MACHINE LEARNING
Science and Technology

PURPOSE-LED PUBLISHING™

Part of the MACHINE LEARNING Series

PAPER • OPEN ACCESS

Conditional Diffusion-Flow models for generating 3D cosmic density fields: applications to *f*(*R*) cosmologies

To cite this article: Julieth K Riveros *et al* 2025 *Mach. Learn.: Sci. Technol.* **6** 035031

View the article online for updates and enhancements.

IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 035031 https://doi.org/10.1088/2632-2153/adf8b1

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
27 March 2025

REVISED
7 July 2025

ACCEPTED FOR PUBLICATION
5 August 2025

PUBLISHED
14 August 2025

PAPER

Conditional Diffusion-Flow models for generating 3D cosmic density fields: applications to $f(R)$ cosmologies

Julieth K Riveros[1], **Paola A Saavedra**[1,2], **Héctor J Hortúa**[1,3,4,*], **Jorge Enrique García-Farieta**[5] **and Ivan Olier**[4]

[1] Laboratorio de Inteligencia Artificial (SavIA Lab), Grupo Signos, Departamento de Matemáticas, Universidad El Bosque, Bogotá, Colombia
[2] Dirección de Ciencias Básicas, Universidad Santo Tomás, Seccional Villavicencio, Villavicencio, Colombia
[3] Instituto de Neurociencias, Universidad El Bosque, Bogotá, Colombia
[4] Data Science Research Centre, Liverpool John Moores University, 3 Byrom Street, Liverpool L3 3AF, United Kingdom
[5] Departamento de Física, Universidad de Córdoba, E-14071 Córdoba, Spain
[*] Author to whom any correspondence should be addressed.

E-mail: H.J.HortuaOrjuela@ljmu.ac.uk

Keywords: cosmology: large-scale structure of Universe, methods: statistical, methods: numerical, diffusion models

Supplementary material for this article is available online

Abstract

Next-generation galaxy surveys promise unprecedented precision in testing gravity at cosmological scales. However, realising this potential requires accurately modelling the non-linear cosmic web. We address this challenge by exploring conditional generative modelling to create 3D dark matter density fields via score-based (diffusion) and flow-based methods. Our results demonstrate the power of diffusion models to accurately reproduce the matter power spectra and bispectra, even for unseen configurations. They also offer a significant speed-up with slightly reduced accuracy, when flow-based reconstructing the probability distribution function, but they struggle with higher-order statistics. To improve conditional generation, we introduce a novel multi-output model to develop feature representations of the cosmological parameters. Our findings offer a powerful tool for exploring deviations from standard gravity, combining high precision with reduced computational cost, thus paving the way for more comprehensive and efficient cosmological analyses.

1. Introduction

The advent of precision cosmology marks a new era in our understanding of the Universe driven by a variety of upcoming missions. Among the key ongoing and forthcoming efforts are the Dark Energy Spectroscopic Instrument (DESI) [1], the Euclid space mission [2, 3], the Legacy Survey of Space and Time (LSST) [4], the Wide-Field Infrared Survey Telescope (WFIRST) [5], and the Square Kilometre Array (SKA) [6]. These experiments aim to provide unprecedented measurements that will constrain cosmological parameters with high precision, with N-body simulations being a crucial component of these efforts. N-body simulations are, in fact, essential for accurately modelling the large-scale structure of the Universe, understanding the evolution of cosmic fields, and interpreting the data from these surveys. N-body simulations play a fundamental role in understanding the physics behind galaxy survey data, as they enable the exploration of cosmic structures across a range of scales. Although the non-linear regime of structure formation can, in principle, be approximated by perturbative-based methods, e.g. standard perturbation theory (see [7–9]), renormalised perturbation theory (see [10, 11]) or effective field theory (see [12–15]), there is currently no single, universally adopted framework. Simulations not only offer insights into the small-scale behaviour of galaxy clustering but also provide a reliable means of investigating the clustering in cosmologies beyond the Λ-Cold Dark Matter (ΛCDM) model. As a result, simulations have become increasingly crucial in exploring modified gravity models, with $f(R)$ models highlighting a minimal but significant modification of Einstein's general relativity (GR). Since deviations from GR are likely to manifest in summary statistics, accurately predicting these statistics is crucial for connecting to theoretical models of structure formation. Despite their advantages, simulations can be computationally expensive, depending on their complexity, such as the richness of physical phenomena included and the resolution of mass and scale. This has led to the rise of emulators based on deep learning algorithms that are designed to quickly and accurately predict cosmological observables. In particular, generative models have emerged as a promising tool in cosmology, especially for enhancing and accelerating the analysis of simulations. Recent advancements in generative models offer a powerful approach to efficiently approximate and generate maps where their summary statistics closely mirror the ones obtained from simulated data. By learning the underlying distribution of the complex, high-dimensional cosmic structures, generative models can potentially provide faster and more scalable solutions while maintaining accuracy, opening new avenues for both theoretical and observational cosmology. Generative models have been widely used in astronomy, such as U-Net/V-Net [16], styled neural network [17], GAN with autoencoders [18, 19], only autoencoders [20, 21], normalising flows [22–24], normalising flows applied in HI Maps [25], Psi-GAN [26], GANs [27–29], conditional GANs [30] GAN using spherical maps [31, 32], GANs emulators as Using Latent Space Interpolation [33], faster emulators [34–36] and emulators for cross-domain cosmological problems [37–39], have demonstrated to be powerful tools for simulating samples from complex probability distributions (for a comprehensive review, see [40, 41]). Recently, the potential of diffusion models has gained increasing attention, with several studies exploring their application in cosmology for emulating satellite galaxy and subhalo populations [42], field emulation and parameter inference [43], emulators [44, 45] and image generation [46–48] among others. Diffusion models have gained significant attention due to their effectiveness in generating high-quality samples [49, 50]. These models define a forward diffusion (noising) process that gradually transforms samples from the target distribution into samples from a standard normal distribution. The reverse diffusion process, which is learned during training, is equivalent to learning the data score, which is why diffusion models are also referred to as score-based generative models. This framework has demonstrated remarkable success in photorealistic image generation, as exemplified by Stable Diffusion, and addresses some of the key limitations of GANs, such as mode collapse—where the model fails to capture all modes of the distribution, see e.g. [51]. In this work, we apply diffusion models to generate cold-dark-matter 3D-density fields of modified $f(R)$ gravity conditioned on cosmological parameters. By leveraging conditional diffusion models, we demonstrate their ability to emulate fast and accurate full 3D density fields while maintaining consistency with the summary statistics, all at a low computational cost, with an accuracy similar to state-of-the-art N-body simulations of modified gravity models.

The outline of this paper is as follows. In section 2, we describe three diffusion methods employed in this research and illustrate the algorithms used for training and deploying the models. Section 3 introduces the modified gravity simulations used for training and evaluating the trained-models, and section 4 we detail our methodology including the neural network, the use of representation learning for including the cosmological parameters as conditioned on the generative models, and the n-point statistics for evaluating the performance of the models. In section 5, we present the results of the predicted observables for the different methods implemented and assess their performance. Finally, in sections 6 and 7, we discuss the results and provide the main conclusions of this research.

2. Preliminaries

This section outlines the basis for conditional diffusion models as emulators for N-body simulations. In addition, we introduce several strategies developed to implement diffusion model flavours.

2.1. Denoising diffusion models

Diffusion probabilistic models (DPMs) have rapidly gained prominence as a highly promising generative technique in recent years. Functioning as latent variable models for sequence modelling, DPMs utilise a latent space with the same dimensionality as the input data. In contrast to generative adversarial networks (GANs) [52], which are not probabilistic models, DPMs offer significant advantages, including excellent parallelisation capabilities and avoidance of adversarial training [51]. This eliminates the well-known challenges of debugging and convergence difficulties frequently encountered with GAN training.

2.2. Denoising DPM (DDPM)

Generative models aim to learn and approximate complex, high-dimensional data distributions [53]. Among these, probabilistic diffusion models have recently emerged as a powerful technique, distinguished by their capacity to transform unstructured noise into highly detailed and structured outputs that closely resemble the training data distribution [51]. This is achieved through a two-stage process: a *forward diffusion process* that progressively adds noise to the data, gradually corrupting its structure, and a *reverse diffusion process* that

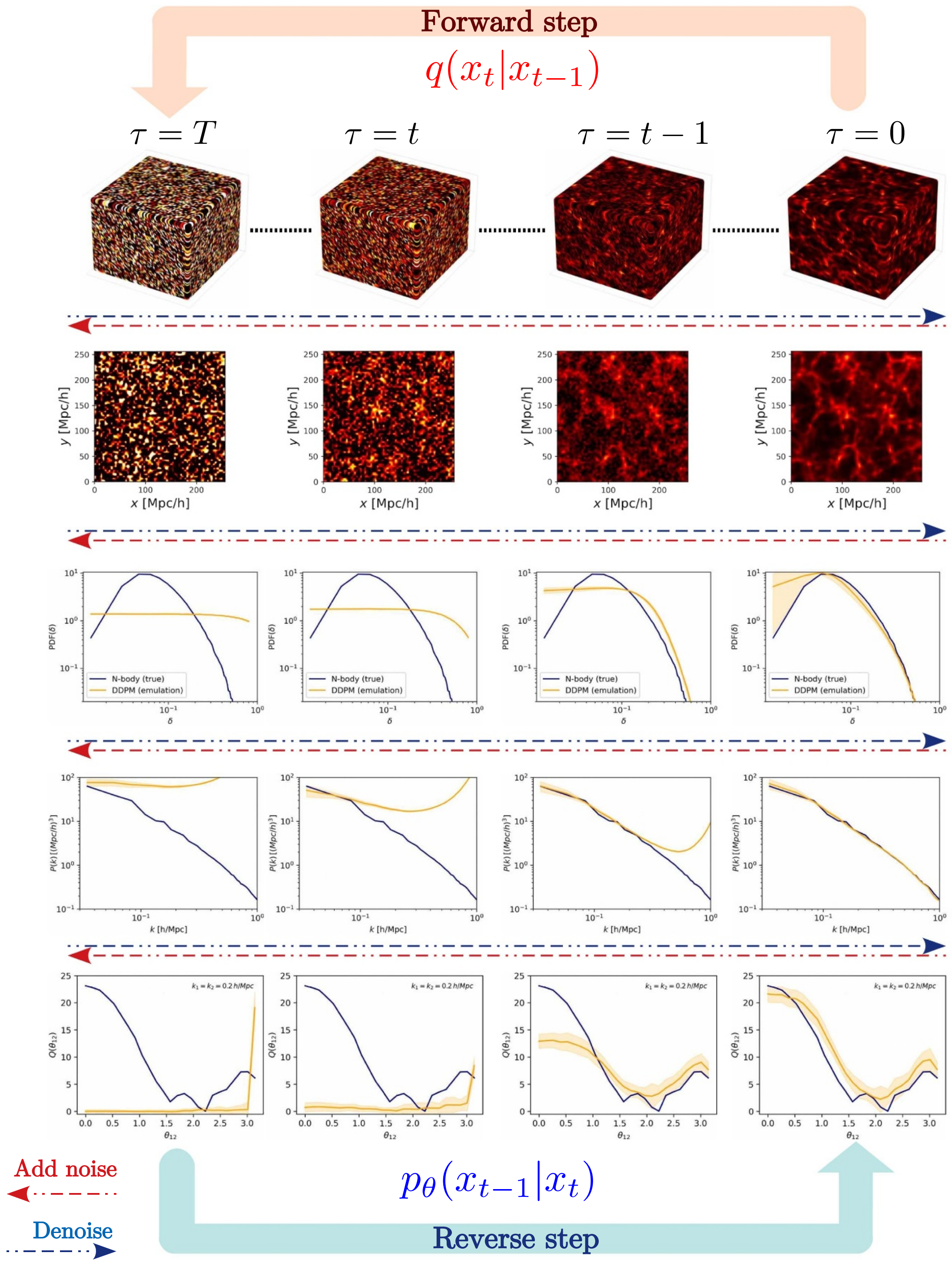


Figure 1. The diagram illustrates the forward and reverse processes in DDPM. The top panel shows the generative 3D density fields created at each time step τ, with their corresponding 2D projections displayed below. The centre panel presents the power spectra (PS), followed by the evolution of one bispectrum configuration. The bottom panel depicts the probability distribution function (PDF) of the voxels (see section 4.3 for definitions of these statistical moments). In the plots, the blue curves represent the true statistical moments, while the orange curves correspond to those computed from the generated simulation at each time step τ.

learns to reverse this corruption, iteratively building coherent structures from the noise as it is showed in figure 1. This bidirectional approach, involving both forward and reverse diffusion, has proven particularly effective for challenging astrophysical scenarios, most notably in generating high-resolution fields [54], reconstruction [55], and as emulators [46, 48, 56], setting new benchmarks for generative modelling performance. The learning aspect of these models involves mastering the reversal of a complex noising process, where progressively more noise is actively added to an initial image $x_0 \sim q(x_0)$ [54]. This noise sequence is executed through a Markov chain of T steps and systematically introduces Gaussian noise at each stage, generating a sequence of noise samples $x_1, \ldots, x_t$. According to [51], during the forward diffusion step, noise is introduced to a sample x_t from the preceding one x_{t-1} and step sizes are regulated by a variance $\{\beta_t \in (0,1)\}_{t=1}^{T}$ as

$$q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right) \tag{1}$$

where $\beta_1 < \beta_2 < \cdots < \beta_t$ regulates the reduction in noise between steps. In equation (1) the process assumes that x_t is conditionally Gaussian with a mean $\sqrt{1-\beta_t}x_{t-1}$ and variance $\beta_t\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The mean term controls how much of x_{t-1} contributes to x_t, while the variance introduces isotropic Gaussian noise with a magnitude determined by β_t [57]. Therefore, by repeatedly applying the forward diffusion process, the image at a specific time t, denoted x_t, can be expressed as a function of the original field x_0. From conditional probabilities, the following joint probability is calculated as

$$q\left(x_{1:T} \mid x_0\right) = \prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right). \tag{2}$$

The term $q(x_{1:T} \mid x_0)$ represents the overall probability of observing the sequence x_1 to x_T. Each factor $q(x_t \mid x_{t-1})$ denotes the probability of transitioning to state x_t from the previous state x_{t-1}, capturing the Markov property, where the next state is determined solely by the current state [57]. A notable feature of this process is that sampling at any arbitrary time step can be achieved in closed form by leveraging the reparametrisation trick [51]. This property allows for direct access to any sample x_t eliminating the need to sequentially compute all $t-1$ previous noisy image

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{3}$$

where $\alpha_t = 1-\beta_t$, with $\bar{\alpha}_t = \Pi_i^t \alpha_i$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The noise added at each step is systematically removed during the reverse diffusion phase [51]. As a result, the process begins with a distribution that contains only noise (the final state of the forward process). Consequently, the noise is removed from the samples step by step, moving in the reverse direction. As stated in [51], the inverse diffusion process considers x_0, and the events are connected through the conditional probability distribution

$$p_\theta\left(\mathbf{x}_{0:T}\right) = p\left(\mathbf{x}_T\right)\prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) \tag{4}$$

being the reverse process equals to

$$p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \Sigma_\theta\left(\mathbf{x}_t, t\right)\right). \tag{5}$$

Here, a neural network with parameters θ is used to compute Eq (4). This express the joint probability distribution $p_\theta(\mathbf{x}_{0:T})$ over a sequence of variables $x_0, x_1, \ldots, x_T$ parametrised by θ [57]. We can divide this joint probability into two parts: the marginal probability of the final state $p(x_T)$, and the product of conditional probabilities $p_\theta(x_{t-1} \mid x_t)$ over all timesteps t from 1 to T [57]. This structure reflects a reverse process, where each state x_{t-1} depends on the subsequent step x_t. The variance is usually selected as $\Sigma_\theta(\mathbf{x}_t, t) = \beta_t\mathbf{I}$ as [51] reported to be the best performance in their results, while the mean $\mu_\theta(\mathbf{x}_t, t)$ is given by

$$\mu_\theta\left(\mathbf{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\mathbf{x}_t, t\right)\right), \tag{6}$$

where $\epsilon_\theta(\mathbf{x}_t, t)$ is the neural network outcome of the noise ϵ present in the sample x_t. The loss function used for this optimisation is given by the expectation value [51]

$$\mathcal{L}_t = \mathcal{E}_{t\sim[1,T], x_0, \epsilon}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right], \tag{7}$$

being $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)$ the neural network prediction of the noise ϵ present in the sample x_t, and $t \sim [1, T]$ the time step drawn from a uniform distribution. By minimising this loss, the model learns to predict and remove the noise at each step, enabling it to reverse the diffusion process during inference and generate realistic data

Algorithm 1. DDPM training.

```
1:  Randomly select a simulation $x_0$ and its cosmological parameters $y$ from the training dataset distribution $q(x_0)$.
2:  Draw a Gaussian-noise sample $\gamma \sim U(0,1)$
3:  if $\gamma < 0.1$ then
4:      Discard conditioning from the dataset $p(x_0, y = \emptyset)$
5:  else
6:      Keep conditioning from the dataset $p(x_0, y)$
7:  end if
8:  Randomly select a time step $t$ in the Markov chain from the uniform distribution $\{1,\ldots,T\}$.
9:  Draw a Gaussian-noise sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.
10: Compute the sample $x_t$ in the $t$th step of the Markov chain as equation (3).
11: Make a gradient descent step with $\nabla_\theta \mathcal{L}_t$ defined in equation (7).
12: Repeat steps 1–5 until converged.
```

Algorithm 2. DDPM sampling.

```
Draw a Gaussian-noise sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.
Choose $\omega = [0,1]$: guidance strength.
Loop through the backward Markov chain:
for $t = T,\ldots,1$ do
    if $t > 1$ then
        $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
    else
        $\epsilon = 0$                                  ▷ No additional noise in the last step
    end if
    Compute $x_{t-1}$ equation (8) where the score estimation is given by equation (12)
end for
Return $x_0$.
```

from random noise. The training algorithm is listed in algorithm 1. Once training is completed, we expect to generate $x_0 \sim p(x_0)$ image from noise. In fact, the model learns to approximate the probability distribution of the training set. Hence, we can sample from this distribution and be able to generate new samples that obey the same features as the training dataset. This can be done by sampling T times equation (5) crossing the Markov chain until $t = 0$ as

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\mathbf{x}_t, t\right)\right) + \sqrt{\beta_t} z, \tag{8}$$

with $z \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. Here, the first term is the mean estimate provided by the neural network equation (6) perturbed by the presence of a Gaussian noise β_t akin to a Langevin sampling step [58]. The inference algorithm is listed in algorithm 2.

2.3. Denoising diffusion implicit models (DDIMs)

DDIMs are implicit probabilistic models associated with DDPMs, since they are trained using the same loss function [59]. DDIMs present an optimised version of DDPM and offer a more efficient and faster solution to the image generation problem. Although it uses the same training objective as DDPM, DDIM introduces non-Markov processes instead of strictly following the Markov approach. This allows DDIM to balance between the quality of the generated samples and processing time. Furthermore, it can create high-quality images faster than DDPM and it performs direct interpolations in latent space and reconstructs observations with minimal error, providing greater flexibility in the generation process [60]. According to [60], the non-Markov inference process is employed in this case, which leads to the same function applied in the DDPM model mentioned above in equation (1). Therefore, the DDIM model generalises the DDPM model and, in turn, allows modifications to the design of the inverse Markov chains. The expression for the non-Markovian conditional probability distribution $p_{(}x_{t-1} \mid x_t, x_0)$ is

$$p\left(x_{t-1} \mid x_t, x_0\right) = \mathcal{N}\left(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\, \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma^2 \mathbf{I}\right). \tag{9}$$

Algorithm 3. DDIM sampling.

```
Create a time subset $\{t_1,..,t_s\} \in \{t_1,..,t_T\}$ with $s \ll T$
Choose $\omega = [0,1]$: guidance strength.
Draw a Gaussian-noise sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.
Loop through a subset of timesteps:
for $t = s,\ldots,1$ do
    if $s > 1$ then
        $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
    else
        $\epsilon = 0$                                  ▷ No additional noise in the last step
    end if
    Compute $x_{t-1}$ equation (11) with $\sigma_t = 0$ and where the score estimation is given by equation (12)
end for
Return $x_0$.
```

According to [60], the processes for the implicit DDIM diffusion models are defined in two phases. In the first phase, the forward diffusion process defines x_0 and transforms it into x_t. Initially, the inference distribution, the non-Markovian forward process is as follows

$$p\left(x_{1:T} \mid x_0\right) = p\left(x_T \mid x_0\right) \prod_{t=2}^{T} p\left(x_{t-1} \mid x_t, x_0\right), \tag{10}$$

where $p(x_{1:T} \mid x_0)$ corresponds to the conditional probability of observing the sequence of variables $x_{1:T}$ evolves from an initial state x_0. It shows that the likelihood of a sequence of observations given the initial conditions can be decomposed in terms of a chain of probabilistic dependencies over time [61]. Equation (9) can be expressed as

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta\left(\mathbf{x}_t, t\right)}{\sqrt{\bar{\alpha}_t}} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\epsilon_\theta\left(\mathbf{x}_t, t\right) + \sigma_t z, \tag{11}$$

with $z \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, $\epsilon_\theta(\mathbf{x}_t, t)$ is the predicted neural network noise ϵ_t at time t, and σ_t is a parameter learning whose variation determines the difference in the posterior distribution. When $\sigma_t = 0$, there is not random sampling and the sample is generated into a deterministic scenario. This is the core of DDIM model. Besides, since it does not need to satisfy the Markov process, a subset $\{t_1,..,t_s\} \in \{t_1,..,t_T\}$ with $s \ll T$ can be created from the original T diffusion time-steps for sampling inference, where s is the number of steps in the new diffusion subset. The inference DDIM algorithm is listed in algorithm 3. DDPM and DDIM primarily differ in their approach to sampling. While DDPM relies on a Markov process and requires many diffusion steps to achieve high-quality results, it tends to be computationally expensive [57]. The DDIM model offers several improvements over DDPM. It can produce higher-quality samples in fewer steps, enhancing efficiency [57]. Moreover, DDIM maintains a consistency property due to its deterministic generative process, ensuring that samples conditioned on the same latent variable share similar high-level features [57]. This consistency also enables DDIM to perform meaningful semantic interpolation within the latent space, resulting in smoother and more interpretable transitions between samples [57].

2.4. Conditioned generation: classifier-free diffusion guidance

While training generative models on the simulation, it is important to generate samples conditioned on the cosmological parameters. To explicit incorporate parameter information into the diffusion process, we employ the Classifier-Free Guidance in our methodology [62]. This technique assumes an unconditional denoising diffusion model $p(x)$ parametrised through an estimator $\epsilon_\theta(x_t, t) = \epsilon_\theta(x_t, t, y = \emptyset)$[6] and a conditional model $p_\theta(x|y)$ parametrised through $\epsilon_\theta(x_t, t, y)$. Both models are trained with the same neural network. In fact, the conditional diffusion model is trained on data (x, y), where the conditioning cosmological parameters y are randomly discarded by $\gamma < 0.1$ (being γ a sample drawn from an uniform distribution [0,1]) such that the model knows how to generate images unconditionally as well. Therefore, the score estimator can be written as [62]

$$\bar{\epsilon}_\theta\left(x_t, t, y\right) = \epsilon_\theta\left(x_t, t, y\right) + \omega\left(\epsilon_\theta\left(x_t, t\right) - \epsilon_\theta\left(x_t, t, y\right)\right), \tag{12}$$

[6] Where for the unconditional model we input a null value $\emptyset$ for the class identifier y when predicting the score.

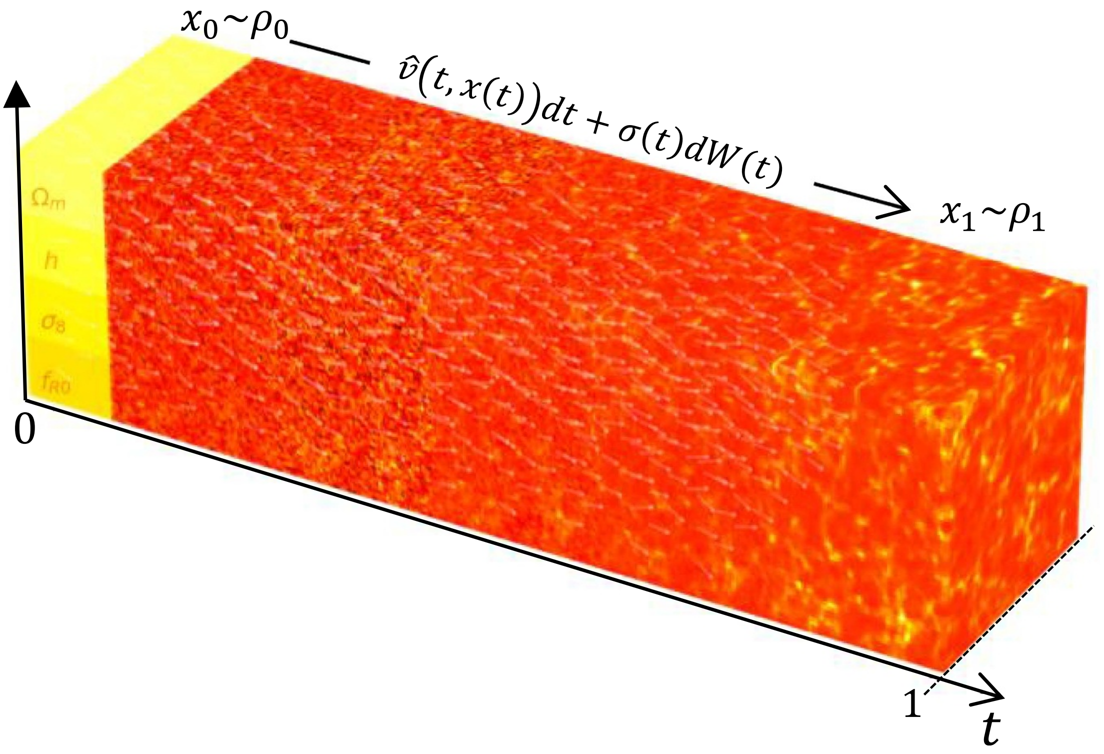


Figure 2. Illustration of the generative model based on SIs, which connects two densities ρ_0 and ρ_1 that represent the feature representation of the cosmological parameters and the 3D density field respectively. The time-dependent probability density $\rho(t)$ that bridges ρ_0 and ρ_1 is found through the the forward stochastic differential equation solutions equation (16) where a drift function is computed by the UNet. Here, the vector field represented with white arrows describes the drift function.

where ω is a parameter that controls the strength of the classifier guidance. In our experiments, we found that $\omega = [0, 0.5]$ provides a suitable range of values to obtain good results. The authors [62] conclude in their studies that the diffusion model needs to have a part dedicated to the unconditional generation task in order to produce classifier-free guided scores effective for sample quality. This methodology only has been used to improve the performance of the model, but also to reduce the memorisation behaviour found in diffusion models [63].

2.5. Stochastic interpolants (SIs)

Although diffusion-based methods have achieved impressive results in areas such as image generation, there is ongoing research into methods that provide exact transport between arbitrary (not just Gaussian) probability densities within a finite time frame. Initially, score-based diffusion models achieved the best results using Stochastic Differential Equations (SDEs) [64]. However, recent research has shown that methods based on ordinary differential equations (ODEs) can achieve comparable or even superior performance if the scoring function is learned effectively. ODE-based methods offer significant advantages, including the availability of an exact and computationally tractable likelihood formula and the straightforward application of established adaptive integration techniques for sampling [65]. A recent generative model based on stochastic dynamics propose the use of SIs x_t that connect a base density ρ_0 to the target ρ_1, but allow for bases that are more general than a Gaussian density. The dynamics can be described as [65]

$$x(t) = \alpha(t)x_0 + \beta(t)x_1 + \sigma(t)W(t), \quad t \in [0, 1], \tag{13}$$

where by construction, it satisfies $x(t = 0) = x_0 \sim \rho_0$, and $x(t = 1) = x_1 \sim \rho_1$. This approach therefore exactly bridge between samples from ρ_0 at $t = 0$, and from ρ_1 at $t = 1$ as it is showed in figure 2. For a large class of densities, ρ_0 and ρ_1 supported on $\mathbb{R}^d$, these distributions are absolutely continuous with respect to the Lebesgue measure and $\rho(t)$ satisfies a family of forward and backward Fokker–Planck equations [64]. Therefore, equation (13) can be realised by many different processes such as ODEs and SDEs, and whose densities at time t are given by $\rho(t)$ [65]. Following the work in [55, 66] let us write the functions under equation (13) as $\alpha(t) = \sigma(t) = 1 - t$, $\beta(t) = t^2$, and $W = \sqrt{t}z$ with $z \sim \mathcal{N}(0, I)$ a Wiener process. The authors in [64, 66] also demonstrate that the velocity field associated with the interpolant, equation (13) takes the form

$$v(t, x_0, x_1) = \dot{\alpha}(t)x_0 + \dot{\beta}(t)x_1 + \dot{\sigma}(t)W(t), \tag{14}$$

Algorithm 4. SDEs training.

```
1: Input: Randomly select a simulation $x_0$ and its labels from the training dataset distribution $q(x_0)$.
2: Randomly select a time step $t$ from the uniform distribution $\{1,\ldots,T\}$.
3: Draw a Gaussian-noise sample $z\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ and build the Wiener process.
4: Compute $x(t)$ and the velocity field defined in equations (13)–(14) respectively.
5: Make a gradient descent step with $\nabla_\theta\mathcal{L}[\hat{v}]$ defined in equation (15) and compute it via Monte
   Carlo sampling.
6: Repeat steps 1–5 until converged.
```

where the dot in the variables represents differentiation with respect to time t. The velocity field can be approximately computed with a neural network $\hat{v}(t,x(t))$ by minimising the loss function

$$\mathcal{L}\left[\hat{v}\right]=\int_0^1 \mathrm{d}t\,\mathcal{E}_{x_0,x_1\sim\rho_0,\rho_1}\left[\left(\hat{v}\left(t,x\left(t\right)\right)-v\left(t,x_0,x_1\right)\right)^2\right]. \tag{15}$$

Once trained, the velocity field will function as the drift term within the SDE [55]

$$\mathrm{d}x\left(t\right)=\hat{v}\left(t,x\left(t\right)\right)\mathrm{d}t+\sigma\left(t\right)\mathrm{d}W\left(t\right), \tag{16}$$

whose solutions are such that $x(t=1)\sim\rho(x_1|x_0)$ and W accounts for another Wiener process. This equation expresses the evolution of $x(t)$ in terms of two components, the first term describes the deterministic part of the dynamics, while the second term accounts for the stochastic component of the process. To suit this approach to our work, x_0 represents a latent representation of the cosmological parameters generated by a neural network while x_1 describes the 3D simulation (see figure 2). Once the model is trained, the velocity field is substituted by the UNet in equation (16), and the initial volume $(x(t=0))$ is given by the feature representation for the cosmological parameters. The time interval $t\in[0,1]$ is divided in 200 steps along which we solve equation (16). The SDE was solved using the Euler second-order method. At the end, the emulator should generate distributions of volumes $x(t=1)$ that resemble the characteristics of the density field conditioned on the cosmological parameters.

3. Dataset: modified gravity simulations

We used a dataset already generated by [67]. These simulations were created with the COmoving Lagrangian Acceleration (COLA) algorithm [68, 69], in particular, the authors used MG-PICOLA[7] [70], a modified version of L-PICOLA [71] that has been extensively tested against full N-body simulations and that extends the gravity solvers to a variety of gravity models.

The dataset consists of 2500 modified gravity simulations varying four cosmological parameters $\Theta=\{\Omega_m, h, \sigma_8, f_{R0}\}$, where h is the reduced Hubble parameter, σ_8 the r.m.s. density fluctuation within a top-hat sphere of 8 Mpc/h radius and f_{R0} the amplitude of the modified gravity function in the Hu and Sawicki model (HS) [72]. The remaining cosmological parameters are set to $\Omega_b=0.048\,206$ and $n_s=0.96$, which correspond to the values reported by [73]. The parameter space is sampled with random numbers uniformly distributed within the specified ranges for each parameter (see table 1). Figure 3 shows the distribution of the 2500 $f(R)$ cosmologies used in this work, presented in a plane projection and highlighting the datasets used for training (light blue dots), testing (orange dots) and validation (green dots). Each simulation follows the dynamics of the particles 128^3 in a small box of comoving side-length 256 h^{-1} Mpc, using 100 timesteps from an initial redshift $z_i=49$ to a redshift $z=0$. The training set comprises 85% of the data (and validation), while the remaining 15% of the data were used for testing. For each simulation, we estimate the density field using a cloud-in-cell particle mesh assignment on a grid with 64^3 voxels. We consider the effective range of the power spectrum up to the Nyquist frequency, k_{Ny}, which in our

[7] The code can be found at https://github.com/HAWinther/MG-PICOLA-PUBLIC.

Algorithm 5. SDEs sampling.

```
Randomly select a simulation $x_0$ and its labels from the training distribution, the trained model $\hat{v}(t_n, x_n)$; and define a grid $t_0 = 0 < t_1 \cdots < t_T = 300$.
Set $\Delta t_n = t_{n+1} - t_n$, $n = 0 : T - 1$.
Create a 3D image representation of the labels
Draw a Gaussian-noise sample $z_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and build the Wiener process.
Set $x_1 = x_0 + \hat{v}(t_0, x_0)\Delta t_0 + \sigma(t_0)\sqrt{\Delta t_0} z_0$.
for $n = 1 : T - 1$ do
    Compute $\hat{v}_{t_n}(x_n, t_n)$ from equation (16).
    Set $x_{n+1} = x_n + \hat{v}(x_n, t_n)\Delta t_n + \sigma(t_n)\sqrt{\Delta t_n} z_n$.
end for
Return: $x_T$.
```

Table 1. The summary of the set-up of the MG simulations. Left: cosmology parameters. Right: set-up parameters used for MG-PICOLA code.

Cosmologies	
Ω_m	[0.1, 0.5]
h	[0.5, 0.9]
σ_8	[0.6, 1.0]
$0.1\log_{10}\lvert f_{R0}\rvert$	[0.4, 0.6]
Ω_b	0.0489
n_s	0.9665

Cosmologies	
Boxsize	256 h^{-1} Mpc
N_p	128^3
Grid force	128^3
IC	2LPT $z_{ini} = 49$
Steps	100
k_{Ny}	1.58 hMpc^{-1}

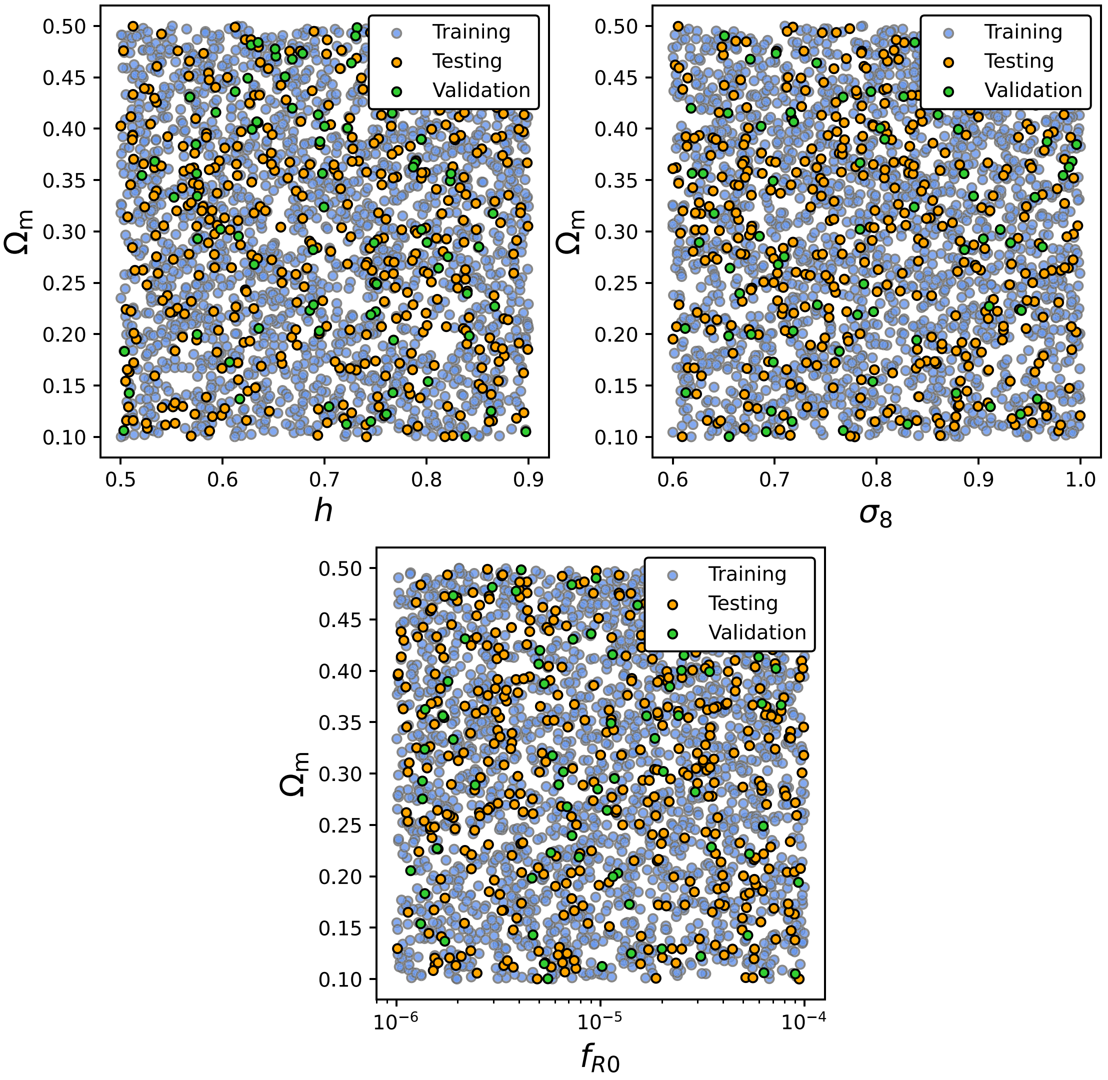


Figure 3. Distribution of the 2500 $f(R)$ cosmologies employed in the diffusion models in selected parameter planes. Light-blue dots represent the training dataset, orange dots the testing dataset, and green dots correspond to the validation dataset. The parameter planes shown are Ω_m versus h, σ_8, and f_{R0}.

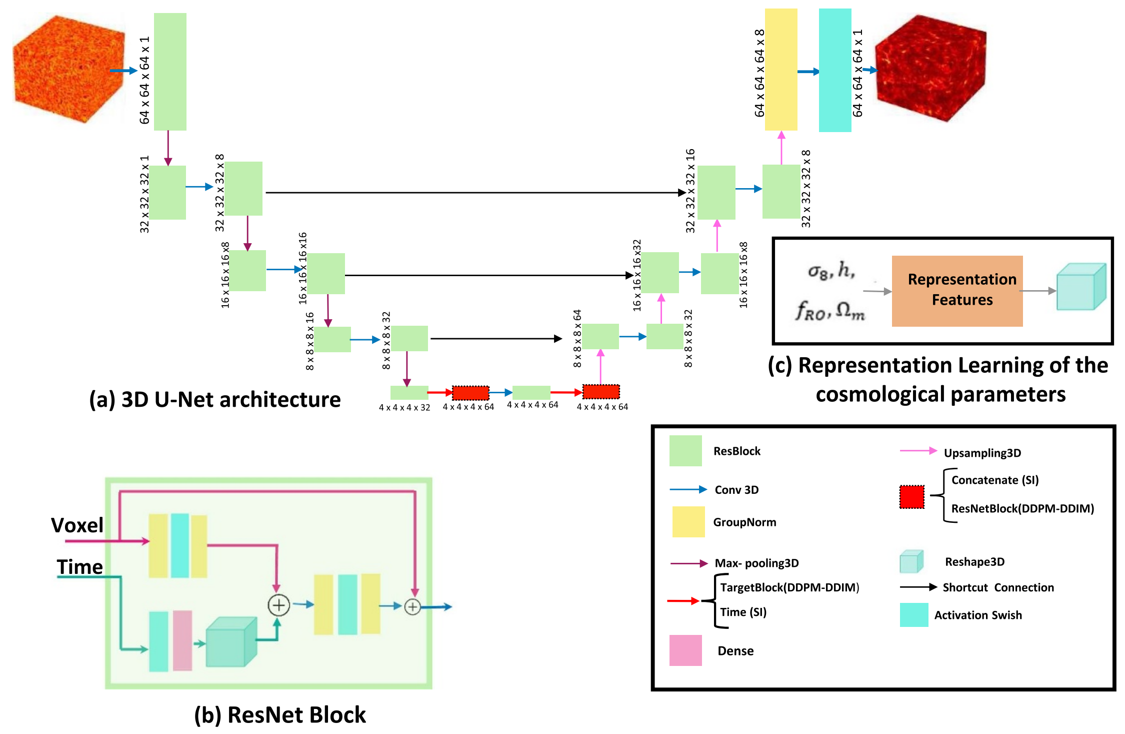


Figure 4. 3D UNet modules employed for all models in the paper. (a) UNet architecture composed of ResNet blocks and connections between blocks at each level of the encoder–decoder. (b) ResNet module schema. (c) Target block schema that transform the parameter space into a 3D representation to be inserted into the UNet.

simulations corresponds to $k \approx 0.75$ Mpc h^{-1}. This resolution results from downsampling the original 128^3 voxel grid of the simulation data, which has a Nyquist frequency of $k_{\mathrm{Ny}} \approx 1.58\,\mathrm{h\,Mpc}^{-1}$.

4. Methodology

4.1. Neural network architecture

While DDPM and DDIM employ neural networks to predict noise at each time step during reverse diffusion, SIs use them to estimate the velocity field $\hat{v}(t, x_t)$. The architecture used in this research for all approaches is the 3D-UNet depicted in figure 4. This model starts with $64 \times 64 \times 64$ voxels with 1 channel, which are passed to a calculating schedule across $T = 1000$ timesteps, geometrically (cosine) interpolating noise levels from a Beta Start of 1×10^{-4} to a Beta End of 0.02. Several experiments were performed using linear, polynomial, and sigmoid functions, however, cosine functions provided the best performance. This UNet consists of an encoder, the middle module, and its decoder. The feature maps of the same pixel level are concatenated via shortcut connections between the encoder and decoder. In the encoder, max-pooling is used to down-sampling layer halves the feature maps, enhancing feature extraction and expanding the receptive field. On the other hand, up-sampling3D in the decoder increases the feature maps, progressively restoring the spatial resolution of the original volume. The output of the decoder part is processed with a group normalisation layer followed by an swish activation and a final convolutional block. In the middle module, we have different configurations depending on the method used (DDPM or SI). For the SI case, four ResNet blocks are sequently used. The ResNet block shown in figure 4(b) processes both the feature maps and the timesteps. The latter is first projected onto an embedding space of dimension 32 using sinusoidal scaling, and then processed through two dense layers of 32 neurons, each with swish activation functions. In case that DDPM (and DDIM) is employed, the middle module consists in two paired ResNet-Target blocks where the cosmological parameter information is inserted into the architecture. Figure 4(c) illustrates the target block schema, where the cosmological parameters are fed into a pre-trained neural network to get a 3D feature representation (explained later in section 4.2) of these parameters. The resultant parameter voxel is then concatenated to the feature maps coming from the encoder part.

4.2. Feature representation for cosmological parameters

We built a 3D volume feature representation for the cosmological parameters to either aggregate it with the simulation boxes along with the time-steps in DDPM (and DDIM) or define the base density $x0 \sim \rho_0$ in the

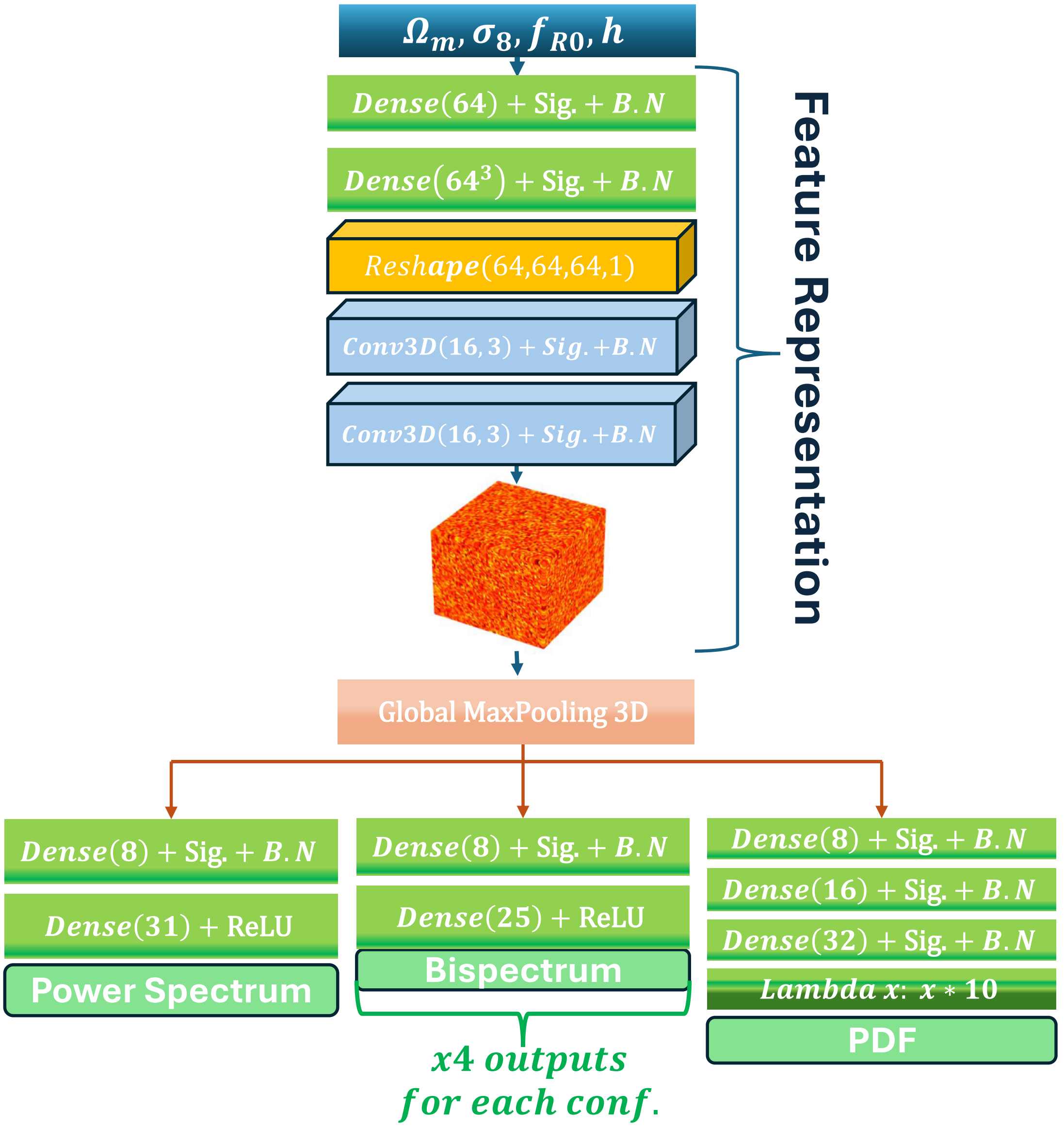

$\Omega_m, \sigma_8, f_{R0}, h$
$Dense(64) + \text{Sig.} + B.N$
$Dense(64^3) + \text{Sig.} + B.N$
$Reshape(64,64,64,1)$
$Conv3D(16,3) + Sig. + B.N$
$Conv3D(16,3) + Sig. + B.N$
Feature Representation
Global MaxPooling 3D
$Dense(8) + \text{Sig.} + B.N$
$Dense(31) + \text{ReLU}$
Power Spectrum
$Dense(8) + \text{Sig.} + B.N$
$Dense(25) + \text{ReLU}$
Bispectrum
x4 outputs for each conf.
$Dense(8) + \text{Sig.} + B.N$
$Dense(16) + \text{Sig.} + B.N$
$Dense(32) + \text{Sig.} + B.N$
$Lambda\ x{:}\ x * 10$
PDF

Figure 5. Multioutput regression task for predicting the power spectra (PS), probability distribution function of the voxels (PDFs), and four different configurations for the bispectra ($[\text{Bis}_1, \ldots, \text{Bis}_4]$) from a set of cosmological parameters. Following the training phase, a submodel is extracted from the layers preceding the Global MaxPool 3D in the neural network. This submodel enables the generation of a three-dimensional feature representation of the cosmological parameters, capturing their complex relationships and working as a conditioner for the generative models.

SI approach. We propose to build this parameter volume based on the so-called representation learning, a powerful technique that enable a neural network to automatically discover and learn the most useful representations of raw data [74]. First, we developed a multioutput regression model using the neural network displayed in figure 5. For this task, we compute the summaries for all train, validation and test volumes such as the power spectra (PS), probability distribution function of the voxels (PDFs), and four different configurations for the bispectra (Bis). Both the multi-output regression and diffusion models were trained on the same training set. For hyperparameter tuning, we used the same validation set for both models, while the test set was reserved exclusively for the final evaluation. Then, we take data pairs ($[\Omega_m$, h, $\sigma_8, f_{R0}]$, [PS, PDFs, $\text{Bis}_1, \ldots, \text{Bis}_4]$) for training the model in a supervised way. The network receives the cosmological parameters as input, which are then processed by two dense layers, each with 64 neurons, followed by a sigmoid activation function and batch normalisation. The output features are reshaped into a $(64, 64, 64, 1)$ volume and passed through three 3D convolutional layers, each employing 16 filters, sigmoid activations, and batch normalisation. Then, a 3D convolutional layer with one filter along with a sigmoid activation is applied generating a 3D representation of the input parameters with dimensions matching the simulation boxes. This sub-neural network yields the 3D representation used in the diffusion models. Following with the neural network architecture, a 3D global max pooling operation is applied to flatten the volume, resulting in six output branches, each corresponding to one of the pre-defined summaries. The optimised network architecture is shown in figure 5. Training is performed using a Huber loss, with weighted losses assigned to the PS and PDF to prioritise their accuracy.

4.3. Training and summary statistics

Simulation data normalisation involved clipping values using the minimum of all maximum values found across the boxes in the training dataset. Subsequently, we applied a logarithmic scale and normalised the data by subtracting the minimum logarithmic value (logmin) and dividing by the range of logarithmic values (logmax–logmin), both calculated from the training data. All models were trained using the Huber loss in equation (7) instead of the standard mean squared error. The models were optimised with the Adam optimiser employing a learning rate of 10^{-4}, a batch size of 16, and training for 30 epochs. Callbacks were implemented to mitigate overfitting. The DDPM model, with approximately 15 million parameters, required approximately nine hours of training on a 16GB Nvidia T4 GPU, while the interpolant model required twelve hours on the same GPU.

4.3.1. A quality metric for generated density fields: n-point statistics

The spatial distribution of dark matter is non-Gaussian, and remarkably little is known about the information encoded in it about cosmological parameters beyond the power spectrum. Therefore, it is crucial that generative models can learn significant information well beyond its power spectrum. Therefore, to illustrate the quality generation of our emulators, we compute some summary statistics that provide information about the Gaussian and non-Gaussian signals. We start using the one-point statistics, commonly known as the probability density function (PDF). The PDF reveals density variations within the simulated volume, identifying overdense regions like galaxy clusters and dark matter halos, as well as underdense regions such as cosmic voids. The values of density contrast δ are binned using logarithmically spaced bins. The PDF of the cosmic density field is then defined as the normalised number of cells as:

$$P(\delta_i) = \frac{N_i}{N_{\text{total}} \Delta\delta}, \tag{17}$$

where N_i is the number of samples in the ith bin, N_{total} is the total number of samples, and $\Delta\delta$ is the width of each bin. The next statistical moment is the matter power spectrum defined as

$$\left\langle \delta(\mathbf{k})\, \delta\left(\mathbf{k}'\right) \right\rangle = (2\pi)^3\, \delta_D\left(\mathbf{k} + \mathbf{k}'\right) P(k), \tag{18}$$

where angular brackets denote ensemble average, δ_D is the 3D Dirac delta function, which enforces the homogeneity of the density statistics, and $\mathbf{k}$, $\mathbf{k'}$ are Fourier modes. The fact that the power spectrum depends only on the magnitude $k \equiv |k|$ is required by isotropy, which allow us to provide information about the Gaussian signal in the data. In addition to the two-point statistics, we also considered the three-point statistics of the density field. These statistics are able to capture any non-Gaussianities in the density field. The matter bispectrum $B(k_1, k_2, k_3)$ is defined as

$$\langle \delta(\mathbf{k}_1)\, \delta(\mathbf{k}_2)\, \delta(\mathbf{k}_3) \rangle = (2\pi)^3\, \delta_D(\mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3)\, B(k_1, k_2, k_3). \tag{19}$$

Unlike the power spectrum, which is only sensitive to the magnitude of Fourier modes, the bispectrum is the lowest-order correlator that is sensitive to phases. Because homogeneity constraints the wavenumbers $(k_1 + k_2 + k_3)$ to form a closed triangle, we can also express the bispectrum as a function of two magnitudes and an angle, i.e. $B(k_1, k_2, \theta)$. It is useful, particularly in analyses of modified theories of gravity to consider the reduced bispectrum

$$Q(k_1, k_2, k_3) = \frac{B(k_1, k_2, k_3)}{P(k_1) P(k_2) + P(k_2) P(k_3) + P(k_1) P(k_3)}, \tag{20}$$

to remove the information that is already contained in the power spectrum. Note that $Q(k_1, k_2, k_3)$ can be written as $Q(k_1, k_2, \theta)$ which define a unique triangle given two out of the three arguments. We use the Pylians3 library to compute these statistics[8].

[8] https://pylians3.readthedocs.io/.

5. Results

Having thoroughly examined the methodologies employed for DDPM (and DDIM) as well as SI models, we now turn our attention to evaluating their performance in generating 3D density fields. To assess the efficacy of the DDPM model, we trained it on the relevant dataset and subsequently generated 50 synthetic samples. These samples were then rigorously compared against a test set instance with identical cosmological parameters, focusing on their summary statistics. The outcomes of this analysis for the DDPM model are illustrated in figure 6. The results demonstrate that the DDPM-generated fields exhibit remarkable consistency with the true field. Not only do they accurately capture the Gaussian signal, but they also successfully recover a diverse range of bispectra configurations. Specifically, two of the bispectra configurations analysed were directly aligned with those used during the training of the feature representation (as detailed in section 4.2), while the remaining configurations represent interpolations between these key points. This highlights the models ability to generalise and produce physically meaningful outputs, even for configurations not explicitly encountered during training. Notice that for lower wavenumbers(k), the predicted power spectrum deviates significantly from the true one. This discrepancy can be attributed to the finite size of the volumes, which inherently imposes a cut-off on large-scale modes. Due to the periodic boundary conditions and the limited spatial extent of the simulation boxes, modes with wavelengths exceeding the box size are effectively truncated. As a result, the power spectrum and bispectra, are influenced by the absence of these large-scale fluctuations. For the latter, we can observe slight deviations in their tails. These limitations are particularly significant in cosmological simulations with a small resolution size, as large-scale modes play a crucial role in shaping the structure of the density fields. As previously discussed, one of the primary drawbacks of DDPM is the significant computational time required to generate samples. This is due to the iterative nature of the process, where the neural network must denoise the image over $T = 1000$ sequential steps. To address this limitation, DDIM was introduced as an alternative approach during the inference process after training the DDPM model. DDIM accelerates the generation process by relaxing the Markovian assumption, as detailed in section 2.3. While this modification substantially reduces inference time, it comes at the cost of a slight degradation in the quality of the generated samples. This trade-off is evident in figure 7, where the bispectra of the DDIM-generated simulations show a noticeable, though modest, deterioration compared to those produced by DDPM. The balance between sample quality and inference time is a critical consideration, particularly in applications requiring the bispectra to constrain cosmological parameters. During the validation phase, this trade-off must be carefully calibrated to ensure that the reduction in computational cost does not compromise the scientific utility of the generated samples. By fine-tuning this balance, DDIM enables the efficient production of a high number of volume samples in a shorter time, making it a practical choice for large-scale simulations despite its minor quality trade-offs. Note that in the DDIM model, the small Fourier modes exhibit behaviour consistent with the ground truth. However, deviations begin to emerge for larger modes, reaching up to 30% error. This discrepancy can be linked to the slight degradation in quality observed in DDIM, as we typically expect precise reconstruction at scales below the Nyquist frequency. Despite this, the power spectrum can be recovered within tens of percent accuracy across the entire range. Finally, the statistics of a generated sample from the SI approach are illustrated in figure 8. It is evident that the SI method yields lower performance compared to the previous model. While SI successfully reconstructs the probability distribution function for the majority of the samples, it struggles to accurately capture higher statistical moments, despite displaying favourable trends and shapes relative to the ground truth.In table 2, we present the evaluation metrics for all models using the entire test dataset. We employ the mean square error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2) to assess the accuracy of the statistical moments derived from the power spectrum, a bispectrum configuration, and the probability density function (PDF). Our results indicate that the DDPM significantly outperforms the other models, achieving the lowest error across all metrics. However, DDPM requires more time to generate volumes compared to the other methods. In contrast, the DDIM generates images in just 9.9 s, making it a practical choice for applications where faster generation is essential such as a parameter constraints. Additionally, we observe that the Stochastic Interpolation (SI) model excels at recovering the PDF and generates data in less than a minute. We think that further refinement through hyperparameter tuning could enhance SI performance, potentially making it a highly accurate and efficient model in terms of both precision and inference time.

6. Discussion

This work has demonstrated the potential of conditional generative modelling to accurately create 3D dark matter density fields, capturing high-order statistical moments with considerable success. Our approach offers a promising avenue for generating realistic cosmological structures, a crucial task for various analyses

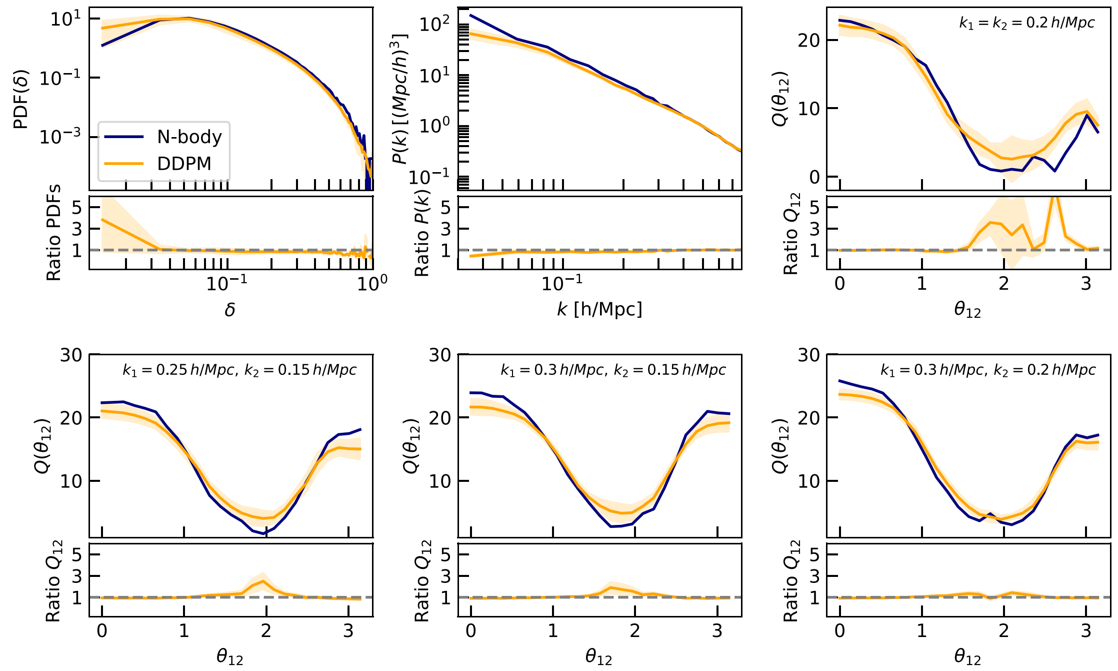


Figure 6. Summary statistics for generated fields with DDPM corresponding to the fiducial value ($\Omega_m = 0.305$, $\sigma_8 = 0.710$, $h = 0.561$, and $0.1\log_{10}|f_{R0}| = 0.43$). Each panel shows the probabilistic distribution function PDF with 50 bins, power spectrum $P(k)$, and four bispectra configuration $Q(\theta_{12})$. Solid orange line represents the mean over 50 generative samples, while the orange region defines the standard deviation. Bottom plots illustrate the percent error for each summary statistics.

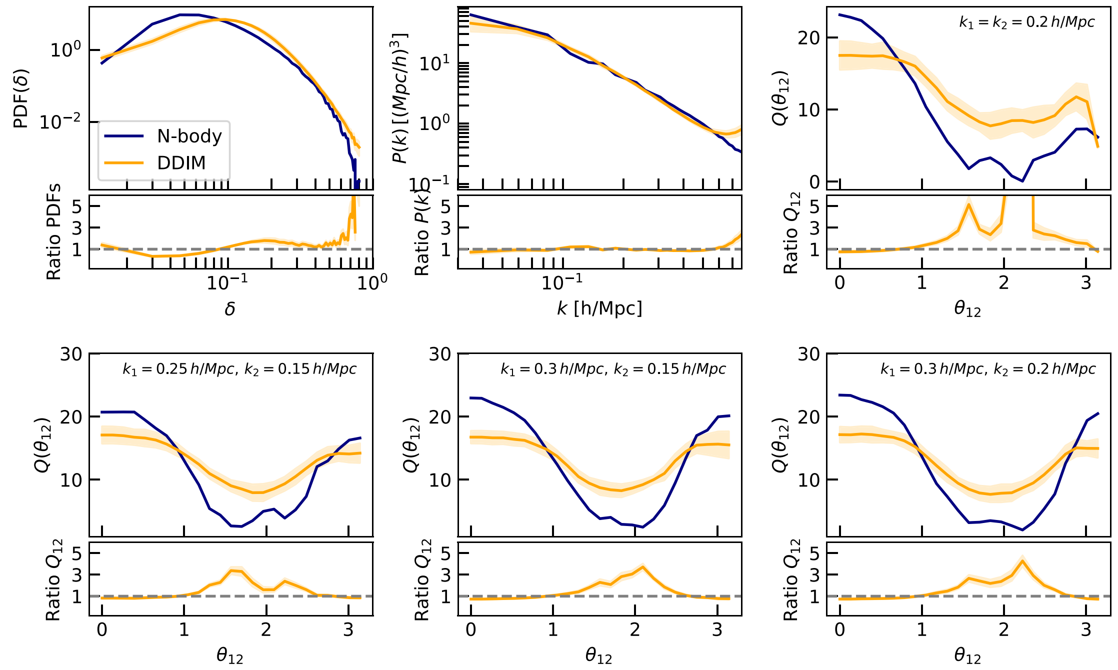


Figure 7. Summary statistics for generated fields with DDIM corresponding to the fiducial value ($\Omega = 0.5$, $\sigma_8 = 0.7$, $h = 0.2$, and $f_{R0} = 0.1$). Each panel shows the probabilistic distribution function PDF with 50 bins, power spectrum $P(k)$, and four bispectra configuration $Q(\theta_{12})$. Solid orange line represents the mean over 50 generative samples, while the orange region defines the standard deviation. Bottom plots illustrate the percent error for each summary statistics.

in cosmology. The demonstrated consistency with higher-order statistics underscores the model ability to capture the complex, non-Gaussian nature of the cosmic web, a significant improvement over methods that rely solely on two-point statistics. This capability is particularly relevant for studying in future phenomena sensitive to the details of structure formation, such as gravity model, galaxy formation, weak lensing and developing parameter estimation [75–77]. However, our current model exhibits limitations, particularly at lower wavenumbers. This increased uncertainty stems from the limited volume of the training simulations.

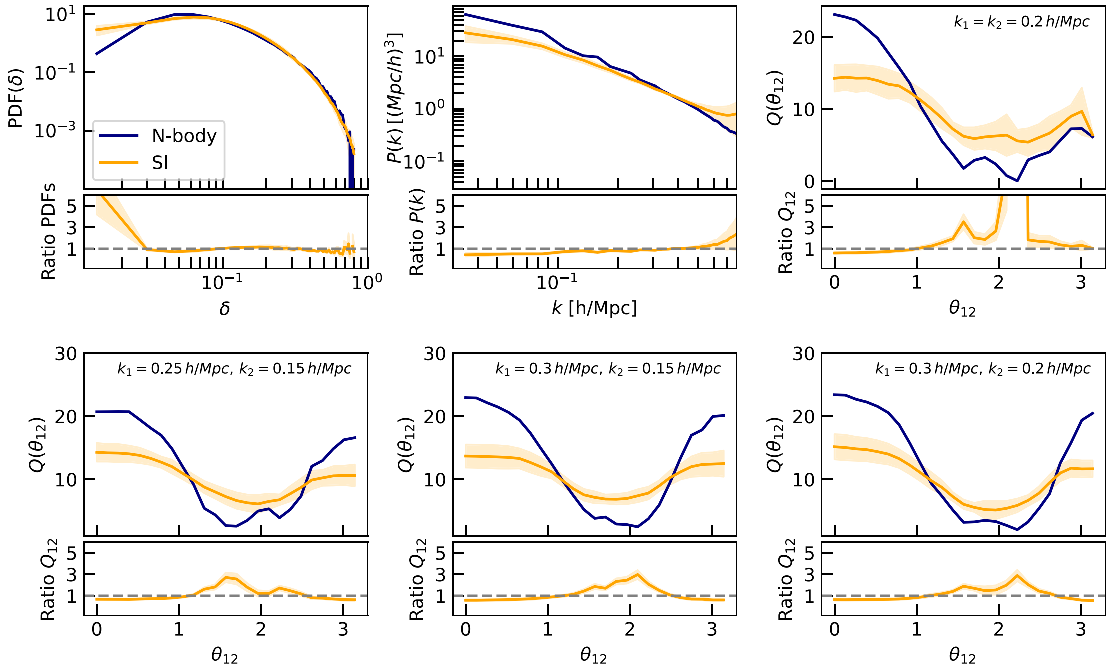


Figure 8. Summary statistics for generated fields with SI corresponding to the fiducial value ($\Omega = 0.5$, $\sigma_8 = 0.7$, $h = 0.2$, and $f_{R0} = 0.1$). Each panel shows the probabilistic distribution function PDF with 50 bins, power spectrum $P(k)$, and four bispectra configuration $Q(\theta_{12})$. Solid orange line represents the mean over 50 generative samples, while the orange region defines the standard deviation. Bottom plots illustrate the percent error for each summary statistics.

Table 2. Assessment of the generative models through the test set, with bold values indicating superior performance. The results show that DDPM achieves the lowest error for most statistical moments, outperforming the majority of the models. However, SI performs best in terms of PDF accuracy. Additionally, DDIM stands out by generating synthetic datasets in just 10 s. Here we present the results specifically for the most standard bispectrum configuration $k_1 = 2k_2$. All metrics are dimensionless due to normalisation.

	Power spectra			Bispectra ($k_1 = 2k_2 = 0.3$)			PDF			
Model	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2	Inference time
DDPM	**61.67**	**1.70**	**0.89**	**6.95**	**2.21**	**0.81**	0.31	0.27	0.83	3 m 20.2 s.
DDIM	69.25	1.86	0.80	16.35	3.42	0.72	1.02	0.53	0.41	**9.9 s**
SI	115.14	2.29	0.72	22.56	3.97	0.64	**0.14**	**0.18**	**0.89**	45.9 s

The relatively small box size restricts the representation of large-scale structures, leading to less accurate predictions on these scales. This limitation highlights the critical need for training data that encompasses a wider range of scales to capture the full spectrum of cosmic structures. Future work will therefore prioritise training our models on significantly larger simulations, which will provide access to a broader range of wave modes and improve the model performance in the low-wavenumber regime. This will be crucial for accurately modelling large-scale structure and its impact on cosmological observables [17, 75, 78]. Recently, people have reported that diffusion models are proposed to produce samples that replicate training data suggesting memorisation behaviour [63]. To handle this issue, we adapt the classifier-free diffusion technique in the diffusion guidance as a proposal given in [79], where they observed that the memorisation can be triggered by conditioning training data on completely random and uninformative labels. Also, we developed different experiments training models only with the augmented dataset (including flip-right-left and Gaussian noise additions), finding quite similar results to those when training data also includes the original images. We plan to transition to latent diffusion models in future studies to address the computational challenges associated with larger simulations. This approach offers a compelling pathway to enhance both training and generation efficiency. By learning a latent space that is perceptually equivalent to the simulation space, we can operate in a lower-dimensional space, significantly reducing the computational cost. The core assumption of latent diffusion, that noise perturbation in simulation and latent spaces are compatible with the generative process, allows for efficient sampling and manipulation of the latent representation. This will enable us to train on larger and more complex simulations, ultimately leading to a more robust and accurate generative model [80, 81]. Furthermore, exploring the impact of baryonic physics and different feedback mechanisms is essential for a complete understanding of structure formation. We intend to extend our analysis by training our models on data from the CAMELS simulation suite [82, 83], utilising different astrophysical feedback prescriptions such as those implemented in the IllustrisTNG [84] simulations. This will allow us to investigate the influence of baryonic processes on the dark matter distribution and to develop a more comprehensive model of cosmic structure. Additionally, we plan to incorporate the effects of observational distortions, such as redshift-space distortions and lensing effects, into our model. This will bring our generated density fields closer to observable quantities and enhance their utility for cosmological analyses. Finally, we are particularly interested in leveraging our generative model for parameter inference. By combining our model with MCMC techniques, we can potentially constrain cosmological parameters and explore the degeneracy between different cosmological models [56, 85, 86]. The ability to generate realistic density fields efficiently opens up new possibilities for exploring the likelihood surface and constraining the cosmological parameters faster.

7. Conclusions

In this work, we have explored the application of conditional generative modelling, specifically using DDPMs, along with their accelerated variant DDIMs, and SIs to generate 3D dark matter density fields. Our analysis demonstrates the significant potential of diffusion models for this task. We demonstrate that DDPM excels in capturing the complex statistical properties of these fields, accurately reproducing both the power spectrum and bispectrum, even for configurations not explicitly encountered during training. However, the computational cost associated with DDPM iterative generation process presents a significant limitation. While DDIM offers a substantial speed-up in sample generation, it comes at the expense of a slight reduction in accuracy, particularly at larger wave modes. This trade-off between speed and accuracy is crucial and must be carefully considered depending on the specific application. Finally, the SI model, while capturing some trends in the bispectrum and successfully reconstructing the probability distribution function for most samples, exhibits lower overall performance compared to both DDPM and DDIM, especially in capturing higher statistical moments. Our quantitative evaluation confirms the superior performance of DDPM, followed by DDIM, highlighting the potential of diffusion-based generative models for cosmological applications. Our emulator is designed to accelerate cosmological inference in scenarios where traditional simulations are computationally expensive. The methods reported in this work might rapidly model modified gravity simulations and test their statistical summaries against observations. These models can also enable efficient sampling of high-dimensional cosmological parameter spaces by replacing iterative simulations with fast-emulated predictions. This emulator is recommended to be used within the trained ranges of cosmological parameters mentioned in table 1, to avoid hallucinations. Future work will focus on mitigating the limitations of DDPMs computational cost, potentially through further refinements of DDIM or exploration of other accelerated sampling techniques, and further exploring the use of these models for parameter inference and other key cosmological tasks. Additionally, addressing the low-wavenumber discrepancies observed in all models due to finite simulation volume will be a priority, requiring the use of larger training simulations. Our code, and scripts used to produce the results in this paper can be found at Github.

Data availability statement

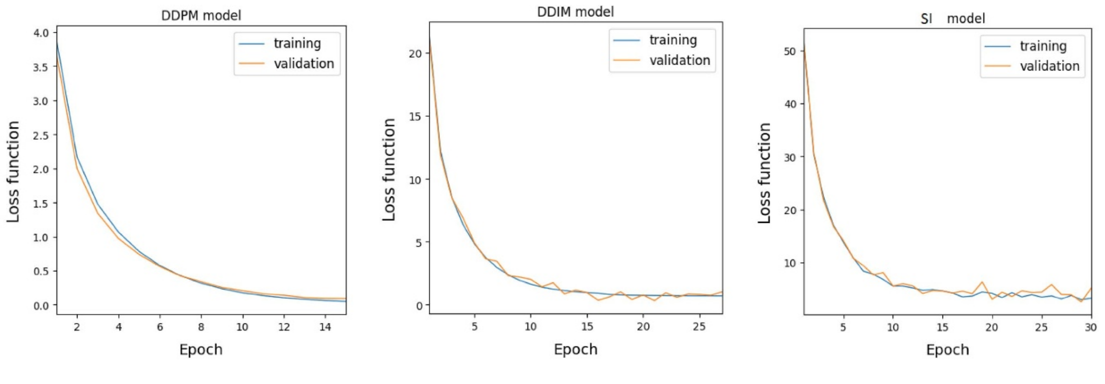
The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.10555349.

Acknowledgments

This paper is based on work supported by the Google Cloud Research Credits program with the Award GCP19980904 granted to HJ Hortúa. JE García-Farieta research was financially supported by the project ‘Plan Complementario de I+D+i en el área de Astrofísica’ funded by the European Union within the framework of the Recovery, Transformation and Resilience Plan—NextGenerationEU and by the Regional Government of Andalucía (Reference AST22_00001).

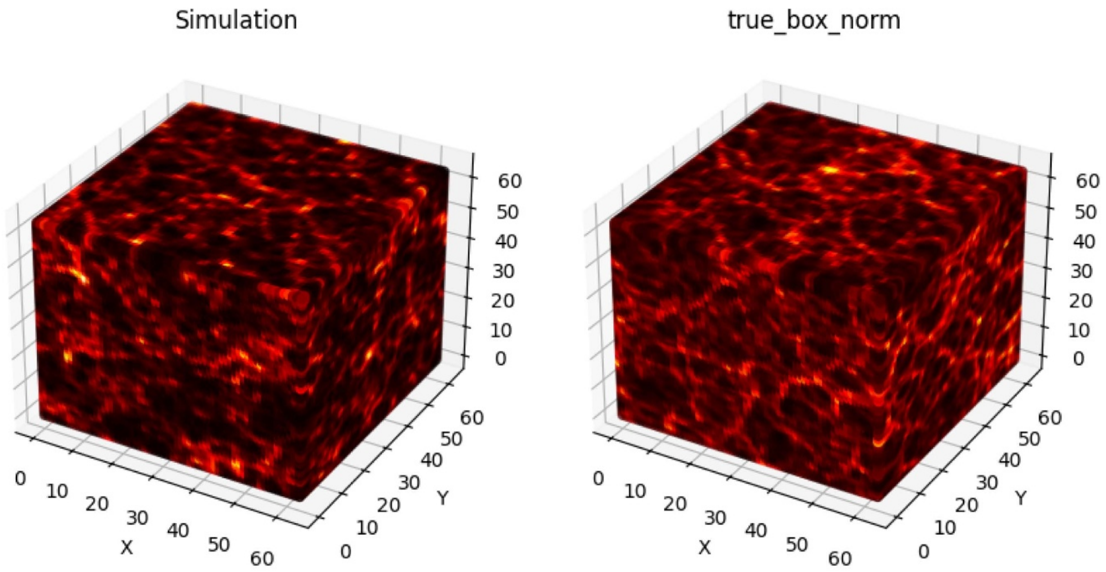
Appendix. Convergence process of loss function

The training loss curves for each model shown in figure A1 reveal distinct convergence behaviours over epochs. DDPM demonstrate significantly faster convergence, consistently reaching lower final loss values compared to DDIM and SI. The latter exhibit higher loss values during training, accompanied by



(a) DDPM model (b) DDIM model (c) SI model

Figure A1. Training loss for each model across epochs. The DDPM models exhibit faster convergence, achieving lower loss values, while DDIM and SI show higher loss values with stochastic fluctuations during training.



(a) Generated box (b) Ground truth box

Figure A2. Comparison between visually the 3D output of the generative model and ground truth.

pronounced stochastic fluctuations. These instabilities may arise from the dependence on stochastic interpolation, which introduces additional variance in gradient updates. This implies that SI requires modified optimisation strategies to achieve performance comparable to that of DDPM. Figure A2 compares the ground truth cosmological simulation box with a synthetic realisation generated by the DDPM model. The model successfully captures key large-scale structures, including filamentary networks, cosmic voids, and high-density regions. Figure A2(a) shows one representative example from multiple realisations produced during inference, demonstrating the models ability to reproduce statistically plausible cosmological fields. Note that the generated boxes appear different from the ground truth because each inference process produces a distinct realisation of the field. However, while individual realisations may vary, their statistical properties remain similar to those of the ground truth.

ORCID iDs

Julieth K Riveros 0009-0009-2627-5516
Paola A Saavedra 0009-0006-6623-1057
Héctor J Hortúa 0000-0002-3396-2404

References

[1] Aghamousa A *et al* (DESI Collaboration) 2016 arXiv:1611.00036
[2] Laureijs R *et al* 2011 arXiv:1110.3193
[3] Amendola L *et al* (The Euclid Theory Working Group) 2018 *Living Rev. Relativ.* **21** 2
[4] Jones R L *et al* (Vera C Rubin Observatory LSST Solar System Science Collaboration) 2020 arXiv:2009.07653
[5] Akeson R *et al* 2019 arXiv:1902.05569
[6] Bacon D J *et al* (Square Kilometre Array Cosmology Science Working Group) 2020 *Publ. Astron. Soc. Aust.* **37** e007
[7] Bernardeau F, Colombi S, Gaztañaga E and Scoccimarro R 2002 *Phys. Rep.* **367** 1–248
[8] Blas D, Garny M and Konstandin T 2014 *J. Cosmol. Astropart. Phys.* JCAP01(2014)010
[9] Osato K, Nishimichi T, Bernardeau F and Taruya A 2019 *Phys. Rev. D* **99** 063530
[10] Crocce M and Scoccimarro R 2006 *Phys. Rev. D* **73** 063519
[11] Carlson J, White M and Padmanabhan N 2009 *Phys. Rev. D* **80** 043531
[12] Carrasco J J M, Hertzberg M P and Senatore L 2012 *J. High Energy Phys.* JHEP09(2012)082
[13] Baumann D, Nicolis A, Senatore L and Zaldarriaga M 2012 *J. Cosmol. Astropart. Phys.* JCAP07(2012)051
[14] Konstandin T, Porto R A and Rubira H 2019 *J. Cosmol. Astropart. Phys.* JCAP11(2019)027
[15] Cabass G, Ivanov M M, Lewandowski M, Mirbabayi M and Simonović M 2023 *Phys. Dark Universe* **40** 101193
[16] Saadeh D, Koyama K and Morice-Atkinson X 2025 *Mon. Not. R. Astron. Soc.* **537** 448–63
[17] Jamieson D, Li Y, de Oliveira R A, Villaescusa-Navarro F, Ho S and Spergel D N 2023 *Astrophys. J.* **952** 145
[18] Ullmo M, Aghnim N, Decelle A and Aragon-Calvo M 2024 arXiv:2403.02171
[19] Ullmo M, Decelle A and Aghanim N 2021 *Astron. Astrophys.* **651** A46
[20] Rothschild T, Nagai D, Aung H, Green S B, Ntampaka M and ZuHone J 2022 *Mon. Not. R. Astron. Soc.* **513** 333–44
[21] Arcelin B, Doux C, Aubourg E and Roucelle C (LSST Dark Energy Science Collaboration) 2021 *Mon. Not. R. Astron. Soc.* **500** 531–47
[22] Kwon K J and Hahn C 2024 *Astrophys. J.* **976** 76
[23] Mootoovaloo A, García-García C, Alonso D and Ruiz-Zapatero J 2025 *Mon. Not. R. Astron. Soc.* **536** 190–202
[24] Rouhiainen A, Giri U and Münchmeyer M 2021 arXiv:2105.12024
[25] Hassan S *et al* 2022 *Astrophys. J.* **937** 83
[26] Bhambra P, Joachimi B, Lahav O and Piras D 2025 *Mon. Not. R. Astron. Soc.* **536** 3138–57
[27] Gondhalekar Y, Bose S, Li B and Cuesta-Lazaro C 2025 *Mon. Not. R. Astron. Soc.* **536** 1408–27
[28] Rodríguez A C, Kacprzak T, Lucchi A, Amara A, Sgier R, Fluri J, Hofmann T and Réfrégier A 2018 *Comput. Astrophys. Cosmol.* **5** 4
[29] Perraudin N, Srivastava A, Lucchi A, Kacprzak T, Hofmann T and Réfrégier A 2019 *Comput. Astrophys. Cosmol.* **6** 5
[30] Schaurecker D, Li Y, Tinker J, Ho S and Refregier A 2021 arXiv:2111.06393
[31] Yiu T W H, Fluri J and Kacprzak T 2022 *J. Cosmol. Astropart. Phys.* JCAP12(2022)013
[32] Perraudin N, Marcon S, Lucchi A and Kacprzak T 2020 arXiv:2004.08139
[33] Tamosiunas A, Winther H A, Koyama K, Bacon D J, Nichol R C and Mawdsley B 2021 *Mon. Not. R. Astron. Soc.* **506** 3049–67
[34] Andrianomena S, Hassan S and Villaescusa-Navarro F 2024 arXiv:2402.10997
[35] Andrianomena S, Villaescusa-Navarro F and Hassan S 2022 arXiv:2211.05000
[36] Kodi Ramanah D, Charnock T, Villaescusa-Navarro F and Wandelt B D 2020 *Mon. Not. R. Astron. Soc.* **495** 4227–36
[37] Diao K and Mao Y 2023 arXiv:2307.04976
[38] Mustafa M, Bard D, Bhimji W, Lukić Z, Al-Rfou R and Kratochvil J M 2019 *Comput. Astrophys. Cosmol.* **6** 1
[39] Zhang X, Lachance P, Dasgupta A, Croft R A C, Di Matteo T, Ni Y, Bird S and Li Y 2024 arXiv:2408.09051
[40] GM H, Gourisaria M K, Pandey M and Rautaray S S 2020 *Comput. Sci. Rev.* **38** 100285
[41] Salakhutdinov R 2015 *Annu. Rev. Stat. Appl.* **2** 361–85
[42] Nguyen T *et al* 2024 arXiv:2409.02980
[43] Mudur N, Cuesta-Lazaro C and Finkbeiner D P 2023 arXiv:2312.07534
[44] Rouhiainen A, Münchmeyer M, Shiu G, Gira M and Lee K 2024 *Phys. Rev. D* **109** 123536
[45] Hassan S and Andrianomena S 2023 arXiv:2311.00833
[46] Mudur N and Finkbeiner D P 2022 Can denoising diffusion probabilistic models generate realistic astrophysical fields? (arXiv:2211.12444)
[47] Mudur N, Cuesta-Lazaro C and Finkbeiner D P 2023 Cosmological field emulation and parameter inference with diffusion models (arXiv:2312.07534)
[48] Zhao X, Ting Y S, Diao K and Mao Y 2023 Can diffusion model conditionally generate astrophysical images? (arXiv:2307.09568)
[49] Croitoru F-A, Hondru V, Ionescu R T and Shah M 2023 *IEEE Trans. Pattern Anal. Mach. Intell.* **45** 10850–69
[50] Cao H, Tan C, Gao Z, Xu Y, Chen G, Heng P-A and Li S Z 2024 *IEEE Trans. Knowl. Data Eng.* **36** 2814–30
[51] Ho J, Jain A and Abbeel P 2020 arXiv:2006.11239
[52] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks (arXiv:1406.2661)
[53] Lamb A 2021 A brief introduction to generative models (arXiv:2103.00265)
[54] Schanz A, List F and Hahn O 2023 Stochastic super-resolution of cosmological simulations with denoising diffusion models (arXiv:2310.06929)
[55] Sabti N, Reddy R, Muñoz J B, Mishra-Sharma S and Youn T 2024 A generative modeling approach to reconstructing 21-cm tomographic data (arXiv:2407.21097)
[56] Mudur N, Cuesta-Lazaro C and Finkbeiner D P 2024 Diffusion-HMC: parameter inference with diffusion model driven Hamiltonian Monte Carlo (arXiv:2405.05255)
[57] Weng L 2021 Diffusion models: a comprehensive overview (available at: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/) (Accessed 9 January 2025)
[58] Welling M and Teh Y W 2011 Bayesian learning via stochastic gradient Langevin dynamics *Int. Conf. on Machine Learning* (available at: https://api.semanticscholar.org/CorpusID:2178983)
[59] Song J, Meng C and Ermon S 2022 Denoising diffusion implicit models (arXiv:2010.02502)
[60] Song J, Meng C and Ermon S 2021 Denoising diffusion implicit models *Int. Conf. on Learning Representations* (available at: https://openreview.net/forum?id=St1giarCHLP)

[61] Zhang G, Ji J, Zhang Y, Yu M, Jaakkola T and Chang S 2023 Towards coherent image inpainting using denoising diffusion implicit models (arXiv:2304.03322)

[62] Ho J and Salimans T 2022 Classifier-free diffusion guidance (arXiv:2207.12598)

[63] van den Burg G J J and Williams C K I 2021 On memorization in probabilistic deep generative models (arXiv:2106.03216)

[64] Albergo M S, Boffi N M and Vanden-Eijnden E 2023 Stochastic interpolants: a unifying framework for flows and diffusions (arXiv:2303.08797)

[65] Albergo M S and Vanden-Eijnden E 2023 Building normalizing flows with stochastic interpolants (arXiv:2209.15571)

[66] Chen Y, Goldstein M, Hua M, Albergo M S, Boffi N M and Vanden-Eijnden E 2024 Probabilistic forecasting with stochastic interpolants and Föllmer processes (arXiv:2403.13724)

[67] García-Farieta J E, Hortúa H J and Kitaura F-S 2024 *Astron. Astrophys.* **684** A100

[68] Tassev S, Zaldarriaga M and Eisenstein D J 2013 *J. Cosmol. Astropart. Phys.* JCAP06(2013)036

[69] Koda J, Blake C, Beutler F, Kazin E and Marin F 2016 *Mon. Not. R. Astron. Soc.* **459** 2118–29

[70] Winther H A, Koyama K, Manera M, Wright B S and Zhao G-B 2017 *J. Cosmol. Astropart. Phys.* JCAP08(2017)006

[71] Howlett C, Manera M and Percival W J 2015 *Astron. Comput.* **12** 109–26

[72] Hu W and Sawicki I 2007 *Phys. Rev. D* **76** 064004

[73] Aghanim N *et al* (Planck Collaboration) 2020 *Astron. Astrophys.* **641** A6

[74] Bengio Y, Courville A and Vincent P 2014 Representation learning: a review and new perspectives (arXiv:1206.5538)

[75] García-Farieta J E, Balaguera-Antolínez A and Kitaura F-S 2024 *Astron. Astrophys.* **690** A27

[76] Ono V, Park C F, Mudur N, Ni Y, Cuesta-Lazaro C and Villaescusa-Navarro F 2024 Debiasing with diffusion: probabilistic reconstruction of Dark Matter fields from galaxies with CAMELS (arXiv:2403.10648)

[77] Andrianomena S, Hassan S and Villaescusa-Navarro F 2024 Cosmological multifield emulator (arXiv:2402.10997)

[78] Sharma D, Dai B, Villaescusa-Navarro F and Seljak U 2024 A field-level emulator for modeling baryonic effects across hydrodynamic simulations (arXiv:2401.15891)

[79] Gu X, Du C, Pang T, Li C, Lin M and Wang Y 2025 On memorization in diffusion models (arXiv:2310.02664)

[80] Rombach R, Blattmann A, Lorenz D, Esser P and Ommer B 2022 High-resolution image synthesis with latent diffusion models (arXiv:2112.10752)

[81] Podell D, English Z, Lacey K, Blattmann A, Dockhorn T, Müller J, Penna J and Rombach R 2023 SDXL: improving latent diffusion models for high-resolution image synthesis (arXiv:2307.01952)

[82] Villaescusa-Navarro F *et al* 2021 *Astrophys. J.* **915** 71

[83] Villaescusa-Navarro F *et al* 2021 arXiv:2109.10915

[84] Springel V *et al* 2017 *Mon. Not. R. Astron. Soc.* **475** 676–98

[85] Hortúa H J, Volpi R, Marinelli D and Malagò L 2020 *Phys. Rev. D* **102** 103509

[86] Hortúa H J, Volpi R and Malagò L 2020 Parameters estimation from the 21 cm signal using variational inference (arXiv:2005.02299)