Diffusion generative modeling for galaxy surveys: emulating clustering for inference at the field level

Carolina Cuesta-Lazaro [*1 2 3] **Siddharth Mishra-Sharma** [*1 4 5]

Abstract

We introduce a diffusion-generative model to describe the distribution of galaxies in our Universe directly as a collection of points in 3-D space, without resorting to binning or voxelization. The custom diffusion model, which employs graph neural networks as the backbone score function, can be used as an emulator that accurately reproduces essential summary statistics of the galaxy distribution and enables cosmological parameter estimation using gradient-based inference techniques. This approach allows for a comprehensive analysis of cosmological data by circumventing limitations inherent to summary statistics-based as well as likelihood-free methods.

1. 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, amongst others, the nonlinear collapse of structures, baryonic processes, and the galaxy-halo connection. 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. While routinely used in cosmological analyses, the 2PCF is not a complete (sufficient) summary of the galaxy clustering signal, and higher-order statistics are required to fully capture the information content of the clustering signal.

Machine learning methods have demonstrated the potential to significantly impact how cosmological data is analyzed, and galaxy clustering is no exception (Makinen et al., 2022; Hahn et al., 2023). 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 estimator of the likelihood of an observed galaxy configuration x given some parametric description θ of the data, $p(x \mid \theta)$ – a *generative model*. Access to the conditional likelihood can be use to sample different field configurations, $x \sim p(x \mid \theta)$, for use in various downstream tasks or as a surrogate model (*emulation*). Additionally, one can use the likelihood to perform *parameter inference* and hypothesis testing with 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, $p(\theta \mid x) = p(x \mid \theta) \cdot p(\theta)/p(x)$.

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 z, in the case of galaxy clustering one characterizing possible initial conditions and their evolution trajectories towards realizing a given observation x – $p(x \mid \theta) = \int \mathrm{d}z\, p(x, z \mid \theta)$. For a collection of galaxies or dark matter halos, constructing a generative model involves modeling the joint probability distribution of the properties (positions, velocities, etc.) of a large number of galaxies, $p\left(\{x_i\}_{i=1}^{N_{\rm gal}}\right)$ – a formidable task.

Machine learning has revolutionized the field of generative modeling, heralding methods that are able to learn complex

[*]Equal contribution [1]NSF Institute of AI and Fundamental Interactions (IAIFI) [2]Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA [3]Center for Astrophysics — Harvard & Smithsonian, 60 Garden Street, MS-16, Cambridge, MA 02138, USA [4]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA [5]Department of Physics, Harvard University, Cambridge, MA 02138, USA. Correspondence to: Carolina Cuesta-Lazaro <cuestalz@mit.edu>, Siddharth Mishra-Sharma <smsharma@mit.edu>.

ICML 2023 Workshop on Machine Learning for Astrophysics, Honolulu, Hawaii, USA. PMLR 202, 2023.

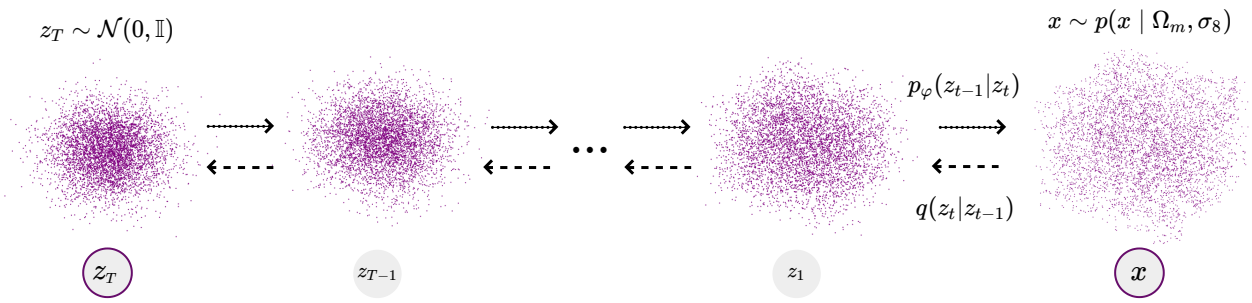

Figure 1. Depiction of the point cloud diffusion model, showing samples from the diffusion process at different time steps. During training, noise is added to a data sample x, that the graph neural network score model learns to denoise. To generate samples, we generate noise from a Gaussian distribution and denoise it iteratively.

data distributions such as those of natural images and human-generated text. Within the sciences, generative modeling has demonstrated potential across domains, showing impressive performance in modeling the distribution of atomistic systems, proteins and biomolecules, and particle jets, to name a few. Much of this success has been enabled through the use of *diffusion models* (Ho et al., 2020; Song et al., 2020) – a class of generative models that learn to denoise a corrupted version of the data by estimating gradients of the data distribution (*score functions*).

Within cosmology, generative modeling has recently been applied in the context of matter density fields (Dai & Seljak, 2022), weak lensing mass maps (Remy et al., 2022), galaxy images (Lanusse et al., 2021; Smith et al., 2022), and strong lensing observations (Adam et al., 2022; Legin et al., 2023). In all cases, the common data modality of 2-D pixelized images is used. While the image representation is appropriate in many cases, the distribution of galaxies is, arguably, ideally represented as a *point cloud* – a set of points in 3-D space, with additional attributes (e.g., velocities, as well as other galaxy properties) attached to them. Pixelization or voxelization necessarily introduces information loss and hyperparameter choices, precluding a full *in-situ* analysis of the 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 as a backbone, faithfully reproduces crucial summary properties of the galaxy field. Furthermore, we show how our model can be used for likelihood-based inference using differentiable optimization techniques.

2. Methodology

We describe, in turn, the diffusion model framework employed, the score neural network, and the likelihood evaluation and parameter inference procedures.

2.1. Diffusion generative modeling

Diffusion models admit several closely related formulations. In one common framing, a neural network $\hat{\epsilon}_\varphi(z_t, t)$ learns to iteratively "de-noise" a corrupted version z_t of the data x_0 from a timestep $t \in [0, T]$ by predicting either the added noise ϵ or the original data point directly. New samples can then be generated by sampling Gaussian random noise z_T and iteratively de-noising it from $t = T$ to $t = 0$. A complementary framing relies on having a neural network $\hat{s}_\varphi(z_t, t)$ estimate the timestep-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 corruption process, $p(z_t) = \mathcal{N}(z_t; x_0, \sigma_t^2)$, the score can be analytically expressed as $\nabla_{z_t} \log p(z_t) = (x_0 - z_t)/\sigma_t^2 = -\epsilon/\sigma_t$. Score- and noise-prediction are hence equivalent up to a timestep-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 towards the original data point. Here, we refer to the noise- and score-prediction networks interchangeably.

Diffusion models are commonly trained directly using (a Monte-Carlo expectation of) score- or noise-matching objectives, e.g. $\int_0^T \mathrm{d}t\, \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbb{I})} \left(\|\epsilon - \hat{\epsilon}_\varphi(z_t, t)\|_2^2 \right)$. While extremely effective for e.g. high-quality image generation, such loss functions do not aim to directly maximize the data log-likelihood $\log p(x)$. Since one of our desiderata is to be able to perform accurate conditional likelihood estimation for parameter estimation purposes, we instead use a

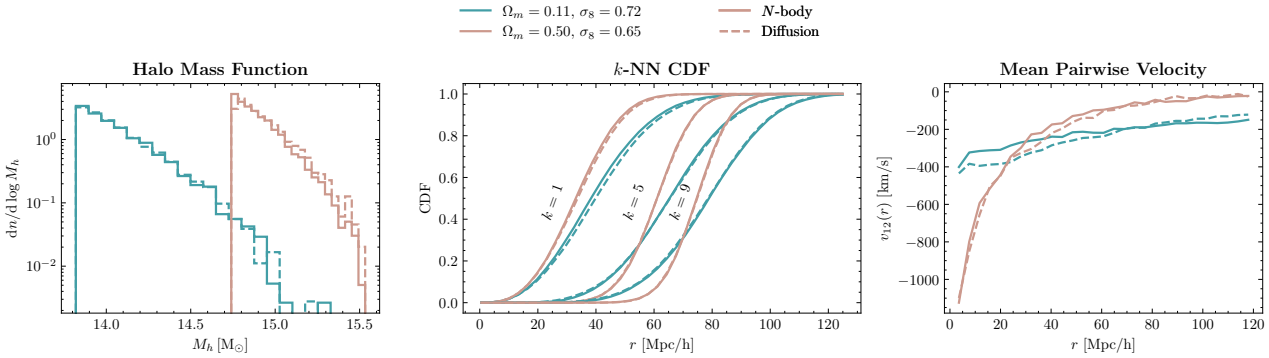

Figure 2. Summary statistics of the samples generated by the diffusion model compared to those of the N-body simulations for two cosmological parameter values from the test set, representing a low (blue) and a high (brown) clustering scenario. For each cosmology, all summary statistics are computed for the same emulated point cloud. From left to right, we show the halo mass function, the k-nearest neighbours statistics, and the mean pairwise velocity.

modified objective which maximizes a lower bound on the log-likelihood.

Considering complementary perspectives on diffusion, Vahdat et al. (2021); Song et al. (2021); Kingma et al. (2023) described how time-weighted sums of the traditional score-matching objective can be used as variational lower bound estimators of the likelihood, and hence as a proxy for maximum-likelihood training. We give a high-level overview of the formalism here; see Luo (2022); Kingma et al. (2023) for further details. We use the efficient and numerically stable implementation from Kingma et al. (2023), 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. Much like in a classical VAE (Kingma & Welling, 2013), the evidence lower bound (ELBO) objective can be used as a variational lower bound on the log-likelihood $\log p(x)$,

$$\mathrm{ELBO}(x) = -\underbrace{\mathbb{E}_{q(z_T|x)}\left[D_{\mathrm{KL}}\left(q\left(z_T \mid x\right) \| p\left(z_T\right)\right)\right]}_{\text{Prior matching}} + \underbrace{\mathbb{E}_{q\left(z_{t_1}|x\right)}\left[\log p\left(x \mid z_{t_1}\right)\right]}_{\text{Reconstruction}} + \underbrace{\mathcal{L}_{\mathrm{dif}}(x)}_{\text{Forward-reverse consistency}} \tag{1}$$

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 $q(z_t \mid x)$ are the (assumed Gaussian) variational posteriors on the noise addition. The diffusion loss $\mathcal{L}_{\mathrm{dif}}(x)$ ensures consistency between the forward (noising) and reverse (de-noising) steps, $\mathcal{L}_{\mathrm{dif}}(x) = -\sum_{t=2}^{T} \mathbb{E}_{q(z_t|x)}\left[D_{\mathrm{KL}}\left(q\left(z_{t-1} \mid z_t, x\right) \| p\left(z_{t-1} \mid z_t\right)\right)\right]$.

It can be shown (Kingma et al., 2023) that the diffusion loss considerably simplifies and can be computing through the Monte-Carlo expectation as $\mathcal{L}_{\mathrm{dif}}(x) = \frac{N_T}{2}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbb{I}),t\sim\mathcal{U}\{0,T\}}\left[w(t)\left\|\epsilon - \hat{\epsilon}_\varphi\left(z_t, t\right)\right\|_2^2\right]$ where N_T is the number of timesteps and $w(t)$ is a specific weighting term that depends on the (learned, monotonic) noise schedule. This can be seen as a weighted form of the traditional de-noising (equivalently, score-matching) objective. The noise-corruption (forward) model is $z_t \sim \mathcal{N}(\alpha_t x, \sigma_t \mathbb{I})$; σ_t and $\alpha_t \equiv \sqrt{1-\sigma_t^2}$, corresponding to a variance-preserving noise model. For brevity, we omit the expression for the weighting term $w(t)$ and the implementation of the learned noise schedule here, referring instead to Luo (2022); Kingma et al. (2023) for details.

2.2. The score model

The score function is a crucial part of the diffusion and model must be chosen sensitive to the data modality and generating process. In our case, we model the distribution of galaxies and their properties as a *point cloud* i.e., a collection of coordinates (positions), optionally with attached attributes (e.g., velocities), $p\left(\{\vec{r}_i; [v_i, \ldots]\}_{i=1}^{N_{\mathrm{gal}}}\right)$. Some requirements for a score model in our case are *(1)* permutation equivariance, *(2)* the ability to process points of arbitrary cardinality, and *(3)* the capacity to effectively model the joint correlation structure of galaxy/halo properties. Variants of the closely related transformer and graph neural networks (GNNs) families satisfy these requirements; here, we show an application using a GNN score model, described below.

We use a variant of the graph convolutional network from Battaglia et al. (2018). A local k-nearest neighbors graph with $k = 20$ is constructed, using the Euclidean distance between coordinates as the distance metric. After linearly projecting the input features z_t into an embedding space of 64 dimensions, 4 message-passing rounds are performed, constructing/updating the edge attributes at each round by passing a concatenation of the sender and receiver node

attributes, edge attributes, as well as global parameters (a combination of timestep embedding and conditioning parameters $\{\Omega_m, \sigma_8\}$) through an MLP. For each node, the neighboring edge attributes are aggregated, concatenated with the node attributes and globals, and passed through another MLP to update the node features. See Appendix A for more details on the graph neural network implementation.

2.3. Generation, likelihood evaluation, and inference

Generation: We can generate new samples from a learned diffusion model by *(1)* sampling an initial random noise configuration $z_T \sim \mathcal{N}(0, \mathbb{I})$, and *(2)* running the reverse diffusion process by sampling $z_t \sim p(z_{t-1} \mid z_t, t)$ until we arrive at $z_0 \equiv x$.

Likelihood evaluation: The diffusion model is trained using a stochastic estimate of the variational maximum-likelihood objective, Eq. (1). The same expression can be used to obtain an estimator of the conditional likelihood $\hat{p}(x \mid \theta)$, ensuring that the ELBO is evaluated a sufficient number of times to obtain a good estimate of the expectation value.

Posterior inference: Given a data sample x, we can use stochastic variational inference (SVI) (Jordan et al., 1999; Blei et al., 2017) to efficiently obtain the approximate posterior. The variational ansatz on the posterior distribution of the cosmological parameters is taken to be a multivariate Normal distribution $q_{\mu,\Sigma} = \mathcal{N}(\mu, \Sigma)$. The mean and covariance are obtained by minimizing the reverse KL-divergence between the true and variational posterior distributions, $D_{\mathrm{KL}}\left(q_{\mu,\Sigma} \,||\, p(\theta \mid x)\right)$. This is again obtained using the tractable evidence lower bound, $\mathrm{ELBO} \equiv \mathbb{E}_{q_{\mu,\Sigma}(\theta)}\left[\log p(x, \theta) - \log q_{\mu,\Sigma}(\theta)\right]$, as the optimization objective. In practice, SVI is implemented using the NumPyro (Phan et al., 2019) package.

3. Experiments and discussion

Dataset and training: We use 2000 N-body simulation boxes from the latin hypercube of the *Quijote* suite (Villaescusa-Navarro et al., 2020) at redshift zero, split 90/10 into training and validation sets. Dark matter halo coordinates are represented as a point cloud, choosing the heaviest 5000 halos by halo mass. The cosmological parameters $\theta = \{\Omega_m, \sigma_8\}$ are used as a conditioning context to the score model. The pipeline is trained using the variational maximum-likelihood objective in Eq. 1. $500,000$ iterations of the AdamW (Loshchilov & Hutter, 2019; Kingma & Ba, 2015) optimizer with peak learning rate 3×10^{-4} and 1000 linear warmup steps are run. We show results for models trained on either *(1)* halo positions only, for a one-to-one comparison with two-point correlations on parameter inference, or *(2)* halo positions, velocities and masses.

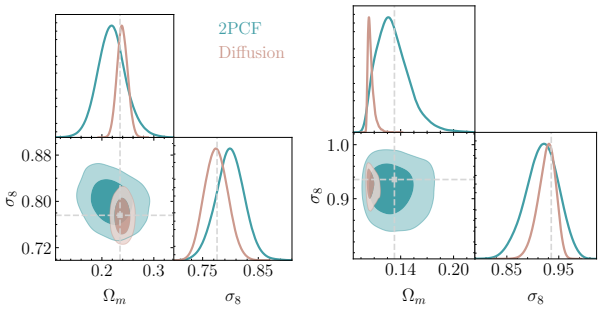

Figure 3. Results of posterior parameter inference using the trained likelihood estimator from the diffusion model, compared with posteriors obtained using the 2PCF. True parameter values are indicated by dashed lines. The diffusion-backed likelihood achieves tighter constraints on the two parameters by using more of the information encoded in the forward model.

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 validation set, for two extreme values of the cosmological parameters. Figure 2 shows the corresponding halo mass function, the cumulative distribution of k-nearest neighbors (k-NNs) and the mean pairwise velocity as a function of pair separation, v_{12}, which shows that the correlation between positions and velocities is correctly modelled. The expected correspondence with held out simulations are seen, over a wide range of scales, and the cosmology dependence is faithfully modeled.

Posterior inference: Figure 3 shows posterior distributions on $\{\Omega_m, \sigma_8\}$ obtained using variational inference for two held out halo sample using the likelihood estimated with the trained diffusion model (brown) and the 2PCF (green), using the diffusion model trained on halo positions only. The 2PCF posterior was obtained by training a normalizing flow to reproduce the posterior distribution of the cosmological parameters given the measured two point functions. As expected, the diffusion-backed likelihood achieves tighter constraints on the two parameters by using more of the information encoded in the forward model. However, we do not find that the posterior estimates obtained with the current model are well-calibrated for Ω_m, and inference can be significantly biased in some cases. An example of this scenario is shown in Figure 3 (right); further details are provided in Appendix B. It takes approximately 300 seconds to obtain a posterior on a Nvidia A100 GPU.

4. Conclusions and prospects

We have introduced a diffusion-generative model that captures the complex, non-Gaussian statistics of the galaxy clustering field along with the underlying cosmology depen-

dence. The model can be efficiently used for fast emulation of N-body simulations via sampling, $x \sim p(x \mid \theta)$, as well as posterior parameter estimation via variational inference or MCMC methods.

To enhance the simulation efficiency of the model and improve the calibration of the recovered posteriors, several properties of the data-generating process, which are learned implicitly, can be explicitly incorporated into the score model and optimization objective: the effect of periodic boundary conditions in the simulation box, and Euclidean (E(3)) symmetry to translations, rotations, and reflections. Since we have a fixed, limited number of simulations these improvements, e.g. the use of symmetry-preserving neural networks (Satorras et al., 2022; Batzner et al., 2022), have the potential to significantly enhance training data-efficiency and the fidelity of our generative model . Finally, scaling our model to observations will require the ability to process sets of a much larger cardinality than considered here. This could be achived, for example, using graph neural networks with hybrid local-global adjacency matrices, and/or using latent diffusion – performing diffusion in an encoded latent space, decoding back to physical space, with the latent representation preserving the desired symmetry properties (e.g., permutation equivariance).

Lastly, 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, *(3)* survey systematics, such as survey masks and fibre collitions. An example of such a forward model has been presented in *SimBIG* (Hahn et al., 2023). 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.

Acknowledgements

This work is supported by the National Science Foundation under Cooperative Agreement PHY- 2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/). 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 Contract Number 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. This research used the following software packages: CORRFUNC (Sinha & Garrison, 2020), FLAX (Heek et al., 2023), GETDIST (Lewis, 2019), JAX (Bradbury et al., 2018), JRAPH (Godwin* et al., 2020), JUPYTER (Kluyver et al., 2016), MATPLOTLIB (Hunter, 2007), NUMPY (Harris et al., 2020), NUMPYRO (Phan et al., 2019), OPTAX (Babuschkin et al., 2020) and WANDB (Biewald, 2020).

References

Adam, A., Coogan, A., Malkin, N., Legin, R., Perreault-Levasseur, L., Hezaveh, Y., and Bengio, Y. Posterior samples of source galaxies in strong gravitational lenses with score-based priors. *arXiv preprint arXiv:2211.03812*, 2022.

Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kapturowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J., Ring, R., Ruiz, F., Sanchez, A., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stokowiec, W., Wang, L., Zhou, G., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL http://github.com/deepmind.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1), may 2022. doi: 10.1038/s41467-022-29939-5. URL https://doi.org/10.1038%2Fs41467-022-29939-5.

Biewald, L. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Dai, B. and Seljak, U. Translation and rotation equivariant normalizing flow (TRENF) for optimal cosmological analysis. *Monthly Notices of the Royal Astronomical Society*, 516(2):2363–2373, jul 2022. doi: 10.1093/mnras/stac2010. URL https://doi.org/10.1093%2Fmnras%2Fstac2010.

Godwin*, J., Keck*, T., Battaglia, P., Bapst, V., Kipf, T., Li, Y., Stachenfeld, K., Veličković, P., and Sanchez-Gonzalez, A. Jraph: A library for graph neural networks in jax., 2020. URL http://github.com/deepmind/jraph.

Hahn, C., Eickenberg, M., Ho, S., Hou, J., Lemos, P., Massara, E., Modi, C., Dizgah, A. M., Blancard, B. R.-S., and Abidi, M. M. Simbig: mock challenge for a forward modeling approach to galaxy clustering. *Journal of Cosmology and Astroparticle Physics*, 2023 (04):010, apr 2023. doi: 10.1088/1475-7516/2023/04/010. URL https://dx.doi.org/10.1088/1475-7516/2023/04/010.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2023. URL http://github.com/google/flax.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models, 2023.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B. (eds.), *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pp. 87 – 90. IOS Press, 2016.

Lanusse, F., Mandelbaum, R., Ravanbakhsh, S., Li, C.-L., Freeman, P., and Póczos, B. Deep generative models for galaxy image simulations. *Monthly Notices of the Royal Astronomical Society*, 504(4):5543–5555, 2021.

Legin, R., Adam, A., Hezaveh, Y., and Levasseur, L. P. Beyond gaussian noise: A generalized approach to likelihood analysis with non-gaussian noise. *arXiv preprint arXiv:2302.03046*, 2023.

Lewis, A. Getdist: a python package for analysing monte carlo samples, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Luo, C. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

Makinen, T. L., Charnock, T., Lemos, P., Porqueres, N., Heavens, A. F., and Wandelt, B. D. The cosmic graph: Optimal information extraction from large-scale structure using catalogues. *The Open Journal of Astrophysics*, 5(1), dec 2022. doi: 10.21105/astro.2207.05202. URL https://doi.org/10.21105%2Fastro.2207.05202.

Phan, D., Pradhan, N., and Jankowiak, M. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

Remy, B., Lanusse, F., Jeffrey, N., Liu, J., Starck, J.-L., Osato, K., and Schrabback, T. Probabilistic mass mapping with neural score estimation. *arXiv preprint arXiv:2201.05561*, 2022.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks, 2022.

Sinha, M. and Garrison, L. H. CORRFUNC - a suite of blazing fast correlation functions on the CPU. MNRAS, 491 (2):3022–3041, Jan 2020. doi: 10.1093/mnras/stz3157.

Smith, M. J., Geach, J. E., Jackson, R. A., Arora, N., Stone, C., and Courteau, S. Realistic galaxy image simulation via score-based generative models. *Monthly Notices of the Royal Astronomical Society*, 511(2):1808–1818, 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34: 1415–1428, 2021.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Villaescusa-Navarro, F., Hahn, C., Massara, E., Banerjee, A., Delgado, A. M., Ramanah, D. K., Charnock, T., Giusarma, E., Li, Y., Allys, E., Brochard, A., Uhlemann, C., Chiang, C.-T., He, S., Pisani, A., Obuljen, A., Feng, Y., Castorina, E., Contardo, G., Kreisch, C. D., Nicola, A., Alsing, J., Scoccimarro, R., Verde, L., Viel, M., Ho, S., Mallat, S., Wandelt, B., and Spergel, D. N. The Quijote Simulations. ApJS, 250(1):2, September 2020. doi: 10.3847/1538-4365/ab9d82.

A. Details on the score model

The graph neural network employed here is composed of four graph-convolution layers $\mathbf{h}^{l+1}, \mathbf{e}^{l+1} = \mathrm{GCL}[\mathbf{h}^l, \mathbf{e}^l, \mathbf{g}^l]$ where $\mathbf{h}^0$ are the embeddings of the halo properties being modelled (either positions, or positions, velocities and masses), the edge features $\mathbf{e}^l$ are learned vectors of dimension 64, and $\mathbf{g}^0$ is the joint embedding of the diffusion time and the cosmological parameters. Layer normalizations is applied after each graph layer.

Each graph-convolution layer is defined as

$$\mathbf{e}_{ij}^{l+1} = \phi_e(\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{e}_{ij}^l), \;\; \mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \phi_h(\mathbf{h}_i^l, \sum_{i \neq j} \mathbf{e}_{ij}^{l+1}) \tag{2}$$

where the edge- and node-update neural networks ϕ_e and ϕ_h are both multilayer perceptrons composed of four hidden layers of dimension 64 and GELU activations.

B. Details on posterior inference

In Figure 4, we compare the mean and variance of the inferred posteriors to the true values for 100 test set simulations. In green, we show the inference resulting from the N-body simulations, showing that the estimation of Ω_m is overconfident, whereas the predicted uncertainties on σ_8 are better calibrated. In brown, we also show the inference results on samples generated with the diffusion model. In this case, the Ω_m uncertainties are better calibrated. This demonstrates that there is still a significant difference between the statistical properties of the generated and true samples when it comes to the conditional dependence on Ω_m. See section 4 for suggestions on how to improve the calibration of the recovered posteriors.

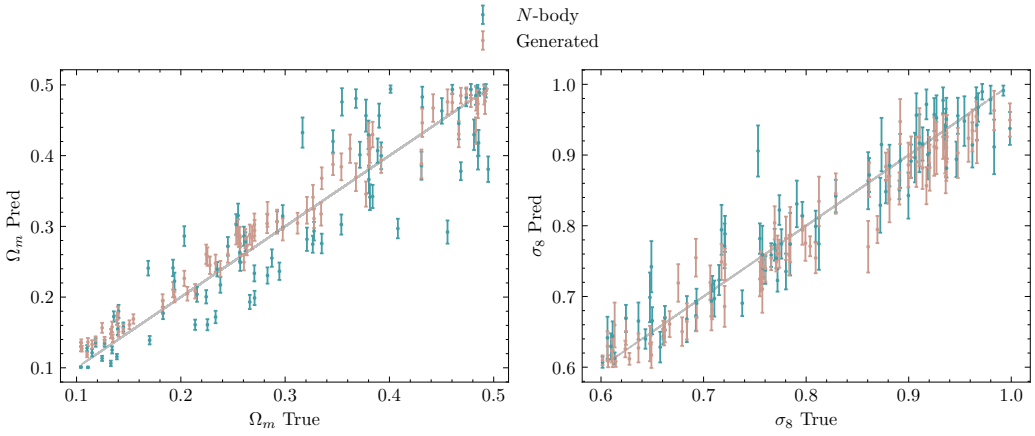

Figure 4. Comparison of the true parameters of the test set simulations to those inferred by the diffusion model, taken as the mean and standard deviation of the inferred posterior distribution. We show the difference between inferring the cosmological parameters from the true N-body simulations (N-body) and the generated diffusion samples (Generated).