



MASTERARBEIT | MASTER'S THESIS

Titel | Title

Differentiable Identification of Cosmic Structures

verfasst von | submitted by

Jim Buisman BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 861

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Astronomie

Betreut von | Supervisor:

Univ.-Prof. Dipl.-Phys. Dr. sc. ETH Oliver Hahn

Mitbetreut von | Co-Supervisor:

Florian List B.Sc. M.Sc. PhD

Abstract

The aim of this thesis is devising and implementing methods for differentiable identification of cosmic structures in cosmological simulations. Their use in combination with automatic differentiation tools such as JAX enables direct analysis of the structure's dependence on cosmological parameters, which is particularly important for applications requiring gradient information, such as Hamiltonian Monte Carlo (HMC), gradient-based optimization, and machine learning techniques. We choose two different structures and approaches for this purpose.

On the largest scales, the universe can be categorized by a few basic structures comprising the 'cosmic web'. This filamentary structure is divided into four distinct environments: clusters, filaments, sheets, and voids. We distinguish them by applying the T-web formalism, which is based on the tidal field tensor. It describes how the gravitational potential deforms in different spatial directions and can classify these environments by its eigenvalues. We are able to analyze the effects of different cosmological parameters on the formation of each environment. The cosmic web can even be approximated for different cosmologies without having to rerun any simulation by leveraging the derivative information to the first order and applying the change linearly. This approach works well enough as long as the changes to the parameter are small enough. Furthermore, the readily available derivatives could be used for parameter constraints using HMC, especially in combination with environment-specific power spectra estimates, though this is only discussed and not tested.

The second approach deals with dark matter haloes, which are gravitationally bound structures that have decoupled from the Hubble expansion and collapsed. We use a convolutional neural network with a U-Net architecture to predict which particles in the initial density field will end up in haloes of a certain mass. These regions in the early universe are known as protohalo patches. The individual particles are assigned their virial halo mass and binned into 10 logarithmically spaced classes. This classification method is automatically differentiable due to the probabilistic predictions of neural networks. The model is able to predict protohalo particles with a precision of 86.74%, though it struggles with the smallest mass bin, which indicates that the simulation resolution is too low to be unambiguous enough for the neural network. The U-Net is able to reproduce the total distribution of haloes remarkably well compared to standard halo finders such as ROCKSTAR, and theoretical halo mass functions (HMF) like Tinker08. The HMF is predicted with a mean difference of 5.41% across the entire mass range. Furthermore, the dependence of these haloes on classical Λ CDM parameters can be captured directly through the use of automatic differentiation. The results show that Ω_b , Ω_c , and σ_8 affect the HMF the most, as well as h and n_s having a negative derivative for larger haloes. Other verified approaches show the same correlations, which proves the success of our classification U-Net.

Kurzfassung

Das Ziel dieser Arbeit ist die Entwicklung und Implementierung von Methoden für differenzierbare Identifikation kosmischer Strukturen in numerischen Simulationen. In Kombination mit automatischer Differenzierung, beispielsweise durch JAX, ermöglicht sie eine direkte Analyse der Abhängigkeit dieser Strukturen von kosmologischen Parametern. Dies ist insbesondere für Anwendungen wichtig, die Gradienten benötigen, wie etwa Hamiltonian Monte Carlo (HMC), Gradientenverfahren und maschinelles Lernen. Für die Umsetzung wählen wir zwei verschiedene Arten von Strukturen und Ansätzen.

Auf den größten Skalen kann das Universum mit nur ein paar grundlegenden Strukturen des sogenannten ‘kosmischen Netzes’ kategorisiert werden. Der Aufbau wird in vier verschiedene Umgebungen unterteilt: Knoten, Filamente, Wände und Hohlräume. Wir unterscheiden sie mithilfe des T-Web-Formalismus, der auf dem Gezeitenfeld-Tensor basiert. Dieser beschreibt, wie sich das Gravitationspotential in verschiedene Raumrichtungen verformt, und klassifiziert die Umgebungen anhand seiner Eigenwerte. Wir analysieren die Auswirkungen unterschiedlicher kosmologischer Parameter auf die Entstehung der jeweiligen Strukturen. Des Weiteren kann das kosmische Netz für verschiedene Kosmologien approximiert werden, ohne die Simulation erneut ausführen zu müssen, indem die erste Ableitung genutzt wird, um die Änderung linear zu addieren. Der Ansatz funktioniert gut, solange die Änderungen der Parameter klein genug bleiben. Darüber hinaus könnten die verfügbaren Ableitungen für Parameterbeschränkung mittels HMC genutzt werden, insbesondere in Kombination mit umgebungsspezifischen Messungen des Leistungsspektrums, wobei dies nur diskutiert, aber nicht getestet wird. Der zweite Ansatz beschäftigt sich mit Dunkle-Materie-Halos, also gravitativ gebundenen Strukturen, die von der Hubble-Expansion entkoppelt und kollabiert sind. Wir verwenden ein neuronales Netzwerk mit einer U-Net-Architektur, um vorherzusagen, welche Teilchen im anfänglichen Dichtefeld letztendlich in Halos einer bestimmten Masse enden werden. Diese Regionen im frühen Universum sind als Protohalo bekannt. Den darin befindlichen Teilchen wird die viriale Masse des Halos zugewiesen und in zehn logarithmisch verteilte Klassen eingeteilt. Durch die probabilistischen Vorhersagen neuronaler Netzwerke ist diese Klassifizierungsmethode automatisch differenzierbar. Das Modell kann Protohalo-Teilchen mit einer Präzision von 86,74% vorhersagen, wobei es die größten Schwierigkeiten mit der kleinsten Massenkategorie hat. Dies deutet darauf hin, dass die Simulationsauflösung für eine eindeutige Klassifikation zu niedrig ist. Das U-Net kann die Massenverteilung der Halos im Vergleich zu typischen Halo-Findern wie ROCKSTAR, als auch theoretischen Halo-Massenfunktionen (HMF) wie Tinker08 bemerkenswert gut reproduzieren. Die HMF stimmt mit einer durchschnittlichen Abweichung von 5,41% über den gesamten Massenbereich mit diesen Methoden überein. Darüber hinaus wird die Abhängigkeit der Halos von klassischen Λ CDM-Parametern untersucht. Die Ergebnisse zeigen, dass Ω_b , Ω_c und σ_8 den größten Einfluss auf die HMF haben, während h und n_s für größere Halos einen negativen Gradienten aufweisen. Andere etablierte Methoden zeigen die gleichen Korrelationen, was den Erfolg unserer U-Net-Methode bestärkt.

Contents

Abstract	i
Kurzfassung	iii
1 Introduction	1
1.1 Parameters	3
1.2 Numerical Simulations	6
1.2.1 Initial Conditions	6
1.2.2 Evolving Dark Matter Structures	8
1.2.3 Differentiable Simulations	10
1.3 Cosmic Structures	13
2 Cosmic Web Environments with T-web	15
2.1 Methods	16
2.1.1 Classification on Initial Conditions	17
2.2 Results	23
2.2.1 Classification on evolved Simulations	23
2.2.2 Differentiability	25
2.3 Further Applications	29
3 Dark Matter Haloes	31
3.1 Methods	33
3.1.1 Halo Particles	39
3.1.2 N-body code	45
3.1.3 Differentiability	48
3.2 Convolutional Neural Networks	49
3.2.1 Historical Introduction	49
3.2.2 Halo Identification with CNNs	51
3.2.3 Classification vs Regression	53
3.2.4 Final Method	56
3.3 Results	61
3.3.1 Halo Mass Function	64
3.3.2 Differentiability	67
3.4 Discussion	71
4 Conclusion	75
Bibliography	79

1 Introduction

The Λ Cold Dark Matter (Λ CDM) model stands as the prevailing cosmological framework for understanding the evolution and structure of the universe (Peebles, 1984; Carroll, 2001). It elegantly combines two key elements: the cosmological constant Λ , representing dark energy - a homogeneous force driving the universe's accelerated expansion (Carroll et al., 1992) - and Cold Dark Matter (CDM), a type of non-interacting matter that clumps together under gravity, providing the necessary structure around which galaxies form (Zwicky, 1933; Zwicky, 1937; Bertone et al., 2005). In addition to these two parts, the cosmological model also accounts for ordinary matter (baryons, leptons) and radiation (photons, neutrinos) to create the universe we see and know. Developed from the combination of observational evidence and theoretical predictions, the Λ CDM model not only explains the observed large-scale structure of the universe (Spergel et al., 2007), but also aligns with measurements of the cosmic microwave background and the distribution of galaxies (Planck Collaboration, 2020). Its unmatched success in describing the data emphasizes the importance of dark energy and dark matter in modern cosmology, offering profound insights into the fundamental nature and evolution of our cosmos.

Modern cosmology is built upon two key hypotheses. The first of these is Albert Einstein's theory of General Relativity (GR, see Einstein, 1917), which provides the framework for describing the dynamics of the universe on the largest scales. GR postulates that gravity is not a force in the traditional sense but rather the curvature of spacetime caused by mass and energy. The equations of GR, specifically the Einstein field equations, relate the geometry of spacetime to the distribution of matter and energy within it. In the context of the Λ CDM model, these equations include the cosmological constant (Λ), which represents the dark energy responsible for the accelerated expansion of the universe (Riess, Filippenko et al., 1998; Perlmutter et al., 1999).

The second hypothesis at the heart of the Λ CDM model is the cosmological principle (Weinberg, 1972), which states that the universe is homogeneous and isotropic on large enough scales ($\gtrsim 100$ Mpc). This means that, when viewed from a sufficient distance, the universe appears the same in all directions (isotropy) and has a uniform composition throughout (homogeneity). This principle simplifies the mathematical modeling of the universe by ignoring the collection of individual stars, galaxies, and galaxy clusters and instead treating it as a smooth and continuous space. It is the basis for all dark energy models, as the accelerated expansion only implies dark energy under the assumption that the cosmological principle holds (Bull et al., 2016). Directly testing this hypothesis is not as straightforward. Homogeneity cannot be directly tested since it is not possible for

1 Introduction

humans to view the cosmos from any other point in the universe. However, if one assumes that the Earth is not special in its location, then isotropy would imply homogeneity. So far, observations of the cosmic microwave background (CMB) have shown that it is isotropic (Planck Collaboration, 2016). By itself, this does not fully prove an isotropic universe, but the additional details and reasons will not be discussed here further (see Bull et al., 2016 for in-depth explanations).

The Cosmological Principle is fundamental to the Λ CDM model because the resulting geometric symmetries simplify equations governing the universe's evolution immensely and justify its description with the Friedmann-Lemaître-Robertson-Walker (FLRW) metric. It is generally written as:

$$ds^2 = -c^2 dt^2 + a(t)^2 d\Sigma^2, \quad (1.1)$$

where ds represents the spacetime interval, c is the speed of light, dt the differential of cosmic time, $a(t)$ the scale factor (the only dynamical degree of freedom), and $d\Sigma$ is the 3-metric of maximally symmetric space (Kumar Aluri et al., 2023).

The FLRW metric is the only exact solution of the Einstein field equations that is globally spatially homogeneous and isotropic at all times (Straumann, 2013). It provides a framework for describing the evolution of the scale factor of the universe ($a = 1/(1+z)$), which determines how distances between objects change over time, where z represents the redshift. The metric takes into account the curvature of space, allowing for different possible geometries of the universe: flat, open, or closed. When combined with the field equations, the FLRW metric leads to the Friedmann equations (Friedmann, 1922), which govern the expansion rate of the universe based on its energy content, including baryonic matter, radiation, dark matter, and dark energy. For flat Λ CDM they can be written as:

$$H^2 \equiv \frac{\dot{a}^2}{a^2} = \frac{8\pi G\rho}{3} + \frac{\Lambda c^2}{3}, \quad (1.2)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \left(\rho + \frac{3p}{c^2} \right) + \frac{\Lambda c^2}{3}, \quad (1.3)$$

where H is the Hubble parameter, which measures the rate of expansion of the universe as a function of time, G is the gravitational constant, ρ is the energy density, p is the energy pressure, and c is the speed of light. The Friedmann equation can also be rewritten in terms of the critical density ρ_{crit} , which represents the energy density required for a flat universe:

$$\rho_{\text{crit}} = \frac{3H^2}{8\pi G}. \quad (1.4)$$

Using this, the total density can be expressed in dimensionless form as the density parameter,

$$\Omega = \frac{\rho}{\rho_{\text{crit}}}, \quad (1.5)$$

where Ω represents the fractional contribution of a given energy component to the total density of the universe. The Friedmann equation in terms of Ω then becomes

$$H^2 = H_0^2 \left(\Omega_m a^{-3} + \Omega_\Lambda \right), \quad (1.6)$$

where Ω_m and Ω_Λ correspond to the density parameters for matter and dark energy, respectively, and H_0 is the Hubble constant today. Additionally, they can also be expressed via the equation of state with

$$w = \frac{p_\Lambda}{\rho_\Lambda}. \quad (1.7)$$

Thus far, measurements found that w is very close to -1 , which is only possible for vacuum energy and hence indicative of a cosmological constant (Carroll, 2001). However, recent studies suggest a preference for time-varying dark energy, highlighting the need for continued exploration (DESI Collaboration, Adame et al., 2024).

1.1 Parameters

Building upon the concepts of General Relativity and the Cosmological Principle, as well as the resulting equations and metrics that describe the dynamics of the universe, it is now time to explore the specific cosmological parameters that define the Λ CDM model. These parameters encapsulate the different components and properties of the universe, such as its energy density, curvature, and rate of expansion. The Λ CDM model is usually characterized by six free parameters, as it represents the simplest model (Lahav and Liddle, 2022) that fits the observed data, in accordance with Occam's Razor. Any other parameters are either fixed at specific values or derived from the six previous parameters. However, it is important to note that the specific set of independent parameters chosen can vary (e.g., WMAP parameters: Spergel et al., 2007 compared to Planck parameters: Planck Collaboration, 2020) depending on the focus of the study or the particular cosmological model used. The most significant parameters for this work are explained in detail in this following section.

Hubble Constant (H_0)

The Hubble constant, denoted as H_0 , quantifies the current rate of expansion of the universe. The phenomenon was first discovered by Edwin Hubble in 1929 by measuring the relation between the recessional velocity of nearby galaxies and their distance from us, as observed in the redshift of light (Hubble, 1929). The value of H_0 is crucial in determining the age and size of the universe. However, its exact value remains a major topic of debate, with discrepancies between measurements from the CMB observations (e.g., Planck Collaboration, 2020) and local distance ladder methods using Cepheid variables and Type Ia supernovae (e.g., Riess, Yuan et al., 2022). This tension is known as the 'Hubble tension' and continues to be one of the biggest issues in modern cosmology (e.g., Kamionkowski and Riess, 2023, Hu and Wang, 2023). h is a dimensionless form of H_0 , which normalizes it to 100 km/s/Mpc.

Matter Density Parameter (Ω_m)

The matter density parameter, Ω_m , represents the total matter content of the universe,

1 Introduction

including both baryonic (ordinary) matter and dark matter: $\Omega_m = \Omega_b + \Omega_c$. CMB data has indicated a flat universe (Planck Collaboration, 2014), which means $\Omega_m + \Omega_\Lambda \simeq 1$. In combination with light-curves from supernovae type Ia (Scolnic et al., 2018) the best-fit values for Ω_m are approximately 0.3, indicating that matter constitutes about 30% of the total energy density of the universe. The other 70% are contributed by dark energy density, Ω_Λ , which is responsible for the accelerated expansion of the universe.

Baryon Density Parameter (Ω_b)

The baryon density parameter, Ω_b , specifically refers to the contribution of ordinary baryonic matter (protons, neutrons, and electrons) to the total energy density of the universe. It is a subcomponent of Ω_m and is crucial to understanding the formation of light elements during Big Bang Nucleosynthesis (BBN) and the subsequent formation of galaxies and stars (Cyburt et al., 2016). By separating the density into its different components, it is possible to better grasp their individual roles in the evolution of the universe. However, this can only be back-tracked until neutrino decoupling (shortly before BBN) because the different density species were able to transform between flavors before that (Akita and Yamaguchi, 2022). The baryonic density is usually estimated by one of two methods: CMB measurements such as the Planck mission, or primordial deuterium abundance (D/H) coupled with BBN theory (Cooke et al., 2016). CMB data has shown the highest accuracy in recent years. It suggests that $\Omega_b \approx 0.05$, indicating that baryons comprise about 5% of the total energy density of the universe (Planck Collaboration, 2020).

Cold Dark Matter Density Parameter (Ω_c)

The second subcomponent of the matter density is dark matter, which is non-baryonic, interacts only through gravity and has a negligible velocity dispersion ('cold'), which is of importance for structure formation (Lahav and Liddle, 2022). It accounts for the majority of the matter density in the universe. Current estimates of the dark matter density parameter, Ω_c suggest it to be ≈ 0.25 , which is five times more than the baryonic density. This implies that dark matter accounts for about 25% of the universe's total energy density (Planck Collaboration, 2020).

Curvature Parameter (Ω_k)

As mentioned above, observational evidence suggests that the universe is flat. However, in order to also account for a different geometry, the spatial curvature parameter, Ω_k , is introduced. It is defined as: $\Omega_k = 1 - \Omega_m - \Omega_\Lambda$. If $\Omega_k = 0$, the universe is indeed flat; if $\Omega_k > 0$, the universe has a positive curvature (closed); and if $\Omega_k < 0$, the universe has a negative curvature (open). Observations from the CMB in combination with baryon acoustic oscillation (BAO) measurements strongly suggest that the universe is very close to flat, with Ω_k near zero (Planck Collaboration, 2020).

Scalar Spectral Index (n_s)

The scalar spectral index, n_s , characterizes the scale dependence of the primordial fluctuations in the universe. It is connected to the primordial power spectrum, which

is typically described by a power law $P(k) \propto k^{n_s-1}$. This spectrum defines how the density perturbations that seeded the large-scale structure of the universe vary with scale (Peiris et al., 2003). A value of $n_s = 1$ corresponds to a scale-invariant spectrum. However, the current understanding of the inflationary period predicts a value slightly below 1, indicating that smaller-scale fluctuations were slightly suppressed compared to larger ones (Baumann, 2009). The observed CMB anisotropies are caused by these field fluctuations, which makes n_s such an important parameter. By carefully measuring the strength of the CMB anisotropies across all angular scales, the Planck Collaboration, 2020 found that $n_s \approx 0.96$, implying a slight tilt in the spectrum towards larger scales. As will be shown later, n_s also has a significant effect on large-scale structure formation.

Amplitude of Matter Fluctuation (σ_8)

The parameter σ_8 describes the amplitude of density fluctuations on a fiducial scale of $8h^{-1}$ Mpc in the universe. It is defined as the root mean squared of the density perturbations at $z = 0$. σ_8 is a measure of the clustering of matter and influences the formation of structures such as galaxies and galaxy clusters (Addison et al., 2013). The Planck Collaboration, 2020 estimate $\sigma_8 = 0.81$.

Equation of State Parameters for Dark Energy (w_0, w_a)

The equation of state (EOS) parameter w has been mentioned above, but in order to account for a changing value with redshift, a simple linear relation can be applied: $w(a) = w_0 + w_a(1 - a)$ (Linder, 2003). The dark energy equation of state parameters, w_0 and w_a , describe the time evolution of dark energy. The parameter w_0 represents the present value of the EOS, while w_a describes the potential evolution of w with the scale factor a . Therefore, $w_a = 0.0$ implies no evolution. In combination with $w_0 = -1.0$ (Planck Collaboration, 2020), it is consistent with a cosmological constant. Recently, DESI Collaboration, Adame et al., 2024 showed a preference for $w_0 > -1$ and $w_a < 0$, highlighting the importance of further investigation.

There are many more parameters that were not mentioned (see Lahav and Liddle, 2022 for more information), but the aforementioned are the most important ones for this particular work, where their effect on cosmic structures in computer simulations is tested. These parameters collectively define the Λ CDM model and provide a comprehensive description of the composition, geometry, and evolution of the universe. Observations, particularly from the CMB, large-scale structure surveys, and supernova measurements, continue to tighten the constraints of these parameters, allowing for increasingly precise tests of the Λ CDM model and potential deviations from it (Perivolaropoulos and Skara, 2022).

1.2 Numerical Simulations

Cosmological simulations have become indispensable in modern astrophysics for studying the structure and evolution of the universe. These simulations are able to model the formation of galaxies, dark matter haloes, and cosmic web structures, producing theoretical data that can be compared against cosmological observations. Early simulations in the 1960s modeled dynamical evolution using only a few hundred particles, focusing primarily on gravitational interactions (von Hoerner, 1960; Aarseth and Hoyle, 1963). They have since evolved to include more complex physical processes, such as hydrodynamics, star formation, and feedback mechanisms (e.g., Pillepich et al., 2018; Pakmor et al., 2023; Schaye et al., 2023). However, this thesis will only focus on dark matter (gravity-only) simulations.

Over the past few decades, advances in computational power have significantly enhanced the scale and resolution of simulations, enabling the modeling of increasingly larger volumes of the universe with higher precision and resolution (Angulo and Hahn, 2022). This allows researchers to probe the distribution of matter on scales ranging from individual galaxies to almost the entire observable universe. As more and more observational data is produced by large-scale dark matter surveys, such as DESI (DESI Collaboration, Aghamousa et al., 2016) or Euclid (Euclid Collaboration et al., 2022), it is more important than ever to have accurate computer simulations available. By comparing the actual observations with the simulations, it is possible to refine cosmological theories and analyze different structure formation processes.

In particular, simulations play a vital role in understanding the non-linear evolution of cosmic structures. While linear perturbation theory can describe the early universe, the growth of structures, such as dark matter haloes or galaxies, requires solving complex nonlinear equations that cannot be done analytically (Bernardeau et al., 2002). This is where simulations excel, providing a numerical approach to studying the impact of these nonlinearities on the observable universe.

1.2.1 Initial Conditions

In order to simulate realistic matter collapse and structure formation, it is paramount to start with accurate initial conditions (ICs). The prevalent theory predicts that the growth of structure started from a Gaussian random distribution of the initial density perturbations. It is motivated by inflationary cosmology and in agreement with the large-scale distribution of galaxies, as well as observations of the CMB (Planck Collaboration, 2016). Therefore, the field can be fully described by the power spectrum $P(k)$, which in turn depends on the cosmological parameters and the model. This means that the post-recombination density field can be expressed as a convolution of the primordial density field with a transfer function $T(k)$. The power spectrum can be written as $P(k) = \alpha k^{n_s} |T(k)|^2$ (Bertschinger, 2001), where α is a normalization constant and n_s

is the (constant) power spectrum index after inflation. In most cases, initial conditions are generated on a Lagrangian grid $\vec{q}$ for computational efficiency, the details of which will be discussed in the following sections. First, the distribution $\hat{\delta}_k$ is obtained by taking a set of complex numbers with uniformly distributed phases ϕ and a Rayleigh-distributed amplitude, where the variance is given by the desired spectrum (Bardeen et al., 1986). Then, the potential $\Phi(\vec{q})$ needs to be generated in real space by Fourier transformation, where the relation is governed by the Poisson equation, $\nabla^2 \Phi \propto \delta$:

$$\Phi(\vec{q}) = - \sum_{\vec{k}} \frac{4\pi G a^2 \bar{\rho}}{k^2} \hat{\delta}_k e^{i\vec{k} \cdot \vec{q}}, \quad (1.8)$$

where G is the gravitational constant, a the scale factor, and $\bar{\rho}$ the mean background density. Since the universe is initially well-described by linear theory, full N-body calculations do not need to be used from the very beginning (Big Bang) but rather from a redshift where the density fluctuations are still small enough. The particle positions $\vec{x}$ and velocities $\vec{v}$ up to that redshift are computed by applying the Zeldovich approximation (Zel'dovich, 1970):

$$\vec{x}(q, z) = \vec{q} - D(z)\Phi(\vec{q}) \quad (1.9)$$

$$\vec{v}(q, z) = \frac{dD(z)}{dt} \nabla \Phi(\vec{q}) \quad (1.10)$$

where $D(z)$ is the linear growth factor at redshift z (Efstathiou et al., 1985). $\vec{x}$ and $\vec{q}$ are often referred to as comoving Eulerian coordinates and Lagrangian coordinates, respectively. The Zeldovich approximation is justified in the early universe when density fluctuations are still in the linear regime. It provides a computationally efficient way to generate initial conditions without requiring a full nonlinear computation from the start (Crocce et al., 2006). However, as simulations increased their resolution, it became apparent that initial conditions created with Zeldovich approximations (first order) exhibited transients that only decayed when choosing a sufficiently high starting redshift. In order to reduce this effect, one can instead create the initial conditions using second-order Lagrangian perturbation theory (2lpt), which has been shown to produce smaller and more rapidly decaying transients (Michaux et al., 2021).

Second-order Lagrangian Perturbation Theory

Lagrangian perturbation theory relates $\vec{x}$ and $\vec{q}$ in the following way:

$$\vec{x} = \vec{q} + \vec{\Psi} \quad (1.11)$$

where $\vec{\Psi}(\vec{q})$ is the displacement field, which can be rewritten as the gradient of a potential Φ . For second order, the potential is made up not only of the gravitational potential Φ , but also contains a second order potential due to components of the deformation tensor (e.g., Hahn and Abel, 2011). This leads to slightly different equations for the Eulerian positions and velocities:

$$\vec{x} = \vec{q} - D_1 \vec{\nabla}_q \Phi^{(1)} + D_2 \vec{\nabla}_q \Phi^{(2)} \quad (1.12)$$

1 Introduction

and

$$\vec{v} = -D_1 f_1 H \vec{\nabla}_q \Phi^{(1)} + D_2 f_2 H \vec{\nabla}_q \Phi^{(2)} \quad (1.13)$$

where $\Phi^{(1,2)}$ are the two potentials, D_1 is the linear growth factor and D_2 the second order given by $D_2 \approx -3D_1^2/7$, H_0 is the Hubble constant and $f_i = d \ln D_i / d \ln a$. The subscripts q indicate partial derivatives with respect to the Lagrangian coordinates.

It is possible to incorporate even higher-order terms, but the increase in accuracy is quite small, whereas the computational load increases significantly (Scoccimarro, 1998). However, it can be worthwhile depending on the applications (Michaux et al., 2021). Overall, the usage of 2lpt enables the simulations to calculate the initial density field at lower redshift while keeping high accuracy. This means that less steps are needed for the more intensive N-body calculations to reach the same final redshift.

1.2.2 Evolving Dark Matter Structures

The evolution of dark matter particles, which only interact through gravity and are therefore noncollisional, is described by the collisionless Boltzmann equation coupled with the Poisson equation. Each particle has three positions, three velocities, and one time parameter assigned. Consequently, a system containing N particles defines a $6N + 1$ dimensional phase space, in which the trajectory is the solution of that particular N-body problem. For many particles, the high dimensionality would lead to a very expensive computation, but at the same time, the two-body relaxation time increases. This is beneficial as simulations containing long relaxation times compared to the time frame of interest allow for a statistical description (Chacon et al., 2020). Using a mean-field description of the dynamical system, it is possible to rewrite it utilizing only one particle distribution function, which reduces the dimension to simply $6 + 1$. This process can be understood as a collisionless fluid under external forces rather than individual particles interacting with each other. The Vlasov equation describes this process:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \vec{\nabla}_x f - \vec{\nabla} \Phi \cdot \vec{\nabla}_v f = 0 \quad (1.14)$$

where $f(\vec{x}, \vec{v}, t)$ is the distribution function of the fluid as a function of position, velocity and time, and Φ is the gravitation potential. This equation is coupled with the Poisson equation:

$$\nabla^2 \Phi(\hat{x}, t) = 4\pi G a^2 [\rho(x, t) - \bar{\rho}(t)] \quad (1.15)$$

For a more complete derivation of these equations, see Dolag et al. (2008). In order to solve these equations and evolve the N-body system in time, one must discretize the continuous variables (position $\vec{x}$ and velocity $\vec{v}$) before applying numerical methods. While the traditional N-body approach discretizes the system into individual particles, alternative techniques exist, such as directly solving the Vlasov-Poisson equations in a 6+1D phase space (e.g., ColDICE by Sousbie and Colombi, 2016). Nonetheless, the following paragraphs will focus on discretized systems. More precisely, it will discuss

the different methods for force computations. Starting from just a basic summation of particle forces, these methods have been improved over the years to allow for faster and more efficient calculations.

Direct Sum

The direct sum method involves calculating the gravitational forces between all pairs of particles in an N-body system:

$$\Phi(\vec{x}) = -G \sum_j \frac{m_j}{(|\vec{x} - \vec{x}_j|^2 + \epsilon^2)^{\frac{1}{2}}} \quad (1.16)$$

where m_j is the mass of an individual particle, while $\vec{x}_j$ is the position. ϵ is the gravitational softening parameter that is needed to suppress the interaction between two particles at small scales. It limits the maximum relative velocity and prevents the particles from sticking together. It is usually set to $\approx 1/50$ of the mean inter-particle separation, but the exact value varies from one simulation to another (Zhang et al., 2019). The direct sum method is straightforward, though it is a bit more complicated to solve for periodic box conditions (see e.g., Angulo and Hahn, 2022). Nonetheless, it is arguably the most accurate, but computationally very expensive, with a complexity of $O(N^2)$. Due to the need to calculate interactions for every pair of particles, this method becomes impractical for large simulations. Despite its high computational cost, direct summing remains useful for small-scale problems or when high precision is required for specific interactions (Spurzem, 1999).

Tree Method

The tree code method uses hierarchical grouping by subdivision of space into cubic cells until each cell only contains a single particle. The force calculations happen at a cell level and go from small to increasingly larger cells (i.e., ‘walking the tree’). This means that volumes containing many particles will be subdivided quite often, while nearly empty space, which obviously has less effect on the majority of particles, contributes only on the highest level. The most common implementation of tree code is the Barnes-Hut algorithm (Barnes and Hut, 1986). It reduces the computational complexity of N-body simulations to $O(N \log(N))$, but increases long-range force errors, which can be controlled by an ‘opening angle’ parameter. Fast multipole (Greengard and Rokhlin, 1987) further pushes the complexity to $O(N)$. The tree code method balances accuracy and computational efficiency, making it suitable for large-scale cosmological simulations where interactions at different scales need to be resolved efficiently.

Particle Mesh

The Particle-Mesh (PM) method combines particle-based and grid-based approaches to calculate gravitational forces. It works by placing a mesh over the entire space and calculating the density from the particle positions. This density field is then Fourier transformed before solving the Poisson equation and obtaining the potential. By interpolating the derivatives of the potentials, one can assign the forces to each individual particle (Klypin and Holtzman, 1997). It is important to realize that each mesh cell can

1 Introduction

contain varying amounts of particles. The number of calculations for the PM method depends less on the amount of particles (N_p) and more on the amount of grid cells (N_g). The Fast Fourier Transform (FFT) is very efficient and reduces the complexity to $O(N_p + N_g \log(N_g))$. The PM method is very effective for large-scale simulations, since the physical space can be increased while keeping N_g constant. Furthermore, the same amount of physical space can be calculated at any arbitrary resolution. However, since the force calculations are limited by the grid size, this method is less accurate for small-scale interactions.

Combinations

Both the tree code and the Particle-Mesh method have clear advantages but also some downfalls. The universe is homogeneous on large enough scales, but smaller scales harbor a platitude of interesting interactions. These smaller scales are often not resolved in PM methods, which is where the next two options come in. In order to increase resolution at certain locations, it is possible to individually modify the grid size, which is called **Adaptive Mesh Refinement** (AMR). It works by solving the Poisson equation at low resolution and then progressively increasing the resolution in volumes where the density is higher (Kravtsov et al., 1997). Another option is to switch from the PM method to the tree code in these specific regions. This hybrid method is called **TreePM** and works by splitting the potential into long- and short-range contributions. The long range is calculated in the Fourier space via PM, while the short range part is solved in the real space by using the tree code method (Xu, 1995). Instead of using the tree code method as the solution for more dense regions, it is also possible to use the direct sum there, which is referred to as Particle-Particle-Particle-Mesh (**P³M**). There are even some further combinations to add the adaptive mesh size to the hybrid methods (e.g., Couchman, 1991). Overall, these combined methods help greatly with achieving higher accuracy across all scales while keeping computational cost low, making them ideal for cosmological simulations that need to capture both large-scale structure and fine details.

1.2.3 Differentiable Simulations

Differentiable simulations represent a modern approach to computational modeling, where the entire simulation pipeline is made differentiable with respect to its input parameters. This means that not only can the physical processes be simulated, but one can also compute gradients of output quantities with respect to initial conditions or physical parameters. This is incredibly helpful for measuring the exact effect certain parameter choices have on specific physical mechanisms. In order for a simulation to be differentiable, all functions should have a derivative (almost) everywhere. In other words, the functions used for the linear and non-linear evolution of dark matter particles need to be continuous and not contain any breaks (e.g., *if*-statement, etc.). Historically, gradients for certain processes were usually calculated by hand. By applying the chain rule and breaking down the entire cosmological simulation into a succession of elementary algorithmic

operations, it was possible to calculate gradients throughout the entire evolution (e.g., Jasche and Lavaux, 2019). Later, automatic differentiation packages took care of a lot of the manual work, making it more feasible to simulate larger and more complicated structure formation processes in a fully differentiable manner. In addition to having gradients for further statistical analysis, these newer codes often relied on GPU instead of CPU, which massively improved performance (e.g., Modi, Lanusse and Seljak, 2021). This shift was largely driven by the "machine learning revolution", which led to the development of differentiable programming frameworks optimized for GPUs. These frameworks not only enabled efficient gradient computations but also facilitated the construction of differentiable forward models, opening new avenues for cosmological simulations and inference.

As just mentioned, treating the simulation as a differentiable function that maps input parameters, such as initial conditions or physical constants, to output observables can help highlight the effect any single parameter has throughout the very complex nonlinear evolution. Furthermore, one can estimate how the final output would change if different initial parameters were chosen without having to actually run this modified simulation. Later in this thesis, these benefits will be used and analyzed in regard to the cosmological parameters and their effect on structures such as the cosmic web and dark matter haloes. The resulting derivatives have many more use cases. The creation of a Fisher matrix is one of them, for example, since this common task is based on gradients, which are automatically available in these simulations and do not have to be calculated separately (Lanzieri et al., 2023). It has also been shown before that there is more information in the data than current statistical approaches (e.g., N-point statistics) are able to extract, which can be refined using more complex Bayesian inference algorithms (such as Variational Inference (VI), Hamiltonian Monte Carlo (HMC), etc., see Seljak et al., 2017). These algorithms are usually impractical for nondifferentiable simulations, as they rely on derivatives of likelihoods with respect to their parameters. Automatic differentiation solves this hurdle and even enables the incorporation of newer and better data from current large-scale structure surveys such as the Rubin Observatory Legacy Survey of Space and Time (LSST, see LSST Science Collaboration et al., 2009), the Dark Energy Spectroscopic Instrument (DESI, see DESI Collaboration, Aghamousa et al., 2016) or Euclid (Euclid Collaboration et al., 2022). These surveys are probing the universe at unprecedented scales and make analyzing the data from the ground up more challenging and time-consuming. Having access to derivatives allows for a more efficient exploration of parameter space in a Bayesian framework, by leveraging gradients to navigate the posterior distribution effectively (Modi, Lanusse, Seljak et al., 2021). These reconstruction problems, in which physical parameters or initial conditions are inferred from final states, are another one of the main advantages of differentiable simulations. This capability significantly enhances the predictive power of simulations and their utility in scientific discovery.

Implementing differentiable simulations is technically challenging. It requires careful reformulation of traditional algorithms to ensure that they are differentiable and that

1 Introduction

their gradients can be computed efficiently. Automatic differentiation tools play a crucial role in this process, enabling the computation of gradients without manual derivation. We will use JAX for this purpose throughout the thesis.

JAX

JAX is a high-performance numerical computing library (Bradbury et al., 2018), that can automatically differentiate native Python and NumPy functions. It aims to transform and accelerate code, bringing powerful capabilities to scientific modeling. With JAX, traditional simulation algorithms can be seamlessly transformed into differentiable versions, allowing the computation of gradients with respect to input parameters. It contains reimplementations of packages such as `jax.numpy` and `jax.scipy` to easily facilitate these benefits. Furthermore, JAX uses the XLA library to compile the code for hardware optimization on GPUs and TPUs. By leveraging the parallel processing capabilities of modern computing architectures, the code becomes far more efficient when dealing with a large number of small calculations at once, as in the case of N-body simulations. The easy implementation makes complex large-scale simulations involving millions of particles much faster and easier to optimize. Apart from automatic differentiation and code transformations, there are also benefits such as Just in Time Compilation (JIT) and many more (see *ibid.* for an overview and further links).

Due to the many advantages and possible applications of JAX, a larger ecosystem of packages has started to form around it to improve specific implementations of this library, such as neural network libraries (e.g., `Flax` by Heek et al., 2024, or `Haiku` by Hennigan et al., 2020) or a gradient processing and optimization library (e.g., `Optax` by DeepMind et al., 2020). In order to specifically help with cosmological simulations, Campagne et al. (2023) developed the `jax-cosmo` library. It has access to many of the advantages of the main JAX library, like vectorized operations and automatic differentiation, while also implementing cosmology-specific functions. The aforementioned paper documents possible applications, such as power spectra or Fisher matrices calculations, as well as gradient-based inference methods.

In summary, JAX provides easy access to highly efficient and fully differentiable simulations. It enables efficient gradient-based optimization, facilitates the solving of inverse problems, and integrates seamlessly with modern computing hardware and machine learning frameworks. This makes it a valuable resource for both theoretical studies and practical applications across a wide range of scientific disciplines.

DISCO-DJ

Throughout this thesis, almost all N-body simulations are computed using the DISCO-DJ (**D**ifferentiable **S**imulations for **C**osmology - **D**one with **J**ax) software package (Hahn

and List, 2025). It features a linear Einstein-Boltzmann solver and nonlinear structure formation models such as n-th-order Lagrangian perturbation theory, a particle-mesh and a Tree-PM N-body code. In order to accommodate the ongoing observational missions, it will also feature a method to model observables and help maximize the information extraction. Furthermore, it contains multiple analysis methods including power spectra, cross-spectra, and more. As the code has not yet been officially released, there are no direct citations or links to the package.

1.3 Cosmic Structures

While the isotropy of the CMB indicates that the universe is very smooth on large scales, as postulated by the cosmological principle, smaller scales reveal many inhomogeneities. Planets, stars, galaxies, or other well-known astronomical objects show just how drastic the matter density can change on these relatively small scales. The formation of all structures is believed to originate from primordial density fluctuations, as seen in the cosmic microwave background (Spergel et al., 2007), which slowly grew over time through gravitational instability (Ciardi and Ferrara, 2005). At larger distances, the distribution of matter is much smoother, but given enough time, it also forms distinct structures. On the largest scales, the universe can be approximated by a handful of basic structures, known as the ‘cosmic web’ (Bond, Kofman et al., 1996). This filamentary structure is usually divided into four distinct types: clusters, filaments, sheets, and voids. These types were first predicted by Zel’dovich (1970) due to particles collapsing and creating distinct regions of varying densities. The theoretical work was later supported by early numerical simulations like Efstathiou et al. (1985), which reproduced this cosmic web. At around the same time, the first galaxy surveys also observed this structure (e.g., de Lapparent et al., 1986). Technological advancements over the years have helped push these surveys to better resolutions, and in turn reveal finer parts of this web structure (e.g., 2MASS Skrutskie et al., 2006).

In general, as the universe evolves, denser regions collapse faster and pull surrounding matter inward. In contrast, voids, which are the least dense structures, have very little attractive force and thus contain almost no galaxies. The little matter within slowly leaves these underdense volumes and flows into flattened planar-like regions, called sheets. These two-dimensional structures form the boundaries between cosmic voids and can span tens of mega-parsecs. Over time, the matter gets funneled further into cylindrical and elongated regions known as filaments. They contain far more matter and house many galaxies (Colberg et al., 2005), but are ultimately just another transitional region for the matter flow. The filaments connect the most dense structures throughout the universe, called clusters. These sphere-like regions are the prominent positions for large galaxy clusters. Even though these cosmic web environments are visible with baryonic matter when working with all sky surveys, the effects are even more pronounced in dark matter simulations.

1 Introduction

Dark matter, which interacts only through gravity, is more diffusely distributed throughout the universe (Colberg et al., 2005) and constitutes the backbone of this structure formation. Since it contains a larger fraction of the total mass, dark matter has to be the driving force for this cosmic web and be dispersed similarly. Due to its lack of electromagnetic interaction, it cannot be observed directly, but by studying these cosmic structures astronomers can infer properties of dark matter, dark energy, and other cosmological information. Thus, the cosmic web offers a powerful framework for understanding both the past and the future of cosmic evolution (Bonnaire et al., 2021). For this reason, the first part of this thesis will focus on the differentiable identification of these cosmic web environments.

The second part will focus on a different type of structure, which will be briefly introduced here. When going from the large-scale structures (clusters, filaments, sheets and voids) to smaller scales, gravitationally bound dark matter becomes decoupled from the Hubble expansion and collapses. Such a structure is called a dark matter halo. While invisible, dark matter haloes are indirectly detected through their gravitational effects on luminous matter, such as galaxies and galaxy clusters. This effect is known as gravitational lensing, as it bends the light from background objects similar to how a glass lens works (Massey et al., 2010). Dark matter haloes provide the gravitational scaffolding for the formation of galaxies and clusters, by providing a site where gas can cool and condense, leading to star formation and galaxy evolution. In almost all cases, the halo seems to radially extend outward further than the galaxies inhabiting them. The first evidence of dark matter in haloes was found through the analysis of rotational curves by Freeman (1970). By estimating the mass of baryonic matter in a galaxy, the maximum radial velocity for gravitationally bound stars could be calculated at different radii. Instead of following the predicted decrease at the outskirts of the galactic disk, the rotation curve seemed to flatten, which is only possible if there is more matter and it extends further outward. Hence, the existence of dark matter in galaxy haloes was substantiated. Years of research have shown that dark matter haloes are a fundamental structure in the evolution of the universe. They vary in size, from those hosting individual galaxies to much larger systems that encompass galaxy clusters, with typical masses ranging from about 10^{11} to $10^{15} M_{\odot}$ (Navarro et al., 1996).

2 Cosmic Web Environments with T-web

This first part of the thesis focuses on the cosmic web and the differentiable identification of its environments. This large-scale structure was first predicted by the Zeldovich approximation, and then reproduced by early numerical simulations, which did so in a discrete way, and were therefore not differentiable. Later, Hahn, Porciani et al. (2007) introduced a new classification method known as ‘T-web’ to identify four distinct environments: voids, sheets, filaments, and clusters. It is based on local gravitational dynamics, which are governed by the tidal field tensor, T_{ij} . The tensor is obtained from the Hessian of the gravitational potential ϕ generated by the matter distribution. In this framework, the cosmic web structures are determined by analyzing the eigenvalues of the tidal field tensor at each point in space. This approach allows for a differentiable classification of these four environments.

The gravitational potential ϕ is related to the matter density field δ through Poisson’s equation:

$$\nabla^2 \phi = 4\pi G a^2 \bar{\rho} \delta, \quad (2.1)$$

where $\bar{\rho}$ is the mean mass density of the universe. By solving Poisson’s equation for the gravitational potential, one can obtain the tidal field tensor

$$T_{ij} = \partial_i \partial_j \phi, \quad (2.2)$$

which describes how the gravitational potential deforms in different spatial directions. The tensor is symmetric, ensuring that its eigenvalues are real, and it fully characterizes the local dynamics in the vicinity of a point in space.

Finally, to classify the cosmic web structures, the number of positive eigenvalues of the tidal field tensor are examined. This number determines the stability of a point with respect to the surrounding gravitational potential. Similarly, the eigenvalues define the dimensions of stable and unstable manifolds, which leads to the following environments:

- *No positive eigenvalues* mean that there are no stable orbits in the region, so all particles spread further apart, leading to a **void**.
- *One positive eigenvalue* is stable along one axis (i.e., contracts along one axis, while expanding along the other two) and forms a **sheet**.
- *Two positive eigenvalues* make a **filament**.

- *All three positive eigenvalues* mean that the region is contracting along all axes and therefore leads to a **cluster**.

Thus, the number of positive eigenvalues provides a direct classification of the local cosmic environment.

2.1 Methods

The cosmic web classification with T-web is achieved in Python using only the packages for `jax.numpy` and DISCO-DJ (including any dependencies). We implemented it directly into the simulation workflow in the following manner. First, the initial conditions (ICs) are generated with DISCO-DJ and evolved to a specified redshift. For the ICs, the gravitational potential ϕ and the matter density field δ , as well as their Fourier transformed counterparts ($\hat{\phi}$ and $\hat{\delta}$), are already stored and can easily be retrieved as a variable. After evolving the simulation using either the particle-mesh N-body code or Lagrangian perturbation theory, these variables are not automatically calculated and have to be derived from the particle coordinates. Luckily, DISCO-DJ contains a function to transform the positions into a density field with `dj.get_delta_from_pos(X_sim)`, which is based on cloud-in-cell interpolation.

At this point, it should be noted that, unlike the original implementation of the T-web classification, we apply no additional smoothing to the density field or the gravitational potential. This choice leaves the length-scale over which the stability of structures is determined completely free.

Next, we apply the Fourier transformation using the real fast Fourier transform `jnp.fft.rfftn`. In order to calculate the gravitational potential, we divide the density field in Fourier space element-wise by the wavenumber components $-k^2$, which effectively applies the inverse of the Laplacian operator in Fourier space. This operation follows directly from Poisson's equation 2.1, where the Laplacian in the Fourier space becomes a multiplication by $-k^2$. By dividing the Fourier transformed density field $\hat{\delta}$ by k^2 , we recover the gravitational potential $\hat{\phi}$ in Fourier space.

To construct the tidal field tensor T_{ij} in real space, it is necessary to calculate the partial second derivative of the gravitational potential (see eq. 2.2). In a similar fashion to above, this can be simplified by staying in the Fourier space, where the equivalent calculation is achieved through the multiplication of the wavenumber components $-k_i k_j$ with $\hat{\phi}$. The wavenumber components exist for each dimension ($k_x k_y k_z$ for the normal 3D case). we use the different combinations of these to construct all parts of a Hessian matrix

$$\widehat{T}_{ij} = \begin{pmatrix} -k_x^2 \hat{\phi} & -k_x k_y \hat{\phi} & -k_x k_z \hat{\phi} \\ -k_y k_x \hat{\phi} & -k_y^2 \hat{\phi} & -k_y k_z \hat{\phi} \\ -k_z k_x \hat{\phi} & -k_z k_y \hat{\phi} & -k_z^2 \hat{\phi} \end{pmatrix}.$$

It should be noted that the matrix is symmetric, so $-k_x k_y \hat{\phi} = -k_y k_x \hat{\phi}$.

Then, we transform each component back to real space using the inverse real fast Fourier transform (`irfft`) to obtain the tidal field tensor T_{ij} . The next step is to calculate the eigenvalues of the tensor at each point, which describe the local deformation of the gravitational field. We implement this using the `jax.numpy.linalg.eigvalsh` function, which efficiently computes the eigenvalues for symmetric matrices, ensuring that the result is real.

Finally, the classification of the cosmic web structures is based on the number of positive or negative eigenvalues. Specifically, we examine how many of the eigenvalues are positive at each point, the particles can be classified into one of the four cosmic structures: voids, sheets, filaments, and clusters (see 2 for the corresponding counts).

This methodology allows for a differentiable classification of the large-scale cosmic web structures. However, before implementing this classification method into an evolved simulation, it is better to test it on a matter density field with a known distribution. In this way, the results can be checked for potential errors. This is the case for a Gaussian random field, where the fractions of the environments have been calculated. Theoretically, sheets and filaments should each make up 42% of the total volume, while voids and clusters occupy 8% (Doroshkevich, 1970).

2.1.1 Classification on Initial Conditions

Since the T-web formalism for structure classification is verifiable, when using the initial conditions of a simulation as input, the code is first tested on the unperturbed density field, before moving on to evolved simulations.

2D Case

In order to further reduce memory and computation time of the first implementation, the 2D simulation of DISCO-DJ is used. The cosmological parameters and initial scale factor are left unchanged and taken from the example notebook.

- $\Omega_c = 0.8$
- $\Omega_b = 0.2$
- $\Omega_k = 0.0$
- $h = 0.6774$
- $n_s = -1.9$

2 Cosmic Web Environments with T-web

- $\sigma_8 = 0.8$
- $w_0 = -1.0$
- $w_a = 0.0$
- $a_{initial} = 5.0$

It is important to mention that the parameters vary quite a bit from the standard Λ CDM model, but this is due to the lower dimensions of a 2D simulation. Since an evolved simulation is not needed, the calculations can already be performed after creating the initial conditions. As mentioned above, both the density field δ and the potential ϕ are automatically calculated in this unevolved state. The method is the same for both these fields, but they are usually only applied to ϕ . Nonetheless, δ will be used as an additional check, as it contains more power on small scales (due to $\hat{\delta} = -k^2\hat{\phi}$). Here is a short overview of the procedure for the simplified case on the initial conditions:

1. The tidal tensor is calculated in Fourier space

$$\widehat{T}_{ij} = \partial_i \partial_j \hat{\phi}$$
2. It is transformed into real space
3. The eigenvalues are calculated
4. Each point is classified by the amount of positive eigenvalues

Since the tidal tensor is symmetrical, the resulting eigenvalues are all real. Furthermore, because of the `jax.numpy.linalg.eigvalsh` function, the eigenvalues are already sorted from the smallest to the biggest value. This makes counting the different possible cases even easier. When working in 2D, there are only two eigenvalues per particle, which results in just three cases: both negative, mixed, or both positive.

Figure 2.1 shows the first implementation of the T-web formalism, but since the density is randomly distributed under these initial conditions, larger connected structures cannot be classified. In contrast, Figure 2.2 shows a bit more information. The conditions are at the same initial time, but since the potential is smoother than the density, the parameters used to create this 2D simulation lead to a more clumped potential on these scales. After applying the T-web method to classify these structures, the regions of lower potential show a higher density of voids (without positive eigenvalues), while the higher potential has more clusters (purely positive eigenvalues).

This shows that the method works visually as expected. The statistical distribution of different structure types is checked in 3D to be consistent with the literature values.

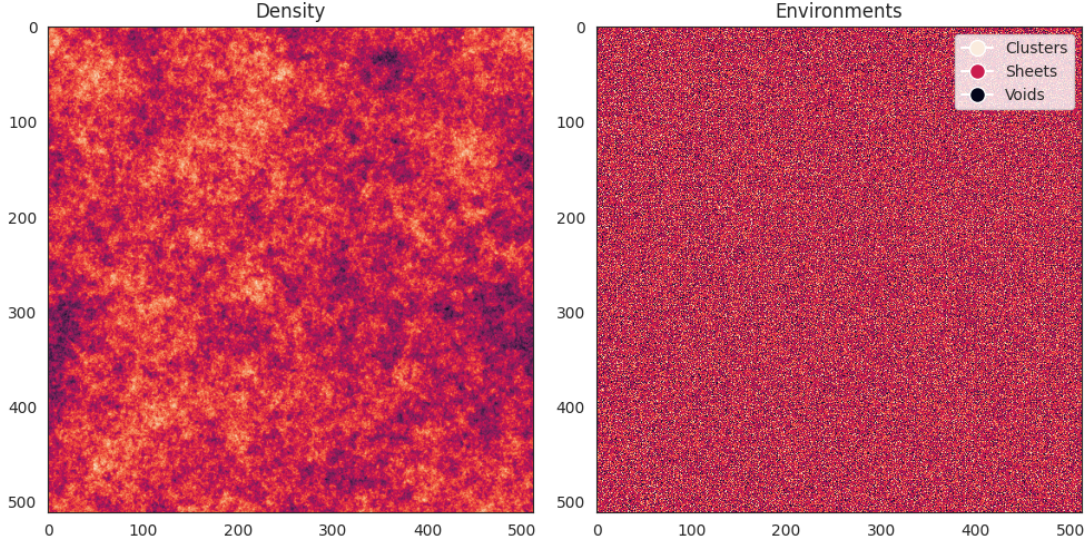


Fig. 2.1. T-web structure classification on 2D initial conditions. Density field (left) where brighter colors represent regions of higher density. Eigenvalue environment of each particle for the same field (right). Since the density of the initial conditions is Gaussian distributed, no large-scale structures of environments can be identified yet.

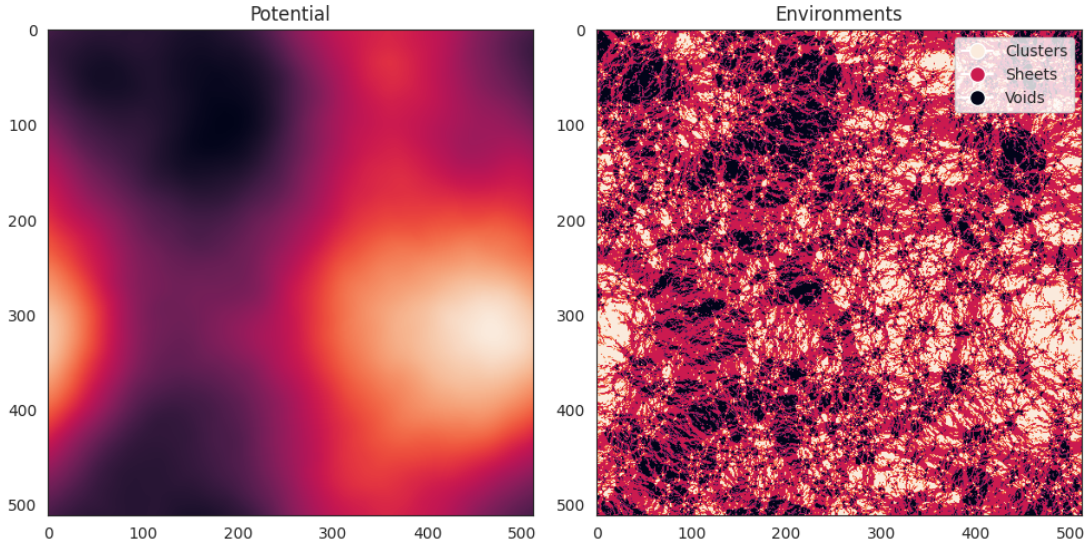


Fig. 2.2. T-web structure classification on 2D initial conditions. Gravitational potential (left) where brighter colors represent a stronger potential. Eigenvalue environment of each particle for the same field (right). Due to the chosen cosmological parameters, the potential has less variance over small scales, which leads to a smoother field. This is also evident in the eigenvalue plot, where regions with a higher potential correspond to more points with purely positive eigenvalues (i.e., these regions will collapse and form clusters).

3D Case

Having qualitatively shown that this method for classifying structure types works in two dimensions, the next step is to apply the T-web formalism to initial conditions in a three-dimensional space. Working in full-dimensional space means that the cosmological parameters do not have to differ as much from the current Λ CDM predictions as they did in the 2D case. DISCO-DJ has some of the most used parameter sets already stored as dictionaries in the code, so they can be chosen more easily. In this first run, the CAMELS parameters are used (Villaescusa-Navarro, Anglés-Alcázar et al., 2021).

- $\Omega_c = 0.3 - 0.049$
- $\Omega_b = 0.049$
- $\Omega_k = 0.0$
- $h = 0.6711$
- $n_s = 0.9624$
- $\sigma_8 = 0.8$
- $w_0 = -1.0$
- $w_a = 0.0$
- $a_{initial} = 0.02$

After generating the initial conditions, the classification procedure stays the same. The additional dimension leads to an additional class, resulting in the four distinct cases (voids, sheets, filaments, and clusters), as described before. In order to better visualize the results, the simulation is cut into slices, and only one of these slices is randomly chosen and shown. The same slice is used for the density and the gravitational potential in order to remain consistent.

When comparing the figures 2.3 and 2.4 with each other, it is clear that the potential ϕ is smoother on large scales than the density δ . The difference between ϕ and δ comes from the fact that the density is the second derivative of the potential, which leads to an additional factor of $-k^2$ when Fourier transformed. This corresponds to a lower spatial variance on small scales. Therefore, the classification works better on the potential than on the density field. Since density fluctuations are much stronger on small scales, the T-web formalism struggles to identify clear structures in the early simulation state. In contrast, the gravitational potential exhibits more distinct maxima and minima due to weaker fluctuations on the same scales. While these differences are reduced in late-time simulations, applying the T-web formalism to the gravitational potential remains the

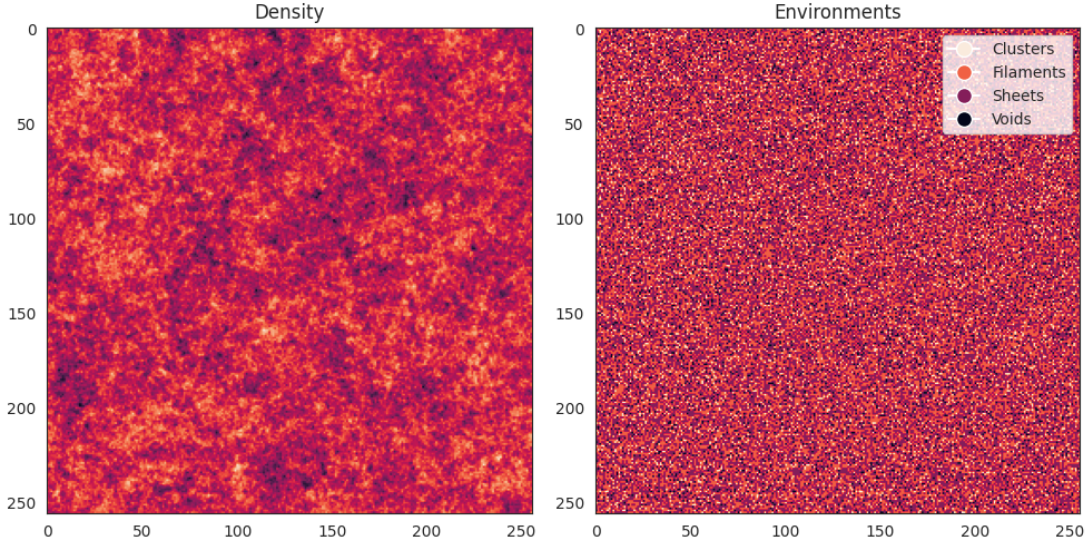


Fig. 2.3. T-web structure classification on full 3D initial conditions. Density field slice (left) where brighter colors represent regions of higher density. Eigenvalue environment of each particle for the same field (right). Since the density in the initial conditions is Gaussian distributed, no large-scale structures of environments can be identified yet.

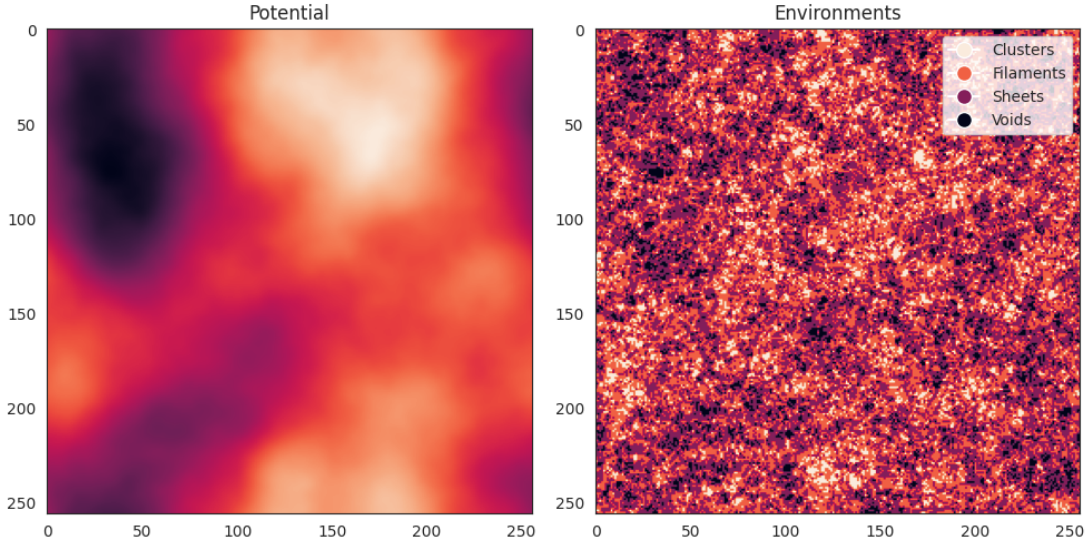


Fig. 2.4. T-web structure classification on full 3D initial conditions. Gravitational potential slice (left) where brighter colors represent a stronger potential. Eigenvalue environment of each particle for the same field (right). The generated initial conditions show clear regions of high and low potential, but it does not correspond directly to the cosmic environments. However, regions of higher potential seem to also show more clusters (white), while the opposite is true for voids (black).

better approach.

So far, these results have only been examined qualitatively. In order to compare them quantitatively, the fraction each structure type occupies in the entire three-dimensional initial conditions is calculated. In the case of the density field, the values are 8.10% for clusters, 41.91% for sheets, 41.90% for filaments, and 8.09% for voids. The potential has similar, but slightly different, values of 7.97% for clusters, 42.04% for sheets, 42.02% for filaments, and 7.97% for voids. As discussed above, ϕ is smoother than δ due to a change in variance over different scales, which therefore has an effect on the distribution in a finite simulation box. For infinitely large fields, the volume fractions of these structure types have been predicted by Doroshkevich (1970) to be 8.00% for clusters and voids and 42.00% for sheets and filaments. In order to further minimize the discrepancy, the fractions are calculated for many different initial conditions and then averaged out. This is equivalent to a larger box size while being much more resource efficient. The following values are the results of 100 separate ICs:

1. Clusters: 7.99%
2. Sheets: 42.01%
3. Filaments: 42.01%
4. Voids: 7.99%

These values are already very close to the predicted volume fractions, and could be further improved with more simulations, or a larger box size. A different approach to change these fractions in a finite box size is to vary the power spectrum of the simulation, since it governs the scale of the variance.

For high power in large scales, it is possible to get a simulation box that contains a far greater fraction of a specific environment than the predicted values for an infinitely large box. This can be achieved by reducing the cosmological parameter $n_s = 0.9624 - 4$. By decreasing the value of the spectral index, the spectrum shifts power from small to large scales, which leads to a more smoothed-out density field on larger scales and increases the cosmic variance. The effect of this is even more evident on the already smoother potential, as can be seen in Figure 2.5. It should be noted that this results in worse volume fractions, but it demonstrates the effect of n_s on the cosmic web environments more clearly. However, one can imagine that increasing this parameter would have the opposite effect and further improve the fractions, especially in combination with multiple simulations. Since this is not necessary, as the method has already been verified enough, we will instead finally move on to applying this classification on evolved simulations.

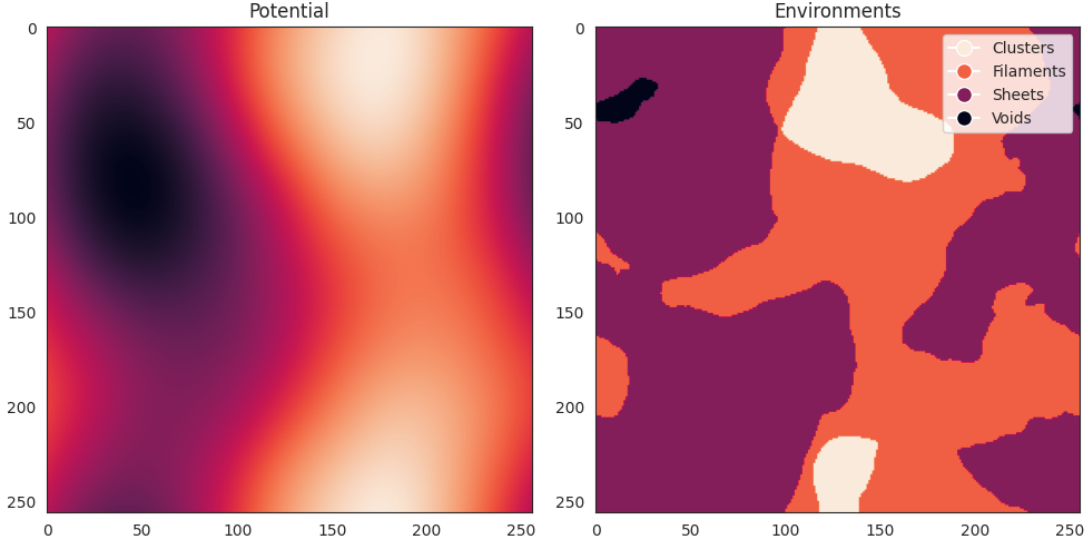


Fig. 2.5. T-web structure classification on full 3D initial conditions with a decreased value for the spectral index ($n_s = 0.9624 - 4$). Gravitational potential (left) and T-web structure classification (right). The smoothed out potential leads to distinct regions for the environments.

2.2 Results

Applying the T-web formalism to the initial conditions is important, since they are Gaussian, so the outcome is already known. As has been shown above, the resulting volume fractions align with the theory, proving that the implementation has been successful. Therefore, it is finally time to apply the same procedure to evolved simulations (where the nonlinear structure formation has led to non-Gaussian fields) and examine the differentiability aspect of this environment classification method.

2.2.1 Classification on evolved Simulations

There are two options for generating late-time simulations within the DISCO-DJ package. The first is done by using Lagrangian Perturbation Theory (LPT). It is faster and less computationally expensive than the full N-body calculations but also less accurate. For high redshifts, the error in second-order LPT is negligible, but it starts to accumulate at later times.

The method for classifying the structures in evolved fields is basically indifferent to the classification of initial conditions. However, the density field and gravitational potential have to be calculated manually from the positions of the evolved particles (see 2.1) before building the Hessian and counting its positive eigenvalues. Noticeably, the fields

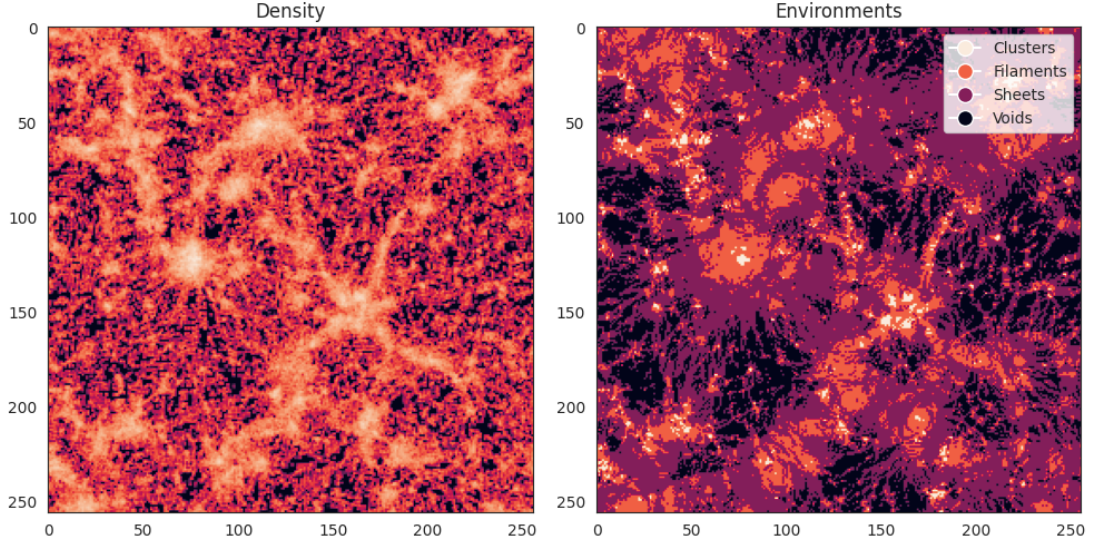


Fig. 2.6. T-web structure classification on a 3D simulation that was evolved to $z=0$ with **Lagrangian Perturbation Theory**. 2D slice of the density field (left) and the same slice for the eigenvalue environments (right). The density field was converted to a potential before applying the structure classification to improve the results.

of late-time simulations have a typical cosmic web look due to the fact that the dark matter particles had time to group together and form connected structures. The T-web environments follow these structures quite clearly. Figure 2.6 shows an LPT-evolved simulation at redshift $z=0$. It shows some regions having collapsed, with the peaks corresponding to cluster environments (purely positive eigenvalues) according to the T-web formalism. They are surrounded by filaments, which in turn are enclosed within sheets. The least dense regions are populated by voids (purely negative eigenvalues).

In order to compare the effects of the simulation method, the exact same cosmological parameters and even the seed for the initial conditions are left unchanged. However, this time the simulation is evolved with N-body PM calculations for 100 steps until redshift 0. By using a particle mesh it is possible to run the full N-body simulation in a computationally efficient way, but the runtime is still slightly longer than pure LPT. Figure 2.7 shows the results and it is immediately clear that the N-body method produces quite different density fields at late times. The T-web environments are not as obvious, but the clusters are much more concentrated. The density is more collapsed into individual strands, which produce a more pronounced cosmic web. In contrast, the less accurate LPT is far smoother, showing how the errors become too large above a certain redshift.

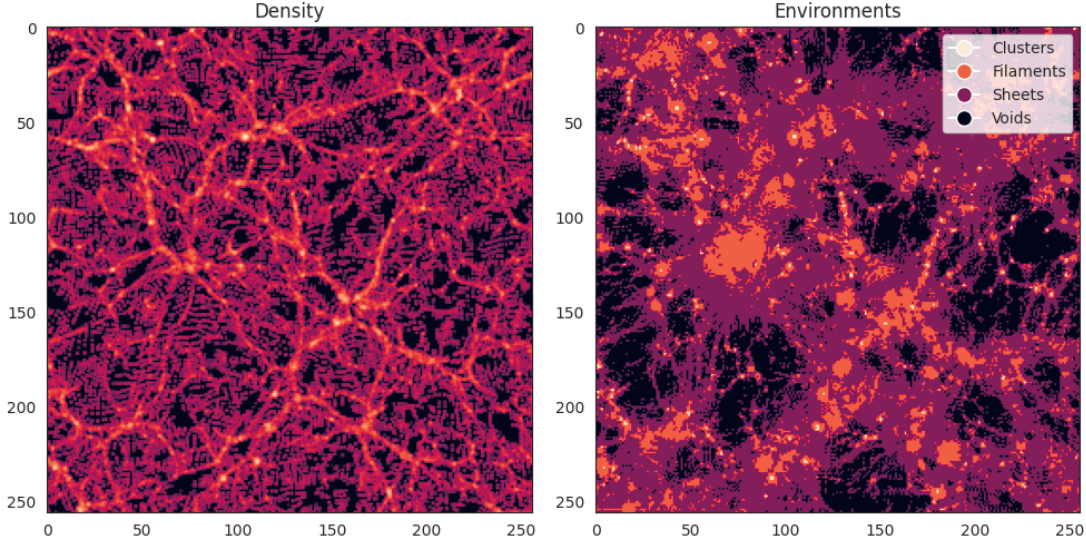


Fig. 2.7. T-web structure classification on a 3D simulation that was evolved to $z=0$ with **N-body**. 2D slice of the density field (left) and the same slice for the eigenvalue environments (right). The density field was converted to a potential before applying the structure classification to improve the results.

2.2.2 Differentiability

Since DISCO-DJ is able to calculate the density field in a fully differentiable manner, it is possible to compute the derivative through the entire simulation. Additionally, the T-web structure classification was purposefully written in a differentiable way, which makes it possible to differentiate the environments with respect to any input (such as cosmological parameters). This is accomplished using JAX by defining a function f that contains all the code and applying the `jax.jacfwd(f)` function to it. It evaluates the Jacobian using the forward-mode of the automatic differentiation. The result has many applications; one of them is a very straightforward method to verify the calculated derivative against finite differences. This next part explores this method and shows some mostly qualitative results. The application involves simulating the environments for a specific cosmology and using the derivative to directly calculate the environments for a slightly different cosmology. This allows the change to be added linearly into the base simulation, eliminating the need to rerun the entire process from the beginning.

$$\text{EigenValue}_{i,\text{new}} = \text{EV}_{i,\text{fiducial}} + \frac{\delta \text{EV}_i}{\delta \theta} \Delta \theta, \quad (2.3)$$

where θ represents any cosmological parameter. At this point, it should be noted that even though the N-body simulation and the eigenvalue (EV) calculations of the Hessian are fully differentiable, classification into different environments is not, since it involves discrete decisions. The derivative with respect to any of the cosmological parameters can

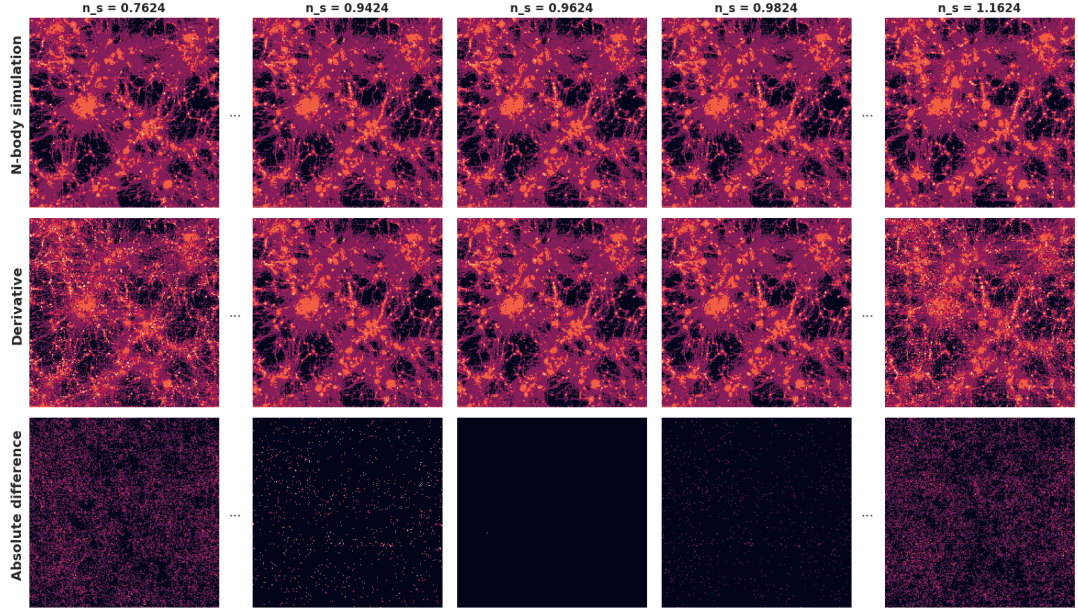


Fig. 2.8. T-web structure classification for different values of n_s . The first row contains the full simulation and structure classification from initial conditions with the adapted cosmological parameters. The second row is done in relation to the fiducial value of $n_s = 0.9624$ by adding $\frac{\delta EV}{\delta n_s} \Delta n_s$ to the reference eigenvalues. The last row shows the absolute difference between both methods.

only be calculated for the eigenvalues, and not the environments themselves. The actual classification has to be done separately after differentiating, as this part only allows for discrete classes. The manner of which does not change; the resulting eigenvalues are simply classified by the usual criteria of positive versus negative amount per particle. It is obviously not as accurate as rerunning the entire N-body simulation with the modified cosmological parameters, as the derivative is only correct for an infinitely small change. Higher-order terms would be needed to increase the accuracy, but this linear approach works well enough for small changes.

Figure 2.8 shows how adding the derivative linearly compares with the actual N-body simulation. The base simulation with fiducial cosmology is shown in the middle column, so the absolute difference is 0 in that case. As the parameter is increasingly varied in both directions, the absolute difference for both methods also increases. The environments do not change drastically for small changes in the spectral index n_s , but a slight trend is noticeable, especially when looking at the second row. As was shown before in Fig. 2.5 a lower value of n_s decreases the variance of the initial density field, which leads to a smoother potential. The resulting environments, therefore, become larger and more clumped together. This can be seen as an increase in both the voids (black) and the clusters (white) for the left most column. A larger spectral index leads to slightly more dispersed environments.

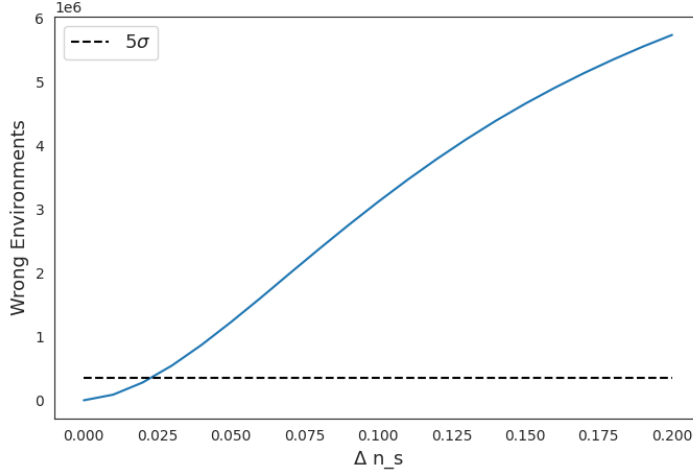


Fig. 2.9. The plot shows how many voxels out of the 256^3 simulation are misclassified for different values of n_s . For small changes in the spectral index, the derivative is predicting the behavior sufficiently well, but it starts to behave worse for larger differences. δn_s should not be larger than around 0.02 in order to stay below the dashed line indicating 5σ .

The derivative method is only a simple proof of concept to quickly visualize the effect of parameters, and the difference to the full N-body simulation becomes increasingly large for modified cosmology. Figure 2.9 shows how many environments are misclassified for increasing values of n_s . The dashed line provides the 5σ mark, where σ is measured as a percentage of misclassified voxels compared to the full simulation box that contains 256^3 particles. The derivative of the structure classification seems to capture the correct trend and works well enough for small changes, but after around $|\Delta n_s| = 0.2$ (which corresponds to the second and fourth columns in Fig. 2.8) the misclassification starts to increase rapidly. This is due to the simplification of a linear relation, which does not represent the actual process. Therefore, this method should not be used to compute the cosmological environments for changing parameters, especially for larger differences to the base simulation. However, the method is still useful for predicting the effect of changing a parameter on the outcome.

Figure 2.10 shows the same procedure as before, but for varying the parameter σ_8 instead of n_s . Furthermore, the slice shown is zoomed in to better compare a larger filament. σ_8 is the amplitude of the linear power spectrum on the scale of $8h^{-1}$ Mpc. In other words, a higher value increases the amplitude of density fluctuations, leading to larger and more massive structures forming earlier in the universe (Guo et al., 2013). This can be seen very well in the figure, as the first row shows the filament increasing in size for higher values of σ_8 . The second row, which approximates the structure classification in relation to the fiducial cosmology in the middle column, is able to capture this trend

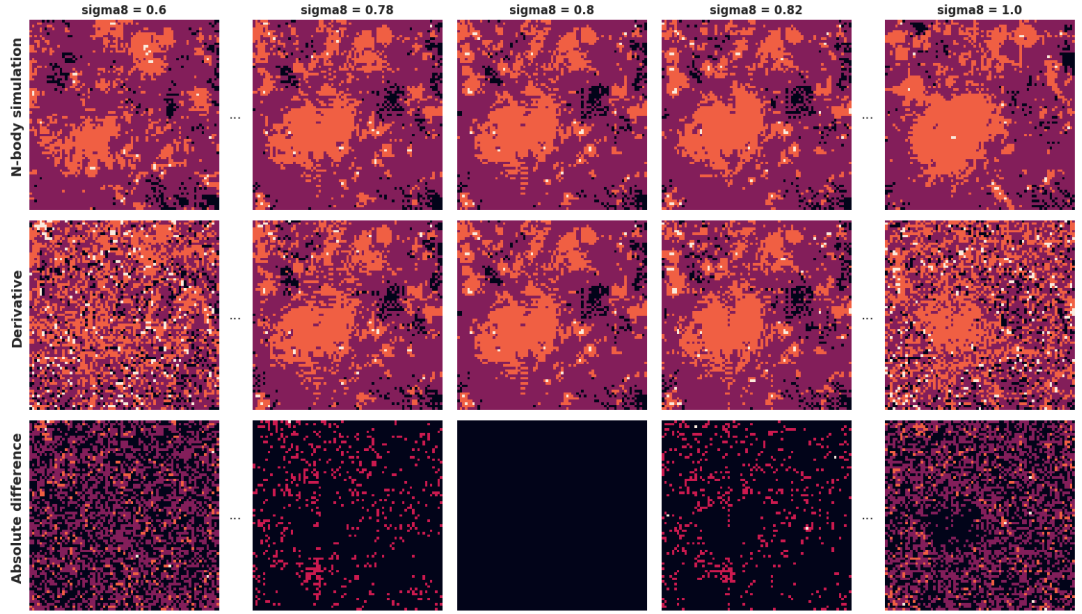


Fig. 2.10. Zoomed in slice of T-web structure classification for different values of σ_8 . The first row contains the full simulation and structure classification from initial conditions with the adapted cosmological parameters. The second row is done in relation to the fiducial value of $\sigma_8 = 0.8$ by adding $\frac{\delta EV}{\delta \sigma_8} \Delta \sigma_8$ to the reference eigenvalues. The last row shows the absolute difference between both methods.

for small changes (the second/fourth column). The method also shows the large filament changing size and shape in a similar manner to the full N-body simulation. The increase in number and size of clusters for the fourth column can also be reproduced from the derivative. In the end, however, the method breaks down again and is not able to classify almost any environments correctly. Nonetheless, it proves useful for very small variations in cosmological parameters, as well as showing the effects a change might have.

2.3 Further Applications

A differentiable structure classification method has many more applications. For example, by splitting the simulation into environments and calculating specific properties separately, it is possible to obtain tighter constraints for parameters of the Λ CDM model. This has been shown by Bonnaire et al. (2021), who used the classification of the structure with T-web to identify the four different environments before estimating the power spectra for each of them individually. The Fischer matrix is used to quantify the information content available from these statistics and compared to the total matter power spectrum. The filaments alone are able to provide better constraints for almost all pairs of cosmological (Ω_m , Ω_b , h , n_s , σ_8 , M_ν) and nuisance (λ_{th}) parameters. By combining the environment-based power spectra, Bonnaire et al. (ibid.) were able to improve their constraints by an average factor of 8.75; with n_s and M_ν showing the greatest improvement. Furthermore, by leveraging differentiable simulations and having direct access to the derivatives, Fisher forecasting for parameter constraints becomes viable and much more efficient (Campagne et al., 2023).

As mentioned in more detail in Section 1.2.3, differentiable simulations can also improve information extraction by using the automatically computed derivatives for Bayesian inference algorithms. One such application is Hamiltonian Monte Carlo (HMC), which utilizes gradient information to explore high-dimensional parameter spaces more efficiently (see Betancourt, 2017 for a full review). There are even more advanced methods that all necessitate derivatives for approximating posteriors, further reducing computation time and enabling scalability to very high dimensions. In that sense, differentiable simulations and cosmic web classification based on eigenvalues could be incorporated into these inference techniques. For instance, environment-specific summary statistics, such as those in Bonnaire et al. (2021), could be efficiently calculated and used for Fisher forecasts or as inputs for Bayesian inference models. This could potentially increase the accuracy of cosmological parameter constraints in an efficient way.

More research on these applications would be beneficial, particularly in combining differentiable environment classifications with modern Bayesian frameworks to tackle increasingly complex cosmological models. Ultimately, as this chapter has shown, differentiable structure classification not only performs computationally efficiently but also opens up new methods for deeper insight into the large-scale structure of the universe.

2 Cosmic Web Environments with T-web

This combination of advanced techniques sets a strong foundation for future explorations in both computational astrophysics and cosmological inference.

3 Dark Matter Haloes

The goal of this chapter is to develop a method for finding dark matter haloes in a differentiable manner. A way to do this is to use a convolutional neural network (CNN) to predict protohalo patches directly from the initial conditions. This kind of machine learning can be done completely differentiable as it maps the input to the output. The training data for the model will be supplied using the methodology shown in the next sections (a short summary of the final method will also be given in Section 3.2.4).

In contrast to the mathematically defined cosmic web environments, dark matter haloes have no clear boundary to distinguish them from the background. As previously mentioned, haloes are gravitationally bound regions that have decoupled from the Hubble expansion and collapsed (Wechsler and Tinker, 2018). Since they cannot be observed directly, analyzing their exact shape and size is basically impossible. Therefore, most studies stick to numerical simulations to investigate their physical properties. N-body simulations showed it was possible to use a universal density profile that only has one variable parameter for haloes of fixed mass: concentration, which quantifies how concentrated the density profile is towards the center (Navarro et al., 1997). The Navarro-Frenk-White (NFW) density profile parameterized the mass distribution inside the haloes. However, dark matter haloes have no clear outer boundary, as the density just continuously drops off towards the edges before reaching the mean density of the smooth background field. In order to compare haloes and produce statistically significant samples, it is necessary to use an objective and unambiguous method to define their radius and mass.

A common approach is to define the halo radius as a function of the critical overdensity, δ_c , which represents the density at which a region starts to collapse. For $\Omega_m = 1$, linear theory predicts uniform, spherical overdense regions to collapse at $\delta_c = 3/20 \cdot (12\pi)^{2/3} \approx 1.686$. They will then contract until they reach a dynamical, also known as virial, equilibrium. The virial theorem in combination with energy conservation predicts that this will happen at approximately $18\pi^2 \approx 178$ (Cole and Lacey, 1996). Different cosmologies and more complex formation processes require numerical solutions for the virial radius, r_{vir} . The most commonly used value is r_{200c} , which stands for the radius with a mean enclosed density 200 times that of the critical density of the universe at that redshift (Bryan and Norman, 1998). Similarly, the halo mass, $M_{200c} = \frac{4\pi}{3} 200 \rho r_{200c}^3$, is the mass enclosed within this radius and is often used to categorize haloes across a wide mass range. This formalism allows for a somewhat consistent comparison between haloes in simulations, despite their diffuse and complicated outer boundaries. Throughout the

3 Dark Matter Haloes

literature, there are also other definitions, so it is always important to compare the results with the same units. In addition to r_{vir} and r_{200c} , there is also r_{200m} , which is based on the mean density of the universe. Furthermore, a wide range of other factors in place of 200 are used frequently. We will use vir for radius and mass in the following sections.

Halo Finders

Even though the extent and physical properties of dark matter haloes have been discussed, identifying them in numerical simulations is still a challenge, but necessary to obtain labeled data for training the neural network. It is usually achieved through the use of ‘halo finders’. These algorithms are designed to detect collapsed, bound structures within large-scale cosmological simulations. There are several methods for identifying haloes, but the two main approaches include spherical overdensity (SO) finders and the Friends-of-Friends (FoF) algorithm. These two methods are the foundation for basically all halo finders today, with combinations and additions that make them better for certain applications. Overall, the algorithms usually work either by locating a density peak and collecting particles around it (SO approach), or by directly linking particles together based on specific criteria (FoF approach).

The classical SO method stems from a paper by Press and Schechter (1974), which describes the structure formation of dark matter haloes as a spherical collapse of matter around local density peaks. Therefore, these methods locate peaks and grow shells around them until the density drops below a chosen value, which is usually determined from a spherical top-hat collapse model. On the other hand, the FoF method (Davis et al., 1985) does not assume any specific shape for the haloes, but instead connects particles that are within a specified linking length (either in 3D space or even in 6D phase space). The proximity condition is similar to the density criterion, but has the main advantage of not assuming spherical symmetry. There are of course many variations for halo finders as well as hybrid approaches. For more information, see Knebe et al. (2013). The haloes in this thesis are identified using the ROCKSTAR halo finder (Behroozi et al., 2013), which is based on the FoF method.

Halo Mass Function

Once the dark matter haloes are found and their properties are measured, statistical analysis can be performed. The most important statistic for this thesis is the halo mass function (HMF):

$$\frac{dn}{d \ln M} = f(\sigma) \frac{\rho_m}{M} \frac{d \ln \sigma^{-1}}{d \ln M}, \quad (3.1)$$

where the HMF $\frac{dn}{d \ln M}$ is given as the number density of halos per logarithmic mass interval; $f(\sigma)$ is known as the halo multiplicity function, often given by the Press-Schechter, Sheth-Tormen, or Tinker form; ρ_m is the mean matter density of the universe; M is the halo mass; and $\sigma(M)$ is the variance of the density fluctuations on the mass scale M . This allows for a comparison between the results of halo finders and theoretical predictions.

The theory of these halo mass functions is founded in the analytical model, often called the Press-Schechter, excursion set approach. Press and Schechter (1974) theorized that the

overdense regions in the initial density field could be used to locate haloes at later times. By combining this assumption with a Gaussian distribution of the initial fluctuations, Bond, Cole et al. (1991) were able to derive a number density $n(m, z)$ of bound objects with mass m at time z . The assumptions and equations leading up to this mass function are explained in many publications (e.g., Mo et al., 2010). At this point, it should be mentioned that this analytical approach helped not only estimate the halo mass function, but also merging histories (e.g., Lacey and Cole, 1993) and clustering properties (e.g., Sheth and Tormen, 2002).

While the Press–Schechter, excursion set approach was very close to the results of numerical simulations, it slightly overestimated small haloes. Sheth, Mo et al. (2001) improved upon this by incorporating an ellipsoidal collapse instead of a spherical one. Later, Tinker et al. (2008) used a large set of simulations with different cosmological parameters, redshifts, box sizes, number of particles, and even distinct codes to further refine the HMF. However, with this modification, the halo mass function became largely an empirical fit to simulations. Newer HMFs are increasingly calibrated to numerical data or directly constructed as emulators (e.g., Bocquet et al., 2020). The results in this thesis are compared to the Tinker08 HMF.

3.1 Methods

This section will explain how to create labeled training data to predict protohalo patches from the initial density field. In order to fully understand the reasons behind the choices in this part of the work, this next section will cover the general pipeline, as well as specific changes to the membership criteria and N-body code. By visualizing the separate steps needed in chronological order, some problems become apparent and can be addressed to improve the final outcome.

The implementation of this dark matter halo finder starts off similarly to the cosmic web structure finding (Chapter 2), in that it also uses the DISCO-DJ package to generate the initial conditions (ICs) and evolves them to their final redshift. The simulation for this current section has a box size of 100 Mpc/ h and a resolution of 128 cubed. The ICs are created using the CAMELS cosmological parameters previously mentioned (see Section 2.1.1) at a redshift of 49. They are then evolved for 100 steps using particle-mesh N-body code until they reach $z = 0$.

DISCO-DJ does not contain its own halo finder, so the data has to be analyzed using a separate package such as ROCKSTAR. For this to work, the current simulation state - which includes the particle positions, velocities, and scale factor - has to be saved in an appropriate format, like the output format of the N-body code called GADGET-4, and exported as an hdf5 file. Thankfully, both these steps can be easily accomplished with the `dj.save_as_hdf5()` function. The resulting file is then processed by ROCKSTAR, which returns a table containing all haloes and subhaloes, as well as a number of their properties. The subhaloes are virialized structures that were found within a larger collapsed halo. At

3 Dark Matter Haloes

the current resolution and box size, there are around 7000 parent haloes and an additional 1000 subhaloes. For subsequent calculations, the subhaloes are excluded to ensure that each particle is unambiguously assigned to either the background or an individual halo. Including subhaloes would lead to ambiguity, as particles might otherwise be associated with both a subhalo and its parent halo. Furthermore, subhalo mass functions are usually steeper than their host/parent counterpart (Jiang and van den Bosch, 2014), due to different formation processes and tidal stripping, so they must be separated from the normal halo mass function. For the purposes of this thesis, the subhaloes are simply disregarded.

The next step uses the columns for the coordinates of the halo center as well as the virial radius. The haloes can then be overlayed on top of the density field. It should be noted that δ has to be computed from the particle positions using cloud-in-cell interpolation, as explained in the T-web Section 2.1.

The first results of this can be seen in Figure 3.1, which shows a random slice of the density field and all the haloes that appear in that slice. Since the haloes have collapsed significantly by $z = 0$, the markers are scaled up by a factor of 500 using the intersecting circle of the virial radius to illustrate their relative size in that slice. The marker colors are not indicative of any physical meaning, since they are just based on the halo id, which is the first column of the properties table calculated by ROCKSTAR. Visualizing the data in this way, while not necessarily needed, can help verify the calculations and conversion are done correctly.

The next step is to calculate the Halo Mass Function (HMF). In the case of simulation data, this step is done trivially by binning all halo masses and dividing their frequencies by the box size, which already represents n . The logarithmic width of the bin represents $\ln M$. The change in number density of haloes in comparison to their mass produces the HMF.

Figure 3.2 shows the results of these two calculation steps. Logarithmic bins are better for visualizing the HMF across the entire range of haloes in the simulation. It would be expected that the amount of haloes increases towards lower masses, as can be seen from the Tinker08 line. However, as the halo mass reaches the size of individual simulation particles, it underestimates the distribution. This is expected as the simulation is not able to produce any haloes with masses in-between these discrete values. Furthermore, the halo finder is not able to accurately identify gravitationally bound particles at these low numbers. For the current resolution and cosmology, the halo mass of $10^{11} M_{\odot}/h$ corresponds to the minimum of only 2 particles. As is apparent in the HMF plot, it only starts to match the theory at around $10^{12} M_{\odot}/h$ at the level of 75%. Therefore, ROCKSTAR is set up to discard any haloes with less than 20 particles for following calculations.

Further analysis of the detected haloes requires a closer look at the individual particles that make up each halo. ROCKSTAR saves this data in a binary field, which can be read using a Python package called `pynbody` (Pontzen et al., 2013) to extract all the relevant

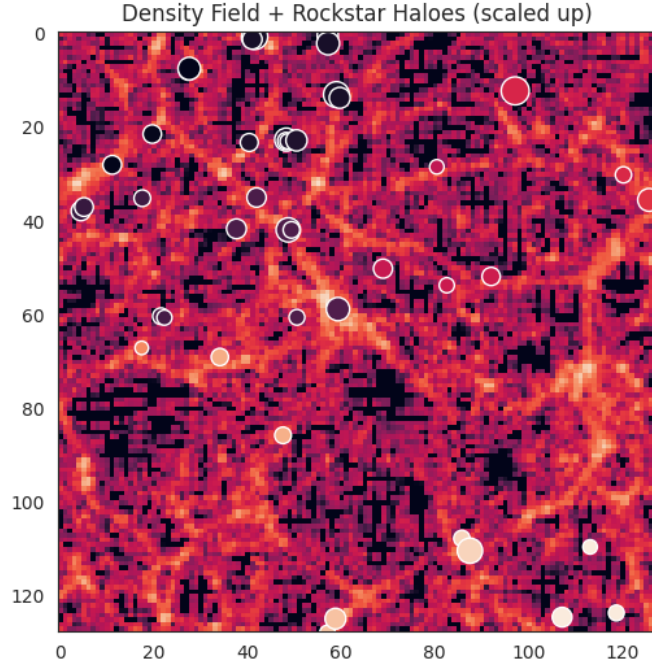


Fig. 3.1. 2D slice of the Density field, where brighter colors represent regions of higher density. The circles represent dark matter haloes. Their colors are based on the id - which is the order in which they were found by **ROCKSTAR** - so they have no physically significant meaning. The marker size is determined by calculating the radius of the spherical halo that intersects with the slice (i.e., halos whose centers lie farther from the slice appear with a smaller effective radius). In order to better see the haloes, the marker size is scaled up by a factor of 500.

3 Dark Matter Haloes

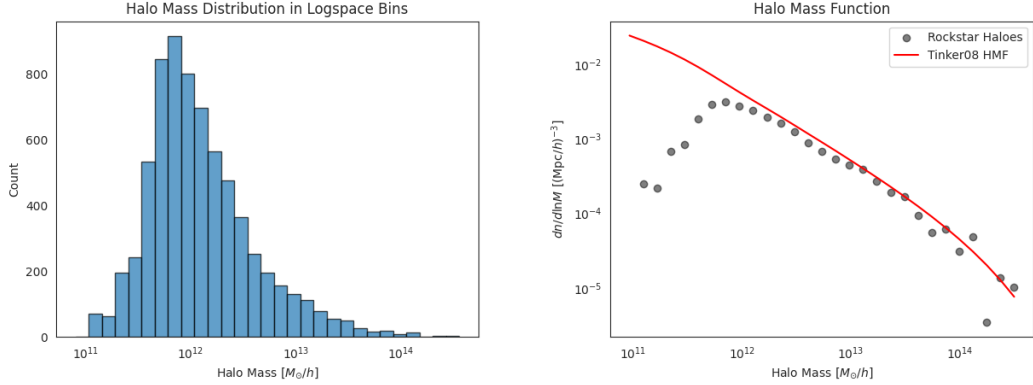


Fig. 3.2. These plots show the distribution of halo masses and compare them to theoretical functions. (Left) This histogram shows the amount of haloes found by **ROCKSTAR** for logarithmically spaced mass bins. (Right) This plot compares the HMF of the haloes from the simulation (gray points) to the analytical Tinker08 line (red line). It seems to fit quite well after around 10^{12} solar masses per h . The deviation at lower masses can be explained by the low resolution of the simulation.

information. It returns an object containing all the haloes and has the FoF particles directly assigned. By iteratively going through every halo and extracting the particles, it is possible to see which regions in the initial conditions ended up creating which haloes. These particles in Lagrangian coordinates are called protohalo patches.

The future halo particles during the initial conditions can be seen in Fig. 3.3 which shows both a 2D slice and the full 3D data cube. During the early times, the haloes inhabit a much larger fraction of the space and are far more extended. This is expected, as the variation in density during the initial conditions is far smaller than at late times, causing the same amount of mass to cover a larger area. Overall, the protohalo patches are relatively separated in the sense that they are not completely mixed up. Instead, they inhabit distinct regions for the most part. It is more difficult to see in the 2D case because the volumes might extend in different direction. Nonetheless, these plots are able to show that even though the formation of dark matter haloes is very complex, to a certain extent, it can be described by individual clouds of denser volumes collapsing into distinct haloes.

Figure 3.4 shows a few randomly selected haloes in the simulation box throughout the time evolution from $z = 49$ to $z = 0$. The final haloes are scaled up by a factor of 500 to better visualize them. Logically, larger protohalo patches form haloes with bigger radii. It also demonstrates that the relative position in the simulation box does not vary significantly in most cases. It should be noted that some protohalo regions might look like they are disconnected by very large distances (for example, the pink particles in the left plot), while in reality they are connected due to the periodic boundary conditions. Overall, the figure helps visualize the formation process during the simulation.

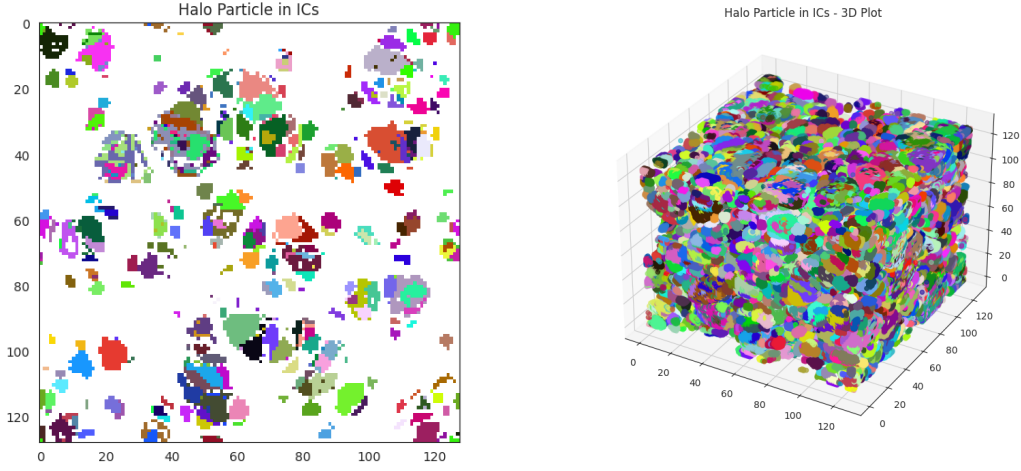


Fig. 3.3. These plots show the protohalo regions at $z = 49$. (Left) A 2D slice of the halo particles in the initial conditions. (Right) The full 3D simulation cube. Each halo has a randomly assigned color to distinguish them from one another. The particles that don't end up in any halo are left transparent.

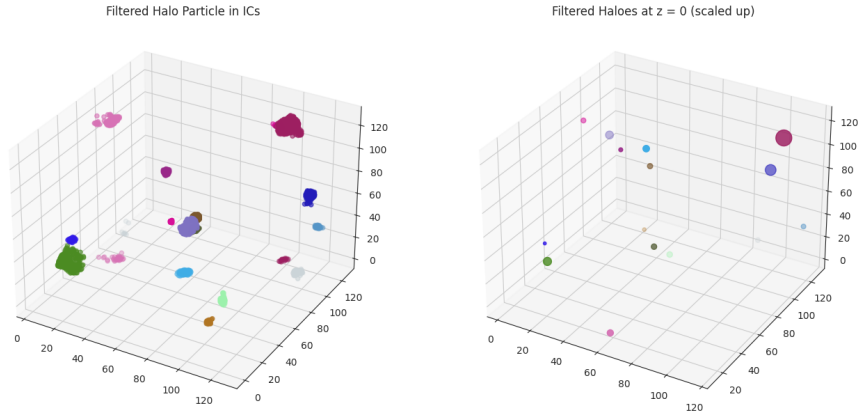


Fig. 3.4. A comparison of randomly selected haloes and their FoF assigned particles at different redshifts. (Left) The protohalo patches at $z = 49$. (Right) The virialized dark matter haloes at $z = 0$. The marker size represents the virial radius scaled up by a factor of 500. These two plots show the evolution from individual particles to gravitationally bound dark matter haloes throughout an entire simulation.

3 Dark Matter Haloes

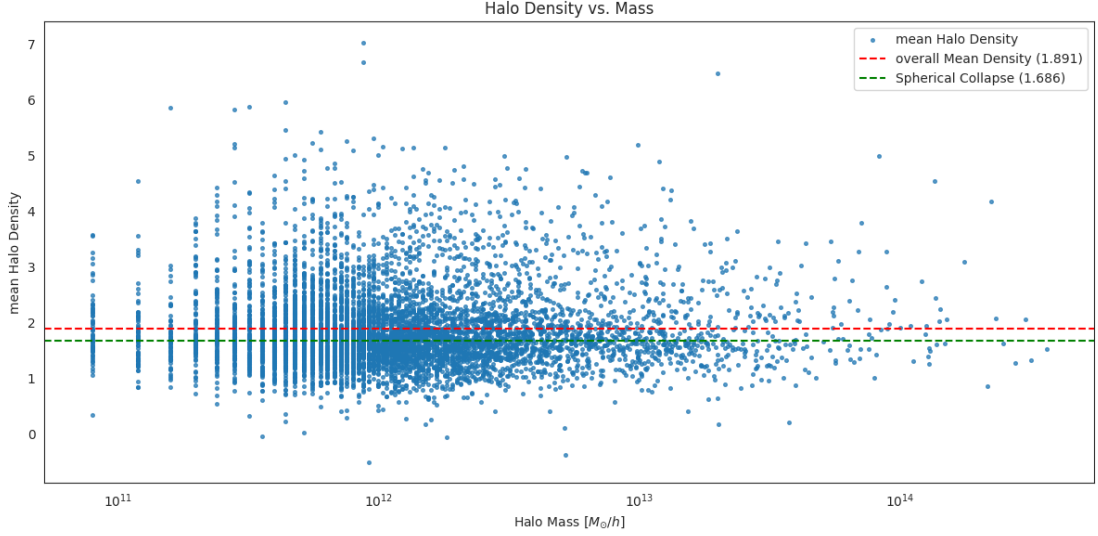


Fig. 3.5. The mean halo density in relation to the mass. The blue dots represent the mean FoF particles density for each halo. The red line is the mean value for all these blue dots, while the green line shows the theoretical value for spherically overdense regions to collapse. Even though the overall mean has a higher density, a lot of individual haloes are still below that value, which might have to do with their masses corresponding directly to the assigned Friends-of-Friends particles, instead of the final m_{vir} property.

Having access to the individual particles, or voxels (i.e., grid cells in the case of particle-mesh simulations) of the haloes also allows for quick analysis of the mean density for each protohalo patch. This is done by simply extracting the individual particles of each halo, summing up their values from the density field, and dividing it by the amount of particles. This is effectively the mean halo density.

The result of this step can be seen in Fig. 3.5, which shows the mean density for all 7000 parent haloes. The discrete vertical lines on the low-mass end are again a sign of the minimum mass for the simulation resolution. The left most line represents all haloes with only two particles, then three and so on, with no masses in between possible. As the number of particles increases, the relative difference becomes smaller and the discreteness effect seemingly disappears around $10^{12} M_{\odot}/h$, aligning with the results of Fig. 3.2. For this reason, it is important to have a minimum mass cutoff for haloes depending on the simulation resolution.

Even though the overall mean of the halo density is indeed above the theoretical value of 1.686 for spherical overdensity (see the beginning of this chapter for more explanation), the large number of data points that fall under the green line indicates that the member particles might not be well suited. This will be analyzed more closely in the following section.

3.1.1 Halo Particles

Taking a closer look at the binary file containing all halo particles for the ROCKSTAR halo finder shows that it stores only the FoF particles needed to identify a halo. This is problematic as the resulting properties are missing the unbinding process (Behroozi et al., 2013). To accurately compare the results, it is necessary to choose a suitable mass definition, such as m_{vir} or m_{200c} , which are more physically motivated and less sensitive to numerical effects than FoF. This is especially important for the halo mass function as the theoretical line shifts depending on the selected definition. It is interesting to note that Ondaro-Mallea et al. (2022) found that FoF mass function is near-universal, meaning it changes little with respect to growth rate and power spectrum shape. m_{vir} shows only slightly worse deviation, and performs much better than m_{200b} or m_{200c} . Due to a combination of these reasons, it was decided that all calculations and analysis will be done with the virial mass from now on. However, the halo finder does not provide a separate list of these particles, so a different approach is needed. As mentioned above, ROCKSTAR also returns a table containing some properties for each halo. Taking the center coordinates and the virial radius, it is possible to manually assign each particle to a host halo. However, one has to be careful with overlapping haloes.

A good example of this can be seen in Figure 3.6. If one were to simply assign each particle within the virial radius to a halo, the order in which the program runs would affect the final outcome. In the case of the image shown, the particles would be assigned to the pink halo first, but once the same criterion is checked for the green one, all the particles in the overlapping region would be switched to the second halo. This can be avoided by having a second criterion. The program initially calculates the distance of each particle to the center of a halo. If the distance is shorter than the radius, it is saved and assigned to the host. In the case where one particle already has a distance saved, it only overwrites it if the new value is smaller. This way, the particles are assigned to the halo with the closest proximity. This method does not take into account the relative halo size and, hence, the gravitational potential, but works well enough for these purposes. It should be noted that one has to include only parent haloes and filter out any subhalo beforehand. Furthermore, one has to take the periodic boundary conditions into account when calculating the distance to the halo, which can be accomplished by evaluating the absolute distance and subtracting it from the box size. The smaller value will be used as the true separation from the halo center:

$$\vec{d} = \min(|\vec{x} - \vec{x}_h|, L - |\vec{x} - \vec{x}_h|), \quad (3.2)$$

where $\vec{x}$ is the position of the particle, $\vec{x}_h$ the position of the halo center, and L is the box size. Furthermore, the program is sped up by parallelizing it with Numba (Lam et al., 2015).

Figure 3.7 shows the effect of these different member particle definitions for protohalo patches at $z = 49$. It is clearly visible that the FoF particles overestimate the mass

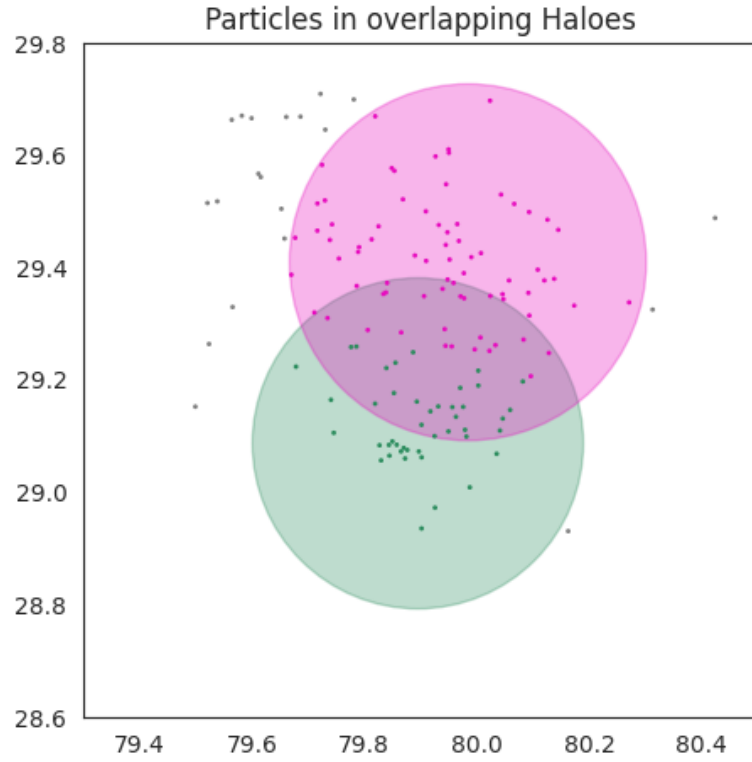


Fig. 3.6. A zoomed in slice of the haloes and particle coordinates at $z = 0$. It shows a strong case of two overlapping haloes and how the particles are assigned in such a case. The first criteria that is checked is if the particles fall inside the virial radius. If outside, they are not part of any halo (gray dots). The second criteria assigns the particles to the closest halo center, making sure that overlapping volumes are split up accordingly.

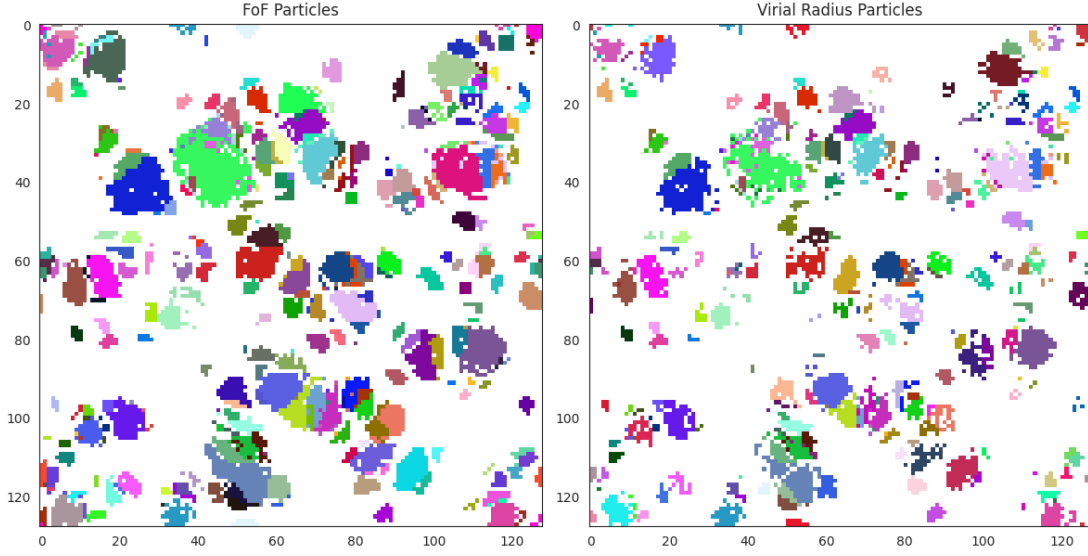


Fig. 3.7. This compares the same 2D slice of the protohalo patches at $z = 49$ for the different particle definitions. (Left) This plot shows the FoF particles that are stored in the binary file after ROCKSTAR has found all haloes. (Right) The same haloes, but only the particles that actually lie within the virial radius during $z = 0$. Each halo id is given a random color. It is immediately apparent that the FoF haloes are larger than the virial counterparts based on the halo finders definitions. Furthermore, the virial patches seem to be missing a lot of particles from within the protohalo regions.

3 Dark Matter Haloes

of haloes compared to their virial radius. The reason for this is likely the unbinding process by the halo finder. This is a step where the **ROCKSTAR** removes certain particles from each halo after it has identified them using the Friends-of-Friends method. The unbinding is based on certain criteria to better fit the physical formation process of each halo. The final halo properties, such as mass, position, radius, etc. are only calculated after this unbinding step. For this reason, it is necessary to rely more on the table of properties than on the member particles. The unbinding seems to mostly remove particles on the border of each patch, which makes sense since those are less likely to end up gravitationally bound compared to the center of a patch. However, it also removes some particles from inside, which is possible if the particles are perturbed enough to leave the halo completely. The complex formation process of these structures becomes more apparent in such cases and shows why the simple theory of spherical overdensity is not enough to explain dark matter haloes completely.

Definition	Min [$M_{\odot}/h$]	Max [$M_{\odot}/h$]
Virial	7.94×10^{10}	3.38×10^{14}
200c	3.97×10^{10}	2.84×10^{14}
FoF Particle sum	7.94×10^{11}	4.03×10^{14}
Virial Particle sum	7.94×10^{10}	3.38×10^{14}

Table 3.1: Halo Masses for different Definitions

Table 3.1 compares the effect of the different definitions quantitatively. The first two rows are taken from the properties table, while the last two are a direct summation of member particles. For this simulation, a single particle has a mass of $3.97 \times 10^{10} M_{\odot}/h$. In order to avoid the problematic low-mass region, **ROCKSTAR** was set to only identify haloes with a minimum of 20 particles. This results in the FoF particle sum having a lower end of $7.94 \times 10^{11} M_{\odot}/h$. Nonetheless, the unbinding can still produce lower values as it seems. The smallest virial mass in the table contains two particles, while the 200c definition even produces a halo of just one single particle. In all cases, the region this close to the resolution limit should be discarded. In general, the table shows the same effect that was visible in Fig. 3.7: the FoF mass overestimates the total mass of a halo. In contrast, the applied method for assigning member particles within the virial radius produces the same mass range as the properties table.

Figure 3.8 compares the same mass definitions, but for each halo individually. The left graph clearly shows the reason why the FoF particle sum should not be used for HMFs based on the virial mass, as there is a large discrepancy in masses. Without the unbinding, the particle sum is much larger than the virial mass. For smaller haloes, this value is up to an order of magnitude. After assigning the virial particles as discussed earlier, the relation looks much more centered on the diagonal line. The spread in both mass distributions is likely due to **ROCKSTAR** using different methods to calculate the virial mass, but the mean indicates that the masses are comparable and should be good enough for the halo mass function.

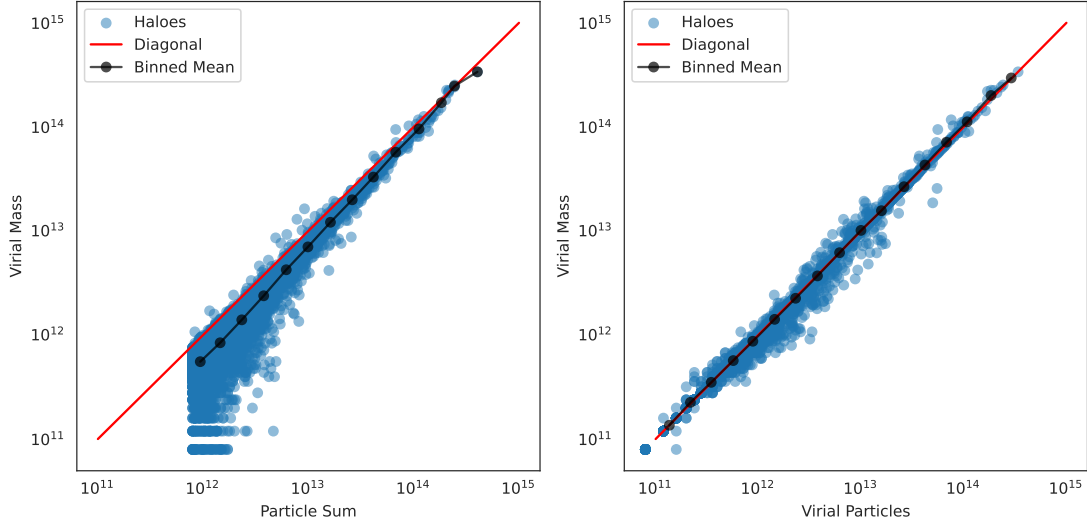


Fig. 3.8. A comparison of different mass definitions for each halo. The blue dots are individual haloes, the red line indicates a one to one relation for the case that two definitions produce the exact same masses, and the black line shows the binned mean of the blue data points. (Left) The virial mass versus the FoF particle sum. It shows that in general the virial mass is lower than the sum of all FoF particles. (Right) The virial mass from the properties table compared to the sum of the manually assigned particles inside the virial radius. Even though there is some scatter, the relation is not strongly biased in any direction, with the mean line matching almost perfectly.

3 Dark Matter Haloes

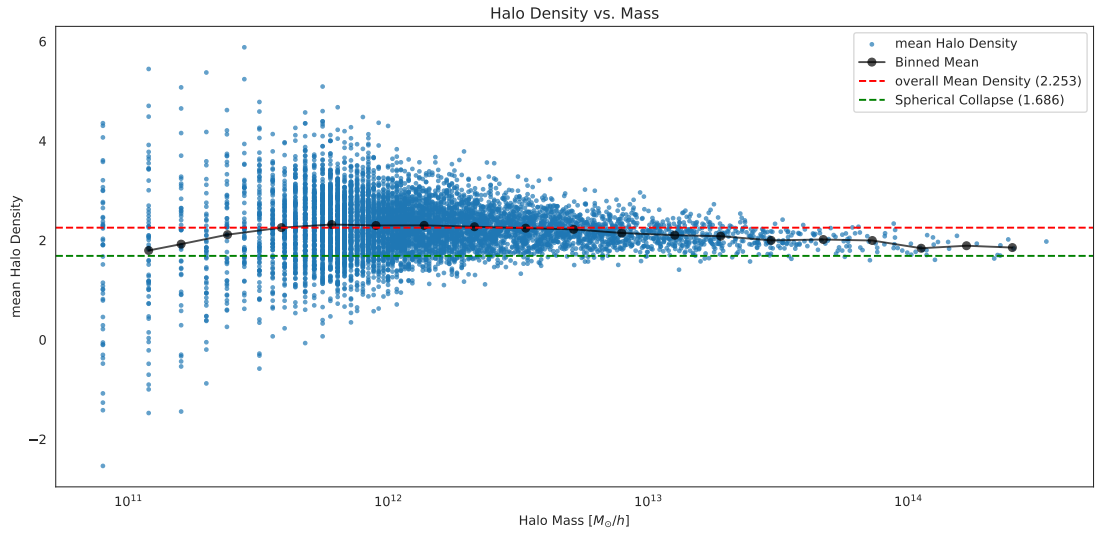


Fig. 3.9. The mean halo density in relation to the mass repeated, but with the virial particles this time. The blue dots represent the mean virial particles density for each halo. The red line is the single mean value for all the blue dots, while the black line shows the binned mean for different masses. The green line indicates the theoretical value for spherically overdense regions to collapse. With the more accurate mass definition, the data-points are almost exclusively above 1.686. The mean values are also significantly higher than before.

Lastly, the previous analysis of mean halo density versus mass can be redone with the new virial particles. Figure 3.9 behaves much better than the FoF particles in Fig. 3.5. At the lowest end of the halo masses, the discrete vertical lines are still apparent. Due to the few particles in these haloes, the data cannot be trusted and should be discarded for any value below $10^{12}M_{\odot}/h$. This region also shows the largest spread in density, but this is expected when calculating the mean of such sparse haloes. Meanwhile, almost all haloes with more than 20 particles contain a mean density above the linear theory value for spherical regions to collapse (1.686). The overall mean for all haloes in that graph is also clearly above that value at 2.253, indicating that the virial particles are much better approaches than the FoF counterparts.

This subsection of the applied methods for finding dark matter haloes and their protohalo patches in Lagrangian space has shown the reasons for the use of specific mass definitions and their effects on the resulting data.

The significance of physics-supported mass definitions and its consistent use have been demonstrated clearly. Although it would have been easier to just leave out the results stemming from the FoF halo particles, it does serve a deeper understanding into halo finders and the complicated formation process of these structures. The following section will look at a final improvement for the current structure classification method.

3.1.2 N-body code

During further analysis of comparing the halo mass function for different cosmologies, it became apparent that the halo finder consistently underestimated the number density by a small margin. This could not be explained by any error in the assignment of virial particles, as the discrepancy was also visible in the properties table produced by **ROCKSTAR**, which required no extra modifications. After some other causes were ruled out, it was decided to compare the results with a completely different N-body code. Naturally, it makes the most sense to use a simulator that has gone through rigorous testing and is widely known in the field of cosmological simulations, such as **GADGET-4** (Springel et al., 2021).

A single random seed is used in **DISCO-DJ** to create two identical initial conditions at $z = 49$. Then, one of them is evolved with **DISCO-DJ** (Disco), while the other is evolved with **GADGET-4** (G4). Both simulations are run up to $z = 0$ and saved as hdf5 files. At this point, the particle positions are almost identical, but there are some slight differences visible. For a more meaningful comparison, the **ROCKSTAR** halo finder analyzes both files, which also shows a distinct halo count, with Disco showing around 10% more haloes than G4. However, the G4 haloes are on average more massive. Next, the virial protohalo patches in Lagrangian coordinates are calculated.

The results of this can be seen in Fig. 3.10. Brighter colors indicate more massive haloes. Contrary to the halo table, it looks like G4 has far more smaller haloes, but more careful analysis shows that Disco produces mostly the same patches, but each one

3 Dark Matter Haloes

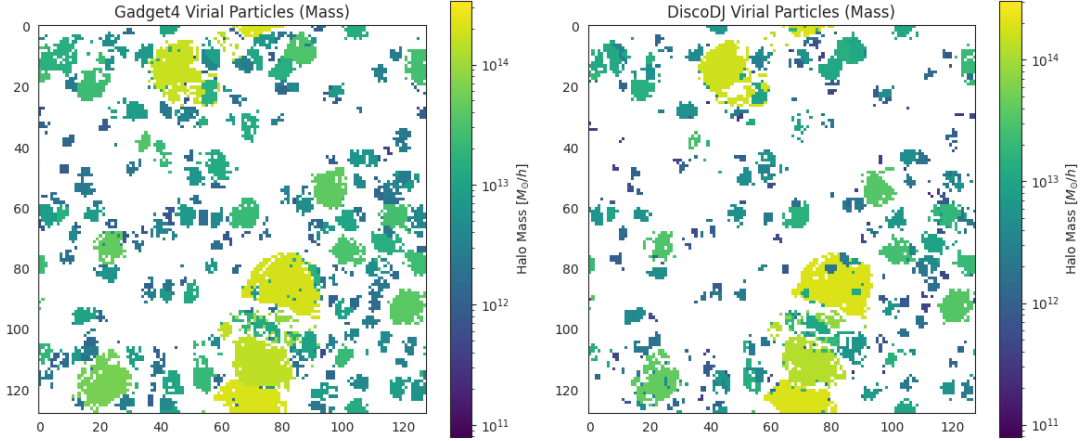


Fig. 3.10. A slice showing protohalo patches at $z = 49$ based on virial particles. The color indicates the mass of the final halo. (Left) This simulation is run with **GADGET-4**. (Right) This simulation has the same seed and initial conditions, but was evolved with **DISCO-DJ** instead. Both plots show similar halo patches when it comes to the more massive haloes, but they diverge significantly for smaller ones. G4 seems to consistently create more massive haloes, which is more noticeable for smaller patches.

contains significantly fewer particles. Even though the haloes are less massive across all scales, the difference is most noticeable for the smaller patches. Furthermore, the sparser protohaloes lead to more low-mass patches. This happens when a region becomes detached from the main halo. Therefore, there are quite a few small haloes in close proximity to larger ones, which simply do not exist in the G4 simulation.

To see which code produces the more accurate results compared to the theory, the halo mass function can be calculated for both simulations separately. This is done exactly the same as previously explained. The results are shown in Fig. 3.11. Similarly to before, the number density below $10^{12} M_{\odot}/h$ diverges due to the resolution of the simulation and should not be used for statistical analysis. However, even above that value, Disco consistently underestimates the amount of haloes. Meanwhile, G4 matches the analytical line very closely. It is also evident that Disco only overestimates the amount of smaller haloes, which leads to the higher halo count noticed from the properties table. G4 on the other hand produces more massive haloes, which is more apparent at the higher mass end. Interestingly, the slope of both codes is very similar for masses between 10^{12} and $10^{14} M_{\odot}/h$, just negatively offset in the case of Disco. This indicates that Disco produces a slightly sparser particle density across the entire simulation. It is clear that it is not able to produce a completely accurate HMF, so G4 should be used for further steps in the methodology.

However, it is not easy to pinpoint the reason for the different results in both codes, as they both use very different methods. The default N-body calculation method in

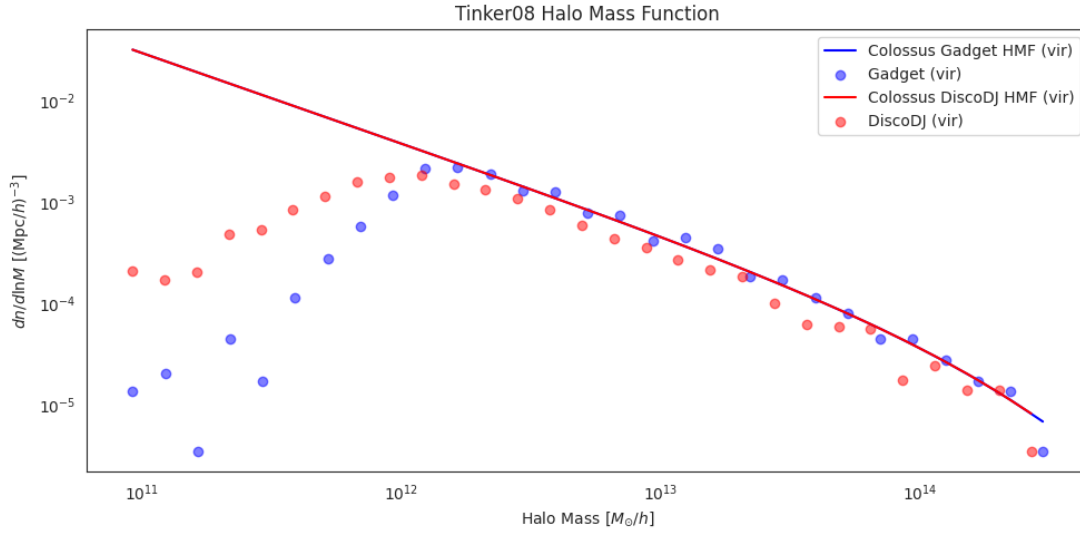


Fig. 3.11. A comparison of the resulting halo mass function for both codes with the theoretical Tinker08 line. The red data points are from the DISCO-DJ simulation, while the blue points come from the GADGET-4 code. The red and blue lines show the expected behavior for this cosmology. Since both simulations are based on the same cosmological input, the lines are exactly the same, with the exception of the mass range (since G4 reaches slightly higher masses). The number density of haloes decreases as their mass increases, with both codes producing very similar slopes, but Disco underestimates the HMF across the entire range at this resolution

3 Dark Matter Haloes

GADGET-4 is **TreePM**, while **DISCO-DJ** relies on a pure PM approach. The key difference is that PM methods apply a force softening at small scales due to their grid-based nature, while **TreePM** uses a hierarchical tree algorithm to calculate small scales separately (see Section 1.2.2 for more information). This means that PM codes underestimate power on small scales, which naturally suppresses small structures, including dark matter haloes, making them appear less massive in **DISCO-DJ**.

Another potential factor is the time-stepping scheme. **G4** employs adaptive time-stepping with significantly more steps to evolve particles to $z = 0$, potentially leading to more accurate results. In contrast, **Disco** uses fewer time steps, making the computation much faster but possibly missing some finer details in structure formation.

However, throughout the process of investigating the problem, it became apparent that it was possible to dampen the effect by increasing the simulation particles to 512^3 . At this resolution **Disco** is able to resolve smaller scales and reduce the difference in halo masses, but at a significantly higher computational cost, making this solution too expensive for training a neural network. There are new approaches, such as Wu, Guo et al. (2024), that provide corrections to haloes found from low-resolution simulations, though they are not explored in this thesis.

In conclusion, the main reason for the discrepancy is the force softening inherent to the PM method in **DISCO-DJ**, which suppresses small-scale structures. Increasing the resolution can reduce this issue, but the computational cost becomes problematic. Therefore, all future simulations will instead be run using **GADGET-4**, which provides more accurate small-scale forces due to its **TreePM** approach.

3.1.3 Differentiability

The previously explained methods showed how to find dark matter haloes in different simulations and calculate a range of varying properties, such as their densities, protohalo patches in Lagrangian space, and their number density per mass bin (HMF). However, neither the structure-finding method nor any further analysis is differentiable yet. This means that it is not possible to make any claims about the effect cosmological parameters have on the structure formation of these haloes from running a single simulation. A way to achieve this is to use the previous methods to create training data for a neural network to predict protohalo patches directly from the initial conditions. The creation of this labeled data does not have to be differentiable itself as long as all the steps before the CNN input and any analysis after the output are differentiable. The entire structure finding process can be differentiated once the training has been completed.

3.2 Convolutional Neural Networks

This following section will give a short introduction to the history of convolutional neural networks and their most important components. It will go into detail about identifying haloes with machine learning by first showing failed approaches and their problems.

3.2.1 Historical Introduction

Convolutional Neural Networks (CNNs) are a type of general artificial neural network, which are based on neurons (also known as nodes). They can be compared to biological neurons, hence the name, but in reality it is simply a function that is applied to an input. Neural networks usually have multiple layers made up of individual neurons. In the original feed-forward networks (e.g., Rumelhart et al., 1986) each node is connected to every following one, making them very inefficient for large datasets like images. CNNs, on the other hand, only share local connections, which effectively means that each node affects only a few neighboring nodes in the next layer. This greatly reduces parameters and speeds up model convergence. They first gained traction with LeNet-1 (LeCun et al., 1989) in the 1990s, which successfully tackled handwritten digit recognition tasks for the postal service. These networks were slowly improved over the next decades before AlexNet (Krizhevsky et al., 2012) achieved a significant leap in accuracy in the ImageNet challenge, demonstrating CNNs' utility in more complex visual tasks such as object recognition. This led to a surge of research into the modern CNN applications, which is still ongoing today. A more comprehensive review of the advances in this field can be found in Li, Yang et al. (2020).

Convolutional neural networks are widely used for datasets with grid-like structures and significant correlations between adjacent grid elements. The most obvious example of this is an image in which each color corresponds to a value on that pixel grid. To understand the basic mode of operation of such a network, it is necessary to know the following elements. The main component of CNNs is the convolution kernel, which is a matrix of values that is applied to a section of the input (element-wise multiplications followed by a sum) to compute a new output. The size of this kernel defines how many neighboring pixels are taken into account during the convolution step. If large-scale effects have to be analyzed, then either a larger kernel or more layers are needed (more on that later). Since the kernel is always smaller than the entire data, it will only be able to convolve a section before it needs to be moved and repeated on a new part. The distance the kernel is moved across the grid between each convolution is referred to as the stride. It affects the density of the convolution operation and can also change the output size. Stride 1 would keep the same size, while stride 2 already halves the height and width of an image. In order to avoid losing information on the edges of images, padding can be used. It adds zero values around the entire data to keep the dimension constant throughout the convolution.

3 Dark Matter Haloes

After the kernel is applied to each section of the data, an activation function is applied to the output. It introduces non-linearity into the model, which helps it separate abstract features (Xu et al., 2015). Typical options include ReLU (Rectified Linear Unit, which replaces negative values with zero while keeping positive values unchanged), Sigmoid (Squashes values into a range between 0 and 1), Tanh (similar to sigmoid but ranges from -1 to 1), Leaky ReLU (similar to ReLU but allows small gradients for negative values to avoid dying neurons), and many more.

A common practice when working with image data is to repeat the convolution followed by an activation function on different spatial resolutions. It is accomplished by down-sampling the feature map to smaller dimensions before repeating the previous processes. A common approach for this is max-pooling, which simply takes the maximum value of a section and gets rid of all other pixels. While this does reduce the size of an image and therefore removes information, it helps with overfitting (Hawkins, 2004).

All three steps combined are known as a layer and can be repeated multiple times to extract more large-scale features and dependencies. In the case of object classification, the reduction in image size is not a problem, since the final output will likely only be a single value corresponding to a word. However, image segmentation (classifying each pixel) needs the same size and shape for both input and output. This is the case for identifying protohalo patches in the initial conditions, as the result should be the same data shape but contain halo patches instead of density values. This can be accomplished with a special CNN architecture known as U-Net.

It was first developed by Ronneberger et al. (2015) for image segmentation (i.e., assigning each part of an image to its corresponding class). The model is a fully convolutional network and consists of an encoder and decoder part. The encoder is made up of multiple layers that each apply convolutions before pooling the data to reduce its spatial dimensions. After reaching the last layer (bottleneck) and applying the convolutions one more time, the data gets up-sampled again the same amount of times until it reaches the original input dimensions. This part of the architecture is known as the decoder. It is almost symmetrical to the encoder, with only the down-sampling (e.g., max-pooling) being exchanged with up-sampling (e.g., transposed convolution). The gradual decrease and increase in spatial resolution helps the model extract features across multiple scales while preserving fine-grained details. In order to help the network with local information, the features of each encoder layer are combined with the decoder layer before the convolutions are applied. This part of the U-Net is known as a skip connection and works as a reminder of the input data throughout the entire network. In general, this type of neural network performs very well with a range of different image to image predictions such as classification, segmentation, reconstruction, etc. And the large increase in implementations of U-Net architectures in research publications (Azad et al., 2022) shows the success it has had in many different scientific fields.

A crucial part of CNNs are trainable parameters, which are updated during training to improve predictions. They primarily include the weights and biases of the convolution kernels, while operations like max-pooling are non-trainable. The weights measure the

significance of every input feature, while the bias is a single value that is added to improve the overall result. For a fully connected layer this can be expressed as

$$y = Wx + b, \quad (3.3)$$

where y and x represent the output and input respectively, W stands for the weights matrix, and b for the bias. The number of trainable parameters depends on the input-channels, kernel size of the convolution, and output-channels. They are updated during training using gradient-based optimization, typically through back propagation combined with stochastic gradient descent (SGD) or variants like Adam Kingma and Ba (2014). It works by processing input data, computing the output predictions, and evaluating a loss function that quantifies the difference between predictions and ground truth labels. The gradients of this loss function with respect to the trainable parameters are then computed using back propagation. These gradients are used to update the parameters in a direction that minimizes the loss, gradually improving the network's accuracy over many iterations.

3.2.2 Halo Identification with CNNs

In the context of this thesis, the U-Net covers all the requirements needed for the challenge of identifying protohalo patches in a differentiable manner. The differentiability is a huge advantage of neural networks and comes from the fact that they only return probabilities, which allows for a smooth transition from one class to another (halo versus background, for example). In the end, this makes it possible to evaluate a halo mass function with respect to any cosmological parameter and the effect they have on the haloes. In theory, this would also allow for analysis of the halo shapes in a differentiable manner. Overall, the idea of a differentiable halo finder can have many different applications, and the neural network approach has been done before. Berger and Stein (2019) for example used a U-Net to map particles from an initial density field directly to the final haloes at $z = 0$. The trained model was able to predict the positions and sizes of haloes quite well. However, they only used the **Peak Patch** algorithm to identify true haloes for training and evaluation, which is an analytical method by Bond and Myers (1996). It works by identifying density peaks and the resulting collapsing region around them in Lagrangian space, before using LPT to move the haloes to their final positions in Eulerian space. It works well enough, but is much simpler than the current numerical halo finders (see Knebe et al., 2013 for comparisons). Furthermore, they had to post-process the U-Net output to better match the results by Peak Patch. They did so by identifying connected regions and placing spheres at their centers. The radius is determined by having a prediction probability cutoff. While this process improves the results, it also removes any differentiability that was possible before. It is worth pointing out that parameter dependence was not their intention. Instead, they tried and managed to create halo catalogs directly from initial density fields, which is orders of magnitude faster than normal numerical methods, while achieving relatively high accuracy (Stein et al., 2019).

3 Dark Matter Haloes

Bernardini et al. (2020) used a slightly different approach. Instead of predicting the collapsed haloes, their network was trained to recognize protohalo patches from the initial density field. This is the same approach that will be used for this thesis. Furthermore, they also used a similar method for calculating the true patches, that is, they first evolved a simulation to redshift 0 and then ran a halo finder (in their case HOP by Eisenstein and Hut, 1998) before tracing the halo particles back to the initial density field. However, their U-Net used regression instead of classification, which means that each pixel was assigned a continuous value between zero and one. In each patch, the network predicted the distance from its border. They achieved this by applying a Euclidean distance transformation to each patch and normalizing it by the maximum value in that halo. Again, the network needed post-processing in order to separate the haloes from one another. In this case, they used an adaptive watershed algorithm to distinguish the regions. While improving the results, this method is also not differentiable.

A more recent approach by López-Cano et al. (2024) also tackles the challenge of identifying protohalo patches from the initial conditions. Their method consists of a two-step neural network pipeline: first, a semantic segmentation network predicts which particles will collapse into haloes (binary prediction), and then an instance segmentation network groups these particles into individual haloes. This is achieved by mapping particles into a pseudo-space where distinct haloes become more easily separable. The actual instance segmentation requires a post-process step with a clustering algorithm from the pseudo-space distribution. Their final protohalo patches are remarkably close to the ground truth. They are even able to reproduce relations of the linear Lagrangian bias parameter or effects of varying parameters without explicitly training their model for these aspects. Nonetheless, this method was also not approached with differentiability in mind, so both the processing in between models (turning the binary prediction probabilities into single class values) and the final clustering algorithm are discrete procedures.

It is worth mentioning that Lucie-Smith et al. (2024) were able to predict halo masses directly from the initial density field using a CNN (not U-Net, however), but their approach is quite distinct. They do not try to identify halo particles, but rather just a single halo mass from a sub-volume of the density field at $z = 99$. While this method does not need post-processing and should be completely differentiable, it is a different and arguably simpler task than a voxel-specific halo finder. Furthermore, the dependence on cosmological parameters is not analyzed, nor any other gradient aspect. Instead, the goal of that paper was understanding the role of anisotropic information for the final mass of a halo.

Overall, the comparison to other literature shows that using the output of a U-Net directly for differentiable dark matter halo statistics is very difficult. The main reason is that connected regions have to be separated from one another to accurately identify each patch, count the total amount of haloes, and calculate their mass. This cannot be done in a differentiable manner, which is why simply classifying the initial density field into two classes (halo versus background) is not sufficient. The regression approach faces a

similar problem. One can imagine that being able to classify each halo separately would eliminate this problem, but training a neural network to achieve this task is much more difficult. For one, the network has to be able to predict a different number of classes each time, as the halo count varies across simulations. Furthermore, networks that are trained for such a task (known as instance segmentation) usually do not have to also learn the process to create the patches. That is likely the reason why López-Cano et al. (2024) also used one network to perform binary prediction and then have a second segment the patches into individual haloes. However, they used additional processing in their pipeline, that would have to be taken care of by the networks directly to ensure differentiability. In fact, this is one of the methods that was tried for this thesis but ultimately failed. The next section will cover some of the other ideas that were discussed and tried, as well as the reasons why they were not successful. The final working method will be covered in the section underneath the next.

3.2.3 Classification vs Regression

The first decision for the neural network is the kind of label (output) that makes the most sense for differentiable identification of dark matter haloes. The two main options are classification and regression, each has its own advantages and disadvantages. Regression is used when the network has to predict a continuous value for each pixel in the output. This is especially effective when the result does not have clear borders but rather shows a trend towards certain regions. Classification is used when each point in the label has to be assigned to a predefined class. In the end, the classes are still based on a continuous prediction, which is crucial for differentiability. This is achieved by predicting a probability for each point and class. The total probability always adds up to one over the classes (i.e., if the network is very certain, one class will have a probability close to 1, while the other are closer to 0). In order to just show the most likely prediction, one can simply choose the class with the highest probability for each data point and discard the rest. Although it is helpful in visualizing the outcome, the process itself is not differentiable, so it should be used with care.

Overall, the individual haloes suggest a classification network, but the uncertain and difficult-to-define borders of these dark matter structures favor the regression approach. Therefore, both methods will have to be tested and analyzed for the specific needs of this U-Net to prove which is superior.

Failed approaches

The easiest way to do this is simplifying the network into just two dimensions and using slices instead of the full 3D density field / protohalo patches. This is as simple as changing the input-, output- and kernel-shape, but makes training the U-Net much faster. Once a network's results look promising, the same method is repeated in the full three dimensions.

3 Dark Matter Haloes

From a quick test in 2D it was apparent that the simplest binary classification works really well, where the input consists of the initial density field and the label is the same map containing just two classes: halo particle (value 1) or background (value 0). In this case the U-Net can learn to accurately predict the features quite quickly. However, there is no easy way to extract the individual haloes from connected patches. Adding a border class to surround each patch helps separate individual haloes (Li, Zhao et al., 2020), but it becomes problematic when patches overlap, since there is no background that could be redefined as a border. The low resolution also makes it unfeasible to use the outer halo particles themselves because small haloes would become border classes completely.

As no clear solution can be found for a differentiable post-processing of the binary classification result, and other papers (like Bernardini et al., 2020) had used regression, this approach is tried next. An immediate advantage of a regression network is that the label or output has many different options and can be adjusted to the specific needs of the application. The first obvious choice is that the background has a value of 0 and the halo patches take some higher numbers. What then remains to be determined is the range and scaling of the values inside the haloes. Since Bernardini et al. (ibid.) already showed that it is possible to use Euclidean distance transformation (EDT), it is tried first. Applying EDT to the halo patches rescales all particles inside in relation to the distance to the background. This means the border has a value of 1 and increases toward the center of the patch. After training it for an appropriate amount of time, it became obvious that the patches are generally more extended (i.e., the background bordering a halo had values higher than 0), and larger haloes were favored. This is because they are more certain, but also because they contain larger values so the network is punished more for wrong predictions. To some extent, that behavior is acceptable since larger haloes should be easier to predict, but at the same time smaller haloes should not be neglected. By rescaling all the EDT patches to have the same maximum (e.g., 1 in the center and decreasing to 0 towards the border), this effect can be counteracted. The result is that smaller haloes appear more often in the prediction. Large patches are still weighted slightly more because they have more particles and therefore contribute more to the loss, but not as excessively anymore. However, the network still struggles with neighboring patches and consistent scaling. The borders often extend further, and instead of reaching a maximum in the center, the entire inner regions usually have a value of around 0.7.

More tests were also performed with different scaling or center definitions, such as creating Voronoi maps and calculating the distance to the center of each patch. For a spherical halo patch, that would be identical to the EDT approach, but for different shapes, the particles that are further away from the center are assigned a lower value. In general, this method leads to a sharper drop-off from the center regions, which made the U-Net predict smaller patches and was overall worse than EDT. Another different approach that turned out to work best was using the center of mass and calculating the distance for each particle in relation to that point. The values are then scaled with a Gaussian and the patch mass, so larger haloes drop off more slowly than smaller ones, yet all the haloes have a maximum of 1 in the center. This method was the first to predict

the full range 0 to 1 almost completely, but neighboring patches still blurred together making it very difficult to distinguish individual haloes.

Nonetheless, even the Gaussian approach did not perform as well as the binary classification U-Net in terms of halo patches versus background. In order to incorporate the strengths of the simple classification with the more meaningful regression, a two-step prediction can be devised, similar to the approach by López-Cano et al. (2024). It works by building and training two separate U-Nets. The first one gets the density field as input and has to predict just two classes (halo or background). The second network uses both the density field and the binary classification as input and converts that into a single regression prediction. Using these two networks sequentially, it is possible to predict the halo positions very well while also distinguishing neighboring haloes using the Gaussian-scaled distances to the center of mass. However, this method turned out to be more problematic and misleading than originally thought. The main mistake was training the regression network on the true binary maps. This is wrong for two reasons. First, turning binary prediction (probabilities) into a binary map is a discrete method, so the differentiability is disrupted in between the two CNNs. Second, it is misleading to train the network on true (ROCKSTAR) patches, but evaluate it on predicted patches. Even if the classification network performs really well, the patches still look different from the true ones, especially around the edges. The correct approach to the two-step network is first training the classification U-Net, then letting it predict the entire training data, and using the initial density field as well as the predicted protohalo patches (probabilities) as the input for the regression U-Net during training. The task immediately becomes a lot more difficult because the predicted probabilities do not align clearly with the true patches, and the uncertainty in the classes propagates through the second network. While this two-step approach seemed extremely successful at first, it ended up being quite troublesome and performs worse than a single regression network.

Before explaining the actually successful method, one last somewhat effective approach will be mentioned as it does point the way to the final approach. Throughout all the previous networks, no matter how well they worked, none of them showed a clear solution to further process the output without losing the differentiability. As a quick reminder, in order to calculate the halo mass function, one needs to count the haloes and separate them into mass bins. This is inherently difficult to do in a continuous manner. The novel approach here is to predict protohalo patches that could be used directly as bin data. This can be achieved by training a binary classification network but including a mass condition. That means that only protohalo patches within a certain mass range are predicted. It is implemented by concatenating the mass range values directly to the density field (in the bottleneck layer). During training, mass ranges within the minimum and maximum halo masses are randomly chosen each epoch, and the labels are filtered to show only these haloes. By letting the CNN predict the same map multiple times with different mass ranges, one can theoretically layer all predictions on top of each other and create one simulation box containing all protohalo patches. Furthermore, halo class particles of an individual prediction can simply be summed up and divided by the mean

3 Dark Matter Haloes

halo mass of that range to produce an average count for a single bin. The method works reasonably well and produces the first differentiable HMF, but overlaying all predictions for a single density field leads to a larger number of haloes than the real simulation has. This happens because the network does not care if it predicts the same halo across multiple mass ranges as each prediction is completely independent. To some extent, that is acceptable because the haloes are based on probabilities, and slight changes in the cosmological parameters or the initial density field could certainly result in a change in halo mass. Nevertheless, the same protohalo patch should not be within multiple mass ranges with a high probability. Overall, the network sometimes underestimates single-mass range predictions while at the same time consistently overestimates the combined number of protohalo patches. Nonetheless, the approach was the last step towards the final U-Net because it showed the effectiveness of binning the haloes by masses, and the importance of not allowing the same particles to independently count towards multiple bins at the same time.

3.2.4 Final Method

Having tried many different options showed exactly which methods worked well and which did not. Combining newly acquired knowledge, especially from the last approach, led to this final network. The main idea is a classification network but using multiple classes directly as binned data. We accomplish it by preparing the labels to categorize the protohalo patches into 10 classes according to their virial halo mass. In that scenario the background is class 0 and the other 9 classes are evenly spaced on a log-scale ranging from the smallest to the most massive halo in the entire training data. This approach has the benefit of using a classification network, which proved to be more accurate and faster at learning features, likely due to having fewer options compared to a full regression network. It also solves the problems related to post-processing as the classes can be used directly as bins for the HMF. Visually, it is also easier to distinguish individual halo patches even when neighboring patches are connected, because most of the time they will belong to two different classes. Maybe most importantly, the classes are no longer independent of each other. Even though the predictions are still based on probabilities, having the network classify all haloes at the same time means that each pixel's class depends on the others. In other words, if the U-Net is very certain that a pixel belongs to a specific class, the probabilities of the other classes for that pixel become lower. The total probability always adds up to one since each particle has to belong to one of these classes. The continuous transition from one class to the next is the basis for this differentiable identification of cosmic structures, enabling a halo mass function that can be evaluated with respect to any cosmological parameter. The exact details of our U-Net will be discussed in more detail in the following sections.

Training Data

Having finally settled on a working neural network, the first thing to do is create training and validation data. One will be used for training the network, meaning the CNN parameters are adapted after every pass through, while the other is only used to make sure the network does not over-fit. After each training epoch, the validation data is also passed through the U-Net, though the parameters are not adjusted, and the loss is calculated by comparing the prediction to the ground truth (also known as label). If the training loss decreases while the validation loss increases, it is a sign of overfitting, and the training will be stopped.

The data itself consists of input and target data cubes. The input is the initial density field of the simulation at redshift $z = 49$, while the protohalo patches are the output target for training. The exact methods to calculate one from the other were already explained in detail in Section 3.1. In short, the particles are evolved forward in time using **GADGET-4** up to $z = 0$. **ROCKSTAR** is then used to identify all haloes as well as properties such as their positions, masses, and many more. A sphere is placed at each center position, and all particles that lie within the virial radius are assigned to that halo. These particles are then traced back to the initial conditions at $z = 49$ to form their protohalo patches. Since the U-Net is based on binned halo masses, all particles in a patch are actually assigned the virial mass of their halo. The binning process is applied during the data preparation for the neural network. After each simulation is computed, all files except for the initial density field and the protohalo patches are discarded to save memory.

Since the motivation behind this project is evaluating the effect cosmological parameters have on the HMF, it is important to allow the network to learn from varying parameters. For this reason, the parameters for all the simulations are predefined using a Latin hypercube. The method was described by McKay et al. (1979) to sample values from a multidimensional distribution. It ensures that all parameters are equally spaced within the defined range. The training data is based on 500 simulations, so a Latin hypercube is created for that number of samples, including the simulation seed and the eight cosmological parameters (Ω_c , Ω_b , Ω_k , h , n_s , σ_8 , w_0 , w_a). It is important to note that Ω_c and Ω_b are the primary parameters in **DISCO-DJ**, so changing either of them affects Ω_m . Similarly, $\Omega_{de} = 1 - \Omega_m$ is implicitly defined such that the cosmology remains flat. The 500 randomly chosen cosmologies are saved as a text file to compare the predicted HMF with the expected theoretical line once training has finished. However, the neural network is not conditioned on the cosmological parameters (i.e., they are not directly used as input). This is important because it forces it to learn everything from the initial density field, which itself depends on the cosmology. The chosen range for each parameter is based on the Quijote simulation suite (Villaescusa-Navarro, Hahn et al., 2020), as well as the Mira-Titan HMF emulator (Bocquet et al., 2020) and Moresco et al. (2012) for Ω_k . Overall, the allowed range is quite large to ensure the network can learn the dependence

3 Dark Matter Haloes

as well as possible while still remaining in relatively realistic regions.

	h	Ω_b	Ω_c	Ω_k	w_0	w_a	n_s	σ_8
Lower Bound	0.55	0.03	0.13	-0.005	-1.3	-2.16	0.85	0.7
Upper Bound	0.85	0.07	0.45	0.005	-0.7	1.3	1.05	0.9

Table 3.2: Cosmological parameter limits for the Latin hypercube.

It should be mentioned that 500 simulations are not enough to train this neural network, but the data is augmented each time it is loaded by the model. A three-dimensional data cube can be rotated and flipped along each axis, which generates pseudo-new cubes for the network, as it is effectively a different input. With these augmentations, each simulation can be represented in 48 unique ways, leading to a total amount of 24,000 data cubes available for training and validation. During each epoch, when a new batch of training data is loaded, the network randomly rotates and flips the cubes. It is important to apply the same transformation to both the input and output to ensure that the network can still accurately learn all the features. Having added the data augmentation, the network does not show any signs of overfitting. Instead, training is stopped whenever the validation does not improve over a predefined number of epochs.

U-Net Architecture

Most of the important components of a convolutional neural network have been explained in the introduction of this chapter, but this next section will cover which exact combination and order was used for this case. The entire model architecture can be seen in 3.12. It consists of the encoder (left side), bottleneck (bottom middle), and decoder (right side), which were created using the Haiku (Hennigan et al., 2020) neural network library for JAX.

All layers consist of a double convolution with a kernel shape of $3 \times 3 \times 3$ and a stride of 1. The padding is defined in a way ('SAME') as to not change the shape during the convolution process. However, this simply extends the data with zeros, which does not take into account periodicity. It will be discussed further in Section 3.4. The output-channels or filters are consistent for both convolutions, but they change across each layer to accommodate the difference in resolution. Because of that, each layer L has $2^L \times n_f$ filters. Tests performed with different values have shown that $n_f = 16$ works quite well for this neural network. The 'Leaky-ReLu' activation function is applied after each convolution, which helps with convergence, as well as exploding or vanishing gradients (Xu et al., 2015).

Each layer in the encoder is composed the following way. First, a double convolution is applied, the result of which is saved for the skip connection to the decoder. Dropout of 0.5 is used, which randomly sets that fraction of neurons to 0. This regularization method prevents overfitting during training and forces neurons to work more independently (Hinton et al., 2012). During evaluation, dropout is not used as it reduces the amount of

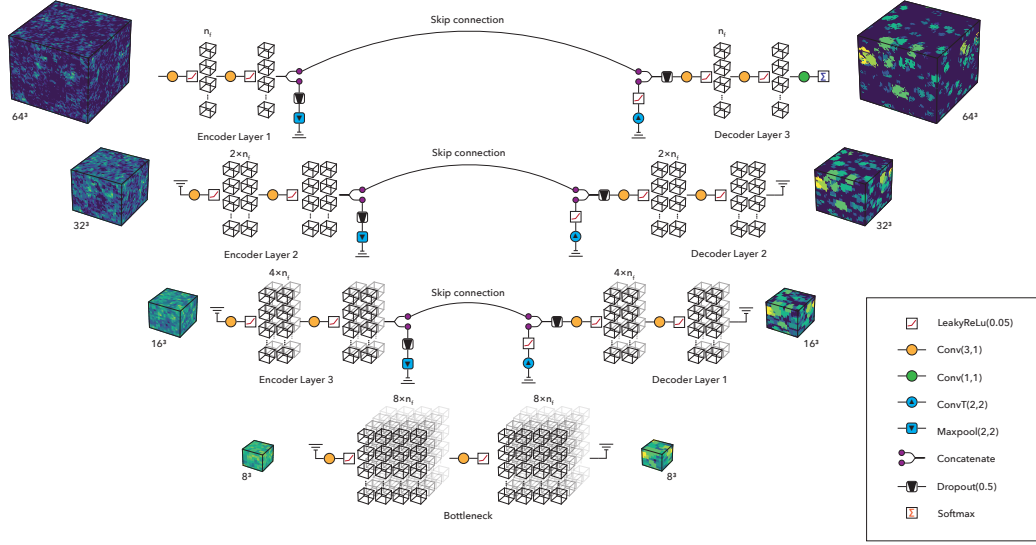


Fig. 3.12. A schematic representation of the 4 layer U-Net architecture used for classification. The legend shows each module and corresponding parameters used, like: (kernel size, stride). The number of filters of the first convolution is $n_f = 16$, changing with each following layer. The final convolution uses 10 filters, which represent the 10 mass bins.

information the network gets to see. The last step of each encoder layer is to reduce the spatial resolution. This is accomplished through max-pooling. This U-Net uses a kernel of $2 \times 2 \times 2$ and a stride of 2, which effectively means the model halves the resolution per axis by only keeping the maximum value out of every 8 (a cube of 2 by 2 by 2) particles. This process is repeated three times before reaching the bottleneck.

At which point the network does not change the spatial resolution any further. Instead, only the double convolution is applied. So far, the data has been down-sampled from 64^3 to 8^3 particles. At the same time, the convolution filters were increased from $n_f = 16$ all the way to 128.

After the bottleneck, the previous procedure is basically reversed. This part is called the decoder. The data is first up-sampled using a transposed convolution with kernel-shape $2 \times 2 \times 2$ and stride 2. In combination with reducing the filters, the overall shape matches that of the corresponding encoder layer. Similarly to the normal convolution, the ‘Leaky-ReLu’ activation function is also used after the transposed one. At this point, the skip connection data from the encoder is concatenated to the decoder. Next, dropout is applied again (during training). Lastly, the double convolution layer is executed before the next layer starts.

After the U-Net has gone through the final decoder layer, the data has a shape of 64^3 with 16 filters (i.e., $64 \times 64 \times 64 \times 16$). In order to match the classification result, a last convolution is applied with kernel $1 \times 1 \times 1$ and stride 1. The important thing here is

3 Dark Matter Haloes

that the output-channels match the number of classes, which in this case is 10. Finally, a softmax function is used to obtain probabilities for the classes that add up to 1.

Having covered the U-Net architecture in detail, there are two more details to address for training the network. The first is the learning rate. This hyper-parameter controls how much the model is updated after every iteration. Each time training data is passed through the model, the loss is calculated by comparing the prediction to the ground truth. Gradient descent (Adam) is then used to update the model parameters in a way to minimize the loss. Larger learning rates allow the network to change these values more, which generally helps it converge faster. However, if the value is too large, the weights can oscillate around the minimum instead of converging toward it (Wu, Liu et al., 2019). Training the U-Net using different learning rates made it apparent that a value of 10^{-4} is almost ideal. Anything larger and the model does not learn any classes other than the background, while lower values usually take many more iterations to achieve the same accuracy. However, during later epochs, when changes to the model parameters do not have to be so large anymore, a lower learning rate proves to be even better. Therefore, a schedule is implemented to gradually decrease the learning rate after each iteration. It decreases around 10% over every 100 epochs.

The last, but arguably most important part of this U-Net is the loss function. There are many different ways to evaluate the difference between prediction and label, so choosing a method that captures the most accurate meaning is of great significance. In order to understand how the loss function interacts with the results, a few common options are discussed first. The simplest way to calculate the difference of a prediction from the ground truth is to compute the absolute difference of each point in the data and add them all up. This is known as the $L1$ loss: $\sum_{i=1}^n |y_i - \hat{y}_i|$. By squaring the error, it punishes outliers more heavily. This is known as the $L2$ loss: $\sum_{i=1}^n (y_i - \hat{y}_i)^2$. In most cases, instead of using the sum of all pixels/voxels, the loss is multiplied by $1/n$. This changes the $L1$ loss into the mean absolute error (MAE) and $L2$ into the mean squared error (MSE). Classification networks often use cross-entropy loss: $-\sum_{i=1}^n y_i \log(\hat{y}_i)$, and the mean equivalent of it (MCE). These losses are commonly used, and there are other slight variations to better suit the purpose, but a problem arises for multi-class classification. In the case of mass-binned protohalo patches, the classes are not physically independent from each other, yet MCE does not take these relations into account. For example, if the true class of a patch is 7, the loss will be the same no matter if the network predicts it to be class 1 or class 6 as both classes are wrong. However, in this case, it is obviously far better if the network predicts a closer class. This is implemented using the squared Earth Mover’s Distance (EMD, see Rubner et al., 2000; Hou et al., 2016), which measures the minimum cost of transforming one probability distribution into another. It gets its name from a physical analogy: there are piles of earth (one probability distribution) and empty holes (another distribution), and the goal is to move the earth to fill the holes with the least amount of work. The effort required depends on both how much earth needs to be moved and how far it has to be moved. In practice, its main difference to the above mentioned loss functions is that it transforms the probabilities of the classes

into cumulative sums with:

$$L_{EMD^2}(p, t) = \sum_{i=1}^C (\text{CDF}_i(p) - \text{CDF}_i(t))^2, \quad (3.4)$$

where C are the classes, p and t the prediction and target respectively, and CDF the cumulative distribution function. In the case of the label, it means that all classes below the truth are 0, while all above are 1. For the prediction, the probabilities are distributed among all the classes, but by applying a cumulative sum, the distribution matches better when the network is closer to the true class, even if it is not spot on. Using the unweighted version of the squared EMD as described above, means that the distance between all classes is treated equally. It would have been possible to weigh the loss function in a way to increase the difference between background (class 0) and lightest mass bin (class 1), but we chose not to impose an artificial weighting scheme that might bias the classifier toward certain mass bins. Instead, we rely on the model to learn the natural distribution of the haloes without explicitly enforcing larger separations between specific classes. Overall, it shows that the choice of an appropriate loss function is extremely important, especially when the classes are correlated. After implementing the squared EMD loss, the U-Net’s results improve dramatically, and patches are almost always predicted to be very close to the true mass bin, if not exactly correct.

3.3 Results

With the methods, model architecture, and hyper-parameters all finalized, the training can be started. Using a Nvidia RTX 3090, a single epoch takes around 8.5 seconds. The U-Net stopped after around 2800 epochs (~ 7 hours) because the loss did not decrease over the last 200 epochs. The final EMD loss of the validation dataset reaches a value of 0.0332. Figure 3.13 shows the evolution of the loss function for both the training and validation data.

The fact that the validation loss does not deviate significantly means that the model is truly learning the process needed to find protohalo patches, and not just memorizing the density fields used for training (overfitting). It should be mentioned that the losses shown here are averaged values for each epoch. Individual iterations show much more scatter. As is usually the case with neural networks, the model improves substantially in the beginning and flattens out rather quickly. The majority is learned within just 500 epochs. However, the U-Net still shows continuous improvements for quite a while, even if very small, before eventually reaching a plateau. Most interesting is the little step in the first 100 epochs. Evaluating the model in that state shows that it predicts the entire map to be class 0 (background). Logically, the model starts off predicting random classes, but almost immediately learns that it can reduce the loss massively by simply predicting class 0 for every voxel. The background is by far the most common class for a particle,

3 Dark Matter Haloes

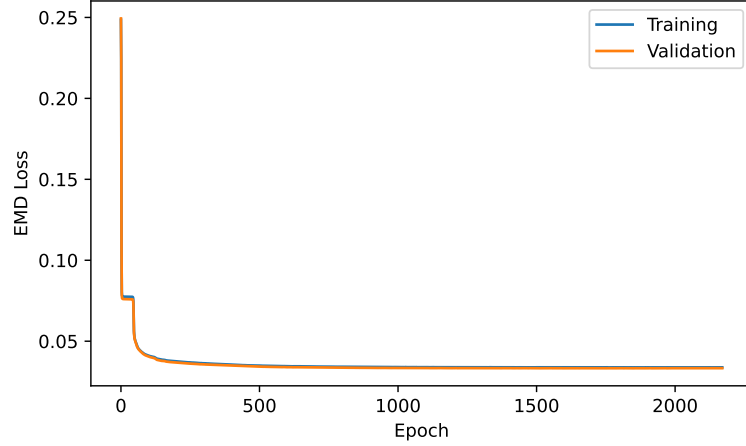


Fig. 3.13. The evolution of the Earth Mover’s Distance loss throughout the training process. There is very little difference between the training (blue) and validation (orange) dataset, which would normally show in the case of overfitting. Most important features are already learned within 500 epochs.

so it makes sense that this is learned first. It then takes quite a while for the network to start predicting the larger patches correctly, after which the second quick drop in EMD loss is visible. The longer the model is trained, the better it becomes at predicting the smaller haloes and patch-border particles, as those are the most uncertain.

Even though the validation data itself is never used to train the network, it is still better to use an independent testing dataset for further evaluation once the model is fully trained. This data is created through the same method from random cosmological parameters within the previously mentioned Latin hypercube. Any further analysis is performed using the test data. Figure 3.14 shows such a simulation and the prediction.

The density field as seen in this image is already preprocessed, which includes a rescaling of the values. When loading the initial conditions, the data is divided by 2.66, which is the average standard deviation of these fields. This helps the CNN converge faster (Huang et al., 2020) because the new data has a simpler standard deviation of 1. The mean is not adjusted, though it is already close to 0 as is. Other than this rescaling and the random orientation during training, the data is only changed to fit the channel and batch dimensions expected by the network. The middle plot of Fig. 3.14 is obviously not used during the prediction and is only shown to assess the accuracy.

The U-Net output is by far not perfect, but it seems to generally be able to identify the protohalo patches. A closer look at the statistical properties of these predictions will be shown later, though the visual maps already demonstrate quite a few interesting facts. First, the network creates smoother protohalo patches, where the edges are seldom disconnected from the main body. Furthermore, it is able to predict larger haloes much better than the smaller ones. Both of these phenomena are likely connected to the same

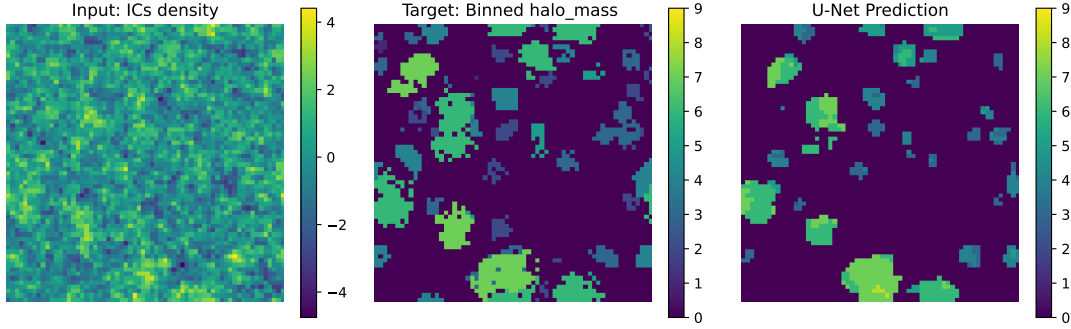


Fig. 3.14. 2D Slices of the trained model prediction. The U-Net works with full 3D maps. The left plot shows the initial density field at $z = 49$ used as input. The middle is the target with the binned protohalo patches (also $z = 49$), known as ground truth. The right image shows the network prediction. While not perfect, it does reproduce most positions, sizes and mass bins fairly accurately.

underlying reason. Haloes are very difficult to define on a particle level. The complex non-linear process which creates these structures is very sensitive to tiny changes in the initial conditions. It is for that reason different halo finders usually do not produce the exact same halo catalogs. The exact borders and edges of these patches usually don't agree either, especially when the numerical simulations are at such a low resolution as here. Since the network cannot say with certainty which particle is part of the halo or the background in those cases, it is safer to guess the background than to risk a wrong prediction.

Another interesting property of the U-Net protohaloes is that patches are not attributed to a single mass bin. Especially the larger ones often consist of multiple different mass particles. They are usually higher in the center and lower along the edges, which minimizes the error when misclassifying background. While this multi-mass patch is physically incorrect, it highlights the differentiable structure finding that utilizes these more fluid mass definitions. However, one should not forget that this single image representation of the prediction reduces the complexity, since only the highest probability class is shown. For further calculations such as the HMF, the raw probabilities can be used instead.

All in all, comparing the full simulation - including DISCO-DJ for the initial conditions, GADGET-4 for the N-body calculations, ROCKSTAR for identifying the haloes, and back-tracking the virial particles to the initial redshift - to the U-Net prediction, it is clear that the neural network is not perfect but still a reasonable imitation with a halo particle precision of 86.74% (not distinguishing between mass classes). However, the model is much faster. A full simulation takes around 4 minutes on the previously mentioned hardware. The U-Net approach is completed in just 1 second. In both cases, this includes the initial condition generation with Disco. If this is left out of the equation, a full simulation still takes around 4 minutes, while a prediction only needs 13.5ms. This is 4 orders of magnitude (17,777) faster, which makes it very efficient in generating

approximate halo catalogs with different cosmological parameters.

3.3.1 Halo Mass Function

The HMF can be calculated quite easily using the trained mass bins. First, the prediction is totaled across the entire box to see number of particles $N_{p,i}$ per class i . Since the probabilities are used directly, a single particle can contribute to multiple classes at once, though only partially, so the total number stays consistent. The mass bins of each class are used to define an average halo mass $\bar{M}_{halo,i}$. Calculating the approximate number of haloes that can be constructed with the particles in each class depends on the mass of a single particle, which in turn depends on the cosmological parameters. It is derived with

$$M_p = \frac{\Omega_m \rho_{crit} V_{box}}{N_p}, \quad (3.5)$$

where M_p is the dark matter mass of a single particle, $\Omega_m = \Omega_b + \Omega_c$ as gravity-only simulations usually do not differentiate between these types of matter, ρ_{crit} is the critical density of the Universe, V_{box} is the volume of the simulation box ($100^3 \text{Mpc}^3/h^3$), and N_p is the total number of particles in the simulation (64^3).

$$\rho_{crit} = \frac{3H^2}{8\pi G} \quad (3.6)$$

where H is the Hubble parameter and G is the gravitational constant. In the units used by ROCKSTAR, this is expressed as:

$$\rho_{crit} = 2.775 \times 10^{11} \left(\frac{M_\odot}{h} \right) / \left(\frac{\text{Mpc}}{h} \right)^3, \quad (3.7)$$

This formulation ensures consistency with the cosmological background density used in simulations.

The estimated halo count for each bin can now be obtained by calculating the total mass of all particles per class i and dividing it by the average halo mass $\bar{M}_{halo,i}$:

$$N_{halo,i} = \frac{N_{p,i} M_p}{\bar{M}_{halo,i}} \quad (3.8)$$

Class 0 can be ignored for this calculation as it only represents the background. With this, the HMF can be calculated exactly as before, mentioned in Section 3.1. The halo count $N_{halo,i}$ is divided by the box size V_{box} to get the number density of haloes n_i . This divided by the logarithmic width of the bins $\Delta \ln M$ represents the HMF:

$$\text{HMF}_i = \frac{N_{halo,i}}{V_{box} \Delta \ln M} = \frac{n_i}{\Delta \ln M} \quad (3.9)$$

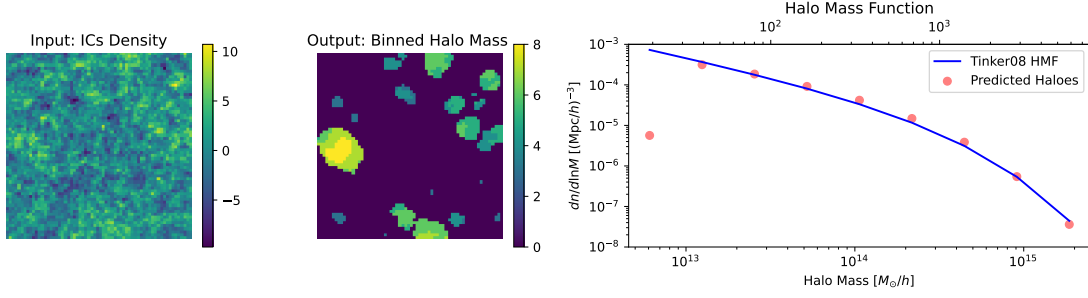


Fig. 3.15. A protohalo prediction and the resulting HMF compared to the theoretical line. The left plot shows the initial density field at $z = 49$ generated with DISCO-DJ. The middle is the network prediction of the binned protohalo patches (also $z = 49$). The right image shows the halo mass function. The red points are the results from the prediction, while the blue line indicates the analytical Tinker08 line. The halo mass is shown on the lower x-axis, while the corresponding particle number is shown at the top. Other than the first mass bin, all points are extremely close to the expected values.

Figure 3.15 shows how the prediction fares against the expected distribution of haloes. The initial density field is not processed in this visualization, as it is not part of any previously generated dataset. Instead, it is shown directly as generated by Disco. The U-Net takes care of the usual data preparation internally. Similarly to the visual comparison of the protohalo patches, the smallest haloes are clearly under-represented. This is likely due to the uncertainty of these haloes. Especially at the current resolution, they contain as little as 20 particles. It is possible that a higher resolution network would be able to handle this bin better. Nonetheless, the totaled probabilities of the other mass bins are able to predict the HMF better than anticipated, achieving a mean difference of just 5.41% for all bins. It seems to follow the theoretical Tinker08 line very closely. This means that even though the model is not able to exactly assign each particle correctly, the overall distribution of probabilities matches almost perfectly. Therefore, the HMF should be a reliable statistic to measure the dependence of cosmological parameters, which is only made possible by this novel approach for a completely differentiable dark matter halo identification.

Figure 3.16 visualizes how much the analytical HMF changes within the parameter limits that were used to train the U-Net (see table 3.2), and whether the network prediction can capture it equally well across the entire range. The initial conditions used for the predictions leave the seed and all parameters unchanged, with the exception of the one for which the limits are tested. The first mass bin should not be taken into account for the reasons mentioned above. Generally, the prediction seems to match the expected line quite well, though it is difficult to tell for some of the parameters, since the effect on the HMF is so small. It is noteworthy that Ω_k , w_0 , w_a are not taken into account by the Tinker08 HMF (Tinker et al., 2008). While they are input parameters

3 Dark Matter Haloes

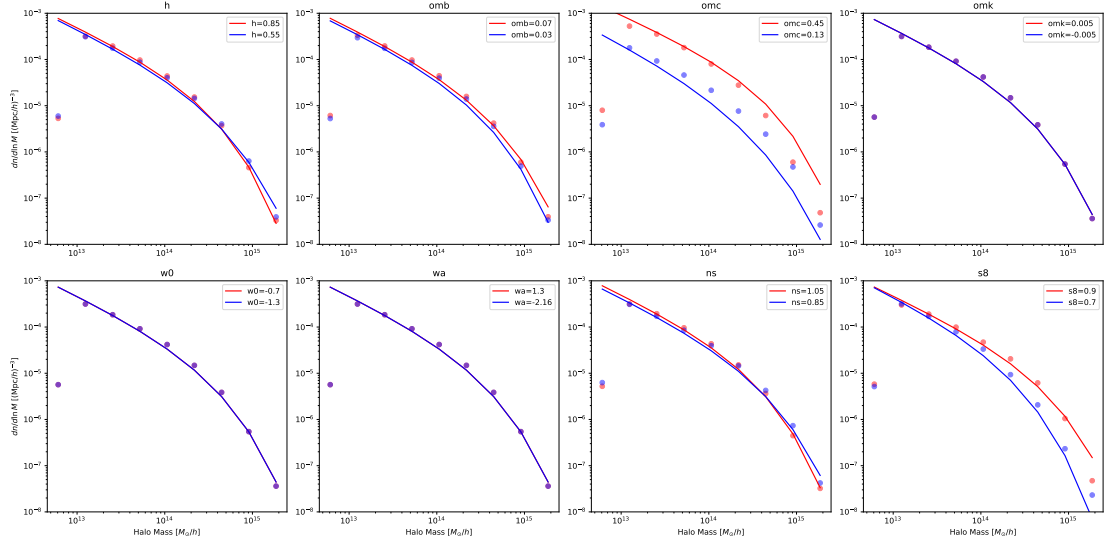


Fig. 3.16. The HMF for all parameter limits of the Latin hypercube that was used to train the neural network. Each plot shows the lower (blue) and upper (red) limits for a different cosmological parameter. Ω_c and σ_8 have the strongest effect within their bounds. h and n_s have a slightly inverse relation for higher masses. In general, the predictions (points) align very well with the Tinker08 line for a single specific seed.

for DISCO-DJ, the current version of the code does not have them implemented, so no change is expected for these three parameters. Nonetheless, they are part of the cosmological parameters, and while they are not expected to show strong influence on the HMF, it would still be interesting to implement this in the future. Furthermore, they are used for later comparisons to results from a separate HMF emulator, so they are still shown for completeness. Ω_c and σ_8 have the largest effects within their bounds, which are matched relatively well by the model predictions, although the dependence is slightly underestimated for Ω_c . Nonetheless, it is clear that an increase in Ω_c leads to a higher number density of haloes across all mass bins. This makes sense as there is more dark matter to form the haloes. Meanwhile, a higher value of σ_8 increases the number density of larger haloes more than those with smaller masses. A higher amplitude of the primordial fluctuations leads to more haloes forming earlier and growing larger, which affects the high-mass end more (Cui, 2024).

Overall, the model holds well against different cosmological parameters within the Latin hypercube used for training. The HMF being most affected by Ω_c and σ_8 also agrees with the literature (e.g., Hung et al., 2021). The next section will look directly at the differentiability aspect of this neural network approach.

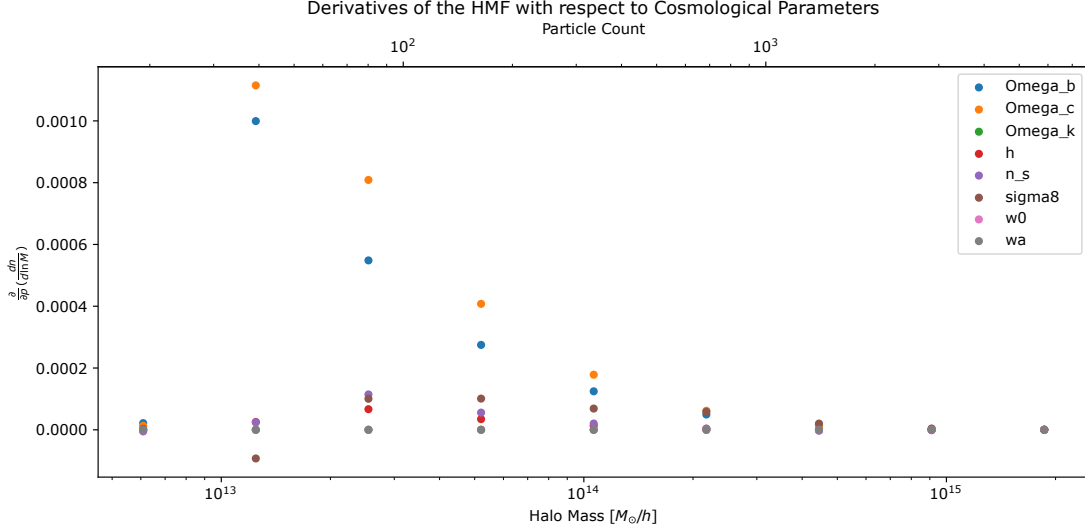


Fig. 3.17. This sensitivity of the halo mass function with respect to the cosmological parameters p . It shows the derivative of the number density for each mass bin and parameter. The halo mass and corresponding particle count is shown along the x-axis. Ω_c and Ω_b show by far the strongest effect on the HMF.

3.3.2 Differentiability

Since the entire process is differentiable and written in JAX (Bradbury et al., 2018), the derivative can be easily computed using `jax.jacfwf()`. This function evaluates the Jacobian using the forward-mode of the automatic differentiation.

Figure 3.17 shows the derivative of the HMF with respect to cosmological parameters. The effect of the parameters on each mass bin is directly compared. The first bin should be ignored due to the unreliability of the model in that range. However, the other bins show some clear effects. There is a general trend of lower masses being affected more strongly, which is likely because number densities decrease toward more massive haloes. It is only logical that the amount these values can exhibit change also decreases with each mass bin. The strongest dependence of the HMF is on Ω_c and Ω_b , which are also the only parameters that have an effect on particle mass for each simulation. Therefore, it makes sense they affect the halo mass function the most.

It might be confusing that Ω_b has a larger effect than σ_8 on the HMF, especially when looking at Fig. 3.16. However, one has to take into account the range of the allowed parameters. The bounds of σ_8 are five times as far apart as the bounds of Ω_b , which is why the absolute change is greater. Nonetheless, Fig. 3.17 is capable of demonstrating the relative change of the HMF through a numerical simulation, which was not possible before.

Furthermore, it also reveals that almost all derivatives are above 0, which means an

3 Dark Matter Haloes

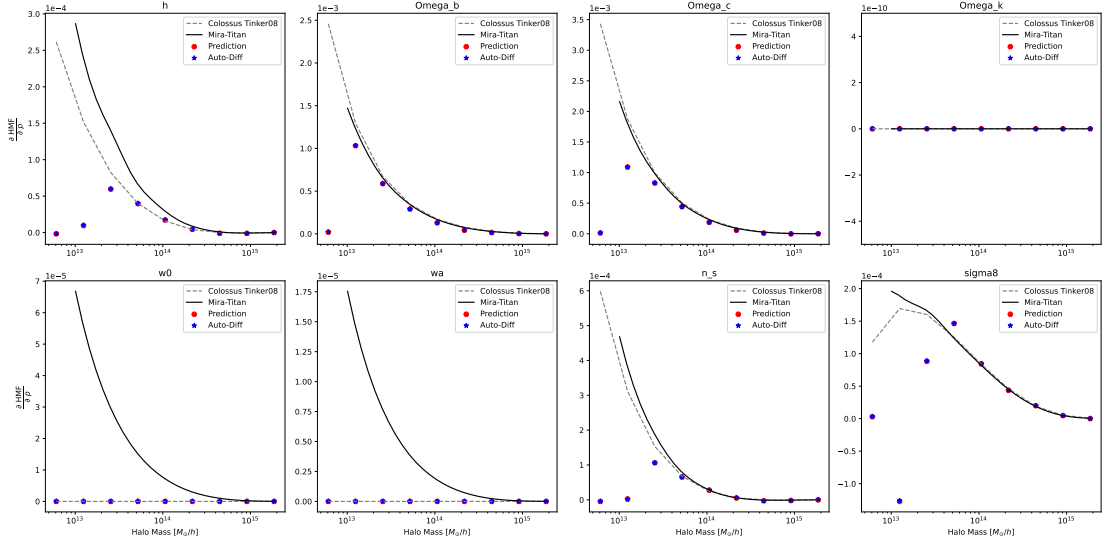


Fig. 3.18. These plots show the sensitivity of the HMF bins to different cosmological parameters. The automatic differentiation (blue stars) of the neural network is compared to the finite difference approach of the same model (red points), Mira-Titan (black line) and Tinker08 (dashed line).

increase in parameter value also increases the number density of haloes at any given size. The only significant outlier to this is the second bin of σ_8 , though h and n_s slightly drop below 0 for higher masses. Since σ_8 just shows this effect for one bin and none of the others, it is possible that the network is mistaken. In order to analyze the effects and accuracy more carefully, the derivatives will be compared to finite differences of the theoretical line.

The finite difference approximation for the derivative is computed using the central difference method. This is given by:

$$\left(\frac{\partial f}{\partial p}\right)_{\text{finite}} = \frac{f(p + \epsilon) - f(p - \epsilon)}{2\epsilon}, \quad (3.10)$$

where f is the HMF, p represents the cosmological parameter, and ϵ is a small perturbation to p . The standard value for p is based on the CAMELS parameters (Villaescusa-Navarro, Anglés-Alcázar et al., 2021) because they lie well within the bounds. ϵ is chosen as 10^{-3} so that none of the limits are exceeded.

This is done for the theoretical Tinker08 line and the model prediction by simply running separate simulations with $f(p \pm \epsilon)$. The other parameters and the seed are left unchanged. Both of these values can then be compared to the automatic differentiation by JAX.

Figure 3.18 shows the comparison of finite differences and automatic differentiation

for multiple methods. As expected, both model data points align extremely well. This just means that the neural network behaves predictably in the range of the chosen ϵ (i.e., the difference of two simulations is similar to the derivative of a single one in between). Unfortunately, the current version of DISCO-DJ does not have a functioning dependence of Ω_k , w_0 or w_a , so these values are expected to be 0 in all bins. Furthermore, w_0 and w_a do not affect the initial density field, so the U-Net would need additional input information about the growth rate, as well as the change of growth rate. Nonetheless, the parameters are still shown for comparison with the other approaches. They are not known to affect the HMF greatly either way, but a future version of Disco and a newly trained CNN with extra information about the growth rate would be able to derive those values as well.

The external HMFs against which the neural network values are compared are created with two separate Python packages. Mira-Titan (Bocquet et al., 2020) uses an emulator-based approach grounded in extensive N-body simulations, whereas COLOSSUS (Diemer, 2018) uses analytical fitting functions, such as Tinker08. Mira-Titan is based on actual simulations and does not depend on any theoretical approach, so it might be argued that these values can be trusted more. However, it only works with M_{200c} instead of M_{vir} , so the masses have to be converted, which is accomplished with COLOSSUS. Furthermore, the Mira-Titan Universe Suite of simulations only encompasses a flat universe $\Omega_k = 0$, making this parameter dependence impossible to compare. However, even though COLOSSUS is able to handle every single parameter, the Tinker08 HMF itself does not account for Ω_k , w_0 , or w_a (Tinker et al., 2008). Lastly, Mira-Titan is not able to emulate masses below $10^{13} M_\odot/h$. This is less problematic though, as it only affects the first bin of the neural network, which is not working well enough either way. In order to have every option, both lines and all parameters are kept for comparison.

Mira-Titan and Tinker08 mostly agree on the parameter dependence, with slight differences at low masses for h and σ_8 . Due to the missing dependence of the current version, w_0 and w_a show no effect on the HMF for the neural network (and similarly, Tinker08) approach, but a clear relation for Mira-Titan. Even though the correlation is quite weak (on scales of just 10^{-5}), it shows that a future version of the neural network approach (trained on N-body code with a functioning dependence and additional input about the growth rate) should also learn these features.

The other U-Net derivatives seem to capture the behavior quite well. It now shows that it is not only the first mass bin that is problematic, but rather the first 2-3. Ω_b and Ω_c arguably work best, with even the second bin being quite close to the lines. h and n_s match above the third mass bin, though h fits the Tinker08 line better than the Mira-Titan HMF emulator. σ_8 seems to be the most problematic of the parameters with a stronger effect. It is now clear that the second bin is incorrect and there should be no negative values for this parameter. The reason for this outlier is not clear, but it is likely that a higher resolution would increase the accuracy at lower masses. Overall, both neural network approaches (automatic differentiation and finite difference) are not perfect but still able to capture the general trend of the relation between cosmological

3 Dark Matter Haloes

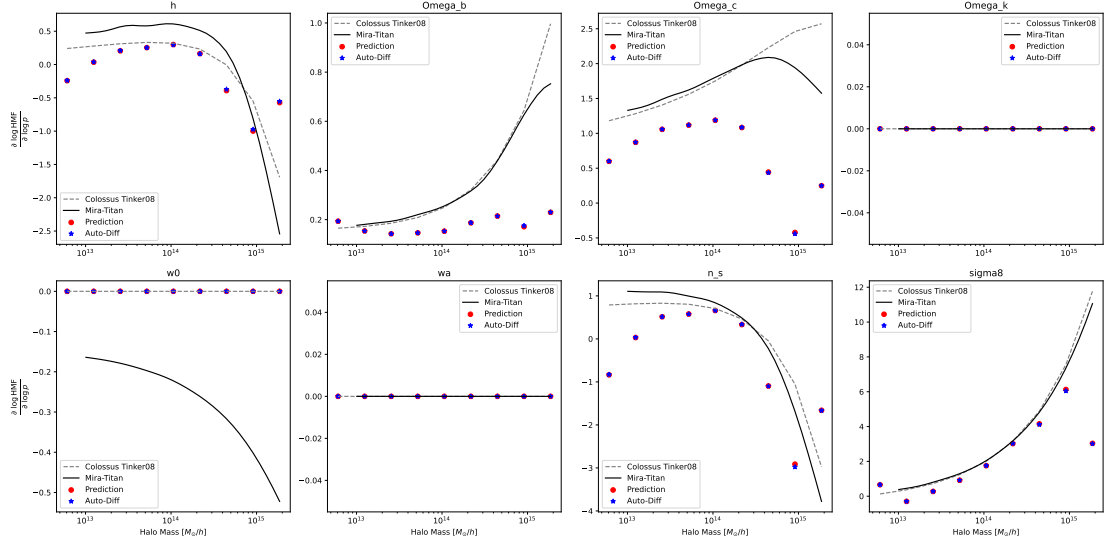


Fig. 3.19. These plots show the **normalized/dimensionless** sensitivity of the HMF bins to different cosmological parameters. The automatic differentiation (blue stars) of the neural network is compared to the finite difference approach of the same model (red points), Mira-Titan (black line) and Tinker08 (dashed line).

parameters and the halo mass function.

Since the magnitude of the derivatives vary so much for each parameter, it is often difficult to compare the results directly. In order to reduce that effect, the derivatives can be normalized by the unperturbed function value $f_0 = f(p)$ and scaled by the parameter value p to ensure dimensionless results.

$$\left(\frac{\partial \log f}{\partial \log p} \right) = \frac{p}{f} \frac{\partial f}{\partial p} \quad (3.11)$$

The approach is the same for finite differences and automatic differentiation. The HMF must be evaluated three times for the finite difference ($f(p)$, $f(p + \epsilon)$, $f(p - \epsilon)$), while automatic differentiation can be done in just one step with JAX.

Figure 3.19 shows the same derivatives as before, but rescaled as logarithmic derivatives to account for differences in parameter magnitudes. This way, the effect of cosmological parameters can be compared more easily. The y-axis is still scaled differently, but not by multiple orders of magnitude anymore. Hence, one can immediately tell which parameters have a weak effect on the HMF and which a strong one.

Ω_k is currently not implemented in any of the methods, but might be in the future, so it is shown for completeness of the parameter space. While w_0 and w_a are also not taken into account by Tinker08 or the U-Net, they are working for Mira-Titan. However, it

is now obvious that the dependence of w_a is simply too weak to be noticeable. w_0 on the other hand, seems to affect the HMF slightly more, which shows the importance of implementing it into the N-body simulator for generating more accurate results.

Ω_b and Ω_c , which both demonstrated strong influence on the HMF, seemed to match better for the direct derivatives of Fig. 3.18. This is likely because they affect the HMF so strongly, which was able to hide their inaccuracy better. Moreover, the deviation from the theoretical lines was also visible in Fig. 3.16, which showed an underestimation for the dependence of Ω_c on the HMF.

Interestingly, the neural network based derivatives of Ω_c are vertically offset, but the general trend is still fairly close to the theoretical lines. This parameter also exhibits a completely different trend at high masses between Tinker08 and Mira-Titan. The emulator shows a decreasing correlation for haloes with $M_{vir} \gtrsim 4 \times 10^{14} M_\odot/h$, which is not apparent in the Tinker08 HMF. The U-Net trend aligns better with Mira-Titan at these masses, though the last bin opposes this trend again.

Both h and n_s underestimate the first 2 mass intervals and overestimate the last one, but they are able to reproduce the negative derivative quite well. This is also obvious in Figure 3.16, though the difference is quite small. All methods agree that the number density of haloes $M_{vir} \gtrsim 2 \times 10^{14} M_\odot/h$ decreases for larger values of these parameters. As was already seen in Fig. 3.18, the U-Net derivatives of h match better with the Tinker08 HMF than with the Mira-Titan emulator, although the reasons are not clear. Out of all the cosmological parameters, σ_8 has by far the strongest dependence in this dimensionless approach. It also clearly shows the best agreement between the model derivatives and the comparison lines. Similarly to the other parameters, the last mass bins deviates further from Mira-Titan and Tinker08. The rest of the masses agree very well. Even the second bin, which was a clear outlier in Fig. 3.18, is only slightly underestimated in this visualization.

Overall, the derivatives seem to align consistently for each cosmological parameter. While they are not perfect, the fact that the neural network is able to capture most correlations proves that this approach for a differentiable halo finder is successful.

3.4 Discussion

As shown, the CNN approach certainly has its applications and there are many more that have not yet been analyzed. For example, one could look at the spatial distribution of a specific class or use the Lagrangian shapes of haloes to determine their spin (White, 1984). The previous tests only encompassed the entire simulation box at once, but it is also possible to select a subset of particles (e.g., a large protohalo patch) and see how they alone get affected by varying cosmological parameters. There are surely many more possibilities for a differentiable structure finder, but one also has to be careful with the downfalls of this current approach. For one, the predicted patches are smoother than the N-body counterparts, which means the neural network struggles with the complexity and

3 Dark Matter Haloes

uncertainty of assigning border particles to the halo. Furthermore, individual protohalo patches cannot be separated easily, since the particles are assigned to multiple different mass bins at once. While this ensures differentiability, it also means that the model does not necessarily produce physically accurate patches. It is theoretically possible for the U-Net to predict a halo of the highest mass class with only a single particle. There is no direct connection between number of particles and mass of halo in the prediction, though it clearly exists in the simulated cases ($M_{halo} = \sum_{p \in \text{Halo}} (N_p M_p)$). However, one could possibly post-process the prediction by not just looking at the highest probability for each particle, but instead also taking into account neighboring voxels to create haloes with matching mass bin and particle count. Though, this would likely not be differentiable.

Along the way, a few more ideas for future improvements became apparent. The most obvious is a higher resolution. While 64^3 worked well as a proof of concept, many details are not resolved, and the smallest haloes contain so few particles that the CNN struggles to predict them at all. With the current hardware, a 128^3 version is possible, but with all steps taking 8 times longer and time constraints, it was not fully tested. Ideally, one would use even more powerful hardware and go directly to 256^3 simulations, but the efficiency of training data creation and training itself would have to be improved in order to fully train the model in a reasonable span of time. Maybe one could then also increase the number of mass bins and hence classes to get more data points for the HMF. A future paper will be released documenting an improved model with a resolution of 256^3 and more mass bins, while providing even better predictions.

Another possible implementation for the future are periodic boundaries. These are used for the actual simulation box, but the U-Net does not actively know about them when applying the convolutions with zero padding. This leads to larger protohalo patches that would normally be connected through the periodic edge, to be viewed as separated smaller haloes. This could theoretically be implemented through periodic padding and different down-/up-sampling methods, but there are some benefits to not implementing periodic boundaries into the CNN.

For one, with the current setup, it is possible to tile a larger simulation box into multiple predictions of the smaller resolution. With this method, a simulation box of size 128^3 could be split up into 8 equal smaller boxes and use each one as input for the neural network before combining the output once more to get back to the original resolution. While this approach is not perfect because the border regions are missing information, it should still work better than with a periodic model. In such a case, the U-Net should have been trained with a loss that does not take the border of the simulation box into account. Instead, it could have simply looked at an inner subsection of the box. In the end, it depends on the exact applications of the model. For lower resolution, the non-periodicity makes more sense because tile-ability is more important, but a higher resolution network should probably use periodic wrapping for the convolutions.

Looking at the parameter dependence of the halo mass function, it is clear that implementing a working w_0 into the N-body code would benefit a possible dependence

analysis. w_a seems to have a negligible effect on the HMF, and Ω_k could not be tested because Mira-Titan itself only used a flat universe. The standard cosmological parameters (h , Ω_b , Ω_c , n_s and σ_8) are already working quite well in this approach, and are likely the most interesting derivatives either way, but having these other parameters implemented would still be beneficial.

A different question that came up during discussions concerned the input of the CNN. Is the density field enough information, or should the parameters also be included directly? In the end, it was agreed that the neural network can be more flexible if the parameters are simply inferred from the initial conditions. This means that this model can be combined with any numerical simulation no matter the choice of parameters. However, this also limits the dependence to cosmological parameters that can be inferred directly from the initial density field, which excludes w_0 and w_a as they would need additional information about the growth rate. There might be some advantages to also include the potential, though that can be calculated from the density field directly, so there is no additional information. Maybe a combination of position and velocity of particles would increase the accuracy, which shows there are still multiple options one could try.

Overall, even though the neural network approach for dark matter haloes is not without its faults, the benefits of having immediate access to the derivatives clearly outweigh them. Especially because this approach can still be improved and should so far only be seen as a proof of concept. Yet, being able to use the protohalo patch predictions directly without any post-processing is already an important accomplishment and surely has many more applications in this field.

4 Conclusion

The main goal of this thesis was implementing differentiable structure finding methods into numerical dark matter simulations and analyzing the benefits of readily available derivatives. This was split into two different parts, covering very different scales of the universe. On a larger scale, the cosmic web was classified into its distinct environments. Meanwhile, dark matter protohalo patches, which cover much smaller scales, were also identified in a differentiable manner. This allowed both types of structure to be found and their dependence on cosmological parameters to be analyzed directly, without having to run multiple simulations.

We classified the cosmic web using a method known as ‘T-web’ (Hahn, Porciani et al., 2007), which is able to identify four distinct environments: voids, sheets, filaments, and clusters. It works by determining the number of positive eigenvalues of the tidal field tensor, which describes how the gravitational potential deforms in different spatial directions. The N-body simulations were computed using DISCO-DJ, which stands for **D**ifferentiable **S**imulations for **C**OsMology - **D**one with **J**ax (Hahn and List, 2025). It was used for generating initial conditions at $z = 49$ and evolving them to $z = 0$ employing both Lagrangian Perturbation Theory (LPT) and the particle-mesh N-body code. We continued to use JAX (Bradbury et al., 2018) modules such as `jax.numpy` to maintain differentiability throughout any post-processing. The gravitational potential was derived from the particle positions in order to construct the tidal field tensor and compute its eigenvalues. By applying the T-web formalism on initial conditions first, we were able to confirm its correct implementation. This works because the volume fraction for each environment is known for a Gaussian random field, which is the case for the initial matter distribution. Our results were within 0.2% of expected volume fractions, which are 8% for clusters and voids, and 42% for sheets and filaments (Doroshkevich, 1970). The slight difference is only due to a finite box size.

The matter fields were evolved with second-order LPT and N-body. Both results were classified using T-web and their environments compared. We were able to show that the more accurate N-body method produced more concentrated structures, as expected.

Furthermore, we were able to compute the dependence of the eigenvalues on the cosmological parameters. We evaluated the Jacobian using the forward-mode of the automatic differentiation implemented in JAX. By adding these derivatives linearly to a single simulation, we were able to produce the cosmic web for slightly different parameters without having to rerun the N-body code. However, this linear dependence quickly breaks down for larger changes in the initial conditions. Nonetheless, it is possible to see the

4 Conclusion

effect each parameter has on the individual environments. Additionally, we discussed a few very promising applications for this implementation of a differentiable method for identifying cosmic web structures. The main advantage of relying on differentiable simulations is having direct access to the derivatives, which makes Fisher forecasting for parameter constraints much more efficient (Campagne et al., 2023). By combining this with environment-specific analysis, it is possible to obtain tighter constraints for parameters of the Λ CDM model (Bonnaire et al., 2021). We did not have enough time to investigate this further, but we were nonetheless able to lay the foundation for future research.

The second and main part of this thesis deals with the development of a differentiable dark matter halo finder. There are many different halo finders (Knebe et al., 2013), but none of them are differentiable because it is inherently difficult to implement. We were able to achieve this through the use of a convolutional neural network (CNN). While there were already similar approaches (e.g., Bernardini et al., 2020, Berger and Stein, 2019, López-Cano et al., 2024), none of them are completely differentiable, as they have to post-process the model output. However, their goal was to be as accurate as possible while utilizing the immense time improvements of neural networks, which they were all able to achieve. We, on the other hand, focus first and foremost on the differentiability aspect of this approach.

This is accomplished by training a U-Net model to classify initial density fields into mass binned protohalo patches. The network is able to predict which particles will end up in a halo of a certain virial mass range. We created training data by generating initial conditions with DISCO-DJ at $z = 49$. Due to DISCO-DJ being a PM code, it underestimated the small-scale forces, which leads to fewer haloes than expected by theoretical approximations, especially at this low spatial resolution. Therefore, the N-body calculations were executed with GADGET-4 (Springel et al., 2021) to reach $z = 0$. ROCKSTAR was then used to identify the haloes through a FoF algorithm. We assigned each halo’s mass to the particles within their virial radius. By simply returning the particles to their original position and binning the masses into 10 classes (with 0 representing the background), we were able to create the label used for training. A total of 500 simulations with different cosmological parameters were created to reinforce the parameter dependence. The training data was augmented by changing their orientation, resulting in 24,000 pseudo-unique simulation boxes. The model was trained until it stopped improving.

The CNN was able to successfully predict the protohalo patches and assign the halo mass to its particles. It generally struggled with lower-mass haloes, due to their unpredictability. Furthermore, the probabilistic approach of the particle classification led to patches being assigned a higher mass inside and lower around the edges, which reduces the loss in case of wrong predictions. While this is not ideal for exact halo identification, it does enable differentiability. We were able to show that the output can be used directly to calculate the halo mass function. Except for the first mass bin, the distribution closely followed the theoretical Tinker08 HMF (Tinker et al., 2008). The dependence of cosmological parameters was analyzed using different approaches. First, the model

successfully reproduced the HMF accounting for the entire parameter space. Secondly, the derivative of the halo mass function with respect to the cosmological parameters was calculated and compared to that of the Mira-Titan HMF emulator (Bocquet et al., 2020), as well as the Tinker08 HMF. It was not possible to evaluate the effect of Ω_k , w_0 , or w_a from the network approach due to incomplete implementation in DISCO-DJ. However, we were able to replicate the dependence of the classical Λ CDM parameters (h , Ω_b , Ω_c , n_s and σ_8) quite well, especially for higher mass bins. We showed that Ω_b , Ω_c , and σ_8 affect the HMF the most out of these parameters within their realistic limits. The dimensionless derivatives were very accurate for σ_8 , but a little offset for Ω_c . There was room for improvement regarding Ω_b . Although not perfect, the U-Net learned the inverse relationship of h and n_s for haloes above $10^{14}M_\odot/h$.

With this in mind, the neural network approach clearly proved its effectiveness as a differentiable dark matter halo finder. We also discussed future changes to further improve this method. Additionally, the analysis does not have to be limited to just the HMF but a similar pipeline can be used to look at spatial distribution of a specific class or examine Lagrangian shapes. We successfully showed that the U-Net can predict protohalo patches without the need of post-processing, while simultaneously making the derivatives available.

In conclusion, we were able to achieve the goals of this thesis and create differentiable structure finding methods. Since cosmic environments can be classified with T-web in such a manner without any loss of information, it is clearly advisable. The immediate access to derivatives makes parameter constraints easier and more efficient. Including environment-specific analysis can even improve these constraints (Bonnaire et al., 2021). The U-Net presents the first halo finder of its kind that can be differentiated completely. However, this method is not without its trade-offs. The exact positions and shapes of haloes are not recovered, so the implementation depends on the application. However, the HMF can be predicted very well, making this approach clearly viable for estimating its dependence on cosmological parameters. Overall, differentiable structure finding has many benefits for cosmology, which surely makes it an important area in research for years to come.

Bibliography

- Aarseth, S. J. and F. Hoyle (June 1963). ‘Dynamical Evolution of Clusters of Galaxies, I’. In: *Monthly Notices of the Royal Astronomical Society* 126.3, pp. 223–255. ISSN: 0035-8711. DOI: [10.1093/mnras/126.3.223](https://doi.org/10.1093/mnras/126.3.223). eprint: <https://academic.oup.com/mnras/article-pdf/126/3/223/8074955/mnras126-0223.pdf>. URL: <https://doi.org/10.1093/mnras/126.3.223>.
- Addison, G. E., G. Hinshaw and M. Halpern (Dec. 2013). ‘Cosmological constraints from baryon acoustic oscillations and clustering of large-scale structure’. In: *Monthly Notices of the Royal Astronomical Society* 436.2, pp. 1674–1683. DOI: [10.1093/mnras/stt1687](https://doi.org/10.1093/mnras/stt1687). arXiv: [1304.6984](https://arxiv.org/abs/1304.6984) [astro-ph.CO].
- Akita, Kensuke and Masahide Yamaguchi (Oct. 2022). ‘A Review of Neutrino Decoupling from the Early Universe to the Current Universe’. In: *Universe* 8.11, 552, p. 552. DOI: [10.3390/universe8110552](https://doi.org/10.3390/universe8110552). arXiv: [2210.10307](https://arxiv.org/abs/2210.10307) [hep-ph].
- Angulo, Raul E. and Oliver Hahn (Dec. 2022). ‘Large-scale dark matter simulations’. In: *Living Reviews in Computational Astrophysics* 8.1, 1, p. 1. DOI: [10.1007/s41115-021-00013-z](https://doi.org/10.1007/s41115-021-00013-z). arXiv: [2112.05165](https://arxiv.org/abs/2112.05165) [astro-ph.CO].
- Azad, Reza et al. (2022). ‘Medical Image Segmentation Review: The Success of U-Net’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, pp. 10076–10095. URL: <https://api.semanticscholar.org/CorpusID:254043694>.
- Bardeen, J. M., J. R. Bond, N. Kaiser and A. S. Szalay (May 1986). ‘The Statistics of Peaks of Gaussian Random Fields’. In: *Astrophysical Journal* 304, p. 15. DOI: [10.1086/164143](https://doi.org/10.1086/164143).
- Barnes, Josh and Piet Hut (Dec. 1986). ‘A hierarchical $O(N \log N)$ force-calculation algorithm’. In: *Nature* 324.6096, pp. 446–449. DOI: [10.1038/324446a0](https://doi.org/10.1038/324446a0).
- Baumann, Daniel (July 2009). ‘TASI Lectures on Inflation’. In: *arXiv e-prints*, arXiv:0907.5424, arXiv:0907.5424. DOI: [10.48550/arXiv.0907.5424](https://doi.org/10.48550/arXiv.0907.5424). arXiv: [0907.5424](https://arxiv.org/abs/0907.5424) [hep-th].
- Behroozi, Peter S., Risa H. Wechsler and Hao-Yi Wu (Jan. 2013). ‘The ROCKSTAR Phase-space Temporal Halo Finder and the Velocity Offsets of Cluster Cores’. In: *Astrophysical Journal* 762.2, 109, p. 109. DOI: [10.1088/0004-637X/762/2/109](https://doi.org/10.1088/0004-637X/762/2/109). arXiv: [1110.4372](https://arxiv.org/abs/1110.4372) [astro-ph.CO].
- Berger, Philippe and George Stein (Jan. 2019). ‘A volumetric deep Convolutional Neural Network for simulation of mock dark matter halo catalogues’. In: *Monthly Notices of the Royal Astronomical Society* 482.3, pp. 2861–2871. DOI: [10.1093/mnras/sty2949](https://doi.org/10.1093/mnras/sty2949). arXiv: [1805.04537](https://arxiv.org/abs/1805.04537) [astro-ph.CO].

Bibliography

- Bernardeau, F., S. Colombi, E. Gaztañaga and R. Scoccimarro (Sept. 2002). ‘Large-scale structure of the Universe and cosmological perturbation theory’. In: *Physics Reports* 367.1-3, pp. 1–248. DOI: [10.1016/S0370-1573\(02\)00135-7](https://doi.org/10.1016/S0370-1573(02)00135-7). arXiv: [astro-ph/0112551](https://arxiv.org/abs/astro-ph/0112551) [[astro-ph](#)].
- Bernardini, M., L. Mayer, D. Reed and R. Feldmann (Aug. 2020). ‘Predicting dark matter halo formation in N-body simulations with deep regression networks’. In: *Monthly Notices of the Royal Astronomical Society* 496.4, pp. 5116–5125. DOI: [10.1093/mnras/staa1911](https://doi.org/10.1093/mnras/staa1911). arXiv: [1912.04299](https://arxiv.org/abs/1912.04299) [[astro-ph.CO](#)].
- Bertone, Gianfranco, Dan Hooper and Joseph Silk (Jan. 2005). ‘Particle dark matter: evidence, candidates and constraints’. In: *Physics Reports* 405.5-6, pp. 279–390. DOI: [10.1016/j.physrep.2004.08.031](https://doi.org/10.1016/j.physrep.2004.08.031). arXiv: [hep-ph/0404175](https://arxiv.org/abs/hep-ph/0404175) [[hep-ph](#)].
- Bertschinger, Edmund (Nov. 2001). ‘Multiscale Gaussian Random Fields and Their Application to Cosmological Simulations’. In: *Astrophysical Journal Supplement* 137.1, pp. 1–20. DOI: [10.1086/322526](https://doi.org/10.1086/322526). arXiv: [astro-ph/0103301](https://arxiv.org/abs/astro-ph/0103301) [[astro-ph](#)].
- Betancourt, Michael (Jan. 2017). ‘A Conceptual Introduction to Hamiltonian Monte Carlo’. In: *arXiv e-prints*, arXiv:1701.02434, arXiv:1701.02434. DOI: [10.48550/arXiv.1701.02434](https://doi.org/10.48550/arXiv.1701.02434). arXiv: [1701.02434](https://arxiv.org/abs/1701.02434) [[stat.ME](#)].
- Bocquet, Sebastian, Katrin Heitmann, Salman Habib, Earl Lawrence, Thomas Uram, Nicholas Frontiere, Adrian Pope and Hal Finkel (Sept. 2020). ‘The Mira-Titan Universe. III. Emulation of the Halo Mass Function’. In: *Astrophysical Journal* 901.1, 5, p. 5. DOI: [10.3847/1538-4357/abac5c](https://doi.org/10.3847/1538-4357/abac5c). arXiv: [2003.12116](https://arxiv.org/abs/2003.12116) [[astro-ph.CO](#)].
- Bond, J. R., S. Cole, G. Efsthathiou and N. Kaiser (Oct. 1991). ‘Excursion Set Mass Functions for Hierarchical Gaussian Fluctuations’. In: *Astrophysical Journal* 379, p. 440. DOI: [10.1086/170520](https://doi.org/10.1086/170520).
- Bond, J. R. and S. T. Myers (Mar. 1996). ‘The Peak-Patch Picture of Cosmic Catalogs. I. Algorithms’. In: *Astrophysical Journal Supplement* 103, p. 1. DOI: [10.1086/192267](https://doi.org/10.1086/192267).
- Bond, J. Richard, Lev Kofman and Dmitry Pogosyan (Apr. 1996). ‘How filaments of galaxies are woven into the cosmic web’. In: *Nature* 380.6575, pp. 603–606. DOI: [10.1038/380603a0](https://doi.org/10.1038/380603a0). arXiv: [astro-ph/9512141](https://arxiv.org/abs/astro-ph/9512141) [[astro-ph](#)].
- Bonnaire, Tony, Nabila Aghanim, Joseph Kuruvilla and Aurélien Decelle (Dec. 2021). ‘Cosmology with cosmic web environments I. Real-space power spectra’. In: *arXiv e-prints*, arXiv:2112.03926, arXiv:2112.03926. DOI: [10.48550/arXiv.2112.03926](https://doi.org/10.48550/arXiv.2112.03926). arXiv: [2112.03926](https://arxiv.org/abs/2112.03926) [[astro-ph.CO](#)].
- Bradbury, James et al. (2018). *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. URL: [http://github.com/google/jax](https://github.com/google/jax).
- Bryan, Greg L. and Michael L. Norman (Mar. 1998). ‘Statistical Properties of X-Ray Clusters: Analytic and Numerical Comparisons’. In: *Astrophysical Journal* 495.1, pp. 80–99. DOI: [10.1086/305262](https://doi.org/10.1086/305262). arXiv: [astro-ph/9710107](https://arxiv.org/abs/astro-ph/9710107) [[astro-ph](#)].
- Bull, Philip et al. (June 2016). ‘Beyond Λ CDM: Problems, solutions, and the road ahead’. In: *Physics of the Dark Universe* 12, pp. 56–99. DOI: [10.1016/j.dark.2016.02.001](https://doi.org/10.1016/j.dark.2016.02.001). arXiv: [1512.05356](https://arxiv.org/abs/1512.05356) [[astro-ph.CO](#)].

- Campagne, Jean-Eric et al. (Apr. 2023). ‘JAX-COSMO: An End-to-End Differentiable and GPU Accelerated Cosmology Library’. In: *The Open Journal of Astrophysics* 6, 15, p. 15. DOI: [10.21105/astro.2302.05163](https://doi.org/10.21105/astro.2302.05163). arXiv: [2302.05163](https://arxiv.org/abs/2302.05163) [astro-ph.CO].
- Carroll, Sean M. (Feb. 2001). ‘The Cosmological Constant’. In: *Living Reviews in Relativity* 4.1, 1, p. 1. DOI: [10.12942/lrr-2001-1](https://doi.org/10.12942/lrr-2001-1). arXiv: [astro-ph/0004075](https://arxiv.org/abs/astro-ph/0004075) [astro-ph].
- Carroll, Sean M., William H. Press and Edwin L. Turner (Jan. 1992). ‘The cosmological constant.’ In: *Annual Review of Astronomy and Astrophysics* 30, pp. 499–542. DOI: [10.1146/annurev.aa.30.090192.002435](https://doi.org/10.1146/annurev.aa.30.090192.002435).
- Chacon, Jazhiel, J. Alberto Vazquez and Ruslan Gabbasov (June 2020). ‘Dark Matter with N-Body Numerical Simulations’. In: *arXiv e-prints*, arXiv:2006.10203, arXiv:2006.10203. DOI: [10.48550/arXiv.2006.10203](https://doi.org/10.48550/arXiv.2006.10203). arXiv: [2006.10203](https://arxiv.org/abs/2006.10203) [astro-ph.CO].
- Ciardi, Benedetta and Andrea Ferrara (Feb. 2005). ‘The First Cosmic Structures and Their Effects’. In: *Space Science Reviews* 116.3-4, pp. 625–705. DOI: [10.1007/s11214-005-3592-0](https://doi.org/10.1007/s11214-005-3592-0). arXiv: [astro-ph/0409018](https://arxiv.org/abs/astro-ph/0409018) [astro-ph].
- Colberg, Jörg M., K. Simon Krughoff and Andrew J. Connolly (May 2005). ‘Intercluster filaments in a Λ CDM Universe’. In: *Monthly Notices of the Royal Astronomical Society* 359.1, pp. 272–282. DOI: [10.1111/j.1365-2966.2005.08897.x](https://doi.org/10.1111/j.1365-2966.2005.08897.x). arXiv: [astro-ph/0406665](https://arxiv.org/abs/astro-ph/0406665) [astro-ph].
- Cole, Shaun and Cedric Lacey (July 1996). ‘The structure of dark matter haloes in hierarchical clustering models’. In: *Monthly Notices of the Royal Astronomical Society* 281, p. 716. DOI: [10.1093/mnras/281.2.716](https://doi.org/10.1093/mnras/281.2.716). arXiv: [astro-ph/9510147](https://arxiv.org/abs/astro-ph/9510147) [astro-ph].
- Cooke, Ryan J., Max Pettini, Kenneth M. Nollett and Regina Jorgenson (Oct. 2016). ‘The Primordial Deuterium Abundance of the Most Metal-poor Damped Lyman- α System’. In: *Astrophysical Journal* 830.2, 148, p. 148. DOI: [10.3847/0004-637X/830/2/148](https://doi.org/10.3847/0004-637X/830/2/148). arXiv: [1607.03900](https://arxiv.org/abs/1607.03900) [astro-ph.CO].
- Couchman, H. M. P. (Feb. 1991). ‘Mesh-refined P 3M: A Fast Adaptive N-Body Algorithm’. In: *Astrophysical Journal Letters* 368, p. L23. DOI: [10.1086/185939](https://doi.org/10.1086/185939).
- Crocce, Martín, Sebastián Pueblas and Román Scoccimarro (Nov. 2006). ‘Transients from initial conditions in cosmological simulations’. In: *Monthly Notices of the Royal Astronomical Society* 373.1, pp. 369–381. DOI: [10.1111/j.1365-2966.2006.11040.x](https://doi.org/10.1111/j.1365-2966.2006.11040.x). arXiv: [astro-ph/0606505](https://arxiv.org/abs/astro-ph/0606505) [astro-ph].
- Cui, Weiguang (June 2024). ‘How much do we know the halo mass function? Predictions beyond resolution’. In: *arXiv e-prints*, arXiv:2406.03829, arXiv:2406.03829. DOI: [10.48550/arXiv.2406.03829](https://doi.org/10.48550/arXiv.2406.03829). arXiv: [2406.03829](https://arxiv.org/abs/2406.03829) [astro-ph.CO].
- Cyburt, Richard H., Brian D. Fields, Keith A. Olive and Tsung-Han Yeh (Jan. 2016). ‘Big bang nucleosynthesis: Present status’. In: *Reviews of Modern Physics* 88.1, 015004, p. 015004. DOI: [10.1103/RevModPhys.88.015004](https://doi.org/10.1103/RevModPhys.88.015004). arXiv: [1505.01076](https://arxiv.org/abs/1505.01076) [astro-ph.CO].
- Davis, M., G. Efstathiou, C. S. Frenk and S. D. M. White (May 1985). ‘The evolution of large-scale structure in a universe dominated by cold dark matter’. In: *Astrophysical Journal* 292, pp. 371–394. DOI: [10.1086/163168](https://doi.org/10.1086/163168).
- de Lapparent, V., M. J. Geller and J. P. Huchra (Mar. 1986). ‘A Slice of the Universe’. In: *Astrophysical Journal Letters* 302, p. L1. DOI: [10.1086/184625](https://doi.org/10.1086/184625).

Bibliography

- DeepMind et al. (2020). *The DeepMind JAX Ecosystem*. URL: <http://github.com/google-deeppmind>.
- DESI Collaboration, A. G. Adame et al. (Nov. 2024). ‘DESI 2024 VII: Cosmological Constraints from the Full-Shape Modeling of Clustering Measurements’. In: *arXiv e-prints*, arXiv:2411.12022, arXiv:2411.12022. DOI: [10.48550/arXiv.2411.12022](https://doi.org/10.48550/arXiv.2411.12022). arXiv: [2411.12022](https://arxiv.org/abs/2411.12022) [astro-ph.CO].
- DESI Collaboration, Amir Aghamousa et al. (Oct. 2016). ‘The DESI Experiment Part I: Science, Targeting, and Survey Design’. In: *arXiv e-prints*, arXiv:1611.00036, arXiv:1611.00036. DOI: [10.48550/arXiv.1611.00036](https://doi.org/10.48550/arXiv.1611.00036). arXiv: [1611.00036](https://arxiv.org/abs/1611.00036) [astro-ph.IM].
- Diemer, Benedikt (Dec. 2018). ‘COLOSSUS: A Python Toolkit for Cosmology, Large-scale Structure, and Dark Matter Halos’. In: *Astrophysical Journal Supplement* 239.2, 35, p. 35. DOI: [10.3847/1538-4365/aeee8c](https://doi.org/10.3847/1538-4365/aeee8c). arXiv: [1712.04512](https://arxiv.org/abs/1712.04512) [astro-ph.CO].
- Dolag, K., S. Borgani, S. Schindler, A. Diaferio and A. M. Bykov (Feb. 2008). ‘Simulation Techniques for Cosmological Simulations’. In: *Space Science Reviews* 134.1-4, pp. 229–268. DOI: [10.1007/s11214-008-9316-5](https://doi.org/10.1007/s11214-008-9316-5). arXiv: [0801.1023](https://arxiv.org/abs/0801.1023) [astro-ph].
- Doroshkevich, A. G. (Oct. 1970). ‘Spatial structure of perturbations and origin of galactic rotation in fluctuation theory’. In: *Astrophysics* 6.4, pp. 320–330. DOI: [10.1007/BF01001625](https://doi.org/10.1007/BF01001625).
- Efstathiou, G., M. Davis, S. D. M. White and C. S. Frenk (Feb. 1985). ‘Numerical techniques for large cosmological N-body simulations’. In: *Astrophysical Journal Supplement* 57, pp. 241–260. DOI: [10.1086/191003](https://doi.org/10.1086/191003).
- Einstein, Albert (Jan. 1917). ‘Kosmologische Betrachtungen zur allgemeinen Relativitätstheorie’. In: *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften*, pp. 142–152.
- Eisenstein, Daniel J. and Piet Hut (May 1998). ‘HOP: A New Group-Finding Algorithm for N-Body Simulations’. In: *Astrophysical Journal* 498.1, pp. 137–142. DOI: [10.1086/305535](https://doi.org/10.1086/305535). arXiv: [astro-ph/9712200](https://arxiv.org/abs/astro-ph/9712200) [astro-ph].
- Euclid Collaboration et al. (June 2022). ‘Euclid preparation. I. The Euclid Wide Survey’. In: *Astronomy and Astrophysics* 662, A112, A112. DOI: [10.1051/0004-6361/202141938](https://doi.org/10.1051/0004-6361/202141938). arXiv: [2108.01201](https://arxiv.org/abs/2108.01201) [astro-ph.CO].
- Freeman, K. C. (June 1970). ‘On the Disks of Spiral and S0 Galaxies’. In: *Astrophysical Journal* 160, p. 811. DOI: [10.1086/150474](https://doi.org/10.1086/150474).
- Friedmann, A. (Jan. 1922). ‘Über die Krümmung des Raumes’. In: *Zeitschrift für Physik* 10, pp. 377–386. DOI: [10.1007/BF01332580](https://doi.org/10.1007/BF01332580).
- Greengard, L and V Rokhlin (1987). ‘A fast algorithm for particle simulations’. In: *Journal of Computational Physics* 73.2, pp. 325–348. ISSN: 0021-9991. DOI: [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9). URL: <https://www.sciencedirect.com/science/article/pii/0021999187901409>.
- Guo, Qi, Simon White, Raul E. Angulo, Bruno Henriques, Gerard Lemson, Michael Boylan-Kolchin, Peter Thomas and Chris Short (Jan. 2013). ‘Galaxy formation in WMAP1 and WMAP7 cosmologies’. In: *Monthly Notices of the Royal Astronomical*

- Society* 428.2, pp. 1351–1365. DOI: [10.1093/mnras/sts115](https://doi.org/10.1093/mnras/sts115). arXiv: [1206.0052](https://arxiv.org/abs/1206.0052) [astro-ph.CO].
- Hahn, Oliver and Tom Abel (Aug. 2011). ‘Multi-scale initial conditions for cosmological simulations’. In: *Monthly Notices of the Royal Astronomical Society* 415.3, pp. 2101–2121. DOI: [10.1111/j.1365-2966.2011.18820.x](https://doi.org/10.1111/j.1365-2966.2011.18820.x). arXiv: [1103.6031](https://arxiv.org/abs/1103.6031) [astro-ph.CO].
- Hahn, Oliver and Florian List (2025). ‘DISCO-DJ’. In: in preparation.
- Hahn, Oliver, Cristiano Porciani, C. Marcella Carollo and Avishai Dekel (Feb. 2007). ‘Properties of dark matter haloes in clusters, filaments, sheets and voids’. In: *Monthly Notices of the Royal Astronomical Society* 375.2, pp. 489–499. DOI: [10.1111/j.1365-2966.2006.11318.x](https://doi.org/10.1111/j.1365-2966.2006.11318.x). arXiv: [astro-ph/0610280](https://arxiv.org/abs/astro-ph/0610280) [astro-ph].
- Hawkins, Douglas M. (Jan. 2004). ‘The Problem of Overfitting’. English (US). In: *Journal of chemical information and computer sciences* 44.1. Copyright: Copyright 2012 Elsevier B.V., All rights reserved., pp. 1–12. ISSN: 0095-2338. DOI: [10.1021/ci0342472](https://doi.org/10.1021/ci0342472).
- Heek, Jonathan, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner and Marc van Zee (2024). *Flax: A neural network library and ecosystem for JAX*. Version 0.9.0. URL: <http://github.com/google/flax>.
- Hennigan, Tom, Trevor Cai, Tamara Norman, Lena Martens and Igor Babuschkin (2020). *Haiku: Sonnet for JAX*. Version 0.0.13. URL: <http://github.com/deepmind/dm-haiku>.
- Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever and Ruslan R. Salakhutdinov (July 2012). ‘Improving neural networks by preventing co-adaptation of feature detectors’. In: *arXiv e-prints*, arXiv:1207.0580, arXiv:1207.0580. DOI: [10.48550/arXiv.1207.0580](https://doi.org/10.48550/arXiv.1207.0580). arXiv: [1207.0580](https://arxiv.org/abs/1207.0580) [cs.NE].
- Hou, Le, Chen-Ping Yu and Dimitris Samaras (Nov. 2016). ‘Squared Earth Mover’s Distance-based Loss for Training Deep Neural Networks’. In: *arXiv e-prints*, arXiv:1611.05916, arXiv:1611.05916. DOI: [10.48550/arXiv.1611.05916](https://doi.org/10.48550/arXiv.1611.05916). arXiv: [1611.05916](https://arxiv.org/abs/1611.05916) [cs.CV].
- Hu, Jian-Ping and Fa-Yin Wang (Feb. 2023). ‘Hubble Tension: The Evidence of New Physics’. In: *Universe* 9.2, 94, p. 94. DOI: [10.3390/universe9020094](https://doi.org/10.3390/universe9020094). arXiv: [2302.05709](https://arxiv.org/abs/2302.05709) [astro-ph.CO].
- Huang, Lei, Jie Qin, Yi Zhou, Fan Zhu, Li Liu and Ling Shao (Sept. 2020). ‘Normalization Techniques in Training DNNs: Methodology, Analysis and Application’. In: *arXiv e-prints*, arXiv:2009.12836, arXiv:2009.12836. DOI: [10.48550/arXiv.2009.12836](https://doi.org/10.48550/arXiv.2009.12836). arXiv: [2009.12836](https://arxiv.org/abs/2009.12836) [cs.LG].
- Hubble, Edwin (Mar. 1929). ‘A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae’. In: *Proceedings of the National Academy of Science* 15.3, pp. 168–173. DOI: [10.1073/pnas.15.3.168](https://doi.org/10.1073/pnas.15.3.168).
- Hung, D. et al. (Apr. 2021). ‘An optical observational cluster mass function at $z \sim 1$ with the ORELSE survey’. In: *Monthly Notices of the Royal Astronomical Society* 502.3, pp. 3942–3954. DOI: [10.1093/mnras/stab300](https://doi.org/10.1093/mnras/stab300). arXiv: [2101.02215](https://arxiv.org/abs/2101.02215) [astro-ph.CO].
- Jasche, J. and G. Lavaux (May 2019). ‘Physical Bayesian modelling of the non-linear matter distribution: New insights into the nearby universe’. In: *Astronomy and*

Bibliography

- Astrophysics* 625, A64, A64. DOI: [10.1051/0004-6361/201833710](https://doi.org/10.1051/0004-6361/201833710). arXiv: [1806.11117](https://arxiv.org/abs/1806.11117) [astro-ph.CO].
- Jiang, Fangzhou and Frank C. van den Bosch (Mar. 2014). ‘Statistics of Dark Matter Substructure: I. Model and Universal Fitting Functions’. In: *arXiv e-prints*, arXiv:1403.6827, arXiv:1403.6827. DOI: [10.48550/arXiv.1403.6827](https://doi.org/10.48550/arXiv.1403.6827). arXiv: [1403.6827](https://arxiv.org/abs/1403.6827) [astro-ph.CO].
- Kamionkowski, Marc and Adam G. Riess (Sept. 2023). ‘The Hubble Tension and Early Dark Energy’. In: *Annual Review of Nuclear and Particle Science* 73, pp. 153–180. DOI: [10.1146/annurev-nucl-111422-024107](https://doi.org/10.1146/annurev-nucl-111422-024107). arXiv: [2211.04492](https://arxiv.org/abs/2211.04492) [astro-ph.CO].
- Kingma, Diederik P. and Jimmy Ba (Dec. 2014). ‘Adam: A Method for Stochastic Optimization’. In: *arXiv e-prints*, arXiv:1412.6980, arXiv:1412.6980. DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980). arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG].
- Klypin, Anatoly and Jon Holtzman (Dec. 1997). ‘Particle-Mesh code for cosmological simulations’. In: *arXiv e-prints*, astro-ph/9712217, astro-ph/9712217. DOI: [10.48550/arXiv.astro-ph/9712217](https://doi.org/10.48550/arXiv.astro-ph/9712217). arXiv: [astro-ph/9712217](https://arxiv.org/abs/astro-ph/9712217) [astro-ph].
- Knebe, Alexander et al. (Oct. 2013). ‘Structure finding in cosmological simulations: the state of affairs’. In: *Monthly Notices of the Royal Astronomical Society* 435.2, pp. 1618–1658. DOI: [10.1093/mnras/stt1403](https://doi.org/10.1093/mnras/stt1403). arXiv: [1304.0585](https://arxiv.org/abs/1304.0585) [astro-ph.CO].
- Kravtsov, Andrey V., Anatoly A. Klypin and Alexei M. Khokhlov (July 1997). ‘Adaptive Refinement Tree: A New High-Resolution N-Body Code for Cosmological Simulations’. In: *Astrophysical Journal Supplement* 111.1, pp. 73–94. DOI: [10.1086/313015](https://doi.org/10.1086/313015). arXiv: [astro-ph/9701195](https://arxiv.org/abs/astro-ph/9701195) [astro-ph].
- Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton (2012). ‘ImageNet classification with deep convolutional neural networks’. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., pp. 1097–1105.
- Kumar Aluri, Pavan et al. (May 2023). ‘Is the observable Universe consistent with the cosmological principle?’ In: *Classical and Quantum Gravity* 40.9, 094001, p. 094001. DOI: [10.1088/1361-6382/acbefc](https://doi.org/10.1088/1361-6382/acbefc). arXiv: [2207.05765](https://arxiv.org/abs/2207.05765) [astro-ph.CO].
- Lacey, Cedric and Shaun Cole (June 1993). ‘Merger rates in hierarchical models of galaxy formation’. In: *Monthly Notices of the Royal Astronomical Society* 262.3, pp. 627–649. DOI: [10.1093/mnras/262.3.627](https://doi.org/10.1093/mnras/262.3.627).
- Lahav, Ofer and Andrew R Liddle (Jan. 2022). ‘The Cosmological Parameters (2021)’. In: *arXiv e-prints*, arXiv:2201.08666, arXiv:2201.08666. DOI: [10.48550/arXiv.2201.08666](https://doi.org/10.48550/arXiv.2201.08666). arXiv: [2201.08666](https://arxiv.org/abs/2201.08666) [astro-ph.CO].
- Lam, Siu Kwan, Antoine Pitrou and Stanley Seibert (2015). ‘Numba: A llvm-based python jit compiler’. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6.
- Lanzieri, Denise, François Lanusse, Chirag Modi, Benjamin Horowitz, Joachim Harnois-Déraps, Jean-Luc Starck and LSST Dark Energy Science Collaboration (LSST DESC) (Nov. 2023). ‘Forecasting the power of higher order weak-lensing statistics with automatically differentiable simulations’. In: *Astronomy and Astrophysics* 679, A61, A61. DOI: [10.1051/0004-6361/202346888](https://doi.org/10.1051/0004-6361/202346888). arXiv: [2305.07531](https://arxiv.org/abs/2305.07531) [astro-ph.IM].

- LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel (1989). ‘Backpropagation Applied to Handwritten Zip Code Recognition’. In: *Neural Computation* 1.4, pp. 541–551. DOI: [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541).
- Li, Huiying, Xiaoqing Zhao, Anyang Su, Haitao Zhang, Jingxin Liu and Guiying Gu (2020). ‘Color Space Transformation and Multi-Class Weighted Loss for Adhesive White Blood Cell Segmentation’. In: *IEEE Access* 8, pp. 24808–24818. DOI: [10.1109/ACCESS.2020.2970485](https://doi.org/10.1109/ACCESS.2020.2970485).
- Li, Zewen, Wenjie Yang, Shouheng Peng and Fan Liu (2020). ‘A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects’. In: *CoRR* abs/2004.02806. arXiv: [2004.02806](https://arxiv.org/abs/2004.02806). URL: <https://arxiv.org/abs/2004.02806>.
- Linder, Eric V. (Mar. 2003). ‘Exploring the Expansion History of the Universe’. In: *Physical Review Letters* 90.9, 091301, p. 091301. DOI: [10.1103/PhysRevLett.90.091301](https://doi.org/10.1103/PhysRevLett.90.091301). arXiv: [astro-ph/0208512](https://arxiv.org/abs/astro-ph/0208512) [[astro-ph](#)].
- López-Cano, Daniel, Jens Stücker, Marcos Pellejero Ibañez, Raúl E. Angulo and Daniel Franco-Barranco (May 2024). ‘Characterizing structure formation through instance segmentation’. In: *Astronomy and Astrophysics* 685, A37, A37. DOI: [10.1051/0004-6361/202348965](https://doi.org/10.1051/0004-6361/202348965). arXiv: [2311.12110](https://arxiv.org/abs/2311.12110) [[astro-ph.CO](#)].
- LSST Science Collaboration et al. (Dec. 2009). ‘LSST Science Book, Version 2.0’. In: *arXiv e-prints*, arXiv:0912.0201, arXiv:0912.0201. DOI: [10.48550/arXiv.0912.0201](https://doi.org/10.48550/arXiv.0912.0201). arXiv: [0912.0201](https://arxiv.org/abs/0912.0201) [[astro-ph.IM](#)].
- Lucie-Smith, Luisa, Hiranya V. Peiris, Andrew Pontzen, Brian Nord and Jeyan Thiyaalingam (Mar. 2024). ‘Deep learning insights into cosmological structure formation’. In: *Physical Review D* 109.6. ISSN: 2470-0029. DOI: [10.1103/PhysRevD.109.063524](https://doi.org/10.1103/PhysRevD.109.063524). URL: <http://dx.doi.org/10.1103/PhysRevD.109.063524>.
- Massey, Richard, Thomas Kitching and Johan Richard (Aug. 2010). ‘The dark matter of gravitational lensing’. In: *Reports on Progress in Physics* 73.8, 086901, p. 086901. DOI: [10.1088/0034-4885/73/8/086901](https://doi.org/10.1088/0034-4885/73/8/086901). arXiv: [1001.1739](https://arxiv.org/abs/1001.1739) [[astro-ph.CO](#)].
- McKay, M. D., R. J. Beckman and W. J. Conover (1979). ‘A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code’. In: *Technometrics* 21.2, pp. 239–245. ISSN: 00401706. URL: <http://www.jstor.org/stable/1268522> (visited on 30/12/2024).
- Michaux, Michaël, Oliver Hahn, Cornelius Rampf and Raul E. Angulo (Jan. 2021). ‘Accurate initial conditions for cosmological N-body simulations: minimizing truncation and discreteness errors’. In: *Monthly Notices of the Royal Astronomical Society* 500.1, pp. 663–683. DOI: [10.1093/mnras/staa3149](https://doi.org/10.1093/mnras/staa3149). arXiv: [2008.09588](https://arxiv.org/abs/2008.09588) [[astro-ph.CO](#)].
- Mo, Houjun, Frank van den Bosch and Simon White (2010). ‘Formation and Structure of Dark Matter Halos’. In: *Galaxy Formation and Evolution*. Cambridge University Press, pp. 319–365.
- Modi, C., F. Lanusse and U. Seljak (Oct. 2021). ‘FlowPM: Distributed TensorFlow implementation of the FastPM cosmological N-body solver’. In: *Astronomy and Computing* 37, 100505, p. 100505. DOI: [10.1016/j.ascom.2021.100505](https://doi.org/10.1016/j.ascom.2021.100505). arXiv: [2010.11847](https://arxiv.org/abs/2010.11847) [[astro-ph.CO](#)].

Bibliography

- Modi, Chirag, François Lanusse, Uroš Seljak, David N. Spergel and Laurence Perreault-Levasseur (Apr. 2021). ‘CosmicRIM : Reconstructing Early Universe by Combining Differentiable Simulations with Recurrent Inference Machines’. In: *arXiv e-prints*, arXiv:2104.12864, arXiv:2104.12864. DOI: [10.48550/arXiv.2104.12864](https://doi.org/10.48550/arXiv.2104.12864). arXiv: [2104.12864](https://arxiv.org/abs/2104.12864) [astro-ph.CO].
- Moresco, Michele, Licia Verde, Lucia Pozzetti, Raul Jimenez and Andrea Cimatti (July 2012). ‘New constraints on cosmological parameters and neutrino properties using the expansion rate of the Universe to $z \sim 1.75$ ’. In: *Journal of Cosmology and Astroparticle Physics* 2012.7, 053, p. 053. DOI: [10.1088/1475-7516/2012/07/053](https://doi.org/10.1088/1475-7516/2012/07/053). arXiv: [1201.6658](https://arxiv.org/abs/1201.6658) [astro-ph.CO].
- Navarro, Julio F., Carlos S. Frenk and Simon D. M. White (May 1996). ‘The Structure of Cold Dark Matter Halos’. In: *Astrophysical Journal* 462, p. 563. DOI: [10.1086/177173](https://doi.org/10.1086/177173). arXiv: [astro-ph/9508025](https://arxiv.org/abs/astro-ph/9508025) [astro-ph].
- (Dec. 1997). ‘A Universal Density Profile from Hierarchical Clustering’. In: *Astrophysical Journal* 490.2, pp. 493–508. DOI: [10.1086/304888](https://doi.org/10.1086/304888). arXiv: [astro-ph/9611107](https://arxiv.org/abs/astro-ph/9611107) [astro-ph].
- Ondaro-Mallea, Lurdes, Raul E. Angulo, Matteo Zennaro, Sergio Contreras and Giovanni Aricò (Feb. 2022). ‘Non-universality of the mass function: dependence on the growth rate and power spectrum shape’. In: *Monthly Notices of the Royal Astronomical Society* 509.4, pp. 6077–6090. DOI: [10.1093/mnras/stab3337](https://doi.org/10.1093/mnras/stab3337). arXiv: [2102.08958](https://arxiv.org/abs/2102.08958) [astro-ph.CO].
- Pakmor, Rüdiger et al. (Sept. 2023). ‘The MillenniumTNG Project: the hydrodynamical full physics simulation and a first look at its galaxy clusters’. In: *Monthly Notices of the Royal Astronomical Society* 524.2, pp. 2539–2555. DOI: [10.1093/mnras/stac3620](https://doi.org/10.1093/mnras/stac3620). arXiv: [2210.10060](https://arxiv.org/abs/2210.10060) [astro-ph.CO].
- Peebles, P. J. E. (Sept. 1984). ‘Tests of cosmological models constrained by inflation’. In: *Astrophysical Journal* 284, pp. 439–444. DOI: [10.1086/162425](https://doi.org/10.1086/162425).
- Peiris, H. V. et al. (Sept. 2003). ‘First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications For Inflation’. In: *Astrophysical Journal Supplement* 148.1, pp. 213–231. DOI: [10.1086/377228](https://doi.org/10.1086/377228). arXiv: [astro-ph/0302225](https://arxiv.org/abs/astro-ph/0302225) [astro-ph].
- Perivolaropoulos, L. and F. Skara (Dec. 2022). ‘Challenges for Λ CDM: An update’. In: *New Astronomy Reviews* 95, 101659, p. 101659. DOI: [10.1016/j.newar.2022.101659](https://doi.org/10.1016/j.newar.2022.101659). arXiv: [2105.05208](https://arxiv.org/abs/2105.05208) [astro-ph.CO].
- Perlmutter, S. et al. (June 1999). ‘Measurements of Ω and Λ from 42 High-Redshift Supernovae’. In: *Astrophysical Journal* 517.2, pp. 565–586. DOI: [10.1086/307221](https://doi.org/10.1086/307221). arXiv: [astro-ph/9812133](https://arxiv.org/abs/astro-ph/9812133) [astro-ph].
- Pillepich, Annalisa et al. (Jan. 2018). ‘Simulating galaxy formation with the IllustrisTNG model’. In: *Monthly Notices of the Royal Astronomical Society* 473.3, pp. 4077–4106. DOI: [10.1093/mnras/stx2656](https://doi.org/10.1093/mnras/stx2656). arXiv: [1703.02970](https://arxiv.org/abs/1703.02970) [astro-ph.GA].
- Planck Collaboration, P. A. R. Ade, N. Aghanim, C. Armitage-Caplan et al. (Nov. 2014). ‘Planck 2013 results. XVI. Cosmological parameters’. In: *Astronomy and*

- Astrophysics* 571, A16, A16. DOI: [10.1051/0004-6361/201321591](https://doi.org/10.1051/0004-6361/201321591). arXiv: [1303.5076](https://arxiv.org/abs/1303.5076) [astro-ph.CO].
- Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud et al. (Sept. 2016). ‘Planck 2015 results. XIII. Cosmological parameters’. In: *Astronomy and Astrophysics* 594, A13, A13. DOI: [10.1051/0004-6361/201525830](https://doi.org/10.1051/0004-6361/201525830). arXiv: [1502.01589](https://arxiv.org/abs/1502.01589) [astro-ph.CO].
- Planck Collaboration, N. Aghanim et al. (Sept. 2020). ‘Planck 2018 results. VI. Cosmological parameters’. In: *Astronomy and Astrophysics* 641, A6, A6. DOI: [10.1051/0004-6361/201833910](https://doi.org/10.1051/0004-6361/201833910). arXiv: [1807.06209](https://arxiv.org/abs/1807.06209) [astro-ph.CO].
- Pontzen, Andrew, Rok Roškar, Greg Stinson and Rory Woods (May 2013). *pynbody: N-Body/SPH analysis for python*. Astrophysics Source Code Library, record ascl:1305.002.
- Press, William H. and Paul Schechter (Feb. 1974). ‘Formation of Galaxies and Clusters of Galaxies by Self-Similar Gravitational Condensation’. In: *Astrophysical Journal* 187, pp. 425–438. DOI: [10.1086/152650](https://doi.org/10.1086/152650).
- Riess, Adam G., Alexei V. Filippenko et al. (Sept. 1998). ‘Observational Evidence from Supernovae for an Accelerating Universe and a Cosmological Constant’. In: *Astronomical Journal* 116.3, pp. 1009–1038. DOI: [10.1086/300499](https://doi.org/10.1086/300499). arXiv: [astro-ph/9805201](https://arxiv.org/abs/astro-ph/9805201) [astro-ph].
- Riess, Adam G., Wenlong Yuan et al. (July 2022). ‘A Comprehensive Measurement of the Local Value of the Hubble Constant with 1 km s^{−1} Mpc^{−1} Uncertainty from the Hubble Space Telescope and the SH0ES Team’. In: *Astrophysical Journal Letters* 934.1, L7, p. L7. DOI: [10.3847/2041-8213/ac5c5b](https://doi.org/10.3847/2041-8213/ac5c5b). arXiv: [2112.04510](https://arxiv.org/abs/2112.04510) [astro-ph.CO].
- Ronneberger, Olaf, Philipp Fischer and Thomas Brox (May 2015). ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’. In: *arXiv e-prints*, arXiv:1505.04597, arXiv:1505.04597. DOI: [10.48550/arXiv.1505.04597](https://doi.org/10.48550/arXiv.1505.04597). arXiv: [1505.04597](https://arxiv.org/abs/1505.04597) [cs.CV].
- Rubner, Yossi, Carlo Tomasi and Leonidas Guibas (Nov. 2000). ‘The Earth Mover’s Distance as a Metric for Image Retrieval’. In: *International Journal of Computer Vision* 40, pp. 99–121. DOI: [10.1023/A:1026543900054](https://doi.org/10.1023/A:1026543900054).
- Rumelhart, David E., Geoffrey E. Hinton and Ronald J. Williams (1986). ‘Learning representations by back-propagating errors’. In: *Nature* 323, pp. 533–536. URL: <https://api.semanticscholar.org/CorpusID:205001834>.
- Schaye, Joop et al. (Dec. 2023). ‘The FLAMINGO project: cosmological hydrodynamical simulations for large-scale structure and galaxy cluster surveys’. In: *Monthly Notices of the Royal Astronomical Society* 526.4, pp. 4978–5020. DOI: [10.1093/mnras/stad2419](https://doi.org/10.1093/mnras/stad2419). arXiv: [2306.04024](https://arxiv.org/abs/2306.04024) [astro-ph.CO].
- Scoccimarro, Roman (Oct. 1998). ‘Transients from initial conditions: a perturbative analysis’. In: *Monthly Notices of the Royal Astronomical Society* 299.4, pp. 1097–1118. DOI: [10.1046/j.1365-8711.1998.01845.x](https://doi.org/10.1046/j.1365-8711.1998.01845.x). arXiv: [astro-ph/9711187](https://arxiv.org/abs/astro-ph/9711187) [astro-ph].
- Scolnic, D. M. et al. (June 2018). ‘The Complete Light-curve Sample of Spectroscopically Confirmed SNe Ia from Pan-STARRS1 and Cosmological Constraints from the Combined Pantheon Sample’. In: *Astrophysical Journal* 859.2, 101, p. 101. DOI: [10.3847/1538-4357/aab9bb](https://doi.org/10.3847/1538-4357/aab9bb). arXiv: [1710.00845](https://arxiv.org/abs/1710.00845) [astro-ph.CO].

Bibliography

- Seljak, Uroš, Grigor Aslanyan, Yu Feng and Chirag Modi (Dec. 2017). ‘Towards optimal extraction of cosmological information from nonlinear data’. In: *Journal of Cosmology and Astroparticle Physics* 2017.12, 009, p. 009. DOI: [10.1088/1475-7516/2017/12/009](https://doi.org/10.1088/1475-7516/2017/12/009). arXiv: [1706.06645](https://arxiv.org/abs/1706.06645) [astro-ph.CO].
- Sheth, Ravi K., H. J. Mo and Giuseppe Tormen (May 2001). ‘Ellipsoidal collapse and an improved model for the number and spatial distribution of dark matter haloes’. In: *Monthly Notices of the Royal Astronomical Society* 323.1, pp. 1–12. DOI: [10.1046/j.1365-8711.2001.04006.x](https://doi.org/10.1046/j.1365-8711.2001.04006.x). arXiv: [astro-ph/9907024](https://arxiv.org/abs/astro-ph/9907024) [astro-ph].
- Sheth, Ravi K. and Giuseppe Tormen (Jan. 2002). ‘An excursion set model of hierarchical clustering: ellipsoidal collapse and the moving barrier’. In: *Monthly Notices of the Royal Astronomical Society* 329.1, pp. 61–75. DOI: [10.1046/j.1365-8711.2002.04950.x](https://doi.org/10.1046/j.1365-8711.2002.04950.x). arXiv: [astro-ph/0105113](https://arxiv.org/abs/astro-ph/0105113) [astro-ph].
- Skrutskie, M. F. et al. (Feb. 2006). ‘The Two Micron All Sky Survey (2MASS)’. In: *Astronomical Journal* 131.2, pp. 1163–1183. DOI: [10.1086/498708](https://doi.org/10.1086/498708).
- Sousbie, Thierry and Stéphane Colombi (Sept. 2016). ‘ColDICE: A parallel Vlasov–Poisson solver using moving adaptive simplicial tessellation’. In: *Journal of Computational Physics* 321, pp. 644–697. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2016.05.048](https://doi.org/10.1016/j.jcp.2016.05.048). URL: <http://dx.doi.org/10.1016/j.jcp.2016.05.048>.
- Spergel, D. N. et al. (June 2007). ‘Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology’. In: *Astrophysical Journal Supplement* 170.2, pp. 377–408. DOI: [10.1086/513700](https://doi.org/10.1086/513700). arXiv: [astro-ph/0603449](https://arxiv.org/abs/astro-ph/0603449) [astro-ph].
- Springel, Volker, Rüdiger Pakmor, Oliver Zier and Martin Reinecke (Sept. 2021). ‘Simulating cosmic structure formation with the GADGET-4 code’. In: *Monthly Notices of the Royal Astronomical Society* 506.2, pp. 2871–2949. DOI: [10.1093/mnras/stab1855](https://doi.org/10.1093/mnras/stab1855). arXiv: [2010.03567](https://arxiv.org/abs/2010.03567) [astro-ph.IM].
- Spurzem, R. (Sept. 1999). ‘Direct N-body Simulations’. In: *Journal of Computational and Applied Mathematics* 109, pp. 407–432. DOI: [10.48550/arXiv.astro-ph/9906154](https://doi.org/10.48550/arXiv.astro-ph/9906154). arXiv: [astro-ph/9906154](https://arxiv.org/abs/astro-ph/9906154) [astro-ph].
- Stein, George, Marcelo A. Alvarez and J. Richard Bond (Feb. 2019). ‘The mass-Peak Patch algorithm for fast generation of deep all-sky dark matter halo catalogues and its N-body validation’. In: *Monthly Notices of the Royal Astronomical Society* 483.2, pp. 2236–2250. DOI: [10.1093/mnras/sty3226](https://doi.org/10.1093/mnras/sty3226). arXiv: [1810.07727](https://arxiv.org/abs/1810.07727) [astro-ph.CO].
- Straumann, Norbert (2013). *General Relativity*. DOI: [10.1007/978-94-007-5410-2](https://doi.org/10.1007/978-94-007-5410-2).
- Tinker, Jeremy, Andrey V. Kravtsov, Anatoly Klypin, Kevork Abazajian, Michael Warren, Gustavo Yepes, Stefan Gottlöber and Daniel E. Holz (Dec. 2008). ‘Toward a Halo Mass Function for Precision Cosmology: The Limits of Universality’. In: *Astrophysical Journal* 688.2, pp. 709–728. DOI: [10.1086/591439](https://doi.org/10.1086/591439). arXiv: [0803.2706](https://arxiv.org/abs/0803.2706) [astro-ph].
- Villaescusa-Navarro, Francisco, Daniel Anglés-Alcázar et al. (July 2021). ‘The CAMELS Project: Cosmology and Astrophysics with Machine-learning Simulations’. In: *Astrophysical Journal* 915.1, 71, p. 71. DOI: [10.3847/1538-4357/abf7ba](https://doi.org/10.3847/1538-4357/abf7ba). arXiv: [2010.00619](https://arxiv.org/abs/2010.00619) [astro-ph.CO].

- Villaescusa-Navarro, Francisco, ChangHoon Hahn et al. (Sept. 2020). ‘The Quijote Simulations’. In: *Astrophysical Journal Supplement* 250.1, 2, p. 2. DOI: [10.3847/1538-4365/ab9d82](https://doi.org/10.3847/1538-4365/ab9d82). arXiv: [1909.05273](https://arxiv.org/abs/1909.05273) [astro-ph.CO].
- von Hoerner, S. (Jan. 1960). ‘Die numerische Integration des n-Körper-Problemes für Sternhaufen. I’. In: *Zeitschrift für Astrophysik* 50, pp. 184–214.
- Wechsler, Risa H. and Jeremy L. Tinker (Sept. 2018). ‘The Connection Between Galaxies and Their Dark Matter Halos’. In: *Annual Review of Astronomy and Astrophysics* 56, pp. 435–487. DOI: [10.1146/annurev-astro-081817-051756](https://doi.org/10.1146/annurev-astro-081817-051756). arXiv: [1804.03097](https://arxiv.org/abs/1804.03097) [astro-ph.GA].
- Weinberg, Steven (1972). *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity*.
- White, S. D. M. (Nov. 1984). ‘Angular momentum growth in protogalaxies’. In: *Astrophysical Journal* 286, pp. 38–41. DOI: [10.1086/162573](https://doi.org/10.1086/162573).
- Wu, Yanzhao, Ling Liu, Juhyun Bae, Ka-Ho Chow, Arun Iyengar, Calton Pu, Wenqi Wei, Lei Yu and Qi Zhang (Aug. 2019). ‘Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks’. In: *arXiv e-prints*, arXiv:1908.06477, arXiv:1908.06477. DOI: [10.48550/arXiv.1908.06477](https://doi.org/10.48550/arXiv.1908.06477). arXiv: [1908.06477](https://arxiv.org/abs/1908.06477) [cs.LG].
- Wu, Yiheng, Hong Guo and Volker Springel (July 2024). ‘Improving the accuracy of halo mass based statistics for fast approximate N-body simulations’. In: *Monthly Notices of the Royal Astronomical Society* 531.4, pp. 4944–4953. DOI: [10.1093/mnras/stae1439](https://doi.org/10.1093/mnras/stae1439). arXiv: [2406.10466](https://arxiv.org/abs/2406.10466) [astro-ph.IM].
- Xu, Bing, Naiyan Wang, Tianqi Chen and Mu Li (May 2015). ‘Empirical Evaluation of Rectified Activations in Convolutional Network’. In: *arXiv e-prints*, arXiv:1505.00853, arXiv:1505.00853. DOI: [10.48550/arXiv.1505.00853](https://doi.org/10.48550/arXiv.1505.00853). arXiv: [1505.00853](https://arxiv.org/abs/1505.00853) [cs.LG].
- Xu, Guohong (May 1995). ‘A New Parallel N-Body Gravity Solver: TPM’. In: *Astrophysical Journal Supplement* 98, p. 355. DOI: [10.1086/192166](https://doi.org/10.1086/192166). arXiv: [astro-ph/9409021](https://arxiv.org/abs/astro-ph/9409021) [astro-ph].
- Zel’dovich, Ya. B. (Mar. 1970). ‘Gravitational instability: An approximate theory for large density perturbations.’ In: *Astronomy and Astrophysics* 5, pp. 84–89.
- Zhang, Tianchi, Shihong Liao, Ming Li and Liang Gao (July 2019). ‘The optimal gravitational softening length for cosmological N-body simulations’. In: *Monthly Notices of the Royal Astronomical Society* 487.1, pp. 1227–1232. DOI: [10.1093/mnras/stz1370](https://doi.org/10.1093/mnras/stz1370). arXiv: [1810.07055](https://arxiv.org/abs/1810.07055) [astro-ph.CO].
- Zwicky, F. (Jan. 1933). ‘Die Rotverschiebung von extragalaktischen Nebeln’. In: *Helvetica Physica Acta* 6, pp. 110–127.
- (Oct. 1937). ‘On the Masses of Nebulae and of Clusters of Nebulae’. In: *Astrophysical Journal* 86, p. 217. DOI: [10.1086/143864](https://doi.org/10.1086/143864).