Point cloud approach to generative modeling for galaxy surveys at the field level

Carolina Cuesta-Lazaro[*,†]

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA and Center for Astrophysics—Harvard & Smithsonian, 60 Garden Street, MS-16, Cambridge, Massachusetts 02138, USA

Siddharth Mishra-Sharma[*,‡]

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA; Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA; and Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA

(Received 22 December 2023; accepted 5 April 2024; published 20 June 2024)

We introduce a diffusion-based generative model to describe the distribution of galaxies in our Universe directly as a collection of points in 3D space (coordinates) optionally with associated attributes (e.g., velocities and masses), without resorting to binning or voxelization. The custom diffusion model can be used both for emulation, reproducing essential summary statistics of the galaxy distribution, as well as inference, by computing the conditional likelihood of a galaxy field. We demonstrate a first application to massive dark matter haloes in the QUIJOTE simulation suite. This approach can be extended to enable a comprehensive analysis of cosmological data, circumventing limitations inherent to summary statistics, as well as neural simulation-based inference methods.

DOI: 10.1103/PhysRevD.109.123531

I. INTRODUCTION

Cosmological data analysis is a multidisciplinary field that involves nuanced interplay between theory and data. Analysis of late-time observables of structure formation is especially challenging due to the high dimensionality of typical data and complexity of the underlying data-generating process, which aims to model, among others, the nonlinear collapse of structures, baryonic processes, and the formation of galaxies in the dark matter cosmic web. An example of such an observable is galaxy clustering—the 3D distribution of galaxies in the Universe—which is a powerful probe of cosmology and galaxy formation.

The galaxy clustering signal is typically quantified by summary statistics like the two-point correlation function (2PCF), which measures the probability of finding a pair of galaxies as a function of their separation in excess of expectation based on a uniform distribution. While routinely used in cosmological analyses, the 2PCF is not a complete (sufficient) summary of the galaxy clustering signal, and other statistics like higher-order correlation functions [1,2], wavelet scattering transforms [3–5], density statistics [6,7], void statistics [8,9], k-nearest neighbor summaries [10], and many others are routinely employed to capture additional information contained in the clustering signal, in particular at smaller scales where nonlinear structure formation is critical to the description of the field. Recent studies [6,11] have shown that the information extracted from existing galaxy surveys can be more than doubled through the use of alternative summary statistics that go beyond the 2PCF.

Machine learning methods have demonstrated the potential to significantly impact how cosmological data are analyzed, and galaxy clustering is no exception [12–14]. More concretely, the ability of neural networks to beat the curse of dimensionality allows for extraction of information about the underlying cosmology without having to manually construct summary statistics to describe the galaxy clustering field.

For galaxy clustering observations, arguably the holy grail is to obtain a reliable likelihood of an observed galaxy configuration x given some parametric description θ of underlying cosmological models of interest, $p(x|\theta)$ which is additionally amenable to sampling—a "generative model." Access to the conditional likelihood can be used to sample different field configurations, $x \sim p(x|\theta)$, for use

[*]These authors contributed equally to this work.
[†]cuestalz@mit.edu
[‡]smsharma@mit.edu

in various downstream tasks or as a surrogate model (emulation). Additionally, one can use the likelihood to perform parameter inference and hypothesis testing using a method of ones choosing. In the context of Bayesian inference, commonly employed in cosmology, the conditional likelihood can be used in conjunction with a prior $p(\theta)$ in order to obtain a estimate of the parameter posterior density, $p(\theta|x) = p(x|\theta) \cdot p(\theta)/p(x)$.

Unfortunately, computing the conditional likelihood is extremely challenging for most observationally interesting scenarios. This is because it requires marginalizing over an essentially infinite-measure space of latent configurations, denoted z, characterizing possible initial conditions and their evolution trajectories toward realizing a given observation x; $p(x|\theta) = \int \mathrm{d}z p(x, z|\theta)$. For a collection of galaxies or dark matter halos, constructing a generative model involves modeling the joint probability density of the properties (positions, velocities, etc.) of a large number of galaxies, $p\left(\{x_i\}_{i=1}^{N_{\mathrm{gal}}}|\theta\right)$, while simultaneously capturing the dependence on cosmology—a formidable task.

Machine learning has revolutionized the field of generative modeling, heralding methods that are able to learn complex data distributions such as those of natural images and human-generated text. Much of this success has been enabled through the use of diffusion models [15,16]—a class of generative models that, colloquially, learn to efficiently denoise a corrupted version of the data. Within the sciences, diffusion models have demonstrated potential across domains, showing impressive performance in modeling the distribution of atomistic systems [e.g., [17]], proteins and biomolecules [e.g., [18–20]], and particle jets [e.g., [21–23]], to name a few. Compared to other generative models, diffusion models tend to be more expressive than variational autoencoders, allow for more flexible architecture and training than normalizing flows, and can estimate approximate likelihoods unlike generative adversarial networks, while still producing diverse samples.

Within cosmology, generative modeling has recently been applied in the context of matter density fields [12,24], initial conditions reconstruction [25], weak lensing mass maps [26], galaxy images [27,28], and strong lensing observations [29,30]. In all cases, the common data modality of 2D or 3D pixelized images or voxelized boxes is used. While the image representation is appropriate in many cases, such as weak gravitational lensing, the distribution of galaxies is, arguably, ideally represented as a *point cloud*—a set of points in 3D space, with additional attributes (e.g., luminosities, velocities, as well as other galaxy properties) attached to them. Pixelization or voxelization necessarily introduces scale cuts, information loss, as well as hyperparameter choices, precluding a full *in-situ* analysis of observed data.

In this paper, we develop a diffusion-generative model with the goal of describing the statistical properties of the distribution of galaxies in our Universe. We focus here on modeling dark matter halos, leaving a more detailed exploration including effects of the galaxy-halo connection and observational effects to future work. We show that our custom diffusion model, which uses either graph neural networks or transformers as a backbone, faithfully reproduces crucial summary properties of the galaxy field with expected cosmological dependence. Furthermore, we show how our model can be used to evaluate the conditional likelihood of a galaxy field.

This paper is organized as follows. We describe our methodology, including an overview of the diffusion modeling framework, the underlying data-processing neural networks involved, and a description of the dataset and training procedure in Sec. II. We showcase generated samples and validate their properties in Sec. III. We describe in detail the methodological limitations of our model and discuss future avenues for improvement in Sec. IV. We conclude in Sec. V.

II. METHODOLOGY

We describe, in turn, the simulation dataset used, the diffusion model framework, and the noise-prediction neural network backbones of our model.

A. Dataset and forward model

Our dataset is derived from the high resolution Latin hypercube set of the QUIJOTE suite of 2000N-body simulations [31] at redshift zero. These simulations follow the evolution of 1024^3 cold dark matter particles in a volume of $(1h^{-1}\ \mathrm{Gpc})^3$ with periodic boundary conditions from $z = 127$ to $z = 0$. QUIJOTE uses the TreePM GADGET-III code and identifies halos using a friends-of-friends [32] halo-finding algorithm. Each simulation in the Latin hypercube varies the cosmological parameters described in Table I and the random phases of the initial conditions simultaneously.

We randomly split the dataset 90%/10% into training and testing sets. The held-out test is used to (1) compute validation metrics over the course of training, and (2) evaluate the cosmological parameter dependence of the trained model. We do not use separate datasets for the two purposes due to the limited total number of available simulations. To assess the ability of our model to capture the effect of cosmic variance, we use a separate set of 50 simulations with varying random phases in the initial conditions and cosmological parameters fixed to the fiducial parameter values specified in Table I.

Dark matter halo coordinates are represented as a 3D point cloud, selecting the heaviest 5000 halos by halo mass. We chose to select halos by number density, as opposed to by choosing a minimum halo mass threshold, since in observations we only have access to the former. We also use the velocity and mass attributes from the halo catalogs for a subset of our experiments to demonstrate the ability of the model to reproduce correlations in a higher-dimensional

TABLE I. Definitions and ranges of the cosmological parameters of the QUIJOTE simulation suite.

Parameter	Interpretation	Range	Fiducial
Ω_m	Matter density	[0.1, 0.5]	0.3175
Ω_b	Baryon density	[0.03, 0.07]	0.049
h	Dimensionless Hubble constant	[0.5, 0.9]	0.6711
σ_8	Amplitude of matter fluctuations in $8h^{-1}$ Mpc spheres	[0.6, 1.0]	0.834
n_s	Spectral index of the primordial power spectrum	[0.8, 1.2]	0.9624

feature space. Examples of samples from the test dataset are shown in the bottom row of Fig. 2.

B. Diffusion-based generative modeling

Diffusion models have emerged as state-of-the-art deep generative models in domains like computer vision, surpassing in flexibility and expressivity models like normalizing flows and variational autoencoders. They admit several closely related formulations. In one common framing [15], a neural network $\hat{\epsilon}_\varphi(z_t, t)$ learns to iteratively "denoise" a corrupted version z_t of the data $x \equiv z_{t=0}$ from a time step $t \in [0, T]$ by predicting either the additive noise ϵ, the original data point x directly, or some combination of the two [33]. New samples can then be generated by sampling Gaussian random noise z_T and iteratively denoising it from $t = T$ to $t = 0$. A complementary framing [16] relies on having a neural network $\hat{s}_\varphi(z_t, t)$ estimate the time-dependent gradient of the data distribution—the so-called score function, $\nabla_{z_t} \log p(z_t)$.

The two formulations are closely related. Considering Gaussian noise addition with variance σ_t^2 as the forward process, $q(z_t|x) = \mathcal{N}(z_t; x, \sigma_t^2)$, the "conditional" score can be analytically expressed as $\nabla_{z_t} \log q(z_t|x) = (x - z_t)/\sigma_t^2 = -\epsilon/\sigma_t$. Score and noise prediction are hence equivalent up to a time-step-dependent scaling. The intuition behind the relative negative sign is that, since the noise ϵ corrupts the data point, moving in its "opposite" direction will maximize the local (in time t) probability of moving toward the original data point. Hence, we refer to the noise- and score-prediction networks interchangeably.

C. Variational diffusion models

Here, we use the "variational diffusion model" formulation [34–36], which frames the diffusion process as a hierarchical variational autoencoder (VAE) with a specific (Gaussian) functional form for the transition probability between latent variable hierarchies in the forward (noise-addition) process. Much as in a classical VAE [37], the evidence lower bound objective can be used as a variational lower bound on the log-likelihood $\log p(x)$. We give a high-level overview of the formalism here; see Kingma *et al.* [36] and Luo [38] for further details.

1. The forward process

The forward (noising) process is defined by the distribution $q(z_t|z_{t-1})$, which also defined the "noise schedule" of the diffusion model. This is a critical part of the model which can have a large impact on the final fidelity and learning dynamics of the model [39]. We take this to be a variance preserving,

$$q(z_t|z_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t} \cdot z_{t-1}, \beta_t\right) \tag{1}$$

which corresponds to

$$q(z_t|x) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot x, \sqrt{1-\bar{\alpha}_t}) \tag{2}$$

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i; \qquad \alpha_t \equiv 1 - \beta_t. \tag{3}$$

This is commonly referred to as the "diffusion kernel" and can be used to conveniently predict a noised data sample at any time step t without going through intermediate times. We further define the signal-to-noise ratio or the mean-to-standard-noise ratio, $\mathrm{SNR}(t) \equiv \bar{\alpha}_t/(1-\bar{\alpha}_t)$.

2. The variational objective

For the diffusion objective, we use the efficient and numerically stable implementation of the variational evidence lower bound (ELBO) from Kingma *et al.* [36] and Kingma and Gao [40]. The ELBO can be written as

$$\begin{aligned}\log p(x) \geq \mathrm{ELBO}(x) = &\underbrace{-\mathbb{E}_{q(z_T|x)}[D_{\mathrm{KL}}(q(z_T|x)\|p(z_T))]}_{\text{Prior matching}} \\ &+ \underbrace{\mathbb{E}_{q(z_{t_1}|x)}[\log p(x|z_{t_1})]}_{\text{Reconstruction}} \\ &+ \underbrace{\mathcal{L}_{\text{diffusion}}(x)}_{\text{Forward-reverse consistency}}\end{aligned} \tag{4}$$

where z_T are the latent random variables at the last noising step, z_{t_1} are the latent variables in the first noising step, and $q(z_t|x)$ are the (assumed Gaussian) variational posteriors on the noise addition. The prior-matching and reconstruction terms are exactly analogous to a classical VAE with a

single bottleneck layer and contain no trainable parameters. The diffusion loss $\mathcal{L}_{\text{diffusion}}(x)$ ensures consistency between the forward (noising) and reverse (denoising) distribution at each step of the hierarchy [38],

$$\mathcal{L}_{\text{diffusion}}(x) = -\sum_{t=2}^{T} \mathbb{E}_{q(z_t|x)}[D_{\text{KL}}(q(z_{t-1}|z_t,x)\|p_\varphi(z_{t-1}|z_t))]. \tag{5}$$

The target denoising step $p_\varphi(z_{t-1}|z_t)$ is learned as an approximation of the ground truth $q(z_{t-1}|z_t,x)$, which corresponds to a local denoising of z_t when we have access to the target image x. For Gaussian diffusion, it can be shown [38] that the ground-truth denoising distribution can be written analytically as a Gaussian,

$$q(z_{t-1}|z_t,x) = \mathcal{N}(z_{t-1};\mu_q(z_t,x),\sigma_q(t)\mathbb{I}), \tag{6}$$

where we omit the functional forms of μ_q and σ_q for brevity. If we also assume a Gaussian functional form for the learned transition distribution $p_\varphi(z_{t-1}|z_t)$, minimizing the Kullback-Leibler (KL) divergence in Eq. (5) reduces to matching the means and variances of the two Gaussians. After some algebra, minimizing the target KL-divergence terms reduce to

$$\begin{aligned}&\arg\min_\varphi D_{\text{KL}}(q(z_{t-1}|z_t,x)\|p_\varphi(z_{t-1}|z_t))\\ &= \arg\min_\varphi \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left[\|\epsilon-\hat{\epsilon}_\varphi(z_t,t)\|_2^2\right]\end{aligned} \tag{7}$$

where $\hat{\epsilon}_\varphi(z_t,t)$ is the noise-prediction neural network that is optimized during training, parametrized by φ. In practice, the sum over time steps in Eq. (5) is computed as an expectation with appropriate scaling,

$$\begin{aligned}&\sum_{t=2}^{T}\mathbb{E}_{q(z_t|x)}[D_{\text{KL}}(q(z_{t-1}|z_t,x)\|p_\varphi(z_{t-1}|z_t))]\\ &\approx \frac{N_T}{2}\mathbb{E}_{t\sim U\{1,T\},q(z_t|x)}[D_{\text{KL}}(q(z_{t-1}|z_t,x)\|p_\varphi(z_{t-1}|z_t))].\end{aligned} \tag{8}$$

This makes diffusion models especially efficient to train—they do not require simulation of the entire trajectory back to the primal Gaussian at every training step unlike, e.g., continuous-time normalizing flows [41].

Finally, the prefactor in Eq. (7) can be elegantly written in terms of the time-dependent log-SNR, $\gamma(t)$, and the time step discretization in Eq. (8) can be taken to the continuum limit, yielding the final diffusion loss

$$\mathcal{L}_{\text{diffusion}}(x) = \frac{1}{2}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbb{I}),t\sim\mathcal{U}(0,T)}\left[\gamma'_\eta(t)\|\epsilon-\hat{\epsilon}_\varphi(z_t,t)\|_2^2\right], \tag{9}$$

where $\gamma'_\eta(t) \equiv \mathrm{d}\gamma_\eta/\mathrm{d}t$ is evaluated via automatic differentiation and the expectation over a standard Gaussian comes via the expectation over $q(z_t|x)$ in Eq. (5). Equation (9) demonstrates that the variational likelihood objective is equivalent to the traditional denoising (noise-prediction) objective [15], with an appropriate weighting prefactor [40].

The noise schedule $\gamma_\eta(t)$ is implicitly parametrized via the log-SNR, modeled as linearly increasing in time between learnable extremal values $\eta = \{\gamma_{\min},\gamma_{\max}\}$ with $\gamma(t=T) = \gamma_{\min}$ and $\gamma(t=0) = \gamma_{\max}$. The noise-prediction neural network and noise schedule parameters $\{\varphi,\eta\}$ are hence simultaneously optimized toward the maximum-likelihood bounding objective.

3. Sample generation

With the learned transition step $p_\varphi(z_{t-1}|z_t;t,\theta)$ at hand, there are several ways of generating new samples. The simplest is perhaps through ancestral sampling: (1) discretize the interval $[0,T]$ to a chosen number of time steps, (2) sample an initial random noise configuration $z_T \sim \mathcal{N}(0,\mathbb{I})$, and (3) run the reverse diffusion process by iteratively sampling $z_t \sim p_\varphi(z_{t-1}|z_t;t,\theta)$ until we arrive at a sample at $z_0 \equiv x \sim \hat{p}(x|\theta)$. We explicitly reinstate $\{t,\theta\}$ dependence here to emphasize that the transition distribution (through the noise-prediction network) is conditioned on the diffusion time t and cosmological parameters $\theta = \{\Omega_m,\sigma_8\}$. An example of a sampled point cloud starting from a 3D Gaussian sample is shown in Fig. 1, along with the intermediate states. The sampled trajectories for a subset of particles are shown via the gray lines. See Appendix A for an alternative deterministic sampling method based on ordinary differential equations (ODEs) that produce smoother trajectories. In Appendix A, we also show how these smoother trajectories vary with a conditioning parameter, Ω_m, to produce more or less clustered point clouds.

4. Likelihood evaluation

The diffusion model is trained using a stochastic estimate of the variational maximum-likelihood objective, Eqs. (4) and (5). The same expression can be used to obtain an estimator of the conditional log-likelihood $\log\hat{p}(x|\theta)$, ensuring that the ELBO is evaluated a sufficient number of times to obtain a good estimate of the expectation value. In more detail, the interval $[0,T]$ is discretized into time steps, and we iteratively draw $z_t \sim q(z_t|z_{t-1},\theta)$ starting from $z_0 = x$, compute the diffusion loss terms in Eq. (7), which are summed up and added to the prior and reconstruction terms in Eq. (4).

D. The score-/noise-prediction models

The noise-prediction function $\hat{\epsilon}_\varphi(z_t,t):\mathbb{R}^{N_{\text{gal}}\times N_f}\to\mathbb{R}^{N_{\text{gal}}\times N_f}$, where N_{gal} denotes the number of tracers and

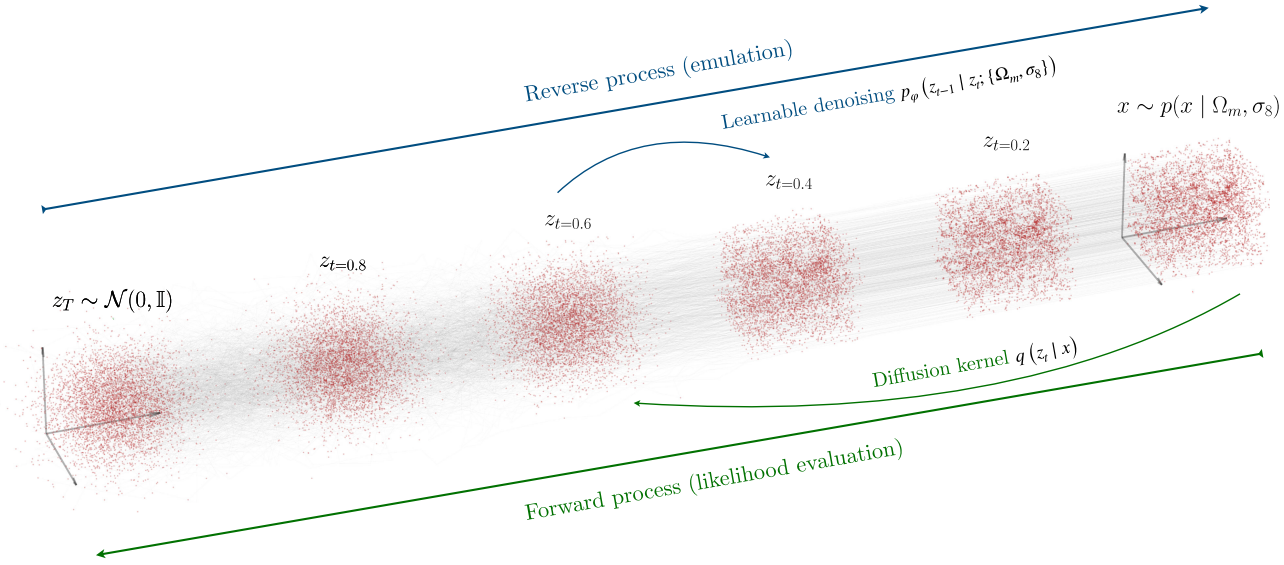

FIG. 1. A schematic overview of the point cloud diffusion model, showing samples from the diffusion process at different diffusion times. During training, noise is added to a data sample x using the diffusion kernel $q(z_t|x)$ and a denoising distribution $p_\varphi(z_{t-1}|z_t)$ is learned. To generate samples, we simulate the reverse process—we sample noise from a standard Gaussian distribution and denoise it iteratively using the learned denoising distribution.

N_f the number of features per tracer, is a crucial part of the diffusion model and must be chosen sensitive to the data modality and generating process. In our case, we model the distribution of tracers and their properties as a "point cloud," i.e., a collection of coordinates (positions), optionally with attached attributes (e.g., velocities), $p\left(\{\vec{r}_i; [\vec{v}_i, \ldots]\}_{i=1}^{N_{\rm gal}} | \theta\right)$. The number of features is either $N_f = 3$ when only modeling tracer coordinates, or $N_f = 7$ when additionally modeling velocities and masses.

The score model must be (1) equivariant to permutations, (2) able to process points of arbitrary cardinality, and (3) able to effectively model the joint correlation structure of galaxy/halo properties. Variants of the closely related transformer and graph neural network (GNN) families satisfy these requirements; here, we show applications using both, described below. All models are implemented using the JAX [42] framework.

1. Graph neural network model

We use a variant of the graph-convolutional network from Battaglia *et al.* [43]. A local k-nearest-neighbors graph with $k = 20$ is constructed using the Euclidean distance between coordinates as the distance metric, accounting for periodic boundary conditions, at each time step in the diffusion process. The relative 3D distances between input node coordinates are used as input graph edge features. The time step t is projected onto a 16-dimensional space via sinusoidal encodings, and the conditioning parameters $\theta = \{\Omega_m, \sigma_8\}$ are linearly projected also onto a 16-dimensional parameter space. They are concatenated to form the global conditioning vector g^0. Both the input node features z_t and edge features $z_{t,i}^{\rm pos} - z_{t,j}^{\rm pos}$ are initially projected into a 16-dimensional latent space via a four-layer multilayer perceptron (MLP; fully connected neural network) with 128 hidden features and Gaussian error linear unit (GELU) activations. All MLPs utilized in the GNN have these same attributes.

Four message-passing rounds are performed, updating the edge attributes e_{ij} at each round by passing a difference of the sender and receiver node attributes, edge attributes, as well as global parameters (a combination of time step embedding and conditioning parameters $\{\Omega_m, \sigma_8\}$) through an MLP. For each node h_i, the neighboring edge attributes are aggregated, concatenated with the node and global attributes, and passed through another MLP to obtain the residual of the updated node features. Featurewise layer normalization [44] is applied after each layer.

The graph-convolutional layers are defined as

$$e_{ij}^{l+1} = \phi_e^l(\text{Concat}[h_i^l - h_j^l, e_{ij}^l, g^0]) \tag{10}$$

$$h_i^{l+1} = h_i^l + \phi_h^l\left(\text{Concat}\left[h_i^l, \sum_{j \in \mathcal{N}(i)_{/i}} e_{ij}^{l+1}, g^0\right]\right) \tag{11}$$

where the edge- and node-update neural networks ϕ_e^l and ϕ_h^l are both MLPs. $\mathcal{N}(i)$ denotes the set of nodes connected to node (i) by an edge. Each edge and node update is additionally parametrized by the global (diffusion time and conditioning) parameters g^0. Finally, the latent node features are projected back onto N_f dimensions via an MLP.

The GNN model integrates an attention mechanism to selectively emphasize relevant features in the graph when updating the edge features—attention scores are computed for each edge and used to scale the edge features. First, for each edge $(i,\ j)$ connecting nodes i and j, an attention logit is calculated using an MLP ϕ_a,

$$l_{ij}^l = \phi_a(\text{Concat}[h_i^l - h_j^l, e_{ij}^l, g^0]). \tag{12}$$

These logits are then normalized across all neighboring edges via sofmax, ensuring that the attention scores for edges emanating from a single node sum to one. For a node i, the attention weight α_{ij} for each edge is given by $\alpha_{ij} = \exp(l_{ij})/\left(\sum_{k\in\mathcal{N}(i)}\exp(l_{ik})\right)$. The edge features are then scaled by these attention weights, $e'_{ij} = e_{ij}\cdot\alpha_{ij}$, resulting in attention-modified edge features which are used subsequently in Eq. (10).

The graph neural network was implemented using the JRAPH [45] package and contains 637,373 trainable parameters.

2. Transformer model

The transformer [46,47] is a sequence-to-sequence model that uses self-attention to process sequences of arbitrary length. We use a encoder-only transformer without positional encodings or causal masking, which makes the model permutation equivariant and able to deal with set-valued data. The input coordinates z_t are linearly projected onto an embedding space of dimension 256, then processed through four transformer layers each consisting of multihead self-attention with four heads and a two-layer MLP of hidden dimension 1024 with GELU activations. A "prelayer" norm configuration, where features are normalized each time the transformer residual stream is read from, was found to be crucial for training stability [48]. The final output is projected down to the dimensionality of the input attributes. In order to condition the score model on time step t as well as the parameters of interest $\theta = \{\Omega_m, \sigma_8\}$, a linear projection of the combined conditioning vector g^0 (described in the GNN model above) is added to the input embeddings. The transformer score model contains 4,776,281 trainable parameters.

Self-attention scales quadratically $\mathcal{O}(N_{\text{gal}}^2)$ with the number of input points, in principle limiting the applicability of this architecture to larger point clouds. For set-valued data, however, a specified number N_{ind} of representative "inducing" points can be learned also via attention—essentially an on-the-fly learned clustering [49]. These are then used for computing the keys and values in the attention mechanism, with the input points projected onto queries, scaling the computation linearly $\mathcal{O}(N_{\text{gal}}\cdot N_{\text{ind}})$ with the cardinality of the point cloud. We implement and test induced attention in our code, achieving similar performance to full attention, but did not find it necessary for computational tractability with our 5000-cardinal point cloud.

III. RESULTS AND DISCUSSION

A. Training

The model is trained using the variational maximum-likelihood objective in Eq. (4). We run 300,000 iterations of the AdamW [50,51] optimizer with peak learning rate 3×10^{-4}, 5000 linear warmup steps, cosine annealing, and a batch size of 16. Boxes are randomly translated and rotated, sensitive to periodic boundary conditions, as a form of data augmentation. We select the checkpoint used downstream as the one with the smallest KL divergence between two-point correlation functions of generated samples and those from the held-out validation set. Further details are provided in Appendix B. We train models on either (1) halo positions only or (2) halo positions, velocities, and masses. Training takes about 12 h on 4 Nvidia A100 GPUs.

B. Conditional sampling

Sampling a point cloud takes ∼5 sec on a single Nvidia A100 GPU using 1000 time steps, scaling sublinearly via vectorization when sampling batches. We show examples of position-only samples from our diffusion model, with a GNN backbone, in the top row of Fig. 2, with the conditioning cosmological parameters $\{\Omega_m, \sigma_8\}$ annotated. Boxes from the test set corresponding to these parameters are shown in the bottom row. Generated boxes are drawn from the same random seed in order to emphasize the effect of parameter conditioning. We see clear signs of clustered structure, with the expected dependence as the cosmological parameters are varied. Although the overall clustering of dark matter particles increases with increasing Ω_m, we here select the most massive 5000 dark matter haloes in each cosmology. In cosmologies with a lower value of Ω_m this selection will lead to smaller mass objects that tend to be near each other, as opposed to big clusters more separated in space for a larger Ω_m cosmology.

A sample from the diffusion model trained on positions, velocities, and masses with the transformer backbone is shown in Fig. 3 (left) to be compared with a box from the test simulation suite with the same underlying cosmology (right). Velocity directions and magnitudes are indicated with gray attached arrows, and the size of the marker is proportional to the mass of the galaxy. Again, clear signs of clustered structure are visible.

C. Summary statistics validation

We verify the quality of the trained generative model, including the dependence on cosmological parameters, by comparing the summary statistics obtained from the generated point clouds with those from a held-out test set.

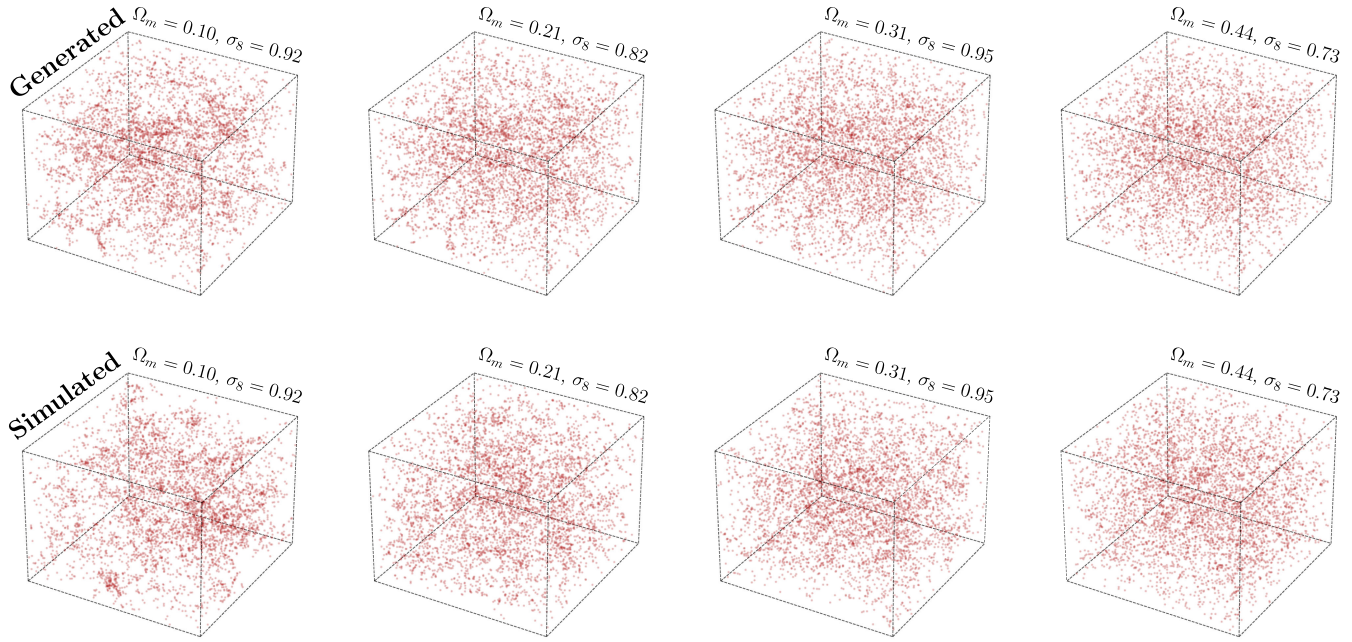

FIG. 2. Examples of point clouds generated from the trained position-only diffusion model (top row) and those from the test set (bottom row), with each column corresponding to the same set of cosmological parameters, indicated. The generated point clouds are drawn from the same random seeds.

In Fig. 4, we evaluate the positions only model trained with the GNN score model described in Sec. II D 1. We compare the parameter dependence of two widely used clustering statistics: the two-point correlation function and the cumulative distribution of k-nearest neighbors (k-NNs), evaluated at five different parameter values of the test set that have been chosen to span the Ω_m parameter space. We compare the mean and variance of 20 diffusion samples to one sample from the N-body. Overall the diffusion model reproduces the trends of the N-body simulations, although due to the lack of varying seeds in the initial conditions for the N-body simulations at varying cosmological parameters we cannot provide a robust quantitative evaluation. Note that the N-body samples also vary other cosmological parameters that are implicity marginalized over in the generated samples, namely, Ω_b, h, and n_s. Hence we do not expect a perfect agreement between the two.

In Fig. 5, we show the corresponding evaluation for the model trained to reproduce the joint probability distribution of halo positions, masses, and velocities, trained with the transformer score model described in Sec. II D 2. Here, we show the mean pairwise velocity distribution (left), to demonstrate that the model describes the joint distribution of velocities and positions faithfully, as well as the

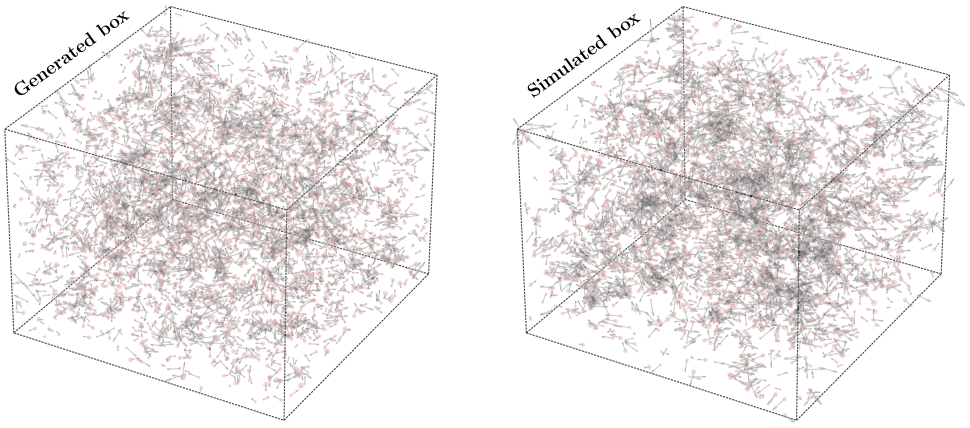

FIG. 3. Example of a point cloud generated from the diffusion model trained on positions, velocities, and masses (top row) and those from the test set (bottom row). Gray arrows correspond to the position and relative magnitude of velocities, and the size of the individual points is proportional to the masses of the galaxies.

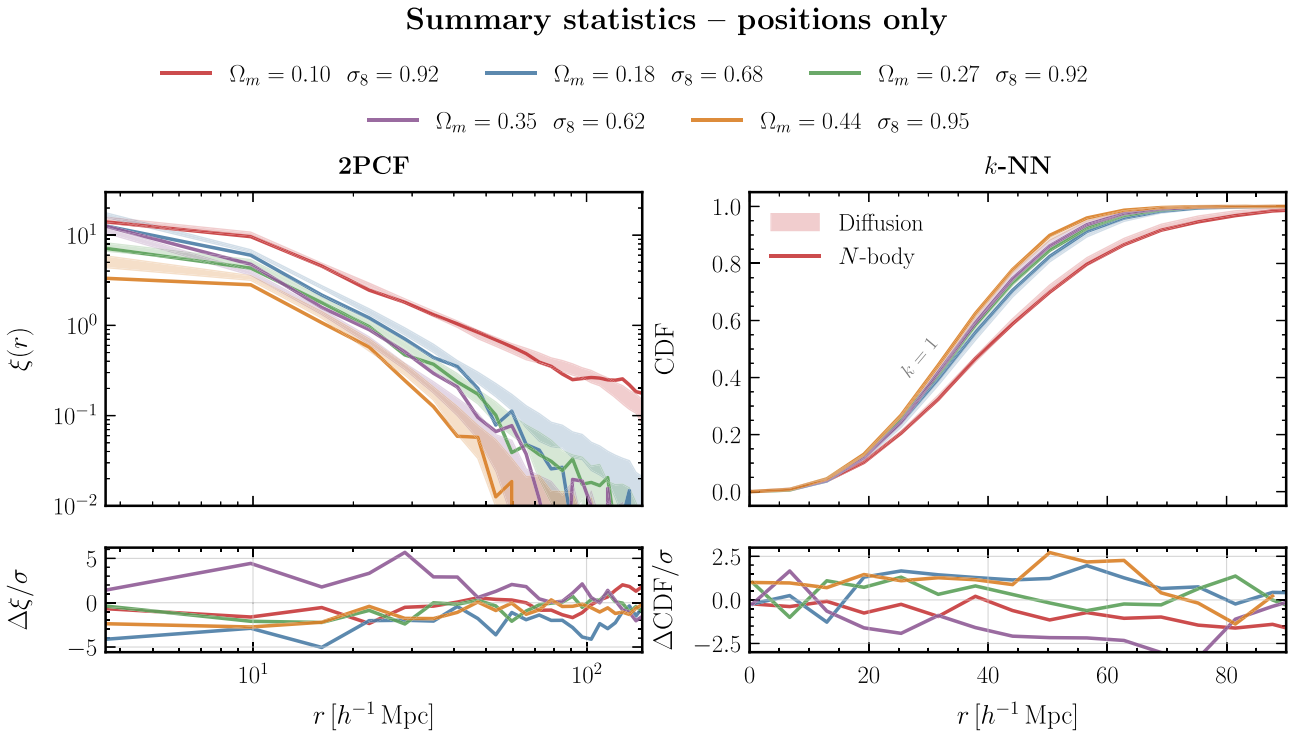

FIG. 4. Summary statistics of the samples generated by the diffusion model compared to those of the N-body simulations for five equally spaced Ω_m values from the test set. For each cosmology, all summary statistics are computed for the same emulated point cloud. Lines are samples from the N-body simulations with different initial conditions, solid contours represent the mean and variance of 20 samples from the diffusion model at that parameter value. On the left, we show the halo two-point correlation function. On the right, the cumulative density function for finding a first neighbor at a given distance from a random point in the simulation volume. Lower panels show the difference between the N-body and the mean of the diffusion samples, in units of the diffusion samples' standard deviation.

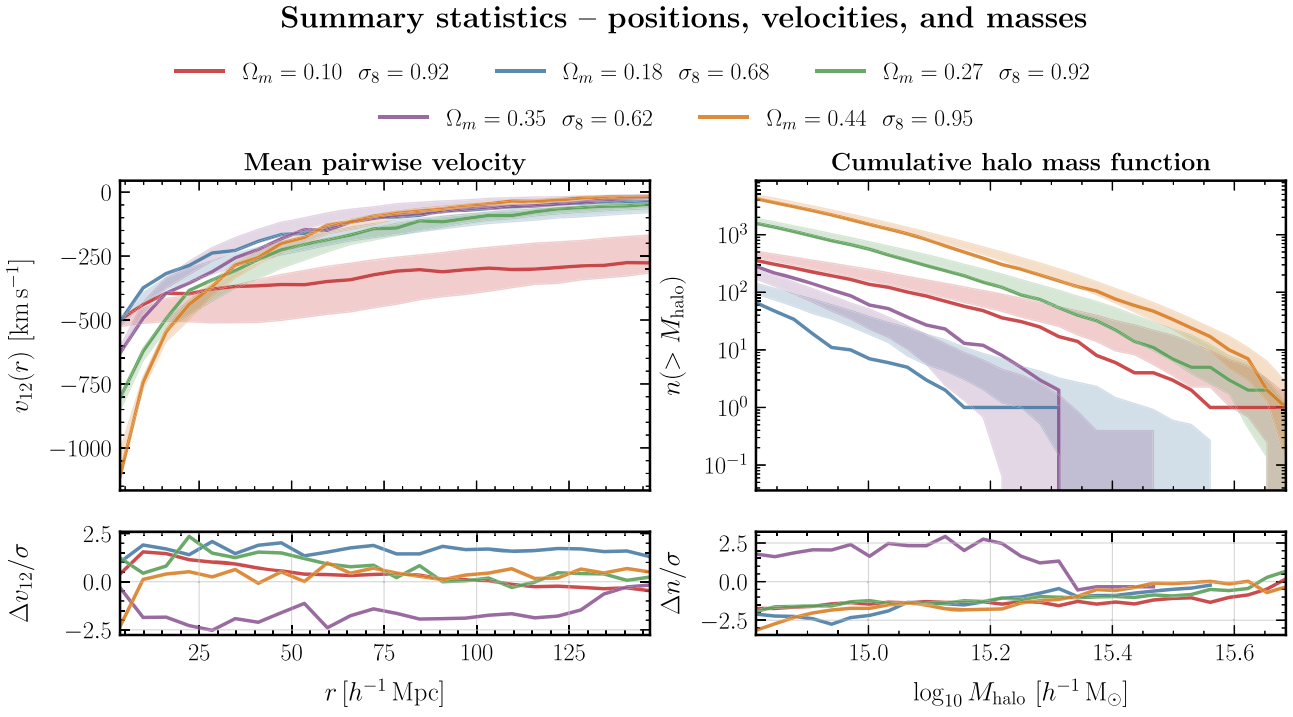

FIG. 5. Velocity (left) and mass (right) summary statistics of the samples generated by the diffusion model compared to those of the N-body simulations for five equally spaced Ω_m values from the test set. On the left, we show the mean pairwise velocity as a function of pair separation. On the right, the cumulative halo mass function.

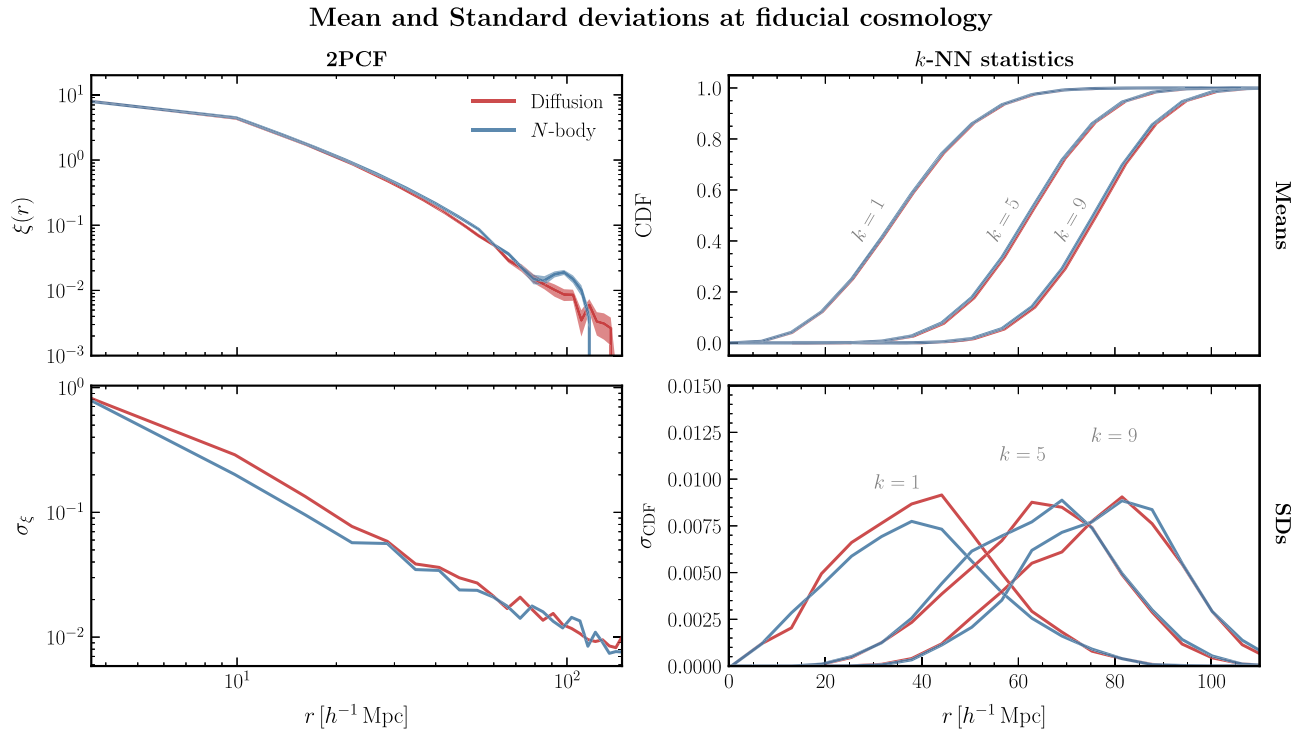

FIG. 6. Mean and variances of the diffusion model and the N-body simulations for the two-point correlation function and the nearest-neighbor statistics at the fiducial parameter values. Blue contours show the mean and variance of the QUIJOTE simulations, whereas red contours show the mean and variance of the diffusion samples at the same parameter values. The upper row shows a comparison of the mean, whereas the lower row shows differences in the standard deviation of the statistics as a function of scale.

cumulative halo mass function (right). We find that the parameter dependence of the mean pairwise velocity can be reproduced over a wide range of scales, whereas the cumulative halo mass function seems to be slightly offset.

Finally, to assess the model's ability to model cosmic variance, we compare the mean and variance of 50 diffusion samples' summary statistics to those of 50 N-body simulations at the fiducial values, shown in Table I. In the first row of Fig. 6, we compare the mean of the 2PCF and of the k-nearest-neighbor statistics for different k values, $k = 1$, 5, 9. Although the model can reproduce the k-NN statistics well, it cannot accurately capture the behavior of the 2PCF at the baryon acoustic oscillation (BAO) scale ($\sim 120h^{-1}$ Mpc). In the second row, we demonstrate that the model can indeed recover the standard deviation of the different summary statistics as a function of scale due to varying initial conditions for both the 2PCF and k-NN statistics. As already mentioned, the diffusion model is only conditioned on Ω_m and σ_8 and therefore is not expected to reproduce the fiducial distributions perfectly.

D. Likelihood calculation

Figure 7 shows 1-σ intervals on the parameters $\{\Omega_m, \sigma_8\}$ for samples from the held-out test set computed using the diffusion-backed approximate log-likelihood $\log \hat{p}(x|\theta)$ for the positions-only model (black data points) and model with all features (red data points). These are computed by varying the dependent parameter on a 1D grid, while keeping the other parameter fixed at the ground-truth value. The learned ELBO in Eq. (4) is evaluated 32 times with 50 discretization time steps $\in [0, T = 1]$ to obtain an estimate of the conditional log-likelihood, which takes $\sim$10 sec when vectorized over the number of evaluations. Additional results substantiating these choices are shown in Appendix C.

Although the points follow the expected trend qualitatively, it is clear that the learned likelihoods are not well calibrated. In particular, they are seen to be overconfident for Ω_m.

This is perhaps not surprising—although we see from Fig. 4 that the model has learned qualitatively good variation across cosmological parameters in the space of tested summaries, 1800 samples is likely too small a training dataset from which to robustly learn this parameter dependence at the field level, even with aggressive data augmentation. It can be seen from Fig. 6 that the generated samples also struggle to reproduce the two-point correlation function at larger scales, suggesting that the large-scale correlation structure, which influences in particular the Ω_m dependence, is not optimally captured.

Given this, we do not rigorously evaluate posteriors distributions on cosmological parameters or compare them

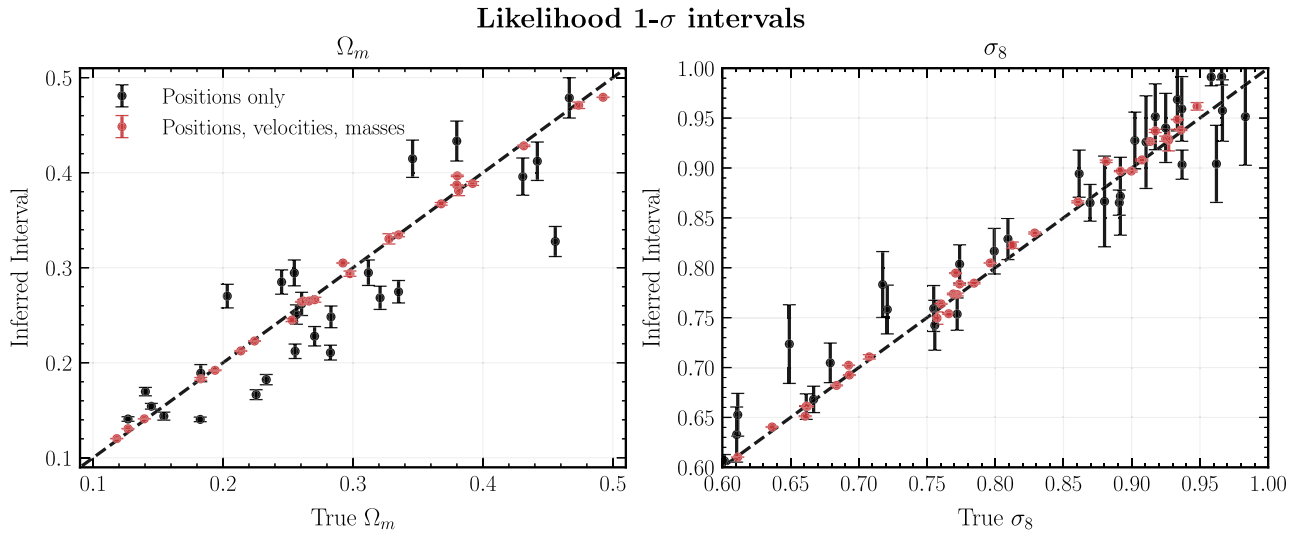

FIG. 7. Intervals corresponding to 1-σ containment from the likelihood profiles for Ω_m (left) and σ_8 (right), obtained by fixing one parameter at its true value and evaluating the likelihood over the other. Intervals for the position-only model (black data points) and model with positions, velocities, and masses (red data points) are shown.

to those obtained using two-point correlation function summaries. We leave model improvements toward better-calibrated likelihoods to future work, some of which we discuss in the next section.

IV. LIMITATIONS AND PROSPECTS

The present paper serves as a proof-of-principle exposition of some of the capabilities enabled by generative modeling in the context of galaxy surveys, i.e., field level emulation and inference. Although we demonstrate the ability to emulate cosmological fields at the point cloud level, our GNN and transformer-backed models are not able to achieve well-calibrated likelihoods to a level that would be satisfactory for downstream applications and also show discrepancies between the emulated and simulated point clouds on large scales. The availability of larger datasets for training is likely to partially alleviate this issue. We also outline promising directions on the methodological side for future study that could significantly improve the fidelity of our generative model and enable scaling to a larger number of points.

(i) Periodic boundary conditions: The target point cloud data are confined to a box with periodic boundary conditions. Our model does not account for either the confinement to a box or periodic boundary conditions at the level of the diffusion model (although note that the graph neighbor calculation does take into account distances across box boundaries). Existing methods used for generation of periodic configurations of materials [52,53] could be leveraged in this direction.

(ii) Physical symmetries: Cosmological data typically encode a great deal of physical symmetry—in our case, Euclidean symmetry associated with the freedom to choose an arbitrary coordinate system. Our model is, on the other hand, manifestly coordinate dependent, relying on propagating absolute coordinates. Although we aim to partly mitigate this through data augmentation (i.e., train-time rotations, translations, and reflections), this is significantly less data efficient than directly baking in these symmetries using symmetry-preserving neural networks [54,55], which have been shown to be provably more robust in other domains, e.g., the study of atomistic systems [56–58]. Although translation- and rotation-invariant neural networks have previously been used within astrophysics for parameter prediction tasks via invariant feature propagation [59], end-to-end "equivariance" is expected to benefit our model more since it does not target prediction of globally invariant properties.

(iii) Physically motivated base distributions: Our diffusion model relies on standard Gaussian diffusion, with the asymptotic latent distribution being a standard Gaussian $z_T \sim \mathcal{N}(0, \mathbb{I})$. This means that the model has to denoise the standard Gaussian into a box. Recent classes of deep generative models, e.g., stochastic interpolants and conditional flow matching [60,61], allow for the base distribution to be arbitrary and also implicitly defined through samples. In our case, using a physically motivated initial distribution, e.g., particles in a box distributed according to a fiducial two-point correlation function, can be set up as a potentially easier learning problem.

(iv) Architecture expressivity: Our fiducial score function is a simple message-passing GNN, which suffers from known lack of expressivity, in particular when modeling long-range correlations. A common culprit is oversmoothing—as the number

of message-passing hops increases, the feature neighborhoods become increasingly similar across nodes, leading node features to collapse to similar values [62]. Indeed, our summary two-point correlation function does not faithfully capture correlation features on large spatial scales, such as the BAO peak (Fig. 6). The use of techniques to explicitly mitigate these issue [63,64] and enhance modeling of long-range correlation could therefore be helpful for generative modeling of cosmic fields as point clouds.

(v) Scalability and hierarchical description of galaxy field: Our diffusion model is used to represent the entire input point cloud field, which we choose to consist of 5000 particles. In practice, the field at large scales is expected to be highly linear and Gaussian, which calls for hierarchical methods that can generate fields described by nonlinear statistics on small scales while conforming to consistent linear descriptions on large scales. Designing the diffusion process in a lower-dimensional latent space via hierarchical down- and up-sampling [65], also possible while preserving Euclidean symmetry [66], could be one avenue toward this.

V. CONCLUSIONS

We introduced a diffusion-based generative model that captures the complex, non-Gaussian statistics of the galaxy clustering field along with the underlying cosmology dependence. The model can be efficiently used for emulation of N-body simulations via sampling, $x \sim p(x|\theta)$, as well as evaluation of the conditional likelihood. While the model qualitatively reproduces essential summary statistics associated with the galaxy field, it can struggle to correctly model the point cloud's correlation structure, in particular on larger spatial scales. We discuss the technical limitations of our model in this direction and avenues for further improvement.

The model presented in this work was trained on the dark matter halo distribution generated by N-body simulations. An application to upcoming galaxy clustering datasets, such as DESI, would require building a forward model for the survey that includes: (1) a model of the galaxy-halo connection, (2) observational effects, such as redshift space distortions and the Alcock-Paczynski effect, and (3) survey systematics, such as survey masks and fiber collisions. An example of such a forward model has been presented in SIMBIG [14]. A diffusion model for galaxy clustering trained on such a forward model could provide strong constraints on the standard ΛCDM cosmological model, as well as a means to test the robustness of its constraints through the analysis of posterior samples and likelihood estimates.

The code used for reproducing the results presented in this paper is available from GitHub [67].

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions). This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics of U.S. Department of Energy under Grant No. DE-SC0012567. The computations in this paper were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University.

APPENDIX A: ALTERNATIVE SAMPLING METHOD: PROBABILITY FLOW ODE

With the trained noise-prediction neural network $\hat{\epsilon}_\varphi(z_t, t)$ at hand, several sampling techniques other than the ancestral sampling used in the main text are possible. Recall from Sec. II B that the noise-prediction network is equal to the local conditional score of the data distribution, up to a sign and noise schedule-dependent scaling, $\nabla_{z_t} \log p(z_t) = \hat{s}_\varphi(z_t, t) = -\hat{\epsilon}_\varphi(z_t, t)/\sigma_t$.

In the continuous-time, stochastic differential equation (SDE) formulation [16], taking the time step discretization to the continuum limit $\delta_t \to 0$, the variance-preserving forward diffusion process in Eq. (1) can be written as

$$z_t = \sqrt{1-\beta_t\Delta_t}\, z_{t-1} + \sqrt{\beta_t\Delta_t}\,\epsilon \tag{A1}$$

$$\approx z_{t-1} - \frac{\beta_t\Delta_t}{2} z_{t-1} + \sqrt{\beta_t\Delta_t}\,\epsilon \tag{A2}$$

with $\epsilon \sim \mathcal{N}(0, \mathbb{I})$. This is an update rule corresponding to the Euler-Maruyama discretization of the SDE

$$\mathrm{d}z_t = -\frac{1}{2}\beta(t) z_t \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}w_t \tag{A3}$$

where w_t represents a Wiener process, also known as Brownian motion. This forward SDE has a corresponding *reverse* SDE which gives the same marginal distribution $p(z_t)$ at all times,

$$\mathrm{d}z_t = \left[-\frac{1}{2}\beta(t) z_t - \beta(t)\nabla_{z_t} \log p(z_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}w_t. \tag{A4}$$

This SDE can be solved with any solver and, in the case presented, via simple discretization (Euler-Maruyama method).

Remarkably, there exists a deterministic reversible process, an ODE (ordinary differential equation), whose trajectories share the same marginal densities $p(z_t)$ as this reverse SDE [16,35],

$$\mathrm{d}z_t = -\frac{1}{2}\beta(t)[z_t + \nabla_{z_t} \log p(z_t)]\mathrm{d}t \tag{A5}$$

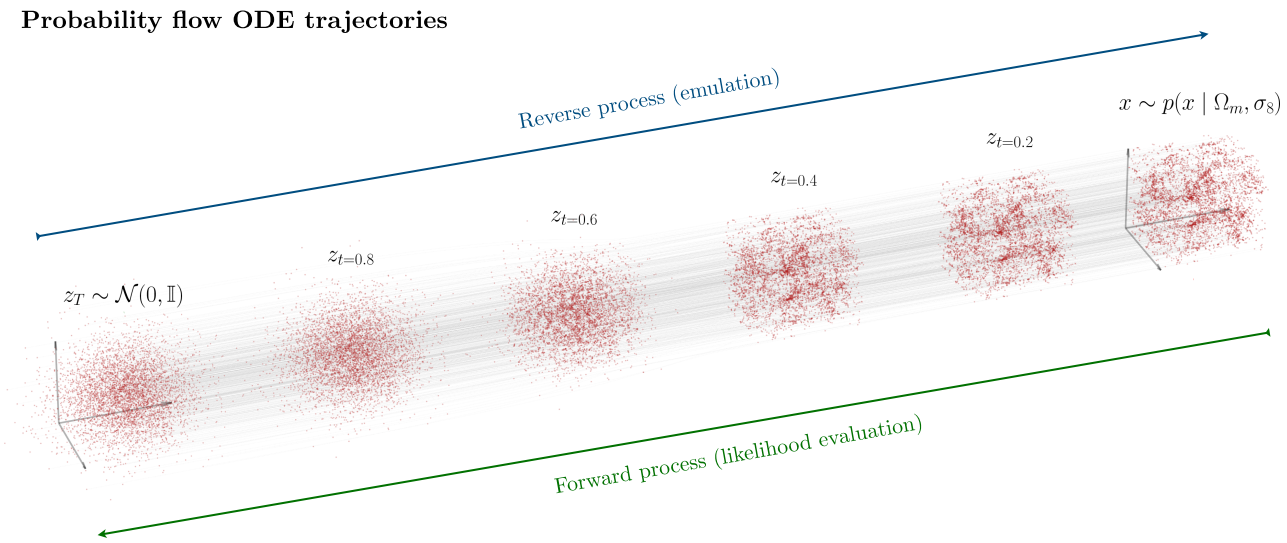

FIG. 8. A sample obtained by solving the probability flow ODE associated with the diffusion SDE. Smooth trajectories from the primal Gaussian (left) to the sampled halo distribution (right) can be seen, in contrast with the stochastic trajectories in Fig. 1.

called the probability flow ODE and which is an instance of a continuous-time normalizing flow [41].

This highlights the fact that diffusion models essentially efficiently train a continuous normalizing flow without requiring simulation the entire forward trajectory. This ODE can be solved using any ODE solver to obtain samples from the data distribution; we show in Fig. 8 a sample and corresponding trajectories for a subset of particles obtained using Heun's second-order method via DIFFRAX [68]. The trajectories can be seen to be smooth, in contrast with those in Fig. 1.

Sampling via solving the probability flow ODE can be used to visualize the dependence of the sampling trajectory on cosmological parameters, unencumbered by the stochastic component. This is shown in Fig. 9, which illustrates 2D projected slices of 400 generated particle coordinates across time steps along with the sampled trajectories for a subset of the particles. Starting from the same initial Gaussian distribution, we condition on different cosmological parameters with $\Omega_m = 0.13$ (red) and $\Omega_m = 0.47$ (blue), keeping fixed $\sigma_8 = 0.8$. Diverging trajectories across diffusion time

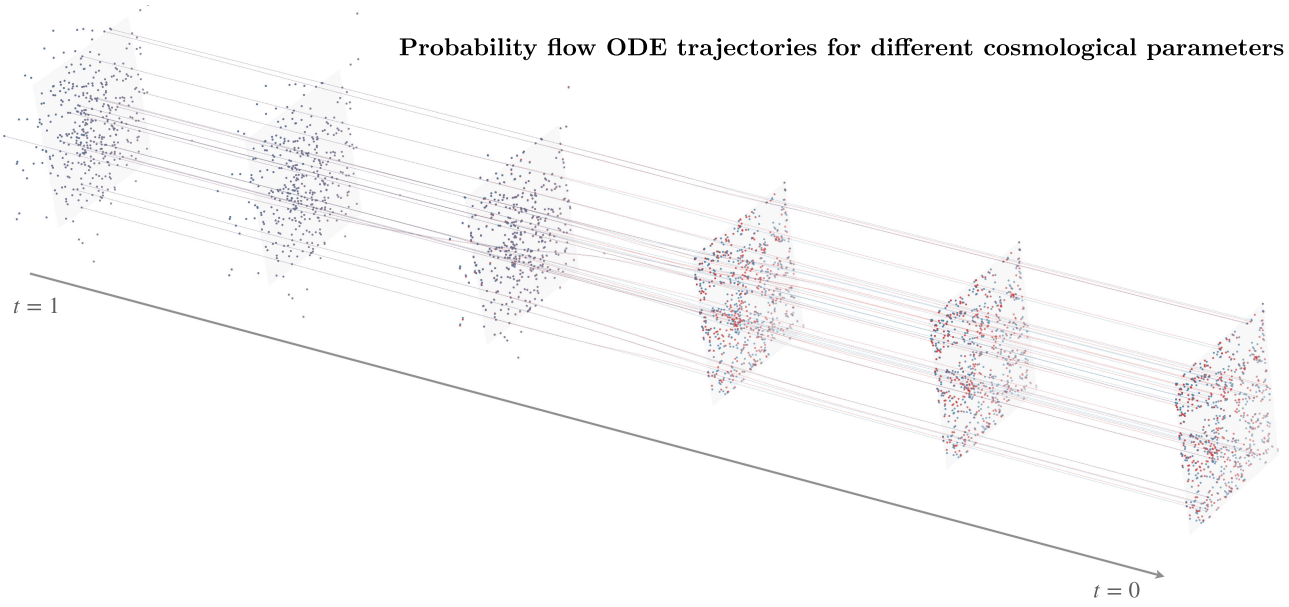

FIG. 9. Probability flow ODE trajectories starting from the same initial Gaussian distribution, conditioned on different cosmological parameters with $\Omega_m = 0.13$ (red) and $\Omega_m = 0.47$ (blue), keeping fixed $\sigma_8 = 0.8$. 2D spatial projections for 400 particles are shown, with trajectories illustrated for a subset of these.

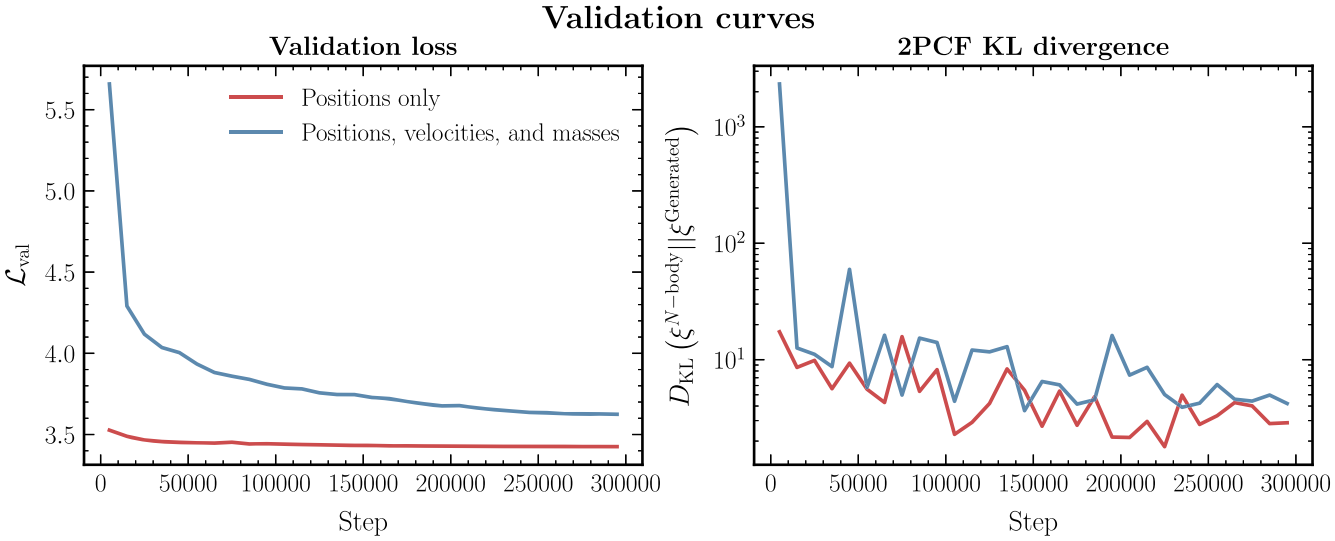

FIG. 10. Validation loss (left) and KL divergence between the true two-point correlation functions and the generated ones (right) as a function of training steps. We show both a graph neural network model trained on halo positions only (red) and a transformer model trained on positions, velocities, and masses (blue).

can be seen, highlighting the parameter dependence of the score model.

APPENDIX B: ADDITIONAL TRAINING DETAILS

In Fig. 10, we show the validation loss curve (evaluated on the held-out test set), together with the KL divergence between the N-body two-point correlation functions on small scales ($r < 55h^{-1}$ Mpc) and the generated ones. The KL divergence is computed assuming that both distributions are Gaussian. We show that a small decrease in the loss value produces a sharp decrease in the KL divergence and therefore a large improvement in the quality of the generated samples. The checkpoint used downstream for each model is chosen as the one with the smallest KL divergence in the validation set.

APPENDIX C: LOG-LIKELIHOOD EVALUATION

The diffusion model is trained by maximizing a stochastic estimate of a variational lower bound on the log-likelihood, the ELBO in Eq. (4). The same expression can be used to compute an estimate of the (conditional) log-likelihood, which we show as delta likelihood profiles in

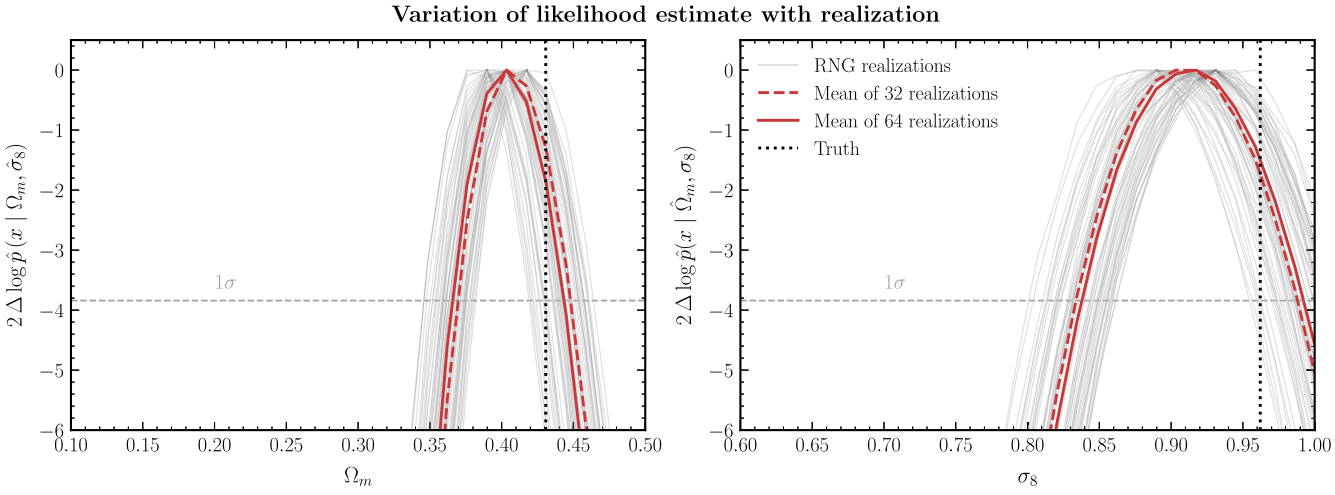

FIG. 11. For a particular sample in the test test, twice the log-likelihood profiles relative to their maximum value, for Ω_m (left) and σ_8 (right). Profiles for different evaluations (gray) as well as averaged over 32 (dashed red) and 64 (solid red) evaluations are shown. Averaging over 32 realizations is seen to give a converged estimate of the conditional likelihood (RNG refers to random number generated).

Fig. 11 when averaged over 32 (dashed red) and 64 (solid red) evaluations for Ω_m (left) and σ_8 (right). Fifty discretization steps are used; we found quantitatively similar results for the mean profiles using a larger number of steps. Different evaluations are shown as gray lines, demonstrating that the variance of the approximate likelihood with respect to the random seed is high relative to the 1σ interval. It can be seen that the relative log-likelihood has converged with 32 evaluations, which we use to show the likelihood profile results in Fig. 7.

Interestingly, variation on the *absolute* log-likelihood estimate was observed to significantly larger, $\mathcal{O}(100)$, compared to the $\mathcal{O}(1)$ variation on the relative conditional log-likelihood shown. An estimate of the raw log-likelihood would therefore require a larger number of evaluations in this case.

[1] S.-F. Chen, H. Lee, and C. Dvorkin, Precise and accurate cosmology with CMB × LSS power spectra and bispectra, J. Cosmol. Astropart. Phys. 05 (2021) 030.

[2] D. Gualdi, H. Gil-Marín, and L. Verde, Joint analysis of anisotropic power spectrum, bispectrum and trispectrum: Application to N-body simulations, J. Cosmol. Astropart. Phys. 07 (2021) 008.

[3] S. Cheng, Y.-S. Ting, B. Ménard, and J. Bruna, A new approach to observational cosmology using the scattering transform, Mon. Not. R. Astron. Soc. **499**, 5902 (2020).

[4] G. Valogiannis and C. Dvorkin, Towards an optimal estimation of cosmological parameters with the wavelet scattering transform, Phys. Rev. D **105**, 103534 (2022).

[5] G. Valogiannis and C. Dvorkin, Going beyond the galaxy power spectrum: An analysis of BOSS data with wavelet scattering transforms, Phys. Rev. D **106**, 103509 (2022).

[6] E. Paillas, C. Cuesta-Lazaro, W. J. Percival, S. Nadathur, Y.-C. Cai, S. Yuan, F. Beutler, A. de Mattia, D. Eisenstein, D. Forero-Sanchez, N. Padilla, M. Pinon, V. Ruhlmann-Kleider, A. G. Sánchez, G. Valogiannis, and P. Zarrouk, Cosmological constraints from density-split clustering in the BOSS CMASS galaxy sample, arXiv:2309.16541.

[7] C. Uhlemann, O. Friedrich, A. Boyle, A. Gough, A. Barthelemy, F. Bernardeau, and S. Codis, It takes two to know one: Computing accurate one-point PDF covariances from effective two-point PDF models, Open J. Astrophys. **6**, 1 (2023).

[8] A. Pisani *et al.*, Cosmic voids: A novel probe to shed light on our Universe, Bull. Am. Astron. Soc. **51**, 40 (2019).

[9] A. J. Hawken, M. Aubert, A. Pisani, M.-C. Cousinou, S. Escoffier, S. Nadathur, G. Rossi, and D. P. Schneider, Constraints on the growth of structure around cosmic voids in eBOSS DR14, J. Cosmol. Astropart. Phys. 06 (2020) 012.

[10] A. Banerjee and T. Abel, Nearest neighbour distributions: New statistical measures for cosmological clustering, Mon. Not. R. Astron. Soc. **500**, 5479–5499 (2020).

[11] G. Valogiannis, S. Yuan, and C. Dvorkin, Precise cosmological constraints from BOSS galaxy clustering with a simulation-based emulator of the wavelet scattering transform, Phys. Rev. D **109**, 103503 (2024).

[12] B. Dai and U. Seljak, Translation and rotation equivariant normalizing flow (TRENF) for optimal cosmological analysis, Mon. Not. R. Astron. Soc. **516**, 2363 (2022).

[13] T. L. Makinen, T. Charnock, P. Lemos, N. Porqueres, A. F. Heavens, and B. D. Wandelt, The cosmic graph: Optimal information extraction from large-scale structure using catalogues, Open J. Astrophys. **5** (2022), 10.21105/astro.2207.05202.

[14] C. Hahn, M. Eickenberg, S. Ho, J. Hou, P. Lemos, E. Massara, C. Modi, A. M. Dizgah, B. R.-S. Blancard, and M. M. Abidi, SimBig: Mock challenge for a forward modeling approach to galaxy clustering, J. Cosmol. Astropart. Phys. 04 (2023) 010.

[15] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. **33**, 6840 (2020).

[16] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations, arXiv:2011.13456.

[17] E. Hoogeboom, V. Garcia Satorras, C. Vignac, and M. Welling, Equivariant diffusion for molecule generation in 3D, arXiv:2203.17003.

[18] J. Ingraham, M. Baranov, Z. Costello, V. Frappier, A. Ismail, S. Tie, W. Wang, V. Xue, F. Obermeyer, A. Beam, and G. Grigoryan, Illuminating protein space with a programmable generative model, bioRxiv (2022), https://www.biorxiv.org/content/early/2022/12/02/2022.12.01.518682.full.pdf, 10.1101/2022.12.01.518682.

[19] J. L. Watson *et al.*, Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models, bioRxiv (2022), https://www.biorxiv.org/content/early/2022/12/14/2022.12.09.519842.full.pdf, 10.1101/2022.12.09.519842.

[20] S. Alamdari, N. Thakkar, R. van den Berg, A. X. Lu, N. Fusi, A. P. Amini, and K. K. Yang, Protein generation with evolutionary diffusion: Sequence is all you need, bioRxiv (2023), https://www.biorxiv.org/content/early/2023/09/12/2023.09.11.556673.full.pdf, 10.1101/2023.09.11.556673.

[21] V. Mikuni, B. Nachman, and M. Pettee, Fast point cloud generation with diffusion models in high energy physics, Phys. Rev. D **108**, 036025 (2023).

[22] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch, and T. Golling, PC-JeDi: Diffusion for particle cloud generation in high energy physics, SciPost Phys. **16**, 018 (2024).

[23] E. Buhmann, C. Ewen, D. A. Faroughy, T. Golling, G. Kasieczka, M. Leigh, G. Quétant, J. A. Raine, D. Sengupta, and D. Shih, EPiC-ly fast particle cloud generation with flow-matching and diffusion, arXiv:2310.00049.

[24] N. Mudur and D. P. Finkbeiner, Can denoising diffusion probabilistic models generate realistic astrophysical fields?, arXiv:2211.12444.
[25] R. Legin, M. Ho, P. Lemos, L. Perreault-Levasseur, S. Ho, Y. Hezaveh, and B. Wandelt, Posterior sampling of the initial conditions of the universe from non-linear large scale structures using score-based generative models, Mon. Not. R. Astron. Soc. **527**, L173 (2023).
[26] B. Remy, F. Lanusse, N. Jeffrey, J. Liu, J.-L. Starck, K. Osato, and T. Schrabback, Probabilistic mass mapping with neural score estimation, Astron. Astrophys. **672**, A51 (2023).
[27] F. Lanusse, R. Mandelbaum, S. Ravanbakhsh, C.-L. Li, P. Freeman, and B. Póczos, Deep generative models for galaxy image simulations, Mon. Not. R. Astron. Soc. **504**, 5543 (2021).
[28] M. J. Smith, J. E. Geach, R. A. Jackson, N. Arora, C. Stone, and S. Courteau, Realistic galaxy image simulation via score-based generative models, Mon. Not. R. Astron. Soc. **511**, 1808 (2022).
[29] A. Adam, A. Coogan, N. Malkin, R. Legin, L. Perreault-Levasseur, Y. Hezaveh, and Y. Bengio, Posterior samples of source galaxies in strong gravitational lenses with score-based priors, arXiv:2211.03812.
[30] R. Legin, A. Adam, Y. Hezaveh, and L. P. Levasseur, Beyond Gaussian noise: A generalized approach to likelihood analysis with non-Gaussian noise, Astrophys. J. Lett. **949**, L41 (2023).
[31] F. Villaescusa-Navarro *et al.*, The Quijote simulations, Astrophys. J. Suppl. Ser. **250**, 2 (2020).
[32] M. Davis, G. Efstathiou, C. S. Frenk, and S. D. M. White, The evolution of large-scale structure in a universe dominated by cold dark matter, Astrophys. J. **292**, 371 (1985).
[33] T. Salimans and J. Ho, Progressive distillation for fast sampling of diffusion models, arXiv:2202.00512.
[34] A. Vahdat, K. Kreis, and J. Kautz, Score-based generative modeling in latent space, Adv. Neural Inf. Process. Syst. **34**, 11287 (2021).
[35] Y. Song, C. Durkan, I. Murray, and S. Ermon, Maximum likelihood training of score-based diffusion models, Adv. Neural Inf. Process. Syst. **34**, 1415 (2021).
[36] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, Variational diffusion models, arXiv:2107.00630.
[37] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, arXiv:1312.6114.
[38] C. Luo, Understanding diffusion models: A unified perspective, arXiv:2208.11970.
[39] T. Chen, On the importance of noise scheduling for diffusion models, arXiv:2301.10972.
[40] D. P. Kingma and R. Gao, Understanding diffusion objectives as the ELBO with simple data augmentation, arXiv:2303.00848.
[41] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, FFJORD: Free-form continuous dynamics for scalable reversible generative models, arXiv:1810.01367.
[42] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, *JAX: Composable transformations of PYTHON+NUMPY programs* (2018).
[43] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, Relational inductive biases, deep learning, and graph networks, arXiv:1806.01261.
[44] J. Lei Ba, J. R. Kiros, and G. E. Hinton, Layer normalization, arXiv:1607.06450.
[45] J. Godwin, T. Keck, P. Battaglia, V. Bapst, T. Kipf, Y. Li, K. Stachenfeld, P. Veličković, and A. Sanchez-Gonzalez, *Jraph: A library for graph neural networks in JAX* (2020).
[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, arXiv:1706.03762.
[47] R. E. Turner, An introduction to transformers, arXiv:2304.10557.
[48] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu, On layer normalization in the transformer architecture, arXiv:2002.04745.
[49] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. Whye Teh, Set transformer: A framework for attention-based permutation-invariant neural networks, arXiv:1810.00825.
[50] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 2019* (2019).
[51] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2015).
[52] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, Crystal diffusion variational autoencoder for periodic material generation, arXiv:2110.06197.
[53] Y. Luo, C. Liu, and S. Ji, Towards symmetry-aware generation of periodic materials, arXiv:2307.02707.
[54] M. Geiger and T. Smidt, e3nn: Euclidean neural networks, arXiv:2207.09453.
[55] V. Garcia Satorras, E. Hoogeboom, and M. Welling, E(n) equivariant graph neural networks, arXiv:2102.09844.
[56] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nat. Commun. **13**, 2453 (2022).
[57] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, Learning local equivariant representations for large-scale atomistic dynamics, Nat. Commun. **14**, 579 (2023).
[58] I. Batatia, D. Péter Kovács, G. N. C. Simm, C. Ortner, and G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, arXiv:2206.07697.
[59] T. L. Makinen, T. Charnock, P. Lemos, N. Porqueres, A. F. Heavens, and B. D. Wandelt, The Cosmic graph: Optimal information extraction from large-scale structure using catalogues, Open J. Astrophys. **5**, 18 (2022).
[60] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden, Stochastic interpolants: A unifying framework for flows and diffusions, arXiv:2303.08797.
[61] A. Tong, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, K. Fatras, G. Wolf, and Y. Bengio, Improving

and generalizing flow-based generative models with minibatch optimal transport, arXiv:2302.00482.
[62] T. Konstantin Rusch, M. M. Bronstein, and S. Mishra, A survey on oversmoothing in graph neural networks, arXiv:2303.10993.
[63] L. Zhao and L. Akoglu, PairNorm: Tackling oversmoothing in GNNs, arXiv:1909.12223.
[64] J. Tönshoff, M. Ritzert, E. Rosenbluth, and M. Grohe, Where did the gap go? Reassessing the long-range graph benchmark, arXiv:2309.00367.
[65] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, arXiv:1706.02413.
[66] C. Fu, K. Yan, L. Wang, W. Y. Au, M. McThrow, T. Komikado, K. Maruhashi, K. Uchino, X. Qian, and S. Ji, A latent diffusion model for protein structure generation, arXiv:2305.04120.
[67] https://github.com/smsharma/point-cloud-galaxy-diffusion.
[68] P. Kidger, On neural differential equations, Ph.D. thesis, University of Oxford, 2021.