Embedding Jets with Maximum Manifold Capacity Representations

Samuel Bright-Thonney[1,2]
[1]Department of Physics, Massachusetts Institute of Technology
[2]The NSF AI Institute for Artificial Intelligence and Fundamental Interactions

Abstract

Self-supervised learning (SSL) has emerged as the dominant paradigm for training particle physics foundation models. Existing methods largely borrow from language modeling (e.g. masked-/next-token prediction) or computer vision (e.g. multi-view similarity objectives). In this work, we explore an alternative technique based on *manifold capacity* theory: a neuroscience-inspired method to quantify the linear separability of manifolds (point clouds) using their intrinsic geometric features. We apply the recently developed Maximum Manifold Capacity Representations (MMCR) technique to learn representations of simulated particle jets from hadron colliders, finding that MMCR matches or slightly surpasses similarity-based objectives (SimCLR) as measured by linear class separability of the learned embeddings. These results position MMCR as a promising approach for representation learning in particle physics, and motivate further study in more complex settings.

1 Introduction

In the era of large-scale machine learning (ML), foundation models are a compelling prospect for analyzing large, complex datasets in the physical sciences. High-energy physics (HEP) stands to benefit enormously from a powerful foundation model, with experiments at the Large Hadron Collider measuring the results of 40 MHz proton-proton collisions in tens-of-millions heterogeneous detector readout channels. A model that efficiently encodes the underlying structure of this data in a lower-dimensional feature space would be a boon to the field, accelerating downstream analysis and potentially discovery. Recent work on HEP foundation models [1–5] has focused on *jets*: collimated sprays of particles resulting from the fragmentation of high-energy quarks and gluons. Most approaches rely on *self-supervised learning* (SSL) techniques, with training objectives borrowed from language modeling (next-token/masked-token prediction) [1, 2] or computer vision (aligning representations of multiple views of the same image) [5].

In this work, we explore a recently proposed SSL strategy for encoding jets based on *manifold capacity theory* [6, 7]. We train encoders for jet data using the *maximum manifold capacity representations* (MMCR) framework [7], a SSL technique that encourages networks to learn compact representations using an objective related to the geometry of "augmentation manifolds" constructed from many views of a given input sample. We evaluate learned representations using linear and shallow nonlinear classifier probes, and compare them to those obtained from SimCLR [8], a popular existing SSL method that has previously been applied to jets [5].

2 Methods

Manifold Capacity Theory Manifold capacity was introduced in [6], and is motivated by *invariant object recognition* in neuroscience: the brain's ability to quickly distinguish object classes (e.g. cats

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

vs. dogs) under many different "views" (viewing angle, breed, etc.). It is hypothesized that these views form "object manifolds" in neuronal activation space, which our brains then process into representations that can be efficiently distinguished; i.e. *linearly separable*. Manifold capacity theory analyzes the linear separability of a collection of manifolds embedded in an ambient space $\mathbb{R}^N$, generalizing the theory of *perceptron capacity* for separability of points [9–11]. Roughly speaking, manifold capacity $\alpha \equiv P/N$ corresponds to the largest number (P) of manifolds that can be embedded in an ambient space of dimension N such that a random dichotomy can, with high probability, be linearly separated by a hyperplane. Capacity naturally depends on the geometry of the manifold in question (e.g. intrinsic dimensionality, spatial extent) and varies accordingly. Ref. [6] developed a statistical mechanical theory for computing α for manifolds represented as finite point clouds[1] in $\mathbb{R}^N$. Closely related to capacity are the manifold *radius* R_M and *dimension* D_M, which measure the spatial extent and dimensionality of the manifold.

MMCR Manifold capacity naturally relates to self-supervised learning: if a neural network can be trained to learn high-capacity representations for the augmentation manifolds of e.g. images, then the learned embeddings may be very useful for downstream tasks. However, computing manifold capacity for a general point cloud is costly and non-differentiable, making it unsuitable for direct optimization. Ref. [7] proposed instead using an approximate formulation of capacity valid for elliptical manifolds: $\alpha \approx \phi(\sum_i \sigma_i)$, where σ_i are the singular values of a manifold point cloud $\mathbf{X} \in \mathbb{R}^{N\times k}$ composed of k vectors in $\mathbb{R}^N$, and ϕ is a monotonically decreasing function [7]. They use this formulation to introduce the **maximum manifold capacity representations** (MMCR) technique, a multi-view self-supervised learning strategy. Given a batch of inputs $\{\mathbf{x}_i\}_{i=1}^B$ and a family of augmentations, MMCR generates k views of each input and feeds them through an encoder network f_θ to obtain d-dimensional representations $\mathbf{z}_i^{(j)}$, $j = 1 \dots k$. These are normalized to the unit sphere $\mathbb{S}^{d-1}$ and the *centroids* $\mathbf{c}_i = \frac{1}{k}\sum_{j=1}^k \mathbf{z}_i^{(j)}$ of each augmentation manifold are arranged into a matrix $\mathbf{C} \in \mathbb{R}^{d\times B}$. The MMCR loss is defined using the nuclear norm $||\cdot||_*$ of this matrix:

$$\mathcal{L}_{\text{MMCR}} = -||\mathbf{C}||_* = -\sum_{i=1}^{\text{rank}(\mathbf{C})} \sigma_i(\mathbf{C}), \tag{1}$$

where σ_i are the singular values of $\mathbf{C}$. The loss is constructed to *maximize* the extent of the "centroid manifold", which implicitly minimizes the extent of the individual augmentation manifolds $\mathbf{Z}_i \in \mathbb{R}^{d\times k}$. This objective is easy to optimize with neural networks and, surprisingly, is sufficient to learn expressive representations despite the approximations involved [7, 12].

SimCLR For our particle physics studies we benchmark MMCR against SimCLR [8], a successful and well-studied SSL technique. Given an encoder f_θ and a batch $\{\mathbf{x}_i\}_{i=1}^B$, *positive pairs* $(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$ are generated as augmented views of the same input, and the SimCLR objective is formulated as:

$$\mathcal{L}_{\text{SimCLR}} = -\sum_{i\in\mathcal{B}} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \tilde{\mathbf{z}}_i)/T)}{\sum_{j\neq i}\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/T)}, \tag{2}$$

where $\mathbf{z} = f_\theta(\mathbf{x})$ are encoder outputs, $\text{sim}(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i \cdot \mathbf{z}_j/||\mathbf{z}_i||||\mathbf{z}_j||$, T is a temperature hyperparameter, and the sum in the denominator runs over all (i, j) in the dual-viewed batch, excluding self-similarity. In addition to the standard augmentation-based SimCLR, we also consider *supervised* SimCLR [13], where positive pairs are constructed from input pairs with the same class label.

3 Experiments

Our experiments follow the general strategy of Refs. [5, 14], training encoders on simulated particle jet data and evaluating learned embeddings via linear and nonlinear classifiers.

Dataset We use the JETCLASS dataset [15, 16], which consists of ten classes of jets from simulated proton-proton collisions. For simplicity we use only five classes: quark/gluon-initiated jets (QCD), W/Z boson jets ($W \to qq'$ and $Z \to q\bar{q}$), hadronic top quark decays ($t \to bqq'$), and Higgs boson decays to b quarks ($H \to b\bar{b}$). Jets are represented as point clouds constructed from their constituent particles, with each carrying kinematic, ID[2], and trajectory information. We use simple 3D kinematic

[1] For point clouds, the continuous manifold corresponds to the *convex hull* of the constituent points.

[2] Particle identification (ID) features are quantities such as charge and particle type.

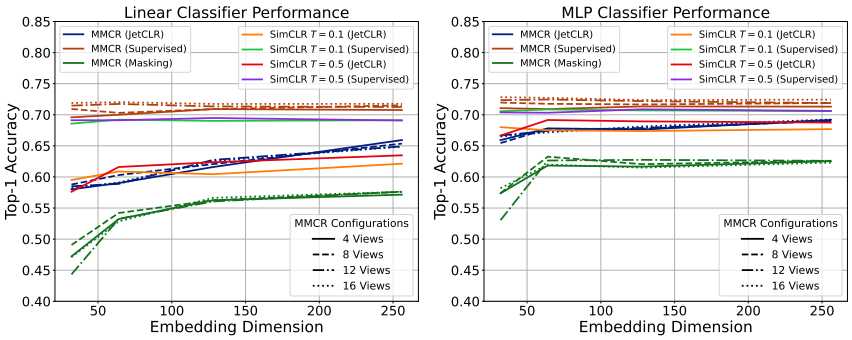

Figure 1: Top-1 accuracy of linear (left) and MLP (right) classifiers as a function of d_{embed} for various MMCR and SimCLR strategies (in legend). MMCR matches or slightly outperforms SimCLR in most cases, with the supervised augmentation strategy performing best overall.

features $(\log p_T, \Delta\eta, \Delta\phi)$ for each particle, where p_T is transverse momentum and $\Delta\eta$ and $\Delta\phi$ are the pseudorapidity and azimuthal angle relative to the jet axis, respectively. We take the 64 highest-p_T particles in each jet and zero-pad those with fewer than 64 constituents.

Encoders To ensure a fair comparison, we train all models with the same transformer encoder backbone [17] with a latent dimension $d_{\text{model}} = 512$, 4 self-attention blocks with 4 heads each, and without tokenization. Self-attention is between all pairs of constituent particles, with masks applied only on padding entries. We obtain the final d_{embed}-dimensional embeddings by averaging all non-padding vectors after the attention layers and applying a shallow MLP. During training we apply an additional *projection head* before computing the loss, as is standard practice in SSL [8, 18, 19]. This head is a simple one-hidden-layer MLP projecting from $d_{\text{embed}} \to d_{\text{embed}}/2$ and is discarded after training, with all downstream evaluation performed on the transformer output.

Augmentations We explore three different augmentation strategies: JetCLR, supervised, and masked. JetCLR [5, 14] generates augmented jets ("views") using three physics-inspired transformations:

- *Rotation*: Constituents are rotated by a random angle $\theta \in [0, 2\pi]$ about the jet axis in the η-ϕ plane.
- *Soft splitting*: Noise sampled from $\mathcal{N}(0, \Lambda/p_{T,i})$ is added to the position (η_i, ϕ_i) of each constituent i with $\Lambda = 100$ MeV, mimicking distortion from detector effects.
- *Collinear splitting*: A randomly chosen 10% of constituent particles are split into pairs with momenta $f_i p_{T,i}$ and $(1 - f_i)p_{T,i}$ and identical (η_i, ϕ_i), where $f_i \sim U(0, 1)$.

For the supervised strategy, views are generated by drawing additional jets with the same class label from the training set. Finally, masked views randomly select and mask 10% of constituent particles according to a probability distribution $p_i \sim 1/p_{T,i}$, inspired by the Masked Particle Modeling (MPM) approach for jet foundation models [1]. We train MMCR encoders with all three augmentation strategies, while for the SimCLR baselines we only use JetCLR and supervised.

Training All models are implemented in PyTorch [20] and trained with the AdamW optimizer [21, 22] with a learning rate of 10^{-4} annealed to 10^{-5} over 100 epochs on a cosine schedule [23]. We use a baseline batch size of 64, with the effective batch size scaling with the number of augmentations per sample used in the MMCR and SimCLR objectives. We train models with $d_{\text{embed}} = 32, 64, 128, 256$, and for MMCR we train variants using $N_{\text{view}} = 4, 8, 12, 16$ augmentations per sample. For SimCLR baselines we use the standard two views for JetCLR, while for supervised the number of positive (same-class) pairs depends on the class-composition of a batch (the loss is still computed pairwise, see [13]). All models are trained on a single A100 GPU.

Evaluation We evaluate our encoders on a common sample of 0.5M jets from the JetClass test set (100k from each class) with a 70/15/15 split. In keeping with standard SSL evaluation practices, we train linear classifiers on embeddings from each model and record top-1 accuracy. We also train shallow MLP classifiers to assess basic non-linear separability. We also train four different *binary* linear classifiers to distinguish QCD jets from W, Z, top, and Higgs jets. This is comparable to

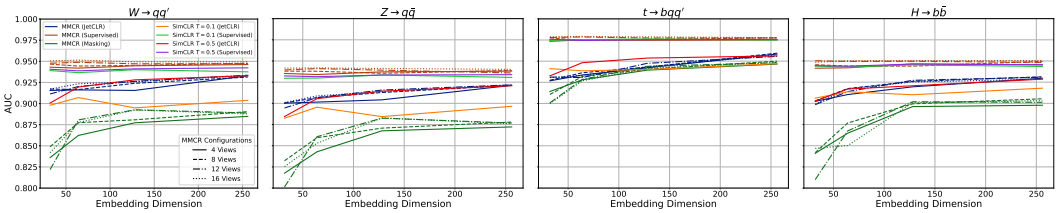

Figure 2: Area under the curve (AUC) classification metrics for linear binary classification of QCD jets versus W, Z, top, and Higgs (from left to right), plotted as a function d_{embed} as in Fig. 1.

the binary tasks considered in past JetCLR work [5, 14], and binary classification against QCD (the dominant background in many collider physics settings) is a standard task in HEP.

4 Results

Figure 1 summarizes the main results of our studies, showing top-1 accuracy for linear (left) and MLP (right) classifiers as a function of d_{embed} for the various MMCR and SimCLR encoders described above. Supervised MMCR and SimCLR models perform best overall on linear evaluation, with MMCR benefitting from larger embedding dimensions and more views per training sample. MMCR appears to slightly outperform SimCLR with large d_{embed} and N_{view}, but the effect is small and may be related to the effective MMCR batch size scaling with the number of views. The differences are smaller still for MLP evaluation, indicating that SimCLR and MMCR achieve similar quality embeddings. MMCR with masking performs the worst, falling significantly short of the JetCLR and supervised variants. Interestingly, the performance gap between supervised and JetCLR shrinks significantly with a nonlinear classifier, suggesting that the physics-inspired JetCLR augmentations are sufficient to learn an expressive – but not *linearly* separable – representation.

Figure 2 reports the area under the curve (AUC) for binary linear classification of W, Z, top, and Higgs jets against QCD. The same conclusions from Fig. 1 hold, with MMCR and SimCLR performing roughly equally. There is a strong difference between classes, however, with top jets being best separated. This is likely due to their three-prong substructure, which is more distinct from QCD's one-prong structure and than the two-prong profiles of W, Z, and $H \to b\bar{b}$. We use only kinematic input features for each particle, omitting particle ID and trajectory information that would help further differentiate the two-prong classes.

To better understand the interplay between SimCLR batch size and effective MMCR batch size, we train compute-matched versions of the models using supervised augmentations. Rather than sampling B elements and generating k views, we instead sample batches with exactly N exemplars from each class. In the MMCR setting, these are treated as N views of each class element, while for SimCLR the batch is treated normally and class labels define the positive pairs. The batch size is then scaled by varying N. In Fig. 3, we compare top-1 accuracy (left) and per-class AUC (middle) for linear/MLP classifiers on MMCR and SimCLR embeddings. We see essentially the same behavior as in Fig. 1, with MMCR and SimCLR performing equally.

Lastly, we compare MMCR and SimCLR in a slightly larger-scale setting by training on all ten JETCLASS classes using a full Particle Transformer architecture with kinematic, particle ID, and trajectory features as described in [16]. Per-class AUCs are plotted in Fig. 3 (right), again showing roughly equivalent performance. MMCR appears to outperform SimCLR for some classes using an MLP classifier, but by a relatively small margin.

5 Conclusion & Outlook

Our studies position MMCR as a viable and promising alternative to SimCLR for self-supervised representation learning in particle physics. We have demonstrated that MMCR embeddings are at least as expressive as those obtained from SimCLR using the same data augmentations, and may improve more with further parameter exploration[3]. MMCR is also *conceptually* distinct: the objective

[3]For example, some of the best-performing image models in [7] used 20-40 augmentations per image.

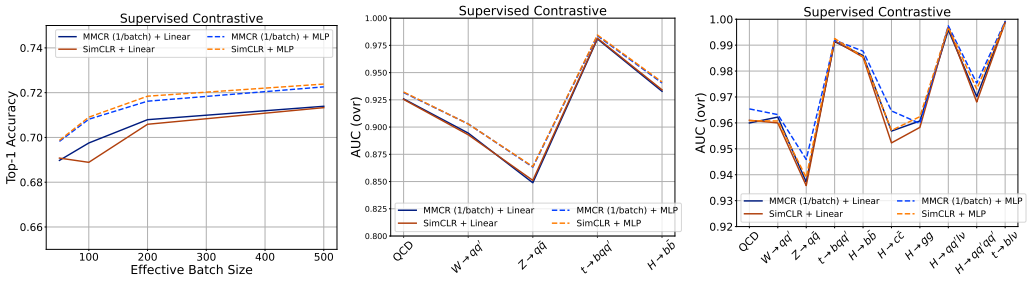

Figure 3: Left: Top-1 accuracy for compute-matched MMCR and SimCLR representations trained with supervised augmentations. Center: Corresponding one-versus-rest AUC for each class. Right: One-versus-rest AUC by class for MMCR and SimCLR models using a ParT [16] encoder with kinematic, PID, and trajectory particle inputs and trained on all classes in the JETCLASS dataset.

is derived from the geometry of multi-view augmentation manifolds of training examples, whereas SimCLR simply promotes alignment between views. This may prove beneficial for complex and highly-structured data such as jets.

These results also lay a strong foundation for further exploration. We intend to perform a comprehensive MMCR parameter-space exploration to better understand the interplay of d_{embed} and N_{view}, with an equivalent scan for SimCLR baselines (batch size, temperature). We will also further explore larger-scale trainings with full particle features (e.g. particle ID, trajectory), larger encoders, and all of the JETCLASS classes, which will improve performance and expressivity. Lastly, we hope to explore *heterogeneous* augmentations, mixing physics inspired augmentations with other well-motivated transformations (e.g. detector distortions, or systematic variations of underlying simulation parameters as in [24]).

References

[1] 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(3):035074, 2024. doi: 10.1088/2632-2153/ad64a8.

[2] Joschka Birk, Anna Hallin, and Gregor Kasieczka. OmniJet-α: the first cross-task foundation model for particle physics. *Mach. Learn. Sci. Tech.*, 5(3):035031, 2024. doi: 10.1088/2632-2153/ad66ad.

[3] Vinicius Mikuni and Benjamin Nachman. Solving key challenges in collider physics with foundation models. *Phys. Rev. D*, 111(5):L051504, 2025. doi: 10.1103/PhysRevD.111.L051504.

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

[5] Barry M. Dillon, Gregor Kasieczka, Hans Olischlager, Tilman Plehn, Peter Sorrenson, and Lorenz Vogel. Symmetries, safety, and self-supervision. *SciPost Phys.*, 12(6):188, 2022. doi: 10.21468/SciPostPhys.12.6.188.

[6] SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003, 2018.

[7] Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. Learning efficient coding of natural images with maximum manifold capacity representations. *Advances in Neural Information Processing Systems*, 36:24103–24128, 2023.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020.

[9] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 2006.

[10] Elizabeth Gardner. Maximum storage capacity in neural networks. *Europhysics letters*, 4(4): 481, 1987.

[11] Elizabeth Gardner. The space of interactions in neural network models. *Journal of physics A: Mathematical and general*, 21(1):257, 1988.

[12] Rylan Schaeffer, Victor Lecomte, Dhruv Bhandarkar Pai, Andres Carranza, Berivan Isik, Alyssa Unell, Mikail Khona, Thomas Yerxa, Yann LeCun, SueYeon Chung, et al. Towards an improved understanding and utilization of maximum manifold capacity representations. *arXiv preprint arXiv:2406.09366*, 2024.

[13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[14] Barry M. Dillon, Radha Mastandrea, and Benjamin Nachman. Self-supervised anomaly detection for new physics. *Phys. Rev. D*, 106(5):056005, 2022. doi: 10.1103/PhysRevD.106.056005.

[15] Huilin Qu, Congqiao Li, and Sitian Qian. Jetclass: A large-scale dataset for deep learning in jet physics, June 2022. URL `https://doi.org/10.5281/zenodo.6619768`.

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

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[19] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[24] Philip Harris, Michael Kagan, Jeffrey Krupa, Benedikt Maier, and Nathaniel Woodward. Re-simulation-based self-supervised learning for pre-training foundation models. *arXiv preprint arXiv:2403.07066*, 2024.