.002	0.809 ± 0.008	0.800 ± 0.019
MPMv2 (Leigh et al., 2025)	0.896 ± 0.001	0.912 ± 0.001	0.817 ± 0.027	0.760 ± 0.016
k_T-view SSL (ours)				
JetCLR-scale	0.884 ± 0.001	$\underline{0.915 \pm 0.001}$	0.803 ± 0.017	0.780 ± 0.023
RINO	$\mathbf{0.908 \pm 0.001}$	$\mathbf{0.916 \pm 0.001}$	$\mathbf{0.838 \pm 0.009}$	$\mathbf{0.847 \pm 0.003}$

4.3 DATA EFFICIENCY

We sweep the JETNET finetuning data fraction over $\{0.1, 1, 10, 50, 100\}\%$ and compare RINO with representative baselines, under the same finetuning protocol (App. A.6) to ensure the comparison reflects representational differences rather than tuning effort. Figure 2 reports OOD top-tagging accuracy on JETCLASS. RINO finetuned on 1% of labels reaches 0.797 ± 0.015 accuracy, matching fully supervised training on the entire JETNET dataset (0.786 ± 0.023) within uncertainty.

RINO has the leading OOD accuracy at all data fractions and improves monotonically. In contrast, some models are non-monotonic: MPMv2, for example, peaks in OOD performance at 10% of the finetuning dataset. This is consistent with the feature-distortion mechanism of Kumar et al. (2022): when pretrained features are strong and the OOD shift is large, finetuning distorts features along in-distribution (ID) directions. More ID labels drive further distortion, which can explain this non-monotonic behavior. Moreover, JetCLR-scale outperforms JetCLR at all label fractions, so the benefit of cross-view alignment persists across data regimes. Full curves are in App. A.14.

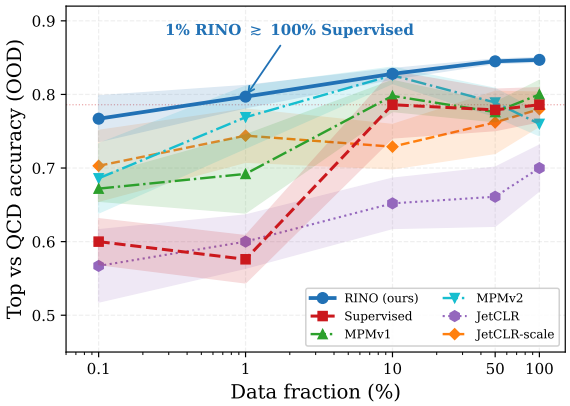

Figure 2: OOD top-tagging accuracy versus JETNET data fraction. RINO with 1% of labels matches supervised training on 100% within uncertainty. Bands show $\pm 1\sigma$ over 10 finetuning seeds (MLP head); the dotted line marks supervised training on 100% of labels.

4.4 CROSS-TOPOLOGY GENERALIZATION

To test whether the learned representations capture general jet substructure rather than task- or dataset-specific features, we evaluate all methods on unseen JETCLASS signal topologies after finetuning only on top tagging from JETNET with a linear head (Table 2). No model sees Higgs samples during any training. We also report AUC as a metric, since the classification threshold is calibrated for the finetuning task and does not necessarily transfer to unseen signals and topologies.

RINO achieves the highest AUC on both $H \to 4q$ (0.894 ± 0.007) and $H \to b\bar{b}$ (0.788 ± 0.016). The $H \to 4q$ result demonstrates genuine cross-topology generalization. Although finetuned only on 3-prong top-quark jets (3 hard subjets from $t \to bqq$), RINO's linear classifier identifies 4-prong Higgs decays ($H \to 4q$) with low variance, demonstrating that k_{T}-view pretraining learns multi-prong substructure itself rather than only topology-specific features.

On $H \to b\bar{b}$, the leading methods cluster within $\sim 2\%$ AUC of each other. This is expected: $H \to b\bar{b}$ discrimination relies on track-level features (impact parameters and secondary-vertex displacements) that distinguish b-quarks from light quarks. Our kinematic-only inputs cannot access these flavor signatures, so all methods are bounded by what kinematic substructure each backbone preserves.

Table 2: Cross-topology transfer (JETNET→JETCLASS, linear head). Models are finetuned on top tagging only and evaluated on unseen JETCLASS signal topologies. Bold marks the best per column; underline marks entries within one standard deviation of the best.

Method	$H \to 4q$		$H \to b\bar{b}$	
	ACC	AUC	ACC	AUC
Supervised	0.707 ± 0.026	0.800 ± 0.029	0.623 ± 0.020	0.726 ± 0.030
SSL baselines				
JetCLR (Dillon et al., 2022)	0.727 ± 0.067	0.803 ± 0.042	$\mathbf{0.706 \pm 0.041}$	0.782 ± 0.018
MPMv1 (Golling et al., 2024)	0.741 ± 0.027	0.819 ± 0.018	0.612 ± 0.026	0.705 ± 0.017
MPMv2 (Leigh et al., 2025)	0.757 ± 0.033	0.859 ± 0.018	0.648 ± 0.023	0.778 ± 0.010
OmniJet-α (Birk et al., 2024)	0.692 ± 0.053	0.797 ± 0.034	0.603 ± 0.026	0.709 ± 0.032
k_{T}-view SSL (ours)				
JetCLR-scale	0.630 ± 0.020	0.819 ± 0.014	0.575 ± 0.015	0.771 ± 0.010
RINO	$\mathbf{0.821 \pm 0.008}$	$\mathbf{0.894 \pm 0.007}$	0.687 ± 0.025	$\mathbf{0.788 \pm 0.016}$

4.5 DESIGN ABLATIONS

Cross-scale DINO objective. Table 3 progressively adds RINO components on top of an iBOT-only variant (single-scale masked prediction, no cross-scale DINO objective). Adding DINO-based multi-

scale distillation across k_T views yields $+14.0\%$, while a final teacher-schedule refinement adds another $+1.0\%$. Therefore, most of the improvement comes from the cross-scale DINO objective. The full progressive decomposition, including view-assignment and PE choices, is in App. A.8.

Multi-scale view alignment. JetCLR-scale and RINO use the same multi-scale views from k_T but differ in the training objective (Table 1). JetCLR-scale outperforms the original JetCLR (0.780 vs. 0.700, $+11.4\%$), confirming that k_T views account for most of the gain regardless of the SSL family. The remaining gap to RINO ($+7.6\%$; $+8.6\%$ for the final model) reflects the benefit of self-distillation over contrastive alignment (discussed in detail in Sec. 5.1).[3]

Clustering algorithm. To isolate the role of the energy-ordered k_T scale sequence, we train variants using Cambridge/Aachen (C/A) clustering (Dokshitzer et al., 1997), which groups by angular proximity, and anti-k_T (Cacciari et al., 2008), which clusters the hardest constituents first. Under the same training objective, C/A yields 0.796 ± 0.012 and anti-k_T yields 0.693 ± 0.026, compared to k_T's 0.839 ± 0.005. All three algorithms produce multi-scale views, but they differ in how they define resolution: k_T coarsens by energy scale (approximately reversing the parton shower), C/A by angular proximity, and anti-k_T by hardness. The ranking $k_T >$ C/A $>$ anti-k_T is consistent with the finding that k_T subjets align more closely with the prongs from the hard process than C/A or anti-k_T subjets (CMS Collaboration, 2025c), confirming that the SSL views help most when the coarse-graining axis matches the physical process that generated the data.

Table 3: Ablation: contribution of DINO self-distillation. All variants share the same polar-binned PE, backbone, and finetuning protocol. Arrows give the relative gain over the previous row.

Configuration	**OOD top ACC**
iBOT only	0.736 ± 0.028
+ Cross-scale self-distillation	0.839 ± 0.005 $\uparrow 14.0\%$
+ Teacher HP tuning	0.847 ± 0.003 $\uparrow 1.0\%$

5 Discussion

5.1 Why Cross-Scale Training Helps

Two controlled comparisons isolate the contributions of k_T views and the SSL objective, as shown in Fig. 3. Replacing JetCLR's augmentations with k_T views while keeping the contrastive loss (JetCLR-scale) improves accuracy from 0.700 to 0.780 ($+11.4\%$). Adding cross-scale distillation over k_T views to the iBOT-only variant gives $+14.0\%$ ($0.736 \to 0.839$). Switching from contrastive alignment to self-distillation adds a further $+5.1\%$ (JetCLR $\to$ iBOT-only) and $+7.6\%$ (JetCLR-scale $\to$ RINO before teacher tuning, 0.839; Table 3).

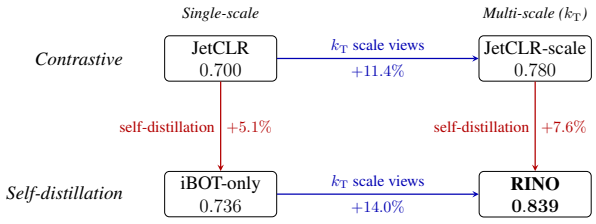

Figure 3: Factorization of RINO's OOD accuracy gain. The top horizontal arrow replaces JetCLR augmentations with k_T scale views; the bottom horizontal arrow adds DINO-based multi-scale distillation across k_T views. Vertical arrows compare contrastive and self-distillation objectives. RINO is shown before teacher tuning (0.839, Table 3); the final model reaches 0.847 (Table 1).

Perhaps surprisingly, the mean cosine similarity across k_T clustering scales (App. A.9, Fig. 5) shows that RINO's backbone is *not* scale-invariant, while the entire model (backbone and head combined)

[3]We compare against the model before teacher tuning (0.839, Table 3) because it shares the pretraining recipe of the iBOT-only variant, whereas teacher tuning was applied only to the final model.

is one of the *most invariant*. Mean cosine similarity alone does not tell the whole story, though, since low similarity can also be inherited from input differences; the structure of the similarity matrix matters. Unlike the scale-ignorant MPMv2 backbone (monotonic decay with distance along the k_{T} ladder) and the scale-invariant JetCLR-scale backbone (all cosines near 1), RINO's backbone is scale-aware: it shows a block structure that follows the student-teacher role assignment. This is permitted by projection-head shielding (Bordes et al., 2023): the loss only constrains $h \circ f$, so the backbone f is free to develop any geometry the projection head h can absorb. RINO's invariance is therefore non-trivial: the head realizes invariance on top of a scale-aware backbone. By contrast, the JetCLR-scale backbone is itself nearly scale-invariant (all cosines ≥ 0.89). For OOD top tagging, the 8.6% accuracy gap between RINO and JetCLR-scale (same views, same architecture) suggests that scale-aware backbones outperform scale-invariant ones. Preserving scale information matters: for example, the top-tagging discriminative signal *is* how jet substructure varies with scale. Top-quark jets resolve into three prongs at $N \sim 3$ and develop characteristic soft-radiation patterns at relatively fine scales, whereas QCD jets do neither.

5.2 Why RINO Generalizes OOD

To measure the JetNet→JetClass domain shift in representation space, we compute the 1-Wasserstein distance (W_1) and squared maximum mean discrepancy (MMD^2) between JetNet and JetClass finetuned embeddings, separately for QCD and top-quark jets (implementation details in App. A.16). Both metrics quantify how far source-domain finetuned representations sit from the target distribution. Table 4 shows that RINO has the smallest W_1 and MMD^2 of all methods (W_1=0.043 and MMD^2=0.007 for QCD, versus at least 0.132 and 0.045 for the baselines), alongside the highest OOD accuracy. RINO's shift is already small before finetuning and does not grow during finetuning (App. A.16).

Table 4: Domain shift in finetuned representations (JetNet vs. JetClass). Lower W_1 and MMD^2 indicate more domain-invariant representations.

Method	W_1 (QCD) ↓	W_1 (top) ↓	MMD^2 (QCD) ↓	MMD^2 (top) ↓	OOD ACC ↑
Supervised	0.746	0.728	0.515	0.564	0.786 ± 0.023
JetCLR	0.132	0.127	0.196	0.191	0.700 ± 0.031
MPMv1	0.241	0.262	0.133	0.162	0.800 ± 0.019
MPMv2	0.375	0.348	0.338	0.305	0.760 ± 0.016
OmniJet-α	0.815	0.752	0.571	0.552	0.751 ± 0.050
JetCLR-scale	0.133	0.145	0.045	0.048	0.780 ± 0.023
RINO	**0.043**	**0.044**	**0.007**	**0.007**	$\mathbf{0.847 \pm 0.003}$

5.3 Limitations

First, our simulation-to-simulation proxy (JetNet→JetClass) may not faithfully reflect the actual simulation-to-data shift. Appendix A.13 outlines the intended workflow on experimental data, but direct real-data validation requires involved physics analyses (*e.g.*, control-region selection, tag-and-probe) and is beyond this work's scope. Second, SSL methods pretrain on JetClass QCD, so pretraining partially overlaps with the test domain; the genuine OOD challenge comes from unseen signal topologies and the mismatch in simulation settings between JetNet and JetClass QCD. Third, the k_{T} inductive bias is task-dependent: RINO's advantage is largest for multi-prong substructure ($t \to bqq$, $H \to 4q$) and narrows for tasks driven by other signatures (*e.g.*, $H \to b\bar{b}$).

6 Conclusion

RINO constructs SSL views at multiple k_{T} clustering scales of the same jet and trains via DINO-style self-distillation. To simulate real-data conditions, we pretrain on JetClass QCD only and finetune on the smaller labeled JetNet before evaluating OOD on JetClass. RINO outperforms baselines on OOD top tagging (Table 1), matches supervised training with $100\times$ fewer labels (Fig. 2), and transfers without retraining to unseen $H \to 4q$ jets (Table 2). A factorization study (Fig. 3) attributes this to both multi-scale view alignment and self-distillation, the latter preserving scale-aware backbone geometry (Sec. 5.1).

AI USE STATEMENT

We used generative AI tools for the four purposes selected in the submission form, detailed below together with how the authors checked the results.

Aiding or polishing writing. We used generative AI tools to edit the paper for readability and to format references. Every AI-suggested edit was shown to the authors as a diff and applied only after their review, modification, and approval.

Retrieval and discovery. We used generative AI tools to search for and identify references. The authors required the tools to quote the supporting passage of each source verbatim, verified these quotes against the source, and checked that each citation supports the claim it is attached to. Every reference entry was checked against its arXiv, INSPIRE, or DOI record.

Research ideation or execution. We used generative AI tools to:

- execute and provide feedback on experiments;
- implement methods in a limited capacity, namely small fixes, bug fixes, and speed-only training optimizations to the authors' code in `https://anonymous.4open.science/r/RINO-95DC`, adapting the open-source baseline implementations to our shared training and evaluation framework, and adapting JetCLR to the k_{T}-view JetCLR-scale variant;
- fix bugs in the data-preprocessing code;
- assist in interpreting results and in checking interpretations, including writing scripts to compute summary statistics (means and standard deviations);
- create and edit software code (*e.g.*, scripts, job configurations, and plotting code); and
- create and modify scientific figures.

The core RINO implementation and the preprocessing pipeline were written by the authors and were largely complete before generative AI tools were used in this project. AI-assisted code, including plotting code, was reviewed and tested by the authors and validated through training and evaluation runs. Derived numbers (*e.g.*, relative gains and test statistics) were recomputed with scripts rather than by hand and checked against the experiment outputs, and implementation details stated in the paper were checked against the code and configuration files. Methodology and results were discussed in weekly group meetings, so that multiple authors reviewed them. We have *not* used generative AI tools to propose or refine hypotheses, develop theoretical models or conceptual frameworks, or assist with translation; generating synthetic datasets, formulating or proving mathematical claims, and qualitative or thematic data analysis are not applicable to this work.

Drafting sections of the paper. We used generative AI tools to draft parts of the text, which the authors substantially rewrote. Every drafted change was reviewed and approved by the authors before inclusion, and multiple authors then independently reviewed the full manuscript or its core parts.

Review and responsibility. The authors have reviewed all AI-assisted work as detailed above. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

RINO aims to reduce reliance on MC simulation by enabling pretraining on real, unlabeled collider data, with no direct societal risks specific to this work.

REPRODUCIBILITY STATEMENT

Code is available at `https://anonymous.4open.science/r/RINO-95DC`. Appendix A.1 lists the pretraining hyperparameters, App. A.2 the loss terms, App. A.3 the architecture and input features, App. A.4 the datasets and preprocessing, App. A.6 the finetuning protocol, App. A.17 the baseline implementations, and App. A.18 the compute resources. All datasets are public (Qu et al., 2022a; Kansal et al., 2021; Kasieczka et al., 2019b).

REFERENCES

Bo Andersson. The Lund Model. *Nucl. Phys. A*, 461:513C, 1987. doi: 10.1016/0375-9474(87)90510-0.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1esx6EYvr.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://arxiv.org/abs/2301.08243.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Jai Bardhan, Radhikesh Agrawal, Abhiram Tilak, Cyrin Neeraj, and Subhadip Mitra. HEP-JEPA: A foundation model for collider physics using joint embedding predictive architecture, February 2025.

James Barnard, Edmund Noel Dawe, Matthew J. Dolan, and Nina Rajcic. Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks. *Phys. Rev. D*, 95(1):014018, 2017. doi: 10.1103/PhysRevD.95.014018.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/3147da8ab4a0437c15ef51a5cc7f2dc4-Paper.pdf.

Jan Beirlant, Edward J Dudewicz, László Györfi, and Edward C van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

Herman J. C. Berendsen. Coarse graining from particles to fluid dynamics. In *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*, pp. 279–296. Cambridge University Press, Cambridge, 2007.

Joschka Birk, Anna Hallin, and Gregor Kasieczka. OmniJet-α: the first cross-task foundation model for particle physics. *Mach. Learn. Sci. Tech.*, 5:035031, 2024. doi: 10.1088/2632-2153/ad66ad.

Alexander Bogatskiy, Timothy Hoffman, David W. Miller, Jan T. Offermann, and Xiaoyang Liu. Explainable equivariant neural networks for particle physics: PELICAN. *JHEP*, 03:113, 2024. doi: 10.1007/JHEP03(2024)113.

Johan Bonilla, Grigorios Chachamis, Barry M. Dillon, Sergei V. Chekanov, Robin Erbacher, Loukas Gouskos, Andreas Hinzmann, Stefan Höche, B. Todd Huffman, Ashutosh V. Kotwal, Deepak Kar, Roman Kogler, Clemens Lange, Matt LeBlanc, Roy Lemmon, Christine McLean, Benjamin Nachman, Mark S. Neubauer, Tilman Plehn, Salvatore Rappoccio, Debarati Roy, Jennifer Roloff, Giordon Stark, Nhan Tran, Marcel Vos, Chih-Hsiang Yeh, and Shin-Shan Yu. Jets and Jet Substructure at Future Colliders. *Frontiers in Physics*, 10:897719, 2022. doi: 10.3389/fphy.2022.897719.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research (TMLR)*, 2023. URL https://openreview.net/forum?id=ZgXfXSz51n.

Johann Brehmer, Víctor Bresó, Pim de Haan, Tilman Plehn, Huilin Qu, Jonas Spinner, and Jesse Thaler. A Lorentz-equivariant transformer for all of the LHC. *SciPost Phys.*, 19(4):108, 2025. doi: 10.21468/SciPostPhys.19.4.108.

Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04: 063, 2008. doi: 10.1088/1126-6708/2008/04/063.

Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. FastJet User Manual. *Eur. Phys. J. C*, 72:1896, 2012. doi: 10.1140/epjc/s10052-012-1896-2.

Esteban Calzetta, Mario Castagnino, and Roman Scoccimarro. Coarse graining approach to quantum cosmology. *Phys. Rev. D*, 45:2806–2813, 1992. doi: 10.1103/PhysRevD.45.2806.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

S. Catani, Yu.L. Dokshitzer, M.H. Seymour, and B.R. Webber. Longitudinally-invariant k_{T}-clustering algorithms for hadron-hadron collisions. *Nucl. Phys. B*, 406:187, 1993. ISSN 0550-3213. doi: 10.1016/0550-3213(93)90166-M.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597, 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

CMS Collaboration. Performance of heavy-flavour jet identification in Lorentz-boosted topologies in proton-proton collisions at $\sqrt{s} = 13$ TeV. *Journal of Instrumentation*, 20(11):P11006, November 2025a. doi: 10.1088/1748-0221/20/11/P11006. URL https://doi.org/10.1088/1748-0221/20/11/P11006.

CMS Collaboration. Calibration of the top and W jet tagging efficiency in 13 TeV data collected by the CMS experiment in 2016–2018. 2025b. URL https://cds.cern.ch/record/2929534.

CMS Collaboration. A method for correcting the substructure of multiprong jets using the Lund jet plane. *JHEP*, 11:038, 2025c. doi: 10.1007/JHEP11(2025)038.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

Sylvain Delattre and Nicolas Fournier. On the Kozachenko–Leonenko entropy estimator. *Journal of Statistical Planning and Inference*, 185:69–93, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Barry M. Dillon, Gregor Kasieczka, Hans Olischlager, Tilman Plehn, Peter Sorrenson, and Lorenz Vogel. Symmetries, safety, and self-supervision. *SciPost Phys.*, 12:188, 2022. doi: 10.21468/SciPostPhys.12.6.188.

Yu.L. Dokshitzer, G.D. Leder, S. Moretti, and B.R. Webber. Better jet clustering algorithms. *Journal of High Energy Physics*, 1997(08):001, September 1997. doi: 10.1088/1126-6708/1997/08/001. URL https://doi.org/10.1088/1126-6708/1997/08/001.

Uriel Frisch. *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press, 1995.

Tobias Golling, Lukas Heinrich, Michael Kagan, Samuel Klein, Matthew Leigh, Margarita Osadchy, and John Andrew Raine. Masked particle modeling on sets: towards self-supervised high energy physics foundation models. *Mach. Learn. Sci. Tech.*, 5:035074, 2024. doi: 10.1088/2632-2153/ad64a8.

Shiqi Gong, Qi Meng, Jue Zhang, Huilin Qu, Congqiao Li, Sitian Qian, Weitao Du, Zhi-Ming Ma, and Tie-Yan Liu. An efficient Lorentz equivariant graph neural network for jet tagging. *Journal of High Energy Physics*, 2022(7), July 2022. ISSN 1029-8479. doi: 10.1007/jhep07(2022)030. URL http://dx.doi.org/10.1007/JHEP07(2022)030.

Fernando F. Grinstein. Coarse grained simulation and dynamic bridging for turbulent mixing predictions. *Computers & Fluids*, 271:106185, 2024. ISSN 0045-7930. doi: 10.1016/j.compfluid.2024.106185. URL https://www.sciencedirect.com/science/article/pii/S0045793024000173.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46:9052, 2024. doi: 10.1109/TPAMI.2024.3415112.

Philip Harris, Jeffrey Krupa, Michael Kagan, Benedikt Maier, and Nathaniel Woodward. Resimulation-based self-supervised learning for pretraining physics foundation models. *Phys. Rev. D*, 111:032010, 2025. doi: 10.1103/PhysRevD.111.032010.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/a2b15837edac15df90721968986f7f8e-Paper.pdf.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *Int. J. Forecast.*, 20:5, 2004. doi: 10.1016/j.ijforecast.2003.09.015.

Helgi I. Ingólfsson, Cesar A. Lopez, Jaakko J. Uusitalo, Djurre H. de Jong, Srinivasa M. Gopal, Xavier Periole, and Siewert J. Marrink. The power of coarse graining in biomolecular simulations. *WIREs Computational Molecular Science*, 4(3):225–248, 2014. doi: 10.1002/wcms.1169.

Mikhail M. Ivanov. Effective field theory for large-scale structure. In Cosimo Bambi, Leonardo Modesto, and Ilya Shapiro (eds.), *Handbook of Quantum Gravity*, pp. 213–260. Springer Nature Singapore, Singapore, 2024. ISBN 978-981-99-7681-2. doi: 10.1007/978-981-99-7681-2_5. URL https://doi.org/10.1007/978-981-99-7681-2_5.

Raghav Kansal, Javier Duarte, Hao Su, Breno Orzari, Thiago Tomei, Maurizio Pierini, Mary Touranakou, Jean-Roch Vlimant, and Dimitrios Gunopulos. Particle cloud generation with message passing generative adversarial networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 23858. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

Raghav Kansal, Carlos Pareja, Zichun Hao, and Javier Duarte. JetNet: A Python package for accessing open datasets and benchmarking machine learning methods in high energy physics. *Journal of Open Source Software*, 8(90):5789, 2023. doi: 10.21105/joss.05789. URL https://joss.theoj.org/papers/10.21105/joss.05789.

Gregor Kasieczka, Tilman Plehn, Anja Butter, Kyle Cranmer, Dipsikha Debnath, Barry M. Dillon, Malcolm Fairbairn, Darius A. Faroughy, Wojtek Fedorko, Christophe Gay, Loukas Gouskos, Jernej Fesel Kamenik, Patrick Komiske, Simon Leiss, Alison Lister, Sebastian Macaluso, Eric Metodiev, Liam Moore, Benjamin Nachman, Karl Nordström, Jannicke Pearkes, Huilin Qu, Yannik Rath, Marcel Rieger, David Shih, Jennifer Thompson, and Sreedevi Varma. The machine learning landscape of top taggers. *SciPost Physics*, 7(1), July 2019a. ISSN 2542-4653. doi: 10.21468/scipostphys.7.1.014. URL http://dx.doi.org/10.21468/SciPostPhys.7.1.014.

Gregor Kasieczka, Tilman Plehn, Jennifer Thompson, and Michael Russel. Top Quark Tagging Reference Dataset, 2019b. URL https://doi.org/10.5281/zenodo.2603256.

Subash Katel, Haoyang Li, Zihan Zhao, Raghav Kansal, Farouk Mokhtar, and Javier Duarte. Learning symmetry-independent jet representations via jet-based joint embedding predictive architecture, 2024. URL https://arxiv.org/abs/2412.05333.

Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chemical Reviews*, 116(14):7898–7936, Jul 2016. ISSN 0009-2665. doi: 10.1021/acs.chemrev.6b00163. URL https://doi.org/10.1021/acs.chemrev.6b00163.

Roman Kogler, Benjamin Nachman, Alexander Schmidt, Lily Asquith, Emma Winkels, Mario Campanelli, Chris Delitzsch, Philip Harris, Andreas Hinzmann, Deepak Kar, Christine McLean, Justin Pilot, Yuta Takahashi, Nhan Tran, Caterina Vernieri, and Marcel Vos. Jet substructure at the Large Hadron Collider. *Reviews of Modern Physics*, 91(4), December 2019. ISSN 1539-0756. doi: 10.1103/revmodphys.91.045003. URL http://dx.doi.org/10.1103/RevModPhys.91.045003.

Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. Energy Flow Networks: Deep Sets for Particle Jets. *JHEP*, 01:121, 2019. doi: 10.1007/JHEP01(2019)121.

Claudius Krause, Ramon Winterhalder, Matthew Feickert, Benjamin Nachman, John Raine, and HEP ML Community. A living review of machine learning for particle physics, July 2026. URL https://doi.org/10.5281/zenodo.21626667.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Matthew Leigh, Samuel Klein, François Charton, Tobias Golling, Lukas Heinrich, Michael Kagan, Inês Ochoa, and Margarita Osadchy. Is tokenization needed for masked particle modeling? *Mach. Learn. Sci. Tech.*, 6:025075, 2025. doi: 10.1088/2632-2153/addb98.

Congqiao Li et al. Accelerating Resonance Searches via Signature-Oriented Pre-training. 2024. URL https://arxiv.org/abs/2405.12972.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

Vinicius Mikuni and Benjamin Nachman. Solving key challenges in collider physics with foundation models. *Phys. Rev. D*, 111:L051504, 2025a. doi: 10.1103/PhysRevD.111.L051504.

Vinicius Mikuni and Benjamin Nachman. Method to simultaneously facilitate all jet physics tasks. *Phys. Rev. D*, 111(5):054015, 2025b. doi: 10.1103/PhysRevD.111.054015.

Farouk Mokhtar, Joosep Pata, Dolores Garcia, Eric Wulff, Mengke Zhang, Michael Kagan, and Javier Duarte. Fine-tuning machine-learned particle-flow reconstruction for new detector geometries in future colliders. *Phys. Rev. D*, 111(9):092015, 2025. doi: 10.1103/PhysRevD.111.092015.

Eric A. Moreno, Thong Q. Nguyen, Jean-Roch Vlimant, Olmo Cerri, Harvey B. Newman, Avikar Periwal, Maria Spiropulu, Javier M. Duarte, and Maurizio Pierini. Interaction networks for the identification of boosted $H \to b\bar{b}$ decays. *Phys. Rev. D*, 102(1):012010, 2020. doi: 10.1103/PhysRevD.102.012010.

Zoltán Nagy and Davison E. Soper. What is a parton shower? *Phys. Rev. D*, 98:014034, Jul 2018. doi: 10.1103/PhysRevD.98.014034. URL https://link.aps.org/doi/10.1103/PhysRevD.98.014034.

Zoltán Nagy and Davison E. Soper. Multivariable evolution in final state parton shower algorithms. *Phys. Rev. D*, 105:054012, Mar 2022. doi: 10.1103/PhysRevD.105.054012. URL https://link.aps.org/doi/10.1103/PhysRevD.105.054012.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Massimo Pietroni, Gianpiero Mangano, Ninetta Saviano, and Matteo Viel. Coarse-Grained Cosmological Perturbation Theory. *JCAP*, 01:019, 2012. doi: 10.1088/1475-7516/2012/01/019.

Huilin Qu and Loukas Gouskos. Jet tagging via particle clouds. *Physical Review D*, 101(5), March 2020. ISSN 2470-0029. doi: 10.1103/physrevd.101.056019. URL http://dx.doi.org/10.1103/PhysRevD.101.056019.

Huilin Qu, Congqiao Li, and Sitian Qian. JetClass: A large-scale dataset for deep learning in jet physics, 2022a.

Huilin Qu, Congqiao Li, and Sitian Qian. Particle Transformer for Jet Tagging. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 18281, 2022b. URL https://proceedings.mlr.press/v162/qu22b.html.

Yi Ren, Shangmin Guo, Wonho Bae, and Danica J. Sutherland. How to prepare your task head for finetuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gVOXZproe-e.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkGuG2R5tm.

Gavin P. Salam. Towards jetography. *The European Physical Journal C*, 67(3-4):637–686, May 2010. ISSN 1434-6052. doi: 10.1140/epjc/s10052-010-1314-6. URL http://dx.doi.org/10.1140/epjc/s10052-010-1314-6.

Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. doi: 10.2140/pjm.1967.21.343.

Jonas Spinner, Victor Bresó, Pim de Haan, Tilman Plehn, Jesse Thaler, and Johann Brehmer. Lorentz-Equivariant Geometric Algebra Transformers for High-Energy Physics. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 22178. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/277628cff838927d869cd1f671328ce0-Paper-Conference.pdf.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pp. 1195–1204, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68053af2923e00204c3ca7c6a3150cf7-Abstract.html.

Jesse Thaler and Ken Van Tilburg. Identifying Boosted Objects with N-subjettiness. *JHEP*, 03:015, 2011. doi: 10.1007/JHEP03(2011)015.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Herve Jegou. Going deeper with Image Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00010. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00010.

Ho Fung Tsoi and Dylan Rankin. jBOT: Semantic jet representation clustering emerges from self-distillation. *SciPost Physics*, 21(3), September 2026. ISSN 2542-4653. doi: 10.21468/scipostphys.21.3.053. URL http://dx.doi.org/10.21468/SciPostPhys.21.3.053.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E. Charron, Gianni de Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS Central Science*, 5(5):755–767, 2019. doi: 10.1021/acscentsci.8b00913. URL https://doi.org/10.1021/acscentsci.8b00913. PMID: 31139712.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3712–3721, 2019. doi: 10.1109/ICCV.2019.00381.

Zihan Zhao, Farouk Mokhtar, Raghav Kansal, Haoyang Li, and Javier Duarte. Large-Scale Pretraining and Finetuning for Efficient Jet Classification in Particle Physics. In *22nd International Workshop on Advanced Computing and Analysis Techniques in Physics Research*, 2024.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

A APPENDIX

The code is provided in `https://anonymous.4open.science/r/RINO-95DC`.

A.1 PRETRAINING HYPERPARAMETERS

Table 5 summarizes the hyperparameters used for RINO pretraining on the QCD-jet subset of JETCLASS. The teacher EMA momentum and the learning-rate scheduler follow cosine schedules over the full 100 training epochs, with warmup as indicated. The pretraining recipe was held fixed across all RINO ablation variants reported in Sec. 4.5, with only the view-construction parameters and the input positional encoding varied between variants. The headline RINO model in Table 1 (0.847 ± 0.003) uses a tuned teacher schedule, a fixed teacher temperature and a lower final EMA momentum (Table 5), chosen by the OOD top-vs-QCD accuracy on JETCLASS; this step contributes the final $+1.0\%$ row of Table 3, and RINO before tuning (0.839 ± 0.005) already exceeds every baseline (App. A.17). Values listed in Table 5 below are those of this final configuration.

Table 5: RINO pretraining hyperparameters (production configuration).

Group	Hyperparameter	Value
Optimization	Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
	Peak learning rate	3×10^{-3}
	Weight decay	0.05
	Gradient clipping (max norm)	3.0
Schedule	Total epochs	100
	LR warmup	epochs 0–19, $10^{-6} \rightarrow 1.0$ (cosine)
	LR main schedule	epochs 20–99, $1.0 \rightarrow 0.05$ (cosine)
	Layer-wise LR decay	0.9 per layer (backbone)
Batching	Total batch size	3000 jets
	Per-GPU batch size	1500 jets (2 GPUs, data parallel)
DINO loss	Student temperature τ_s	0.10 (fixed)
	Teacher temperature τ_t	0.04 (fixed)
	ablation variants	$0.04 \rightarrow 0.07$, epochs 0–19 (cosine)
	Teacher centering	Sinkhorn–Knopp
	Loss weight	$\lambda_{\mathrm{DINO}} = 1.0$
iBOT loss	Mask ratio	0.25 (unclustered view)
	Mask embedding	learnable d_{model}-dim vector
	Loss weight	$\lambda_{\mathrm{iBOT}} = 1.0$
KoLeo regularizer	Applied to	student `[CLS]` embeddings (teacher-scale views)
	Loss weight	$\lambda_{\mathrm{KoLeo}} = 0.1$
Teacher EMA	Update rule	$\bar{\theta} \leftarrow m\,\bar{\theta} + (1-m)\,\theta$, per batch
	Momentum schedule	$m : 0.995 \rightarrow 0.997$ over 100 epochs (cosine)
	ablation variants	$m : 0.995 \rightarrow 0.999$ over 100 epochs (cosine)

A.2 PRETRAINING LOSS TERMS

This section writes out the three terms of Eq. 2; hyperparameter values are listed in Table 5. For a view x of a jet, $P_s(x)$ and $P_t(x)$ denote the student and teacher output distributions over the K prototypes of a projection head (App. A.3), obtained with temperatures τ_s and τ_t. The teacher distribution is normalized with Sinkhorn–Knopp (Sinkhorn & Knopp, 1967; Caron et al., 2020) and receives no gradient. We write the cross-entropy as

$$H(a, b) = -\sum_{k=1}^{K} a_k \log b_k, \tag{3}$$

and each loss term below is averaged over the jets in a mini-batch.

DINO term. Let $x^{(v)}$ be the jet at view v, $\mathcal{T} = \{6, 8, 16\}$ the teacher views, and $\mathcal{S}$ the student views (the teacher views plus three student-only views sampled per batch from $\{2, 3, 4, \text{uncl.}\}$). Using the DINO head on the `[CLS]` output,

$$\mathcal{L}_{\text{DINO}} = \frac{1}{|\mathcal{P}|} \sum_{(t,s)\in\mathcal{P}} H\big(P_t(x^{(t)}),\, P_s(x^{(s)})\big), \qquad \mathcal{P} = \{(t, s) \in \mathcal{T} \times \mathcal{S} : s \neq t\}, \tag{4}$$

so each student view is aligned with every teacher view (Caron et al., 2021) at a different scale.

iBOT term. The iBOT term acts on the unclustered view x only (App. A.8). A random set $\mathcal{M}$ of 25% of the particles is masked in the student's input, giving $\tilde{x}$, and the teacher sees the unmasked jet. Using the iBOT head on the output token of each masked particle i (Zhou et al., 2022),

$$\mathcal{L}_{\text{iBOT}} = \frac{1}{|\mathcal{M}|} \sum_{i\in\mathcal{M}} H\big(P_t(x)_i,\, P_s(\tilde{x})_i\big). \tag{5}$$

KoLeo term. Following DINOv2 (Oquab et al., 2024; Sablayrolles et al., 2019), for the ℓ_2-normalized student `[CLS]` representations $\hat{z}_1, \ldots, \hat{z}_B$ of the B jets in a mini-batch at the same teacher-scale view,

$$\mathcal{L}_{\text{KoLeo}} = -\frac{1}{B} \sum_{i=1}^{B} \log\Big(\min_{j\neq i} \|\hat{z}_i - \hat{z}_j\|_2\Big), \tag{6}$$

averaged over teacher-scale views. It spreads different jets within a view and never compares two views of the same jet.

A.3 Architecture Details

Backbone. The shared transformer encoder backbone has 8 layers with hidden dimension $d_{\text{model}} = 256$, 16 attention heads, and a feed-forward expansion ratio of 4 (FFN hidden dimension 1024), totaling 8.4M trainable parameters (plus 2.6M parameters in the polar-binned positional encoding lookup table). Activations are SwiGLU (Shazeer, 2020) in the FFN sub-layers, normalization is RMSNorm (Zhang & Sennrich, 2019) applied pre-residual, and residual paths use LayerScale (Touvron et al., 2021) with learnable per-channel scaling initialized to 0.01. A learnable `[CLS]` token is prepended to each input sequence and serves as the global jet representation. Following Darcet et al. (2024), we additionally append four learnable *register* tokens that are not used as outputs but stabilize the attention maps. Variable-length particle sequences are handled with attention masking; the maximum sequence length is 90 particles per jet (the 90 highest-p_{T} particles are kept).

Input features (per particle). Each constituent is represented by the seven features listed in Table 6. All features are pre-standardized through a fixed affine map $x \mapsto (x-\mu)/\sigma$. The per-feature centering μ and scale σ are estimated once from the *median* and *interquartile range* (IQR) of the corresponding distribution on the JetClass QCD pretraining set, rather than from the empirical mean and standard deviation; this robust estimator is insensitive to the long tails typical of jet kinematic spectra. The resulting (μ, σ) values are held fixed during both training and inference, and used unchanged on all downstream datasets, so that the same backbone can be applied without renormalizing the input distribution. Quantities quoted in $(\Delta\eta, \Delta\phi)$ space elsewhere in this appendix (notably the radial parameters of the positional encoding) are in these standardized units, so for example $\Delta\eta = 1$ corresponds to a raw rapidity offset of 0.14.

Positional encoding. The polar-binned positional encoding maps each constituent's $(\Delta\eta, \Delta\phi)$ position to a learned embedding via a 100×100 angular grid (radial $\times$ azimuthal bins) with logarithmic radial spacing centered on the jet axis. Concretely, the radial coordinate $r = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$ is mapped through $\log(1 + r/r_{\text{scale}})$ before quantization into 100 bins on the interval $[0, \log(1 + r_{\max}/r_{\text{scale}})]$, while the azimuthal coordinate $\alpha = \arctan2(\Delta\phi, \Delta\eta)$ is binned uniformly on $[-\pi, \pi)$. The two radial parameters are not free: r_{scale} is set to the median of the per-constituent radial distribution on the pretraining set (so half of all constituents fall in the high-resolution region near the knee of the log compression), and $r_{\max}$ is set to the 99th percentile of the same distribution (so extreme tail constituents are clamped rather than allowed to consume a large share of bins). The

Table 6: Per-particle input features and the fixed standardization (μ, σ) applied by the input pipeline. The model receives $(x - \mu)/\sigma$ for each feature.

Feature	**Definition**	μ	σ
$\log p_{\mathrm{T}}$	$\log p_{\mathrm{T},i}$, constituent transverse momentum	1.7	1.8
$\log E$	$\log E_i$, constituent energy	2.0	1.8
$\log(p_{\mathrm{T}}/p_{\mathrm{T}}^{\mathrm{jet}})$	$\log(p_{\mathrm{T},i}/p_{\mathrm{T}}^{\mathrm{jet}})$, relative p_{T}	-4.7	1.8
$\log(E/E^{\mathrm{jet}})$	$\log(E_i/E^{\mathrm{jet}})$, relative energy	-4.7	1.8
$\Delta\eta$	$\eta_i - \eta_{\mathrm{jet}}$, rapidity offset	0.0	0.14
$\Delta\phi$	$\phi_i - \phi_{\mathrm{jet}}$ (wrapped to $[-\pi, \pi)$), azimuthal offset	0.0	0.14
ΔR	$\sqrt{(\Delta\eta_i)^2 + (\Delta\phi_i)^2}$, angular distance to jet axis	0.14	0.25

two radial parameters are quoted in the standardized $(\Delta\eta, \Delta\phi)$ units of Table 6; on the JETCLASS QCD pretraining set this gives $r_{\mathrm{scale}} \approx 0.96$ and $r_{\max} \approx 5.60$, corresponding to $\sim$0.13 and $\sim$0.78 in raw $(\Delta\eta, \Delta\phi)$ space, within the $R = 0.8$ jet cone. The combination of polar parameterization and log radial binning therefore concentrates resolution where the constituent density is highest, by construction.

Bin collisions matter because the role of the positional encoding in an encoder-only iBOT setup is precisely to break what would otherwise be a complete degeneracy among masked constituents. With a generative decoder (as in MAE-style methods), positional queries fed to the decoder disambiguate which masked token to reconstruct. With our encoder-only architecture, the only signal a masked token carries is its mask embedding plus its positional embedding; without PE, all masked tokens are identical and iBOT cannot distinguish them. A bin collision restores this degeneracy locally: two masked constituents that fall in the same bin become indistinguishable. Empirically, adding polar-binned PE helps with training stability and improves OOD accuracy from 0.828 to 0.839 on cluster views (Table 12), consistent with reduced bin collisions aiding iBOT training.

We measure, for each candidate scheme, the per-particle collision rate (fraction of real constituents sharing a bin with at least one other constituent of the same jet) and the per-jet collision rate (fraction of jets containing at least one such collision). The polar grid is markedly more bin-efficient than a Cartesian grid of comparable size because the constituent $(\Delta\eta, \Delta\phi)$ distribution is concentrated near the jet axis (the median jet has 32 constituents, see App. A.4). A representative comparison is shown in Table 7.

Table 7: Bin-collision rates (%) for positional-encoding schemes across grid sizes, measured on 5×10^5 JETCLASS QCD jets. Lower collision rates reduce ambiguity among masked iBOT tokens. The production choice is Polar 100×100 (bold).

Scheme *(patches)*	20×20 *400*	50×50 *2,500*	80×80 *6,400*	$\mathbf{100\times100}$ ***10,000***	120×120 *14,400*
Per-particle collision rate (%)					
Cartesian	64.76	35.98	22.19	16.82	13.13
Log-Cartesian	42.70	13.61	6.27	4.19	3.00
Polar	32.96	9.87	4.52	**3.03**	2.16
Per-jet collision rate (%)					
Cartesian	99.93	99.44	97.49	94.55	90.13
Log-Cartesian	99.13	83.55	59.55	46.79	37.26
Polar	90.01	59.14	39.59	**31.19**	24.94
Mask–mask collision rate (% of masked particles, 25% mask)					
Cartesian	38.37	14.71	7.42	5.17	3.80
Log-Cartesian	16.99	3.82	1.59	1.02	0.72
Polar	12.38	2.78	1.16	**0.76**	0.53

Three observations emerge. First, all three schemes improve monotonically with grid size, and the same ranking (Polar best, Log-Cartesian intermediate, Cartesian worst) holds at every grid size and on all three metrics. Second, most of the improvement over uniform Cartesian comes from *density-*

matched binning (Log-Cartesian alone reduces the 100×100 per-particle rate from 16.82% to 4.19%, a $4\times$ reduction); the polar parameterization adds a further $\sim1.4\times$ refinement ($4.19\% \to 3.03\%$) on top, by aligning the bin boundaries with the rotational symmetry of the jet. Third, the mask–mask rate (bottom panel) is the operationally relevant figure for iBOT, and it is substantially smaller than the per-particle rate at every grid size: at the production Polar 100×100, only 0.76% of masked particles share a bin with another masked particle (vs. 3.03% per-particle overall), so the degeneracy iBOT actually faces during training is roughly $4\times$ lower than the full per-particle rate would suggest.

Polar 100×100 is a deliberate middle ground. Going further to Polar 120×120 would reduce the per-particle collision rate only from 3.03% to 2.16% and the per-jet rate from 31.19% to 24.94%, while requiring 44% more patches and a correspondingly larger learnable PE table (~1.13M extra parameters at $d_{\text{model}} = 256$). Conversely, dropping to Polar 80×80 would save 36% of the patches but increase the per-particle rate from 3.03% to 4.52%. The 100×100 choice therefore lies at the knee of the diminishing-returns curve, capturing nearly all of the achievable reduction without paying for the long tail of marginal improvements.

Projection heads. Following DINOv2 (Oquab et al., 2024), the DINO and iBOT projection heads are independent three-layer MLPs with hidden width 1024 and bottleneck dimension 256, GELU activations, an ℓ_2-normalization layer, and a weight-normalized linear output of dimension 2048. The DINO head operates on the `[CLS]` representation, whereas the iBOT head operates on per-particle outputs at masked positions.

A.4 Dataset Details

JetClass. JetClass (Qu et al., 2022a) is a large-scale jet classification benchmark with 10 classes spanning QCD, Higgs, electroweak, and top topologies. The hard process is generated with MadGraph5_aMC@NLO, parton showering and hadronization use Pythia, and detector effects are modeled with Delphes in the CMS configuration; jets are clustered from Delphes E-Flow objects with the anti-k_{T} algorithm at $R = 0.8$ and selected with $p_{\text{T}} \in [500, 1000]$ GeV and $|\eta| < 2$. See Qu et al. (2022a;b) for full details. We use the QCD subset (10M training jets) for pretraining and the signal classes for downstream evaluation.

Pretraining set (QCD only). We pretrain exclusively on the QCD subset (10M training jets), excluding all signal classes. The median QCD jet contains 32 constituents (mean 35.2 ± 17.2; Fig. 4): only 11% of jets have ≤ 16 constituents and 0.13% have ≤ 6, confirming that the coarse views ($N \in \{2, 3, 4\}$) are genuine coarse-grainings for nearly all jets. Subjet clustering is computed once during preprocessing with FastJet (Cacciari et al., 2012) using the exclusive k_{T} algorithm with jet radius $R = 0.8$, and the four-momenta of the resulting subjets are stored on disk for the stopping points $N \in \{2, 3, 4, 6, 8, 16\}$.

Downstream evaluation sets. We finetune on JetNet (Kansal et al., 2021; 2023) ($\sim$360K jets covering gluon, light-quark, and top topologies, with jet $p_{\text{T}} \approx 1$ TeV, restricted to 0.8–1.6 TeV, and at most 30 constituents per jet, zero-padded). JetNet parton-level events are generated with MadGraph5_aMC@NLO, then showered and hadronized with Pythia (Monash 2013 tune), with a simplified detector-level description instead of Delphes. It therefore differs from JetClass in p_{T} range and detector-level simulation. We additionally use the Top Quark Tagging dataset (Kasieczka et al., 2019a;b) (~1.2M training jets: top vs. QCD, with $p_{\text{T}} \in [550, 650]$ GeV and up to 200 constituents), which uses Pythia for shower/hadronization and Delphes for detector effects with a different (ATLAS-like) configuration than JetClass. For OOD evaluation we test on the JetClass test set, restricted to the relevant signal class versus QCD (*e.g.*, top vs. QCD for the main comparison). All downstream evaluation samples are disjoint from the pretraining sample; the model never sees their labels during pretraining and never sees their non-QCD signal classes at all. The JetNet$\to$JetClass transfer is OOD in three respects: different p_{T} range, different constituent count distribution (at most 30 vs. up to 183), and different detector-level simulation (simplified vs. Delphes).

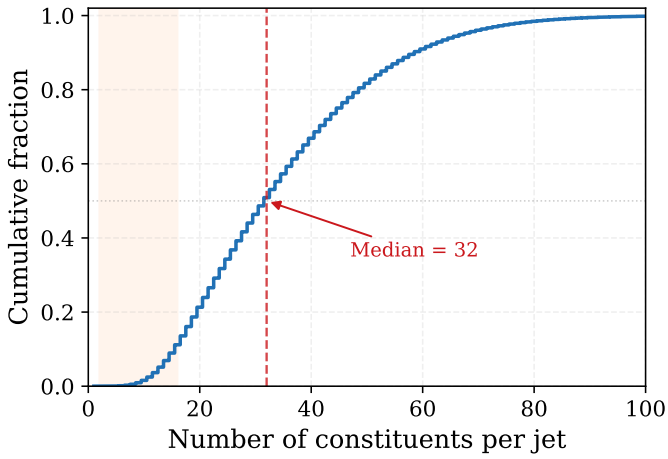

Figure 4: Cumulative distribution of constituent multiplicity for QCD jets in JETCLASS (12.5M jets across train, validation, and test splits). The shaded region marks the k_T view range $N \in \{2, \ldots, 16\}$; only 11% of jets have ≤ 16 constituents, confirming that the coarse views are genuine coarse-grainings.

A.5 PHYSICAL SCALES OF THE k_T VIEWS

The hierarchy is not learned but imposed by the number of subjets per view, N. Each clustering step does indeed have a meaningful energy scale: the $N{+}1 \to N$ merge scale $q_N = \sqrt{d_{\text{merge}}(N)}$ (Catani et al., 1993), where $d_{\text{merge}}(N)$ is the k_T distance (d_{ij} of Eq. 1 or d_{iB}) at the clustering step from $N{+}1$ to N subjets. For JETCLASS QCD jets with more than 16 constituents, we report the median and IQR of q_N and of $p_T^{\text{soft}}/p_T^{\text{jet}}$, the transverse momentum of the softest of the N subjets divided by that of the parent jet (Table 8).

Table 8: Physical scales of the exclusive-k_T views: merge scale q_N and softest-subjet momentum fraction, median [IQR] over JETCLASS QCD jets with more than 16 constituents.

N	q_N [GeV]	$p_T^{\text{soft}}/p_T^{\text{jet}}$
2	9.91 [6.07, 17.09]	0.23 [0.12, 0.36]
3	5.70 [3.69, 9.01]	0.088 [0.042, 0.156]
4	3.89 [2.57, 5.99]	0.044 [0.021, 0.084]
6	2.25 [1.48, 3.39]	0.018 [0.009, 0.034]
8	1.49 [0.95, 2.26]	0.010 [0.006, 0.019]
16	0.46 [0.24, 0.76]	0.0033 [0.0021, 0.0054]

The median scale q_N falls monotonically from 9.91 to 0.46 GeV, while the resolved softest-subjet fraction falls from 0.23 to 0.0033. Thus, the fixed N does indeed correspond to a meaningful distribution of physical scales. Two results show that RINO benefits from this ordering. First, the scale-structure analysis (App. A.9) finds that the backbone retains resolution information while the projection head aligns across scales. Second, with all other choices fixed, the intended teacher $\{6, 8, 16\}$/student-only $\{2, 3, 4, \text{uncl.}\}$ assignment gives 0.839 ± 0.005, versus 0.823 ± 0.006 when the roles are swapped and 0.730 ± 0.009 when both roles mix scales (Table 12).

A.6 FINETUNING PROTOCOL

Table 9 gives the locked finetuning recipe used in all main results in Sec. 4, identical across pretrained backbones. The supervised baseline shares the same optimizer, scheduler, and head architecture but is trained end-to-end from random initialization without the frozen-backbone warmup. For SSL methods we adopt the LP-FT (linear-probe-then-finetune) strategy (Kumar et al., 2022; Ren et al., 2023): the classification head is trained for the first 5 epochs with the backbone frozen, after which

the backbone is unfrozen and trained jointly with the head at the same learning rate. This was found to give substantially smaller seed-to-seed variance than full finetuning from epoch 0, while matching its peak accuracy. Accuracy uses a fixed decision threshold of 0.5. The JETCLASS test sets are class-balanced, while JETNET has a QCD:top ratio of 2:1, which the positive-class weight of 2.0 compensates (Table 9).

Table 9: Finetuning hyperparameters (locked across all backbones and datasets).

Hyperparameter	**Value**
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Head learning rate	1×10^{-4}
Backbone learning-rate factor (after unfreeze)	1.0 (*i.e.*, same as head)
Weight decay	0.05
LP-FT: backbone freeze epochs	5
LR warmup epochs	10
LR scheduler	cosine with warmup
Dropout (in head)	0.1
Label smoothing	0.0
Early-stopping patience	5 epochs (on validation loss)
BCE positive-class weight	2.0 for JETNET; 1.0 for TOP QUARK TAGGING
Head architecture	2-layer MLP, hidden $[256, 128]$, ReLU, dropout 0.1
Pooling	mean over particle tokens (for backbones with PE)
Seeds per configuration	10 for JETNET and 6 for TOP QUARK TAGGING

A.7 ROBUSTNESS TO FINETUNING REGULARIZATION

All main results use one finetuning recipe for every method (Table 9).[4] To test whether RINO's lead depends on this shared recipe, we vary the weight decay and dropout over five (weight decay, dropout) settings, $(0.01, 0.1)$, $(0.05, 0.1)$, $(0.10, 0.1)$, $(0.05, 0.3)$, and $(0.05, 0.5)$, for supervised training, MPMv1, MPMv2, and RINO with the default backbone and 100% of the JETNET labels, keeping all other settings of Table 9. Table 10 reports the default setting (the Table 1 value) together with the lowest (Worst) and highest (Best) mean OOD accuracy across the five settings. Best is selected on JETCLASS accuracy, which requires target labels that are unavailable in applications to real data, so it is a retrospective best case rather than a deployable choice.

For supervised training and MPMv1, no setting improves on the default, so the shared recipe does not disadvantage them. RINO's worst setting (0.843) exceeds the best setting of every other method, and its default leads the strongest target-selected baseline (MPMv2, 0.810) by 0.037. Its accuracy spans 0.009 across the settings, comparable to MPMv1 (0.010) and smaller than supervised training (0.015) and MPMv2 (0.050).

Table 10: Sensitivity of OOD top-tagging accuracy ($\mu \pm \sigma$ over 10 finetuning seeds; JETNET→JETCLASS, MLP head, 100% of JETNET labels) to the finetuning weight decay and dropout. Default is the Table 1 value (weight decay 0.05, dropout 0.1); Worst and Best are the lowest and highest mean over the five settings of App. A.7. Best is selected on JETCLASS accuracy and is a retrospective upper bound. Bold marks the best per column.

Method	**Default**	**Worst**	**Best (target-selected)**
Supervised	0.786 ± 0.023	0.771 ± 0.033	0.786 ± 0.023
MPMv1 (Golling et al., 2024)	0.800 ± 0.019	0.790 ± 0.018	0.800 ± 0.019
MPMv2 (Leigh et al., 2025)	0.760 ± 0.016	0.760 ± 0.016	0.810 ± 0.017
RINO	$\mathbf{0.847 \pm 0.003}$	$\mathbf{0.843 \pm 0.005}$	$\mathbf{0.852 \pm 0.003}$

[4]The supervised baseline uses the same hyperparameters but is trained end-to-end, without the LP-FT frozen-backbone warmup (App. A.6).

A.8 Extended Ablation Results

Table 11 traces the progressive construction of the RINO recipe. Table 12 reports independent comparisons along the positional-encoding, view-assignment, and clustering-algorithm axes. In the view sets, uncl. denotes the unclustered jet (all particles). OOD ACC (accuracy) entries are mean $\pm$ standard deviation over 10 seeds with an MLP head; the k-NN and linear-probe (LP) columns are the frozen-backbone diagnostics of Sec. 4.1. The final model (row 5) uses a refined teacher schedule (fixed $\tau_t = 0.04$, final EMA momentum 0.997) compared to rows 1–4, which use a cosine warmup to $\tau_t = 0.07$ and final momentum 0.999.

Table 11: Progressive construction of the RINO recipe. Each row builds on the previous row.

Step	Configuration	k-NN	LP	OOD ACC
1.	DINO only, mixed views, no PE	0.859	0.923	0.677 ± 0.016
2.	+ iBOT	0.897	0.918	0.776 ± 0.018 ↑ 14.6%
3.	+ k_{T} multi-scale views, p_{T}-rank PE	0.905	0.916	0.828 ± 0.004 ↑ 6.7%
4.	+ Polar-binned PE	0.919	0.919	0.839 ± 0.005 ↑ 1.3%
5.	+ Teacher HP tuning (production)	0.908	0.916	0.847 ± 0.003 ↑ 1.0%

Table 12: Independent comparisons of design choices. Each block varies one axis while holding others fixed. Arrows give the relative change (%) with respect to the first row of each block.

Axis	Configuration	k-NN	LP	OOD ACC
PE choice | mixed views, DINO+iBOT				
	No PE	0.897	0.918	0.776 ± 0.018
	p_{T}-rank	0.902	0.919	0.777 ± 0.028 ↑ 0.1
	Polar-binned	0.909	0.922	0.730 ± 0.009 ↓ 5.9
PE choice | k_{T} multi-scale views, DINO+iBOT				
	p_{T}-rank	0.905	0.916	0.828 ± 0.004
	Polar-binned	0.919	0.919	0.839 ± 0.005 ↑ 1.3
View assignment | polar-binned PE, DINO+iBOT				
(sets shown are teacher views and student-only views; the student also sees the teacher's views)				
	Teacher $\{6, 8, 16\}$**, student** $\{2, 3, 4, \mathrm{uncl.}\}$	0.919	0.919	0.839 ± 0.005
	Swapped: teacher $\{2, 3, 4\}$, student $\{6, 8, 16, \mathrm{uncl.}\}$	0.903	0.922	0.823 ± 0.006 ↓ 1.9
	Mixed-scale: teacher $\{2, 4, 8, 16\}$, student $\{3, 6, \mathrm{uncl.}\}$	0.909	0.922	0.730 ± 0.009 ↓ 13.0
Clustering algorithm | polar-binned PE, DINO+iBOT				
	k_{T}	0.919	0.919	0.839 ± 0.005
	C/A	0.878	0.909	0.796 ± 0.012 ↓ 5.1
	Anti-k_{T}	0.799	0.906	0.693 ± 0.026 ↓ 17.4

Mixed-scale views. The "mixed-scale" view assignment reshuffles clustering scales between teacher and student so that both roles span the full range: the teacher sees $\{2, 4, 8, 16\}$ and the student additionally sees $\{3, 6, \mathrm{uncl.}\}$. This preserves the use of multi-scale k_{T} views but breaks the hierarchy in which the teacher is restricted to resolved intermediate scales and the student-only views sit at both extremes (few subjets and unclustered). The 13.0% drop from hierarchical to mixed-scale views under the same DINO+iBOT loss (Table 12) shows that the specific hierarchical assignment is the important design choice.

iBOT loss scale. In both RINO and the iBOT-only ablation, the iBOT loss is applied only on the full unclustered particle view. We do not train iBOT to reconstruct masked constituents in the N-subjet clustered views, so that the multi-scale part of RINO is isolated in the DINO objective. Applying masked prediction across clustered views is a natural future extension.

DINO and iBOT are complementary. Neither $\mathcal{L}_{\mathrm{DINO}}$ nor $\mathcal{L}_{\mathrm{iBOT}}$ produces a strong representation alone: DINO-only (mixed views, no PE) reaches 0.677 and iBOT-only reaches 0.736 (Table 11, row 1; Table 3). Combined, they reach 0.776 (mixed views) and 0.839 (k_{T} views), exceeding what either

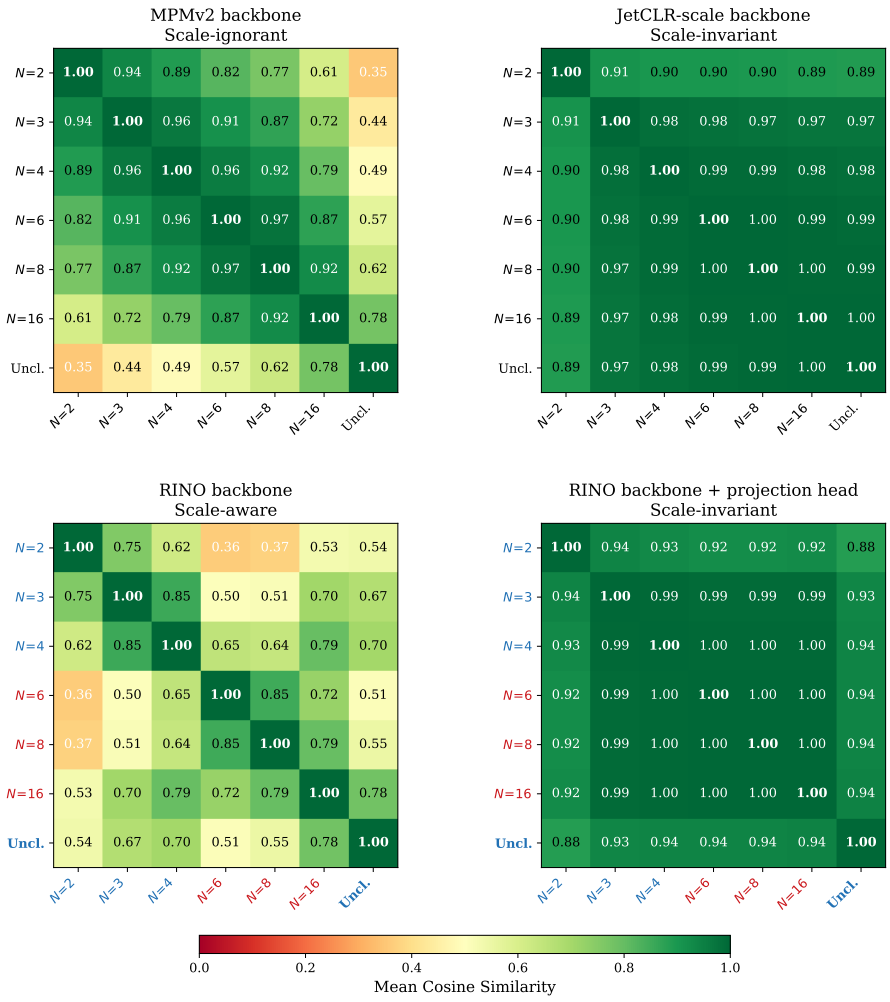

Figure 5: Per-jet mean cosine similarity across k_T clustering scales. Left to right, top to bottom: MPMv2 shows monotonic decay (input-similarity pattern); JetCLR-scale converges to near-invariance; RINO backbone retains scale-dependent geometry organized along the student/teacher role assignment; RINO backbone + projection head realizes cross-scale invariance in the 2048-dimensional head output, including the unclustered view.

component achieves on its own. $\mathcal{L}_{\text{iBOT}}$ thus acts as an auxiliary co-training signal that sharpens representation quality when paired with $\mathcal{L}_{\text{DINO}}$.

PE interaction with views. Polar-binned PE hurts on mixed views (0.730 vs. 0.776 without PE) but helps on cluster views (0.839 vs. 0.828 with p_T-rank PE). This interaction is specific to the view assignment: polar-binned PE encodes spatial position within each subjet view, and its benefit appears only when the teacher-student hierarchy is respected. Under the mixed assignment, polar-binned PE degrades accuracy by 5.9%, while under the hierarchical assignment it improves accuracy by 1.3%.

A.9 Scale Structure

A natural question is how models' representations encode the parton-shower hierarchy. Figure 5 compares the per-jet mean cosine similarity matrix between clustering scales for three models.

A scale-ignorant baseline such as MPMv2 shows the pattern produced by input similarity alone, where, for a given clustering scale, cosine similarity decreases monotonically as the other scale moves away along the k_T ladder in either direction, ranging from 0.92–0.97 between adjacent clustered

scales to 0.35 at the most distant pair ($N=2$ vs. the unclustered jet). This monotonic decay reflects the trivial correlation between subjet count and input feature overlap, without any learned cross-scale structure. In contrast, JetCLR-scale produces near-invariant representations across scales, with all cross-scale cosines in $[0.89, 1.00]$.

RINO shows a third, scale-aware pattern that matches neither baseline. The $N=2$ row is particularly informative: $N=2$ is more similar to the unclustered view (0.54) and $N=16$ (0.53) than to the intermediate teacher-target scales $N=6$ (0.36) and $N=8$ (0.37), despite $N=6$ and $N=8$ being closer to $N=2$ in subjet count. This non-monotonic structure tracks the student-teacher role assignment instead of the trivial input-similarity axis. Adding the projection head makes RINO nearly scale-invariant, with all cross-scale cosines ≥ 0.88 in the head output, including the unclustered view (Fig. 5, bottom right).

Swapped teacher-student roles. To test whether this non-monotonic backbone geometry is causally shaped by the teacher-student role assignment or is a spurious artifact of representation quality, we train a variant with swapped roles: teacher $\{2, 3, 4\}$, with the student additionally seeing $\{6, 8, 16, \text{uncl.}\}$. Figure 6 compares the `[CLS]` cosine similarity matrices side by side.

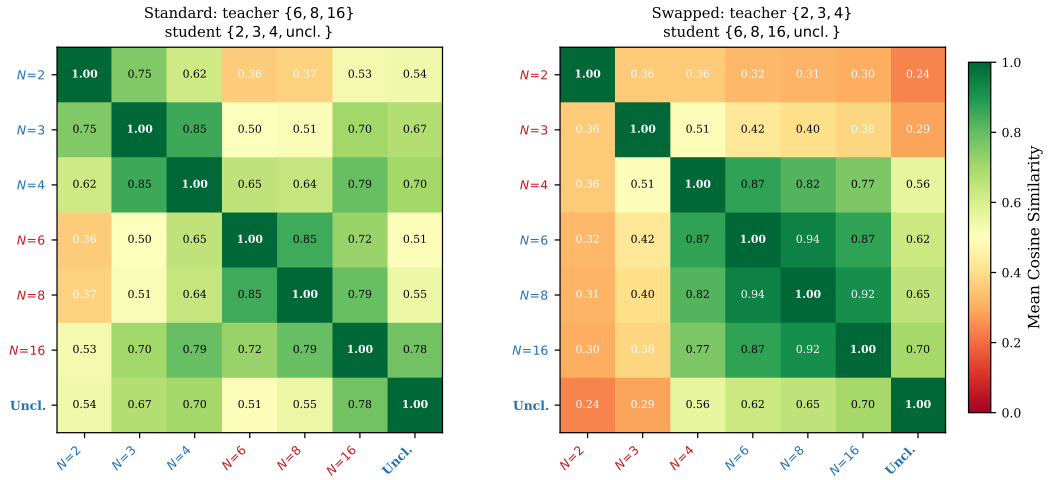

Figure 6: `[CLS]` cosine similarity across k_T scales for the standard (left) and swapped (right) teacher-student assignments. Red = teacher scales, blue = student scales. The grouping of $N=4$ changes with its role: as a student-only scale (left) it is closest to $N=3$ (cosine 0.85, vs. 0.65 to $N=6$); as a teacher scale (right) it is closest to $N=6$ (cosine 0.87, vs. 0.51 to $N=3$).

Under the standard assignment, the student-only scale $N=4$ is closest to $N=3$ (0.85), with cosines 0.65, 0.64, and 0.79 to the teacher scales $N=6$, 8, and 16. Under the swapped assignment, where $N=4$ is a teacher scale, it is closest to $N=6$, 8, and 16 (0.87, 0.82, 0.77) and separates from $N=3$ (0.51). The scales $N=6$, 8, 16 remain mutually similar under both assignments (0.72–0.85 standard, 0.87–0.94 swapped). The grouping of $N=4$ thus follows its role, showing that the backbone geometry is shaped by the role assignment.

A.10 Cross-Topology Transfer with MLP Head

Table 13 reports the MLP head variant of the cross-topology evaluation, including the finetuning topology $t \to bqq$. RINO leads on $t \to bqq$. On $H \to 4q$, RINO and JetCLR-scale are statistically tied (0.828 ± 0.006 vs. 0.824 ± 0.020), as are RINO and MPMv2 on $H \to b\bar{b}$ (0.697 ± 0.009 vs. 0.692 ± 0.029). JetCLR-scale shows a large gap between MLP and linear heads ($H \to 4q$: 0.824 vs. 0.630), while RINO's are nearly identical (0.828 vs. 0.821), indicating that self-distillation produces linearly accessible class boundaries.

Table 13: Cross-topology transfer ACC (JETNET→JETCLASS, MLP head).

Method	$t \to bqq$	$H \to 4q$	$H \to b\bar{b}$
Supervised	0.786 ± 0.023	0.699 ± 0.038	0.623 ± 0.028
SSL baselines			
JetCLR	0.700 ± 0.031	0.696 ± 0.028	0.686 ± 0.015
MPMv1	0.800 ± 0.019	0.769 ± 0.006	0.674 ± 0.017
MPMv2	0.760 ± 0.016	0.780 ± 0.025	0.692 ± 0.029
OmniJet-α	0.751 ± 0.050	0.684 ± 0.028	0.600 ± 0.017
k_{T}-view SSL (ours)			
JetCLR-scale	0.780 ± 0.023	0.824 ± 0.020	0.639 ± 0.014
RINO	$\mathbf{0.847 \pm 0.003}$	$\mathbf{0.828 \pm 0.006}$	$\mathbf{0.697 \pm 0.009}$

A.11 TOP QUARK TAGGING→JETCLASS TRANSFER RESULTS

Table 14 reports top-tagging accuracy on the JETCLASS test set for models finetuned on the TOP QUARK TAGGING dataset (Kasieczka et al., 2019a;b). This is a transfer between datasets with a smaller domain gap than JETNET→JETCLASS, not the in-distribution TOP QUARK TAGGING benchmark, so the values are not comparable to results on the standard TOP QUARK TAGGING test set. The leading methods converge to ~ 0.93 accuracy, indicating that this benchmark is less discriminating than the JETNET→JETCLASS setting used for the main comparison. Nevertheless, scale factors (App. A.12), which directly quantify how strongly a classifier depends on simulation-specific structure, separate the methods, with RINO's closest to unity.

Table 14: TOP QUARK TAGGING→JETCLASS transfer: top-tagging accuracy ($\mu \pm \sigma$ over 6 seeds) on the JETCLASS test set of models finetuned on TOP QUARK TAGGING. These are transfer results, not comparable to in-distribution results on the standard TOP QUARK TAGGING test set. The leading methods saturate near 0.93. Bold marks the best per column; underline marks entries within 1σ of the best.

Method	**MLP head**	**Linear head**
Supervised	0.929 ± 0.001	0.929 ± 0.001
SSL baselines		
JetCLR (Dillon et al., 2022)	0.921 ± 0.001	0.921 ± 0.001
MPMv1 (Golling et al., 2024)	0.929 ± 0.002	0.927 ± 0.001
MPMv2 (Leigh et al., 2025)	0.932 ± 0.001	$\underline{0.933 \pm 0.001}$
OmniJet-α (Birk et al., 2024)	0.760 ± 0.009	0.756 ± 0.008
k_{T}-view SSL (ours)		
JetCLR-scale	$\mathbf{0.936 \pm 0.003}$	$\mathbf{0.935 \pm 0.003}$
RINO	$\underline{0.933 \pm 0.001}$	$\underline{0.933 \pm 0.001}$

Table 15 extends this to cross-topology transfer: models finetuned on TOP QUARK TAGGING for top tagging are evaluated on the unseen $H \to 4q$ signal from JETCLASS. When finetuned on TOP QUARK TAGGING (smaller domain gap with overlapping p_{T} range), cross-topology transfer is much more uniform: the four methods evaluated span $H \to 4q$ AUC 0.877–0.900, narrower than the 0.803–0.894 of the same four methods on JETNET→JETCLASS (Table 2), and RINO (0.896 ± 0.002) is within one standard deviation of the leading MPMv2 (0.900 ± 0.005). This contrasts with JETNET→JETCLASS (Table 2), where RINO leads by $+4.1\%$, consistent with its advantage being specific to larger domain shifts.

A.12 SCALE-FACTOR PROXY (TOP QUARK TAGGING→JETCLASS)

Scale factors (SFs) are a standard downstream component of collider analyses: efficiencies measured in data control regions are divided by their simulated values and used to correct simulated yields (App. A.13). They therefore directly quantify how strongly a classifier depends on simulation-specific structure. We additionally use the TOP QUARK TAGGING dataset to construct a proxy experiment

Table 15: TOP QUARK TAGGING→JETCLASS cross-topology AUC (linear head, 6 seeds). With the smaller domain gap, the four methods reach similar AUC.

Method	$t \to bqq$	$H \to 4q$
MPMv2	$\mathbf{0.982 \pm 0.001}$	$\mathbf{0.900 \pm 0.005}$
RINO	0.981 ± 0.000	0.896 ± 0.002
MPMv1	0.980 ± 0.000	0.889 ± 0.001
JetCLR	0.976 ± 0.001	0.877 ± 0.005

that measures them, using TOP QUARK TAGGING as source ("simulation") and JETCLASS as target ("data"), with the TOP QUARK TAGGING-finetuned models of Table 14. We partition the source signal-score cumulative distribution function (CDF) into six disjoint intervals (the last containing the highest-scoring 5% of jets), apply the same score boundaries to the target, and compute the target/source efficiency ratio in each. For each finetuning run, let s be the signal score, with CDF F over the TOP QUARK TAGGING test signal jets and w-quantile q_w:

$$F(s) = P_{\mathrm{src}}(s' \le s \mid \text{signal}), \qquad q_w = F^{-1}(w), \tag{7}$$

with $q_0 = -\infty$ and $q_1 = +\infty$. An interval $[a, b)$ thus selects jets with $q_a \le s < q_b$, *i.e.*, source signal efficiencies $1 - F(s)$ between $1 - b$ and $1 - a$; the last interval, $[0.95, 1]$, is the tightest working point ($1 - F(s) \le 0.05$). For each interval $[a, b)$, the signal efficiency in domain $d \in \{\mathrm{src}, \mathrm{tgt}\}$ and the scale factor are

$$\varepsilon^{d}_{[a,b)} = P_d(q_a \le s < q_b \mid \text{signal}), \qquad \mathrm{SF}_{[a,b)} = \frac{\varepsilon^{\mathrm{tgt}}_{[a,b)}}{\varepsilon^{\mathrm{src}}_{[a,b)}}, \tag{8}$$

so $\mathrm{SF} = 1$ means that the target populates that interval exactly as the source does. Table 16 lists the scale factors per bin and their mean absolute departure from unity.

Table 16: TOP QUARK TAGGING→JETCLASS scale factors (Eq. 8) per interval of the source signal-score CDF (the last containing the highest-scoring 5% of jets) and their mean absolute departure from unity (last column). Entries are mean ± standard deviation over finetuning runs; bold marks the value closest to unity per bin.

Method	$[0, 0.3)$	$[0.3, 0.5)$	$[0.5, 0.7)$	$[0.7, 0.9)$	$[0.9, 0.95)$	$[0.95, 1]$	$\langle\lvert\mathrm{SF} - 1\rvert\rangle \downarrow$
Supervised	1.120 ± 0.053	0.947 ± 0.022	0.933 ± 0.023	0.960 ± 0.041	0.958 ± 0.073	$\mathbf{0.972 \pm 0.128}$	0.059
SSL baselines							
JetCLR	1.134 ± 0.032	0.942 ± 0.012	0.903 ± 0.020	0.898 ± 0.023	0.964 ± 0.029	1.256 ± 0.084	0.114
MPMv1	1.162 ± 0.047	$\mathbf{0.966 \pm 0.013}$	0.892 ± 0.014	0.884 ± 0.030	0.947 ± 0.043	1.117 ± 0.083	0.098
MPMv2	0.928 ± 0.024	0.867 ± 0.021	0.898 ± 0.008	1.019 ± 0.022	1.278 ± 0.048	1.999 ± 0.112	0.267
OmniJet-α	1.073 ± 0.029	0.938 ± 0.020	0.949 ± 0.031	0.975 ± 0.026	$\mathbf{1.001 \pm 0.046}$	1.110 ± 0.159	0.054
k_{T}-view SSL (ours)							
JetCLR-scale	$\mathbf{0.980 \pm 0.051}$	0.871 ± 0.030	0.884 ± 0.020	0.980 ± 0.028	1.186 ± 0.086	1.967 ± 0.282	0.240
RINO	1.029 ± 0.057	0.963 ± 0.027	$\mathbf{0.978 \pm 0.018}$	$\mathbf{1.001 \pm 0.059}$	1.031 ± 0.084	1.029 ± 0.115	**0.025**

RINO's 2.5% average deformation is less than half the next-lowest value (5.4%), indicating substantially less dependence on simulation-specific structure.

A.13 PATH TO EXPERIMENTAL DATA

Figure 7 shows the intended use of RINO in an experimental analysis, in three stages.

1. Pretrain on collision data. RINO can learn jet representations from unlabeled experimental data, since its pretraining uses no labels. Nevertheless, in this work, for more straightforward evaluations, pretraining uses simulated JETCLASS QCD jets to mimic the QCD-dominated data. Including a fraction of simulated jets and also specialized control regions to isolate more signal-like topologies for the pretraining are both interesting areas for future work.

2. Finetune on labeled simulations. The pretrained backbone is finetuned on simulated signal and background jets, where labels are available, as in our JETNET and TOP QUARK TAGGING experiments.

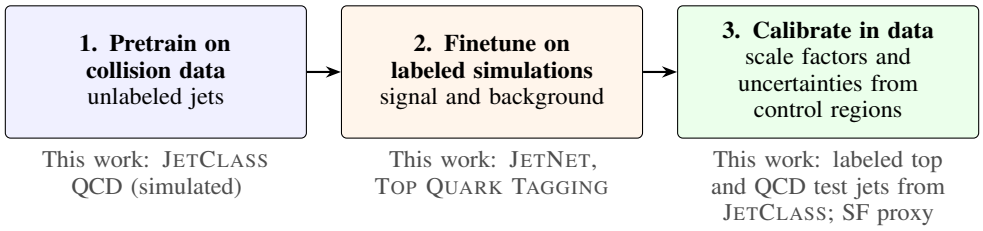

Figure 7: Intended deployment of RINO. Labels enter only in stage 2, and stage 3 is the standard calibration used for supervised taggers. The bottom row lists the simulated stand-ins used in this work, where JETCLASS top jets and all JETCLASS labels enter only in stage 3, for evaluation.

3. Calibrate in data. Signal efficiencies (CMS Collaboration, 2025b) and background mistag rates are measured in dedicated control regions. For top tagging, semileptonic-$t\bar{t}$ tag-and-probe can measure signal efficiency, while QCD-enriched multijet regions constrain mistag rates. The resulting data-to-simulation scale factors (SFs) correct the predicted yields; detector, shower, and hadronization variations enter the final likelihood as systematic uncertainties. Appendix A.12 gives example SFs measured in a simulation-to-simulation proxy.

A.14 FULL DATA-EFFICIENCY CURVES

Figure 8 shows the complete data-efficiency curves for all seven methods evaluated in Sec. 4.3. This includes the JetCLR-scale variant, which isolates the effect of replacing JetCLR's generic augmentations with k_T scale views within the same contrastive objective family. OmniJet-α is omitted from the main-text figure for clarity; its behavior is qualitatively similar to most other baselines (non-monotonic, high variance).

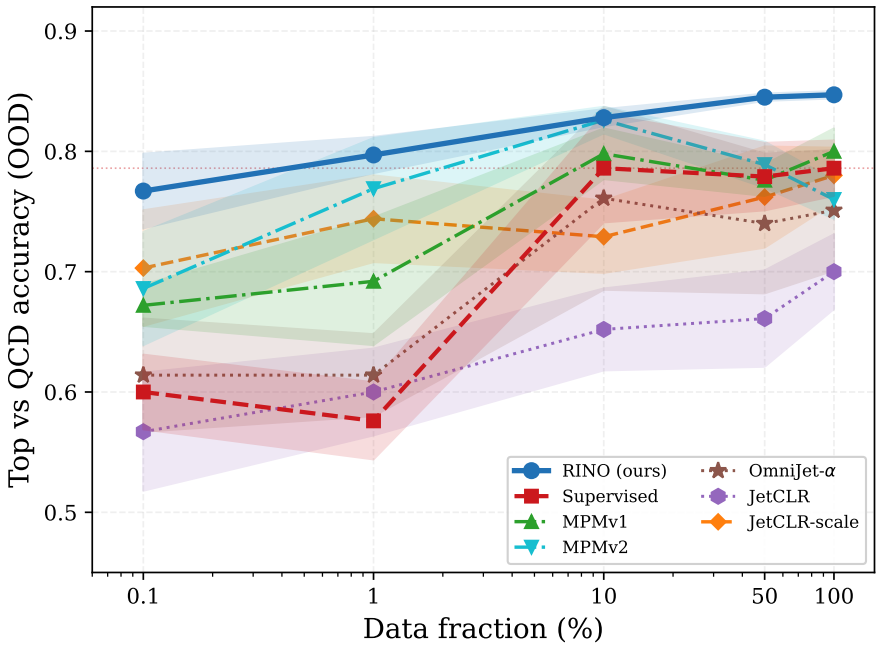

Figure 8: Full data-efficiency curves (all seven methods) with $\pm 1\sigma$ bands over 10 seeds.

Non-monotonic behavior under distribution shift. Several baselines exhibit non-monotonic data-efficiency curves: MPMv2 peaks at 10% and declines at 50–100%. The mechanism is discussed in Sec. 4.3. To rule out a batch-size artifact, we re-ran MPMv2 at 50% with batch size 256 (vs. the default 1024): accuracy changed from 0.789 to 0.761 ± 0.027, indicating that the non-monotonicity

is not simply a batch-size effect. We would like to emphasize that the supervised baseline gains no JETCLASS accuracy beyond 10% of the JETNET labels (0.786, 0.779, and 0.786 at 10%, 50%, and 100%), which is not classical overfitting: its held-out in-distribution JETNET (source) accuracy stays flat or rises with the label fraction. The model is thus not memorizing its training set but specializing to simulation-specific features that do not transfer well. RINO is designed to prevent this: physically, k_{T} coarsening absorbs low-scale radiation into harder subjets (Salam, 2010), so requiring agreement across scales discourages reliance on soft, low-scale details, precisely where parton-shower and hadronization modeling is approximate and generator-dependent (Nagy & Soper, 2018; Barnard et al., 2017). RINO improves monotonically across all label fractions and leads at every fraction; the only other monotonic curve, JetCLR, stays far lower (0.700 ± 0.031 at 100%). This is consistent with RINO's $\sim 3\times$ smaller domain shift in representation space (Table 4).

A.15 MLP VS. LINEAR HEAD DIAGNOSTIC

Table 17 compares MLP and linear head performance on the JETNET→JETCLASS OOD benchmark. The significance column reports the Welch t-statistic $t = (\mu_{\mathrm{MLP}} - \mu_{\mathrm{Linear}})/\sqrt{\sigma^2_{\mathrm{MLP}}/n + \sigma^2_{\mathrm{Linear}}/n}$ with $n = 10$ seeds per condition. Degrees of freedom follow the Welch–Satterthwaite approximation (≈ 11 for RINO, whose two variances differ, and $\approx$ 12–17 for the other methods). RINO is the only method with a significantly positive t ($+3.0$, two-tailed $p \approx 0.01$), indicating exploitable nonlinear structure; JetCLR and MPMv2 show clear degradation ($t < -3$, $p < 10^{-3}$).

Table 17: MLP vs. linear head on OOD top tagging (JETNET→JETCLASS).

Method	**MLP**	**Linear**	t
RINO	$\mathbf{0.847 \pm 0.003}$	$\mathbf{0.838 \pm 0.009}$	$\mathbf{+3.0}$
Supervised	0.786 ± 0.023	0.784 ± 0.033	$+0.2$
MPMv1	0.800 ± 0.019	0.809 ± 0.008	-1.4
MPMv2	0.760 ± 0.016	0.817 ± 0.027	-5.7
JetCLR	0.700 ± 0.031	0.791 ± 0.048	-5.0
JetCLR-scale	0.780 ± 0.023	0.803 ± 0.017	-2.5
OmniJet-α	0.751 ± 0.050	0.749 ± 0.080	$+0.1$

A.16 DOMAIN SHIFT IN REPRESENTATION SPACE

Protocol. Both metrics are computed on $5{,}000$ jets per domain. W_1 is averaged over embedding dimensions; the squared maximum mean discrepancy (MMD^2, biased estimator) uses an RBF kernel $\exp(-\gamma\|x - y\|^2)$ with fixed $\gamma = 1/d$, where d is the embedding dimension.

Table 18 reports the same metrics as Table 4 on the pretrained backbones, before finetuning, with signed changes indicating whether finetuning increases ($+$) or decreases ($-$) the gap. RINO already has the smallest pretrained shift on three of the four metrics (*e.g.*, W_1 (QCD): 0.055, vs. at least 0.092 for the baselines). JetCLR-scale, which shares RINO's k_{T} views, is the other method with a small pretrained shift: the two k_{T}-view methods have the two smallest values on *every* metric. JetCLR-scale's shift increases during finetuning but stays relatively low (Table 4), while RINO's decreases further on all four metrics (*e.g.*, W_1 (QCD): $0.055 \to 0.043$). MPMv2 is the only other method whose gap decreases on all four. The remaining baselines do not consistently benefit from finetuning: the changes are mixed for JetCLR and MPMv1.

These trends connect to the feature-distortion picture of Kumar et al. (2022) (Sec. 4.3): finetuning updates the features of in-distribution (JETNET) jets more than those of OOD (JETCLASS) jets, which can widen the gap between the two domains. For RINO, whose pretrained representations are already close across domains, finetuning instead brings them even closer together. This suggests that the small pretrained shift comes from the k_{T} views, and provides further evidence for the physical picture of App. A.14: k_{T} coarsening absorbs low-scale radiation into harder subjets (Salam, 2010), so requiring agreement across scales discourages reliance on soft, low-scale details, where parton-shower and hadronization modeling is approximate and generator-dependent (Nagy & Soper, 2018; Barnard et al., 2017).

Table 18: Domain shift in *pretrained* representations (JETNET vs. JETCLASS), before finetuning. Colored values give the change *after* finetuning (Table 4): − (green) = finetuning reduces the gap, + (red) = finetuning increases it.

Method	W_1 (QCD) ↓	W_1 (top) ↓	MMD^2 (QCD) ↓	MMD^2 (top) ↓
JetCLR	0.128 +.004	0.125 +.002	0.223 −.027	0.212 −.021
OmniJet-α	0.181 +.634	0.246 +.506	0.095 +.476	0.193 +.359
MPMv1	0.232 +.009	0.274 −.012	0.120 +.013	0.183 −.021
MPMv2	0.409 −.034	0.374 −.026	0.427 −.089	0.409 −.104
JetCLR-scale	0.092 +.041	0.071 +.074	0.017 +.028	**0.009** +.039
RINO	**0.055** −.012	**0.052** −.008	**0.012** −.005	0.011 −.004

A.17 BASELINE IMPLEMENTATION DETAILS

All SSL baselines are implemented under a controlled implementation and finetuning setup so that the main variables across methods are the pretraining objective and view-construction strategy. This setup reduces confounders from depth, width, normalization, and activation choices that would otherwise conflate model capacity with pretraining quality. Table 19 summarizes the key implementation choices. We do not include J-JEPA (Katel et al., 2024) or HEP-JEPA (Bardhan et al., 2025) in our main comparison: J-JEPA reports modest gains over training from scratch, and HEP-JEPA's code has not been released, precluding direct reproduction in our controlled-backbone protocol. Among existing methods, J-JEPA is the closest conceptual relative of RINO: it reclusters each jet into C/A subjets and predicts the embeddings of target subjets from context subjets within a single jet, but does not align representations *across* different coarse-graining scales. Our ablation (Sec. 4.5) shows that cross-scale k_T views account for the majority of the gain ($+11.4\%$) regardless of the SSL family, suggesting that the multi-scale view construction dominates the choice of joint-embedding objective.

Table 19: Baseline implementation summary. All methods share the same 7 kinematic input features (Table 6), the same QCD pretraining data, and the same finetuning recipe (App. A.6).

Method	**Backbone**	**Attention**	**Input**	**Loss**
RINO	JTE enc.	Bidir.	Continuous	DINO+iBOT
Supervised	JTE enc.	Bidir.	Continuous	Cross-ent.
MPMv1	JTE enc.	Bidir.	Continuous	Masked VQ pred.
MPMv2	JTE enc.	Bidir.	Continuous	Masked K-means + L1 + flow
JetCLR	JTE enc.	Bidir.	Continuous	NT-Xent
JetCLR-scale	JTE enc.	Bidir.	Continuous	NT-Xent+MAE
OmniJet-α	JTD dec.	Causal	VQ tokens	Next-token pred.

Shared components. The jet transformer encoder (JTE) backbone is identical across all encoder-based methods: 8 layers, $d_{\text{model}} = 256$, 16 heads, SwiGLU FFN, RMSNorm, LayerScale, and 4 register tokens. MPMv1 and OmniJet-α share a VQ-VAE tokenizer with a 512-code codebook, trained once on the QCD pretraining set and frozen during SSL pretraining. All methods are pretrained on the same QCD-only data.

Hyperparameter parity. Each baseline uses the core hyperparameters from its original publication (loss weights, masking ratios, temperature), which we assume are near-optimal for the respective objectives. RINO's teacher temperature and EMA momentum were selected on the OOD top-vs-QCD accuracy on JETCLASS. This step contributes $+1.0\%$ (Table 3), and RINO before tuning (0.839 ± 0.005) already exceeds every baseline's MLP-head accuracy in Table 1 (best: MPMv1, 0.800 ± 0.019). All encoder-based methods share the same backbone, and all methods share the same data and finetuning protocol, so the comparison isolates the pretraining objective and view construction as much as possible.

OmniJet-α. OmniJet-α is the only decoder-based method. The original model (Birk et al., 2024) uses a 3-layer GPT-style decoder with 8 heads and an 8,194-token vocabulary. Our adaptation replaces

this with an 8-layer jet transformer decoder (JTD) that mirrors the JTE in depth, width, normalization, and activation, differing only in the causal attention mask required by the autoregressive objective. This controls for model capacity when comparing the decoder baseline to encoder-based methods. We use the shared 512-code VQ-VAE tokenizer rather than the original vocabulary. Because causal attention imposes an ordering on an inherently unordered set (jet constituents have no natural sequence), the decoder can only aggregate information left-to-right. We use mean pooling over all token outputs as the jet representation, which empirically outperforms the `last_token` alternative.

MPMv1 and MPMv2. Both follow the masked-particle modeling framework of Golling et al. (2024) and Leigh et al. (2025), adapted to use the shared JTE backbone. The backbone receives *continuous* input features (the same 7 kinematic features as RINO), projected through a linear embedding layer. MPMv1 uses the VQ-VAE only as a *target generator* for a cross-entropy classification loss at masked positions. MPMv2 uses no VQ-VAE: at masked positions, it predicts each constituent's cluster in the 16,384-centroid K-means codebook released with Leigh et al. (2025), regresses the continuous features directly (L1 loss), and models them with a conditional normalizing flow. Mean pooling is used because MPM's per-particle masked-prediction objective provides no training signal for a `[CLS]`.

JetCLR and JetCLR-scale. JetCLR (Dillon et al., 2022) uses the NT-Xent contrastive loss with standard jet augmentations (rotations, translations, p_{T}-dependent smearing, and collinear splittings). JetCLR-scale isolates the contribution of multi-scale view construction: it replaces JetCLR's generic augmentations with the same k_{T} clustering views used by RINO, while keeping a contrastive objective. Scales are grouped into three tiers: coarse ($N \in \{2, 3, 4\}$), intermediate ($N \in \{6, 8\}$), and fine ($N = 16$ or unclustered). Each training batch samples one scale from each of two randomly chosen tiers, producing a two-view pair with a guaranteed cross-scale gap (*e.g.*, $N = 3$ vs. $N = 16$). The NT-Xent loss encourages agreement across these paired scales, and an MAE reconstruction auxiliary at the fine scale stabilizes low-data training. The JetCLR-to-JetCLR-scale comparison (0.700 vs. 0.780) confirms that the k_{T} views account for most of the gain.

A.18 Compute

Pretraining a single RINO model on the QCD subset of JetClass ($\sim 10^7$ jets, 100 epochs) takes approximately 45 minutes per epoch on 2× NVIDIA A100-80GB, for a wall-clock cost of ∼80 hours per pretraining run. Finetuning on JetNet takes ∼2 minutes per epoch on a single NVIDIA 3090 or A10, for at most 55 epochs with early stopping; runs typically converge within ∼25 epochs (∼50 minutes wall-clock per seed). Finetuning on the Top Quark Tagging dataset takes ∼15 minutes per epoch on 2× 3090/A10, again capped at 55 epochs with early stopping (∼25 epochs and ∼6 wall-clock hours per seed in practice).

Table 20 compares per-method pretraining wall-clock cost on the same QCD-only JetClass subset. All encoder-based methods use the shared JTE backbone (OmniJet-α uses the matched JTD). MPMv1 and OmniJet-α additionally require a VQ-VAE tokenizer (∼8h, trained once and shared).

Table 20: Per-method pretraining wall-clock cost (QCD-only JetClass subset, estimated for 100 epochs). † Requires a shared VQ-VAE tokenizer (∼8h additional, trained once).

Method	Time/epoch (min)	Total (h)
JetCLR	∼10	∼20
JetCLR-scale	∼25	∼45
MPMv1†	∼30	∼50
MPMv2	∼40	∼65
OmniJet-α†	∼20	∼30
iBOT-only	∼50	∼80
RINO	∼45	∼80