

OMNICOSMOS: Transferring Particle Physics Knowledge Across the Cosmos

Vinicius Mikuni^{1,*}, Ibrahim Elsharkawy^{2,3,†} and Benjamin Nachman^{4,5,‡}

¹*Kobayashi-Maskawa Institute, Nagoya University, Aichi 464-8602, Japan*

²*National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

³*Department of Physics, University of Toronto and Vector Institute, Toronto, Ontario M5S 1A1, Canada*

⁴*Department of Particle Physics and Astrophysics, Stanford University, Stanford, California 94305, USA*

⁵*Fundamental Physics Directorate, SLAC National Accelerator Laboratory, Menlo Park, California 94025, USA*



(Received 3 May 2026; accepted 24 August 2026; published 15 October 2026)

Foundation models build an effective representation of data that can be deployed on diverse downstream tasks. Previous research showed that the OMNILEARNED, a foundation model for collider physics, could significantly advance discovery potential across collider experiments. In this paper, we show that foundation models trained on collider data can transfer to cosmology. Specifically, we improve the prediction of cosmological parameters and halo and galaxy velocities in different datasets from COSMOBENCH. To the best of our knowledge, this is the first time a collider physics model has been shown to generalize across scientific fields.

DOI: [10.1103/85gr-9vw6](https://doi.org/10.1103/85gr-9vw6)

I. INTRODUCTION

Current and near-term cosmological surveys are revolutionizing our understanding of dark matter and dark energy. So far, cosmological insight from survey data has been obtained by first compressing the data and then comparing the low-dimensional summaries to theoretical predictions. Modern machine learning has catalyzed a movement to go beyond these summaries to perform *field-level analysis* in order to use all of the available information [1]. While these approaches have the potential to significantly improve the precision of survey data, progress in this direction requires two currently incomplete components: (1) accurate simulations describing all relevant length scales and timescales and (2) machine learning methods that can accommodate the structure of cosmological data and simulations. While it may be that machine learning will help address (1), especially beyond the N -body evolution of dark matter, we focus on the second challenge. In particular, cosmological data are composed of variable-length sets of objects (e.g., galaxies) in space (called a *point cloud* in machine learning), and simulating entire universes is so computationally demanding that inference must proceed with only a relatively small number of synthetic universes.

Industry-driven neural network architectures have focused on regular grids (for images) and sequences (for text). One can always voxelize the universe onto a regular grid or sequence

to make direct use of such approaches. Instead, we employ an architecture that is designed for point clouds—It can process variable-length inputs and is invariant under permutations of the inputs (set versus sequence). State-of-the-art methods of this kind are based on transformers [2]. While highly effective, transformers also require a lot of data to leverage the capacity of large models. This is a significant challenge for cosmological inference, where large sets of synthetic universes are unavailable.

We propose to address these challenges by building on a point-cloud foundation model designed for particle collider data. Like cosmology data, collider data are also variable-length sets. Due to quantum mechanics, each collision produces a stochastic number of outgoing particles, each defined by a three-momentum along with other attributes. Fully simulating a particle collision is computationally expensive, but much less so than simulating an entire universe. There are also multiple levels of simulation fidelity. Our hypothesis is that a sufficiently expressive foundation model for collider physics that is fine-tuned on a relatively small number of cosmological simulations will outperform classical approaches as well as machine learning models trained from scratch on the cosmological simulations. While the physical laws most relevant for cosmology and those most relevant for particle collisions are not the same, the datasets share enough commonalities to justify this transfer learning objective.

To test this hypothesis, we use the OMNILEARNED [3–5] foundation model and the COSMOBENCH [6] datasets. OMNILEARNED is a point edge transformer (PET) trained using supervised and unsupervised learning on one billion labeled and unlabeled hadronic jets from various real and synthetic particle collisions. OMNILEARNED has been shown to effectively transfer between low- and high-fidelity simulations, between simulations and real data, and between different collision systems. In this paper, we demonstrate that the same model can be useful for fundamental physics data

*Contact author: vmikuni@hepl.phys.nagoya-u.ac.jp

†Contact author: ibrahim.elsharkawy@mail.utoronto.ca

‡Contact author: nachman@stanford.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

that are unrelated to particle jets. The COSMOBENCH dataset is a set of simulated point clouds for benchmarking tools for inferring parameters in the standard model of cosmology (Λ CDM). In particular, three different simulators are used to build coarse-grained [7] universe simulations. Points represent (effective) dark matter halos or galaxies. Merger histories are present in one of the datasets, but we focus on the present-time tasks [8]. Even though gravity is deterministic, the cosmology simulations are stochastic like the collider simulations because of the initial conditions of the universe. In addition to data, COSMOBENCH also includes classical and machine learning benchmark algorithms that we use as references. Most research related to foundation models in particle physics describe the data using specialized tokenization and are trained via self-supervised learning [9–15]. While these models are also promising candidates for investigating knowledge transfer between particle physics and cosmology, they introduce additional challenges. Because the learned tokenization depends on the set of features used to describe particle interactions, a new tokenizer must be defined for the cosmological data. This contrasts with OMNILEARNED, where replacing a single layer is sufficient to accommodate the different input representation. Moreover, most tokenizers require an explicit ordering of the inputs. Since the ordering of the input halos should not affect the regression performance, a point-cloud representation that is permutation invariant by design is better aligned with the structure of the data.

A number of related works have touched upon some aspects relates to our research. For example, a number of studies have explored point-cloud/graph-based machine learning models applied to cosmology [16–35]. COSMOBENCH includes one such model, and we also compare our foundation model to a setup with the same architecture, but trained from scratch. This allows us to disentangle benefits from an optimized architecture and from pretraining. The philosophy of OMNILEARNED is to encode most of the physics in the model implicitly through pretraining. Explicitly equivariant models have been proposed for cosmological inference [36–51] and it would be interesting to explore in the future whether pretraining outperforms them as it seems to for colliders [5]. There are many active efforts to create foundation models specific to cosmology [52–58], but we are not aware of any trained to ingest entire universes as point clouds. The approach to uncertainty quantification in COSMOBENCH is relatively simple and it would be interesting to explore the interplay between advanced uncertainty quantification (UQ) (see, e.g., Refs. [59,60] and references within) and foundation models in the future.

The paper is organized as follows. Section II describes the OMNILEARNED model architecture and the modifications used to adapt to cosmology problems. Section III then introduces the dataset used to train OMNICOSSOS. Results for different tasks and datasets are then described in Sec. IV. The paper ends with conclusions and outlook in Sec. V.

II. KNOWLEDGE TRANSFER

In particle physics, the set of reconstructed particles produced after a collision is used to study how fundamental particles interact with each other. At high-energy colliders like

the large hadron collider (LHC), a ubiquitous object produced through particle interactions is a jet or a collimated spray of particles. Produced through the interactions of colored particles via the strong force, jets are fundamental research objects as they provide context regarding the properties of quantum chromodynamics as well as the underlying physics process that emerges from the particle collision. Due to their abundance and complexity, many machine learning applications in particle physics were initially proposed for jets [9,61,62], including convolutional neural networks (CNNs) [63], recursive neural networks (RNNs) [64], graph neural networks (GNNs) [65,66], generative adversarial networks (GANs) [67], deep sets [68], and attention [69]/transformers [70,71]. The current state of the art [5,71–73] represents jets as point clouds, or unordered sets of objects in a metric space. These point clouds are processed with transformer models that can account for long-range correlations between particles and can be scaled effectively to bigger models trained on more data.

The OMNILEARNED model [3–5] uses particles clustered into jets as part of the inputs and employs different tasks to create a rich point-cloud representation that can be efficiently adapted to different datasets and tasks. The underlying neural network model, named PET-v2, consists of a mixture of local and global transformer layers tasked to learn local and global correlations between particles. Crucially, generic transformer layer logits A_{ij} are augmented using physics-inspired interaction bias terms B_{ij} ,

$$A_{ij}^* = A_{ij} + B_{ij}, \quad B_{ij} \in \mathbb{R}, \quad (1)$$

shown to improve performance even in the presence of large training datasets. The model is then trained using one billion jets containing a mixture of real particle collisions, as well as simulated particle interactions with different degrees of simulation fidelity. The result of this pretraining stage represents the core foundation model that we refer to OMNILEARNED. To adapt to specific applications, we fine-tune OMNILEARNED to the specific dataset and task to be solved. The first challenge is to identify an efficient strategy to adapt OMNILEARNED to a completely different dataset. Whereas particles used as inputs to OMNILEARNED are described by their four-momenta, the dark matter halo simulations used in this work are described by their spatial location in three-dimensional (3D) coordinates. Moreover, the relevant tasks considered in the pretraining phase of OMNILEARNED are point-cloud classification and generation, while in this case, we are interested in the regression of different cosmological parameters and the halo velocity prediction for each input object. Last, an average jet contains $\mathcal{O}(10\text{--}100)$ particles while we consider up to 5000 of the most massive halos in the COSMOBENCH datasets. We address all these challenges by introducing OMNICOSSOS, with a general adaptation strategy shown in Fig. 1.

First, we address the differences in object representation between point clouds. In the PET-v2 model, the initial input features for each point in the point cloud are transformed using a feature embedding that maps the inputs to a large latent space. Fully connected layers with nonlinear activations take as inputs only the initial feature representation, applying

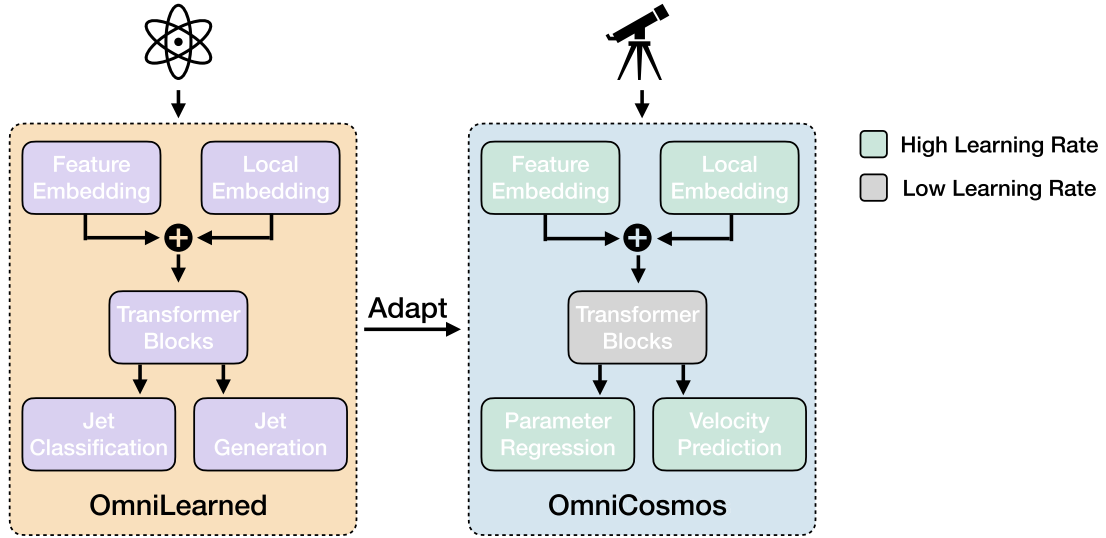


FIG. 1. Summary of the knowledge adaptation strategy between the OMNILEARNED model, trained on particle collisions, to cosmology tasks named OMNICOSSOS.

the same transformation to all points in the point cloud to ensure permutation equivariance of the outputs.

A local embedding that includes the local particle information is determined in parallel using the information of the k nearest neighbors of each input point. For each object $\vec{x}_i$, we identify the k nearest neighbors $\vec{x}_j$ using the Euclidean distances in input space. Next, we augment the original input features by including relevant functions $f(\vec{x}_i, \vec{x}_j)$ calculated between pairs of points. At this step, we are able to include tailored functions specific to the physics problem and dataset used. While in OMNILEARNED we include several features that are relevant for particle physics, such as the Lorentz invariant mass of the combination of two particles, we instead identify three geometrical features to be included in addition to the difference $\vec{x}_i - \vec{x}_j$. We include the absolute distance $d(\vec{x}_i, \vec{x}_j) = \|\vec{x}_i - \vec{x}_j\|_2$, the cosine distance $d_{\cos}(\vec{x}_i, \vec{x}_j) = 1 - \frac{\vec{x}_i \cdot \vec{x}_j}{\|\vec{x}_i\|_2 \|\vec{x}_j\|_2}$, and the cosine distance between the initial halo $\vec{x}_i$ and the difference $\vec{x}_i - \vec{x}_j$, described by $d_{\cos}(\vec{x}_i, \vec{x}_i - \vec{x}_j) = 1 - \frac{\vec{x}_i \cdot (\vec{x}_i - \vec{x}_j)}{\|\vec{x}_i\|_2 \|\vec{x}_i - \vec{x}_j\|_2}$. The resulting pairwise features considered are

$$f(\vec{x}_i, \vec{x}_j) = [\vec{x}_i - \vec{x}_j, d(\vec{x}_i, \vec{x}_j), d_{\cos}(\vec{x}_i, \vec{x}_j), d_{\cos}(\vec{x}_i, \vec{x}_i - \vec{x}_j)]. \quad (2)$$

These pairwise features are used as inputs to a transformer block, where the attention mechanism attends to only particles $\vec{x}_i$ and their k neighbors. Last, we take the average response across the k neighbors to define the new representation for each point.

The final latent representation for each point in the point cloud is the sum of the two previously described operations, the one that applies the same learnable function to all particles and the one that leverages the local neighborhood.

To create OMNICOSSOS with OMNILEARNED pretraining, we start by loading all the pretrained weights of OMNILEARNED in the new network architecture. Whenever weights are incompatible due to differences in input

shapes, randomly initialized weights are used. These correspond to only 0.09% of the total model weights. Since the new layers require more training steps for convergence, we set the learning rate of the newly initialized layers higher than the learning rates of pretrained layers. This also has the effect of properly retaining the strong geometric priors learned in pretraining while preventing “catastrophic forgetting.”

The second challenge is to address the new training tasks. Whereas OMNILEARNED focuses on classification and particle generation, we are interested in regressing cosmological parameters and predicting the three-dimensional halo velocity for each object in the point cloud. While these tasks are remarkably different in scope, they are similar in output. In the classification task, we predict class scores for each jet based on all available classes. For regression, the only modification needed is to change the output size to match the number of parameters to predict. For the velocity prediction, we can repurpose the particle generator network, which originally assigned an output prediction per particle, to instead predict the halo velocities. In this case, again, we only need to replace the last layer with a new one to correct for the difference in output shape. This layer is newly initialized, and thus the learning rate assigned is also higher.

The last challenge is to address the difference in the number of points between the original jet data and the current datasets. Since OMNILEARNED does not enforce a fixed particle multiplicity, no additional modification is required. However, the tenfold increase in the number of particles restricts the number of transformer blocks that can be used due to memory requirements. Because of this, we study only the adaptation of the small OMNILEARNED model, with eight transformer blocks and a total of around two million trainable parameters.

The result of the knowledge adaptation between the OMNILEARNED model, initially trained on particle collisions, is the OMNICOSSOS model, used to address multiple tasks in cosmology.

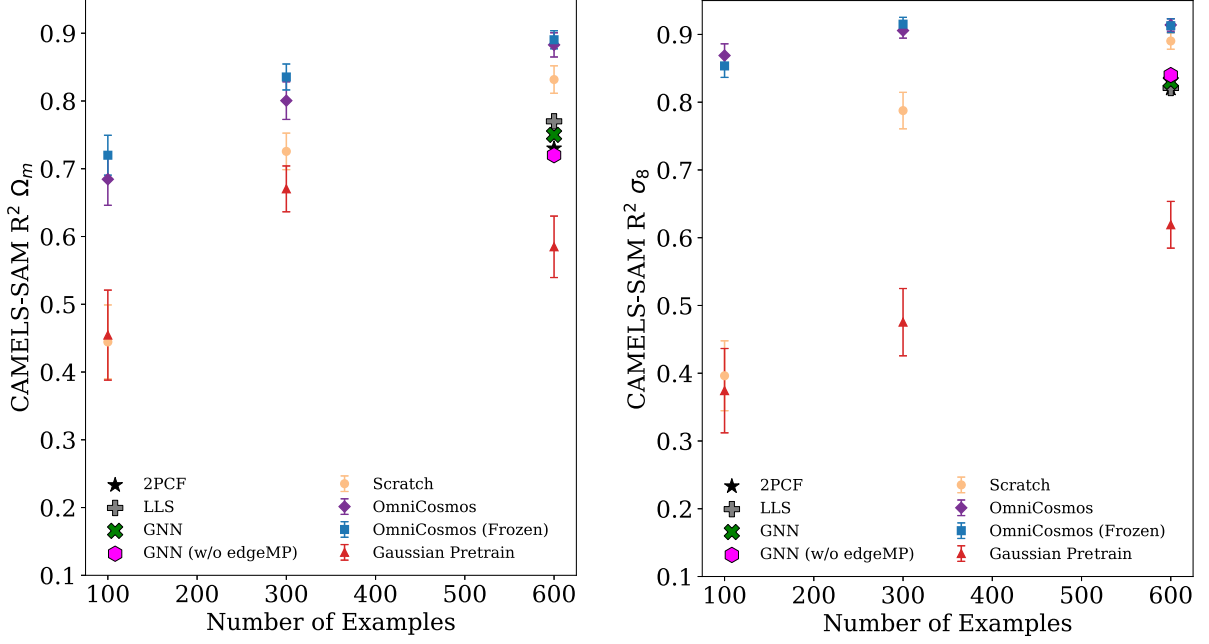


FIG. 2. Results for the prediction of different cosmological parameters obtained using the CAMELS-SAM simulations.

III. COSMOBENCH DATASET

The COSMOBENCH [6] dataset introduces large-scale cosmological simulations of dark matter halos and galaxies using multiple physics simulators, amounting to more than 41 million core hours to produce. In this study, we consider two datasets from COSMOBENCH: the QUIJOTE [74] and CAMELS-SAM [75] simulation suites. In both datasets, we are interested in estimating the values of cosmological parameters Ω_m and σ_8 , representing the fraction of the total energy density of the universe made up of matter (including both dark matter and normal matter) and the amplitude of matter density fluctuations, respectively. From random initial conditions, the different programs use N -body simulations to evolve the systems to present time. The CAMELS-SAM simulations consist of dark-matter-only N -body simulations postprocessed with a semianalytic galaxy-formation model with a box size of 100 cMpc/ h and dark matter particle mass of $\sim 10^8 M_\odot/h$. The QUIJOTE simulations use a box of size 1000 cMpc/ h and consider dark matter particles of masses . Due to the lower resolution, the QUIJOTE simulations are cheaper to create compared to CAMELS-SAM, where the higher mass resolution allows it to resolve smaller, highly nonlinear scales, at the expense of simulating a smaller volume. In both datasets, we use up to 5000 most massive halos available in the simulations. We use the official dataset split for each dataset, consisting of 600/204/196 samples for training, validation, and testing in the CAMELS-SAM dataset and 19651/6550/6551 for the QUIJOTE simulations.

IV. RESULTS

We evaluate the performance of OMNICOSSOS to regress cosmological parameters in the CAMELS-SAM simulations using different amounts of available training samples. We quantify the performance using the coefficient of determina-

tion R_y^2 for a parameter y determined as

$$R_y^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\bar{y}_i - y_i)^2}, \quad (3)$$

where the model prediction $\hat{y}$ is compared against the true value y with mean value $\bar{y}$ for each entry i in the test dataset. We determine the uncertainties on R_y^2 by bootstrapping the test data 1000 times with replacement, reporting the mean value as the final prediction and the standard deviation of the bootstraps as the uncertainty. We directly compare our results with the ones provided in Ref. [6] without retraining.

Inputs are standardized before the training. We test different hyperparameters for both OMNICOSSOS and the model trained from scratch, i.e., with all trainable weights initialized from random weights. Since the network parameters are fixed from the original OMNILEARNED model, we consider different values of learning rate, batch size, and the multiplicative factor between the lower learning rate and higher learning rate layers in the OMNICOSSOS fine-tuning. The settings corresponding to the best-performing models are reported in the final results.

TABLE I. Comparison between the performance reported for different algorithms on the CAMELS-SAM dataset. Bold results represent the algorithm with highest performance.

$R^2 \uparrow$	Ω_m	σ_8
2PCF	0.73 ± 0.03	0.82 ± 0.02
LLS	0.77 ± 0.03	0.82 ± 0.02
GNN	0.75 ± 0.03	0.83 ± 0.02
GNN (w/o edgeMP)	0.72 ± 0.03	0.84 ± 0.02
Scratch	0.83 ± 0.02	0.89 ± 0.01
OMNICOSSOS	0.88 ± 0.01	0.91 ± 0.01
OMNICOSSOS (frozen)	0.89 ± 0.01	0.91 ± 0.01
Gaussian pretrain	0.58 ± 0.04	0.61 ± 0.03

To determine the size of the local neighborhood from each point, we use the value of $k = 10$. The results for the prediction of both Ω_m and σ_8 are shown in Table I and Fig. 2 as a function of the number of examples in the training set. In addition to the comparison between the model trained from scratch and OMNICOSSOS, we consider two additional ablation results: one where the transformer layers of the original OMNILEARNED model are frozen, i.e., are not updated during the adaptation step, and one where the same architecture is first pretrained on a synthetic point cloud. The first ablation is used to study how useful are the learned representations from the OMNILEARNED model when adapting to new datasets; the second is to understand how useful is the particle physics representation used to pretrain the model. The synthetic point cloud consists of ten equally populated classes, with each class represented by four independent Gaussian-distributed features. For class $c \in \{0, \dots, 9\}$ and feature $f \in \{0, \dots, 3\}$, the corresponding Gaussian mean and standard deviation are defined as

$$\mu_{c,f} = -2.5 + \frac{4c}{9} + \frac{f}{3} \quad (4)$$

and

$$\sigma_{c,f} = \left(0.5 + \frac{c}{9}\right) \left(0.8 + \frac{0.4f}{3}\right). \quad (5)$$

Both the feature means and standard deviations increase monotonically with the class label and result in features with comparable ranges as the ones used to pretrain the OMNILEARNED model. As in the original OMNILEARNED model, pretraining jointly targets two tasks: classifying the input point clouds and generating them. The synthetic pretraining dataset contains ten million point clouds, each composed of 150 points, chosen to match the typical point-cloud size used to pretrain the OMNILEARNED model. During the adaptation phase, the transformer blocks of this pretrained model are also kept frozen to enable the direct comparison with the OMNILEARNED model. We observe the performance obtained by both the fine-tuned OMNICOSSOS model and the model trained from scratch to surpass previous benchmarks when using the full available training dataset. Interestingly, with only 100 examples of the available simulations OMNICOSSOS is already able to match or improve upon the performance from previous benchmarks. Moreover, we see a strong benefit from the fine-tuning strategy as opposed to training from scratch, where at all dataset sizes, the OMNICOSSOS model is able to obtain better predictions of the parameters, showing promising results even at the very small dataset size of 100 simulations. We also observe that the full fine-tuning strategy and the strategy in which the transformer blocks are frozen yield similar results within uncertainties. This indicates that the representations learned during pretraining of the OMNILEARNED model are also useful for cosmology. This conclusion is further supported by the results obtained with the Gaussian pretraining sample, which match the model trained from scratch at small dataset sizes and improve only slightly as more data are added, since the most expressive part of the architecture remains frozen.

Next, we evaluate the predictions of the halo velocities using only the input positions. We employ again the R^2 met-

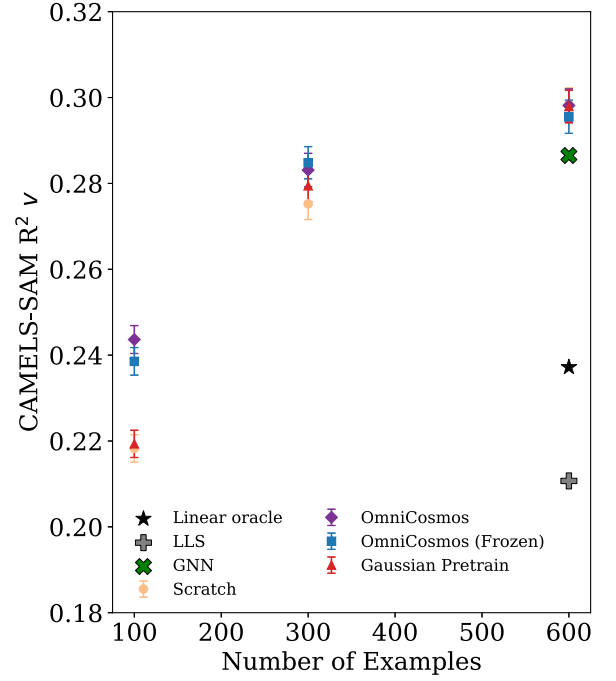


FIG. 3. Results for the halo velocity prediction using the CAMELS-SAM simulations.

ric to evaluate the results, but sum over all halos and the velocity coordinates, resulting in a single metric value per simulation. The results are shown in Fig. 3. Similarly to the regression task, we observe good performance from both OMNICOSSOS and the model trained from scratch, surpassing previous benchmarks when all available training events are used. This task also highlights the benefits of using more complex deep learning models as opposed to simpler summaries, with the GNN benchmark introduced in Ref. [6] also showing strong performance. The improvements observed from OMNICOSSOS compared to the model trained from scratch are reduced, although still present at all dataset sizes. At small dataset sizes, both OMNICOSSOS models with full fine-tuning or frozen backbones are significantly better than the model trained from scratch that shows similar performance to the Gaussian pretraining model. This indicates that at small dataset sizes, the learned representation from OMNICOSSOS is beneficial to improve the prediction quality, whereas the Gaussian pretraining does not yield significant benefits. These results are encouraging and show that models trained on large collider physics datasets encode relevant point-cloud structure that can be transferred to other scientific fields.

Next, we investigate the ability of our model to determine the same set of cosmological parameters but using the QUIJOTE simulations. We increase the number of neighbors considered to $k = 20$, as we observe a small gain in performance using a bigger neighborhood, and scan different hyperparameter settings for both the models training from scratch and different pretraining and adaptation strategies. The results of the best-performing models are shown in Table II and Fig. 4 for the regression of Ω_m and σ_8 using different dataset sizes. Given the bigger simulation samples, we can

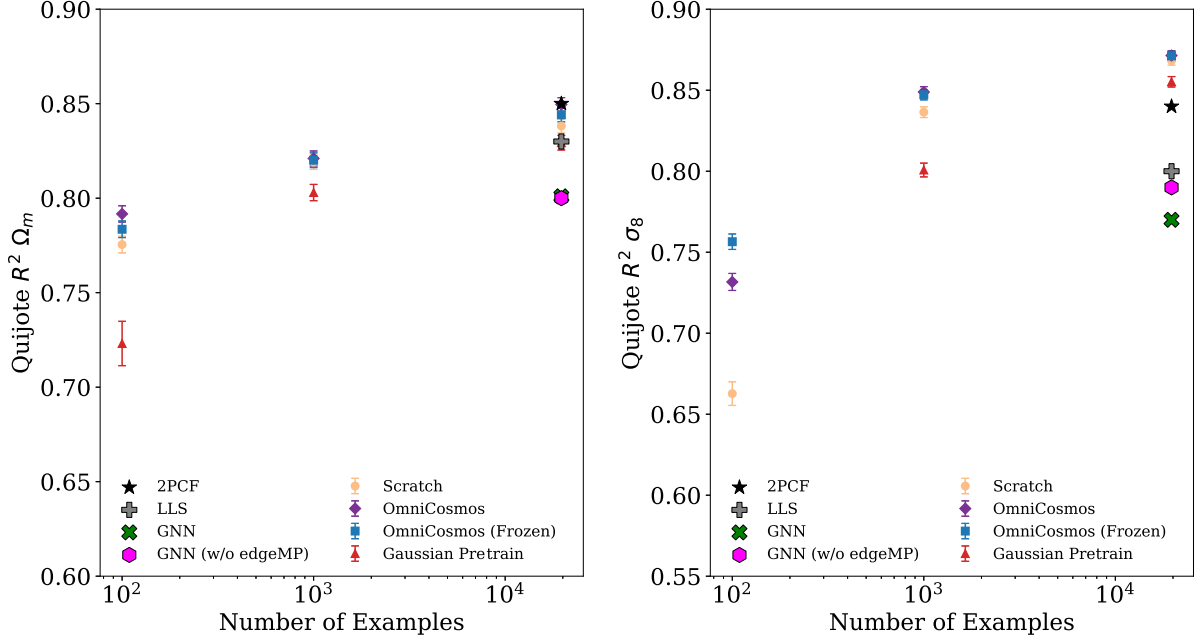


FIG. 4. Results for the prediction of different cosmological parameters obtained using the QUIJOTE simulations.

evaluate the model scaling across simulation sizes of different orders of magnitude. Similarly to the results observed using the CAMELS-SAM simulation, the performance obtained by OMNICOSSOS either matches the best-performing benchmarks or surpasses them. In particular, OMNICOSSOS is able to match previous benchmarks for σ_8 using less than 10% of the available simulations. Interestingly, for Ω_m , OMNICOSSOS matches the prediction from a two-point correlation function (2PCF), but otherwise greatly improves on the previous deep learning model (GNN). We again observe the model with frozen transformer layers performing equally well as the full fine-tuning OMNICOSSOS, in agreement with the results obtained with the CAMELS-SAM simulations. The fine-tuned model pretrained on the synthetic dataset, however, performs significantly worse. In particular, when only 100 examples are used, the adapted model fails to predict σ_8 and instead returns approximately the average value present in the training samples, irrespective of the input data. This results in an R^2 value close to zero, which is therefore not shown in Fig. 4.

TABLE II. Comparison between the performance reported for different algorithms on the QUIJOTE dataset. Bold results represent the algorithm with highest performance.

$R^2 \uparrow$	Ω_m	σ_8
2PCF	0.85 ± 0.004	0.84 ± 0.004
LLS	0.83 ± 0.004	0.80 ± 0.004
GNN	0.80 ± 0.004	0.77 ± 0.005
GNN (w/o edgeMP)	0.80 ± 0.004	0.79 ± 0.005
Scratch	0.838 ± 0.003	0.868 ± 0.003
OMNICOSSOS	0.849 ± 0.003	0.871 ± 0.003
OMNICOSSOS (frozen)	0.844 ± 0.004	0.871 ± 0.003
Gaussian pretrain	0.829 ± 0.004	0.855 ± 0.003

This behavior remains consistent across all hyperparameter scans performed in this work. These results indicate not only that the synthetic dataset fails to produce useful representations, but also that it harms generalization, leading to worse performance than a model initialized with random weights. Last, we evaluate the velocity prediction using the QUIJOTE simulations with results shown in Fig. 5. In previous benchmarks, a simple linear least squares (LLS) model obtained

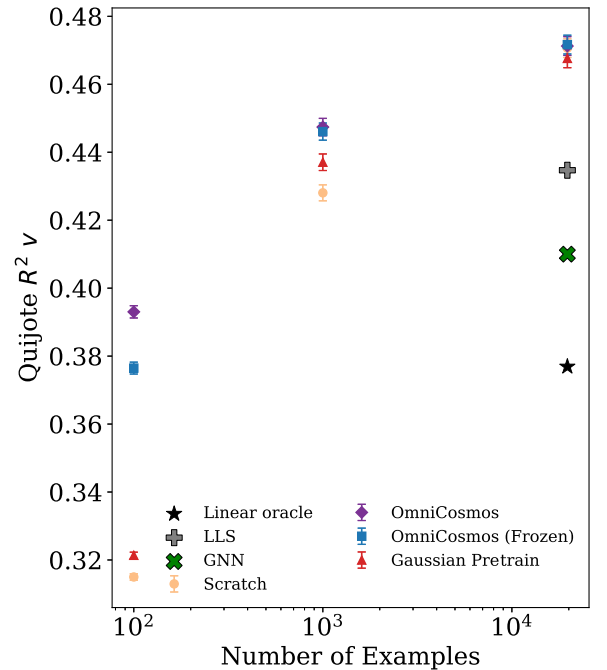


FIG. 5. Results for the halo velocity prediction using the QUIJOTE simulations.

TABLE III. Comparison between the performance reported for different algorithms for halo velocity prediction on the CAMELS-SAM and QUIJOTE datasets. Bold results represent the algorithm with highest performance.

$R_v^2 \uparrow$	CAMELS-SAM	QUIJOTE
Linear oracle	0.2372	0.3769
LLS	0.2107	0.4347
GNN	0.2865	0.4100
Scratch	0.298 $\pm$ 0.004	0.470 $\pm$ 0.003
OMNICOSSOS	0.298 $\pm$ 0.004	0.471 $\pm$ 0.003
OMNICOSSOS (frozen)	0.295 $\pm$ 0.004	0.471 $\pm$ 0.003
Gaussian pretrain	0.298 $\pm$ 0.004	0.467 $\pm$ 0.003

the best predictions for the halo velocities. OMNICOSSOS is able to further improve on previous results, surpassing the LLS results using less than 10% of the available dataset. Moreover, we again observe the benefits from the knowledge transfer. Compared to the model trained from scratch, OMNICOSSOS is able to show superior performance up to thousands of simulation samples while converging to the same results when more data are used. Similarly to the CAMELS-SAM results, the model pretrained on the synthetic Gaussian dataset performs more similarly to the model trained from scratch. In contrast, OMNICOSSOS achieves strong performance with both full fine-tuning and frozen transformer layers. The results for the velocity prediction in both datasets studies are listed in Table III.

V. CONCLUSION AND OUTLOOK

In this work, we introduced OMNICOSSOS, a machine learning model that leverages a pretrained network trained on large particle physics datasets to perform different tasks in cosmology. We show how the representation learned by particle interactions can be transferred to a new dataset by quickly adapting the input representation as well as including geometrical insight in the model architecture. Even though the original OMNILEARNED model was trained on point clouds with one order of magnitude fewer points and for different tasks, the OMNICOSSOS model is often able to achieve superior performance compared to the same model trained from scratch. We show the benefits of the pretraining strategy by comparing the results obtained with two additional scenarios: one where the transformer blocks of the model are kept frozen during adaptation and one where the same architecture is pretrained on a synthetic Gaussian dataset. We observe that the model with frozen backbone achieves similar performance compared to the full OMNICOSSOS model, indicating that the learned relationship between points in the point cloud from collider physics is transferable to cosmology. This observation is further supported by the results obtained using the synthetic dataset, which often underperform compared to OMNICOSSOS. We evaluate OMNICOSSOS using simulations available from COSMOBENCH. In particular, for computationally expensive simulations of galaxy halos such as CAMELS-SAM, OMNICOSSOS is able to improve upon all previous benchmarks using half the available dataset. In

the QUIJOTE simulations, OMNICOSSOS is able to match or surpass all other benchmarks, often using less than 10% of the available data. Since large-scale simulations are computationally expensive, the performance observed by OMNICOSSOS has the potential to greatly reduce the number of simulations necessary to achieve precision for cosmology.

The adaptation step requires fewer than 0.1% of the original model weights to be replaced with randomly initialized weights. Moreover, while pretraining on collider physics data requires substantial computational resources, adapting the OMNILEARNED model is considerably less expensive. For instance, adapting OMNILEARNED to the full CAMELS-SAM dataset takes approximately 8 GPU hours for each task. Although slower, this remains comparable to the 1–3 GPU hours reported for the GNN in Ref. [6]. When only 100 simulated samples are used, adaptation takes approximately 4 GPU hours while matching or surpassing all baselines for parameter regression. For the QUIJOTE simulations, adaptation to the full dataset takes approximately 32 GPU hours, comparable to the 1-day training time reported in Ref. [6].

Although we observe performance improvements with hundreds of examples, the differences between the model trained from scratch and OMNICOSSOS are reduced when thousands of training examples are available. In this work, we explore a simple adaptation strategy where the learning rates of different parts of the model are modified or frozen. For future work, we plan to investigate alternative strategies to perform the adaptation step, using other techniques introduced in the machine learning community to adapt foundation models to specific problems, such as Refs. [76,77].

Finally, beyond the regression of cosmological parameters and halo velocity prediction, it would be interesting to directly use the generative capabilities of OMNILEARNED in the adaptation step of OMNICOSSOS for point-cloud generation, where the use of a pretrained model can potentially reduce the number of simulated datasets needed to achieve high-fidelity generation, resulting in fast and accurate surrogate models for cosmology.

ACKNOWLEDGMENTS

We thank Mariel Pettee for helpful insights during earlier discussions about related projects. We acknowledge the use of Claude 3.5 Sonnet for help building some of the references, which have all been manually checked. V.M. was supported by JST EXPERT-J, Japan Grant No. JPMJEX2509. B.N. was supported by the U.S. Department of Energy (DOE), Office of Science under Contract No. DE-AC02-76SF00515. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 using NERSC Awards No. ERCAP0034229, No. HEP-ERCAP0021099, and No. HEP-ERCAP0028249.

DATA AVAILABILITY

The code for this paper can be found at [78]. The data that support the findings of this article are openly available from the CosmoBench dataset [6].

- [1] F. Leclercq, Field-level inference in cosmology, in Les Houches Summer School 2025: The Dark Universe, [arXiv:2509.13435](#).
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17* (Curran Associates Inc., Red Hook, NY, 2017), pp. 6000–6010.
- [3] V. Mikuni and B. Nachman, Solving key challenges in collider physics with foundation models, *Phys. Rev. D* **111**, L051504 (2025).
- [4] V. Mikuni and B. Nachman, Method to simultaneously facilitate all jet physics tasks, *Phys. Rev. D* **111**, 054015 (2025).
- [5] W. Bhimji, C. Harris, V. Mikuni, and B. Nachman, Foundation model framework for all tasks involving jet physics, *Phys. Rev. D* **113**, 032020 (2026).
- [6] N. Huang, R. Stiskalek, J.-Y. Lee, A. E. Bayer, C. C. Margossian, C. K. Jespersen, L. A. Perez, L. K. Saul, and F. Villaescusa-Navarro, CosmoBench: A multiscale, multiview, multitask cosmology benchmark for geometric deep learning, [arXiv:2507.03707](#).
- [7] It will be interesting to study how the results change with finer granularity/higher multiplicity cosmological simulations.
- [8] It may be interesting to explore connections between pretrained collider models trained using parton shower histories for such tasks in the future.
- [9] M. Feickert and B. Nachman, A living review of machine learning for particle physics, [arXiv:2102.02770](#).
- [10] J. Birk, A. Hallin, and G. Kasieczka, OmniJet- α : The first cross-task foundation model for particle physics, *Mach. Learn.: Sci. Technol.* **5**, 035031 (2024).
- [11] A. Hallin, Foundation models for high-energy physics, in 2nd European AI for Fundamental Physics Conference, [arXiv:2509.21434](#).
- [12] P. Harris, M. Kagan, J. Krupa, B. Maier, and N. Woodward, Resimulation-based self-supervised learning for pretraining physics foundation models, *Phys. Rev. D* **111**, 032010 (2025).
- [13] T. Golling, L. Heinrich, M. Kagan, S. Klein, M. Leigh, M. Osadchy, and J. A. Raine, Masked particle modeling on sets: Towards self-supervised high energy physics foundation models, *Mach. Learn.: Sci. Technol.* **5**, 035074 (2024).
- [14] M. Leigh, S. Klein, F. Charton, T. Golling, L. Heinrich, M. Kagan, I. Ochoa, and M. Osadchy, Is tokenization needed for masked particle modelling? [arXiv:2409.12589](#).
- [15] J. Bardhan, R. Agrawal, A. Tilak, C. Neeraj, and S. Mitra, HEP-JEPA: A foundation model for collider physics using joint embedding predictive architecture, [arXiv:2502.03933](#).
- [16] S. Ravanbakhsh, J. Oliva, S. Fromenteau, L. Price, S. Ho, J. Schneider, and B. Poczos, Estimating cosmological parameters from the dark matter distribution, in *Proceedings of the 33rd International Conference on Machine Learning*, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, 2016), Vol. 48, pp. 2407–2416.
- [17] T. L. Makinen, T. Charnock, P. Lemos, N. Porqueres, A. Heavens, and B. D. Wandelt, The Cosmic Graph: Optimal information extraction from large-scale structure using catalogues, *Open J. Astrophys.* **5** (2022).
- [18] P. Villanueva-Domingo and F. Villaescusa-Navarro, Learning cosmology and clustering with cosmic graphs, *Astrophys. J.* **937**, 115 (2022).
- [19] N. S. M. de Santi *et al.*, Robust field-level likelihood-free inference with galaxies, *Astrophys. J.* **952**, 69 (2023).
- [20] A. Roncoli, A. Čiprijanović, M. Voetberg, F. Villaescusa-Navarro, and B. Nord, Domain adaptive graph neural networks for constraining cosmological parameters across multiple data sets, in *Proceedings of the 37th Conference on Neural Information Processing Systems*, [arXiv:2311.01588](#).
- [21] E. Massara, F. Villaescusa-Navarro, and W. J. Percival, Predicting interloper fraction with graph neural networks, *J. Cosmol. Astropart. Phys.* **12** (2023) 012.
- [22] H. Shao *et al.*, Robust field-level inference of cosmological parameters with dark matter halos, *Astrophys. J.* **944**, 27 (2023).
- [23] C. Cuesta-Lazaro and S. Mishra-Sharma, Point cloud approach to generative modeling for galaxy surveys at the field level, *Phys. Rev. D* **109**, 123531 (2024).
- [24] C. K. Jespersen, M. Cranmer, P. Melchior, S. Ho, R. S. Somerville, and A. Gabrielpillai, Mangrove: Learning galaxy properties from merger trees, *Astrophys. J.* **941**, 7 (2022).
- [25] J. F. Wu and C. K. Jespersen, Learning the galaxy-environment connection with graph neural networks, in *Proceedings of the 40th International Conference on Machine Learning*, [arXiv:2306.12327](#).
- [26] C.-Y. Chuang, C. K. Jespersen, Y.-T. Lin, S. Ho, and S. Genel, Leaving no branches behind: Predicting baryonic properties of galaxies from merger trees, *Astrophys. J.* **965**, 101 (2024).
- [27] J. F. Wu, C. K. Jespersen, and R. H. Wechsler, How the galaxy–halo connection depends on large-scale environment, *Astrophys. J.* **976**, 37 (2024).
- [28] T. Nguyen *et al.*, How DREAMS are made: Emulating satellite galaxy and subhalo populations with diffusion models and point clouds, *Astrophys. J.* **997**, 336 (2026).
- [29] P. Villanueva-Domingo, F. Villaescusa-Navarro, D. Anglés-Alcázar, S. Genel, F. Marinacci, D. N. Spergel, L. Hernquist, M. Vogelsberger, R. Dave, and D. Narayanan, Inferring halo masses with graph neural networks, *Astrophys. J.* **935**, 30 (2022).
- [30] M. Cranmer, P. Melchior, and B. Nord, Unsupervised resource allocation with graph neural networks, in *Proceedings of the 34th Conference on Neural Information Processing Systems* (PMLR, 2021).
- [31] T. Wang and P. Melchior, Graph neural network-based resource allocation strategies for multi-object spectroscopy, *Mach. Learn.: Sci. Technol.* **3**, 015023 (2022).
- [32] J. W. Park, S. Birrer, M. Ueland, M. Cranmer, A. Agnello, S. Wagner-Carena, P. J. Marshall, and A. Roodman (LSST Dark Energy Science), Hierarchical inference of the lensing convergence from photometric catalogs with Bayesian graph neural networks, *Astrophys. J.* **953**, 178 (2023).
- [33] A. Chatterjee and F. Villaescusa-Navarro, Cosmology from point clouds with dark matter halos from the Quijote simulations, *Astrophys. J.* **985**, 132 (2025).
- [34] J. Balla, S. Mishra-Sharma, C. Cuesta-Lazaro, T. Jaakkola, and T. Smidt, A cosmic-scale benchmark for symmetry-preserving data processing, in *The 3rd Learning on Graphs Conference*, [arXiv:2410.20516](#).
- [35] V. Krishnaraj, A. E. Bayer, C. K. Jespersen, and P. Melchior, Transfer learning beyond the Standard Model, *J. Cosmol. Astropart. Phys.* **06** (2026) 026.
- [36] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, DeepSphere: Efficient spherical convolutional neural network with

- HEALPix sampling for cosmological applications, *Astron. Comput.* **27**, 130 (2019).
- [37] M. Defferrard, N. Perraudin, T. Kacprzak, and R. Sgier, DeepSphere: Towards an equivariant graph-based spherical CNN, in *ICLR Workshop on Representation Learning on Graphs and Manifolds* (ICLR, 2019).
- [38] T. S. Cohen, M. Geiger, J. Koehler, and M. Welling, Spherical CNNs, *Proceedings of the International Conference on Learning Representations* (ICLR, 2018).
- [39] J. Ocampo, M. A. Price, and J. D. McEwen, Scalable and equivariant spherical CNNs by discrete-continuous (DISCO) convolutions, in *International Conference on Learning Representations* (ICLR, 2023).
- [40] C. Esteves, A. Makadia, and K. Daniilidis, Spin-weighted spherical CNNs, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., Red Hook, NY, 2020), Vol. 33, pp. 8614–8625.
- [41] N. Krachmalnicoff and M. Tomasi, Convolutional neural networks on the HEALPix sphere: A pixel-based algorithm and its application to CMB data analysis, *Astron. Astrophys.* **628**, A129 (2019).
- [42] J. A. P. Castro, H. J. Hortúa, J. E. García-Farieta, and R. A. Hurtado, Bayesian inference of primordial magnetic field parameters from CMB with spherical graph neural networks, *Universe* **12**, 34 (2026).
- [43] B. Dai and U. Seljak, Translation and rotation equivariant normalizing flow (TRENF) for optimal cosmological analysis, *Mon. Not. R. Astron. Soc.* **516**, 2363 (2022).
- [44] Y. Jagvaral, F. Lanusse, and R. Mandelbaum, Geometric deep learning for galaxy-halo connection: A case study for galaxy intrinsic alignments, *Mon. Not. R. Astron. Soc.* **542**, 2560 (2025).
- [45] D.-A. Onutu, Y. Zhao, J. Vanschoren, and V. Menkovski, Score matching on large geometric graphs for cosmology generation, *arXiv:2508.16990*.
- [46] L. Thiele, M. Cranmer, W. Coulton, S. Ho, and D. N. Spergel, Predicting the thermal Sunyaev–Zel’dovich field using modular and equivariant set-based neural networks, *Mach. Learn.: Sci. Technol.* **3**, 035002 (2022).
- [47] K. Minartz, Y. Poels, S. M. Koop, and V. Menkovski, Equivariant neural simulators for stochastic spatiotemporal dynamics, in *Proceedings of the 37th Conference on Neural Information Processing Systems* (NIPS, CA, 2023).
- [48] S. Villar, D. W. Hogg, K. Storey-Fisher, W. Yao, and B. Blum-Smith, Scalars are universal: Equivariant machine learning, structured like classical physics, in *Advances in Neural Information Processing Systems*, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., Red Hook, NY, 2021), Vol. 34, pp. 28848–28863.
- [49] M. de los Rios, S. Di Gioia, F. Iocco, and R. Trotta, Dark matter profiles of in silico galaxies: Deep learning inference, *J. Cosmol. Astropart. Phys.* **05** (2026) 094.
- [50] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen, 3D steerable CNNs: Learning rotationally equivariant features in volumetric data, in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2018), Vol. 31.
- [51] J. E. Gerken, J. Aronsson, O. Carlsson, H. Linander, F. Ohlsson, C. Petersson, and D. Persson, Geometric deep learning and equivariant neural networks, *Artif. Intell. Rev.* **56**, 14605 (2023).
- [52] B. Xia, N. Ramachandra, A. I. Wells, S. Habib, and J. Wise, Multi-modal foundation model for cosmological simulation data, *arXiv:2510.07684*.
- [53] L. Parker, F. Lanusse, S. Golkar, L. Sarra, M. Cranmer, A. Bietti, M. Eickenberg, G. Krawezik, M. McCabe, R. Morel, R. Ohana, M. Pettee, B. Régaldou-Saint Blancard, K. Cho, and S. A. Ho, AstroCLIP: A cross-modal foundation model for galaxies, *Mon. Not. R. Astron. Soc.* **531**, 4990 (2024).
- [54] L. Parker *et al.*, AION-1: Omnimodal foundation model for astronomical sciences, in *The 39th Annual Conference on Neural Information Processing Systems* (NIPS, 2025).
- [55] H. W. Leung and J. Bovy, Towards an astronomical foundation model for stars with a transformer-based model, *Mon. Not. R. Astron. Soc.* **527**, 1494 (2023).
- [56] T. D. Nguyen *et al.*, AstroLLaMA: Towards specialized foundation models in astronomy, *arXiv:2309.06126*.
- [57] M. Walmsley, I. V. Slijepevic, M. Bowles, and A. M. M. Scaife, Towards galaxy foundation models with hybrid contrastive learning, *arXiv:2206.11927*.
- [58] M. Walmsley *et al.*, Scaling laws for galaxy images, *arXiv:2404.02973*.
- [59] R. Chakkappai *et al.*, Fair universe Higgs uncertainty challenge, *arXiv:2509.22247*.
- [60] FAIR Universe Collaboration, FAIR Universe—Weak lensing ML uncertainty challenge: Competition overview, neurIPS 2025 Machine Learning Competition, 2025, https://fair-universe.lbl.gov/tutorials/WL_Competition_Overview.pdf.
- [61] A. J. Larkoski, I. Mout, and B. Nachman, Jet substructure at the Large Hadron Collider: A review of recent advances in theory and machine learning, *Phys. Rep.* **841**, 1 (2020).
- [62] A. Butter *et al.*, How to GAN LHC events, *SciPost Phys.* **7**, 075 (2019).
- [63] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, Jet-images—Deep learning edition, *J. High Energy Phys.* **07** (2016) 069.
- [64] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson, Jet flavor classification in high-energy physics with deep neural networks, *Phys. Rev. D* **94**, 112002 (2016).
- [65] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe, and G. Rochette, Neural message passing for jet physics, in *Proceedings of the Deep Learning for Physical Sciences (DLPS 2017)* (NIPS, Long Beach, CA, 2017).
- [66] H. Qu and L. Gouskos, Jet tagging via particle clouds, *Phys. Rev. D* **101**, 056019 (2020).
- [67] L. de Oliveira, M. Paganini, and B. Nachman, Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis, *Comput. Software Big Sci.* **1**, 4 (2017).
- [68] P. T. Komiske, E. M. Metodiev, and J. Thaler, Energy flow networks: Deep sets for particle jets, *J. High Energy Phys.* **01** (2019) 121.
- [69] V. Mikuni and F. Canelli, ABCNet: An attention-based method for particle tagging, *Eur. Phys. J. Plus* **135**, 463 (2020).
- [70] V. Mikuni and F. Canelli, Point cloud transformers applied to collider physics, *Mach. Learn.: Sci. Technol.* **2**, 035027 (2021).

- [71] H. Qu, C. Li, and S. Qian, Particle transformer for jet tagging, [arXiv:2202.03772](#).
- [72] Y. Wu, K. Wang, C. Li, H. Qu, and J. Zhu, Jet tagging with more-interaction particle transformer, [Chin. Phys. C](#) **49**, 013110 (2025).
- [73] J. Brehmer, V. Bresó, P. de Haan, T. Plehn, H. Qu, J. Spinner, and J. Thaler, A Lorentz-equivariant transformer for all of the LHC, [SciPost Phys.](#) **19**, 108 (2025).
- [74] F. Villaescusa-Navarro *et al.*, The Quijote simulations, [Astrophys. J. Suppl. Ser.](#) **250**, 2 (2020).
- [75] L. A. Perez, S. Genel, F. Villaescusa-Navarro, R. S. Somerville, A. Gabrielpillai, D. Anglés-Alcázar, B. D. Wandelt, and L. Y. A. Yung, Constraining cosmology with machine learning and galaxy clustering: The CAMELS-SAM suite, [Astrophys. J.](#) **954**, 11 (2023).
- [76] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, LoRA: Low-rank adaptation of large language models, [arXiv:2106.09685](#).
- [77] Q. Sun, E. Cetin, and Y. Tang, Transformer-squared: Self-adaptive LLMs, [arXiv:2501.06252](#).
- [78] <https://github.com/ViniciusMikuni/OmniLearned>.