University of Nottingham

Department of Physics and Astronomy

MSc. Machine Learning in Sciences

Graduation thesis
Submitted September 2023, in partial fulfillment of the conditions for the award of the degree

2D and 3D diffusion to simulate the universe

Scientific supervisor: **Dr. Adam Moss**

Rajat Goyal, September 2023

Contents

Acknowledgements

First and foremost, I express my deepest gratitude to my supervisor, Dr. Adam Moss, whose guidance, patience, and unwavering support have been instrumental in shaping this dissertation. Your keen insights, feedback, and encouragement throughout this journey have been invaluable.

I am indebted to the Physics Department's faculty and staff for creating an environment conducive to learning and exploration. Their collective wisdom and dedication to academic excellence have provided me with a solid foundation on which to build my research.

Special thanks go to my team members Charlie Davies and Davlesh Jodhun, for the countless discussions, brainstorming sessions, and obstacles we have been through that kept me motivated and diligent. Their companionship throughout this whole degree and not just this dissertation is what kept me going through thick and thin. I can't imagine doing this without their friendship and support.

I would also like to extend my gratitude to Anshuman Singh, Samridh Aggarwal and Sarthak Siddha for their unwavering belief in my capabilities, even when I doubted them, played a pivotal role in reaching this milestone. I am especially thankful to my parents, for their endless love, sacrifices, and for instilling in me the values of hard work and persistence.

In the grand scheme of this academic journey, each one of you has played a significant part, and for that, I am eternally grateful.

Thank you.

Abstract

In our dissertation, we explored the potential of machine learning, focusing on 2D and 3D diffusion models, as a tool to simulate the vast expanse of n-bodies in the universe. Traditional simulation methods, while effective, are computationally intensive. Our aim was to find a more efficient yet accurate alternative. The results were promising: our machine learning models replicated the power spectrum of the original simulations and in a fraction of the usual computational time but there is still room for improvement. This achievement underscores the potential of machine learning in cosmological research. By delving into advanced diffusion methods, such as latent and point voxel diffusion, we gained deeper insights into the complexities of cosmic structures. These innovative methods present a groundbreaking approach to understanding the universe's evolution and matter behaviour. While our findings mark a significant step forward, there's ample scope for further research, especially with the advent of more sophisticated models, improved hardware, and novel techniques. In essence, our research has laid a robust foundation, highlighting the transformative role machine learning can play in universe simulations, while also being one of the first model working directly on the coordinates of particles in n-body simulations instead of voxelizing it first.

1 Introduction

Unveiling the universe's secrets is a monumental task in science. At the heart of this challenge is cosmology, which delves into the universe's vastness, its beginnings, its makeup, and its future path. The universe is made up of countless particles. Basic particles like protons and neutrons form the matter we're familiar with. There's also "dark matter," an unseen substance that influences the universe, and dark energy, a mysterious force that speeds up the universe's growth.

Traditionally, computer simulations have been pivotal in our quest to understand these phenomena. They predict the universe's behaviour based on established physical laws like gravity, hydrodynamics, and cosmological expansion equations. However, they often suffer from computational challenges due to the vast number of particles and interactions involved, leading to substantial time and resource requirements.

Recent advancements in machine learning algorithms have revolutionized the field of cosmological simulations. ML is a field of artificial intelligence that specializes in recognizing patterns and making predictions based on data. When applied to universe simulations, ML drastically changes the game. Instead of starting from scratch and making exhaustive calculations every time, ML models can be trained on existing datasets. Once trained, they can then predict certain outcomes in the universe much faster than traditional simulations. This makes research not only more efficient but also broadens the scope of what can be explored in a given time. The ability to "learn" from data means that each successive simulation can be more accurate, making the entire process iterative and continually refined.

A key concept here is diffusion, which describes how things spread. Imagine adding food colouring to water; its dispersal illustrates diffusion. "Latent" and "point voxel" diffusion take this a step further. Latent diffusion [15] looks at the underlying patterns of spread, while point voxel diffusion [36] examines the spread at a detailed level, focusing on specific points or "voxels." This approach lets scientists depict cosmic structures in great detail, leading to more refined universe simulations.

This dissertation aims to merge machine learning with advanced diffusion techniques, offering a sharper tool to probe the universe's complexities.

1.1 Aim and motivation

In cosmology, density fields help us understand how matter is spread across the universe. These fields can be categorized into two types: Gaussian and non-Gaussian. A Gaussian density field means that the spread of disturbances follows a specific pattern known as the Gaussian distribution. On the other hand, non-Gaussian fields don't follow this pattern. For Gaussian fields, we use tools like the power spectrum to describe them. However, for non-Gaussian fields, which are more commonly observed, we don't have a standard tool. [32]

A primary goal of studying these fields is to gain a deeper understanding of the universe's fundamental physics, its origins, and its future. One challenge is determining the best statistical method to get accurate values for certain cosmological parameters. [32] Currently, the power spectrum is the main tool we use for this. However, research has shown that there's more information available, especially at smaller scales, which the power spectrum might miss. By using neural networks, which are trained to sift through all potential statistical methods, we can extract this additional information. These methods have shown that we can get even more precise values for cosmological parameters than just using the power spectrum. But, these simulations are resource-intensive, often taking a very long time to run. [32]

This dissertation aims to address this challenge. We propose using diffusion models and machine learning (ML) algorithms to create 2D and 3D simulations of the universe. If these new models are accurate, it indicates that our approach can provide deeper insights into the universe's makeup, helping advance our understanding of its fundamental physics while also saving a lot of time.

2 Background

2.1 Cosmology and the Universe

Cosmology is the scientific exploration of the universe's large-scale properties, including its beginnings, development, and ultimate destiny. It examines the complex relationships between matter, energy, space, and time, offering a holistic view of the cosmos from an astrophysical perspective.

2.1.1 Big Bang Theory:

At the heart of cosmology is the Big Bang Theory. This theory suggests that the universe originated from a single point of immense density around 13.8 billion years ago [13]. Evidence for this theory includes the redshift seen in distant galaxies, showing they are receding from us, and the Cosmic Microwave Background (CMB) radiation, which is the leftover warmth from the Big Bang.

2.1.2 Dark Matter:

Dark matter is a mysterious component of the universe. It doesn't emit, absorb, or reflect light because it doesn't interact with electromagnetic forces. However, it does exert gravitational forces and is thought to make up about 27% of the universe [2].

Its presence is inferred mainly from its gravitational influence on galaxies and their clusters. One key piece of evidence is the way stars in galaxies rotate. Observations show that stars, especially those further from a galaxy's center, move at speeds suggesting the galaxy has more mass than just the visible stars and gas. This unseen mass is believed to be dark matter [24].

Additionally, gravitational lensing provides more evidence. This phenomenon occurs when the gravitational pull from a massive object, like a galaxy cluster, bends and magnifies light from objects behind it. The observed lensing effects indicate that these clusters have more mass than just the visible matter, further pointing to the presence of dark matter [29].

2.1.3 Dark Energy:

Dark energy is a perplexing component of the universe, acting in contrast to dark matter by driving the universe's accelerated expansion. It's thought to constitute approximately 68% of the universe [2].

The existence of dark energy was postulated based on observations of faraway supernovae. In the 1990s, two separate groups of astronomers found that, contrary to earlier beliefs, the universe is expanding at an increasing rate. This unexpected acceleration couldn't be accounted for by just matter, leading to the introduction of the dark energy concept [22].

The Lambda Cold Dark Matter (ΛCDM) model is the prevailing cosmological model that encompasses both dark energy and cold dark matter. In this model, Λ stands for the cosmological constant. This constant represents a type of energy spread evenly throughout space and is linked to dark energy [20].

Dark matter and dark energy, though invisible and mysterious, play pivotal roles in shaping the universe. Their effects, from the rotation of galaxies to the accelerated expansion of the cosmos, have profound implications for our understanding of the universe's structure and fate.

Cosmology, with its intricate theories and observations, offers profound insights into the universe's workings. The continuous advancements in this field, backed by real-world research and studies, provide a deeper understanding of the cosmos, from the vast expanse of galaxies to the enigmatic dark matter and dark energy.

2.2 Simulations of the Universe

Cosmological simulations are computer-based models designed to mimic the universe's progression from its beginnings to now. These models are essential in studying how cosmic entities, like galaxies and their clusters, as well as the vast empty spaces between them, come into being. Such simulations offer a digital platform to compare theoretical concepts of cosmic development with actual observation [26].

2.2.1 Cosmological Particles in Simulations:

In these simulations, the universe is often depicted using a large collection of particles. These particles don't represent typical elements like protons or electrons but stand for aggregates of matter. The primary particles used in these simulations include:

1. **Baryonic Particles:** These stand for regular matter, such as stars, galaxies, and gas formations. They contribute to the creation of observable cosmic patterns.

2. **Dark Matter Particles:** These symbolize the elusive dark matter. Influenced by gravity, they group together and serve as the foundational framework for cosmic formations [8].

3. **Cosmic Radiation Particles:** These depict different radiation types, including the Cosmic Microwave Background, offering insights into the universe's state shortly after its inception.

2.2.2 Physical Concepts used in simulations:

Several fundamental physical concepts govern these simulations:

1. **Gravity:** This force between particles is explained by Newton's gravitational law and, for vast cosmic scales, by Einstein's theory of relativity [9].

2. **Hydrodynamics:** This principle dictates how gases move and interact in space, based on the Navier-Stokes equations [28].

3. **Thermodynamics:** This oversees how matter heats up or cools down in the universe.

4. **Radiative Transfer:** This concept illustrates the journey of radiation across the universe and its impact on the matter it meets [19].

The simulations use the Friedmann equations to detail how the universe expands and include the cosmological constant to explain the faster expansion caused by dark energy. [10].

2.2.3 Mathematical Foundations:

Cosmological simulations are anchored by a set of core equations that depict the dynamics of matter and energy in the universe. Below, we explore the primary mathematical foundations driving these simulations:

1. **Newton's Law of Universal Gravitation:**

   $$F = \frac{Gm_1m_2}{r^2}$$

   This equation describes the gravitational force F between two masses m_1 and m_2 separated by a distance r. G is the gravitational constant. In cosmological simulations, this equation governs the gravitational interactions between particles, ensuring that structures like galaxies and clusters form correctly [26].

2. **Friedmann Equations:**

   $$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho - \frac{k}{a^2}$$

   The Friedmann equations describe how the scale factor a of the universe evolves over time. $\dot{a}$ is the rate of change of the scale factor, G is the gravitational constant, ρ is the density of the universe, and k is the curvature of space. These equations are central to understanding the expansion of the universe [10].

3. **Navier-Stokes Equations:**

   $$\rho\left(\frac{\partial v}{\partial t} + v \cdot \nabla v\right) = -\nabla p + \mu\nabla^2 v + \rho g$$

   These equations describe the motion of viscous fluid substances. In the context of cosmological simulations, they govern the behavior of gases. ρ is the fluid density, v is the velocity field, p is the pressure, μ is the dynamic viscosity, and g is the gravitational acceleration [28].

4. **Equation for Radiative Transfer:**

   $$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + n \cdot \nabla I_\nu = j_\nu - \alpha_\nu I_\nu$$

   This equation describes how radiation I_ν at frequency ν travels and interacts with matter. j_ν is the emission coefficient, α_ν is the absorption coefficient, n is the direction of propagation, and c is the speed of light. It's crucial for understanding phenomena like the Cosmic Microwave Background [11].

Cosmological simulations rely on foundational equations to effectively model the universe's progression. Through sophisticated computational methods, these equations allow scientists to simulate cosmic phenomena, evaluate theories, and deepen our understanding of the universe's intricacies.

2.2.4 Significance of Simulations:

Cosmological simulations are invaluable for several reasons:

- They enable scientists to simulate and examine otherwise unobservable cosmic events.
- They shed light on the characteristics of dark matter and dark energy.
- They enhance our knowledge of the universe's early states.
- They produce forecasts that, when compared to actual observations, deepen our grasp of cosmology.

Cosmological simulations, grounded in basic physics, give us a glimpse into the universe's progression. Through modeling essential cosmic particles and their interactions, these simulations help us understand how cosmic structures form and function, revealing the universe's enigmas. These simulations act as a bridge between theoretical predictions and observational data, offering profound insights into the universe's structure and evolution.

2.3 N-body simulations:

This dissertation focuses on utilizing N-body simulations to study the universe. The term "N-body" denotes a system with N interacting entities, which, in astrophysics, could be stars, galaxies, or even dark matter particles. Primarily, these entities interact through gravitational forces, although other forces can be considered [1]. N-body simulations numerically track the movement of these entities, mainly under gravitational influence. In such simulations, each entity interacts with all others, resulting in a total of $\frac{N(N-1)}{2}$ interactions.

The intricacy of N-body simulations largely stems from the non-linear gravitational interactions among dark matter particles. Even though dark matter is elusive and only detectable through its gravitational impact, it's crucial in determining the universe's large-scale structure. Therefore, accurately simulating dark matter's distribution and behaviour is vital for producing authentic representations of the universe's development [5].

2.3.1 Mathematical Formulation:

The gravitational force F_{ij} between two particles i and j is described by Newton's law of gravitation:

$$F_{ij} = \frac{Gm_i m_j (r_j - r_i)}{|r_j - r_i|^3}$$

[7]

Where:

- G is the gravitational constant.
- m_i and m_j are the masses of bodies i and j respectively.
- r_i and r_j are their respective position vectors.

The net force on body i is the sum of forces due to all other bodies. Using Newton's second law, $F = ma$, one can derive the equations of motion for each body.

2.3.2 Gadget software for N-body simulations:

Gadget is a well-known software utilized for N-body simulations. Notable projects, including CAMELS [32] and Quijote [33], have employed it. However, N-body simulations come with their set of challenges, such as:

Computational Demand: The pairwise interactions scale with O(N^2). As the number of bodies (N) increases, direct computation becomes increasingly challenging [16].

Sensitivity: The system's behaviour can be highly influenced by its starting conditions. For instance, in scenarios like the three-body problem, slight changes can lead to unpredictable outcomes [27].

No General Analytical Solution: While a two-body system has a straightforward analytical solution, systems with three or more bodies don't have a general closed-form solution, making them more complex to analyze [4].

2.3.3 Efficiency methods implemented by GADGET:

Time-stepping methods: Time-stepping techniques are employed to progress the system in distinct time segments. These techniques adjust the positions and speeds of particles based on the forces they experience. By understanding the forces on each particle (often calculated using approaches like the Barnes-Hut algorithm), these methods transition the system's state over time. This is advantageous as it permits areas with significant dynamic activity, such as dense galaxy cores, to be processed with shorter time intervals for precision. Conversely, less active regions can utilize longer intervals, enhancing computational efficiency. Some simulations, like those utilizing GADGET, adopt adaptive time-stepping, allowing particles to have varied time intervals based on their specific dynamic situations. [25]

TreePM algorithms: The TreePM algorithm offers an efficient approach to estimate gravitational forces in N-body simulations. It employs the Tree code for nearby forces and leverages the Barnes-Hut algorithm to streamline computations. Instead of calculating gravitational interactions between each particle pair, which would require O(N^2) complexity, the Barnes-Hut method reduces it to O(NlogN) by clustering distant particles and estimating their collective gravitational impact. Additionally, the algorithm integrates the particle-mesh method for distant forces, determining the gravitational potential on a grid via Fourier methods. This combined strategy is especially effective for expansive cosmological simulations. [3]

In essence, the TreePM algorithm focuses on the effective calculation of gravitational forces, whereas time-stepping methods dictate the system's temporal evolution based on these forces and optimize the duration of these steps for efficiency. Both elements are vital in numerous N-body simulations, addressing distinct facets of the simulation procedure.

In 2005, the Millennium Simulation stood out as a significant N-body simulation in cosmology. It modeled the universe's progression from just post-Big Bang to today within a cubic space. Using over 10 billion particles to symbolize dark matter, it effectively illustrated the development of galaxies, galaxy clusters, and the expansive gaps in the cosmic fabric. [25].

Despite these optimisations, the process of these simulations is quite taxing computationally. Hence recently the utilisation of generative models is the trending approach in the research community as it substantially reduces the computation cost and time needed by learning the trends between the positions of these particles and inferencing statistical relations between them instead of solving complex pairwise physical equations to calculate the acceleration and position of dark matter particles in n-body simulations. In this dissertation the generative model called latent diffusion is demonstrated for this purpose. As the name suggests, it makes use of the latent space to sample the distribution and through various time steps transforms it into complex structures like image generation. Latent space is lower dimensional representation of the input features hence making the computation and training time much faster. Diffusion models are also more interpretable compared to other generative models as we can derive information of each time step to understand the working of the process under the hood of the model.

3 Literature Review

As the objective of this dissertation is to generate non-distinguishable copies of the simulation maps of the universe using diffusion, the need for the dataset of simulation of the universe arises. Two research projects **CAMELS** [32] AND **QUIJOTE** [33] obtained pretty decent n-body simulations that are compatible with the task.

The **CAMELS dataset** [32] comprises a vast number of simulations, encompassing a wide range of cosmological and astrophysical models. These simulations collectively encompass over 100 billion entities, including dark matter particles, gas elements, stars, and black holes. The aggregate volume of these simulations exceeds $(400h^{-1}Mpc)^3$. Nevertheless, it is important to acknowledge that the occurrence of exceptional entities, such as substantial voids or gigantic clusters, is fundamentally constrained within the collective volume due to the fact that each simulation only encompasses a comoving volume of $(25h^1Mpc)^3$. [32]

The primary objective of the CAMELS project is to offer theoretical forecasts for a certain observable or field, contingent upon the variables of cosmology and astrophysics.

$$S(z) = f(\vec{\theta}_c, \vec{\theta}_a, z)$$

The function $S(z)$ is defined as the dependence of a generic statistic (such as a power spectrum) on the redshift z, as well as the cosmological parameters denoted by $\vec{\theta}_c$ and the astrophysical parameters denoted by $\vec{\theta}_a$. The aforementioned equation is applicable to a summary statistic, but it can be extended to encompass 2D or 3D density fields denoted as $F(\vec{x})$:

$$F(\vec{x}, z) = g(\vec{\theta}_c, \vec{\theta}_a, \delta(\vec{x}), \delta(\vec{y}))$$

where $\delta(\vec{x})$ and $\delta(\vec{y})$ represent the initial circumstances at position $\vec{x}$ and its surrounding environment $\vec{y}$ $(\vec{y} \neq \vec{x})$. [32]

The CAMELS suite comprises a total of 4,233 numerical simulations. Roughly 50% of the simulations in question are classified as N-body, with the remaining portion being characterised as state-of-the-art (magneto-)hydrodynamic simulations that incorporate the subgrid physics models proposed by IllustrisTNG and SIMBA. Starting from the initial redshift $z = 127$ up until the present day, the simulation tracks the progression of 256^3 dark matter particles with a mass of $6.49 \times 10^7 \frac{(\Omega_m - \Omega_b)}{0.251} h^{-1} M$. These particles are contained within a periodic box with a comoving volume equivalent to $(25h^{-1}\text{Mpc})^3$. The selection of the volume and particle number in this study represents a compromise between two factors. Firstly, it aims to ensure a sufficient number of simulations to adequately explore the parameter space, particularly for machine learning applications where at least 1,000 variations are desired. Secondly, it takes into account the computational expense associated with larger volume simulations that are necessary to accurately model intricate processes involved in galaxy formation. [32]

Snapshots are collected at various redshifts, including $z = 6, 5, 4, 3.5$, as well as an additional 30 redshifts that are logarithmically distributed between $(1 + z) = 4$ and $(1 + z) = 1$. The variable z denotes the redshift, which defines the extent to which the light emitted by an entity has been stretched due to the expansion of the universe. A greater redshift value corresponds to a greater spatial distance and an older time period inside the universe, while a redshift value of zero signifies the present time. The cosmological parameters in all simulations are held constant at the following values: $\Omega_b = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $M_v = 0.0$ eV, $w = -1$, $\Omega_K = 0$. [32]

where:

- Ω_b: It corresponds to the baryon density parameter which is the proportion of the overall energy density of the universe that consists of ordinary matter.
- h: It is the Hubble parameter, which is dimensionless. It is related to the Hubble constant, which is a parameter that defines the rate at which the universe is expanding.
- n_s: It is the spectral index shown for tiny fluctuations in the density of the universe right after the big bang (primordial scalar perturbations). It explains the primary state of the universe and the manner in which density fluctuations exhibit variation across different scales. A score of 0.9624 suggests a substantially scale-invariant spectrum.
- M_v: This value shows the mass of the neutrino. Neutrinos are elementary particles characterised by their negligible mass. In the current context, the assigned value of 0.0 eV is indicative of the assumption made in these simulations that neutrinos had no mass.
- w: It is the parameter associated with the state of dark energy. The value of -1 is associated with a cosmological constant, which represents the most basic concept of dark energy.
- Ω_K: It is the parameter representing the density of curvature. A numerical value of zero denotes a state of a flat universe, signifying that the spatial geometry corresponds to the principles of Euclidean geometry.

The parameters Ω_m and σ_8 are subject to variation in different simulations. A broad range of parameters has been employed in this study to minimise any biases on the neural network's output: $\Omega_m \in [0.1, 0.5]$, $\sigma_8 \in [0.6, 1.0]$.
Where:

- Ω_m: It refers to the matter density characteristic. It denotes the proportion of the overall energy density of the universe that consists of matter, including both ordinary matter and dark matter. If Ω_m exceeds 1, it indicates a closed universe that might ultimately contract. Conversely, if Ω_m is below 1, the universe is open and will keep expanding indefinitely. A value of Ω_m precisely at 1 suggests a flat universe that will continue its expansion, albeit at a diminishing pace. Present-day observations lean towards our universe being nearly flat.
- σ_8: This measure defines the magnitude of variations in matter density at scales of $8h^{-1}$ Mpc. The clumpiness of the large-scale structure in the universe can be determined by this measure. A higher σ_8 means more clustering, implying that the universe is lumpier on large scales, while a lower σ_8 means less clustering.

In the N-body simulations, the only variables that undergo alteration are these two parameters, in addition to the initial random seed value which governs the characteristics of the Gaussian density field at the outset. In studies of the universe using computer simulations, researchers often change a few key settings to see how they affect the results. Two of these settings, Ω_m and σ_8, are especially important for understanding the large patterns we see in the universe. By changing these settings, scientists can explore different possible versions of the universe and see how these changes might influence the development of galaxies and other structures. At the same time it's simpler for researchers to keep most settings the same and only change a few at a time. If they changed too many settings all at once, it would be hard to tell which one caused a particular result. By focusing on just a few key settings, they can get clearer answers. IllustrisTNG, a type of simulation done in CAMELS research, solves the coupled equations of gravity using an N-body tree particle mesh approach. [32]

The CAMEL simulations have certain limitations. Firstly, due to the mass and spatial resolution of CAMELS, it's not possible to examine scales smaller than approximately $1h^{-1}$kpc. Additionally, only halos with a dark matter mass exceeding $6.5 \times 10^{9}(\Omega_m - \Omega_b)/0.251h^{-1}M$ have a minimum of 100 dark matter particles. This means that CAMELS isn't suitable for studies that focus on the distribution of matter at very tiny scales, like the sub-halos in the Milky Way, which could provide insights into the nature of dark matter. Secondly, the simulation's volume is quite limited, being just $(25h^{-1}\text{Mpc})^3$. This means CAMELS doesn't account for long-wavelength modes, which play a crucial role in the formation of large structures like galaxy clusters and in determining the correct normalization of the matter power spectrum across all scales. [32]

Another research paper investigated is the **Quijote Simulations** by Villaescusa-Navarro et al. [33]. The Quijote simulations were developed to offer the scientific community a substantial dataset of cosmological simulations. In this research, we unveil the Quijote suite, marking a significant milestone as the most comprehensive set of full N-body simulations considering its specific mass and spatial detail. The suite boasts 44,100 simulations, covering over 7,000 different cosmological scenarios. At a singular redshift level, these simulations account for more than 8.5 trillion particles. The computational effort for these simulations surpassed 35 million CPU hours, generating in excess of 1 Petabyte of data. It's important to highlight that our simulations operate at a comparatively lower resolution, identifying halos with masses roughly above $3 \times 10^{12}h^{-1}M$ (for high resolution) or around $2 \times 10^{13}h^{-1}M$ (for the standard resolution). Higher resolution simulations for the Quijote suite were not feasible due to resource constraints. Our choice of this resolution was influenced by two primary factors: the need to run an extensive array of simulations for a precise Fisher matrix evaluation, and our strategy to initially explore a broad cosmological expanse, followed by the application of machine learning to refine the resolution of these simulations. [33]

The Quijote suite consists entirely of N-body simulations. These were executed using the TreePM software Gadget-III, which is an enhanced version of Gadget-II [25]. The starting conditions for every simulation are set at $z = 127$. To derive the initial matter power spectrum and transfer functions, we adjusted the $z = 0$ matter power spectrum and transfer functions sourced from CAMB [17]. For models that include neutrinos with significant mass, the method from [35] is utilized for rescaling. However, for models where neutrinos are considered to have no mass, a more standard rescaling approach is used. This approach is represented by the equation:

$$Pm(k, zi) = \left(\frac{D(z)}{D(zi)}\right)^2 Pm(k, z = 0),$$

Where $D(z)$ denotes the growth factor at a specific redshift z, f represents the growth rate, and the value of γ is approximately 0.6 in the context of the ΛCDM model. The Lambda Cold Dark Matter (ΛCDM) model, commonly known as the standard cosmological model, provides insights into the large-scale structure and progression of the Universe. In this framework:

- Lambda (Λ) symbolizes the cosmological constant linked to dark energy, which is thought to drive the Universe's accelerating expansion.
- Cold Dark Matter (CDM) pertains to a theoretical type of dark matter that has slow movement relative to light's speed. According to the ΛCDM model, the Universe's composition is roughly 5% regular matter (like stars, planets, and humans), 27% dark matter, and 68% dark energy. This model aligns with various astronomical data, including observations of

the cosmic microwave background, patterns of galaxy clusters, and observations of distant supernovae.

All the simulations cover a cosmic space of $1(h^{-1}Gpc)^3$. Most of these simulations track the development of 512^3 CDM particles (and an additional 512^3 for those with massive neutrinos), which is our standard resolution. However, there are also simulations with 256^3 (lower resolution) and 1024^3 (higher resolution) CDM particles. The gravitational softening distance is determined as 1/40 of the average distance between particles, translating to $100, 50$, and $25h^{-1}kpc$ for low, standard, and high-resolution simulations, respectively. This gravitational adjustment is consistent for both CDM and neutrino particles. Data is captured at redshifts of $0, 0.5, 1, 2$, and 3. Additionally, we preserve the initial conditions and the methods to recreate them.

For our reference model, the cosmological parameters are set as follows: $\Omega_m = 0.3175$, $\Omega_b = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $\sigma_8 = 0.834$, $M_\nu = 0.0$eV, and $w = -1$. These values align well with the most recent findings from the Planck Collaboration (as of 2018). (the meaning of these values are explained in the section above where talking about CAMELS project).

After discussing the detailed datasets from the CAMELS and Quijote projects, we've set the stage for our main research focus. Now, we'll shift our attention to the heart of this dissertation: diffusion. In the next sections, we'll dive deeper into what diffusion is, and how it's relevant to our study. There are two significant types of latent diffusion: ddpm and ddim. DDPM refers to Denoising Diffusion Probabilistic Models. It is a type of generative model that simulates a diffusion process to transform data into noise gradually and then reverses this process to generate new data samples. It can be summarised as a forward and reverse Markov chain process. A Markov chain is a sequence of random variables where the probability distribution of each variable depends only on the value of the previous variable. In the context of DDPM, the data at each timestep is a random variable, and the transition from one timestep to the next is governed by a Markov chain. [6]

Forward process (diffusion):

In the **Denoising Diffusion Probabilistic Models (DDPM)** by Ho et al. [15], the forward process is a pivotal mechanism that incrementally introduces noise to the data over a series of steps. One can imagine the forward process as a spreading or diffusion activity. As steps progress, the data undergoes more corruption, deviating further from its original state. The data transforms into a state resembling pure noise. This systematic corruption of data sets the stage for its counterpart, the reverse process, which seeks to restore the original data from its noisy state.

The equation for the forward process is represented by:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) := \mathcal{N}(x_t; (1-\beta_t)x_{t-1}, \beta_t I)$$

Here:

- $q(x_{1:T}|x_0)$ is the forward process, which is a Markov chain that describes how the data x evolves over T time steps given the initial state.
- $\mathcal{N}$ represents the Gaussian (Normal) distribution.
- β_t is the variance or noise schedule, which dictates how much noise is added at each time step t.

- I is the identity matrix.

The forward process is essentially a sequence of Gaussian distributions where the mean and variance are determined by β_t and the previous state x_{t-1}.

Noise schedule dictates the intensity of noise introduced at every step. Commonly, initial values of β_t are set low and gradually increase. This progressive addition of noise ensures a systematic corruption of the data, enabling the model to master the task of data denoising at different corruption levels.

There are three types of noise schedule used typically:

1. **Linear:** The linear noise schedule is characterized by a steady, linear increase (or decrease) in the amount of noise added to the data at each timestep.

$$\beta_t = \beta_{\text{start}} + t \times \text{rate}$$

   Where:

   - β_{start} is the initial noise level at $t = 0$.
   - rate is the rate of increase (or decrease) of noise per timestep.

2. **Cosine:** The cosine function, known for its wavy pattern, when tailored for a noise schedule, provides a gentle transition between its start and end points.

$$\beta_t = \frac{1 - \cos(\pi t/T)}{2}$$

   Where:

   - T represents the total timesteps.

3. **Sigmoid:** The sigmoid function, characterized by its S-shaped curve, can transform any real number into a value between 0 and 1.

$$\beta_t = \frac{1}{1 + e^{-k(t-t_0)}}$$

   Here:

   - k is the slope parameter.
   - t_0 marks the midpoint of the curve.

A significant property of the forward process is its ability to sample x_t at any timestep t in a closed form. This is represented by:

$$q(x_t|x_0) = \mathcal{N}(x_t; \alpha_t \bar{x}_0, (1 - \alpha_{\bar{t}})I)$$

Where:

$$\alpha_t := 1 - \beta_t$$

$$\alpha_{\bar{t}} := \prod_{s=1}^{t} \alpha_s$$

It's a powerful feature because it enables the model to generate new data points based on the original data x_0 and the noise added up to that point.

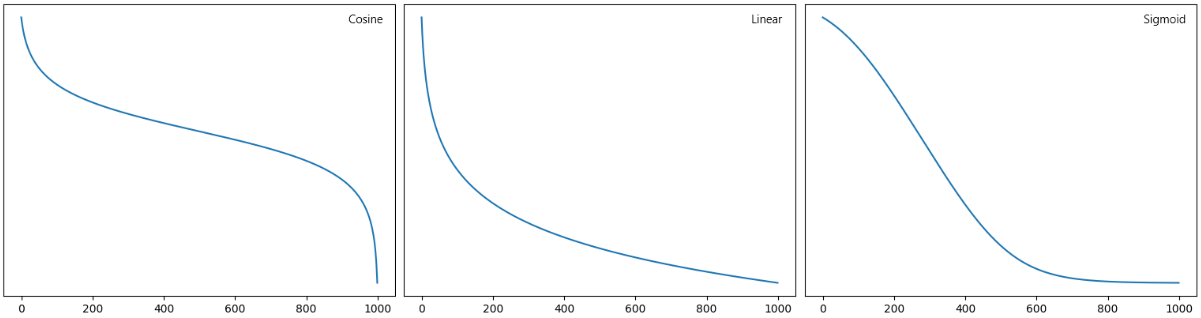


Figure 1: *Visual representation of various noise schedules*

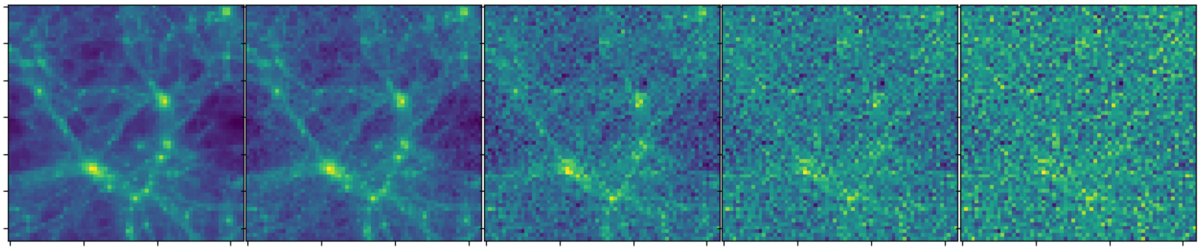


Figure 2: *Example of how noise schedule corrupts an image with noise in various time steps*

Reverse Process (denoising):

The reverse process is essentially the inverse of the forward process. The model attempts to iteratively denoise and reconstruct the original data or generate new samples that resemble the original data distribution. The reverse process is modeled as a series of conditional distributions, each representing the distribution of the data at a previous timestep given the data at the current timestep. The heart of the reverse process is the denoising function, typically represented by a neural network. This network predicts the mean and variance for the data at the previous timestep based on the current noisy data and the current timestep. This continues until the original data or a new sample resembling the original data distribution is generated.

Mathematical Formulation: Being a generative model, it makes use of latent space information via the joint distribution. The joint distribution introduces the latent variables $x_1, \ldots, x_T$ and describes their dynamics. The final latent variable x_T is assumed to follow a standard Gaussian distribution.

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

The joint distribution $p_\theta(x_{0:T})$ serves as a bridge between the observed data and its latent counterparts.

This equation can be dissected into:

- An initial distribution for the last latent variable $p(x_T)$, which is rooted in a standard Gaussian.
- A compounded series of conditional distributions that trace the relationship across sequential timesteps.

Then the model makes use of conditional distributions to capture the temporal relationships between consecutive timesteps. They describe how the data evolves over time, transitioning from one timestep to the next.

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

This equation defines the conditional distribution of the data at the previous timestep x_{t-1} given the data at the current timestep x_t. It's Gaussian (or normal) in nature, parameterized by:

- $\mu_\theta(x_t, t)$: The mean of the distribution, which is a function of the current data x_t and the timestep t. It's determined by a parameterized function, likely a neural network.
- $\Sigma_\theta(x_t, t)$: The covariance matrix of the distribution, also a function of the current data and timestep.

This paper introduces a variational bound on the negative log likelihood for the denoising process.

$$E\left[-\log p_\theta(x_0)\right] \leq E_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$

This equation establishes a bound on the negative log likelihood of the observed data x_0. The expectation is taken over the distribution q, and the equation essentially compares the joint distribution of the observed data and latent variables $p_\theta(x_{0:T})$ with the conditional distribution of the latent variables given the observed data $q(x_{1:T}|x_0)$.

The bound is decomposed into two parts:

$$E_q\left[-\log p(x_T) - \sum_{t\geq 1} \frac{\log p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] =: L$$

- The log likelihood of the final latent variable x_T.
- The sum of log likelihood ratios comparing the reverse process (conditional distributions) with the forward process.

The variational bound L can be rewritten in terms of KL divergences, which measure the difference between two probability distributions:

$$E_q\left[D_{KL}(q(x_T|x_0)\|p(x_T))\right] + \sum_{t>1}\left[D_{KL}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t))\right] - \left[\log p_\theta(x_0|x_1)\right]$$

This equation compares the true and variational distributions using KL divergence for each data point.

1. $DKL(q(x_T|x_0)||p(x_T))$: This is the Kullback-Leibler (KL) divergence between the distribution q of the final state x_T given the initial state x_0 and the target distribution p of x_T. This term measures how different the predicted final state is from the target final state.
2. $\sum_{t>1} DKL(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$: This is a summation of KL divergences for each time step t. It measures the difference between the distribution q of the state x_{t-1} given the next state x_t and the initial state x_0, and the target distribution p_θ of x_{t-1} given x_t.
3. $-\log p_\theta(x_0|x_1)$: This term represents the negative log likelihood of the initial state x_0 given the next state x_1. It measures how likely the initial state is given the next state.

The context is about comparing two probability distributions using the Kullback-Leibler (KL) divergence. Specifically, the comparison is between the model's predicted distribution $p_\theta(x_{t-1}|x_t)$ and the "forward process posteriors" which are represented by $q(x_{t-1}|x_t, x_0)$. The forward process posteriors are distributions that describe the state x_{t-1} given the subsequent state x_t and the initial state x_0.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\right)$$

where

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}\beta_t}}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t} x_t$$

and

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

The mean $\tilde{\mu}_t$ is a weighted combination of the initial state x_0 and the subsequent state x_t. The weights are determined by the parameters α_t, $\bar{\alpha}_{t-1}$, and β_t.

The variance $\tilde{\beta}_t$ is adjusted based on the parameters $\bar{\alpha}_{t-1}$ and $\bar{\alpha}_t$.

The equations describe a process where the state of a system at time $t-1$ is influenced by both its initial state x_0 and its subsequent state x_t. The influence of these states is determined by the parameters α_t, $\bar{\alpha}_{t-1}$, and β_t. The distribution q captures the uncertainty about the state x_{t-1} given the information from x_0 and x_t.

After a few calculations and transformations, the final loss function used to compare the predicted data with observed data comes out to be something similar to the mean squared error where we subtract the difference between observed Gaussian noise with actual Gaussian noise at the time of denoising and sampling:

$$E_{x_0,\varepsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left(\varepsilon - \varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, t\right)\right)^2\right]$$

The field of 3D generative models is still emerging, with much room for advancement. A recent study by Perraudin et al. [21] **explored the use of GANs** to create 3D N-body simulations. In the research, they found challenges in managing large data samples using standard GAN designs. The data's wide range required specific adjustments, and achieving high accuracy was crucial for real-world cosmological applications. Additionally, representing rare training dataset features and addressing mode collapse added to the training complexities.

Diffusion models have been successful in avoiding the limitations of GAN such as mode collapse by learning the data distribution comprehensively and going for a probabilistic approach rather than an adversarial one. But for 3D diffusion models only a handful researches demonstrated satisfactory outcomes. However, as highlighted in the research paper **PVCNN** by Liu et al. [18] these models come with certain limitations. The 3D datasets can be represented in two ways conventionally:

1. **Point Clouds:** A point cloud is a collection of data points in a 3D space. Each point represents a location in that space. Point clouds are often used in 3D modeling and computer graphics to represent the external surface of an object.

2. **Voxels:** Voxels (short for "volume pixels") are the 3D equivalent of 2D pixels. They represent values in a 3D grid. Voxels are often used in volumetric rendering, where the interior of an object is as important as its surface.

In the early stages of this concept, voxel-based methods and point cloud-based methods were used but both methods had their own shortcomings:
Voxel grids are memory-intensive. Their memory requirements grow cubically with an increase in dimension. This makes it challenging to scale them to high resolutions. And decreasing the resolution leads to significant information loss because during the voxelization, multiple points would be merged together.

As for the **point based method**, it needs to compute the neighbours of the points. But unlike the 2D representation, the neighbours are no longer continuously on the memory as some neighbours might be in the middle or end or in a separate tensor. Also relative positions of the neighbours are fixed either hence forcing the kernel to do calculations dynamically. Therefore organising these neighbours requires a lot of memory access making it very computationally expensive and time consuming. [37]

4 Methodology

4.1 Dataset

4.1.1 IllustrisTNG

The *IllustrisTNG N-body simulations* in *CAMELS* project [32] were used as the dataset for 2D diffusion models in this dissertation. It consists of a map of the universe comprising of **405 images** of the resolution **256** × **256** stored as a numpy array. For pre-processing, the images were normalised to a range of **[-1,1]** and downsized to a resolution of **64** × **64**.

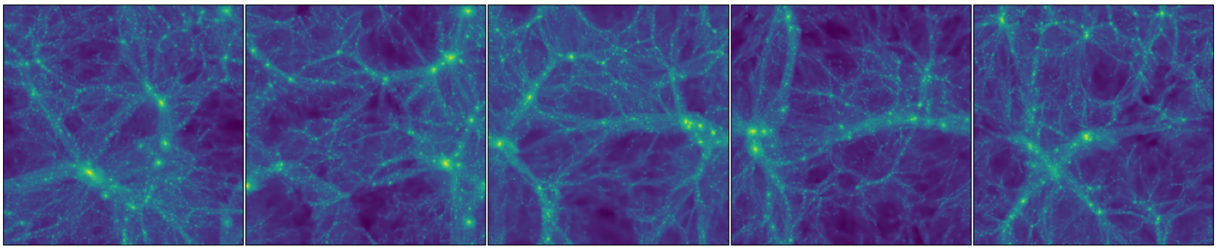


Figure 3: *The 2D dataset input of resolution* 256 × 256 *to DDPM model*[32]

4.1.2 Quijote

For 3D diffusion models, the simulation map from *QUIJOTE* project was picked. A single snapshot of **1GPC^{3}** was selected. For pre-processing, the snapshot was split into **8000 point clouds** being **50Mpc^{3}** each. Only the point clouds consisting of **15-17k** particles were kept and the rest were removed. These point clouds were downsampled to 15k particles first and then further randomly sampled **2048 particles** out of each point cloud. This was done as 15k particles in one point cloud were too much information model for diffusion to learn and it struggled with learning the dense and sparse regions due to it. The point clouds were also normalised as per **standard scaler** to have 0 mean and 1 as variance.

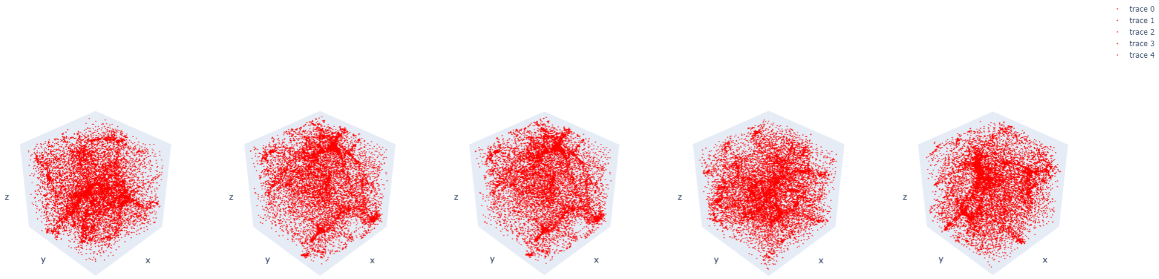


Figure 4: *The 3D dataset input to PVD model containing 15k particles*

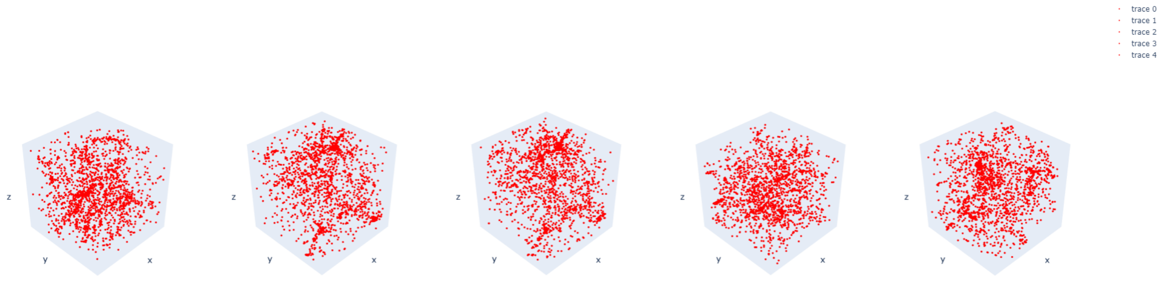


Figure 5: *The normalised 3D dataset input to PVD model containing 2048 particles*

The simulation conditions for both datasets are explained in Literature Review (3).

4.2 Neural network used for denoising process in DDPM (2D diffusion)

Unet architecture was used for the denoising process. The neural network was named Unet after the U shape of its architecture as the first component is downsampling which essentially decreases in size as we go deeper into the architecture and then upsampling blocks which restores the images resolution size from a lower spatial dimension to its original input dimension, hence outputting desirable result. [23]

As we move deeper into the network through the downsampling path, the spatial dimensions of the feature maps reduce, but the depth (number of channels) increases. This allows the network to extract more complex and abstract features from the input image. By reducing the spatial dimensions, the receptive field of the neurons in the deeper layers increases, allowing them to "see" larger portions of the input image. [23]

After capturing the broader context in the downsampling path, the upsampling path restores the spatial dimensions of the feature maps, aiming to produce a segmentation map with the same resolution as the original input image. While the downsampling path captures the broader context, the upsampling path focuses on precise localization, ensuring that the boundaries and details in the segmentation map are accurate. [23]
And to do these tasks, Unet implements two special types of blocks which are residual blocks and self-attention blocks to retain important information even while working on huge resolution inputs or deep neural networks:

Residual blocks: Deep neural networks frequently encounter an issue known as the vanishing gradients problem. This problem can hinder the network's learning process, as the gradients become too small for effective training. To combat this challenge, researchers introduced the concept of residual connections. These are essentially shortcuts that allow gradients to bypass certain layers and flow from one layer to another that's further along in the network. By doing so, a layer's input is directly added to its output, forming what's termed a "residual connection." It enhances the flow of gradients throughout the network hence allowing a model with hundreds of layers to train effectively. [14]

Self-attention block: Self-attention, as previously described, allows the model to weigh the importance of different spatial locations in a feature map, enabling the network to focus on relevant parts of the image. The block consists of Query (Q), Key (K), and Value (V) projections, which are typically implemented as 1x1 convolutions. The self-attention block allows each position in the input feature map to attend to all other positions, capturing global context. This is particularly beneficial in tasks where spatial relationships and long-range dependencies are crucial. [30]

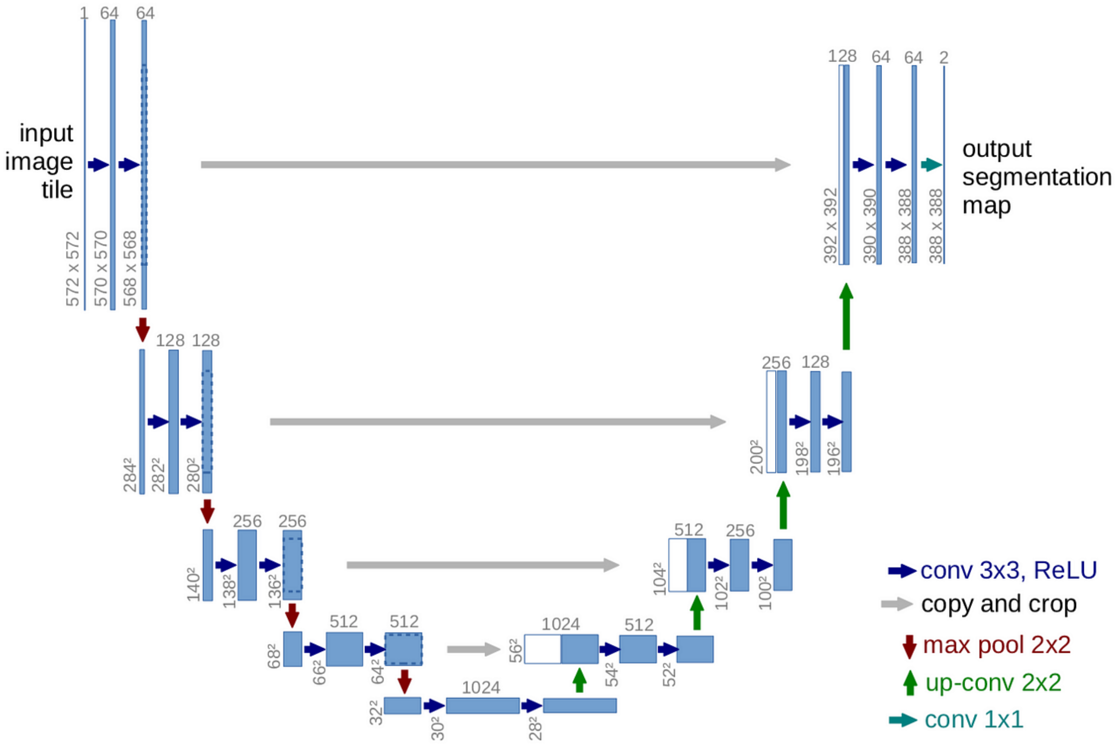


Figure 6: *UNet architecture*
[23]

Hyperparameters:

- number of input channels = 1,
- number of output channels = 1,
- number of resnet layers per Unet block = 2,
- number output channels for each Unet block = (128, 128, 256, 256, 512, 512),
- number of resnet downsampling blocks = 5 with 6th block connected with self attention block
- number of resnet upsampling blocks = 5 with 6th block connected with self attention block
- number of epochs = 80
- noise schedules = linear, cosine, sigmoid
- number of time steps = 1000

4.3 PVD

The PVD makes use of Point-voxel convolution instead of going for traditional 3d generative models like voxel based or point based due to their obvious shortcomings discuss in literature review above. [37] Point voxel convolution makes use of both point clouds and voxelization to retain valuable information while also working in branch hence making the runtime much faster. The point-based branch is used to run over the high resolution features while the voxel-based branch deals with lower resolution features. Point-based branch as dealing with high resolution is capable of extracting fine grained features which are still available in the input while the voxel-based grid focuses on extracting information about the neighbours. Hence by combining these methods, PVCNN makes use of small memory footprint advantage from point based methods and regularity from voxel based models. [37] The voxel branch learns to capture large continuous parts while the point branch captures isolated discontinuous details better.

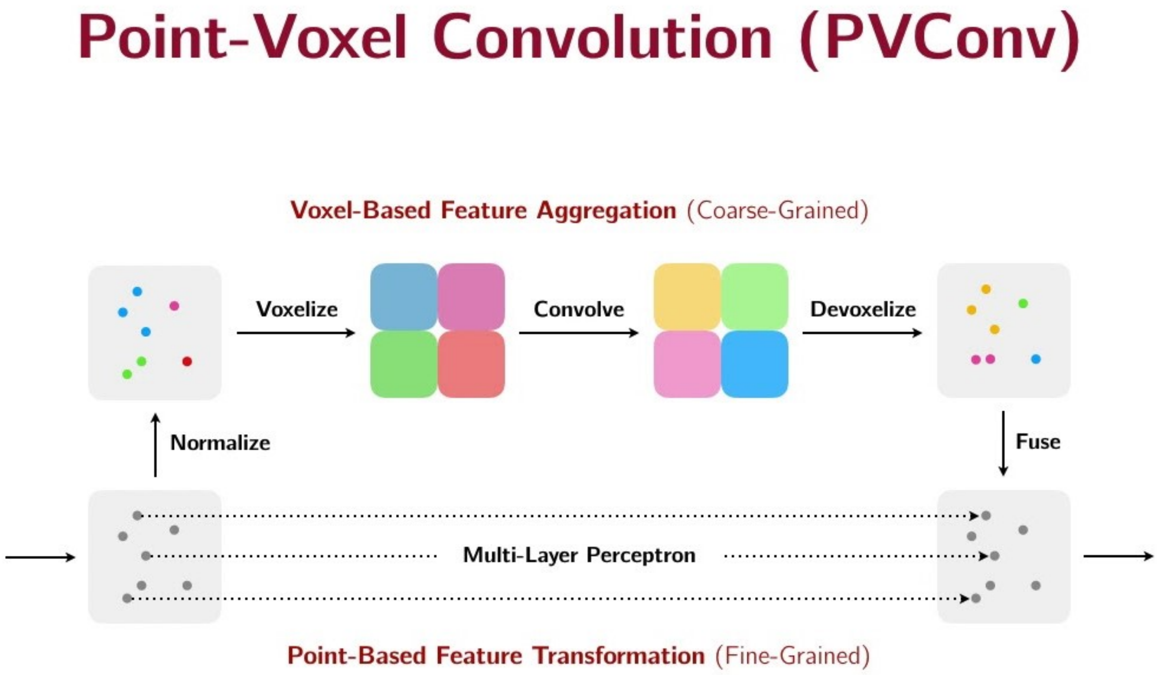

Point-Voxel Convolution (PVConv)
Voxel-Based Feature Aggregation (Coarse-Grained)
Voxelize
Convolve
Devoxelize
Normalize
Fuse
Multi-Layer Perceptron
Point-Based Feature Transformation (Fine-Grained)

Figure 7: *PVCnn architecture explaining how the model makes use of both voxel-based and point-based methods to retain information in higher and lower resolution efficiently*[18]

Key components of PVD:

1. **SharedMLP:** [37]
    - **Purpose:** A shared multi-layer perceptron (MLP) is designed to apply the same set of weights to multiple points simultaneously. This ensures that the same transformation is applied to each point in the point cloud, preserving the spatial structure.
    - **Usage:** In the context of point cloud processing, SharedMLP allows the model to learn point-wise features without being affected by the order or arrangement of points.
2. **PVConv (Point Voxel Convolution):** [18]

- **Purpose:** PVConv is a novel convolution layer that combines the features of both point-based and voxel-based representations. It aims to leverage the benefits of both representations to enhance the model's performance.
- **Usage:** By converting point clouds into voxel grids and then applying convolutions, PVConv can capture local structures more effectively. The combination of point and voxel features provides richer contextual information for tasks like completion.

3. **PointNetSAModule & PointNetAModule:** [18]
   - **Purpose:** These modules are from the PointNet++ architecture. The Set Abstraction (SA) module captures local structures by grouping nearby points and abstracting them into a single point. The Attention Module (AModule) focuses on specific features in the data.
   - **Usage:** These modules allow the model to hierarchically process point clouds, capturing both local and global structures. This hierarchical processing is crucial for tasks like object recognition and segmentation in point clouds.
4. **Attention:** [31]
   - **Purpose:** The attention mechanism allows the model to focus on specific parts of the input data that are more relevant to the task at hand.
   - **Usage:** In the context of point cloud processing, attention can help the model focus on crucial parts of an object or scene, enhancing its ability to recognize or complete structures.
5. **Swish Activation:**
   - **Purpose:** Swish is a self-gated activation function that has been found to perform better than traditional activation functions like ReLU in certain tasks.
   - **Formula:**

     $$\text{Swish}(x) = x \cdot \sigma(x)$$

     Where:
     - x is the input to the function.
     - $\sigma(x)$ is the sigmoid function, defined as:

       $$\sigma(x) = \frac{1}{1 + e^{-x}}$$
   - **Usage:** It introduces non-linearity into the model. It also is non-monotonic (i.e., it doesn't always increase or always decrease) enabling it to learn complex patterns and relationships in the data.

Hyperparameters:

- blocks used in architecture = PointNet, PointNet++, PVConv, PointNetSAModule, PointNetAModule
- time model was trained for = 105 hours
- number of epochs = 10000
- noise schedules = linear, cosine
- number of time steps = 1000

4.4 Hardware Used

4.4.1 MLiS machines:

- CPU = Intel(R) Xeon(R) W-2135 CPU @ 3.7GHz with 6 cores and 12 threads
- GPU = 2 Nvidia gtx 2080ti
- RAM = 64gb

4.4.2 Google Colab:

- CPU = Intel Xeon CPU with 2 vCPUs (virtual CPUs)
- GPU = NVIDIA Tesla T4 GPU
- RAM = 13gb

4.5 Software used

- Python
- Anaconda
- Jupyter Lab
- Google Colab
- Python libraries like numpy, matplotlib, plotly, pytorch, trimesh, os, scipy

4.6 Power spectrum

In simulations, the arrangement of matter can be converted into a different format called Fourier space using a technique known as the Fourier Transform. Here, each distinct pattern, termed a wavevector, indicates a particular size of variation.

The term “power spectrum”, represented as $P(k)$, describes the intensity of these variations based on their size, which is determined by the wavevector’s magnitude k. Essentially, it calculates the consistent strength or spread of changes in density at every size level.

By comparing the power spectrum from simulations with that from real-world data, like galaxy studies or background cosmic radiation, we can refine our models of the universe and evaluate the accuracy of theories about the universe’s structural development.

$$P(k) = |F(k)|^2$$

Where:

- $P(k)$ is the power spectrum as a function of wavevector k (or frequency, depending on the context).
- $F(k)$ is the Fourier transform of $f(t)$.
- $|F(k)|^2$ represents the square magnitude of the Fourier transform.

5 Results

The results were obtained by running the code on *MLiS Machines* with *GTX2080ti GPU*.

5.1 2D Diffusion (CAMELS Dataset)

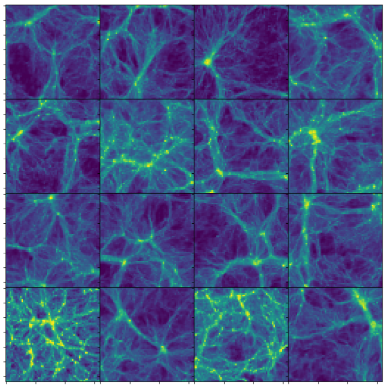


Figure 8: *N-body simulations obtained running Linear noise schedule on DDPM model*

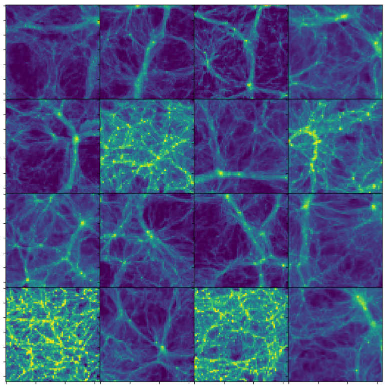


Figure 9: *N-body simulations obtained running Cosine noise schedule on DDPM model on IllustrisTNG*

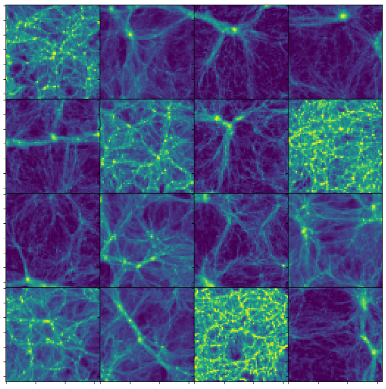


Figure 10: *N-body simulations obtained running Sigmoid noise schedule on DDPM model on IllustrisTNG*

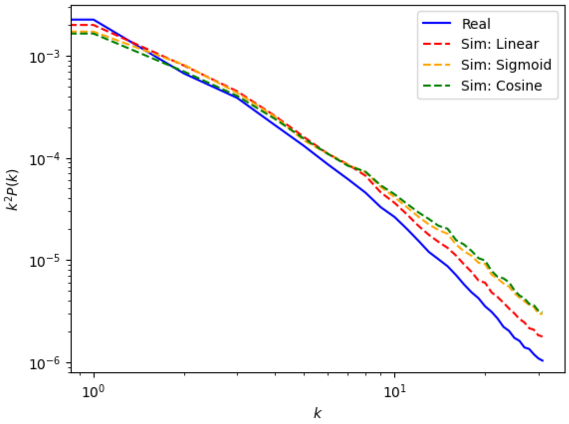


Figure 11: *Power spectrum for the simulations obtained by running DDPM model on 3 various noise schedules: linear, sigmoid, cosine*

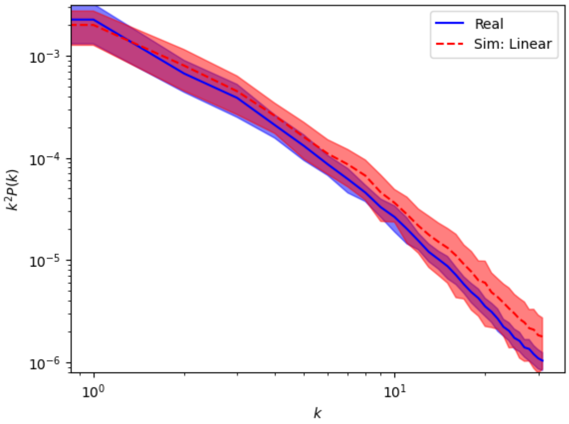


Figure 12: *Comparison between power spectrum of real data and simulated data while considering their envelopes*

Upon examining Figure 11, it is evident that the Linear noise schedule yielded the most accurate simulations, with its power spectrum closely aligning with that of the actual dataset. A deeper analysis, as presented in Figure 12, further confirms this observation. When comparing the envelopes of the power spectrum from the real dataset to that generated using the linear noise schedule, the model effectively replicated the characteristics of the IllustrisTNG simulations.

5.2 3D Diffusion (QUIJOTE Dataset)

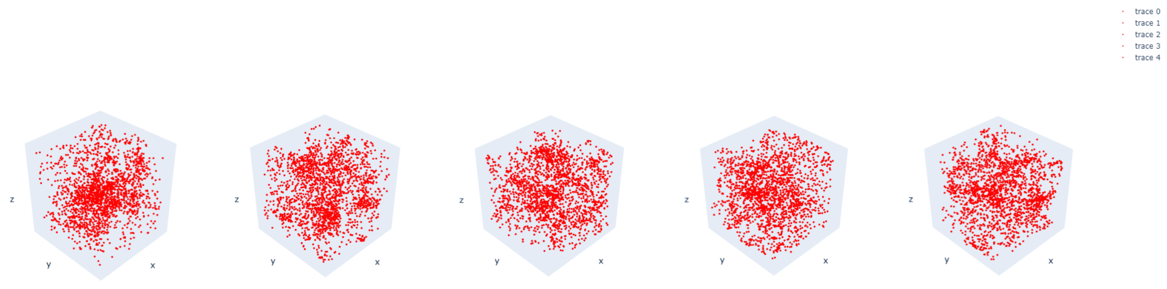
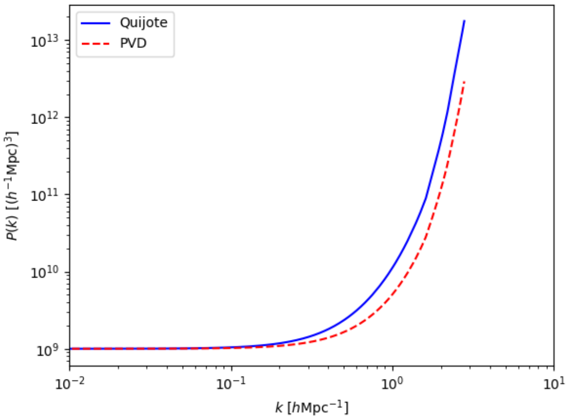


Figure 13: *N-body simulations generated using PVD model on Quijote*



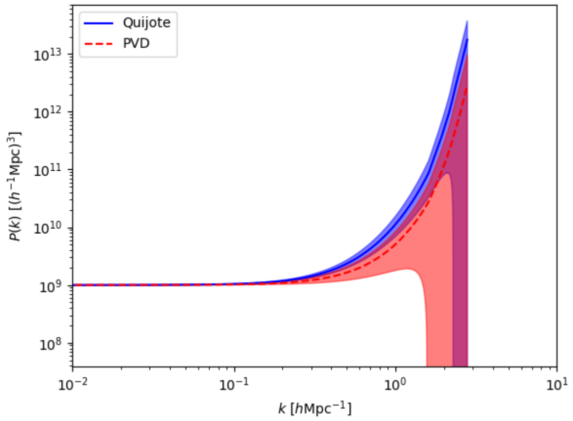


Figure 14: *Comparison between power spectrum of real data and simulated data*

Figure 15: *Comparison between power spectrum of real data and simulated data while considering their envelopes*

Analysing Figure 13, it can be seen that point voxel diffusion model was successful in replicating the patterns from the original Quijote simulations as dense and sparse regions inside the point clouds are clearly visible in the output simulations hence proving that the model did learn information fairly good despite the difficulties. Also the comparison between the power spectrum of real dataset (quijote) and images generated by PVD model (Figure 14) are quite close to each other hence further establishing their statistical relationship as we can see that the power spectrum of pvd generated images fall under the envelope region of real data meaning it is well under its standard deviation (Figure 15).

6 Discussion

The primary objective of our model was to replicate simulations using diffusion techniques that align closely with the power spectrum of the original simulation dataset. Our observations from the 2D dataset provided by CAMELS indicated that the power spectrum of our simulations closely mirrored the original. Specifically, the simulation's power spectrum was encompassed within the standard deviation envelope of the real data, suggesting a statistically significant correlation between the two image sets. Similarly, the 3D diffusion model (PVD) exhibited comparable results with the Quijote simulations in terms of wave number and wave power.

This study successfully employed the direct coordinates of particles within the point clouds, bypassing the use of voxels. This allowed the architecture to capture features at a foundational level, enhancing the model's precision in generating new samples. To the best of our understanding, this is the inaugural research introducing such a method for employing diffusion or PVD in n-body simulations, marking its novelty. The model's capability to accurately reflect both densely and sparsely populated areas within the point clouds highlights the viability of this method even though better results can still be obtained by tweaking architecture and parameters. This indicates the model's proficiency in discerning features, irrespective of the region's attributes or the position of particles, emphasizing its significance for academic research.

This successful replication, achieved in significantly less computational time and cost, underscores the dissertation's primary achievement. It demonstrates that machine learning algorithms can extract statistical insights from universe simulations without delving deep into the intricate mathematics involved. This is especially valuable when the mathematical computations become unfeasible due to extensive parameters or dimensions.

However, like all studies, ours had its limitations. We observed that the model's performance could be significantly enhanced with superior hardware. During our research, we experimented with different platforms and hardware configurations. Each upgrade in hardware facilitated more complex model architectures and reduced runtime. For instance, while the PVD model ran for 105 hours, it produced results comparable to the 35 million hours of Quijote simulations. Given diffusion models' inherent time-intensive nature, extended runtimes can potentially lead to better feature learning and improved image denoising. Increasing the number of time steps might further refine this denoising process.

Future improvements could also stem from expanding the input dataset and fine-tuning hyperparameters. Due to hardware limitations, the image resolution from CAMELS dataset was downsampled to 64×64 from 256×256 resulting in loss of information and the images being blurry and for Quijote, only a fraction of simulation was utilised. Linear noise schedule was observed to work the best for the models used in this project but exploring innovative noise introduction methods in the diffusion process could be a promising avenue for improvement as it's used for both diffusion and denoising process.

Lastly, the denoising architecture itself presents opportunities for enhancement. IDDPM (Improved Denoising Diffusion Probabilistic Models) which is an enhanced version of the original DDPM (Denoising Diffusion Probabilistic Models) can be used for 2D generative modelling. The improvements in IDDPM lead to more stable training dynamics, faster convergence, and potentially better-quality generated samples. By incorporating recent advances in the field, IDDPM can more efficiently handle datasets with complex distributions, offering optimized inference for real-time applications. Additionally, IDDPM often outperforms DDPM in standard generative modeling metrics, such as FID and Inception Score. However, the extent of improvement can vary based on the specific context or dataset, making empirical evaluation crucial.

The field of 3D generation is rapidly evolving, with frameworks like LION [34] emerging as potential game-changers. LION operates as a variational autoencoder (VAE) and boasts a hierarchical latent space, merging a global shape latent representation with a point-centric latent space. By training two hierarchical DDMs within these spaces, the hierarchical VAE method showcases improved performance over DDMs that work directly on point clouds. Yet, the point-based latents remain optimal for DDM-centric modeling. Another promising model is Control3Diff [12], which merges the strengths of diffusion models and 3D GANs. Control3Diff uniquely models the latent distribution, potentially conditioned on external inputs, granting direct control during diffusion. Its ability to model latent distributions and provide direct control during the diffusion process makes it a potential candidate for future research.

In conclusion, our study has laid the groundwork for leveraging machine learning in universe simulations. While our results are promising, the field is ripe for further exploration and refinement.

7 Conclusion

In this dissertation, our primary objective was to harness the capabilities of machine learning, specifically 2D and 3D diffusion models, to simulate the universe. The challenges posed by traditional simulation methods, particularly their extensive computational demands, necessitated an exploration of alternative approaches. Our results have demonstrated the efficacy of our chosen methods.

Our models successfully replicated the power spectrum of the original simulations, achieving this with remarkable accuracy. This accomplishment was realized in a fraction of the computational time traditionally required, highlighting the efficiency and precision of our approach. Even though there is still scope for improvement, the ability of our models to replicate these simulations still validates the potential of machine learning in this domain but also paves the way for more efficient and expansive research in cosmology.

The exploration of advanced diffusion methods, such as latent and point voxel diffusion, has provided a nuanced understanding of cosmic structures. These methods, when combined with machine learning, offer a revolutionary approach to simulating and understanding matter behavior and evolution in the universe.

While our research has yielded promising results, the potential for further refinement and exploration remains. The introduction of more sophisticated models, enhancements in hardware, and innovative noise introduction methods all present opportunities for future research.

In summary, this dissertation has successfully achieved its primary aim, demonstrating the transformative potential of machine learning in universe simulations. As we move forward, the fusion of these computational techniques with traditional cosmological knowledge holds the promise of deeper insights and a more profound understanding of the universe's intricacies.

References

[1] Aarseth, S. J. and Aarseth, S. J. [2003], *Gravitational N-body simulations: tools and algorithms*, Cambridge University Press.

[2] Ade, P. A., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., Banday, A., Barreiro, R., Bartlett, J., Bartolo, N. et al. [2016], 'Planck 2015 results-xiii. cosmological parameters', *Astronomy & Astrophysics* **594**, A13.

[3] Barnes, J. and Hut, P. [1986], 'A hierarchical o (n log n) force-calculation algorithm', *nature* **324**(6096), 446–449.

[4] Barrow-Green, J. [1997], *Poincaré and the three body problem*, number 11, American Mathematical Soc.

[5] Bertschinger, E. [1998], 'Simulations of structure formation in the universe', *Annual Review of Astronomy and Astrophysics* **36**(1), 599–654.

[6] Brooks, S., Gelman, A., Jones, G. and Meng, X.-L. [2011], *Handbook of markov chain monte carlo*, CRC press.

[7] Chandrasekhar, S. [2005], *Principles of stellar dynamics*, Courier Corporation.

[8] Davis, M., Efstathiou, G., Frenk, C. S. and White, S. D. [1985], 'The evolution of large-scale structure in a universe dominated by cold dark matter', *Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 292, May 15, 1985, p. 371-394. Research supported by the Science and Engineering Research Council of England and NASA.* **292**, 371–394.

[9] Einstein, A. et al. [1916], 'The foundation of the general theory of relativity', *Annalen Phys* **49**(7), 769–822.

[10] Friedmann, A. [1922], 'Über die krümmung des raumes', *Z. Phys.* **10**, 377–386.

[11] Fukue, J. [2012], 'Radiative transfer in anisotropic scattering media', *Publications of the Astronomical Society of Japan* **64**(6), 132.

[12] Gu, J., Gao, Q., Zhai, S., Chen, B., Liu, L. and Susskind, J. [2023], 'Learning controllable 3d diffusion models from single-view images', *arXiv preprint arXiv:2304.06700* .

[13] Hawking, S. [1988], *A Brief History of Time.*, Bantam Books.

[14] He, K., Zhang, X., Ren, S. and Sun, J. [2016], Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

[15] Ho, J., Jain, A. and Abbeel, P. [2020], 'Denoising diffusion probabilistic models'.
URL: *https://arxiv.org/pdf/2006.11239.pdf*

[16] Hockney, R. W. and Eastwood, J. W. [2021], *Computer simulation using particles*, crc Press.

[17] Lewis, A., Challinor, A. and Lasenby, A. [2000], 'Efficient computation of cosmic microwave background anisotropies in closed friedmann-robertson-walker models', *The Astrophysical Journal* **538**(2), 473.

[18] Liu, Z., Tang, H., Lin, Y. and Han, S. [2019], 'Point-voxel cnn for efficient 3d deep learning', *Advances in Neural Information Processing Systems* **32**.

[19] Mihalas, D. and Mihalas, B. [1984], ‘Foundations of radiation hydrodynamics oxford university press’.

[20] Peebles, P. J. E. and Ratra, B. [2003], ‘The cosmological constant and dark energy’, *Reviews of modern physics* **75**(2), 559.

[21] Perraudin, N., Srivastava, A., Lucchi, A., Kacprzak, T., Hofmann, T. and Réfrégier, A. [2019], ‘Cosmological n-body simulations: a challenge for scalable generative models’, *Computational Astrophysics and Cosmology* **6**, 1–17.

[22] Riess, A. G., Filippenko, A. V., Challis, P., Clocchiatti, A., Diercks, A., Garnavich, P. M., Gilliland, R. L., Hogan, C. J., Jha, S., Kirshner, R. P. et al. [1998], ‘Observational evidence from supernovae for an accelerating universe and a cosmological constant’, *The astronomical journal* **116**(3), 1009.

[23] Ronneberger, O., Fischer, P. and Brox, T. [2015], U-net: Convolutional networks for biomedical image segmentation, *in* ‘Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18’, Springer, pp. 234–241.

[24] Rubin, V. C. and Ford Jr, W. K. [1970], ‘Rotation of the andromeda nebula from a spectroscopic survey of emission regions’, *Astrophysical Journal, vol. 159, p. 379* **159**, 379.

[25] Springel, V. [2005], ‘The cosmological simulation code gadget-2’, *Monthly notices of the royal astronomical society* **364**(4), 1105–1134.

[26] Springel, V. [2010], ‘E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh’, *Monthly Notices of the Royal Astronomical Society* **401**(2), 791–851.

[27] Sussman, G. J. and Wisdom, J. [1992], ‘Chaotic evolution of the solar system’, *Science* **257**(5066), 56–62.

[28] Tremblin, P., Chabrier, G., Padioleau, T. and Daley-Yates, S. [2022], ‘Nonideal self-gravity and cosmology: Importance of correlations in the dynamics of the large-scale structures of the universe’, *Astronomy & Astrophysics* **659**, A108.

[29] Tyson, J. A., Valdes, F. and Wenk, R. [1990], ‘Detection of systematic gravitational lens galaxy image alignments-mapping dark matter in galaxy clusters’, *Astrophysical Journal, Part 2-Letters (ISSN 0004-637X), vol. 349, Jan. 20, 1990, p. L1-L4.* **349**, L1–L4.

[30] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. [2017], ‘Attention is all you need’, *Advances in neural information processing systems* **30**.

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. [2023], ‘Attention is all you need’.

[32] Villaescusa-Navarro, F., Anglés-Alcázar, D., Genel, S., Spergel, D. N., S. Somerville, R., Dave, R., Pillepich, A., Hernquist, L., Nelson, D., Torrey, P., Narayanan, D., Li, Y., Philcox, O., La Torre, V., Maria Delgado, A., Ho, S., Hassan, S., Burkhart, B., Wadekar, D., Battaglia, N., Contardo, G. and Bryan, G. L. [2021], ‘The camels project: Cosmology and astrophysics with machine-learning simulations’, *The Astrophysical Journal* **915**, 71.
URL: *https://arxiv.org/pdf/2010.00619.pdf*

[33] Villaescusa-Navarro, F., Hahn, C., Massara, E., Banerjee, A., Delgado, A. M., Ramanah, D. K., Charnock, T., Giusarma, E., Li, Y., Allys, E. et al. [2020], 'The quijote simulations', *The Astrophysical Journal Supplement Series* **250**(1), 2.

[34] Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S. and Kreis, K. [2022], 'Lion: Latent point diffusion models for 3d shape generation', *arXiv preprint arXiv:2210.06978* .

[35] Zennaro, M., Bel, J., Villaescusa-Navarro, F., Carbone, C., Sefusatti, E. and Guzzo, L. [2017], 'Initial conditions for accurate n-body simulations of massive neutrino cosmologies', *Monthly Notices of the Royal Astronomical Society* **466**(3), 3244–3258.

[36] Zhou, L., Du, Y. and Wu, J. [2021*a*], '3d shape generation and completion through point-voxel diffusion'.
URL: *https://arxiv.org/pdf/2104.03670.pdf*

[37] Zhou, L., Du, Y. and Wu, J. [2021*b*], 3d shape generation and completion through point-voxel diffusion, *in* 'Proceedings of the IEEE/CVF International Conference on Computer Vision', pp. 5826–5835.