Learning the Halo Occupation Distribution over Multiple Cosmologies with Diffusion Models

Antoine Bourdin[a], Laurence Perreault-Levasseur[b,c]

[a]Department of Physics, McGill University; [b] Trottier Space Institute;
[c] Department of Physics, Université de Montreal

Abstract

Large-scale hydrodynamic simulations of the universe are powerful tools for modeling the evolution of baryonic and dark matter, shedding light on galaxy formation, dark matter structures, and the cosmic web. However, these simulations are computationally intensive, consuming on the order of 100 million CPU hours. In contrast, N-body simulations are more computationally efficient as they model fewer effects, only including the gravitational interactions between baryonic matter and dark matter particles. It is common practice in the field to estimate hydro-simulations by adding baryonic matter onto these N-body simulations via models known as Halo Occupation Distribution (HOD). These empirical models estimate galaxy distributions, yet they fall short in capturing the crucial effects of the galaxy environment, which significantly shapes galaxy formation. Moreover, their precision is becoming inadequate for upcoming observational missions due to their inability to properly capture non-linear physics on small scales. To bridge this gap, we propose employing machine learning score-based diffusion models which have the ability to learn high dimensional distributions. We utilize the Camels Multified Dataset—a collection of one thousand hydrodynamic and N-body simulations spanning various cosmological and astrophysical parameters—to train a diffusion network. This network generates samples representing the associated galaxy distribution from a given N-body simulation. Furthermore, using architectures known as convolutional neural networks we aim to capture environmental factors. Through this approach, we present a probabilistic model which produces galaxy counts for small scale N-body simulation over multiple cosmologies, while being consistent with hydrodynamic simulations.

I. Introduction

Within the past decade, advancements in computational power have revolutionized cosmology, empowering the creation of unprecedentedly realistic simulations of our universe. These advancements have been particularly notable through the use of hydrodynamic simulations [1–3]. These simulations follow the cosmological model Lambda-Cold Dark Matter (ΛCDM) which incorporates baryonic matter, dark matter and dark energy (represented by Λ). The term cold for dark matter means that it weakly interacts with baryonic matter, but is still influenced by gravity. These hydro-simulations have a set percentage of baryonic and dark matter and are evolved from the early universe up until nowadays. During their evolution dark matter particles clump together forming over densities known as halos. Overtime these halos may merge violently and grow in size, or accumulate smaller concentrations of dark matter within the larger halos known as subhalos. Baryonic matter is gravitationally attracted to these subhalos with galaxy formations happening within them and clusters of galaxies being found within halos. Currently, hydro-simulations are the only type of simulations which are able to model galaxies whose properties match the ones we observe in our universe. This makes them one of the most valuable tools we have to study galaxy formation from the early universe until present day. However, they come at the price of being computationally expensive with one hydro-simulation requiring on the order of one-hundred million CPU hours. This is due to them simultaneously evolving a large number of non-linear processes.

The counterpart of hydro-simulations are N-body simulations. N-body simulations only take into account gravitational interactions between dark matter and baryons. Therefore, N-body simulations do not contain galaxies as thermal and electromagnetic interactions are needed to model their formation. This makes N-body simulations computationally cheaper on the order of a thousand CPU hours. Since galaxy formation happens within dark matter halos, cosmologists seek to evolve these cheaper N-body simulations and then add galaxies on top of the dark matter. To do so we need a connection between dark matter halos and galaxies. Halo Occupation Distributions (HOD) are empirical models which estimate the number of galaxies within halos based on their mass. Many of the currently used HOD models were developed in the early 2000's [4, 5]. However, it has been shown that the level of precision of current HOD models is below what we need it to be in order to extract valuable information for upcoming surveys. This is partially due to their inability to model small scales correctly. Furthermore, it was shown that predicting galaxy counts solely based on halo mass is a very simplistic assumption and that taking into account the environment outside the halo can improve the precision of HODs [6].

The motivation for our work is the following. It has been shown that convolutional neural networks (CNN) are able to outperform current HOD models [7]. It is supposed part of this success is due to CNN's ability to capture information about the halo and the environment around it instead of just depending on its mass. However, these networks were not probabilistic in nature meaning a distribution and an uncertainty could not be extracted from them. Recent advances in

machine learning have introduced probabilistic generative models known as score-based diffusion models [8]. These models learn distributions and are able to produce samples which can be used to characterize uncertainties. Furthermore, it was shown that these diffusion models are able to marginalize over different cosmological parameters [9]. Meaning they can produce samples whose features are dependent on the parameters of the ΛCDM model. This could enable future work to relate the spatial distribution of galaxies we observe with the ones generated by the model. Since the samples from the diffusion model are dependent on cosmological parameters it could allow us to infer the parameters of the ΛCDM model which describe our universe.

Within this thesis we demonstrate that these diffusion models allow us to have a probabilistic model between dark matter and galaxy counts over multiple cosmologies. Furthermore, we show that our model is consistent with modern hydrodynamic simulations on small scales. We note that this is not feasible with current HOD models. We use the LH CAMELS Multifield Dataset [10], which contains 1000 hydro-simulations and N-body simultaions: with each pair having the same cosmological parameters and initial conditions. Each of the 1000 simulations have different cosmological parameters. The network is trained to learned the distribution of galaxy counts (based on the results of the hydro-simulations) conditioned on a dark matter simulation over different cosmologies.

II. Background

A. Halo Occupancy Distribution

Within this section we give a brief overview of the development of Halo Occupancy Distribution (HOD) models along with the baseline model often used nowadays. The equations describing the HOD model were not derived from first principles but rather by observing the correlation between dark matter halos and galaxy properties in simulations. HODs were developed in the 2000's and at the time there was a very limited amount of hydrodynamic simulations due to computational limitations. Instead Semi-Analytic (SA) models at the time played a key role. SA models included various types of baryonic physics but these were approximations of astrophysical processes and did not fully evolve non-linear dynamics like in hydro-simulations. This allowed to simulate galaxy formation while having significantly lower runtime compared to hydro-simulations.

One of the first SA model was Galform [11]. The basis was to create what are known as merger trees. Given a dark matter simulation evolved to present day it estimated how the halos had merged over time since the early universe via Monte Carlo simulations. Based on the initial configurations of the halos, it placed gas within them which overtime would collapse to form galaxies. Galform was able to predict the gas mass, galaxy counts, speed of galaxies and galaxy luminosities for each halo. However, the price to pay for approximating non-linear processes was that the parameters which affected galaxy formation, such as galaxy ellipticity, became free parameters. Properly constraining

the values of these free parameters is crucial to simulate galaxies whose properties match the ones we observe in our universe. However, due to the large number of free parameters within Galform, cosmologists were not able to fully constrain all of them.

Nonetheless, Galform gave a computationally efficient framework to explore the correlation between dark matter halo mass and the number of galaxies within halos. It was found that below a threshold halo mass no galaxies were found. After that threshold the galaxies which populate the halos can be separated into central and satellite galaxies. The first galaxy to populate a halo would be a central galaxy. It would be located at the minima of the gravitational potential of the halo. Then if other galaxies were to form within the halo, they would orbit around the central galaxy. The expected number of central and satellite galaxies within a halo of mass M respectively were found to follow the relations [5],

$$\begin{aligned} \langle N_{\text{cen}}(M)\rangle &= \frac{1}{2}\left[1+\text{erf}\left(\frac{\log M-\log M_{\text{min}}}{\sigma_{\text{logM}}}\right)\right] \\ \langle N_{\text{sat}}(M)\rangle &= \langle N_{\text{cen}}(M)\rangle\left(\frac{M-M_0}{M_1}\right)^{\alpha} \end{aligned} \tag{1}$$

where M_{min} is the minimum halo mass that can host a central galaxies, erf is the Gauss error function and σ_{logM} is a parameter that controls how fast $\langle N_{\text{cen}}(M)\rangle$ increases before plateauing at one. This function demonstrates the threshold behavior previously discussed, and $\langle N_{\text{cen}}(M)\rangle$ can be seen as a smoothed step function. Furthermore, we see that $\langle N_{\text{sat}}(M)\rangle$ is a smooth continuation of $\langle N_{\text{cen}}(M)\rangle$ and it follows a power law controlled by the parameter α. Similarly, M_0 is a parameter which sets the minimum halo mass in order to have satellite galaxies. This parameter is usually set to the halo mass that guarantees the halo has a central galaxy, meaning the mass when $\langle N_{\text{cen}}(M)\rangle = 1$. Finally, the parameter M_1 controls how fast $\langle N_{\text{cen}}(M)\rangle$ increases.

Equation (1) only gives expected number for central and satellite galaxies. If we were to populate a dark matter simulation with galaxies using HODs we need a distribution associated with each expected value. To determine if a halo has a central galaxy the Bernoulli distribution is used. The Bernoulli distributions only has outcomes 0 or 1 (failure or success) with probability p of returning a 1. In our case, the probability is $p = \langle N_{\text{cen}}(M)\rangle$ and drawing from the distribution a 1 represents a halo being assigned a central galaxy while the outcome of 0 means the halo does not have a central galaxy. As for the satellite galaxies they follow a Poisson distribution with expected value $\lambda = \langle N_{\text{sat}}(M)\rangle$. The reason for satellite galaxies following a Poisson distribution is due to observations depicting that galaxies are roughly distributed in a Poisson manner [12]. These distributions are what give HODs their probabilistic nature, and why applying an HOD multiple times to the same N-body simulation gives different possible realizations of galaxy counts.

The baseline HOD described is fully characterized with the five parameters in equation (1). Although simple, this model saw success in the 2000s when it was able to model the large scale

clustering of galaxies in the SDSS survey [4]. However, the five parameters do not have an agreed upon value. Each parameters value is changed to best fit certain observations or simulations, meaning the model does not generalize. Furthermore, different cosmological parameters from the ΛCDM model are not included in the parameterization of HODs, even though it is known that different cosmological parameters will affect how matter clusters and hence how galaxies form.

B. Neural Networks

In essence, neural networks are function approximators which are optimized to fit to a set of data. Neural networks have three important elements: an architecture, a loss function, and an optimizer. One of the simplest architectures is a multi-layer perceptron (MLP), which takes an input vector $x \in \mathbb{R}^{\mathbb{D}}$ and then passes it through a series of linear transformations (with matrices) and non-linear functions. This series of transformations can be used to estimate complex functions [13]. The non-linear functions are predetermined, while the entries of the matrices, known as weights, are unknown and need to be optimized to best fit to the data.

Let us have a set of n input vectors $\{x_i\}_{i=1}^n$ which are commonly known as a training set in the literature. For the sake of an example let us say that x_i are the observed luminosity of galaxies. Now, for each input vector we need its corresponding output denoted by the set $\{f(x_i)\}_{i=1}^n$. So for each observed galaxy luminosity we also have the deduced galaxy mass. Now, what if we wanted to predict the galaxy mass for a galaxy luminosity which isn't in our training set. Then we need a fitting function that relates any galaxy luminosity to its mass. This fitting function is what is being learned by the neural network.

To do so take an architecture, such as an MLP, give it a point in your training set x_1 (a galaxy luminosity). This architecture is untrained, meaning the weights are not optimized to represent the fitting function. Therefore, the output of the MLP will be a value which is very far from the true output $f(x_1)$; the associated galaxy mass. We can characterize the distance between the value predicted by the network denoted by $N_\theta(x_1)$, where θ represent the weights of the network, and the true output $f(x_1)$ by calculating the euclidean distance between these two points. This formally is known as a loss function and takes the form,

$$\mathcal{L} = ||N_\theta(x) - f(x)||_2^2 \tag{2}$$

for any point x in the training set. The euclidean norm is represented by $||\cdot||_2$ and it is commonly squared in the literature. This doesn't lose any relative information about distance, but slightly reduces computational memory. Now, we want this loss to be minimized so that our network learns a fitting function which represents our data. To do so we now need to update the weights of the network.

Using the chain rule we can calculate the gradient of the loss with respect to each weight in the

network. This is denoted by $\nabla_{\theta_k}\mathcal{L}$ where θ_i denotes the k-th weight (the k-th matrix element). We can now use this gradient to change all of the weights in our network towards values that will lead to a lower loss. We do so with the formula,

$$\theta'_k = \theta_k - \eta \nabla_{\theta_i}\mathcal{L} \tag{3}$$

where θ'_k are the updated weights and η is a parameter known as the learning rate. The learning rate tells us how big of a step we take in the space of weights. A learning rate which is too large can cause us to overshoot the parameters which minimize the loss. If the learning rate is too low it may take a lot of iterations to reach the minimum. There is no formal way to find the correct learning rate, apart from optimizing the same architecture with different learning rates and seeing which one achieves a lower loss.

To summarize, we pass our input vector x_i into our architecture and compare the network prediction $N_\theta(x_i)$ with the true value $f(x_i)$ using the loss function. Using the gradients of the network weights with respect to the loss we then update (optimize) the network weights with equation (3). We repeat this over our entire training set multiple times until the loss plateaus at a minimum value. This ensures that we have learned a fitting function which best represents our training data. We can now pass new values outside of the training set (new galaxy luminosity) and predict in a principled manner the associated output (new galaxy mass). It is common practice to separate data for which we have the true values into a training set and testing set. The training set is used in the procedure described above. The testing set is passed through the network after its been trained to see if the learned fitting function generalizes properly to data that was not seen during training.

C. Convolutional Neural Networks and the U-Net

A different type of architecture compared to the MLP discussed previously is the convolutional neural network (CNN). The key principal behind a CNN is the ability to learn local spatial information within images. This information is captured with what are known as filters. As depicted in Figure 1 a filter is a matrix which is superimposed on a part of an image. The component values of the filters are multiplied with the corresponding component values of the image and summed together in a similar fashion to a dot product. The output value from this superposition is then stored as an element of an output matrix. The filter is then moved to another patch of the image and repeats the operation until the output matrix is completely filled. This output matrix is most commonly known in the literature as a feature map. The filters essentially summarize local information in the image into a single value which is stored in the feature map.

In an MLP the weights of the network were the matrix entries which created linear transformations. In the case of a CNN, the weights are the different component values in a filter. A CNN can

have multiple filters. Each filter gets passed over the image and creates a feature map. Each filter is responsible for learning key local spatial correlations within the data, which is why the network optimizes the filter values. In a similar fashion to the MLP where the linear transformation is followed by a non-linear transformation, here after applying the filter to an image each component of the feature map goes through a non-linear transformation. CNNs are still fitting functions but they give us the additional ability to capture spatial correlations.

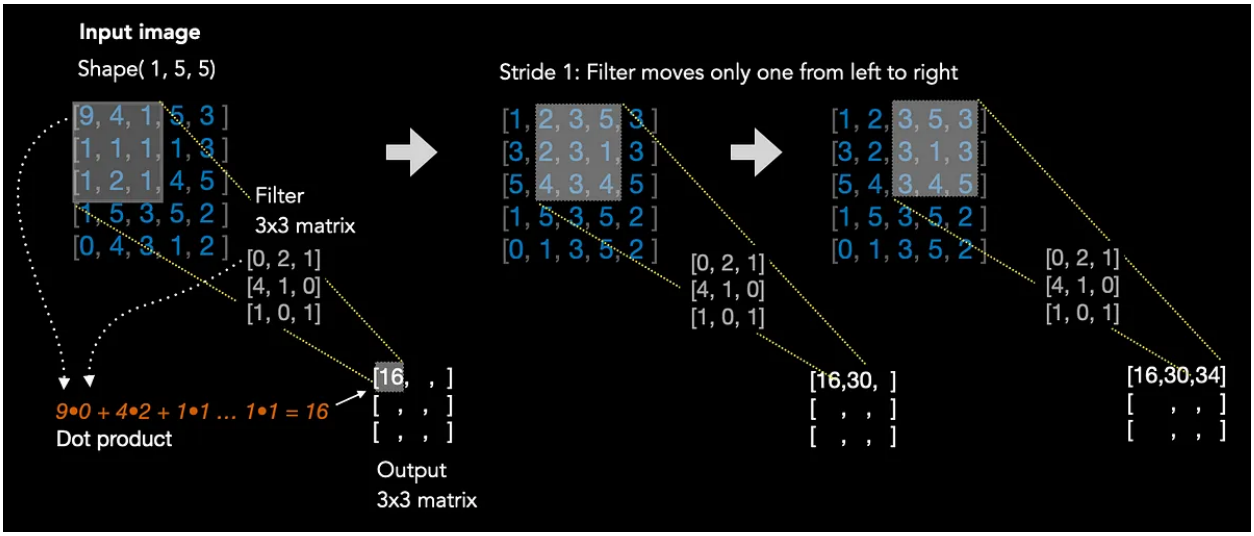

FIG. 1. Filters of size 3x3 being dot producted with the input image, hence, summarizing local information. This operation is repeated on the whole image creating an output matrix of size 3x3 known as a feature map.

CNNs play a key role in one of the most important architectures in machine learning which is known as the U-Net. The U-Net applies a CNN's filters to an image and creates feature maps. Then it applies another CNN with its own new filters to the previously created feature maps. This procedure, illustrated in Figure 2, known as down sampling is repeated a number of times until we have a very small feature map known as a latent space. The latent space can be seen as a data compression of our original image. Furthermore, the different feature maps of different size can be seen as learning different frequency information in an image like a Fourier transform [14]. Let us take the first filter directly applied on the image to be of size F×F. This first filters learn very local information so high frequency information. The created feature map summarizes the information of that F×F

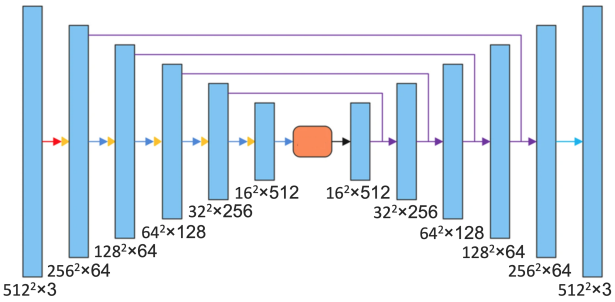

FIG. 2. Example of a U-net. The input image starts on the left and goes through a series of down sampling into a latent space represented by the orange block. The blue rectangles represent the feature maps. The skip connections in purple bring information from the down sampling, and starting from the latent space the network up samples to output a vector field.

patch but with one value. It has reduced the information to a lower resolution meaning a lower frequency. Hence, at each down sampling we learn information about a larger spatial area in the original image meaning we learn lower frequency information.

The second part of the U-net is the upsampling where we go from the smallest feature map and progressively produce an output of the same dimension as the input. This part of the U-net is not learning features in the original image, but rather trying to match the function that would minimize the loss. However, it does so with the help of skip-connections (in purple in Figure 2) which bring in information about the specific frequency which was learned during downsampling. To summarize, a U-net generates a D dimensional output for a D dimensional input. Meaning it learns a fitting function which is a vector field. The down sampling information learns key correlations at different frequencies within the data, and compress it down to a small feature map. This compression is then used to progressively create the vector field along with the help of skip connects which hold different frequency information.

D. Stochastic Differential Equations

We will now shift the focus to stochastic differential equations. These will play a key role in our ability to define score-based models. These score-based models in turn will also require the machine learning knowledge previously developed. Let us first discuss ordinary differential equations (ODE). Perhaps the most well known ODE in physics is Newton's second law,

$$\frac{d^2x}{dt^2} = \frac{F}{m}. \tag{4}$$

Its solution for a scenario with constant force F and mass m is of the form,

$$x(t) = x_0 + v_0 t + \frac{Ft^2}{2m}. \tag{5}$$

Formally known as the equation of motion, $x(t)$ describes the position of an object at a certain time for a certain initial position x_0 and initial velocity v_0. We highlight that the solution to the ODE is a function that returns a single value at a fixed time.

Now let us bring up another concept: randomness. A common example is a random walk. Consider discretized time intervals in steps of δ. Similarly, consider a discretized real line with intervals of $\sqrt{\delta}$. For every time increment δ we either move spatially in the positive direction $\sqrt{\delta}$ with probability 0.5, or in the negative direction $-\sqrt{\delta}$ with probability 0.5. This can be written as,

$$X_i = \begin{cases} \sqrt{\delta}, & \text{probability } \frac{1}{2} \\ -\sqrt{\delta}, & \text{probability } \frac{1}{2} \end{cases} \tag{6}$$

where X_i is a random variable which is a mathematical object defined by a set of possible values along with a probability distribution over this set [15]. Here the set is $\{-\delta, \delta\}$ and the probability

distribution is the discrete probability distribution of 0.5 for either outcome.

Assuming we start at time and position zero, then after n time steps, $t = n\delta$, we can define the position of the process as

$$w(t) = w(n\delta) = \sum_{i=1}^{n} X_i. \tag{7}$$

Now we can no longer describe the position of $w(t)$ with a single value due to the probability associated with each step. Instead we have a set of possible values at time t, or in other words we have a distribution. We can characterize the distribution of position of $w(t)$ by calculating the expected value and variance respectively,

$$\mathbb{E}[w(t)] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} (0.5\sqrt{\delta} - 0.5\sqrt{\delta}) = 0 \tag{8}$$

$$\begin{aligned} \text{Var}\left[w(t)\right] &= \text{Var}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \text{Var}[X_i] \\ &= \sum_{i=1}^{n} 0.5(\sqrt{\delta} - \mathbb{E}[X_i])^2 + 0.5(-\sqrt{\delta} - \mathbb{E}[X_i])^2 = \sum_{i=1}^{n} \delta = n\delta = t \end{aligned} \tag{9}$$

We are justified in taking the expectation and variance inside the sum in equation (8) and (9) because each step is independent (the probability from the previous or future step do not alter the discrete probability distribution describing the current step), hence we can simply calculate the expectation and variance of X_i.

We have just established the expectation and variance of $w(t)$ for discrete time steps. Now, for continuous time we just need to consider an infinite amount of steps n. Which means that we now have infinite X_i. The central limit theorem tells us that in the limit of infinite random variables with finite expectation and variance the distribution converges to a normal distribution [16]. Therefore, in continuous time,

$$w(t) \sim \mathcal{N}(0, t) \tag{10}$$

where $\mathcal{N}$ denotes the normal distribution, and its parameters are the mean and variance. Hence, at any time t the position of the process $w(t)$ is normally distributed with mean zero and variance t. We have just derived what is known as the standard Wiener process; also called Brownian motion.

We are now equipped to discuss stochastic differential equations (SDE). The general form of SDEs is [17],

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(X_t, t)\mathrm{d}w \tag{11}$$

where X_t is a time dependent random variable in $\mathbb{R}^d$, f is a function in $\mathbb{R}^d$ known as the drift term, g is a function in $\mathbb{R}^{d\times d}$ known as the diffusion term and $\mathrm{d}w$ is the infinitesimal form of the Wiener process in $\mathbb{R}^d$. When deriving the Wiener process we were effectively solving equation (11) in one dimension with $f = 0$ and $g = 1$. It is key to understand that because of the $\mathrm{d}w$ term the solution X_t to an SDE is a distribution and no longer a function which returns a single value at time t like for ODEs.

Analytical solutions to SDEs can be complex to derive, and the reader should not try to solve equation (11) like an ODE. This is because the infinitesimal Wiener process is not integrable in the standard Riemann or Lebesgue manner, and Itô calculus is required. However, since the methodology for solving SDEs is not fundamental in order to understand this work (and very lenghty) we will not be developing it here. The interested reader is referred to chapter 4 of [17].

The main takeaway from this section, is that SDEs are differential equations which describe processes which involve randomness. We mathematically encompass this randomness via solutions which are distributions which are a consequence of the probabilistic nature of the Wiener process.

E. VE SDE

In this section we describe the characteristics of a specific SDE which will be of interest for our work. The Variance Exploding (VE) SDE [8] takes the form,

$$\mathrm{d}x = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}\mathrm{d}w \quad \text{where} \quad \sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \tag{12}$$

meaning it has no drift term and its diffusion term is only time dependent. Please note that x here is a random variable, I am changing the notation from X_t to x in order to prevent the cluttering of upcoming expressions. The variables $\sigma_{\min}$ and $\sigma_{\max}$ are parameters to be set. How to set these parameters will be described in section II F. It turns out that the solution to this SDE is [8],

$$p_{0t}(x(t)|x(0)) = \mathcal{N}\left(x(t); x(0), \sigma^2(t)\right). \tag{13}$$

This should be read as given an initial position at time zero $x(0)$, the probability of being at position $x(t)$ is given by the normal distribution with mean $x(0)$ and variance $\sigma^2(t)$ evaluated at $x(t)$. This however, can be extended to entire distributions. Let us have an initial distribution p_0 which we perturb with p_{0t} then at time t our original distribution is described by

$$p_t = p_0 * p_{0t} \tag{14}$$

where $*$ denotes a convolution between the two distributions. One can see equation (13) as being applicable to samples (points) of p_0 while equation (14) applies to the underlying parent distribution.

Now this describes, a process which happens forward in time starting at $t = 0$. It was shown by

[18] that SDEs with affine drift and diffusion terms have an associated time-reversed process. For the VE SDE, the reverse process exists and is

$$\mathrm{d}x = -g^2(t)\nabla_x \log\, p_t(x)\mathrm{d}t + g(t)\mathrm{d}\bar{w} \quad \text{where} \quad g(t) = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}} \tag{15}$$

and $\mathrm{d}\bar{w}$ is the reverse time Wiener process which is still defined by equation (10) but time goes in the opposite direction. There is no closed form solution to the reverse process and hence can only be solved numerically with a starting point at some time $T > 0$ and evolved to $t = 0$ with discretized time step Δt,

$$x_{i-1} = x_i - g^2(t_i)\nabla_{x_i} \log\, p_{t_i}(x_i)\Delta t + g(t_i) z\sqrt{-\Delta t} \tag{16}$$

where Δt is assumed to be negative and $\mathrm{d}\bar{w}$ is discretized with $z\sqrt{-\Delta t}$ where $z \sim \mathcal{N}(0, -\Delta t)$. The closed form solution of the VE SDE for the forward process and the time reversibility is key for score-based models.

F. Score-Based Diffusion

The goal of score-based diffusion models is to learn the underlying data distribution of samples via the mathematics of SDEs along with the help of neural networks. In real life, we do not have access to the parent distribution of the data we collect. Instead we have observations which if sampled in an unbias manner hold information about their underlying parent distribution.

To begin let us consider the unknown parent distribution of some data p_{data} along with a collection of N samples $\{x_i\}_{i=1}^N$. Using the VE SDE we can forward evolve all of the samples with equation (13). This procedure in the literature is usually known as noising the data. This is because the process is simply adding random fluctuations which get larger with time due to the exponential form of $\sigma(t)$ in equation (12).

We choose to evolve our samples on the set time interval $t \in [0, 1]$. Now, equation (14) tells us how a distribution at $t = 0$ is affected by the noising process. Furthermore, we see that if p_{0t} is much larger than p_0 at some time t, then effectively the perturbed distribution p_t is equivalent to p_{0t}. We want this to happen for $t = 1$ such that we have effectively transformed p_{data} into the normal distribution described by p_{0t} in equation (13). It was shown that this happens by setting the parameter $\sigma_{\max}$ in $\sigma(t)$, to the largest euclidean distance between any two sample points [19]. Furthermore, it can be approximated that information about the mean of p_{data} is lost when p_{0t} is large enough and we can estimate the distribution at $t = 1$ to have mean zero. Since $\sigma(t = 1) = \sigma_{\max}$ then we have,

$$p_{t=1} = \mathcal{N}\left(0, \sigma_{\max}^2\right). \tag{17}$$

We now have a closed form distribution at $t = 1$, and we can easily sample from it since it is a

normal distribution. We can now numerically evolve points backward in times from the normal distribution at $t = 1$ until the data distribution at $t = 0$ via equation (15). We have effectively created new samples from the underlying data distribution. We see that $\sigma(t = 0) = \sigma_{\min}$ which is the amount of noise in our samples and should be set to the required precision. This is commonly known as the denoising process since we are going from a white noise Gaussian to samples.

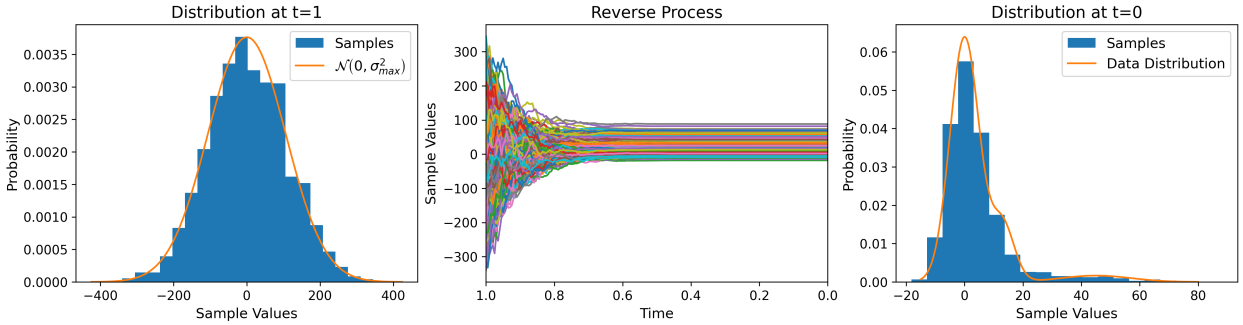

FIG. 3. One dimensional sampling of score-based diffusion. We defined a multi-modal data distribution whose outline can be seen on the right most panel by the orange line. We trained an MLP to learn the score at different times using equation (19). The training data used is 3000 points sampled from the data distribution. We use the parameters $\sigma_{\min} = 0.001$ and $\sigma_{\max} = 106$. The leftmost panel shows the starting distribution to generate new samples. The histogram is of 2000 points sampled from the normal distribution defined at $t = 1$. The second panel shows the reverse process for each of the 2000 points using equation (16) with $\Delta t = 0.005$ from t=1 to t=0. The rightmost panel shows the histogram formed by the backward evolved samples. We see that the generated samples lie within the underlying data distribution as wanted.

We have described the sampling process, however to do so we need the quantity $\nabla_x \log\, p_t(x)$ as can be seen in equation (16). This is known as the score of the distribution at different perturbation times. This is where neural networks come into play. We train a network which takes in data point at a certain time $x(t)$ along with the time t and outputs the score of the perturbed distribution at point x. The logical loss function to train the network is [20, 21],

$$\mathcal{L} = \mathbb{E}_{x\sim p_t(x)} \, ||s_\theta(x,t) - \nabla_x \log\, p_t(x)||_2^2 \tag{18}$$

where $s_\theta(x(t), t)$ is our score network which is being optimized to reduce the expectation between the learned distribution and the true data distribution at any perturbation time. However, we don't actually have access to the true data distribution and cannot calculate p_t. However, it was proven that this loss can be rewritten as [8, 22],

$$\mathcal{L} = \mathbb{E}_{x(0)} \, \mathbb{E}_{p_{0t}(x(t)|x(0))} \, ||s_\theta(x(t),t) - \nabla_{x(t)} \log\, p_{0t}(x(t)|x(0))||_2^2 \tag{19}$$

which can be computed since we have a closed form solution for p_{0t}. The network is trained by uniformly sampling times in the interval $[0, 1]$ and forward evolving already existing samples to the sampled time with equation (13). These forward evolved samples along with their time are passed to the network which optimizes for the score at that time using the loss defined in equation (19).

This is done repetitively such that the samples are evolved to many different times in the interval $[0, 1]$ in order to smoothly learn the score. We show a simple example of score based diffusion in Figure 3.

Now, the strength of score-based diffusion models, is that they can learn the distribution of high dimensional data. As was explained with equation (11), these SDE's can evolve data in $\mathbb{R}^D$, where the dimension D is arbitrary and finite. Similarly, a U-Net can learn the vector field used for the score on arbitrarily sized data. Hence, these models hold the ability to learn the high dimensional distribution the size of images or in our case 3D simulations. One of the main limitations on the dimension of the distribution that can be learned are compute capabilities.

III. Method

The architecture used within our work is the NCSS++ U-net from [8], however we turn the 2D convolutions into 3D convolutions in order to process the 3D simulations. Likewise to [9], we make the architecture conditional. Meaning that during training it takes in a dark matter simulation, the galaxy counts from the hydro-simulation forward evolved to different times, and the corresponding time for the score-based diffusion. Even though the architecture has multiple inputs it only returns one vector field which is the score for the galaxy counts at different times. We only learn the score for the galaxy counts because we want to generate different samples of the galaxy counts for a fix given dark matter simulation just like an HOD. The dark matter simulation is passed through the U-Net because we want the filters to learn important local spatial correlations in both the dark matter simulation and the galaxy counts, along with correlations between the two. When creating new galaxy counts the filters look at the correlations in the given dark matter simulation and use this to determine where to add galaxies in the sample during the reverse diffusion. Looking at the spatial correlation also means looking at the environment in the dark matter simulation, hence allowing the model to not solely focus on halo mass.

The data used are the 1,000 pairs of hydro-simulation and N-body simulations in the LH suite of the Camels Multifield Dataset (CMD) [10] which follow the galaxy formation model of the Astrid simulations [1]. We split these pairs of 1,000 simulations into 900 simulations which are used for training. The remaining 100 N-body simulations are used to generate samples of galaxy counts which we compare with the corresponding galaxy counts from the hydro-simulations. We identify galaxies in the hydro-simulation to create the galaxy counts via the Friends-of-Friends method to identify halos and the SubFind method to find subhalos. Candidate subhalos are said to host a galaxy if their halo mass is greater than $10^{9.5} M_\odot$ and contain baryonic gas. The galaxies identified within the subhalos are then resolved on a 3D grid of 32^3. The simulations in CMD are evolved over a physical volume of $(25\ \text{Mpc/h})^3$. This gives the grid a spatial resolution of about 0.78 Mpc/h which is equivalent to a Nyquist frequency of $k_{\text{Nyq}} \approx 4.02$. This leads us into the small scale regime of $k > 1$ where non-linear dynamics are prominent. Our diffusion model trained on this data had

$\sigma_{\max} = 593$ based on the maximum euclidean distance in the training set and $\sigma_{\min} = 0.001$ which is below the integer precision we need to resolve individual galaxies.

We note that the N-body simulations have two varying cosmological parameters: Ω_m and σ_8 which respectively describe the total matter density and the clustering of matter. The hydro-simulations have the same varying cosmological parameters but also vary astrophysical parameters which effect galaxy formation. Notably ASN_1 and ASN_2 which respectively control star formation rate and galactic wind, while AAGN_1 and AAGN_2 respectively effect black hole kinetic and thermal energies. However, we cannot say that our model is able to marginalize over different astrophysical parameters as the only information we have when generating the galaxy counts is the given dark matter simulations. Since these simulations are only affected by cosmological parameters then our model can only marginalize over the distribution of cosmological parameters. The effect that astrophysical parameters may have on the galaxy count is captured within the uncertainties of the distribution of galaxy counts. Even though, we will be presenting our results with both cosmological and astrophysical parameters for completeness, we do not claim to be able to marginalize over both cosmological and astrophysical parameters when sampling.

IV. Results and Discussion

A. Diffusion and Discrete Data

We apply the diffusion model with the parameters discussed in the methodology section and present visual results in Figure 4 for one cosmology. We compare the galaxy count generated by the model with those of the hydro-simulation for a given dark matter map from an N-body simulation. We see that the score-based model is able to produce samples which has the same structure as the hydro-simulation. It is able to properly predict where there should be voids with no galaxies and find when it suddenly increases from a non-zero count. Furthermore, we see that standard deviation is higher in regions where there are many galaxies while for voids the deviation drops significantly. This is a reasonable behavior as the model should have more uncertainty on how many galaxies to inpaint on top of the dark matter simulation in high density regions, instead of in voids where we really expect the model to always predict zero or low galaxies with well constrained uncertainty. We can also observe that the patterns in the standard deviation are very similar to those in the dark matter simulation as this is the information used to create galaxy counts. However, in the actual samples the model is able to recover the relevant correlations which are consistent with the hydro-simulation without bringing in unexpected spatial correlations from the dark matter.

As can be observed, from the color scheme for the galaxy counts, the samples generated and the mean over the 100 samples all have a lower galaxy count than the hydro-simulation. This behavior was not just observed for this specific cosmology but for every 100 cosmology in our testing set. The score-based model is not able to predict high galaxy counts. Based on the various experience

of members in the group with score-based models, it is known that this has never been a problem for other applications. We pinpointed the cause for this behavior to be the fact that the galaxy counts are discrete integer values. As described in section II F score-based models use SDEs which are defined over continuous space. However, since discrete values are a subset of continuous values this framework should theoretically still be applicable. Within low dimensional toy problems where we define our own analytic data distribution it was indeed observed that we could generate discrete data distributions. However, when the score is learned by a neural network this becomes more challenging.

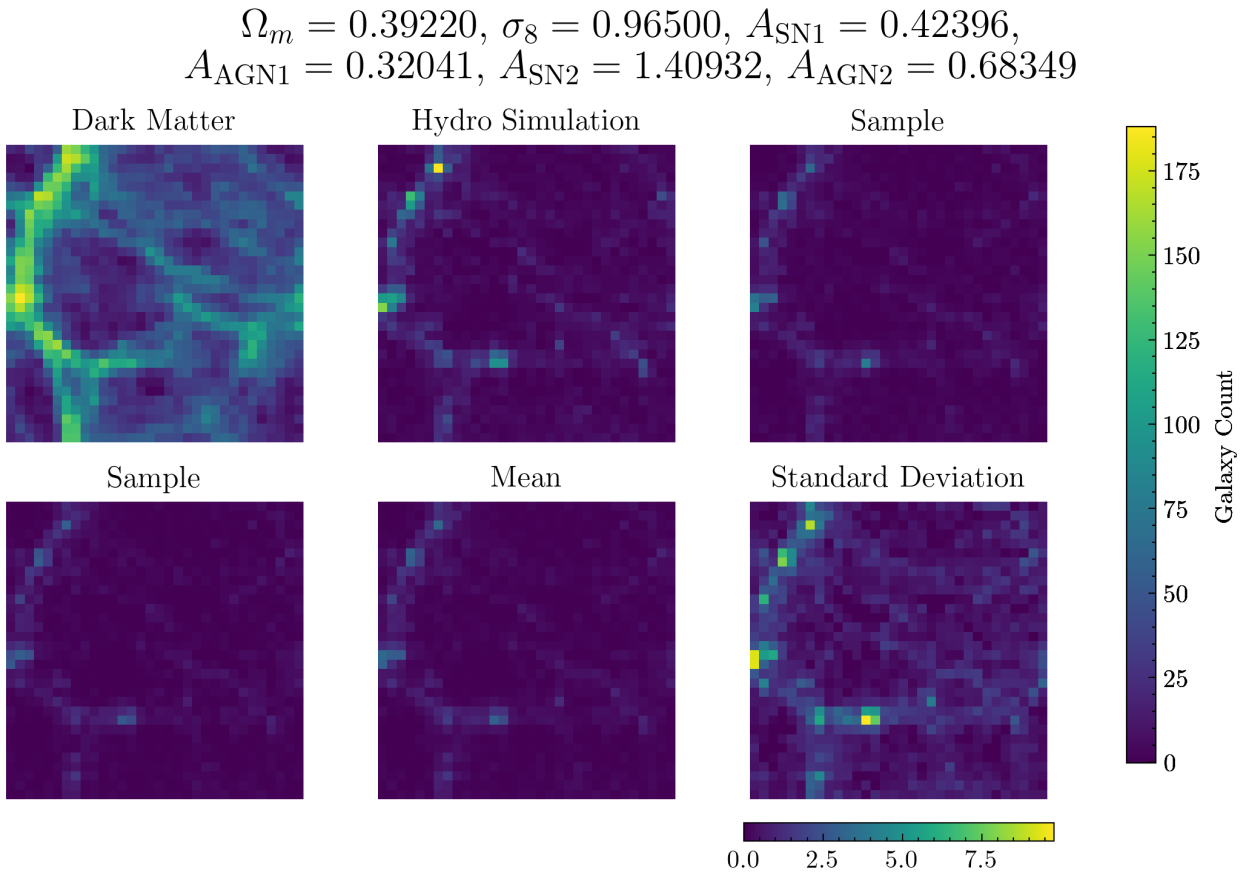

FIG. 4. The first panel in the top left corner shows the dark matter from the N-body simulation. The next panel is the galaxy counts from the associated hydro-simulation. We then present 2 samples out of the 100 which were generated by our model conditioned on the dark matter for this specific cosmology. Finally, we present the mean and standard deviation over our 100 samples. The cosmological and astrophysical parameters from the hydro-simulation are presented in the heading. We note that all of the plots are projected views of the 3D cubes summed along their third axis. The dark matter has its own color scheme (with no colorbar presented). The hydro-simulation, generated samples and their means all have the same color scheme set by the colorbar labeled Galaxy Count on the right of the figure. Finally, the standard deviation subplot has its own color scheme labeled below it.

In the diffusion process at $t = 0$ we have the underlying parent distribution. Now if we think in one dimensional space, a discrete probability distribution described in continuous space is a set of delta functions at integer values with the amplitude of each of the delta function describing the

probability of observing that integer value. Delta functions are problematic because their gradients are infinite. So predicting the score, the gradient of the log of the distribution, should lead to an infinite gradient which would cause errors in a computer. In reality, this does not happen because at $t = 0$ we actually have a σ_{min}. Meaning we have the delta functions convolved with a Gaussian distribution of variance σ_{min} from the solution of the SDE as explained in equation (14). Therefore, at $t = 0$ we have a very narrow Gaussian distribution with a high valued score but its not infinite. These high value scores can still cause instabilities if they are not learned properly. The reason being that the score guides the reverse diffusion, so if the network does not properly predict these high valued scores the sample will not be within the underlying parent distribution at $t = 0$. However, we found that pushing training for very long times (up to 4 days) allowed the network to see enough of these near zero times during training and be able to properly learn these high value scores without causing instabilities.

Unfortunately, this was only one side of the double edge sword. Another prominent issue for our specific application is that in most of the grid points in the cubes the galaxy count is zero. This is a consequence of how matter evolves in our universe. Matter clusters together with gravity in concentrated locations and most of the remaining space doesn't have galaxies. This is also the case in the simulations, the majority of the grid points in the 3D simulations have zero galaxies and there are tight groupings on small physical distance with high galaxy counts. No matter how big we made the architecture or how long we trained the score-model it was never able to capture these high counts happening on small scales.

Faced with these complications we also sought frameworks which apply these score-based models on discrete spaces. However, score-based diffusion with neural networks is rather novel with the first formal paper being published in 2021 [8]. After doing a literature review, the methodology for score-based models in discrete space is currently not well established. Furthermore, it currently cannot scale properly to a 3D simulation as we need. Therefore, we needed a different approach to fix our issue.

B. High Frequency Information

Inspired from the techniques used for Variational Diffusion Models (VDM) [23] we were able to modify our architecture, in a very simple manner, in order to better capture the high galaxy counts on small scales. The VDM paper argued that even though U-Nets learn different frequency information, the highest frequency information learned by the first feature maps is suppressed by low frequency information. This is because the U-Net has a lot more feature maps which contain lower frequency information compared to the first feature map. Furthermore, the latent space which is the starting point to create the vector field (the score) contains low frequency information. Even though, the skip connections bring in these higher frequency information they are not well captured by the network.

Hence, the VDM paper suggests adding inputs to the architecture, specifically a sin and cos projection at high frequencies of the data whose score we want to learn. This can be written out as $y_1 = \sin(2\pi f x)$ and $y_2 = \cos(2\pi f x)$ where x in our case would represent galaxy counts while y_1 and y_2 would be the sin and cos projections of our galaxy counts respectively at frequency f. We note that this sin and cos projection are just the components of a Fourier basis at frequency f. Hence, the idea established in the VDM paper is to add inputs that only have high frequency information. Meaning it propagates high frequency information to the different feature maps in the U-Net including the latent space.

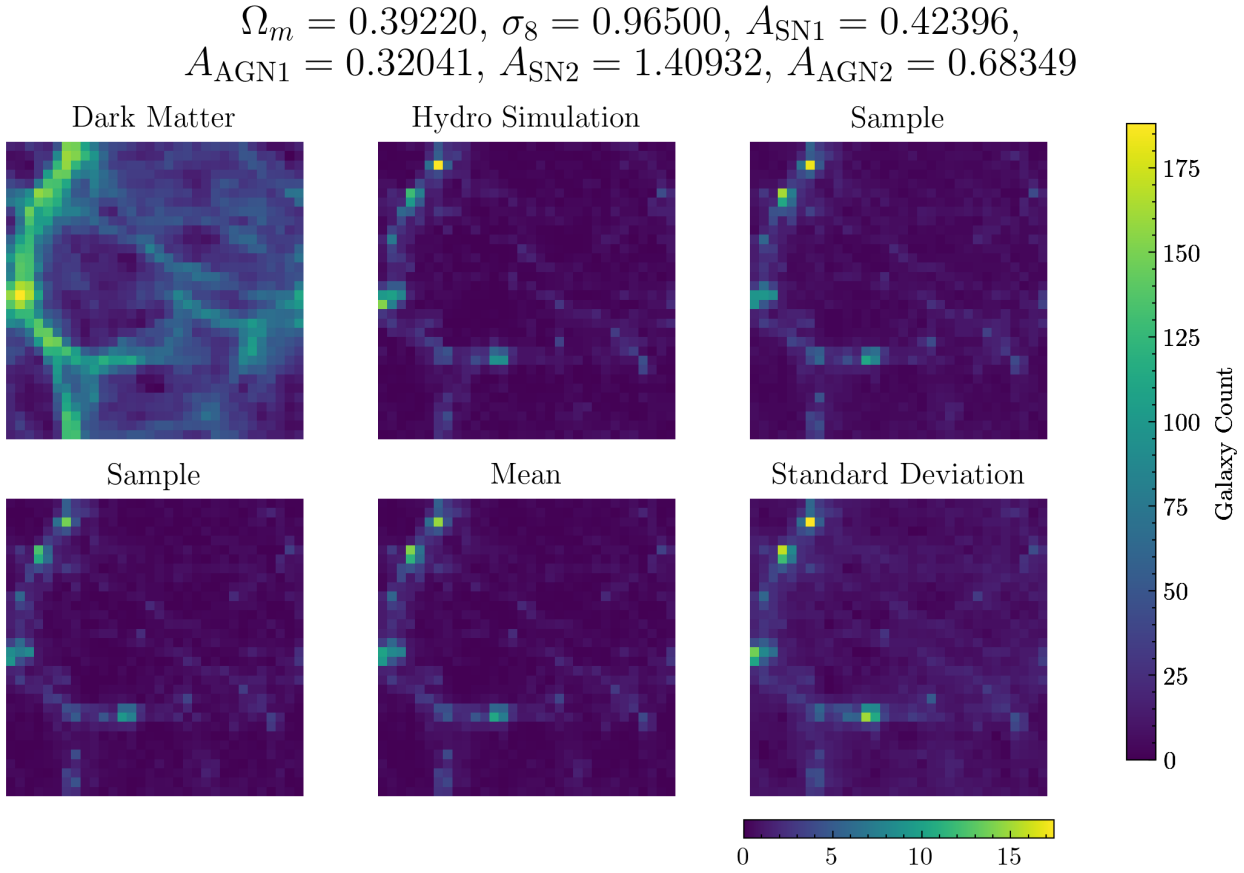

FIG. 5. We present 100 samples generated with the new architecture which encodes high frequency information. Our samples are conditioned on the same N-body simulation and therefore the same cosmological parameters as in Figure 4. We see that our samples now reach galaxy counts on the same order of magnitude as the hydro-simulation.

In our case this means that the reason we are unable to capture the high galaxy counts, is because the U-Net is predominantly learning the low frequency information in our simulations. As explained, the majority of galaxy counts are near zero. Hence, on the large scale (low frequency) the simulations on average have a very low galaxy count. To be able to resolve the high galaxy counts which happens on small scales in tight spatial groupings the network needs this improved way of learning features on small scales (high frequency). The VDM paper found that the optimal high frequencies for the projections were $f = 64, 128$ in units of 1/pixel.

We apply this sin and cosine projection at these two frequencies within our architecture and generate 100 samples for the same cosmology considered in Figure 4. Our new samples are displayed in Figure 5. We see that the samples are now able to predict galaxy counts on the same order of magnitude as the hydro-simulation, without losing any of the correlations which were present in the previous model as discussed in section IV A. Furthermore, we look at the power spectra of the samples and compare them to the power spectra of the hydro-simulation. The power spectra tells us if the number of galaxy counts on different scales from our samples are coherent with the hydro-simulation. We also look at the cross correlation which gives us information about if we predict galaxy counts at the same position as the hydro-simulation on different scales, hence, providing phase information. If the two are completely in phase then the cross correlation factor is one and zero means there is no positional correlation between the samples and the hydro-simulation.

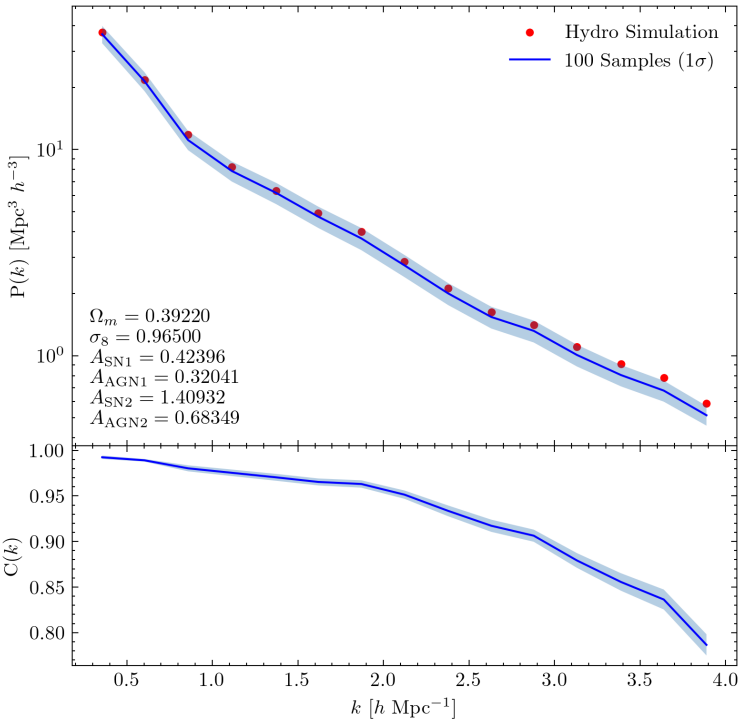

FIG. 6. The top panel shows the power spectra of the 100 samples generated with the new architecture compared to the power spectrum of the hydro-simulation. The bottom panel is the cross correlation between the samples and the hydro-simulation.

Figure 6 shows the power spectra and cross correlation between the 100 generated samples and the hydro-simulation for the cosmology being considered in Figure 5. We see that the power spectra of the generated galaxy counts from the N-body simulation for most scales are within one standard deviation of the power spectra of the hydro-simulation. We see that at about $k > 3$ the samples are within two standard deviation from the hydro-simulation. It is expected for the model to have difficulty perfectly reconstructing the galaxy counts at these small scales due to both the limitation of the U-Net's ability to learn at small scales and because the physics becomes highly non-linear which makes it harder to learn the correlation between dark matter and galaxy count. We see similar behavior in the cross correlation with a correlation between 0.9 and 1 up until $k < 3$, and then a progressive decrease. Although presented for one cosmology in Figure 6 we provide power spectra and cross correlations over different cosmologies in Appendix (VI) which demonstrate similar behavior.

We note that our model does not generalize yet to all 100 cosmologies in our testing set. We are currently investigating this, specifically we are trying to have a better understanding of why the specific frequencies f=64, 128 made such a big difference. We would prefer to have a physical reasoning for the choice of frequency, such as for example the one that corresponds to the Nyquist frequency. It is clear that this high frequency projection is key in making our model work, and we believe that developing a stronger physical intuition would allow us to make the model robust to any cosmology.

V. Conclusion

We have presented a new methodology which utilizes score-based diffusion models and machine learning to populate dark matter in N-body simulations with galaxy counts. Our model is probabilistic in nature and is currently able to generalize to a subset of cosmological parameters. Our generated galaxy counts are constrained to the same level of precision as a hydrodynamic simulations up to the wave number 4 h/Mpc. We highlight the fact that current HOD models do not have the ability to generalize across different cosmologies and achieve the same level of precision as our model on the small scales being considered. Our work has the potential of outperforming all models currently established in the literature.

The two current bottlenecks of the model are the discreteness of the galaxy counts and properly learning small scale information. We found that discrete data create large values in the score at times near zero which could cause samples to not be within the underlying parent distribution. Our best remedy for these are to train for many days in order to best learn the score. This however makes it expensive to run experiments on different models. Furthermore, we established that to learn high galaxy counts on small scales requires improving the way U-Nets learn high frequency information. To do so we do a sin and cosine projection of the galaxy counts at a set frequency. The frequencies selected in the VDM paper do not have any physical reasoning behind them and we wish to explore this further in the hopes of making our model generalize to all cosmologies.

Apart from providing a model which could improve upon current HODs, this work is also part of the collaboration Learning the Universe. If the model generalizes to all cosmologies it could be used on real data. The hope would be to use it to determine the cosmological parameters of our universe using galaxy distributions at present day. Cosmologists have been able to infer these parameters from the Cosmic Microwave Background and strong gravitational lenses. However,this could provide a novel framework for inferring cosmological parameters from galaxies and compare if we can better constrain these parameters.

[1] S. Bird, Y. Ni, T. Di Matteo, R. Croft, Y. Feng, and N. Chen, The ASTRID simulation: galaxy formation and reionization, Monthly Notices of the Royal Astronomical Society **512**, 3703 (2022), https://academic.oup.com/mnras/article-pdf/512/3/3703/43286842/stac648.pdf.

[2] R. Davé, D. Anglés-Alcázar, D. Narayanan, Q. Li, M. H. Rafieferantsoa, and S. Appleby, simba: Cosmological simulations with black hole growth and feedback, Monthly Notices of the Royal Astronomical Society **486**, 2827 (2019), https://academic.oup.com/mnras/article-pdf/486/2/2827/28524018/stz937.pdf.

[3] D. Nelson, V. Springel, A. Pillepich, V. Rodriguez-Gomez, P. Torrey, S. Genel, M. Vogelsberger, R. Pakmor, F. Marinacci, R. Weinberger, L. Kelley, M. Lovell, B. Diemer, and L. Hernquist, The illustristng simulations: public data release, Computational Astrophysics and Cosmology **6**, 2 (2019).

[4] Z. Zheng, A. L. Coil, and I. Zehavi, Galaxy evolution from halo occupation distribution modeling of deep2 and sdss galaxy clustering, The Astrophysical Journal **667**, 760 (2007).

[5] Z. Zheng, A. A. Berlind, D. H. Weinberg, A. J. Benson, C. M. Baugh, S. Cole, R. Davé, C. S. Frenk, N. Katz, and C. G. Lacey, Theoretical models of the halo occupation distribution: Separating central and satellite galaxies, The Astrophysical Journal **633**, 791 (2005).

[6] B. Hadzhiyska, S. Bose, D. Eisenstein, L. Hernquist, and D. N. Spergel, Limitations to the 'basic' HOD model and beyond, Monthly Notices of the Royal Astronomical Society **493**, 5506 (2020), https://academic.oup.com/mnras/article-pdf/493/4/5506/32967532/staa623.pdf.

[7] X. Zhang, Y. Wang, W. Zhang, Y. Sun, S. He, G. Contardo, F. Villaescusa-Navarro, and S. Ho, From dark matter to galaxies with convolutional networks (2019), arXiv:1902.05965 [astro-ph.CO].

[8] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations, in *International Conference on Learning Representations* (2021).

[9] R. Legin, M. Ho, P. Lemos, L. Perreault-Levasseur, S. Ho, Y. Hezaveh, and B. Wandelt, Posterior sampling of the initial conditions of the universe from non-linear large scale structures using score-based generative models (2023), arXiv:2304.03788 [astro-ph.CO].

[10] F. Villaescusa-Navarro, S. Genel, D. Anglé s-Alcázar, L. Thiele, R. Dave, D. Narayanan, A. Nicola, Y. Li, P. Villanueva-Domingo, B. Wandelt, D. N. Spergel, R. S. Somerville, J. M. Z. Matilla, F. G. Mohammad, S. Hassan, H. Shao, D. Wadekar, M. Eickenberg, K. W. K. Wong, G. Contardo, Y. Jo, E. Moser, E. T. Lau, L. F. M. P. Valle, L. A. Perez, D. Nagai, N. Battaglia, and M. Vogelsberger, The CAMELS multifield data set: Learning the universe's fundamental parameters with artificial intelligence, The Astrophysical Journal Supplement Series **259**, 61 (2022).

[11] S. Cole, C. G. Lacey, C. M. Baugh, and C. S. Frenk, Hierarchical galaxy formation, Monthly Notices of the Royal Astronomical Society **319**, 168 (2000), https://academic.oup.com/mnras/article-pdf/319/1/168/3734609/319-1-168.pdf.

[12] P. J. E. Peebles, *The Large-Scale Structure of the Universe* (Princeton University Press, Princeton, 1981).

[13] K. Hornik, M. Stinchcombe, and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks **3**, 551 (1990).

[14] C. Si, Z. Huang, Y. Jiang, and Z. Liu, Freeu: Free lunch in diffusion u-net (2023), arXiv:2309.11497 [cs.CV].

[15] N. Van Kampen, *Stochastic Processes in Physics and Chemistry*, North-Holland Personal Library (Elsevier Science, 1992).

[16] H. Pishro-Nik, *Introduction to Probability, Statistics, and Random Processes* (Kappa Research, LLC, 2014).

[17] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*, Institute of Mathematical Statistics Textbooks (Cambridge University Press, 2019).

[18] B. D. Anderson, Reverse-time diffusion equation models, Stochastic Processes and their Applications **12**, 313 (1982).

[19] Y. Song and S. Ermon, Improved techniques for training score-based generative models, in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 12438–12448.

[20] A. Hyvärinen, Estimation of non-normalized statistical models by score matching, Journal of Machine Learning Research **6**, 695 (2005).

[21] Y. Song, S. Garg, J. Shi, and S. Ermon, Sliced score matching: A scalable approach to density and score estimation, in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, Proceedings of Machine Learning Research, Vol. 115, edited by R. P. Adams and V. Gogate (PMLR, 2020) pp. 574–584.

[22] P. Vincent, A Connection Between Score Matching and Denoising Autoencoders, Neural Computation **23**, 1661 (2011), https://direct.mit.edu/neco/article-pdf/23/7/1661/851298/neco_a_00142.pdf.

[23] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, Variational diffusion models (2023), arXiv:2107.00630 [cs.LG].

VI. Appendix

We provide further power spectra and cross correlation plots between our samples and the hydro-simulations over different cosmologies as supporting material for our ability to generalize on a subset of cosmologies.