Fast and realistic large-scale structure from machine-learning-augmented random field simulations

Davide Piras [1,2]★ Benjamin Joachimi[1] and Francisco Villaescusa-Navarro[3,4]

[1]*Department of Physics and Astronomy, University College London, Gower Street, London, WC1E 6BT, UK*
[2]*Département de Physique Théorique, Université de Genève, 24 quai Ernest Ansermet, 1211 Genève 4, Switzerland*
[3]*Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA*
[4]*Department of Astrophysical Sciences, Princeton University, 4 Ivy Lane, Princeton, NJ 08544, USA*

ABSTRACT

Producing thousands of simulations of the dark matter distribution in the Universe with increasing precision is a challenging but critical task to facilitate the exploitation of current and forthcoming cosmological surveys. Many inexpensive substitutes to full N-body simulations have been proposed, even though they often fail to reproduce the statistics of the smaller non-linear scales. Among these alternatives, a common approximation is represented by the lognormal distribution, which comes with its own limitations as well, while being extremely fast to compute even for high-resolution density fields. In this work, we train a generative deep learning model, mainly made of convolutional layers, to transform projected lognormal dark matter density fields to more realistic dark matter maps, as obtained from full N-body simulations. We detail the procedure that we follow to generate highly correlated pairs of lognormal and simulated maps, which we use as our training data, exploiting the information of the Fourier phases. We demonstrate the performance of our model comparing various statistical tests with different field resolutions, redshifts, and cosmological parameters, proving its robustness and explaining its current limitations. When evaluated on 100 test maps, the augmented lognormal random fields reproduce the power spectrum up to wavenumbers of $1\,h\,\mathrm{Mpc}^{-1}$, and the bispectrum within 10 per cent, and always within the error bars, of the fiducial target simulations. Finally, we describe how we plan to integrate our proposed model with existing tools to yield more accurate spherical random fields for weak lensing analysis.

Key words: methods: statistical – software: simulations – large-scale structure of Universe – dark matter.

1 INTRODUCTION

The best current model to describe our Universe is the ΛCDM model, which prescribes the existence of a cosmological constant Λ associated with dark energy, together with cold dark matter (CDM) and ordinary matter (baryons; see e.g. Dodelson 2003). In particular, the ΛCDM model predicts that dark matter is about five times more abundant than ordinary matter, with galaxies forming along the cosmic web structure woven by dark matter, made of filaments connecting different clusters, all surrounded by voids. While its gravitational effects are observed by many probes, dark matter remains a mystery, with multiple experiments still ongoing to shed light on its nature (see e.g. Trimble 1987; Bertone, Hooper & Silk 2005; Buchmueller, Doglioni & Wang 2017; de Swart, Bertone & van Dongen 2017, and references therein).

The most common tool to analyse and track the origin and evolution of dark matter structures are cosmological N-body simulations (Holmberg 1941; Navarro, Frenk & White 1996; Tormen 1997; Jenkins et al. 1998; Springel 2005; Springel et al. 2005; Boylan-Kolchin et al. 2009; Angulo et al. 2012; Villaescusa-Navarro et al. 2021, 2020; Chacón, Vázquez & Gabbasov 2020, and references

therein). In its basic formulation, an N-body simulation is run by putting a certain number of massive particles in a cubic box, imposing periodic boundary conditions and letting gravity be the only force acting on the particles through its gravitational potential, governed by the Poisson equation (Springel 2005). The initial conditions of the Universe are usually approximated with a Gaussian density field,[1] which can be entirely summarized by a given power spectrum, i.e. by the Fourier counterpart of the correlation function between different particles in the simulation. Starting from high redshift, the position and velocity of the particles are updated iteratively until today ($z = 0$), while various snapshots are taken at different redshifts.

Several methods to run an N-body simulation are available, with different levels of complexity, approximation, and speed (Hockney & Eastwood 1988; Chacón et al. 2020). These include the direct resolution of the equation of motion for each particle (Mikkola & Aarseth 1993), approximated methods like the tree code method (Barnes & Hut 1986; Callahan & Kosaraju 1992), or mean-field approaches like standard (Klypin & Holtzman 1997) or adaptive (O'Shea et al. 2004) particle mesh. In general, though, N-body simulations are computationally expensive to run, and usually require access to

★ E-mail: dr.davide.piras@gmail.com

[1]Hereafter, 'initial conditions' will only refer to the initial conditions of N-body simulations.
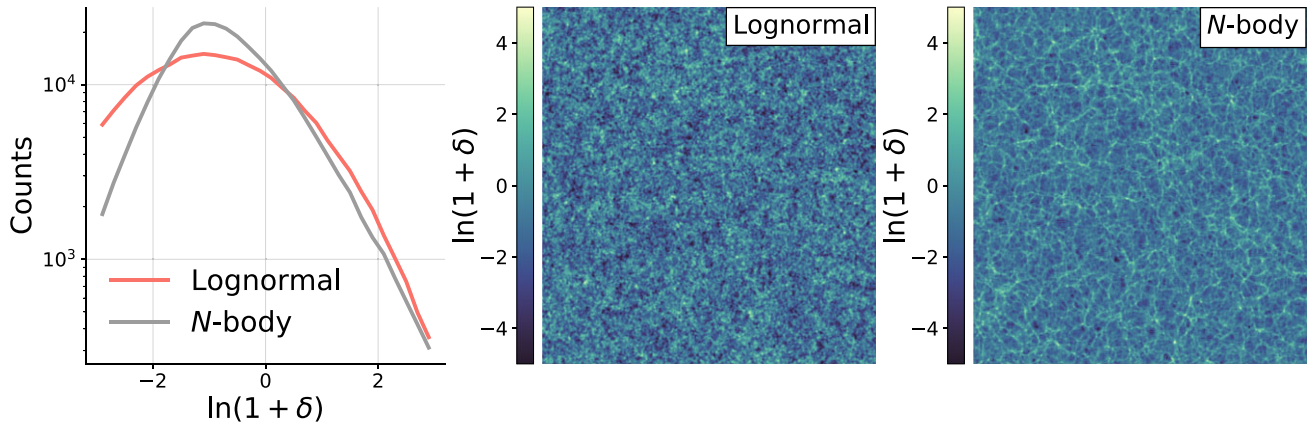
Figure 1. *Left-hand panel*: histograms of the matter overdensity δ, defined in equation (1), for a lognormal random field (red) and an *N*-body simulation dark matter density field (grey). *Middle and right-hand panels*: square maps of a lognormal (middle) and *N*-body (right) density fields, with a side of 512 pixels, corresponding to a comoving length of $1\,h^{-1}$ Gpc. The depth of these fields is $\simeq 1.9\,h^{-1}$ Mpc. In these maps, we clipped the maximum and minimum values before applying a logarithm to reduce their dynamic range; the symbol 'ln' indicates the natural logarithm throughout this paper. The right-hand-side plot is a slice of a simulation from the Quijote suite (Villaescusa-Navarro et al. 2020), while the middle plot, obtained following the procedure described in Section 3.1, represents its lognormal counterpart. The goal of this paper is to train a machine learning model (described in Section 3.2) to transform the lognormal map to the more realistic *N*-body map, thus improving the statistical power of the fast lognormal approximation.

high performance computing hardware. This limits the possibility of fully exploring the impact of different cosmological parameters on the dark matter evolution in our Universe, and hinders statistical analyses of the large-scale structure (see e.g. Taylor, Joachimi & Kitching 2013; Taylor & Joachimi 2014): *N*-body simulations are essential to associate a covariance matrix to real measurements, and thousands of simulations are required to obtain accurate estimates of such matrices.

In recent years, many cheaper approximations have been proposed, which try to capture both the large-scale structure of the cosmic web and its smaller-scale details. These approximations often rely on Lagrangian perturbation theory (Buchert 1992; Buchert & Ehlers 1993; Buchert 1994), and can produce accurate dark matter halo mock catalogues and dark matter density fields (Monaco et al. 2002; Monaco et al. 2013; White, Tinker & McBride 2013; Kitaura, Yepes & Prada 2013; Chuang et al. 2014; Tassev, Zaldarriaga & Eisenstein 2013; Tassev et al. 2015; Howlett, Manera & Percival 2015; Rizzo et al. 2017; Tosone et al. 2020, 2021). While being capable of capturing the large-scale structure statistics with fewer computational resources, these methods usually fail to accurately produce the correct small-scale statistics. Although such approximate mocks are used to estimate covariance matrices in current large-volume data sets when not enough high-resolution simulations are available, to date no inexpensive exact alternative to *N*-body realizations exists.

Another typical approximation to describe (dark) matter fields is found by resorting to a lognormal random field, which represents the simplest alternative to running an entire *N*-body simulation (Coles & Jones 1991; Peebles 1993; Taruya et al. 2002; Percival, Verde & Peacock 2004; Hilbert, Hartlap & Schneider 2011; Xavier, Abdalla & Joachimi 2016). A lognormal random field can be easily obtained from a Gaussian random field (see Section 3.1 for further details), and can be entirely described by a small number of parameters; moreover, a lognormal variable has a skewed distribution which is suited for e.g. the matter overdensity field, whose values range from −1 in voids to values much higher than 1 in clustered dense regions. However, as reported in Xavier et al. (2016) and shown in Fig. 1, the lognormal approximation comes with its

own limitations, and fails to reproduce the correct matter density distribution.

Machine learning (ML) techniques have also been proposed to replace expensive *N*-body simulations. In Rodríguez et al. (2018), generative adversarial networks (GANs; Goodfellow et al. 2014) were successfully trained to generate slices of *N*-body simulations, and Mustafa et al. (2019) applied the same technique to weak lensing convergence maps. Perraudin et al. (2019) and Feder, Berger & Stein (2020) then extended the application of GANs to 3D boxes, proving that, while challenging to train, GANs can capture both large- and small-scale features, and are capable of accurately recovering the statistical information contained in the training data. He et al. (2019) and Alves de Oliveira et al. (2020), on the other hand, showed that it is possible to train a U-shaped neural network architecture (U-net; Ronneberger, Fischer & Brox 2015) to map simple linear initial conditions to the corresponding final evolved fields, correctly learning the non-linear growth of structures under the gravitational influence. Kaushal et al. (2021) additionally used Lagrangian perturbation theory to evolve such initial conditions and only learn the difference in the density fields at $z = 0$. In these latter works, it was also shown that such architectures can perform well even on input data obtained from different cosmological parameters than the training data, thus demonstrating the appealing feature of being able to extrapolate outside the training distribution. Other works have explored the use of super-resolution techniques to *N*-body simulations (Kodi Ramanah et al. 2020; Li et al. 2021), the application of normalizing flows (e.g. Papamakarios et al. 2019) as generative models of the large-scale structure (Rouhiainen, Giri & Münchmeyer 2021; Dai & Seljak 2022), wavelet phase harmonics statistics to produce realistic 2D density fields (Allys et al. 2020), or combinations of ML-inspired techniques with more traditional methods to improve the accuracy of fast *N*-body solvers (Dai, Feng & Seljak 2018; Dai et al. 2020; Dai & Seljak 2021; Böhm et al. 2021).

While being useful, all the previous approaches still require a relatively high amount of computational resources, might not scale well to high-resolution fields, or introduce many approximations that prevent them from being used reliably in place of full *N*-body simulations. In this paper, we show that it is possible to improve the

lognormal approximation by means of ML techniques, with the long-term goal of integrating our approach with the *Full-sky Lognormal Astro-fields Simulation Kit* (FLASK; Xavier et al. 2016), in order to be able to cheaply generate more realistic high-resolution full-sky density fields.

For this purpose, we start from the Quijote *N*-body simulation suite (Villaescusa-Navarro et al. 2020), which offers thousands of realizations of a single cosmological parametrization, as well as hundreds of simulations at different values of the cosmological parameters. We devise a pipeline to create lognormal density fields which are the approximated counterpart of the simulated density fields. By construction, these lognormal fields have the same power spectrum as the one from the fiducial *N*-body simulations, and the phases of the underlying Gaussian fields are taken from the initial conditions of the simulated fields (all details are reported in Section 3.1). Having the pairs of lognormal and corresponding simulated density fields, we draw from image-to-image translation techniques based on convolutional neural networks and adversarial training, in order to obtain a model that can map simple lognormal fields to more realistic density fields (see Fig. 1). We extensively validate our model by measuring first-, second-, and higher-order statistics, obtaining good agreement, almost always within 10 per cent, on all scales. We additionally show that we can train our model using simulations run over a latin hypercube of cosmological parameters, obtaining a good generalization performance over different cosmologies. While a more extensive study for different redshifts and higher resolutions will be explored in future work, these encouraging results indicate that providing the model with a lognormal field as the starting point significantly improves the model's generalization performance. Additionally, we show that starting from lognormal maps with the correct power spectrum naturally leads to good performance at the power spectrum level, and does not allow the model to 'collapse' and learn single modes in the data, a known problem of generative adversarial networks (Metz et al. 2017).

The paper is structured as follows. In Section 2, we describe the Quijote simulation data, on which this work is based. In Section 3, we detail the procedure that we apply to obtain the training data, and describe the image-to-image translation technique that we employ in this work. In Section 4, we present the results for different resolutions of the density fields, as well as for different values of redshift and cosmological parameters, and demonstrate the performance of our model through a wide range of statistical tests. We conclude in Section 5 with a summary of our work, planned improvements, and an outline of possible future applications of our model.

2 DATA

In this work, we use the Quijote simulation suite (Villaescusa-Navarro et al. 2020). This set of *N*-body simulations includes 15 000 realizations following 512^3 dark matter particles in a box with comoving length of $1\,h^{-1}$ Gpc, with the matter density parameter $\Omega_m = 0.3175$, the baryon density parameter $\Omega_b = 0.049$, the Hubble parameter $h = 0.6711$, the scalar spectral index $n_s = 0.9624$, the root mean square of the matter fluctuations in spheres of radius $8\,h^{-1}$ Mpc, $\sigma_8 = 0.834$, and the dark energy equation of state parameter $w = -1$; neutrinos are considered massless. These simulations were run using the TREEPM code GADGET-III, which is an improved version of GADGET-II (Springel 2005). We consider snapshots of both the initial conditions ($z = 127$) and today ($z = 0$), as well the $z = 1$ snapshot and the latin-hypercube simulations at $z = 0$ for further validation of our model (see Section 4.5).

In each *N*-body simulation, we convert the information on the particles' position to a continuous random field through a mass assignment scheme. We analyse the matter overdensity field $\delta(\boldsymbol{x})$, defined as:

$$\delta(\boldsymbol{x}) = \frac{\rho(\boldsymbol{x})}{\bar{\rho}} - 1, \tag{1}$$

with $\rho(\mathbf{x})$ being the matter density field at each position $\boldsymbol{x}$, and $\bar{\rho}$ being the mean density in the volume of the simulation.

Following Chaniotis & Poulikakos (2004), Jing (2005), and Sefusatti et al. (2016), we consider a regular grid of points in all three directions. The continuous overdensity field is obtained by interpolating the discrete overdensity field on this grid, i.e. by evaluating the continuous function

$$\tilde{\delta}(\boldsymbol{x}) = \int \frac{\mathrm{d}\boldsymbol{x}'}{(2\pi)^3} W(\boldsymbol{x} - \boldsymbol{x}') \delta(\boldsymbol{x}') \,, \tag{2}$$

with $W(\mathbf{x})$ being the weight function describing the number of grid points to which every particle is assigned. We choose the piecewise cubic spline interpolation scheme, i.e. we explicitly write the weight function as $W(\mathbf{x}) = W_{1D}(x_1/H)W_{1D}(x_2/H)W_{1D}(x_3/H)$, with H being the grid spacing, x_1 (x_2, x_3) being the x (y, z) direction, and W_{1D} being the unidimensional weight function

$$W_{1D}(s) = \begin{cases} \frac{4-6s^2+3|s|^3}{6} & \text{if } 0 \leq |s| < 1 \,; \\ \frac{(2-|s|)^3}{6} & \text{if } 1 \leq |s| < 2 \,; \\ 0 & \text{otherwise} \,; \end{cases} \tag{3}$$

we refer the reader to Sefusatti et al. (2016) for more details. We consider both a grid with $N_{high}^3 = 512^3$ pixels and $N_{low}^3 = 128^3$ pixels, and present the results in Section 4.3 and Section 4.4, respectively.

3 METHOD

Our goal is to obtain 2D projected density lognormal fields corresponding to slices of the QUIJOTE simulations, in order to train a model that can take as input a lognormal map and predict a more realistic density field with the same statistics as the simulated one. In the following sections, we describe the procedure that we follow to obtain such a data set (Section 3.1), and the machine learning algorithm that we employ to learn the transformation (Section 3.2).

3.1 Obtaining the training data

Since the long-term goal of the project is to increase the accuracy in large-scale structure description of random field maps on the sphere like the ones produced by FLASK (Xavier et al. 2016), we choose to work with slices of the density field rather than the full 3D boxes. We slice a given box along the third axis, and obtain multiple square density fields from a single simulation (128 in the low-resolution case, and 512 in the high-resolution case); the width of each slice is $1000\,h^{-1}$ Mpc$/128 \simeq 7.8\,h^{-1}$ Mpc in the former case, and $1000\,h^{-1}$ Mpc$/512 \simeq 1.9\,h^{-1}$ Mpc in the latter case. Our choice of different thicknesses aims to demonstrate that our approach can work at different resolutions and different projection depths. Since we consider 800 simulations in the low-resolution case, and 200 in the high-resolution case, we are left with 102 400 maps in both instances. We also consider the initial conditions of the 3D boxes, namely the *N*-body simulations at $z = 127$, which we slice in the same way.

In order to create the lognormal counterpart of the more realistic maps, we start by measuring the power spectrum of each simulation's slice at $z = 0$, which we wish to impose on the lognormal fields. We

recall here that the 2D matter power spectrum $P(k)$ can be implicitly defined through the Fourier transform $\delta(\boldsymbol{k})$ of the matter density contrast $\delta(\boldsymbol{x})$,[2] defined as in equation (1):

$$\langle\delta(\mathbf{k})\delta(\mathbf{k}')\rangle = (2\pi)^2\, P(k)\delta_\mathrm{D}(\mathbf{k}+\mathbf{k}')\,, \tag{4}$$

where $\langle\,\cdot\,\rangle$ denotes an average over the whole Fourier space, $k = |\boldsymbol{k}|$, and $\delta_\mathrm{D}(\cdot)$ indicates the Dirac delta function (Dodelson 2003); this in turn yields the estimator

$$\hat{P}(k) = \frac{1}{N_\mathrm{modes}(k)}\sum_{|\boldsymbol{k}|=k}|\delta(\boldsymbol{k})|^2\,, \tag{5}$$

where $N_\mathrm{modes}(k)$ is the number of modes in each k bin, and the sum is performed over all $\boldsymbol{k}$ vectors whose magnitude is k. The definition in equation (4) implies that $P(k)$ is the Fourier counterpart of the 2D matter correlation function $\xi(r)$, with $r = |\boldsymbol{r}|$, i.e.

$$P(k) = \int \xi(r)e^{-\mathrm{i}k\cdot r}\,\mathrm{d}^2\boldsymbol{r}\,, \tag{6}$$

where $\xi(r)$ is defined as

$$\xi(r) = \langle\delta(\mathbf{x})\delta(\mathbf{x}+\mathbf{r})\rangle\,, \tag{7}$$

with $\langle\,\cdot\,\rangle$ representing the average over all locations $\boldsymbol{x}$ in the plane in this case.

In order to generate a lognormal random field with a given power spectrum, we follow the procedure of Coles & Jones (1991) and Percival et al. (2004). We start by converting the measured power spectrum to the matter correlation function $\xi_\mathrm{LN}(r)$, then we calculate the corresponding Gaussian correlation function,

$$\xi_\mathrm{G}(r) = \ln\left[1 + \xi_\mathrm{LN}(r)\right]\,, \tag{8}$$

transform it back to Fourier space and create a Gaussian random field realization δ_G on a grid with this power spectrum and the required resolution (N_low or N_high). It is well known that a zero-mean Gaussian field is entirely specified by the given power spectrum, which only depends on the absolute value of the Fourier coefficients: this means that the Fourier phases can be uniformly sampled from the $[0, 2\pi]$ interval (Coles & Chiang 2000; Chiang & Coles 2000; Watts, Coles & Melott 2003). Crucially, when generating the Gaussian random field, we employ the set of phases of the Gaussian initial conditions of the Quijote simulation realization. In this way, the final lognormal density fields will have a high level of correlation with the density fields obtained from the simulations: while the amount of correlation is limited due to the evolution from $z = 127$ to $z = 0$, the Pearson correlation coefficient between pairs of maps can be as high as 0.5, if we smooth the fields with a Gaussian kernel on scales of about $50\,h^{-1}$ Mpc, while it is consistent with 0 if using completely random phases. Therefore, we argue that our choice facilitates learning the mapping from random fields to N-body slices.

Finally, we obtain the lognormal field δ_LN by calculating for each grid point

$$\delta_\mathrm{LN} = \exp\left(\delta_\mathrm{G} - \sigma_\mathrm{G}^2/2\right) - 1 \tag{9}$$

where σ_G is the standard deviation of the Gaussian field. For all these operations, we employ the PYTHON package NBODYKIT (Hand et al. 2018). A flowchart representing the steps followed to produce the training data is reported in Fig. 2.

We observe two limitations due to the fact that we measure the power spectrum from a finite-resolution grid. First, by relying on the boxes only, we are capable of surveying only a limited range in k,

namely no larger than $k \in [0.025\,h\,\mathrm{Mpc}^{-1}, 1\,h\,\mathrm{Mpc}^{-1}]$ in the high-resolution case, and $k \in [0.025\,h\,\mathrm{Mpc}^{-1}, 0.3\,h\,\mathrm{Mpc}^{-1}]$ in the low-resolution case. In order to access larger scales (i.e. lower k values), we concatenate the measured power spectrum with the theoretical one obtained with CLASS (Blas et al. 2011) for $k \in [10^{-5}\,h\,\mathrm{Mpc}^{-1}, 0.025\,h\,\mathrm{Mpc}^{-1}]$: this makes the procedure outlined in the previous paragraphs more stable numerically.

Second, we observe a mismatch in power in the lognormal fields with respect to the imposed power spectrum. We attribute this discrepancy to the fact that when converting the Quijote initial conditions (obtained using second-order Lagrangian perturbation theory) to a density field, the mass assignment scheme and non-vanishing non-linearities arising from perturbation theory introduce extra spurious correlation in the phases. We correct for this effect, which actually introduces non-Gaussian features and is more pronounced at higher resolution, by iteratively rescaling the input power by the ratio of the output and target power at each k, until the mismatch across a sample of 100 random maps is smaller than 1 per cent at all k values. We checked that this iterative adaptation scheme effectively removes the power mismatch, leaving the final performance of the model unaffected at prediction time: The results presented in Section 4 do not change significantly if, after training the model, we give it as an input a slice of a 3D lognormal field generated with completely random Fourier phases. We further remark that this correction would not be necessary if we had access to a perfectly Gaussian density field of the initial conditions.

We are left with pairs of square density field maps (dubbed δ_LN and δ_SIM), which we use as the training (80 per cent), validation (10 per cent), and test (10 per cent) data, further discussed in the next section. This split is done at simulation level, so that the test, validation, and training data sets are completely independent. Note that to reduce the correlations between slices coming from the same simulation cube, we shift the pixels by a random amount along both the first and second axis, independently for each pair of maps, assuming periodic boundary conditions. It could also be possible to randomly rotate and flip the slices in order to augment the training data; while we found it is not needed in our set-up, we defer further investigations to future work. Before feeding the pairs into the neural network architecture described in the next section, we additionally pre-process each map by calculating $\ln(1 + \delta)$ to decrease the dynamic range of each density value δ.

3.2 Image-to-image translation

As discussed in Section 1, machine learning generative techniques have extensively been applied to N-body simulations. In this work, we aim at mapping lognormal fields to more realistic fields, hence, we employ the *pix2pix* network structure, first proposed in Isola et al. (2017). The model is composed of two parts, as sketched in Fig. 3; all implementation details are reported in Appendix A. The first part is a U-net (Ronneberger et al. 2015), which takes as an input a lognormal map δ_LN, obtained and pre-processed as described in Section 3.1. The map is passed through various convolutional layers to yield a compressed feature map, which is then upsampled back to the original resolution. Crucially, these upsampling steps are concatenated with the corresponding downsampled feature maps, which allow various scales to be accessible in the output map; removing these skip connections significantly impairs the performance of the model. We call the output map the *generated* map δ_GEN.

We want the generated map to carry the same statistical information as the δ_SIM density field. We tested that minimizing a simple

---

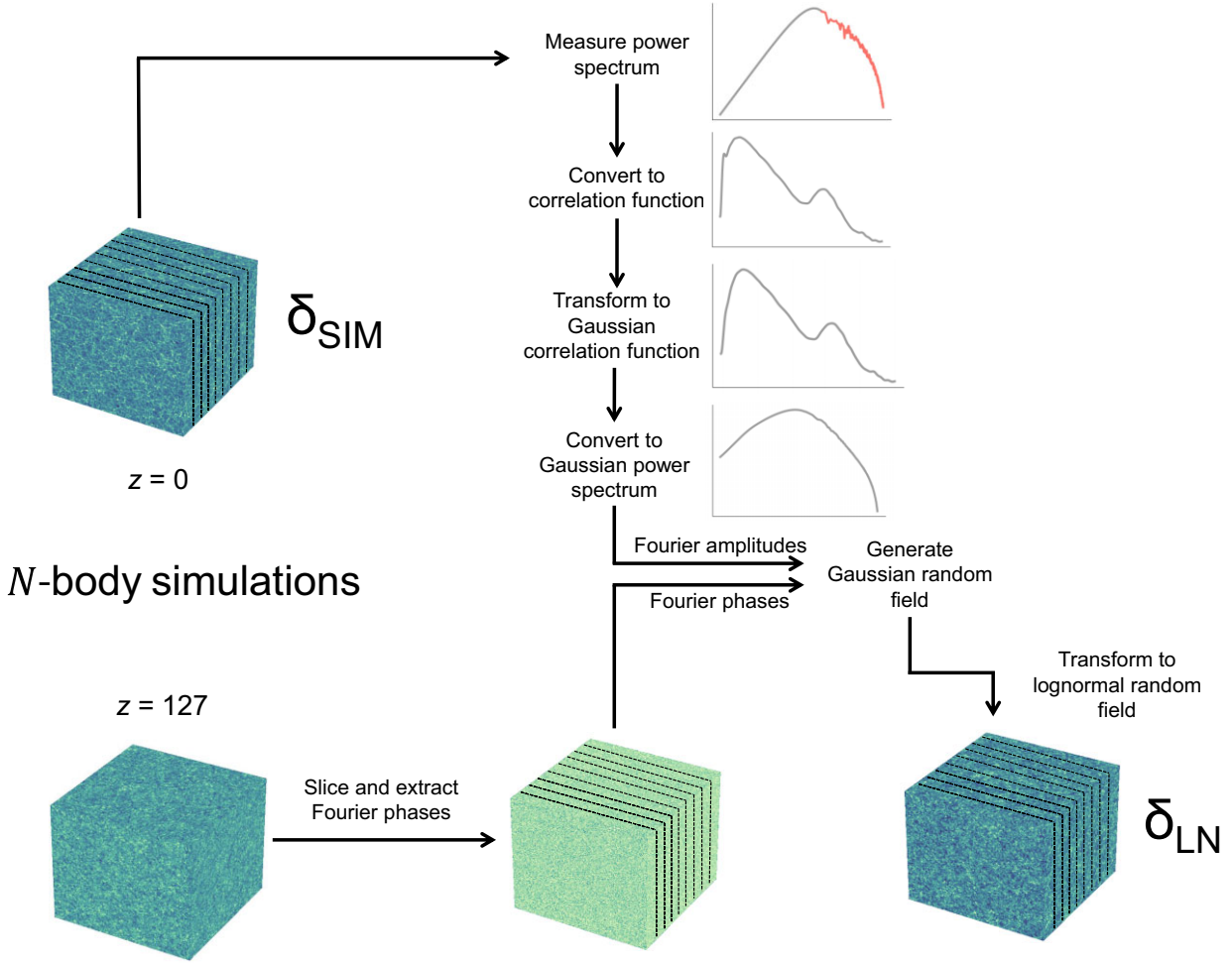[2]From now on, we use $\mathbf{k}$, $\mathbf{x}$, and $\mathbf{r}$ to indicate projected 2-D vectors.

Figure 2. Flowchart of the steps to create the training data, as described in Section 3.1. We first measure the power spectrum of each slice of the $z = 0$ boxes (in red in the top panel), which is concatenated with the theory power spectrum obtained using CLASS (Blas, Lesgourgues & Tram 2011; in grey in the top panel). We then generate a lognormal random field with this power spectrum, following Coles & Jones (1991), Percival et al. (2004); the term 'Gaussian correlation function' indicates $\xi_G(r)$ as in equation (8); the corresponding 'Gaussian power spectrum' is obtained using equation (6). Crucially, when generating the underlying Gaussian field, we use the Fourier phases of the initial conditions of the N-body simulation, which consist of a Gaussian random field at $z = 127$. In this way, the lognormal field displays increased correlation with the N-body field. The final training data consist of pairs of lognormal (δ_{LN}) and simulated (δ_{SIM}) density fields, with either low (side $N_{low} = 128$) or high (side $N_{high} = 512$) resolution, as explained in Section 2. The machine learning model employed to learn the mapping from δ_{LN} to δ_{SIM} is presented in Section 3.2.

ℓ^1 or ℓ^2 norm between δ_{GEN} and δ_{SIM} is not sufficient to yield accurate results. For this reason, following Isola et al. (2017), we employ a second convolutional block as a discriminator, and express the loss in the framework of adversarial training. In the standard GAN framework (Goodfellow et al. 2014), the generator network G is trained together with the discriminator network D until an equilibrium where neither G or D can improve their performance is reached: while G attempts to generate realistic images, D tries to distinguish between real and fake examples. Since we found this framework to be particularly unstable during training, we actually implemented the Wasserstein GAN with gradient penalty (WGAN-GP; Arjovsky, Chintala & Bottou 2017; Gulrajani et al. 2017), which we found superior both in performance and training stability. In this framework, a generator G is trained alongside a critic C to minimize the following cost function:

$$\mathbb{E}_{\delta_{GEN}}\left[C(\delta_{GEN})\right] - \mathbb{E}_{\delta_{SIM}}\left[C(\delta_{SIM})\right] + \lambda_1 \mathbb{E}_{\hat{\delta}}\left[\left(||\nabla_{\hat{\delta}} C(\hat{\delta})||_2 - 1\right)^2\right]$$

$$+ \lambda_2 ||\delta_{GEN} - \delta_{SIM}||_2^2, \tag{10}$$

where $\delta_{GEN} = G(\delta_{LN})$, $\mathbb{E}_{\delta_{GEN}}$, and $\mathbb{E}_{\delta_{SIM}}$ indicate the expectation value over samples of the generated and simulated maps (usually estimated through sample averages), respectively, $\hat{\delta}$ represents a linear combination of δ_{GEN} and δ_{SIM},[3] $|| \cdot ||_2$ indicates the ℓ^2 norm, and λ_1 and λ_2 are two positive hyperparameters that allow us to tune the amount of regularization given by the gradient penalty and the ℓ^2 norm, respectively. In short, equation (10) indicates that we wish to minimize the Wasserstein-1 (or earth mover) distance between the real data and generated data distributions, while constraining the gradient of the critic network to be close to unity; this is needed since the formulation of the Wasserstein distance as in the first two terms of equation (10) only holds when the critic is a 1-Lipschitz function,

---

[3]In particular, $\hat{\delta} = \delta_{SIM} + u(\delta_{GEN} - \delta_{SIM})$, with $u \sim U(0, 1)$, where $U(0, 1)$ indicates the uniform distribution between 0 and 1. This linear combination means that we are constraining the gradient norm to be 1 only along lines connecting real and fake data, which should be sufficient to guarantee good experimental results (Gulrajani et al. 2017).
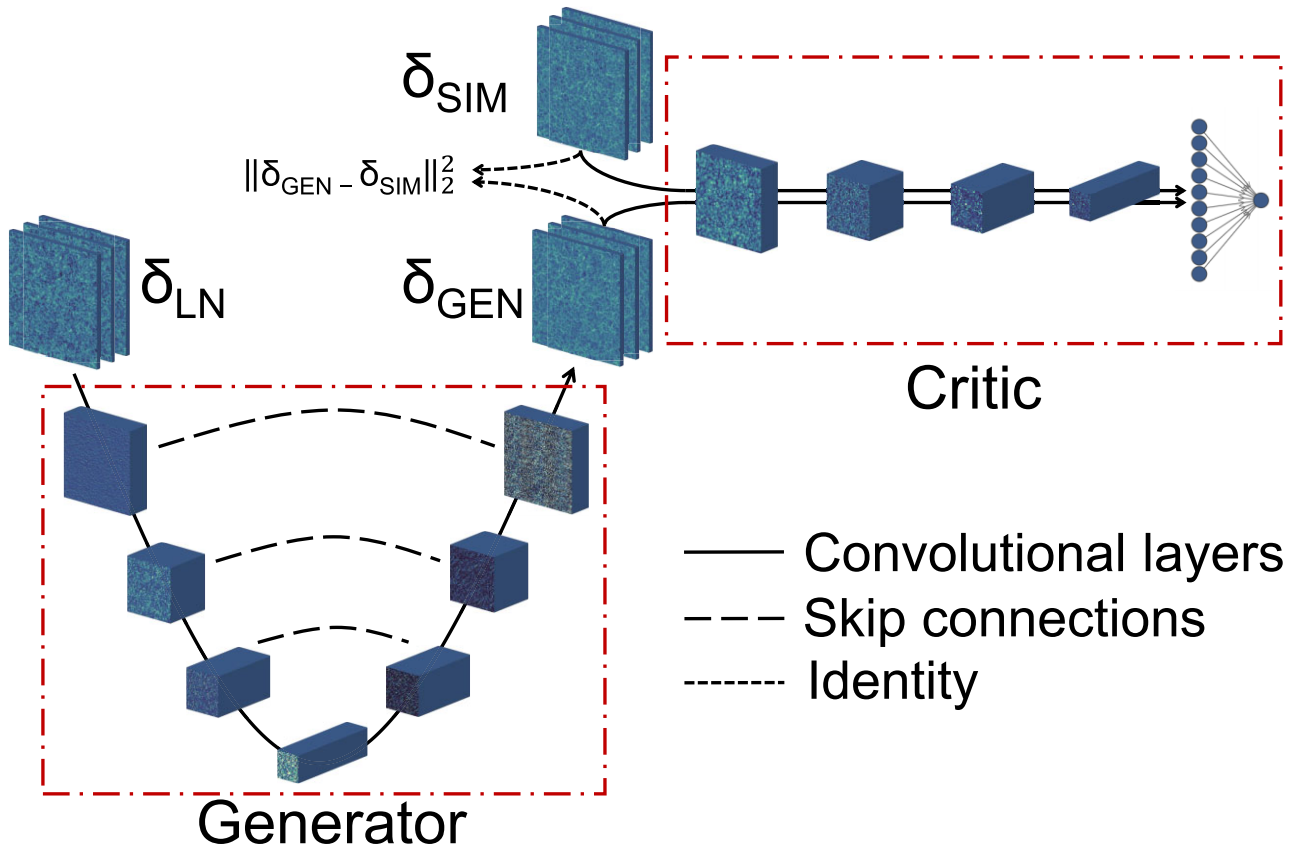
Figure 3. A representation of the generative model employed in this work, as described in Section 3.2. Following Isola et al. (2017), we have two convolutional neural networks, the generator (bottom left) and the critic (top right). We feed the lognormal maps through the generator, which is a U-net (Ronneberger et al. 2015), that first downsamples and then upsamples each image using various convolutional layers, with all details reported in Appendix A. To improve the performance of the model, each upsampling step is concatenated with the output of a downsampling step, as indicated by the dashed lines (*skip connections*). The output of the generator, dubbed δ_{GEN}, is then compared with the target data δ_{SIM} by the critic network, which is again made of various convolutional layers, ending with a dense layer in order to have a single output. We chose this architecture based on those described in the literature (e.g. Isola et al. 2017); a full investigation over different architecture designs is beyond the scope of this paper. The critic and generator networks are trained together, minimizing the loss function of equation (10). Note that in addition to the standard adversarial loss, we include a penalty term in the form of the mean squared error between the generated and target maps, which we found to significantly improve the performance of our model; this is indicated by the short-dashed lines (*identity*).

i.e. when its gradient is bound (see Gulrajani et al. 2017, for more details). We observe that in the standard WGAN-GP formulation $\lambda_2 = 0$, while in our case, we found it key to minimize the ℓ^2 norm between simulated and generated maps as well in order to obtain improved results.

To train the networks, we use the Adam optimizer (Kingma & Ba 2014) with learning rate 10^{-5}; we set the additional Adam hyperparameters $\beta_1 = 0$ and $\beta_2 = 0.9$, following Gulrajani et al. (2017), and refer the reader to Kingma & Ba (2014) and Gulrajani et al. (2017) for more details. We feed the data in batches of 32 pairs at each iteration, and train our model for 10 epochs (150 in the low-resolution case), where each epoch consists of feeding the entire training set through the network. For each batch, we update the critic parameters $n_{critic} = 10$ times, and the generator parameters only once. Multiple iterations of the critic are usually set to ensure its optimality while still allowing the generator to learn (Arjovsky et al. 2017; Gulrajani et al. 2017); in our work, we only explored $n_{critic} = 5$ and $n_{critic} = 10$, and used the latter since it showed slightly better results. Each epoch takes about 0.5 h (4 h) for the low (high) resolution case, on a Tesla P100 GPU; after training, mapping a lognormal map through the generator

takes $\mathcal{O}(1\ s)$ on the same hardware, and can be efficiently done in batches.

We save the model after each epoch. In order to select the best model amongst the saved ones, for each of them, we run the statistical tests described in Section 4.2, and measure the mean percentage difference between the target and predicted maps for randomly sampled maps of the validation set. The best model is chosen as the one which minimizes the sum of the mean percentage differences over all tests; the results are then shown on maps from the test set. In the high-resolution case only, we actually found that the trained model can generate maps whose power spectrum is significantly different (more than 10 per cent) from the input and target ones, which we attribute to instabilities of the WGAN-GP framework; we show one such example in Appendix B. For this reason, we propose a ranking system that takes all the predictions from the test set, and orders them based on the mean difference between the input power spectrum and the predicted power spectrum, since they must match by construction. We select the best 100 maps according to this metric, and discuss possible ways to make the model more stable in Section 5. We envision that in a realistic scenario where an arbitrary number of lognormal maps can be generated with the
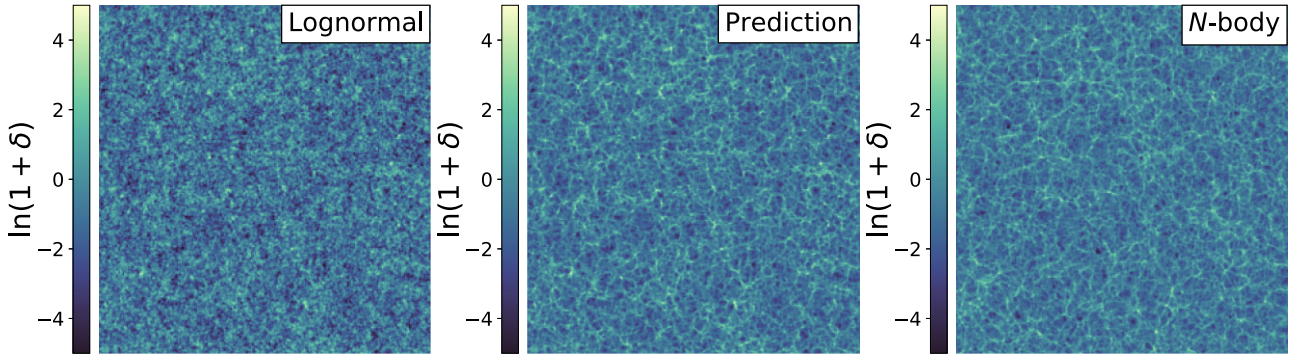
Figure 4. The lognormal (left) and *N*-body (right) density fields as in Fig. 1, with the prediction of our model (middle) given the lognormal field. In these maps, we clipped the maximum and minimum values before applying the logarithm to reduce their dynamic range. The model is described in Section 3.2. We remark that we are not interested in an exact match of the middle and right-hand panels, as we explain in Section 4.1, and thoroughly test that the predicted fields carry the same statistical information as the *N*-body maps from Section 4.2.

goal of producing *m* augmented fields, one could iteratively generate augmented lognormal maps through our model and discard those whose precision is below a desired threshold, until all *m* maps are produced.

We show the results of our best models in the next section. These models are found with $\lambda_1 = 100$ and $\lambda_2 = 10$; we defer a full grid search over these hyperparameters to future work.

4 RESULTS

In this section, we validate the performance of the trained model by comparing the statistics of the generated and simulated maps. In Appendix C, we also show that our model is not affected by mode collapse.

4.1 Qualitative comparison

While the appearance of the maps is irrelevant for the purpose of our statistical analysis, a visual inspection is none the less useful to intuitively understand whether our model is on the right track to learn the *N*-body features. In Fig. 4, we show a lognormal map, its *N*-body counterpart, and the prediction of our model given the lognormal map for the high-resolution case.

Our goal is not to obtain an exact visual match between the model's prediction and the *N*-body map, given that we used the random phases of the $z = 127$ simulations, which only have partial correlations with the $z = 0$ slices. For the applications, we focus on (discussed in Section 5), the actual position of peaks and voids in the lognormal map is irrelevant, since it is dictated by the random sampling of the phases: We only aim to generate maps which carry the same statistical signal as the *N*-body maps on average, improving on the lognormal approximation. We observe that while the predicted field does not match the *N*-body pattern pixel by pixel, the model has learnt the correct morphology of the large-scale structure on top of the lognormal field.

4.2 Statistics

While visual inspection of the generated maps against the target ones is a necessary zeroth-order test to provide intuition on whether the model was adequately trained, it is then fundamental to compare the summary statistics of interests and carefully quantify their agreement. We compare the generated and simulated fields through four different summary statistics, which we briefly describe here.

4.2.1 Pixel counts

The first test consists of binning the pixels of the generated and target density fields into a histogram. While the lognormal distribution is a good approximation of the simulated fields, there is a significant difference between the two (see e.g. Fig. 1). We show in panel (a) of Figs 5 and 6 the performance of our model with respect to the pixel counts for high and low resolution, respectively.

4.2.2 Power spectrum

We compare the power spectrum as defined in equation (4) for the simulated maps and the ones predicted by our model given the lognormal maps. While it could be argued that this is a trivial task (given that the input and output maps have the same power spectrum by construction), it is not obvious that our model does not modify the power spectrum information while learning the new density distribution, and as anticipated at the end of Section 3.2, we actually found some failures of the trained model which yield discrepant power spectra in the high-resolution case. We therefore compute the power spectra and show the results in panel (b) of Figs 5 and 6, for high and low resolution, respectively. Since *N*-body simulations are mainly used to associate a covariance matrix to real measurements, we have also computed power spectrum covariance matrices for all data sets and models we considered. These are very similar for all cases, and while we do not show them for brevity, they further validate the performance of our model.

4.2.3 Bispectrum

To probe the non-Gaussian features of the density fields, we measure the matter bispectrum of the maps, i.e. the counterpart of the three-point matter correlation function in Fourier space. The matter bispectrum $B(k_1, k_2, k_3)$ for a 2D field is defined implicitly as (see e.g. Sefusatti et al. 2006)

$$\xi(r) = \langle \delta(\mathbf{x})\delta(\mathbf{x} + \mathbf{r}) \rangle , \qquad (11)$$

where $\delta(\mathbf{k})$ indicates the Fourier transform of the matter overdensity $\delta(\mathbf{x})$, $k_i = |\mathbf{k}_i|$, and all $\mathbf{k}_i$ vectors are in the plane of the simulation box slices. To further assess that our model correctly captured the information beyond the power spectrum, we also measure the reduced matter bispectrum [$Q(k_1, k_2, k_3)$, see e.g. Liguori et al. 2010], defined
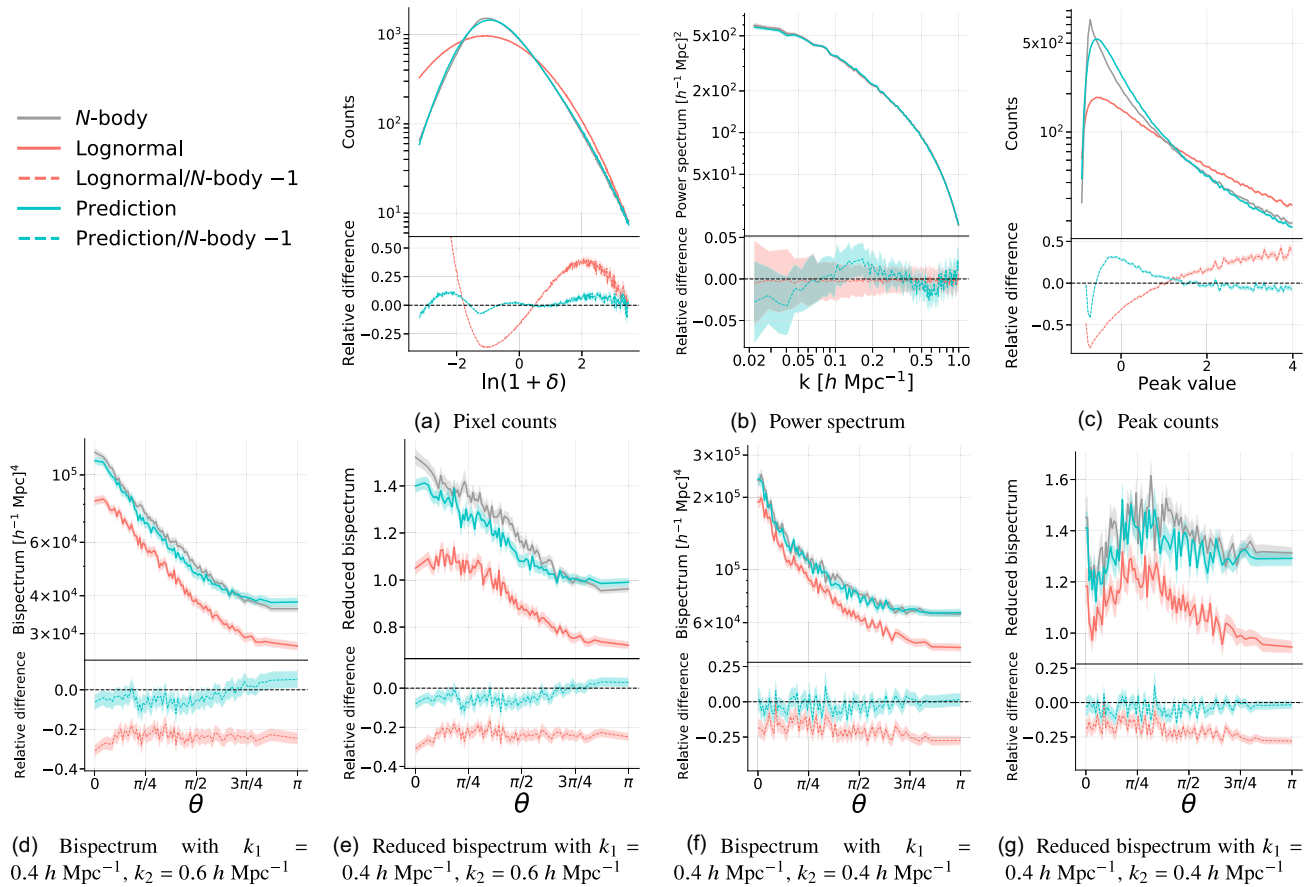
Figure 5. Comparison of the statistical tests described in Section 4.2 for the lognormal (δ_{LN}, in red), N-body (δ_{SIM}, in grey), and predicted (δ_{GEN}, in cyan) maps, considering a resolution of $N_{high} = 512$. The performance is measured at the bottom of each panel by calculating the relative difference of N-body against predicted and lognormal maps (dashed lines). All solid lines indicate the mean values over 100 maps, and the error bars represent the error on the mean (or propagated error, in the case of the relative differences). We observe that, except for the range $\delta < 0$ in panel (a) and (c) and some individual θ values in panels (d)–(g), the prediction always matches the target statistics within the error bars, performing significantly better than the lognormal approximation.

as:

$$Q(k_1, k_2, k_3) = \frac{B(k_1, k_2, k_3)}{P(k_1)P(k_2) + P(k_1)P(k_3) + P(k_2)P(k_3)}. \quad (12)$$

We measure bispectra and reduced bispectra for different configurations depending on the resolution; different triangle configurations usually probe different inflationary models (Liguori et al. 2010), and one must include as many configurations as possible to break degeneracies when inferring cosmological parameters (Bergé, Amara & Réfrégier 2010). Moreover, different bispectra configurations can shed light on the size of collapsing regions, as well as on the relative position of clusters and voids in the large-scale structure (Munshi et al. 2020).

We calculate bispectra and reduced bispectra based on an estimator of the binned bispectrum (see e.g. Bucher, Racine & van Tent 2016); we consider the centroid of each bin as the value of $\boldsymbol{k}$ at which the bispectrum is evaluated. We report the results in panels (d)–(g) of Figs 5 and 6 (for high and low resolution, respectively) as a function of the angle θ between the vectors $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$.

4.2.4 Peak counts

To further assess whether the model has correctly learnt the most non-Gaussian features of the simulated density fields, we verify that the peak counts of the generated and target maps match within the error bars. A peak is defined as a density pixel which is higher than the 8 surrounding pixels. Peak count statistics have been shown to carry significant cosmological information, especially in weak lensing studies, as they trace the most dense regions (Pires et al. 2009; Pires, Leonard & Starck 2012; Dietrich & Hartlap 2010; Marian et al. 2011; Mainini & Romano 2014; Lin & Kilbinger 2015a, b; Lin, Kilbinger & Pires 2016; Kacprzak et al. 2016; Shan et al. 2018; Martinet et al. 2018; Harnois-Déraps et al. 2021). We bin the peak values for both the simulated and target maps, and compare them in panel (c) of Figs 5 and 6, for high and low resolution, respectively.

4.3 High resolution

In Fig. 5, we compare the performance of the predictions of our model against the target maps, for the case with 512^2 pixels. We run the statistical tests on 100 maps sampled from the test set as described at the end of Section 3.2; the solid lines show the mean values and the dashed areas represent the error on the mean.

In panel (b), we show that the trained model is capable of preserving the correct power spectrum on all scales from $0.025\,h\,\mathrm{Mpc}^{-1}$ to $1\,h\,\mathrm{Mpc}^{-1}$, with percentage differences going no higher than 3 per cent, and always within the error bars. At the same time, the model improves on the lognormal approximation as far as the pixel counts and peak counts are concerned, with however significant differences in particular for $\delta < 0$ in the latter case. We believe
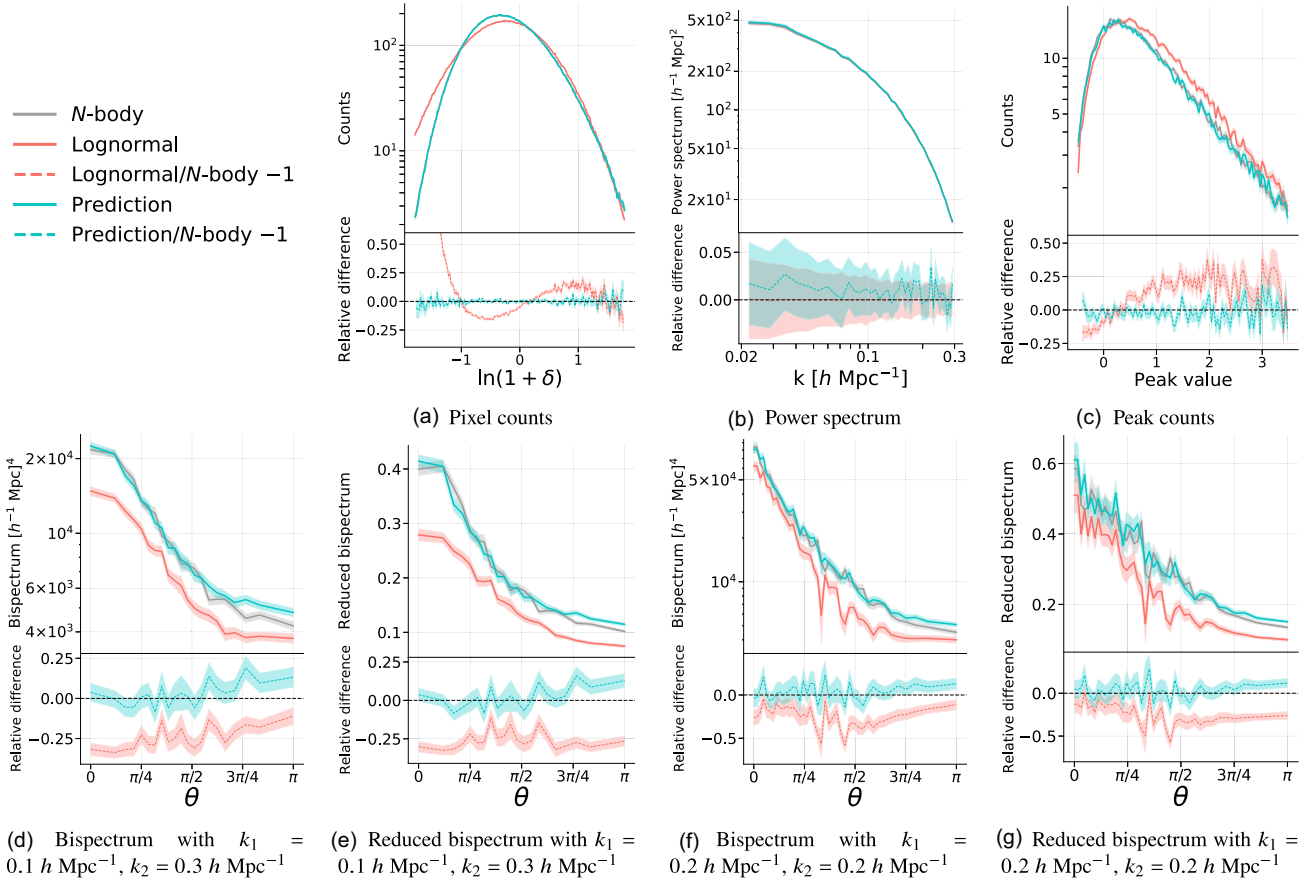
Figure 6. Same as Fig. 5, with a lower field resolution $N_{low} = 128$. The solid lines indicate the mean values over 100 maps. We observe that the model's performance is almost always within the 5 per cent range, except for the bispectra, where significant differences are present at high θ; we discuss these discrepancies in Section 4.4. Despite these differences, our model still outperforms the lognormal approximation.

that the performance in this case could be ameliorated by e.g. exploring different network architectures. In panels (d)–(g), we show the results for the (reduced) bispectrum, for $k_1 = 0.4 \, h \, \text{Mpc}^{-1}$, $k_2 = 0.6 \, h \, \text{Mpc}^{-1}$ – panels (d) and (e) – and for $k_1 = 0.4 \, h \, \text{Mpc}^{-1}$, $k_2 = 0.4 \, h \, \text{Mpc}^{-1}$ – panels (f) and (g). The performance is very good overall, with the percentage difference between target and predicted maps being within the error bars except for a few individual values of θ, significantly improving on the lognormal approximation.

4.4 Low resolution

In Fig. 6, we compare the performance of the predictions of our model against the target maps, for the case with 128^2 pixels. We use 100 randomly sampled maps from the test set. We observe good agreement between predicted and target maps for the pixel counts, power spectrum, and peak counts statistics, with the power spectrum in particular being almost always within 2 per cent. As far as the bispectra are concerned, we consider two configurations, one with $k_1 = 0.1 \, h \, \text{Mpc}^{-1}$, $k_2 = 0.3 \, h \, \text{Mpc}^{-1}$ – panels (d) and (e) – and for $k_1 = 0.2 \, h \, \text{Mpc}^{-1}$, $k_2 = 0.2 \, h \, \text{Mpc}^{-1}$ – panels (f) and (g). Since the fields have a lower resolution, the scales we probe are larger than in Section 4.3. We observe a good agreement overall, except at high θ: We argue that to improve the performance of the model, we could use the same ranking approach based on the power spectrum as in the high-resolution case.

4.5 Redshift and cosmology dependence

So far, we have shown the performance of our model on a given fiducial set of cosmological parameters and redshift, from which the training data were obtained. However, in order for the method to become practical, it is critical to assess whether the performance degrades when the model is tested on lognormal maps obtained with a different cosmology or at different z. Examples of good generalization properties of machine learning models applied to cosmological problems include He et al. (2019), Kaushal et al. (2021), Shao et al. (2022).

We checked that the performance of our model does not degrade much when acting on fields with slightly different (within 2 per cent) values of Ω_m and σ_8, even though a more complete analysis on bigger variations is required. We additionally verified that feeding our model, trained using maps at $z = 0$, with lognormal maps at $z = 0.5$ or $z = 1.0$ does not yield satisfactory results, with percentage errors going well above 50 per cent. This failure is not unexpected: The different dynamic range of the lognormal maps at different redshifts highlights that our model is not capable of extrapolating to such different input values.

To overcome these limitations, we use the Quijote simulations run on a latin hypercube of the cosmological parameters, which are publicly available together with the simulations run at the fiducial cosmology (Villaescusa-Navarro et al. 2020). We consider 800 of such simulations at low resolution $N_{low} = 128$ and $z = 0$, and repeat the procedure described in Section 3.1 to generate a data set of
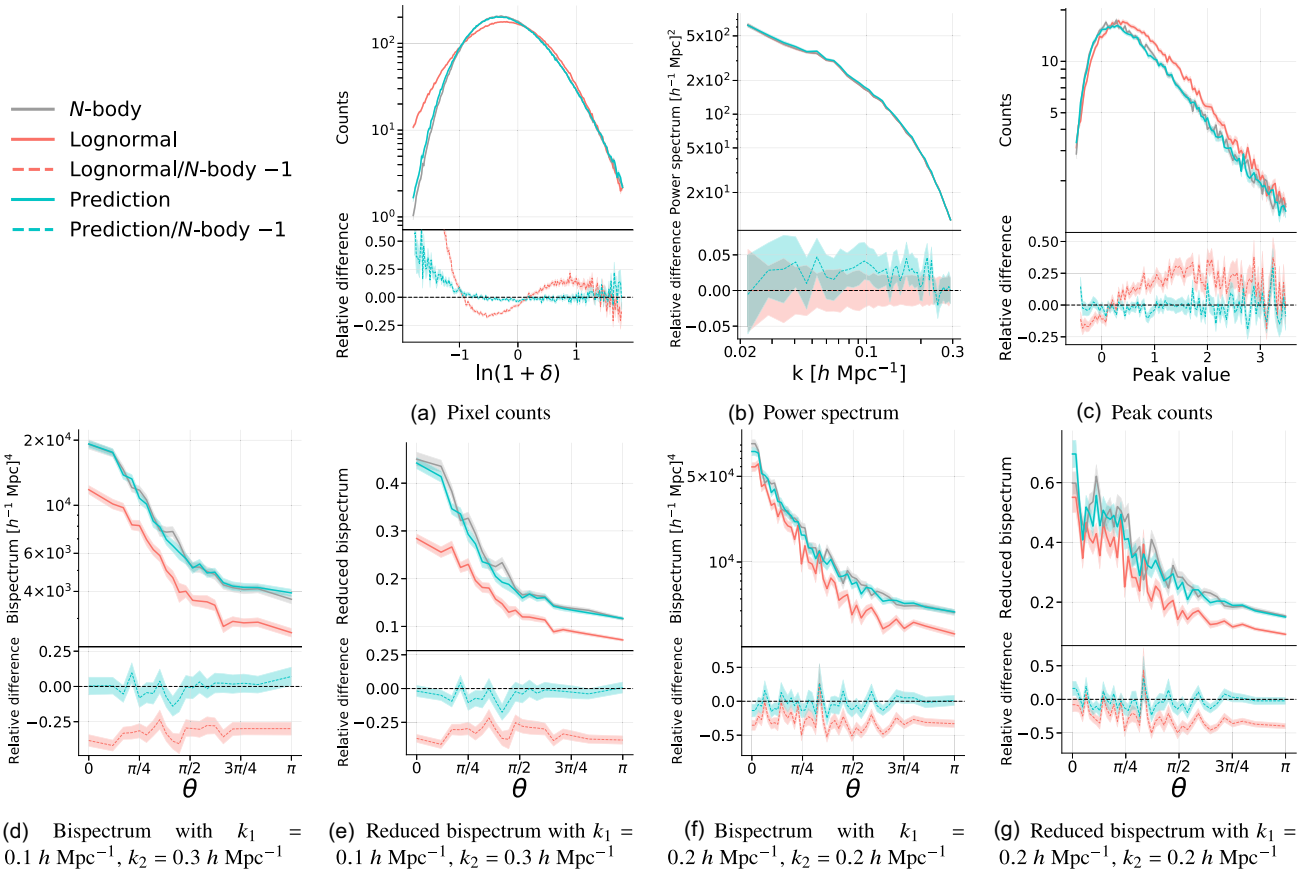
Figure 7. Same as Fig. 6, but for a model trained on latin-hypercube simulations and applied to a different cosmology, as described in Section 4.5. Except at low values of δ in panel (a), the results show good agreeement between the model predictions and the target *N*-body fields, demonstrating that our model has good generalization performance across different cosmologies.

highly correlated pairs of lognormal and *N*-body fields. For each of the first 700 simulations, we keep 90 slices for training, 19 for validation, and 19 for testing (not used in this instance), and train the same model described in Section 3.2 for 150 epochs. Note that, we do not provide explicitly the label corresponding to the different cosmological parameters during training. We select the best model according to the best performance on the summary statistics over the validation set, as in the fiducial case.

We test the best model by applying it to 100 lognormal maps generated at the fiducial cosmology (with results shown in Appendix D), as well as at the cosmological parameters of one random simulation from the test set ($\Omega_m = 0.1987$, $\Omega_b = 0.0446$, $h = 0.5601$, $n_s = 1.1707$, $\sigma_8 = 0.7665$) in Fig. 7. These results, while slightly worse than those presented in Fig. 6 since e.g. the power spectrum shows up to 5 per cent discrepancies, indicate that the model trained on the latin-hypercube simulations has a good generalization performance, which extends to cosmologies that were never shown at training or validation time. We plan to extend these encouraging results to higher resolutions and different redshifts in future work.

Other possible solutions to obtain a good generalization performance include normalizing the maps, either after or before training the model. For instance, one could rescale the field at $z = 1$ through the linear growth factor (Eisenstein, Hu & Tegmark 1999) to $z = 0$, and then invert this transformation after feeding the map through the generator trained as above. However, this approach would ignore the non-linear scales, and directly dividing each pixel value by the linear growth factor could lead to unphysical fields with $\delta <$

-1. Alternatively, instead of the dark matter overdensity field, we could consider the corresponding peak height field, calculated by measuring for each pixel $1.686/\sigma(M)$, where $\sigma(M)$ is the mass enclosed within a given scale; since the peak height is known to extrapolate better, having a weaker dependence on cosmological parameters (Press & Schechter 1974; Bond et al. 1991; Percival 2005; Kravtsov & Borgani 2012), we expect a model trained on this field to have an improved generalization performance.

Finally, we show that with our current set-up we can successfully train a second model on data generated as described in Section 3.1 with $z \neq 0$. We show the results for a model trained on fields at $z = 1$, which have a lower contrast and less non-linearity than $z = 0$, in Fig. 8 for the low-resolution case, with good performance overall. We also expect that it would be possible to train a conditional model by providing the redshift 'label' together with the input lognormal map, thus obtaining a conditional WGAN-GP (see e.g. Mirza & Osindero 2014; Yiu, Fluri & Kacprzak 2021); such a model could be trained e.g. on maps with $z = 0$ and $z = 1$, and then used to predict maps at $z = 0.5$, similarly to Chen et al. (2020). All these points indicate that it will be possible to make our model conditional on z and different cosmological parameters; we defer these studies to future work.

5 CONCLUSIONS

In this paper, we employed the Quijote simulations as a starting point to train a machine learning model that is capable of transforming projected lognormal realizations of the dark matter density field to
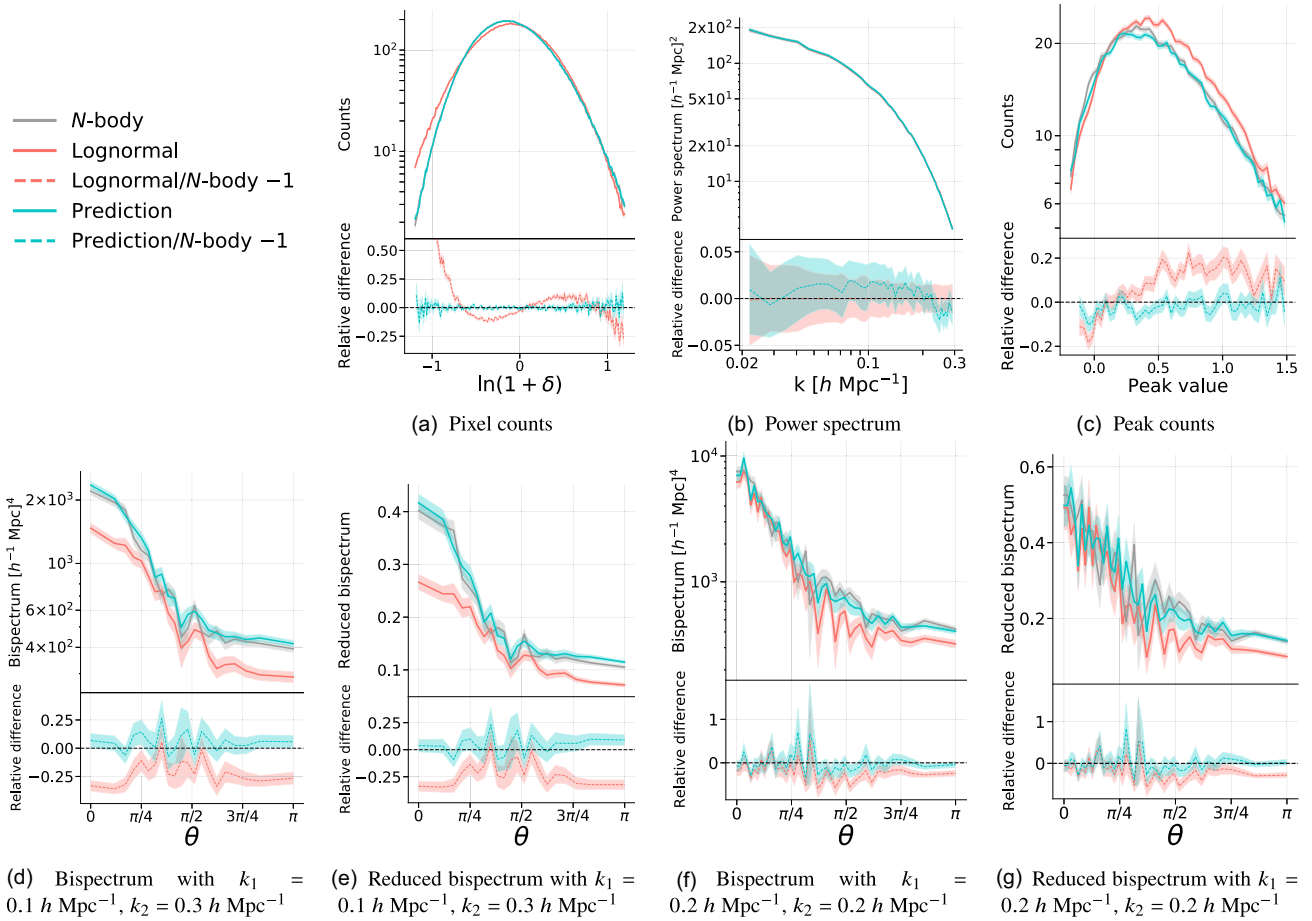
Figure 8. Same as Fig. 6, for a different model trained on data at redshift $z = 1$. We observe a good overall performance of our model, which generally outperforms the lognormal approximation.

more realistic samples of the dark matter distribution. We employed state-of-the-art image-to-image translation techniques, combining convolutional neural networks and adversarial training, to learn such a model, and extensively validated its performance through a thorough set of statistical tests. We observed a significant reduction in the error of non-Gaussian features like peak counts and bispectra, from tens of per cent for the pure lognormal model to no more than 10 per cent obtained by our model in most cases; the latter frequently shows an order of magnitude improvement over the former. Furthermore, the mapping is extremely fast, taking $\mathcal{O}(1\ s)$ on a single GPU.

In order to avoid running large suites of *N*-body simulations, the proposed method has to generalize well to other redshifts and cosmologies. We demonstrated that it is possible to train a model on simulations run over a latin hypercube of cosmological parameters and have good performance on the fiducial cosmology as well as on unseen cosmologies. We outlined a few promising avenues to investigate in order to extend these results to different redshifts and to higher resolutions. Moreover, while in this work we trained different models for different resolutions of the density field, we also expect an improved model to be able to deal with a varying slice thickness.

We plan to extend this work to random fields on the sphere, and integrate it into the FLASK package developed in Xavier et al. (2016). We aim to extend our approach to spherical random fields by iteratively applying our model to square patches of the sky, thus providing the community with a tool to quickly generate

realistic dark matter realizations that overcome the limitations of the lognormal approximation. We also plan to compare our approach to a direct generation of spherical fields by means of spherical convolutional layers, as proposed e.g. for mass maps in Yiu et al. (2021).

Additionally, we believe that the image-to-image technique outlined in this paper could be applied to augment analytical approximations to *N*-body simulations (like L-PICOLA; Howlett et al. 2015, or FastPM; Feng et al. 2016), as well as semi-analytic models of galaxies, which, in the same vein as lognormal random fields, provide a fast approximation to hydrodynamical simulations by modelling complicated baryonic processes (White & Frenk 1991; Kauffmann, White & Guiderdoni 1993; Cole et al. 1994; Somerville & Primack 1999; Lacey 2001). In such instances, one could e.g. train a model to learn the mapping between an *N*-body simulation augmented with semi-analytical models and the corresponding hydrodynamical simulation. We further plan to explore the possibility to employ the data set described in this work to reduce the variance in the statistics of large-scale structure observables using a small number of expensive simulations (Chartier et al. 2021; Chartier & Wandelt 2021; Ding et al. 2022), as well as to replace our WGAN-GP model with either a possibly more stable GAN version (Kwon et al. 2021), or with a more compact model, like the one proposed in the context of Lagrangian deep learning (LDL; Dai & Seljak 2021), using graph neural networks (GNNs; see e.g. Zhou et al. 2018 for a review) or through normalizing flows (e.g. FFJORD; Grathwohl et al. 2018, or

more recently TRENF; Dai & Seljak 2022). This will be investigated in future work.

DATA AVAILABILITY STATEMENT

The original data underlying this article (i.e. the Quijote simulations) are available through the *globus* server, as detailed in the documentation at this link. The byproduct data obtained for this work will be shared on reasonable request to the corresponding author. The code to reproduce this work is publicly available in this GitHub repository (`https://github.com/dpiras/leap_of_lognormal`, or 🔗).

REFERENCES

Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/
Allys E., Marchand T., Cardoso J. F., Villaescusa-Navarro F., Ho S., Mallat S., 2020, Phys. Rev. D, 102, 103506
Alves de Oliveira R., Li Y., Villaescusa-Navarro F., Ho S., Spergel D. N., 2020, preprint (arXiv:2012.00240)
Angulo R. E., Springel V., White S. D. M., Jenkins A., Baugh C. M., Frenk C. S., 2012, MNRAS, 426, 2046
Arjovsky M., Chintala S., Bottou L., 2017, Proceedings of the 34th International Conference on Machine Learning, vol. 70. ICML, p. 214
Barnes J. E., Hut P., 1986, Nature, 324, 446
Bergé J., Amara A., Réfrégier A., 2010, ApJ, 712, 992
Bertone G., Hooper D., Silk J., 2005, Phys. Rep., 405, 279
Blas D., Lesgourgues J., Tram T., 2011, JCAP, 2011, 034
Böhm V., Feng Y., Lee M. E., Dai B., 2021, Astron. Comput., 36, 100490
Bond J. R., Cole S., Efstathiou G., Kaiser N., 1991, ApJ, 379, 440
Boylan-Kolchin M., Springel V., White S. D. M., Jenkins A., Lemson G., 2009, MNRAS, 398, 1150
Bucher M., Racine B., van Tent B., 2016, JCAP, 2016, 055
Buchert T., 1992, MNRAS, 254, 729
Buchert T., 1994, MNRAS, 267, 811
Buchert T., Ehlers J., 1993, MNRAS, 264, 375
Buchmueller O., Doglioni C., Wang L.-T., 2017, Nat. Phys., 13, 217
Callahan P. B., Kosaraju S. R., 1992, in Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing, STOC '92. Association for Computing Machinery, New York, p. 546

Chacón J., Vázquez J. A., Gabbasov R., 2020, Rev. Mex. Fís. E, 17, 241
Chaniotis A. K., Poulikakos D., 2004, J. Comput. Phys., 197, 253
Chartier N., Wandelt B. D., 2021, MNRAS, 509, 2220
Chartier N., Wandelt B. D., Akrami Y., Villaescusa-Navarro F., 2021, MNRAS, 503, 1897
Chen C., Li Y., Villaescusa-Navarro F., Ho S., Pullen A., 2020, preprint (arXiv:2012.05472)
Chiang L.-Y., Coles P., 2000, MNRAS, 311, 809
Chuang C.-H., Kitaura F.-S., Prada F., Zhao C., Yepes G., 2014, MNRAS, 446, 2621
Cole S., Aragon-Salamanca A., Frenk C. S., Navarro J. F., Zepf S. E., 1994, MNRAS, 271, 781
Coles P., Chiang L.-Y., 2000, Nature, 406, 376
Coles P., Jones B., 1991, MNRAS, 248, 1
Dai B., Seljak U., 2021, Proc. Natl. Acad. Sci., 118, e2020324118
Dai B., Seljak U., 2022, MNRAS, 516, 2363
Dai B., Feng Y., Seljak U., 2018, JCAP, 2018, 009
Dai B., Feng Y., Seljak U., Singh S., 2020, JCAP, 2020, 002
de Swart J. G., Bertone G., van Dongen J., 2017, Nat. Astron., 1, 0059
Dietrich J. P., Hartlap J., 2010, MNRAS, 402, 1049
Ding Z. et al., 2022, MNRAS, 514, 3308
Dodelson S., 2003, Modern Cosmology. Academic Press, San Diego
Dumoulin V., Visin F., 2016, preprint (arXiv:1603.07285)
Eisenstein D. J., Hu W., Tegmark M., 1999, ApJ, 518, 2
Feder R. M., Berger P., Stein G., 2020, Phys. Rev. D, 102, 103504
Feng Y., Chu M.-Y., Seljak U., McDonald P., 2016, MNRAS, 463, 2273
Fukushima K., 1980, Biol. Cybern., 36, 193
Glorot X., Bordes A., Bengio Y., 2011, in Gordon G., Dunson D., Dudík M., eds, Proceedings of Machine Learning Research, Vol. 15, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. PMLR, Fort Lauderdale, FL, USA, p. 315
Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14. MIT Press, Cambridge, MA, USA, p. 2672
Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA, USA
Grathwohl W., Chen R. T. Q., Bettencourt J., Sutskever I., Duvenaud D., 2019, 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, https://openreview.net/forum?id=rJxgknCcK7
Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A. C., 2017, in Guyon I., von Luxburg U., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., eds, Advances in Neural Information Processing Systems. Curran Associates, Inc., p. 30
Hand N., Feng Y., Beutler F., Li Y., Modi C., Seljak U., Slepian Z., 2018, AJ, 156, 160
Harnois-Déraps J., Martinet N., Castro T., Dolag K., Giblin B., Heymans C., Hildebrandt H., Xia Q., 2021, MNRAS, 506, 1623
Harris C. R. et al., 2020, Nature, 585, 357
He S., Li Y., Feng Y., Ho S., Ravanbakhsh S., Chen W., Póczos B., 2019, Proc. Natl. Acad. Sci., 116, 13825
Hilbert S., Hartlap J., Schneider P., 2011, A&A, 536, A85
Hockney R. W., Eastwood J. W., 1988, Computer simulation using particles. Hilger, Bristol
Holmberg E., 1941, ApJ, 94, 385
Howlett C., Manera M., Percival W. J., 2015, Astron. Comput., 12, 109
Hunter J. D., 2007, Comput. Sci. Eng., 9, 90
Ioffe S., Szegedy C., 2015, Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. JMLR.org, Lille, France, p. 448
Isola P., Zhu J.-Y., Zhou T., Efros A. A., 2017, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 5967
Jenkins A. et al., 1998, ApJ, 499, 20
Jing Y. P., 2005, ApJ, 620, 559
Kacprzak T. et al., 2016, MNRAS, 463, 3653
Kauffmann G., White S. D. M., Guiderdoni B., 1993, MNRAS, 264, 201

Kaushal N., Villaescusa-Navarro F., Giusarma E., Li Y., Hawry C., Reyes M., 2021, ApJ, 930, 9

Kingma D. P., Ba J., 2015, in LeCun Y., Yoshua B., eds, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings

Kitaura F.-S., Yepes G., Prada F., 2013, MNRAS, 439, L21

Klypin A., Holtzman J., 1997, preprint(astro-ph/9712217)

Kodi Ramanah D., Charnock T., Villaescusa-Navarro F., Wandelt B. D., 2020, MNRAS, 495, 4227

Kravtsov A. V., Borgani S., 2012, ARA&A, 50, 353

Krizhevsky A., Sutskever I., Hinton G. E., 2017, Commun. ACM, 60, 84

Kwon D., Kim Y., Montúfar G., Yang I., 2021, preprint (arXiv:2110.14150)

Lacey C., 2001, in Umemura M., Susa H., eds, ASP Conf. Ser. Vol. 222, The Physics of Galaxy Formation. Astron. Soc. Pac., San Francisco, p. 273

LeNail A., 2019, J. Open Source Softw., 4, 747

Le Cun Y. et al., 1989, IEEE Commun. Mag., 27, 41

Li Y., Ni Y., Croft R. A. C., Di Matteo T., Bird S., Feng Y., 2021, Proc. Natl. Acad. Sci., 118, e2022038118

Liguori M., Sefusatti E., Fergusson J. R., Shellard E. P. S., 2010, Adv. Astron., 2010, 1

Lin C.-A., Kilbinger M., 2015a, A&A, 576, A24

Lin C.-A., Kilbinger M., 2015b, A&A, 583, A70

Lin C.-A., Kilbinger M., Pires S., 2016, A&A, 593, A88

Maas A., Hannun A., Ng A., 2013, in Proceedings of the International Conference on Machine Learning, Atlanta, Georgia

Mainini R., Romano A., 2014, JCAP, 2014, 063

Marian L., Hilbert S., Smith R. E., Schneider P., Desjacques V., 2011, ApJ, 728, L13

Martinet N. et al., 2018, MNRAS, 474, 712

Metz L., Poole B., Pfau D., Sohl-Dickstein J., 2017, in 5th International Conference on Learning Representations, ICLR, April 24, 2017. Conference Track Proceedings, Toulon, France

Mikkola S., Aarseth S. J., 1993, Celest. Mech. Dyn. Astron., 57, 439

Mirza M., Osindero S., 2014, preprint (arXiv:1411.1784)

Monaco P., Theuns T., Taffoni G., Governato F., Quinn T., Stadel J., 2002, ApJ, 564, 8

Monaco P., Sefusatti E., Borgani S., Crocce M., Fosalba P., Sheth R. K., Theuns T., 2013, MNRAS, 433, 2389

Munshi D., Namikawa T., Kitching T. D., McEwen J. D., Takahashi R., Bouchet F. R., Taruya A., Bose B., 2020, MNRAS, 493, 3985

Mustafa M., Bard D., Bhimji W., Lukić Z., Al-Rfou R., Kratochvil J. M., 2019, Comput. Astrophys. Cosmology, 6, 1

Navarro J. F., Frenk C. S., White S. D. M., 1996, ApJ, 462, 563

O'Shea B. W., Bryan G., Bordner J., Norman M. L., Abel T., Harkness R., Kritsuk A., 2004, preprint(astro-ph/0403044)

Papamakarios G., Nalisnick E., Jimenez Rezende D., Mohamed S., Lakshminarayanan B., 2021, Journal of Machine Learning Research, 22, 1

Peebles P. J. E., 1993, Principles of Physical Cosmology. Princeton Univ. Press, Princeton

Percival W. J., 2005, A&A, 443, 819

Percival W. J., Verde L., Peacock J. A., 2004, MNRAS, 347, 645

Perraudin N., Srivastava A., Lucchi A., Kacprzak T., Hofmann T., Réfrégier A., 2019, Comput. Astrophys. Cosmology, 6, 5

Pires S., Starck J.-L., Amara A., Réfrégier A., Teyssier R., 2009, A&A, 505, 969

Pires S., Leonard A., Starck J.-L., 2012, MNRAS, 423, 983

Press W. H., Schechter P., 1974, ApJ, 187, 425

Rizzo L. A., Villaescusa-Navarro F., Monaco P., Munari E., Borgani S., Castorina E., Sefusatti E., 2017, JCAP, 2017, 008

Rodríguez A. C., Kacprzak T., Lucchi A., Amara A., Sgier R., Fluri J., Hofmann T., Réfrégier A., 2018, Comput. Astrophys. Cosmology, 5, 11

Ronneberger O., Fischer P., Brox T., 2015, Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, vol. 9351, Springer, p. 234

Rouhiainen A., Giri U., Münchmeyer M., 2021, preprint (arXiv:2105.12024)

Sefusatti E., Crocce M., Pueblas S., Scoccimarro R., 2006, Phys. Rev. D, 74, 023522

Sefusatti E., Crocce M., Scoccimarro R., Couchman H. M. P., 2016, MNRAS, 460, 3624

Shan H. et al., 2018, MNRAS, 474, 1116

Shao H. et al., 2022, ApJ, 927, 85

Somerville R. S., Primack J. R., 1999, MNRAS, 310, 1087

Springel V., 2005, MNRAS, 364, 1105

Springel V. et al., 2005, Nature, 435, 629

Taruya A., Takada M., Hamana T., Kayo I., Futamase T., 2002, ApJ, 571, 638

Tassev S., Zaldarriaga M., Eisenstein D. J., 2013, JCAP, 2013, 036

Tassev S., Eisenstein D. J., Wandelt B. D., Zaldarriaga M., 2015, preprint (arXiv:1502.07751)

Taylor A., Joachimi B., 2014, MNRAS, 442, 2728

Taylor A., Joachimi B., Kitching T., 2013, MNRAS, 432, 1928

Tormen G., 1997, MNRAS, 290, 411

Tosone F., Neyrinck M. C., Granett B. R., Guzzo L., Vittorio N., 2020, MNRAS, 498, 2663

Tosone F., Neyrinck M. C., Granett B. R., Guzzo L., Vittorio N., 2021, MNRAS, 505, 2999

Trimble V., 1987, ARA&A, 25, 425

Villaescusa-Navarro F. et al., 2020, ApJS, 250, 2

Villaescusa-Navarro F. et al., 2021, ApJ, 915, 71

Watts P., Coles P., Melott A., 2003, ApJ, 589, L61

White S. D. M., Frenk C. S., 1991, ApJ, 379, 52

White M., Tinker J. L., McBride C. K., 2013, MNRAS, 437, 2594

Xavier H. S., Abdalla F. B., Joachimi B., 2016, MNRAS, 459, 3693

Yamaguchi K., Sakamoto K., Akabane T., Fujimoto Y., 1990, in The First International Conference on Spoken Language Processing, ICSLP, November 18-22, 1990. ISCA, Kobe, Japan

Yiu T. W. H., Fluri J., Kacprzak T., 2021, JCAP, 2022, 40

Zhou J. et al., 2020, AI Open, 1, 57

APPENDIX A: MODEL ARCHITECTURE

In its basic formulation, a layer in a convolutional neural network (CNN; see e.g. Fukushima 1980; Krizhevsky, Sutskever & Hinton 2017) is made of a certain number of square filters, each associated to learnable parameters, usually called weights. During training, each filter is convolved through each input data point: This means that the dot product of the learnable weights and the input pixels is calculated, representing a single output for that particular filter. Repeating this operation while moving the filter across the input data creates an

Table A1. Size of each layer's output in the generator and the critic neural networks, detailed in Section 3.2 and Appendix A, for the high-resolution case. The low-resolution architecture is built analogously.

	Size	Comments
Generator	$512 \times 512 \times 1$	Input size; δ_{LN}
	$256 \times 256 \times 64$	
	$128 \times 128 \times 128$	
	$64 \times 64 \times 256$	
	$32 \times 32 \times 512$	
	$64 \times 64 \times 512$	First upsampling step
	$128 \times 128 \times 256$	
	$256 \times 256 \times 128$	
	$512 \times 512 \times 64$	
	$512 \times 512 \times 1$	Linear activation; δ_{GEN}
Critic	$512 \times 512 \times 1$	Input size; either δ_{SIM} or δ_{GEN}
	$256 \times 256 \times 32$	
	$128 \times 128 \times 64$	
	$64 \times 64 \times 128$	
	$63 \times 63 \times 512$	
	$62 \times 62 \times 1$	
	3844	Flattening; input to dense layer
	1	

output map, which is then passed through an activation function to introduce non-linearities in the network. This operation is done for multiple filters, and each output map becomes the input to the following convolutional layer. Stacking convolutional layers allows one to extract progressively larger scales from the input data, and represents a more efficient implementation of a neural network with respect to standard dense layers when dealing with high-dimensional data like images (Le Cun et al. 1989; Goodfellow, Bengio & Courville 2016).

As anticipated, our model, depicted in Fig. 3, consists of two neural networks. The first neural network (the generator) contains four downsampling blocks, followed by four upsampling blocks. Each downsampling block first pads the input data assuming periodic boundary conditions, and then applies a convolution operation with 4x4 filters. There are 64 convolutional filters in the first place, and this number doubles for each block. Note that no pooling layers are present (Yamaguchi et al. 1990), and we are able to reduce the dimensionality of the extracted feature maps by shifting each filter by two pixels in both directions; in other words, we set a stride of 2. The compressed map is then symmetrically upsampled using the transposed convolution operation (Dumoulin & Visin 2016). At each block, each feature map is concatenated with the corresponding downsampled map by simply stacking them along the last spatial axis; this is done in order to better learn the representations at each level (Ronneberger et al. 2015). The activation function used after each downsampling layer is the rectified linear unit (ReLU; Glorot,

Bordes & Bengio 2011), while for the upsampling blocks, we found the leaky ReLU (Maas, Hannun & Ng 2013) with $\alpha = 0.3$ to perform better. A final convolutional layer with linear activation function outputs the generated map δ_{GEN}. Note that all downsampling and upsampling blocks include batch normalization (Ioffe & Szegedy 2015), which during training subtracts the batch mean and divides by the batch standard deviation, in order to make the training procedure more stable. The second neural network is done similarly, with three downsampling blocks followed by two convolutional layers with Leaky ReLU as the activation function, and a final dense layer with a single output and a linear activation function. Input and output shapes for each layer are reported in Table A1. We implement our neural networks in TENSORFLOW (Abadi et al. 2015), and will make the trained models available upon acceptance of this work.

APPENDIX B: HIGH-RESOLUTION MODEL FAILURES

In Fig. B1, we show an example field in the high-resolution case for which the prediction has a power spectrum with an average 20 per cent disagreement with respect to the expected one. We attribute such problems to instabilities in the WGAN-GP model we considered, and describe a possible ranking system that addresses this problem in Section 3.2.
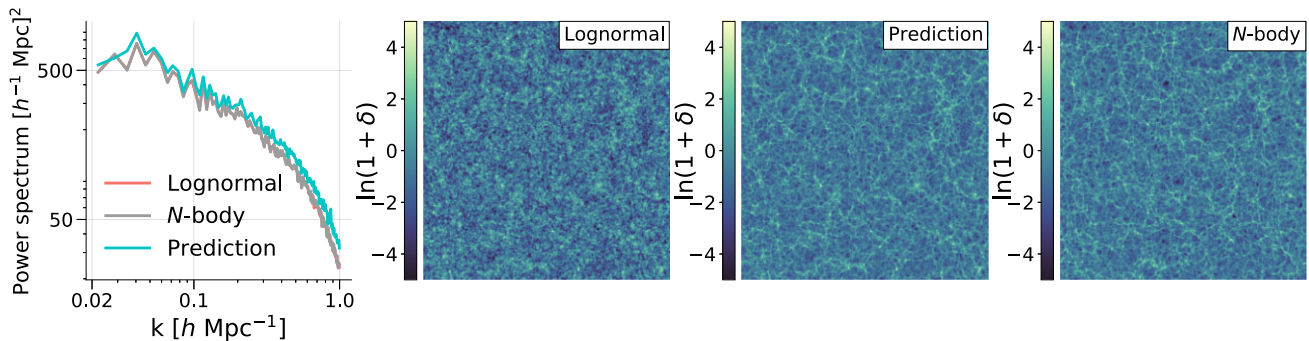

Figure B1. High-resolution example where the power spectrum, shown in the left-hand panel for the three fields in the other panels, is significantly in disagreement (more than 20 per cent) between the prediction (cyan) and the *N*-body (grey). The lognormal power spectrum (red) agrees with the *N*-body one by construction. Given these failure modes, we employed the ranking system described in Section 3.2 to select the predictions whose power spectrum agrees with their lognormal counterpart.

APPENDIX C: MODE COLLAPSE

As we explained in Section 1 and Section 3.2, our choice of providing the model a lognormal field as input, as well as the choice of the WGAN-GP loss function, should prevent the model from memorizing the training set, or only focus on single modes of the data. Following Mustafa et al. (2019), we provide evidence that the generator is capable of producing diverse maps and is not affected by mode collapse.

In Fig. C1, we show three random predictions from the test set, and the closest map in the training set according to pixel-wise distance; we focus on low-resolution data only to limit the computational cost.

Despite showing similar texture, and having summary statistics in agreement (not shown for brevity), the maps are clearly different, thus confirming that our model is immune to mode collapse.

APPENDIX D: LATIN-HYPERCUBE MODEL APPLIED TO FIDUCIAL COSMOLOGY

In Fig. D1, we show the summary statistics results for the model trained on latin-hypercube simulations and applied to fields at the fiducial cosmology, as described in Section 4.5. The redshift is fixed at $z = 0$.
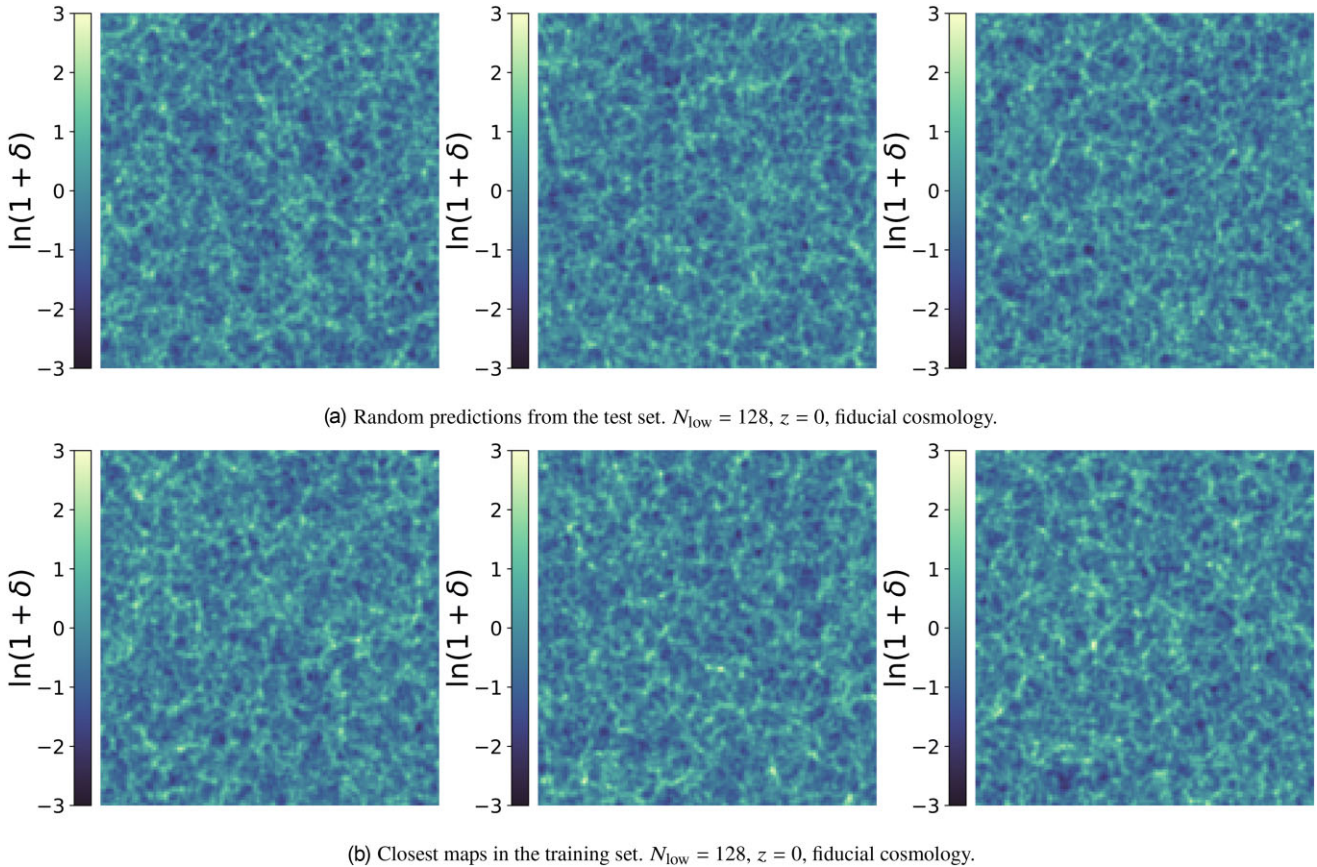

(a) Random predictions from the test set. $N_{\text{low}} = 128$, $z = 0$, fiducial cosmology.

(b) Closest maps in the training set. $N_{\text{low}} = 128$, $z = 0$, fiducial cosmology.

Figure C1. *First row*: three random predictions from the test set, in the low-resolution case at $z = 0$. *Second row*: each panel represents the field in the training set which is closest to the corresponding panel in the first row according to pixel-wise distance. As can be seen, while the maps have the same texture (and summary statistics in agreement, not shown here for brevity), they are visually different, thus confirming that our model is not affected by mode collapse.
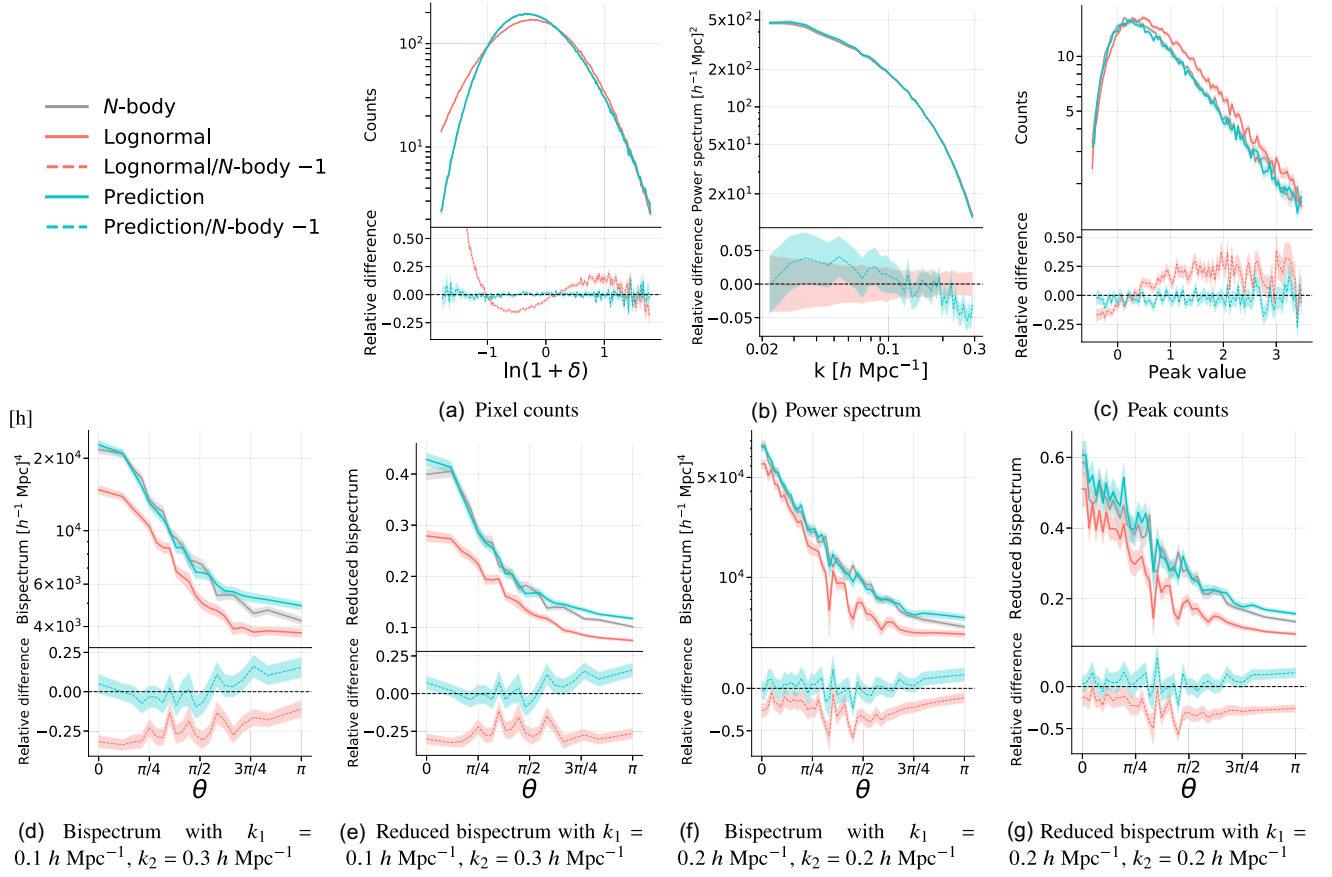
(a) Pixel counts

(b) Power spectrum

(c) Peak counts

(d) Bispectrum with $k_1 = 0.1\,h\,\mathrm{Mpc}^{-1}$, $k_2 = 0.3\,h\,\mathrm{Mpc}^{-1}$

(e) Reduced bispectrum with $k_1 = 0.1\,h\,\mathrm{Mpc}^{-1}$, $k_2 = 0.3\,h\,\mathrm{Mpc}^{-1}$

(f) Bispectrum with $k_1 = 0.2\,h\,\mathrm{Mpc}^{-1}$, $k_2 = 0.2\,h\,\mathrm{Mpc}^{-1}$

(g) Reduced bispectrum with $k_1 = 0.2\,h\,\mathrm{Mpc}^{-1}$, $k_2 = 0.2\,h\,\mathrm{Mpc}^{-1}$

Figure D1. Same as Fig. 6, but in this instance, the model was trained on latin-hypercube simulations, as described in Section 4.5. Despite a small degradation in performance, visible e.g. at high k for the power spectrum, there is good agreement overall, further validating our approach.

This paper has been typeset from a TEX/LATEX file prepared by the author.