2022

TOWARD DEEP LEARNING EMULATORS FOR MODELING THE LARGE-SCALE STRUCTURE OF THE UNIVERSE

Neerav Kaushal
Michigan Technological University, kaushal@mtu.edu

TOWARD DEEP LEARNING EMULATORS FOR MODELING THE LARGE-SCALE STRUCTURE OF THE UNIVERSE

By

Neerav Kaushal

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Applied Physics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2022

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Applied Physics.

Department of Physics

Dissertation Advisor:	*Dr. Elena Giusarma*
Committee Member:	*Dr. Robert Nemiroff*
Committee Member:	*Dr. Francisco Villaescusa-Navarro*
Committee Member:	*Dr. Mauricio Reyes*
Committee Member:	*Dr. Petra Huentemeyer*
Department Chair:	*Dr. Ravindra Pandey*

Dedication

To my Ma, Papa and Meera

who were a constant source of support, motivation, and inspiration for me at each step of my academic life. They are the beacons that always guided me, led me, and pushed me through all my difficult times. They have always been the most understanding people I have ever had and will ever need. This work would not have been possible without my Ma, who always instilled in me the habits of dreaming big, being ambitious, and shooting for the stars; without my Papa, who constantly reminded me of the honor of having a "Dr." in front of your name; and without Meera, who has always been with me during the sinusoidal phases of my PhD and constantly helped me be a better version of myself every day.

Contents

List of Figures

List of Tables

Author Contribution Statement

I am the standalone contributor of Chapters 1, 2, 3 and 7 and the main contributor of Chapter 4. I am not the main contributor of Chapter 6 and my contribution is limited to section 6.4.5.3. Chapters 4, 6 and 7 are the works published in peer-reviewed journals.

Acknowledgments

Many people deserve my heartfelt gratitude for making this work possible.

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Elena Giusarma, for giving me the opportunity to conduct research and providing me with invaluable guidance throughout my PhD. She has constantly supported me and this work would not have been possible without her. I am very thankful to her for the time and patience she put in during the entire duration of my research and thesis, and for her meticulous review of every piece of work I wrote, even during her hardships. She has always put her entire trust in me when it came to research quality, ethics, and direction. In addition to being a great advisor, she is also a really great person who has always been very kind to me and understanding of my situations. She always understood the importance of my family in my life and gave me much-needed time to visit them.

I would also like to sincerely thank Dr. Robert Nemiroff. His dynamism, vision, sincerity, and motivation have deeply inspired and shaped me. He has taught me the importance of humility, eagerness to learn, and not shying away from putting forward a viewpoint based entirely on logic, even if my knowledge of the concept is not robust. It has always been my great pleasure and privilege to converse with him.

Besides, I extend my heartfelt gratitude to my department chair, Dr. Ravindra Pandey, for always being the most kind, understanding, supportive, and approachable person. He constantly checked up on me and gave numerous pieces of pragmatic and personal advice. I thank him for his consistently motivating and inspiring words that provided a very supportive environment for me here at Michigan Tech.

I am also grateful to all my friends, especially Cameron Shock, who has always been on my side as an honest critic and a true friend. I thank him for making unarguably the lengthiest scientific conversations I ever had an absolute delight. I would also like to thank our office manager, Claire Wiitanen, for all her administrative help, timely communication, and afternoon coffees.

Finally, I would like to thank my PhD committee for their time, support, feedback, and careful reviews of my work.

List of Abbreviations

ΛCDM	Lambda Cold Dark Matter
AI	Artificial Intelligence
ANN	Artificial Neural Network
BCE	Binary Crossentropy
CAMB	Code for Anisotropies in the Microwave Background
CDM	Cold Dark Matter
CMB	Cosmic Microwave Background
CNN	Convolutional Neural Networks
COLA	COmoving Lagrangian Acceleration Code
DESI	Dark Energy Spectroscopic Instrument
DL	Deep Learning
DM	Dark Matter
GAN	Generative Adversarial Networks
GD	Gradient Descent
GPU	Graphical Processing Unit
ICs	Initial Conditions
LPT	Lagrangian Perturbation Theory
LSS	Large-scale Structure

LSST	Large Synoptic Survey Telescope
MACHO	Massive Compact Halo Objects
MSE	Mean Square Error
ML	Machine Learning
NECOLA	Neural Enhanced COLA
NN	Neural Networks
ReLU	Rectified Linear Unit
SVM	Support Vector Machine
WIMP	Weakly Interacting Massive Particles
ZA	Zeldovich Approximations

List of Publications

This thesis is based on the following publications:

Chapter 4

NECOLA: Towards a Universal Field-level Cosmological Emulator

Kaushal, N., Villaescusa-Navarro, F., Giusarma, E., Li, Y., Hawry, C., & Reyes, M.

The Astrophysical Journal, 930(2), 115 (2022).

DOI: https://doi.org/10.3847/1538-4357/ac5c4a

Chapter 5

νGAN: Conditional GAN-based Emulator for Cosmic Web Simulations with Massive Neutrinos

Kaushal, N. & Giusarma, E.

In writing

Chapter 6

The CAMELS project: public data release

Villaescusa-Navarro, Francisco ; Genel, Shy ; Anglés-Alcázar, Daniel By Villaescusa-Navarro, F., Genel, S., Daniel, A. et al.

e-Print Archive: astro-ph/2201.01300 (2022)

DOI: https://doi.org/10.48550/arXiv.2201.01300

Chapter 7

A Novel RID Algorithm of Muon Trajectory Reconstruction in Water Cherenkov Detectors

Kaushal, N.

The Astrophysical Journal, 936(2), 120. (2022)

DOI: https://doi.org/10.3847/1538-4357/ac8798

Other peer-reviewed papers not discussed in the thesis:

Toward the Detection of Relativistic Image Doubling in Water Cerenkov Detectors

Kaushal, N., & Nemiroff, R. J.

The Astrophysical Journal, 898(1), 53. (2020)

DOI: https://doi.org/10.3847/1538-4357/ab98fa

Toward the Detection of Relativistic Image Doubling in Imaging Atmospheric Cherenkov Telescopes

Nemiroff, R. J., & Kaushal, N.

The Astrophysical Journal, 889(2), 122. (2020)

DOI: https://doi.org/10.3847/1538-4357/ab6440

Analyses of scissors cutting paper at superluminal speeds

Kaushal, N., & Nemiroff, R. J.

Physics Education, 54(6), 065008. (2019)

DOI: https://doi.org/10.1088/1361-6552/ab3d9f

Towards a Universal Field-level Cosmological Emulator

Kaushal, N., Villaescusa-Navarro, F., Giusarma, E., Li, Y., Hawry, C., & Reyes, M.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021) (2021)

(https://ml4physicalsciences.github.io/2021/files/NeurIPS_ML4PS_2021_47.pdf)

Abstract

Multi-billion dollar cosmological surveys are being conducted almost every decade in today's era of precision cosmology. These surveys scan vast swaths of sky and generate tons of observational data. In order to extract meaningful information from this data and test these observations against theory, rigorous theoretical predictions are needed. In the absence of an analytic method, cosmological simulations become the most widely used tool to provide these predictions in order to test against the observations. They can be used to study covariance matrices, generate mock galaxy catalogs and provide ready-to-use snapshots for detailed redshift analyses. But cosmological simulations of matter formation in the universe are one of most computationally intensive tasks. Faster but equally reliable tools that could approximate these simulations are thus desperately needed. Recently, deep learning has come up as an innovative and novel tool that can generate numerous cosmological simulations orders of magnitude faster than the traditional simulations. Deep learning models of structure formation and evolution in the universe are unimaginably fast and retain most of the accuracy of conventional simulations, thus providing a fast, reliable, efficient and accurate method to study the evolution of the universe and reducing the computational burden of current simulation methods.

In this dissertation, we will focus on deep learning-based models that could mimic the

process of structure formation in the universe. In particular, we focus on developing deep convolutional neural network models that could learn the present 3D distribution of the cold dark matter and generate 2D dark matter cosmic mass maps. We employ summary statistics most commonly employed in cosmology and computer vision to quantify the robustness of our models.

This dissertation is organized as follows: Chapters 1, 2 and 3 discuss the basics and detailed overview of cosmology, cosmological simulations and deep learning. Chapter 4 presents a convolutional neural network model that maps 3D dark matter distribution from fast and mildly accurate COLA simulations to slow and accurate N-body simulations. In chapter 5, we develop a novel neural network generative model that could create statistically independent and robust mass maps of cold dark matter distribution in the universe. Chapter 6 discusses the CAMELS project, a suite of $4,233$ high resolution N-body and hydrodynamics cosmological simulations that combines astrophysics and cosmology. Finally, we discuss a novel and efficient algorithm that could reconstruct the muon trajectories in water Cerenkov detectors in chapter 7.

Chapter 1

An Overview of Cosmology

Our universe evolved over billions of years from uniform fluctuations in matter density to its current state. This present form of our universe has characteristic features that include cosmic filaments, voids and galaxy clusters [10]. Studies of the large-scale structure of the universe provide us with useful information about the nature of gravity, dark matter and dark energy, and the composition of the universe. This information is then used to revise the formulation of various theories of structure formation and evolution which results in a reformed understanding of the finer workings of the universe. In this chapter, we define the ΛCDM model of cosmology and explore the most important observables used to extract information about our universe.

1.1 Basic Concepts

1.1.1 Scale factor

The relative expansion of the universe is parameterized by a dimensionless quantity called the cosmic scale factor. The scale factor is denoted by a and embodies the evolution of the universe and describes the scaling up of all physical distances in the cosmos including the separation of galaxies, and the wavelengths of photons. The distance between the coordinates of two points is called the **comoving distance** (r), and the distance that roughly corresponds to where a distant object would be at a specific moment of cosmological time is called the **proper or physical distance** ($x(t)$). When the universe expands, the comoving distance remains the same but the proper distance increases, and is given by

$$x(t) = a(t)\, r, \tag{1.1}$$

A depiction of scale factor is shown in figure 1.1

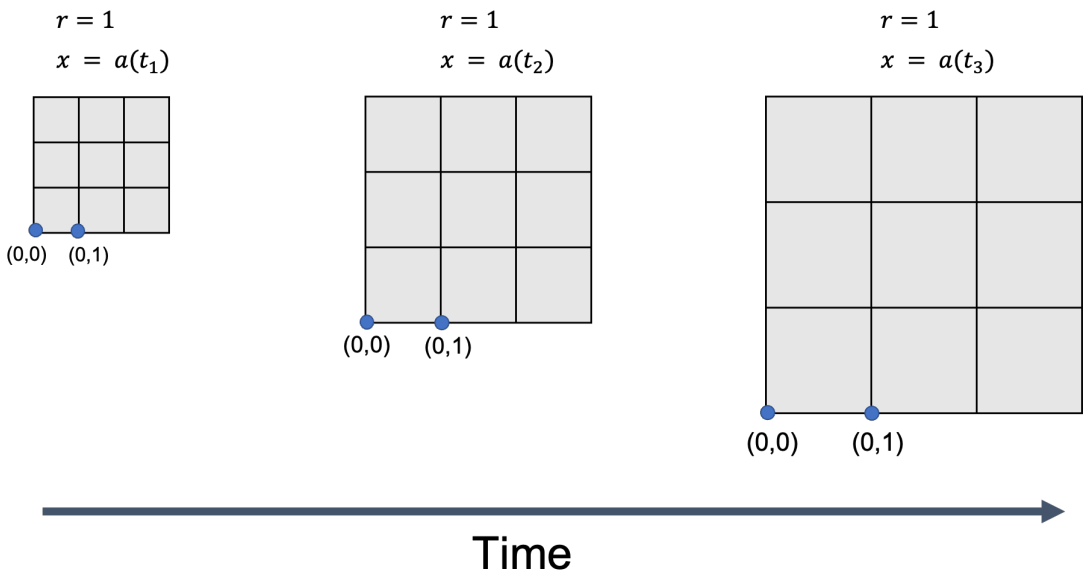

Figure 1.1: The comoving distance (r) between the points on a hypothetical grid remains the same as the universe expands. The physical distance (x) though, is proportional to the scale factor at that time ($a(t)$) times the comoving distance (see equation 1.1). Note that $t_1 < t_2 < t_3$. As the scale factor gets larger over time, so does the physical distance. By convention, the scale factor is assumed to be 1 at the present time i.e., $a(t_0) = 1$.

1.1.2 Redshift

The expansion of the universe causes the conformal stretching of photons emitted by the distant objects at earlier times. This means that the wavelength of the light emitted, $\lambda_{emitted}$, from a source at an earlier time ($t_{earlier}$) has been stretched to $\lambda_{observed}$ at the present time ($t_{present}$) due to this expansion. From this, we can define the redshift of a source as:

$$z = \frac{\lambda_{observed} - \lambda_{emitted}}{\lambda_{emitted}}. \tag{1.2}$$

This implies that the wavelength that we observe today is larger than the wavelength emitted originally by a factor of $a(t_{present})/a(t_{earlier})$. This relates redshift to the scale factor as:

$$1 + z = \frac{a(t_{present})}{a(t_{earlier})}. \tag{1.3}$$

As $a(t_{present}) = 1$ by convention, the above equation reduces to

$$a(t_{earlier}) = \frac{1}{1+z}. \tag{1.4}$$

This means that the light we observe from a source at redshift z was emitted when the universe was smaller than today by a factor of $1 + z$. The scale factor at that time is given by equation 1.4.

1.1.3 Hubble's Law

The Hubble parameter measures how rapidly the scale factor changes and is defined as :

$$H(t) = \frac{1}{a}\frac{da}{dt} = \frac{\dot{a}}{a}. \tag{1.5}$$

The value of the Hubble parameter evaluated at the current time is called the **Hubble constant** and is denoted by H_0. At low redshifts, the Hubble constant relates the recessional velocities of galaxies and their distances from the observer through the **Hubble's Law** [11] as:

$$v = H_0\, d. \tag{1.6}$$

It is conventional to parameterize H_0 in distance-based measurements as $h = H_0/100$. The units of H_0 are km/(sec Mpc).

1.1.4 Homogeneity and Isotropy

We know that the matter in the universe is clustered in the form of stars grouping together to form galaxies on smaller scales and galaxies grouping together to form galaxy structures on larger scales. Despite the observed fact that the universe

is clumpy and contains clustered matter at all scales, cosmologists very frequently use a basic assumption in their analyses of cosmological observables, known as the Cosmological Principle. It states that:

The universe is both homogeneous and isotropic on the largest scales.

This principle results from the understanding that the forces in nature are expected to act uniformly throughout the universe and therefore should not produce any observable irregularities in the large-scale structure of the distribution of matter in the universe, which itself evolved from a gaussian distribution of primordial matter density field. The absence of a preferred location in the universe is termed as homogeneity. It means that at any given point in time, the universe will look the same at every single point in space. Isotropy, on the other hand, means that there is no preferred direction in the universe. That is, from our current location, no matter which direction we look into, the universe will always appear to be the same. Both homogeneity and isotropy may appear to be separate concepts but are well interconnected in reality.

1.1.5 Matter Power Spectrum

The matter density in the universe at any time t can be expressed in terms of the mean density and a local fluctuation as follows:

$$\rho(\vec{x}) = \bar{\rho}(1 + \delta(\vec{x})), \tag{1.7}$$

where $\delta(\vec{x})$ is the overdensity at location $\vec{x}$. Expanding $\delta(\vec{x})$ into Fourier modes, we obtain

$$\delta(\vec{x}) \equiv \frac{\rho(\vec{x}) - \bar{\rho}}{\bar{\rho}} = \int \delta(\vec{k}) e^{-i\,\vec{k}.\vec{x}} d^3k, \tag{1.8}$$

where k is the wavenumber given by $k = 2\pi/\lambda$. We define the two-point **correlation function**, $\xi(x)$, as the excess probability of finding two galaxies separated by a distance x compared to a random distribution of galaxies. It can be thought of as a clumpiness factor - the higher the value for some distance scale, the more clumpy the universe is at that distance scale. The power spectrum is most commonly defined as the fourier transform of the two-point correlation function:

$$\xi(r) = \int \frac{d^3k}{(2\pi)^3} P(k) e^{i.\vec{k}.(x-x')} \tag{1.9}$$

The power spectrum is thus given by averaging over fourier space as:

$$< \delta(\vec{k})\ \delta(\vec{k'}) > \ = \ \ 2\pi^3\ P(k)\ \delta(\vec{k} - \vec{k'}), \tag{1.10}$$

which gives,

$$P(k) = |\delta^2(\vec{k})|. \tag{1.11}$$

The matter power spectrum is the square of amplitude of the fourier modes of the matter density perturbation. The ΛCDM power spectrum asymptotes to $P(k) \sim k^1$ for small k, and $P(k) \sim k^{-3}$ for large k. Figure 1.2 shows the linear matter power spectrum for PLANCK[1] [1] cosmology obtained using CAMB [12] at different redshifts.

According to the theory of inflation [13–16], the universe underwent a rapid expansion during the very early times which stretched the quantum mechanical fluctuations to macroscopic scales. Since quantum mechanical fluctuations are random, the primordial density perturbations can be well described by a gaussian random field. The primordial power spectrum is parameterized as a power law $P_{primordial}(k) \propto k^n$, with $n = 1$ corresponding to scale-invariant spectrum proposed by Harrison and Zeldovich [17, 18].

[1]The Planck satellite is the European Space Agency's first mission to study the origins of the universe. It surveyed the microwave sky measuring the cosmic microwave background (CMB), the afterglow of the Big Bang, and the emission from gas and dust in our own Milky Way galaxy.

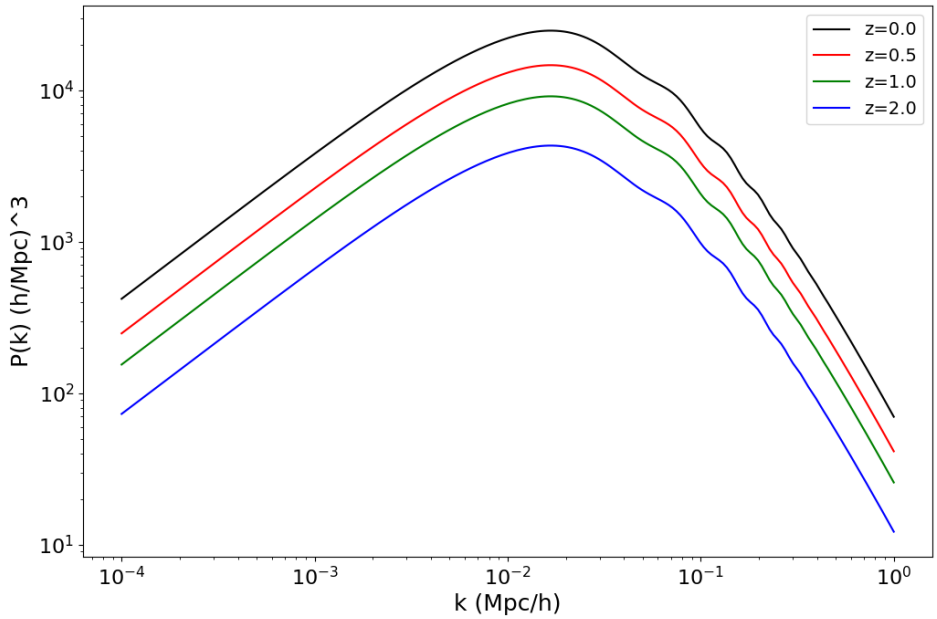

Figure 1.2: The linear power spectrum at various redshifts in the history of the universe. The cosmological parameters from PLANCK [1] are used.

1.1.6 Bispectrum

Huge cosmological information is extracted from various large-scale measurements of the universe through the use of two-point correlation function, or it's fourier transform, the power spectrum. The power spectrum fully characterizes a gaussian random field with zero mean and hence is a very important tool for studying the density field of the universe. However, the multi-scale and nonlinear nature of some physical processes like gravitational dynamics and galaxy biasing induces signatures of non-gaussianity in the density field of the universe. This non-gaussianity manifests itself

in the large-scale observations in the form of elongated filaments, compact galaxy clusters, and very prominent underdense regions called cosmic voids. These features of the cosmic web can not be fully characterized by the 2-point statistics, and higher order statistics are needed to study them.

Similar to equation 1.10, at the three-point level,

$$< \delta(\vec{k}_1)\delta(\vec{k}_2)\delta(\vec{k}_3) > \equiv \ \delta_D(\vec{k}_{123})\ B(\vec{k}_1, \vec{k}_2, \vec{k}_3), \tag{1.12}$$

where $\delta(\vec{k})$ is the overdensity in the fourier space and $\vec{k}_{123} \equiv \vec{k}_1 + \vec{k}_2 + \vec{k}_3$.

In the matter density field of the universe, the bispectrum captures the amount of matter in various triangle configurations, and hence is primarily used to study elongated galaxy filaments.

1.2 The ΛCDM Model

The Lambda-CDM (ΛCDM) model, also known as the Concordance model or Standard model of cosmology, is the current theoretical framework that describes our universe with just six parameters. It is also called the Concordance model or the

Standard model of Cosmology. Current high-precision cosmological observations including the Cosmic Microwave Background (CMB) [19], the observations of distant supernovae [20], and the large-scale structure of galaxies [21] have stunningly confirmed this model. This model states that rooted in the event of Big Bang, the universe evolved from an almost uniform distribution of matter and energy to its present day state. In other words, the early universe was almost entirely smooth and had only small fluctuations/ripples in matter density (manifested in the Cosmic Microwave Background (CMB) [19] observations). These initially minuscule overdense fluctuations in the density attracted more and more mass as the universe expanded [22] and gave birth to the present day cosmic web of matter. The ΛCDM model includes a cosmological constant (Lambda or Λ) and cold dark matter (CDM) as the most prominent constituents. The cosmological constant makes up most of the part of the universe (around $\sim 68\%$)[1] and provides an explanation for the universe's observed accelerating expansion. The Dark Matter (DM) makes about 85% of the entire matter in the universe which is about 27% of the total content of the universe, and can neither reflect, nor absorb nor transmit light. The composition of dark matter is presently unknown and there are numerous hypotheses of what it might consist of, including but not limited to Massive Compact Halo Objects (MACHOs), Weakly Interacting Massive Particles (WIMPs), and Axions. Various experiments like Axion Dark Matter Experiment (ADME) [23, 24], Korea Invisible Mass Search (KIMS)

[25, 26], and DarkSide [27, 28] are presently underway to detect dark matter signatures from the outer universe. Dark matter provides an explanation for the dynamics of galaxies and galaxy clusters that appear to have more gravitational attraction than expected from the models of galaxy formation, dynamics and gravitational lensing. The ordinary matter, also known as the baryonic matter, consists of all the matter we can directly see or observe, like the galaxies, stars, interstellar dust and gas, planets and every kind of matter we can think of and make up only about 5% of the universe. A pie chart of the composition of the universe is shown in Figure 1.3.

1.3 Cosmological Parameters

ΛCDM relates many observed phenomena to six apparently arbitrary parameters determinable from observation. These are as follows:

† **The Hubble parameter (H)**: It is the normalized rate of expansion and is given as $H = \dot{a}/a$, where a is the cosmic scale factor (see 1.1.3).

† **Total Matter density parameter (Ω_m)**: It describes the actual density of all the matter (dark and ordinary) in the universe relative to the critical density[2], and is given as $\Omega_m = \rho/\rho_{critical}$. Here, $\rho_{critical}$ is defined as $\rho_{critical} = 3H^2/8\pi G$

[2]The critical density is the matter density of a spatially flat universe. It is the density at which the universe is at balance, and neither expands due to the global expansion nor contracts due to gravity. It's current value is given by 9×10^{-27} kilograms per cubic meter.

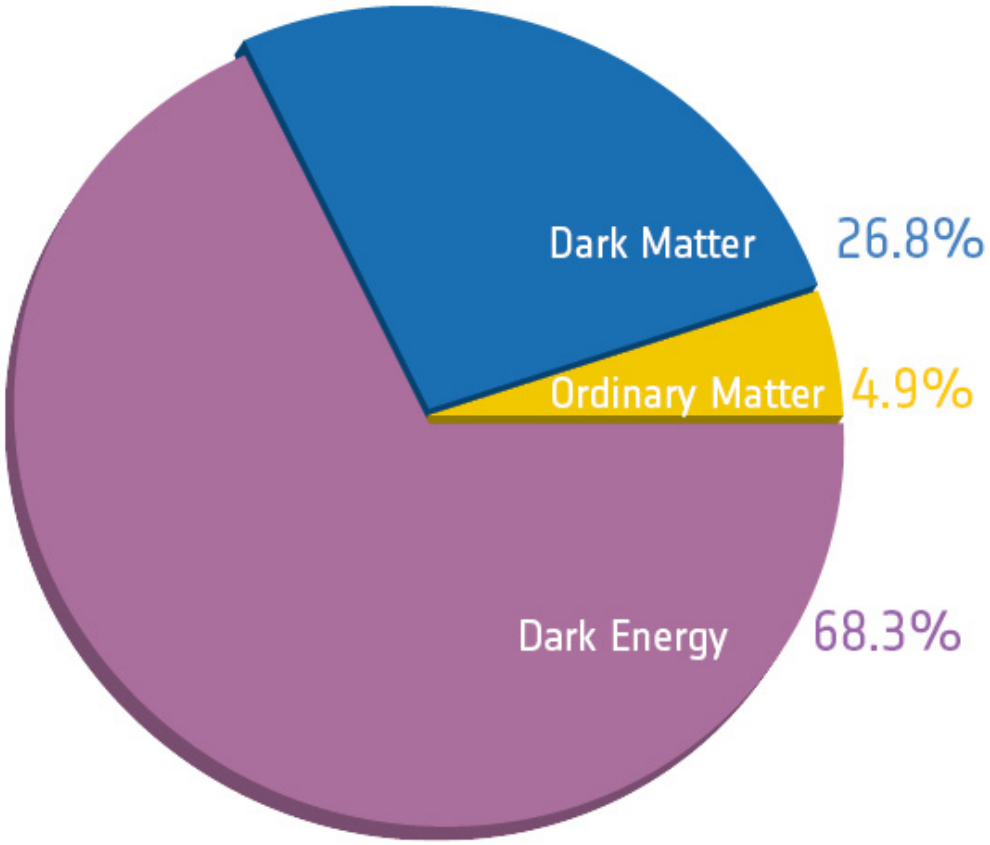

Figure 1.3: The composition of our universe. Image credit: European Space Agency under CC BY-SA 3.0 IGO license.

where H is the Hubble's parameter and G is the universal gravitational constant.

† **Baryonic matter density parameter (Ω_b)**: It describes the actual density of the ordinary (baryonic) matter relative to the critical density, and is given as $\Omega_b = \rho_b/\rho_{critical}$. Note that the total energy density of the universe is given by $\Omega_m + \Omega_\Lambda = 1$, where $\Omega_m = \Omega_{cdm} + \Omega_b + \Omega_\nu$. Ω_{cdm} and Ω_ν are the cold dark matter density parameter and the neutrino density parameter respectively.

† **Scalar spectral index (n_s)**: The scale-dependence of the Cosmic Microwave Background power spectrum constrains the slope of the primordial scalar power spectrum, which is conventionally parameterized by the power-law index n_s, called the scalar spectral index. An $n_s = 1$ corresponds to the scale-invariant spectrum.

† **Amplitude of density fluctuations (σ_8)**: It measures the amplitude of density fluctuations (the linear power spectrum) at the scale of 8 $h^{-1}Mpc$.

† **Age of the universe (t_0)**: The present day age of the universe has been found to be 13.8 billion years and is another significant cosmological parameter.

The six parameters of the ΛCDM model provides an astonishingly accurate description of the universe. The values of these cosmological parameters from PLANCK [9] observations are listed in Table 1.1.

Table 1.1
A list of cosmological parameters in the standard model of the cosmology and their experimentally calculated values from Planck TT, TE, EE+lowE+lensing [9].

Parameter	Description	Value
Ω_m	Total matter density	0.3158
Ω_b	Baryonic matter density	0.0494
n_s	Linear spectral index	0.96605
σ_8	Variance in matter fluctuations at 8 Mpc/h	0.8120
t_0	Age of the universe	13.7971
H_0	Hubble Constant	67.32

1.4 Cosmological Observables

1.4.1 Cosmic Microwave Background (CMB)

The cosmic microwave background (CMB) is the leftover radiation from the Big Bang. The universe was filled with hot plasma of particles (mostly neutrons, electrons and protons) and radiation (photons) when it was born around 13.8 billion years ago. The photons in this plasma continuously interacted with the free electrons as the rate of expansion of the universe was smaller than the rate of scattering of photons with electrons. Therefore, the photons could not travel long distances and the early universe was opaque. As the universe continued expanding, it cooled down and around 380,000 years ago, it's temperature dropped down to around 3000 K. At this temperature, the electrons combined with protons to form neutral hydrogen atoms. The photons could now travel unhindered into the expanded volume of the universe and the universe became transparent. Due to the further expansion of the universe for billions of years, the wavelengths of these photons now grew (redshifted) to the microwave region of electromagnetic spectrum (roughly 1 mm) and the universe is currently cooled to around 2.725 K. These photons that last scattered $380,000$ years ago fill the universe today and create a background glow that can be detected by far-infrared and radio telescopes. This is called the Cosmic Microwave Background

(CMB) radiation, and is shown in Figure 1.4.

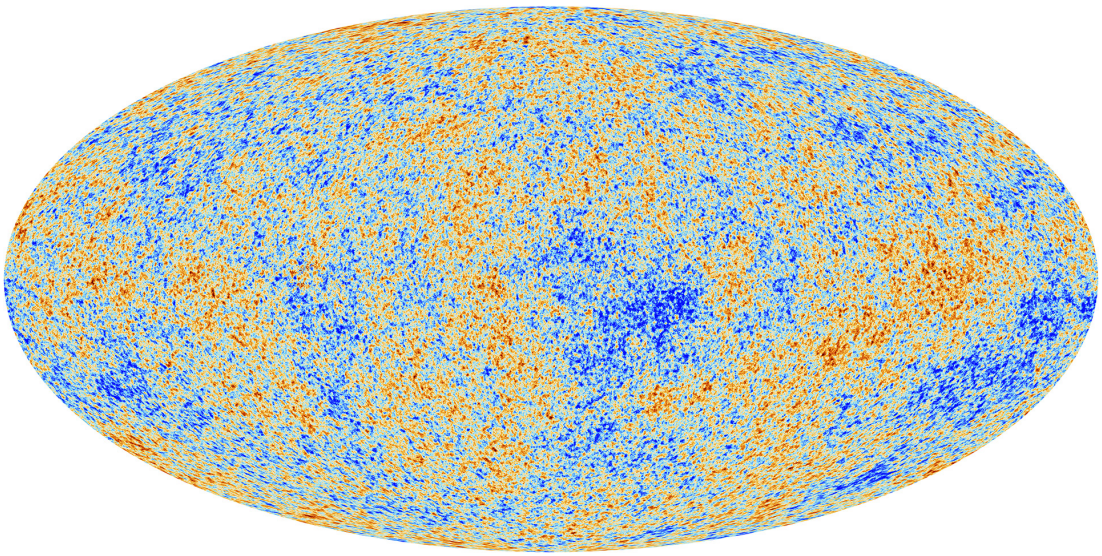

Figure 1.4: The anisotropies in the Cosmic Microwave Background radiation. The figure shows the variations/fluctuations in the temperature field of the universe that correspond to the first light of the universe when it was $380,000$ years old. The orange parts correspond to the regions of higher temperatures while the blue parts represent the colder regions of the universe. Image credit: European Space Agency under CC BY-SA 3.0 IGO license.

The CMB is often characterized by an angular power spectrum of its anisotropies (CMB anisotropies) showing the temperature variation for coefficients of a multipole expansion of the temperature over the celestial sphere. From the angular power spectrum, scales in the early universe can be determined, and values such as the six parameters of the ΛCDM model can be checked for consistency with the CMB.

1.4.2 The Large-Scale Structure

The Large Scale Structure (LSS) of the universe refers to the intricate pattern of galaxies and other cosmological matter on very large scales, typically much larger than the scales of individual galaxies or groups of galaxies. The matter is pulled together on smaller scales to form bigger structures (like gravity pulling together gas to form stars) and the same process happens at larger scales (gravity pulling together stars to form galaxies) and a hierarchy of scale is thus developed. This leads to the formation of vast, correlated structures of matter that are billions of light years in length. The distinct features of this large-scale structure are galaxy clusters, filaments and cosmic voids. These together form a spiderweb-like pattern spanning thousands of millions of light years and thus are collectively called the 'Cosmic Web'. Galaxy clusters are the clusters of galaxies that consist of hundreds or thousands of galaxies, hot plasma and large amounts of invisible dark matter. They cluster together in a hierarchical fashion to make what are called the superclusters or the clusters of galaxy clusters. Filaments are long sheets, walls or needle-like structures made from thousands to millions of galaxies. They are typically hundreds of millions of light years in length and around 20 million light years thick. They are one of the largest structures found in the universe and give the appearance of honey-comb structures in the cosmic web. Cosmic voids, on the other hand, are the vast underdense regions of the universe with negligible structure and thus almost no matter density. The average

density of the cosmic voids is around one-tenth the average density of the universe and a typical void can stretch anywhere from 20 million to hundreds of millions of light years. The abundance of voids can be used to test the non-gaussianity of primordial perturbations, which constrains the models of inflation [29] and they also provide stringent tests for galaxy formation models because of an almost lack of galaxies [30] in them. Figure 1.5 shows the large scale structure formed by the galaxies out to around 2 billion light years. The point where the two slices intersect at the center of the figure is the observational point i.e., Earth.

Over time, the universe keeps getting more and more clustered as gravity keeps pulling more and more matter together. The LSS observations tell us about the strength of gravity in the universe. Different galaxies at different distances from us correspond to different times in the history of our universe as the light from those galaxies takes time to reach us and the speed of light remains constant in space. This means that the further we look out into space, the further we are actually looking back in time. Figure 1.6 shows a simulation of cold dark matter in an expanding universe under the effect of gravity alone. The observable for LSS is the galaxy power spectrum, $P_{gal}(k)$ which is theoretically modelled as:

$$P_{gal}(k) = b^2 P_m(k) + P_s. \tag{1.13}$$

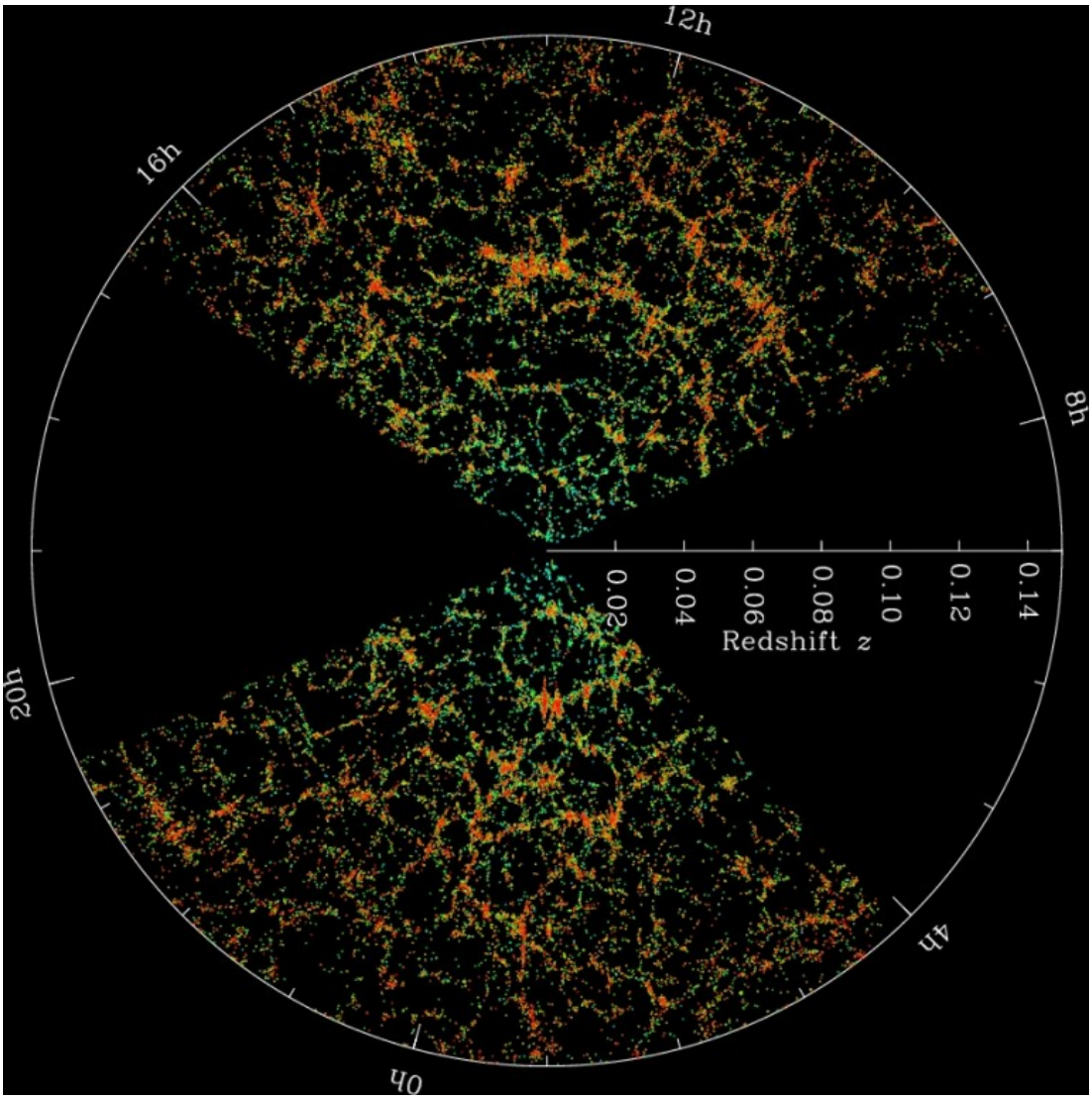

Figure 1.5: A map of the galaxies discovered by the Sloan Digital Sky Survey (SDSS). Each red point in the map is an individual galaxy. The network of galaxies forming thread-like filaments and thick points of intersection i.e., clusters are evident. The black, empty regions inside the web denote the underdense regions, the cosmic voids. A larger redshift corresponds to the galaxies farther from the Earth (center of the figure) and thus much earlier in time. Image credit: M. Blanton and the Sloan Digital Sky Survey.

Here, $P_m(k)$, b and P_s are the matter power spectrum at scale k, the galaxy bias (scale-independent) and the shot noise contributions respectively. The bias reflects

the fact that galaxies act as biased tracers of the underlying dark matter distribution, and P_s arises from the discrete point-like nature of the galaxies as tracers of the dark matter [31].

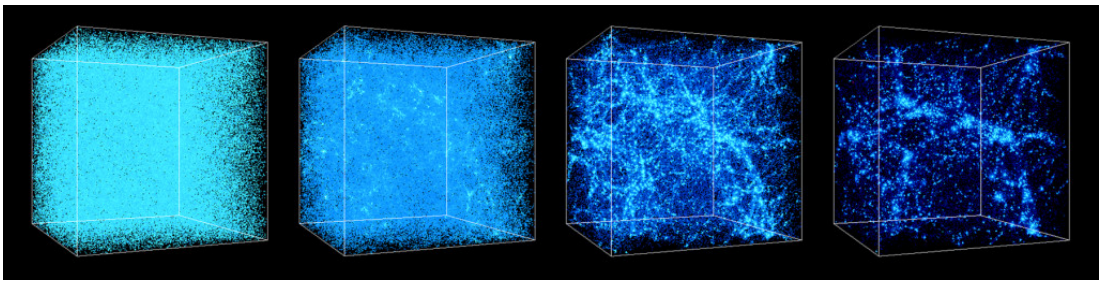

Figure 1.6: A simulation showing the evolution of the universe over time. In earlier times (left), the matter density was almost uniform and then gravity started pulling matter together to form larger patterns. Clustered regions became more clustered and sparse regions became much sparse. Note that the amount of clustering in the present day (far right) is much larger than the one during earlier times (far left). The same is true for sparsity. A web-like structure is thus evident. Also note that as the fabric of the universe itself expands over time, the box shown in the simulation also gets bigger over time. But as the simulation uses comoving coordinates and periodic boundary conditions, the box appears to be the same size. Image credit: National Center for Supercomputer Applications by Andrey Kravtsov (The University of Chicago) and Anatoly Klypin (New Mexico State University).Visualizations by Andrey Kravtsov.https://astro.washington.edu/n-body-shop

Chapter 2

Cosmological Simulations

Current cosmological surveys of large-scale structure take numerous observations each second. In order to extract meaningful information from these surveys, we need rigorous theoretical predictions. However, galaxy structure formation and evolution is highly complicated due to the multi-physics and multi-scale nature of the study. The only way to deal with these complexities is to make use of computer simulation methods. Computer simulations of cosmological evolution are a very indispensable tool in the hands of astrophysicists that help us track the evolution of billions of particles, be it dark matter, galaxies, gas particles or neutrinos, over the course of billions of years. As dark matter acts as the skeleton on which the galaxies form, it also behaves as the backbone of these simulations and is therefore a key component of these simulations.

Based on type of particles, cosmological simulations are usually divided into dark matter-only simulations, such as N-body simulations, and dark matter plus baryons simulations, such as hydrodynamical simulations [32]. Figure 2.1 shows a snapshot of an N-body dark matter-only simulation of our universe. A snapshot refers to the particular distribution of the positions, velocities or accelerations of all the particles in the entire volume of the simulation during a specific instant of time in the history of the evolution of the universe.

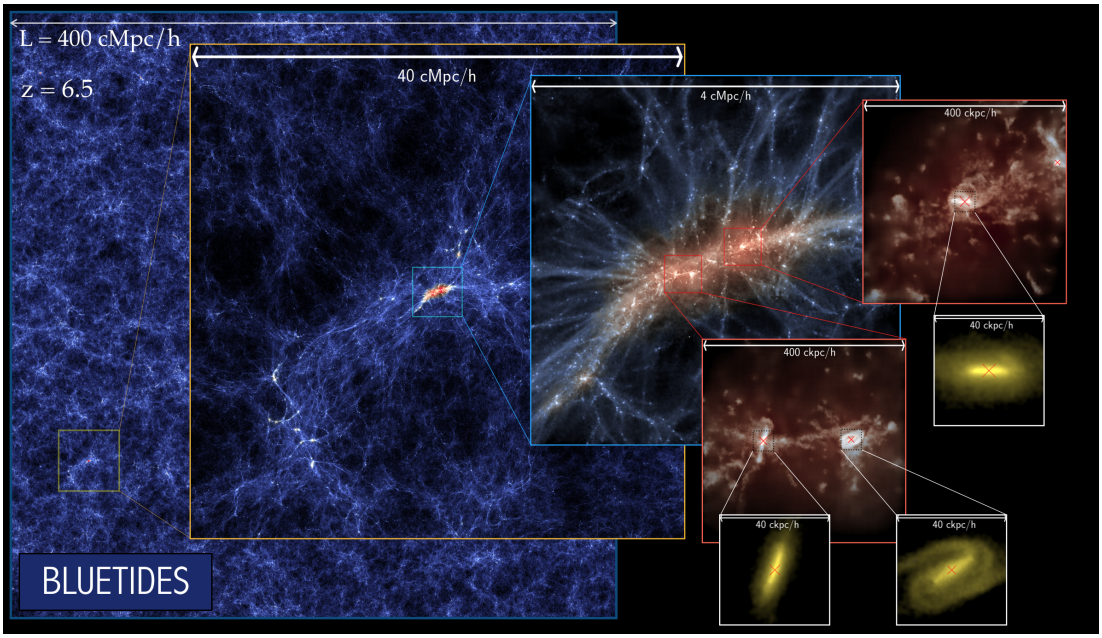

Figure 2.1: A snapshot from the BlueTides simulations. The figure above shows a zoomed-in view of the most massive dark matter halo (10^{13} M_{Sun}) at a redshift of $z = 6.5$. The length of the simulation region is $400c\ Mpch^{-1}$. Image credit: The BlueTides Project[2].

2.1 N-body Simulations

The N-body simulations are a suite of cosmological simulations in which the Cold Dark Matter (CDM) particles are evolved under the effect of gravity alone. CDM refers to that component of Dark Matter (DM) which is known to neither reflect, nor transmit nor absorb any light and has excessively slow thermal velocity, hence termed cold. Dark matter builds the backbone for the formation of galaxies, which are expected to form at the centers of DM overdensities, called halos. An N-body simulation evolves a large number of massive CDM particles interacting with each other through Newtonian gravity only and since our universe began with matter distributed almost uniformly, the simulation starts with particles only slightly perturbed from a uniform grid and nearly at rest. Most cosmological simulations employ Newtonian rather than relativistic gravity, which provides a good approximation since linear structure growth is identical in the matter-dominated regime in the two theories, and non-linear large-scale structure induces velocities far below the speed of light. See [33] for a technical review of N-body simulations.

Typically, we take N^3 (where N usually ranges from 100 to 2000) particles in a comoving cosmological volume of size roughly between 100 Mpc to $10,000$ Mpc. The initial conditions (ICs) of the universe which refer to the initial three-dimensional positions of all of these particles, are typically sampled from a gaussian random field

having a specific power spectrum. Power spectrum is a quantity that denotes the amount of structure at different scales in the universe. It is of utmost importance in cosmology as it is predicted directly in the cosmological models incorporating inflation and dark matter (see sections 1.1.5 and 4.4.1.2). After that, invoking the laws of Newtonian gravity and including the properties of dark energy and various other physical effects, these particles are displaced from their initial condition positions. During this process of evolution which is quantified in timesteps or redshifts, the initial density field (which is gaussian) becomes increasingly non-gaussian and leads to the formation of complex networks of matter, giving rise to prominent structures such as halos, filaments, sheets and voids [34, 35]. Dark matter only N-body simulations numerically solve the Poisson's equation which is a very computationally intensive task in itself. This is because the forces on each one of these billions of simulating particles due to the rest need to be recalculated at each timestep of the evolution. This needs to be done in short time intervals to retain the precision of the approximations and requires that the updates to these particle positions be frequent. Currently, the speed of these simulations is a large bottleneck for cosmological surveys like DESI, EUCLID or LSST.

2.2 QUIJOTE Simulations

Quijote [3] is a suite of $44,100$ simulations that are specifically created to extract the cosmological information embedded in small, nonlinear modes enabling tighter constraints of the cosmological parameters. These simulations are primarily designed to easily quantify the information content of different statistics into the fully nonlinear regime. All Quijote simulations are N-body simulations only, run using TreePM code GADGET-III, which is an improved version of GADGET-II [36]. The initial conditions of all the simulations are generated at $z = 127$. The input matter power spectrum and transfer functions are obtained by rescaling the $z = 0$ matter power spectrum and transfer functions from CAMB [1] [37]. From the input matter power spectrum and transfer functions, displacements and peculiar velocities employing the Zel'dovich approximation [38] (for cosmologies with massive neutrinos) or second-order perturbation theory (for cosmologies with massless neutrinos) are computed. The displacements and peculiar velocities are then assigned to particles that are initially laid on a regular grid.

All simulations have a cosmological volume of 1 $(h^{-1}Gpc)^3$. The majority of the simulations follow the evolution of 512^3 CDM particles (plus 512^3 for simulations

[1]CAMB is a cosmology code for calculating cosmological observables, including CMB, gravitational lensing, source count and 21cm angular power spectra, matter power spectra, transfer functions and background evolution. The code is in Python, with numerical code implemented in fast modern Fortran.

with massive neutrinos), which is considered to be the fiducial configuration and serves as the benchmark against with simulations run with variations in parameters are compared. The snapshots are generated and saved at redshifts 0, 0.5, 1, 2, and 3. A snapshot from the Quijote simulations is shown in Figure 2.2.

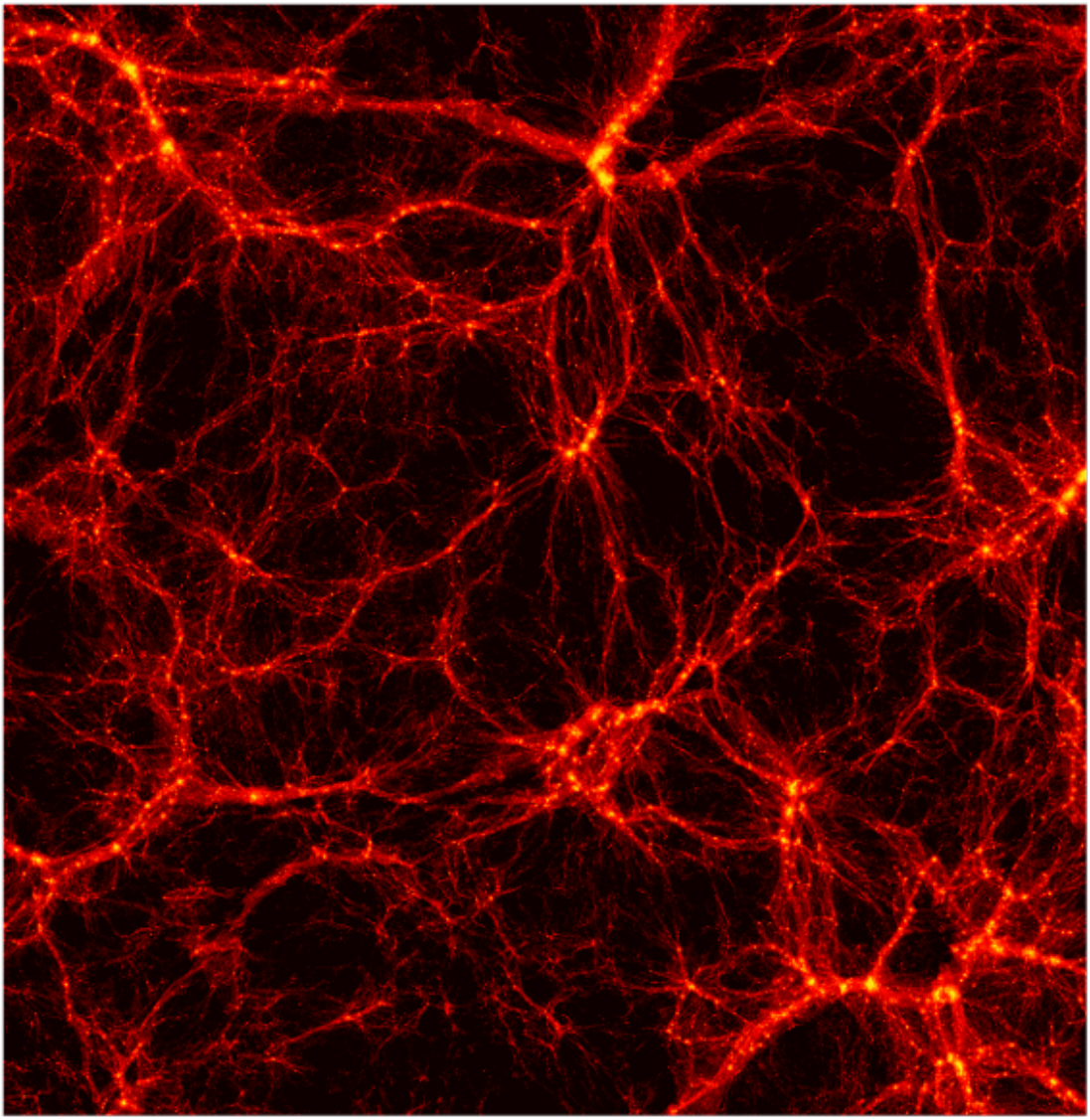

Figure 2.2: A two-dimensional snapshot from the QUIJOTE suite generated using the N-body code GADGET-III. Image credit: The Quijote Simulations[3].

2.3 COLA: Fast LPT-based Simulations

One of the fast approximations to the N-body codes is using Lagrangian Perturbation Theory (LPT) methods to solve the equations of structure formation and evolution. All N-body simulation codes perform numerous timesteps in order to obtain a much reasonable approximation to structure formation both at large and small scales. However, Lagrangian Perturbation Theory (LPT) or its modifications very well describe the large scales ($\sim 100 Mpch^{-1}$ at $z = 0$).

In N-body codes, the time integration for the large scales solves for the linear growth factor. Therefore, by using only a few timesteps leads to a bad estimate of the linear growth factor which miscalculates the power at large scales. But the exact value of the linear growth factor is well-known (see for example, [22]). This fact is made use of in various approximate methods of N-body simulations that use LPT and the large and small scales in N-body codes are decoupled. The large scales are evolved using second-order LPT and the small scales using a full-blown N-body code. This decoupling is enabled by transforming the system to a frame of reference that is comoving with the LPT observers and recasting the equations accordingly. This allows to take large N-body time-steps, thus saving a lot of computations, while at the same time keeping the accuracy on the largest scales.

COLA or COmoving Lagrangian Acceleration method [39] is an example of such an LPT-based approximation to the traditional N-body method for solving for the large scale structure (LSS) in a frame that is comoving with observers following trajectories calculated in LPT. COLA can generate large ensembles of accurate mock halo catalogs very cheaply, that are used to study weak lensing and galaxy clustering. These catalogs are essential for performing detailed error analysis for ongoing and future surveys of LSS. A snapshot of the density field of the Universe generated with the COLA code is shown in Figure 2.3.

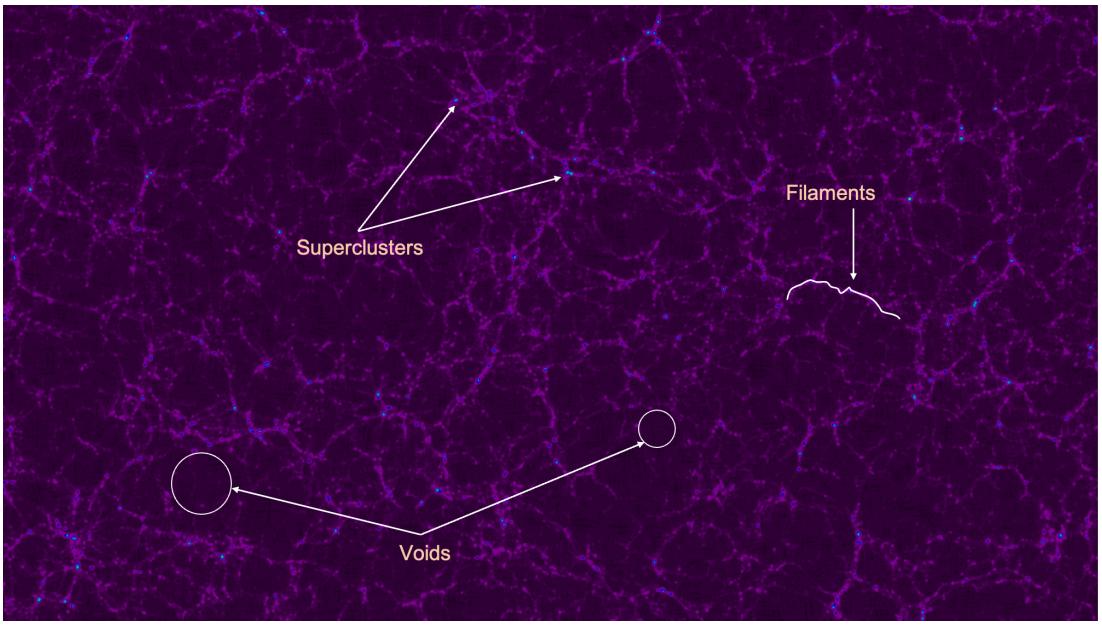

Figure 2.3: A cosmic web of the Universe. It represents the overdensity field of the cold dark matter particles in a region of side 1 $Gpch^{-1}$. CDM particles were placed on a grid and the system was evolved from a redshift of $z = 9$ up to a redshift of $z = 0$ using the COLA method with the standard cosmological model parameters given in Table 1.1.

This thesis makes use of both the QUIJOTE and COLA simulations. The two prominent codes to implement COLA are L-PICOLA [40] and MG-PICOLA [41]. MG-PICOLA contains support for simulations with massive neutrinos and has been used in the works described in this thesis.

Chapter 3

Deep Learning

Deep learning (DL) is a subset of machine learning that uses Artificial Neural Networks (ANNs) to carry out various supervised, unsupervised and reinforcement learning algorithms. It essentially uses a neural network (NN) with three or more layers and these NNs attempt to mimic the behavior of the human brain, although far from matching its ability, and thus "learn" from large amounts of data. Additional hidden layers help in optimizing and refining the accuracy of the network.

Deep learning is extensively used in numerous Artificial Intelligence (AI) applications and services and lie behind various day-to-day products and services such as voice assistants, language translators, recommender systems, fraud financial detectors, spam checkers, self-driving cars, etc. Deep learning enables us to perform various analytical

and physical tasks and improving automation all without any human intervention.

3.1 Supervised and Unsupervised Learning

Supervised learning is a type of machine learning technique that uses labeled datasets. These datasets are designed to train or "supervise" the models. The models use these labeled inputs and outputs and in an attempt to minimize the error between the actual and predicted labels, learn the representations of these datasets over time.

Supervised learning can be broadly classified into two types of problems:

1. **Classification**: It is the process of classifying the data into different classes or categories, such as classifying images into dogs or cats or classifying the handwritten digits into their actual numeric digit forms, classifying an email as spam or non-spam, etc. They can be further divided into various types such as binary classification, multi-class classification, etc. Some examples of classification algorithms are linear classifiers, support vector machines (SVMs), decision trees, random forests, and convolutional neural network (CNN)-based classifiers.

2. **Regression**: This is another type of supervised learning that uses an algorithm to understand the relationship between dependent and independent variables

in a data. Rather than classifying the data into various classes, a regression model uses the existing data to train and predict numerical values based on different data points. They are used, for example, to predict the future prices of stocks, to predict the house market rates, or changes in health trends in a demographic, etc. Prominent regression algorithms are linear regression, polynomial regression, and lasso regression.

3.2 Neural Networks

Deep learning neural nets, also called the artificial neural nets, use a combination of weights, biases and inputs to try to simulate the learning process of the human brain. These elements work together to accurately recognize, classify, and describe objects within the data. These networks contain millions of neurons or nodes, each learning an aspect of data, like a curve, an edge or some depth, for example, in case of image data.

These networks contain a number of nodes that are connected to each other and form a layer. Numerous layers are connected in an hierarchical fashion, such that each layer builds upon the previous layer to refine and optimize the output of the task that can be a categorization or a prediction. The output of each layer acts as the input of the next layer. The flow of computations through the network is called

forward propagation. The input and output layers of a neural network are 'visible' layers while the layers in between are hidden from the user and are thus called the hidden layers. The input layer is the first layer where the deep learning model ingests the data for processing and the output layer is the final layer that makes a prediction or a classification. A typical neural network with three hidden layers is shown in Figure 3.1.

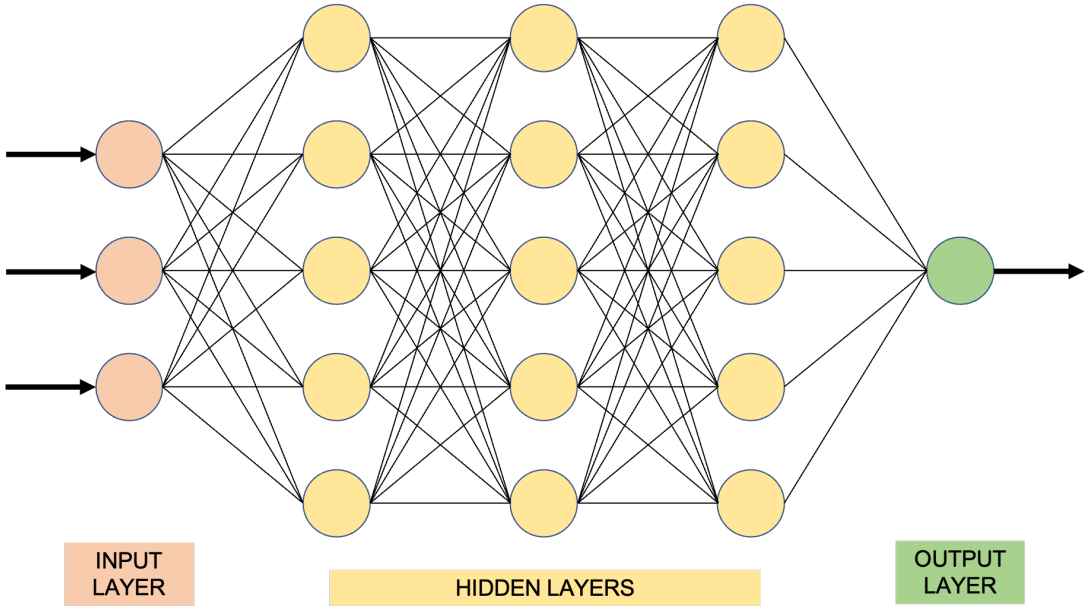

Figure 3.1: A typical neural network. The first layer (orange) is the input layer that represents the data, followed by three hidden layers (yellow) and finally the output layer (green). Note how each neuron or node in a layer is connected to all the nodes of the next layer. This ensures that all the information extracted by one layer is passed on to all possible nodes in the next layer which combine it to form higher-level representations.

Another important process in deep learning algorithms is backpropagation. The mode or network uses optimization algorithms like Gradient Descent (GD) to calculate the error in predictions and compute the loss function. It then propagates this loss

function back through the network to adjust the weights and biases of each node. This is called the training process of the network. The forward pass of data and backward pass of loss function allow the network to effectively learn the representations in the data by continuing to reduce the error in predictions. This makes the algorithm more accurate over time.

Although Figure 3.1 shows a simple neural network, practical deep learning networks are usually highly complex and deep[1] in architecture as the real world data can be very big and complicated. The more the number of hidden layers, the deeper the network is. Figure 3.2 shows various deep neural nets for the ImageNet [42–45] data. The topmost network in the figure is resnet, a very deep feed-forward neural network with hundreds of layers and various skip connections to jump over some layers.

3.3 Loss Functions

A loss function (or cost function, loosely) is a metric that we evaluate to quantify how happy or unhappy we are with our model. It compares the predictions of our model with the actual ground truth and helps in updating the weights and biases of the model.

Loss function is a measure of the accuracy of the neural network with respect to a

[1]Depth refers to the number of hidden layers used in the network.

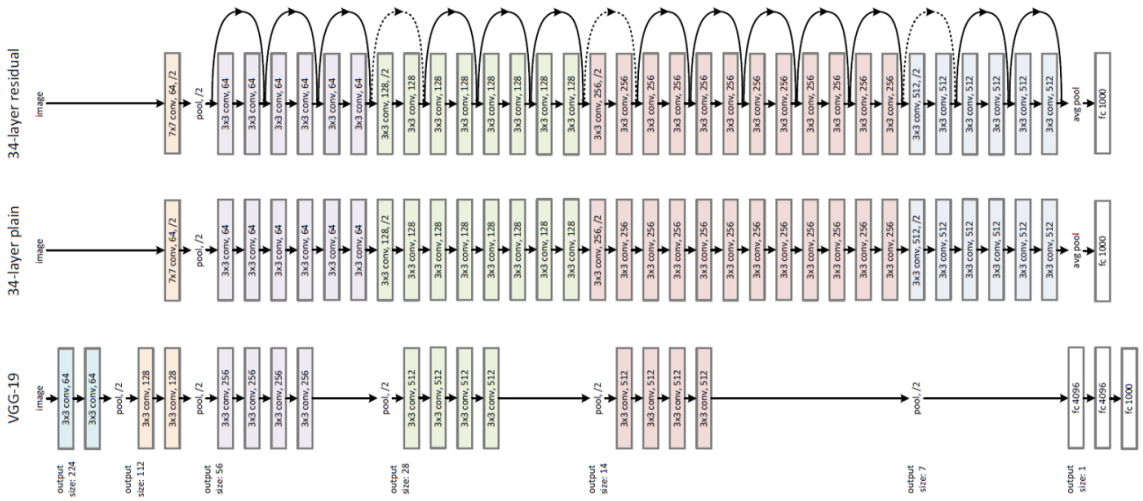

Figure 3.2: Example network architectures for ImageNet data. Bottom: the VGG-19 model [4] (19.6 billion FLOPs) as a reference. Middle: a plain network with 34 parameter layers (3.6 billion FLOPs). Top: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. Image credit: He, K. et al (Proceedings of the IEEE conference on computer vision and pattern recognition 2016).

given training sample and expected output. It provides the performance of a neural network as a whole. In deep learning, the goal is to minimize the cost function. For that, we use the concept of gradient descent.

The loss function in a neural network quantifies the difference between the expected outcome and the outcome produced by the machine learning model. From the loss function, we can derive the gradients which are used to update the weights. The average over all losses constitutes the cost. A loss function is denoted by L.

"The cost function reduces all the various good and bad aspects of a possibly complex system down to a single number, a scalar value, which allows candidate solutions to be ranked and compared" (Page 155, [46]).

3.3.1 Mean Squared Error (MSE) Loss

Mean squared error (MSE) is the most commonly used loss function and is specifically used in regression. The loss is the mean overseen data of the squared differences between true and predicted values, or writing it as a formula.

$$L(y, \widehat{y}) = \frac{1}{N} \sum_{i=0}^{N} (y - \widehat{y})^2 \tag{3.1}$$

where $\hat{y}$ is the predicted value. MSE is sensitive towards outliers and given several examples with the same input feature values, the optimal prediction will be their mean target value. This should be compared with Mean Absolute Error, where the optimal prediction is the median. MSE is thus good to use if we believe that the target data, conditioned on the input, is normally distributed around a mean value, and when it's important to penalize outliers extra much. MSE is also good when we want large errors to be significantly (quadratically) more penalized than small ones.

3.3.2 Binary Crossentropy (BCE) or Log Loss

Binary crossentropy is a loss function that is used in binary classification tasks. These are tasks that answer a question with only two choices (yes or no, A or B, 0 or 1,

left or right). Several independent such questions can be answered at the same time, as in multi-label classification or in binary image segmentation. Formally, this loss is equal to the average of the categorical crossentropy loss on many two-category tasks.

The binary crossentropy loss function calculates the loss of an example by computing the following average:

$$\text{Loss} = -\frac{1}{\text{output}} \sum_{i=1}^{\text{output size}} y_i \,.\, log(\hat{y}_i) + (1 - \hat{y}_i) \,.\, log(1 - \hat{y}_i) \tag{3.2}$$

where $\hat{y}_i$ is the ith scalar value in the model output. y_i is the corresponding target value, and output size is the number of scalar values in the model output. This is equivalent to the average result of the categorical crossentropy loss function applied to many independent classification problems, each problem having only two possible classes with target probabilities y_i and $1 - y_i$.

3.4 Semantic Segmentation

Semantic Image Segmentation refers to the task in computer vision in which we label specific regions of an image in accordance with what is being shown in it. The aim of semantic segmentation is to read an image and label each pixel of that image into

a corresponding class of what is being represented by that pixel. As the predictions are for every pixel of the image, it is also called dense prediction or a pixel-wise classification task. An example of semantic segmentation is shown in Figure 3.3.

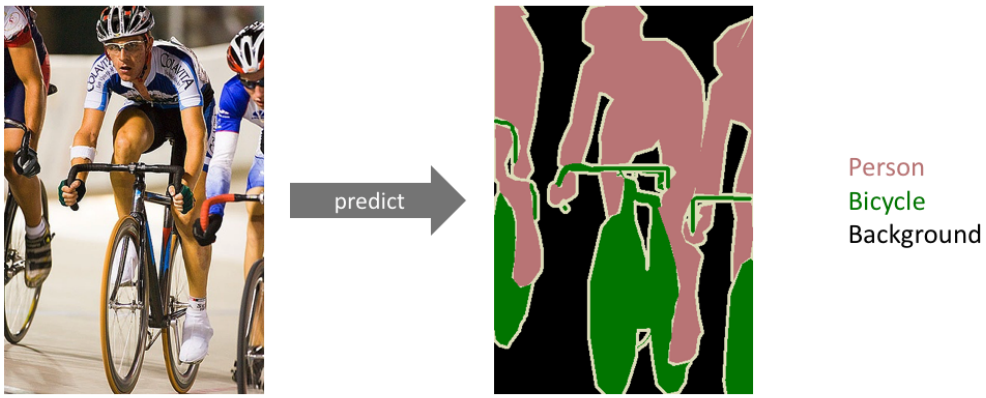

Figure 3.3: Various regions of the image (left) are segmented to corresponding classes (right) by a semantic image segmentor; the bicycles are in green, the people in light pink, and the background in black. Image credit: The PASCAL VOC Dataset.

It is to be noted that in semantic segmentation, we do not separate the instances of the same class; we only care about denoting a class or category of each pixel. In other words, if our image consists of two non-connected islands in an ocean, semantic segmentation will classify the ocean as one class and the two islands as another class. There exists a different class of models, known as instance segmentation models, which do distinguish between separate objects of the same class. So for the same image, instance segmentation will return three classes; one for the ocean and one each for the two islands. The difference between semantic and instance segmentation

is shown in Figure 3.4.

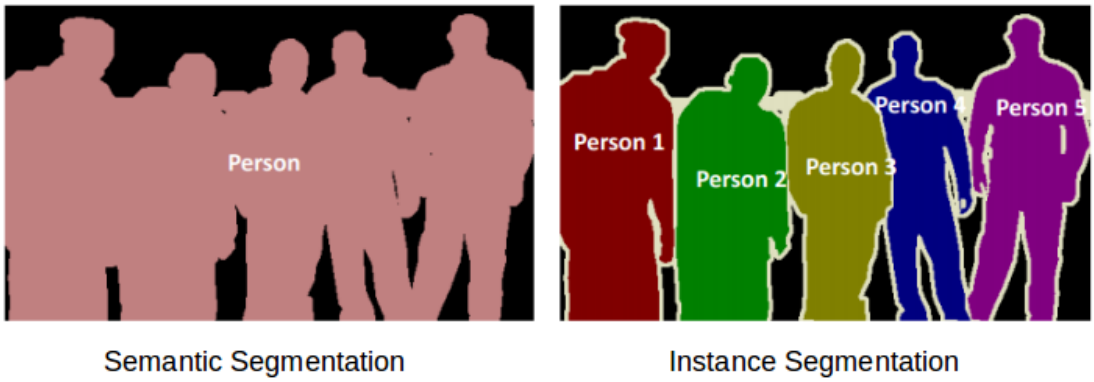

Figure 3.4: Difference between semantic segmentation (left) and instance segmentation (right). The left figure shows only two classes: the background in black and the five people in the foreground in light pink. The figure on the right shows seven classes: one for background (black), one each for the five persons (red, green, dull yellow, blue and purple) and one for some hands on some shoulders (silver). Image credit: The PASCAL VOC Dataset.

There are various applications of segmentation models including but not limited to:

1. **Autonomous vehicles**: Cars need to be equipped with the required perception to understand, estimate, and learn from their surroundings so that self-driving cars can safely integrate into our existing roads. An example of this is shown in Figure 3.5.

2. **Medical Image Diagnostics**: Analysis of various scans like MRI, CT, X-ray and Ultrasound of brain, heart, lungs and other organs performed by radiologists can be augmented by machines which helps in reducing the time required to run concerned diagnostic tests.

3. **Geographic Information Systems**: In Geographic Information Sciences

(GIS), semantic segmentation can be used for land cover classification or for extracting roads or buildings from satellite imagery. An example of this is shown in Figure 3.6.

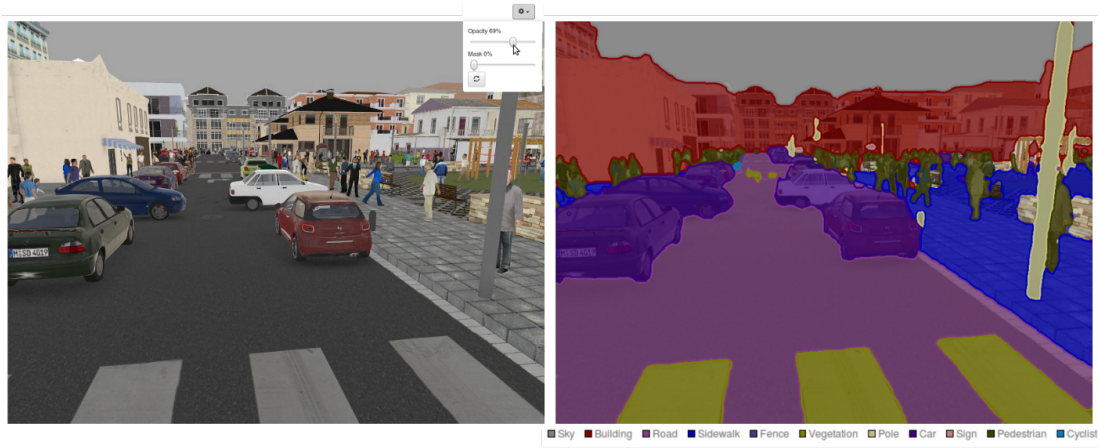

Figure 3.5: A screenshot of a scene in traffic (left) and it's real time segmented (right). Image credit: The SYNTHIA-Rand-CVPR16 dataset[5].

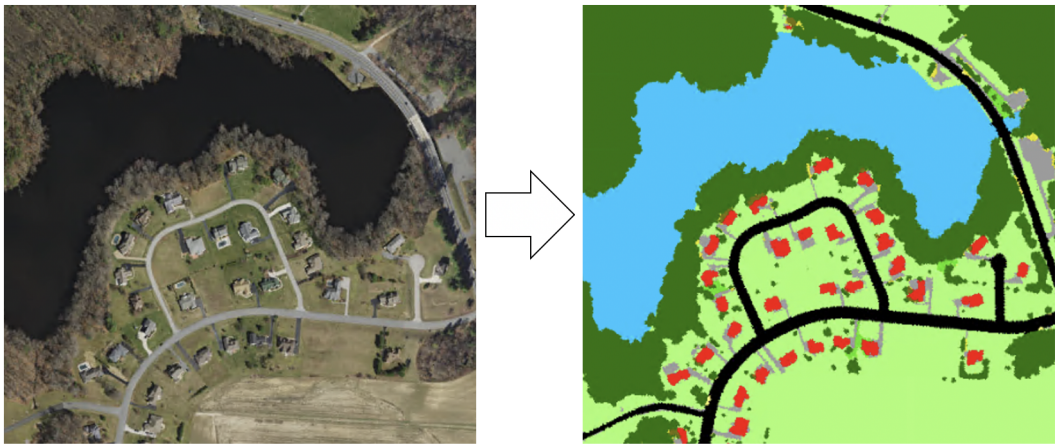

Figure 3.6: A satellite image on the left after feeding through a trained image segmentation model segments various regions of the image where pixels match closely with each other. Image credit: The ArcGIS Developers.

3.5 Convolutional Neural Networks

A Convolutional Neural Network (ConvNet or CNN hereafter) is a deep learning algorithm which takes an input image, assigns importance (learnable weights and biases) to various aspects or objects in the image and is able to differentiate one from the other. The architecture of a ConvNet is analogous to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. This is depicted in Figure 3.7.

Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area. A ConvNet is able to successfully capture the spatial and temporal dependencies in an image through the application of relevant filters.

ConvNets are being used almost everywhere with some of their applications being in but not limited to Image Classification, Image Retrieval, Image Detection, Image Segmentation, Face Recognition, Video Classification, Pose Recognition, Game playing, Image Captioning and Stylistic Artworks. They are called convolutional networks because their basic operation is related to the convolution of two signals which is the element-wise product and sum of a filter and the signal.

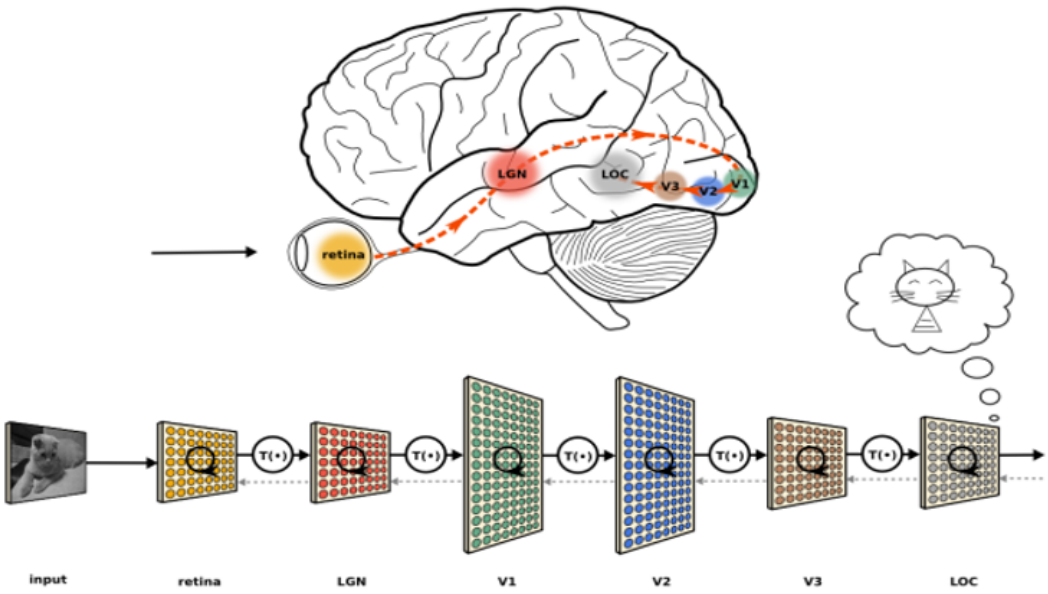

Figure 3.7: Our eye observes a particular image and the neural networks interpret the information from the field of vision in a hierarchical manner. LGN, V1, V2, V3, etc. can be thought of as the activation maps after the output of previous layer passes through a hidden layer and selectively activating some neurons. Each information from the previous layer is passed on to the next layer which combines them together to form higher-order representations. This series of information extraction carries on until the information in the raw image is fully extracted and the image is finally interpreted by the brain after passing through the last layer. Image credit: Jonas Kubilius under CC BY 4.0 license.

3.5.1 Filters

A lot of the details of what makes up an image is actually contained in its edges or outlines. It's one of the reasons why we can easily distinguish objects in cartoon sketches. ConvNets also extract features from images by detecting edges, which represent image features. They use filters which are sets of learnable weights that

detect spatial patterns such as edges or curves in an image by detecting the changes in intensity values of the image. A filter can be thought of as storing a single template or pattern. An example of the application of a filter on an image is shown in figure 3.8.

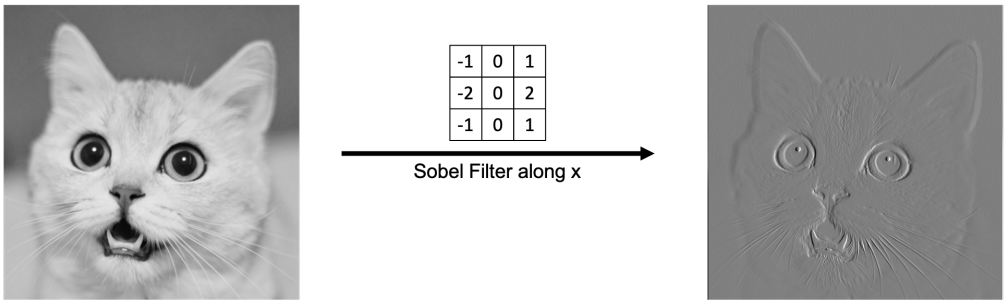

Figure 3.8: The application of the **Sobel filter** along the x-axis in an image. This filter will detect edges in the image.

While the first few layers of a CNN are comprised of edge detection filters (low level feature extraction), deeper layers often learn to focus on specific shapes and objects in the image. Although the filters are hand-engineered in primitive methods, with enough training, ConvNets have the ability to learn these filters by itself.

3.5.2 Activation functions: Rectified linear unit (ReLU)

The input data (x) in a neural network is passed through a linearity to obtain $wx+b$, where w are the weights and b are the biases. After passing through this linearity, an

activation function (ϕ) is responsible for transforming this summed weighted input from the node into the activation of the node or output for that input as:

$$\text{output} = \phi(wx + b) \tag{3.3}$$

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It is given as:

$$\text{ReLU}(x) = \max(0, x) \tag{3.4}$$

Figure 3.9 shows the ReLU function.

ReLU has become the most prominent activation function for many types of neural networks as the models implementing ReLU are easier to train and less computationally intensive compared to other activations. Various modifications of ReLU include Leaky ReLU, Parametric ReLU, and Gaussian-error linear unit (GELU).

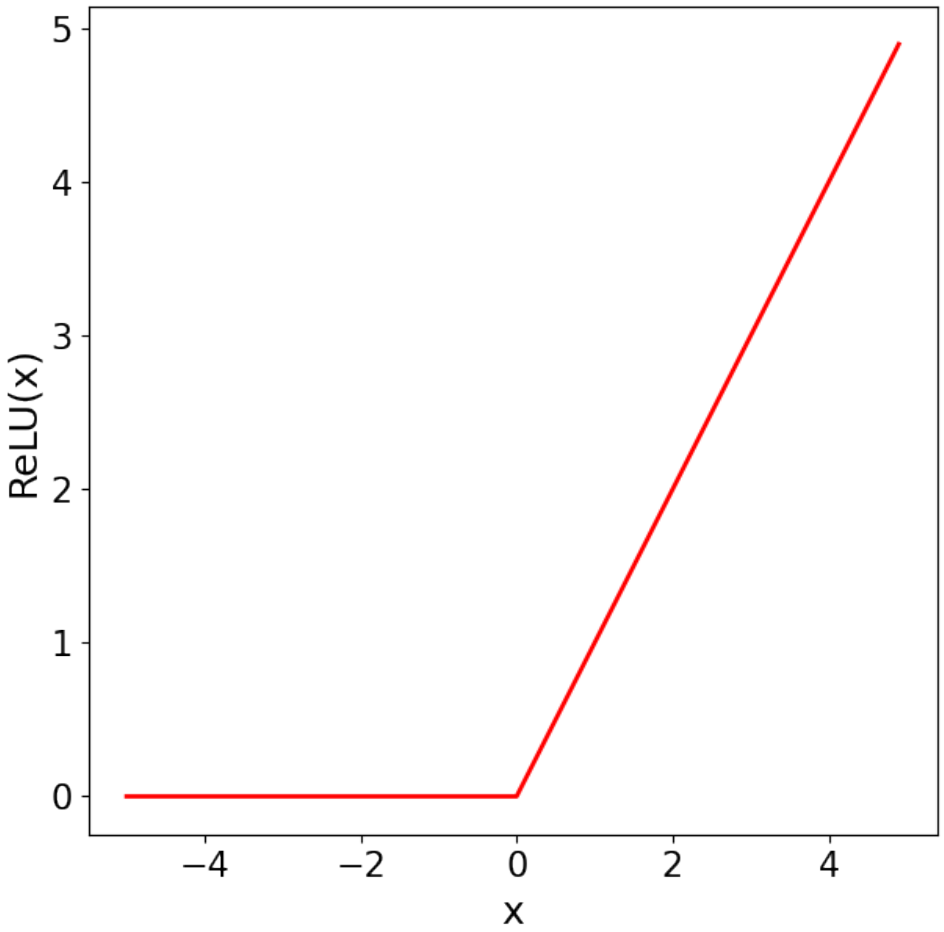

Figure 3.9: The ReLU activation function.

3.5.3 Pooling

The output feature maps (the outputs of neural network layers) are sensitive to the location of the features in the input. This sensitivity limits the ability of the network to generalize well over the entire training data. One approach to address this issue

is to down sample the feature maps. The down sampled feature maps then become more robust to the changes in the position of the feature in the image. This is known as local translation invariance.

Pooling layers are one of the downsampling techniques that summarize the presence of features in the patches of the feature maps. Max pooling and average pooling are two common pooling techniques that summarize the average presence of a feature and the most activated presence of a feature respectively.

Max Pooling is a pooling operation that calculates the maximum value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. It is usually used after a convolutional layer. Figure 3.10 shows the max pooling operation.

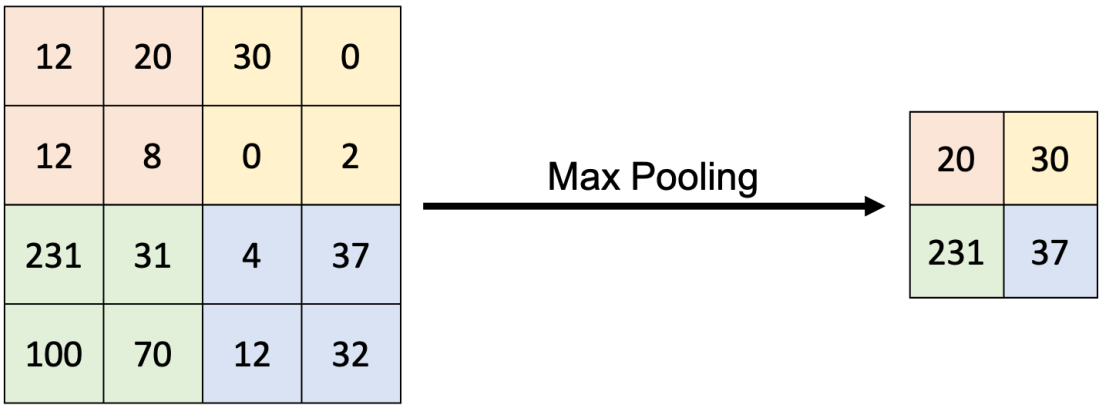

Figure 3.10: The Max Pooling operation. The most activated features (here pixel values) in each 2×2 region of the original matrix on the are extracted.

3.5.4 U-Net

U-Net [47] is a CNN-based architecture that was designed by Ronneberger, Fischer and Brox for fast and precise segmentation of images in the field of biomedical sciences. Semantic segmentation is also known as pixel-wise classification, where we classify each pixel of an image as belonging to a particular class. In the field of semantic segmentation, it is still one of the most popular end-to-end architectures and while performing extremely well, has surpassed many other models in various challenges. Figure 3.11 shows the architecture of a U-Net.

It consists of a contracting path and an expansive path. The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3×3 convolutions, each followed by a rectified linear unit (ReLU) and a 2×2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes. In total, the

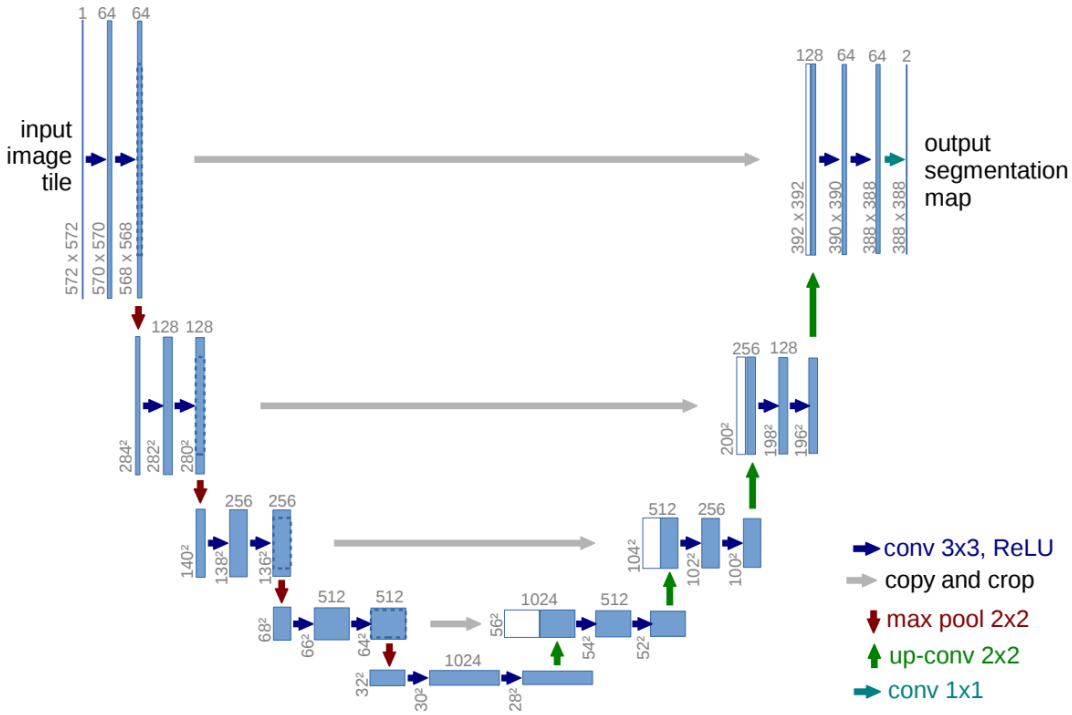

Figure 3.11: The U-Net architecture. The left path is encoding or downsampling which forgets the information about "where" in the figure and tries to extract the information of "wha" in it. After the bottleneck at the bottom, the decoding or upsampling starts. This again reconstructs the "where" of the information in the image and loses the "what" of it. The output is the segmented map of the input image tile. Image credit: The original U-Net paper.

network has 23 convolutional layers.

3.6 Generative Adversarial Networks (GANs)

A Generative Adversarial Network, or GAN [48] for short, is a type of deep learning approach to generative modeling using neural networks like convolutional neural nets

(CNNs) or densely connected (linear) layers.

Generative modeling refers to the unsupervised learning technique in machine learning that involves automatically discovering and learning the regularities or patterns in data, or understanding the underlying statistical distribution of the data in such a way that the generative model can generate or output new examples that can no longer be distinguished from the real samples and seem to come from the original dataset.

GANs work in a game-theoretic approach, unlike a conventional neural network. It consists of two networks which are adversaries to each other; one of them is a Generator (G) and the other is the Discriminator (D). The whole GAN network tries to learn generating new fake data from a training distribution through this two-player game between adversaries, who are in a constant battle throughout the training process. Since an adversarial learning method is adopted, we need not care about approximating intractable density functions.

The goal of the generator (G) is to generate real-looking data (or images in case of computer vision tasks) while being trained on the generator network. The discriminator (D) on the other hand, tries to classify examples as either real (from the domain of the training data) or fake (generated by the generator). Both G and D are battling constantly wherein G tries to fool D by generating images that look more and more real and D tries not to be fooled and classify the images generated by G as fakes as

much as possible. To generate very real-looking images, a really good G is needed otherwise it will never be able to fool D and convergence will never be achieved. Similarly, if D is not good, it will classify both the fake and the real images as real, even if the fake images do not make any sense. This means that the model is never actually trained and hence never produces the desired output. The algorithm works as follows:

† The input is a random noise vector (z) that can be Gaussian distributed (usually drawn from a unit-normal distribution $N(0, 1)$), uniform distributed or can be some other structured input.

† The generator G, takes the latent vector z and parameterized by a neural network, gives the output $G(z)$.

† The discriminator D, also parameterized by a neural network, inputs real samples x and fake samples $G(z)$ i.e., the samples generated by G, and outputs scores $D(x)$ and $D(G(z))$ respectively. Each score represents the belief of discriminator in the sample being real i.e. coming from the distribution of the real data, $p_{data}(x)$. This score, when scaled to $[0, 1]$ can also be loosely interpreted as an implicit likelihood of the data given D i.e. $p(x|D)$.

† The predictions of D are compared to the actual, true labels and a loss is computed ($L(D, G)$).

† This loss is backpropagated, first through D and then through G to update the weights and biases of both.

† Steps 1-5 are repeated over several epochs while iterating through the entire dataset.

The loss function in case of GANs is given by:

$$\min_G \max_D L(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[1 - \log D(G(z))], \tag{3.5}$$

where $\mathbb{E}$ is the expectation function.

As a GAN is effectively a min-max problem, G and D compete so that G minimizes and D maximizes the above loss function. A typical GAN workflow is shown in Figure 3.12.

One of the disadvantages of GANs is that they are more unstable to train as both the networks (G and D) are trained from a single backpropagation. In addition, GANs are therefore very sensitive to the choice of objective function as well as the choice of hyperparameters. Furthermore, GANs can not be used to perform any inference queries.

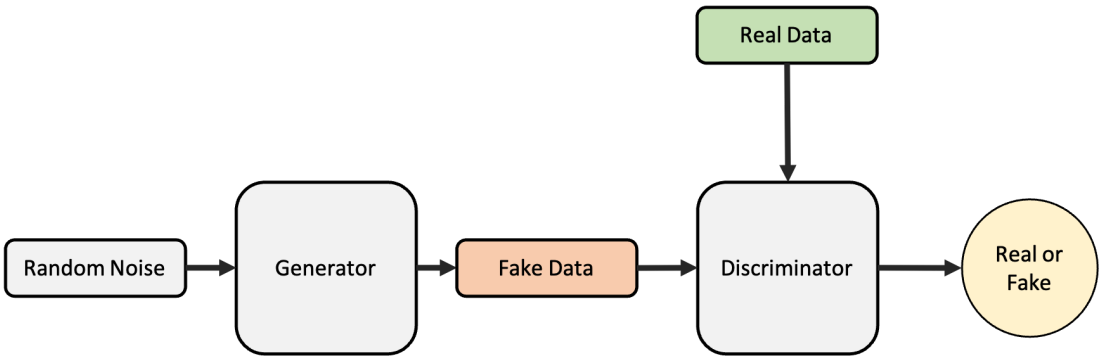

Figure 3.12: The Generator takes a random noise vector and outputs samples of fake data. The Discriminator compares this fake data from the Generator with the real data and outputs the degree of realness of the fake data.

In spite of these shortcomings, GANs are an exciting and rapidly changing field, delivering on the promise of generative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks such as translating photos of summer to winter or day to night, and in generating ultra-realistic photos of objects, scenes, and people that even humans cannot tell are fake.

3.6.1 Conditional GANs

GANs are a type of generative models that are capable of generating new random plausible data samples similar to the training dataset. But they suffer from a serious practical drawback. There is no way to control the types of images other than trying to figure out the complex relationship between the latent space input to the generator

and the generated images. Conditional GANs or CGANs [49] are a modified form of conventional GANs that involve the conditional generation of data by a generator model. Data generation can be conditional on a class label if available, allowing the targeted generation of data of a given type. This is shown in Figure 3.13.

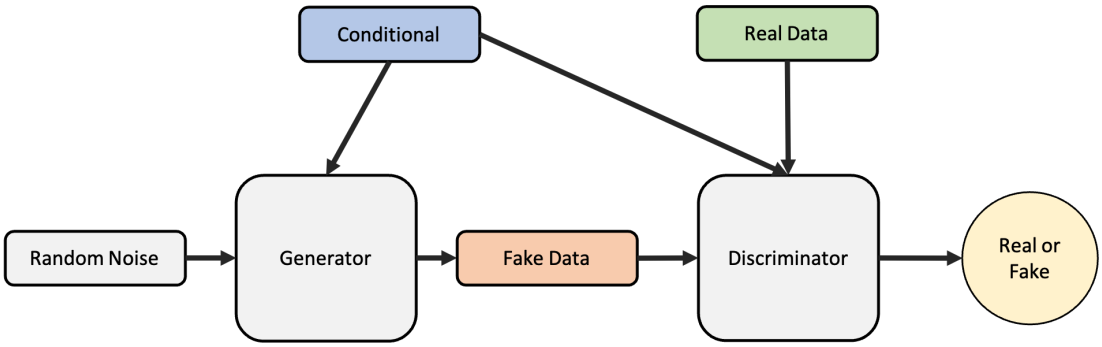

Figure 3.13: A typical Conditional Generative Adversarial Network (CGAN). A random noise and the condition is fed to the Generator which then generates fake data. The Discriminator takes the fake data, the real labels/conditions, and the real data in order to determine the probability of the fake data being real.

A Conditional GAN provides two added advantages to the model:

1. **Faster convergence**: The convergence of the model will be faster. The random distribution that the fake images follow will start to have a distribution.

2. **Controlled output**: The output of the Generator can be controlled at any time by providing the label for the data that needs to be generated.

3.7 Hyperparameters

In Machine and Deep learning, hyperparameters are the variables which determine the network structure (e.g. the number of hidden layers) and the variables which determine how the network is trained (e.g. the learning rate). Hyperparameters are set before training i.e. before optimizing the weights and bias. There are basically two kinds of hyperparameters: 1) related to the network structure, and 2) related to the training process.

3.7.1 Network hyperparameters

Hyperparameters that characterize the architecture of the network are called network hyperparameters. They determine the strength, capacity, robustness and depth of the network. Various network hyperparameters include the number of hidden layers, dropout, initialization of the network weights, and activation functions.

3.7.2 Training hyperparameters

These hyperparameters control various features of the training including but not limited to speed, converging efficiency, optimization type, and the training time. Various

hyperparameters used for training a network include the learning rate, number of epochs and the batch size.

The learning rate controls the amount by which the weights and biases of a model are changed in response to the estimated error. A small learning rate could result in long training times while a large learning rate could lead to an unstable training process or the model never reaching the global minima of the loss function. An epoch occurs when the entire data passes through the model once. For example, if a model needs to be trained on $50,000$ images in batches of 100 images, an epoch occurs when all the $50,000$ images pass through the model once. The batch size is thus 100 and the number of batches in one epoch is $50,000/100 = 500$.

There are various methods available in literature for finding the most optimized hyperparameters for a given learning problem such as manual search, grid search, random search, and Bayesian optimization.

Chapter 4

NECOLA: Towards a Universal Field-level Cosmological Emulator

In this Chapter, we develop a deep convolutional neural network that models the process of structure formation in the universe.

4.1 Abstract

We train convolutional neural networks to correct the output of fast and approximate N-body simulations at the field level. Our model, Neural Enhanced COLA, –NECOLA–, takes as input a snapshot generated by the computationally efficient

COLA code and corrects the positions of the cold dark matter particles to match the results of full N-body Quijote simulations. We quantify the accuracy of the network using several summary statistics, and find that NECOLA can reproduce the results of the full N-body simulations with sub-percent accuracy down to $k \simeq 1\ h\mathrm{Mpc}^{-1}$. Furthermore, the model, that was trained on simulations with a fixed value of the cosmological parameters, is also able to correct the output of COLA simulations with different values of Ω_m, Ω_b, h, n_s, σ_8, w, and M_ν with very high accuracy: the power spectrum and the cross-correlation coefficients are within $\simeq 1\%$ down to $k = 1\ h\mathrm{Mpc}^{-1}$. Our results indicate that the correction to the power spectrum from fast/approximate simulations or field-level perturbation theory is rather universal. Our model represents a first step towards the development of a fast field-level emulator to sample not only primordial mode amplitudes and phases, but also the parameter space defined by the values of the cosmological parameters.

4.2 Introduction

In order to extract valuable information about fundamental physics from cosmic surveys, we need theoretical predictions to confront the collected data. On semi-linear scales, analytic tools like perturbation theory [50] can be used to provide such theoretical predictions. However, on non-linear scales, where a large amount of cosmological information resides [e.g. 51–70], numerical simulations become necessary.

Cosmological simulations can be classified into two broad categories: 1) N-body simulations that model the matter field accounting only for the force of gravity, and 2) hydrodynamic simulations that model not only gravity but also fluid hydrodynamics and astrophysical effects such as the formation of stars and feedback from black holes. While computationally more efficient than hydrodynamic simulations, N-body simulations are still expensive, and running large sets or high-resolution simulations require a significant computational cost [e.g. 51, 71–79]. To overcome this, several methods have been developed that are much less computationally demanding but come at the expense of being less accurate (e.g., ALPT [80], PThalos [81], PINOCCHIO [82], FastPM [83], COLA [40, 84, 85], EZMOCKS [86], FlowPM [87], PATCHY [88], log-normal models [89, 90], HALOGEN [91], MUSCLE-UPS [92], QPM [93], HaloNet [94] and mass-Peak Patch [95, 96]).

Being able to run fast and accurate simulations is of main importance in cosmology in order to provide the theoretical predictions needed to retrieve the maximum information from cosmological surveys. In this work, we try to build a bridge between the fast and approximate COLA simulations, and the expensive and accurate full N-body simulations using deep learning. Deep learning techniques have been used more recently to generate superresolution realizations of the full phase-space matter distribution of the Universe from the low-resolution N-body simulations [97, 98]. We build on the work of [99] and [100] who used neural networks to find the mapping between the displacement field generated by the Zel'dovich approximation to the one from fast

and full N-body simulations, respectively. In this work, we train convolutional neural networks to correct the particle positions from COLA simulation snapshots to match those of full N-body Quijote simulations. The most important conclusion of our work is that our model seems to be universal, i.e., once trained on simulations with a fixed value of cosmological parameters, our network is able to correct the particle positions of COLA simulations with any other cosmology with surprising accuracy: the power spectrum is accurate at the 1% level down to $k = 1\ h\mathrm{Mpc}^{-1}$.

This paper is organized as follows. In Section 4.3, we describe the simulations we use and the architecture of our neural network model. We present the results of the trained network in Section 4.4. Finally, we draw our conclusions in Section 4.5.

4.3 Methods

In this section, we describe the two types of simulations we used, together with the model architecture and the training procedure.

4.3.1 Simulations

4.3.1.1 Full N-body simulations

We made use of the Quijote full N-body simulations [51] to both train and test the model. The simulations used in this work follow the evolution of 512^3 cold dark matter (CDM) particles (plus 512^3 neutrino particles in the case of massive neutrino cosmologies) from $z = 127$ down to $z = 0$ in a periodic volume of $(1000\ h^{-1}\mathrm{Mpc})^3$. We train the network using a set of 100 simulations from the fiducial cosmology, where the values of the cosmological parameters are fixed to: $\Omega_\mathrm{m} = 0.3175$, $\Omega_\mathrm{b} = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $\sigma_8 = 0.834$, $w = -1$, $M_\nu = 0.0$ eV. These simulations are only different in the value of the initial random seed.

We test the accuracy of our network on simulations with very different cosmologies to the one used in the training. For this, we made use of 100 of 2,000 simulations of the latin hypercube contained in the Quijote simulations, where the values of the cosmological parameters span the range $\Omega_\mathrm{m} \in [0.1, 0.5]$, $\Omega_\mathrm{b} \in [0.03, 0.07]$, $h \in [0.5, 0.9]$, $n_s \in [0.8, 1.2]$ and $\sigma_8 \in [0.6, 1.0]$. In these simulations, not only the set of values of the cosmological parameters is different, but the initial random seed varies as well. Furthermore, we test the accuracy of our network on models with massive neutrinos and on models where the dark energy equation of state is $w \neq -1$, making

use of Quijote simulations labeled M_ν^+, M_ν^{++}, M_ν^{+++}, w^+, and w^{-1}. On average, each of the Quijote simulations used in this work required ~ 500 CPU hours to run. We refer the reader to [51] for further details on the Quijote simulations.

4.3.1.2 Approximate N-body simulations

The fast and approximate simulations we use in this work are run with the COmoving Lagrangian Acceleration (COLA) [84] method, that combines second-order Lagrangian perturbation theory (2LPT) [101] on large scales with N-body methods on small scales. In particular, we use the MG-PICOLA [41] package. For each Quijote full N-body simulation, we run a COLA simulation by matching 1) the number of particles, 2) the set of values of the cosmological parameters, and 3) the value of the initial random seed, which gives rise to identical initial Gaussian field for both Quijote and COLA. These simulations require fewer time steps than the full N-body simulations and are therefore much more computationally efficient. Each COLA simulation is run with 30 time steps equally spaced in log from $z = 9$ down to $z = 0$. On average, these simulations only take 3 CPU hours to run.

4.3.2 Model

4.3.2.1 Input and Target

Let us write the displacement vector of a particle as $\vec{d} = \vec{x}_f - \vec{x}_i$, where $\vec{x}_f$ and $\vec{x}_i$ are the final ($z = 0$) and initial (Lagrangian) position of the particle. Our goal is to train a neural network to correct the positions of the particles generated by COLA, to match them with those from a full N-body simulation, i.e.

$$\vec{x}_{f,\mathrm{Nbody}} = g(\vec{x}_{f,\mathrm{COLA}}) \tag{4.1}$$

where g is an unknown function. Note that the right-hand side of Eq. 4.1 should not be taken as the position of the particular particle considered, but also of all its neighboring particles. To preserve translational equivariance, we use displacement vectors instead of absolute particle positions. Thus, the input to the network is $\vec{d}_{\mathrm{COLA}}$, rather than $\vec{x}_{f,\mathrm{COLA}}$. The network is trained to learn $\vec{d}_{\mathrm{Nbody}} - \vec{d}_{\mathrm{COLA}} = \vec{x}_{f,\mathrm{Nbody}} - \vec{x}_{f,\mathrm{COLA}}$.

4.3.2.2 Model Architecture

We follow Alves de Oliveira et al. [102] and use a V-Net [103] based model that consists of 2 downsampling and 2 upsampling layers connected in a "V" shape. Blocks of two 3^3 convolutions connect the input, the resampling, and the output layers. 1^3 convolutions are added over each of these convolution blocks to realize a residual connection. We add batch normalization after every convolution except the first one and the last two, and leaky ReLU activation with a negative slope of 0.01 after every batch normalization, as well as the first and the second to last convolutions. The last activation in each residual block acts after the summation, following Milletari et al. [103]. As in U-Net/V-Net, at all resolution levels (with the exception of the bottleneck levels), the inputs to the downsampling layers are concatenated to the outputs of the upsampling layers. All layers have a channel size of 64, except for the input and the output, that have 3 channels (the displacement vector along each cartesian coordinate), as well as those after concatenations (128-channeled). Finally, the input ($\vec{d}_{\mathrm{COLA}}$) is directly added to the output, so that the network could learn the corrections to match the target ($\vec{d}_{\mathrm{Nbody}} - \vec{d}_{\mathrm{COLA}}$). Stride-2 2^3 convolutions and stride-12 2^3 transposed convolutions are used in downsampling and upsampling layers, respectively. A diagram of the network architecture is shown in Figure 4.1.

Following Alves de Oliveira et al. [102], we minimize a loss function given by $L =$

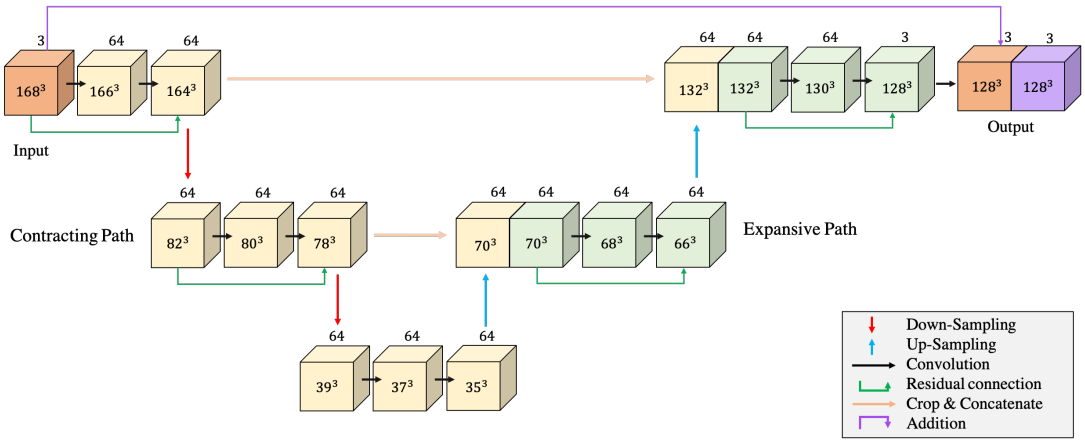

Figure 4.1: The diagram shows the architecture of our model, NECOLA. The top leftmost cube (orange) represents the input and the top rightmost cubes (orange and purple) represent the output. The cubes in yellow and green represent various multi-channel feature maps. The number inside each cube represents the size of the feature map while the number on the top of each cube represents the number of channels in the map. See section 4.3.2.2 for more details on the convolution operations. Figure taken from [6] under a CC BY license.

$\log_e(L_\delta L_\Delta^\lambda)$, where L_δ is the Mean Squared Error (MSE) loss on $n(\mathbf{x})$ (the particle number in voxel $\mathbf{x}$) and L_Δ is the MSE on the displacement vector $\vec{d}$. With this loss function, we are able to train the model to make accurate predictions in both Lagrangian and Eulerian spaces. By combining the two losses with logarithm rather than summation, we can account for their absolute magnitudes and trade between their relative values. λ here serves as a weight on this trade-off of relative losses and $\lambda = 1$ works pretty well in our case.

The input cannot be fed into the network at once due to the big size of the data (3×512^3), and we thus divide it into smaller chunks first. We crop the data into subcubes of size 3×128^3, corresponding to a simulation box of length 250 h^{-1}Mpc.

In order to preserve the physical translational equivariance, no padding has been used in the 3^3 convolutions, which results in an output that is smaller than the input in spatial size. This limitation is compensated by padding the input cubes periodically with 20 voxels on each side so that the effective spatial size of the input becomes 3×168^3. Furthermore, data augmentation is implemented to enforce the equivariance of displacement fields under rotational and parity transformations. We use the Adam optimizer [104] with a learning rate of 0.0001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and reduce the learning rate by half when the loss does not improve for 3 epochs. The model is trained on 70 realizations for 100 epochs and the remaining realizations are used for validation (20) and final testing (10). From now on, we will refer to this model as NECOLA, from Neural Enhanced COLA, in order to avoid any confusion with the model by Alves de Oliveira et al. [102], which uses Zel'dovich simulations as input and a different value of λ. Note that the model architecture of NECOLA is the same as that of Alves de Oliveira et al. [102].

4.3.3 Benchmark models

In order to compare the predictions of our model, we have used four different benchmarks:

† **COLA**. This benchmark represents the results of running the COLA simulation

itself.

† **ZA**. In this case, the positions of the particles at $z = 0$ are computed using the Zel'dovich approximation.

† **mod(ZA)**. This benchmark represents our model but trained on ZA simulations as input and correcting the output to match the target N-body simulations.

† **NN(ZA)**. This benchmark is the model developed by Alves de Oliveira et al. [100], that takes as input ZA simulations and corrects the output to match full N-body simulations. We refer the reader to Alves de Oliveira et al. [100] for further details on this model.

4.4 Results

In this section, we investigate the performance of our model. We first make use of several summary statistics to quantify the accuracy of our model for simulations with the same cosmology as the one used to train the network. Then, we investigate how well does our network extrapolate to other cosmological models.

4.4.1 Fiducial cosmology

We first present the results of testing the network on simulations that have the same cosmology as the one used for its training.

4.4.1.1 Visual comparison

Before quantifying the accuracy of the network using summary statistics, we perform a visual inspection of its output. In Fig. 4.2, we show the distribution of matter at $z = 0$ from the full N-body simulation (top row), the COLA simulation (middle row), and NECOLA (bottom row).

While looking at large scales, the agreement between the three methods is really good, but when we look at small scales, some differences are visible. In the case of COLA, the output is more diffuse and halos do not exhibit a high concentration in their centers, in contrast to the corresponding N-body simulation. On the other hand, NECOLA produces much sharper results, clearly defining the positions and boundaries of dark matter halos.

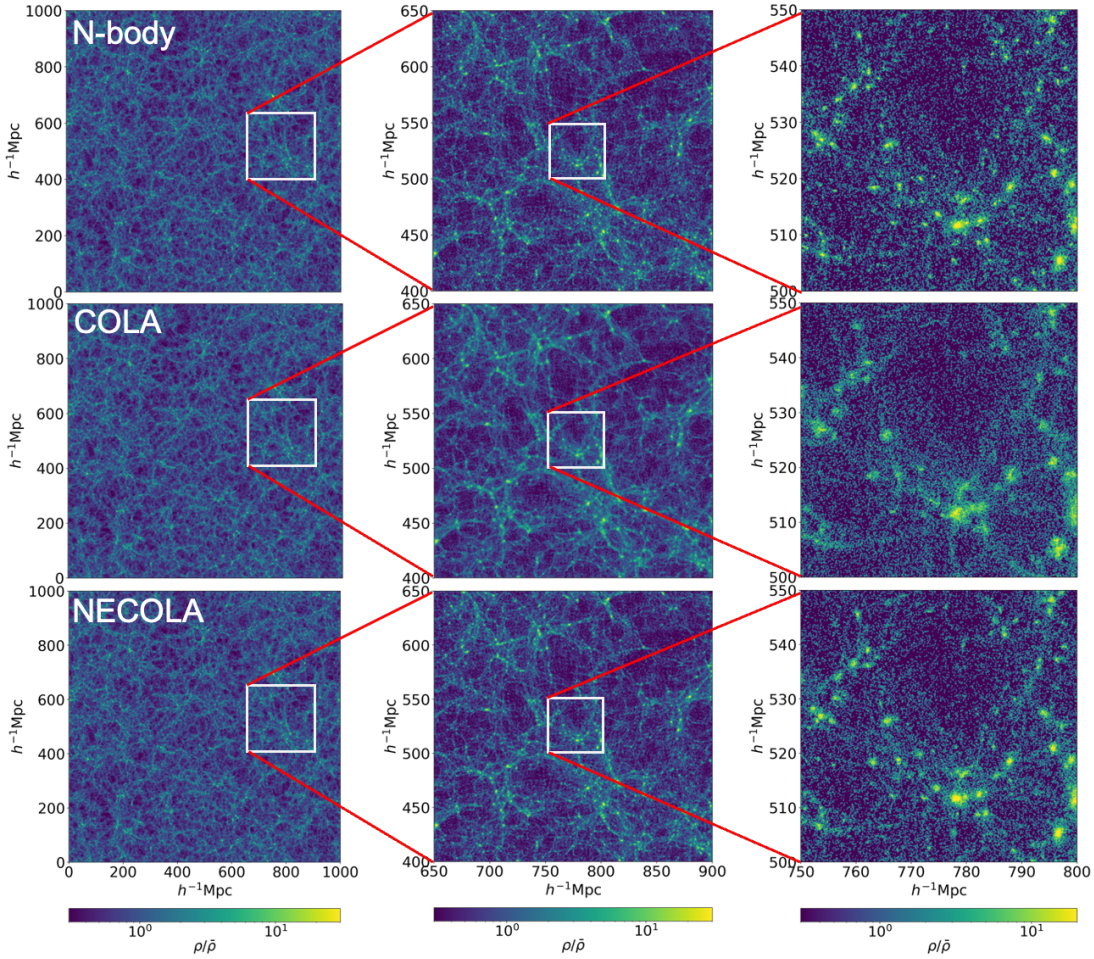

Figure 4.2: The figure shows the cold dark matter density fields for the target N-body simulations (top), the input/benchmark COLA simulations (middle) and the predictions of our model (bottom), at a scale of 1000 Mpc h^{-1} (left column), 250 Mpc h^{-1} (middle column) and 50 Mpc h^{-1} (right column). Each figure is a zoomed-in image of the white box in the figure on its left. Figure taken from [6] under a CC BY license.

4.4.1.2 Power spectrum

The power spectrum is defined as the Fourier transform of the 2-point correlation function (2PCF), which measures the excess probability of finding a pair of random

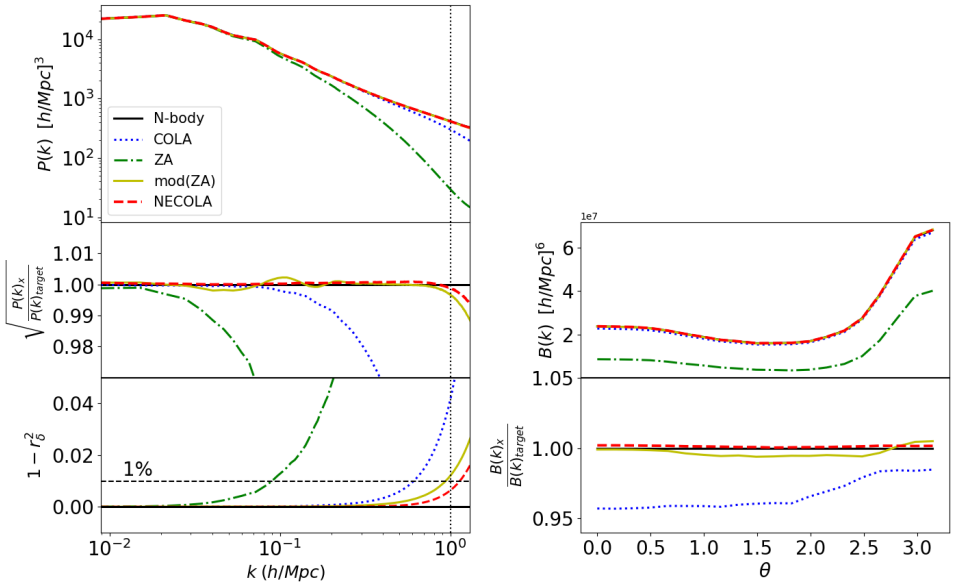

Figure 4.3: The left plot shows the 3D matter power spectrum (top), the transfer function (middle) and the cross-correlation coefficient (bottom), while the right plot shows the bispectrum for $k_1 = 0.15\ h\text{Mpc}^{-1}$ and $k_2 = 0.25\ h\text{Mpc}^{-1}$ (top) and the bispectrum ratio (bottom) for the target N-body simulations (solid black), the COLA simulations (dotted blue), the ZA approximations (dash-dotted green), mod(ZA) (solid yellow), and NECOLA (dashed red). As can be seen, NECOLA outperforms all benchmarks in all cases. Figure taken from [6] under a CC BY license.

galaxies (or points) at a given separation compared to the one from a random distribution. The power spectrum is one of the most important summary statistics used in cosmology since for Gaussian density fields (like the one our Universe resembles on large, linear scales), it fully characterizes the statistical properties of the field.

In the top-left panel of Fig. 4.3, we show with a solid black line the average power spectra from 10 Quijote simulations of the test set. The dotted blue line shows the average power spectrum from the corresponding COLA simulations, while the green

dot-dashed line outputs the average power spectrum of Zel'dovich-evolved simulations. The solid yellow and dashed red lines show the average power spectrum from mod(ZA) and NECOLA, respectively. As can be seen, the worst model is the one that only employs the Zel'dovich approximation, followed by the COLA simulation.

In order to better visualize the differences between the output of the N-body simulation and the networks, we plot in the middle-left panel of Fig. 4.3 the transfer function, defined as

$$T(k) = \sqrt{\frac{P_{\rm pred}(k)}{P_{\rm target}(k)}}, \tag{4.2}$$

where $P_{\rm pred}(k)$ and $P_{\rm target}(k)$ are the average matter power spectra of the predictions and the target density fields respectively. Values close to 1 indicate a better agreement between the prediction and the target. As can be seen, both networks achieve a sub-percent accuracy on the power spectrum down to $k = 1\ h{\rm Mpc}^{-1}$, though the results obtained from NECOLA are slightly more accurate. We note that in the case of the Quijote simulations, it does not make sense to look into much smaller scales than $k \sim 1\ h{\rm Mpc}^{-1}$, as those are not numerically converged in the simulations due to mass resolution [51].

We note that there exist state-of-the-art power spectrum emulators such as COSMIC EMU [105–107], FRANKEN EMU[108] and MIRA TITAN [109, 110] that are computationally faster and more accurate in estimating the power spectrum but are not used in our comparisons as the primary objective of our work is to provide a field-level

emulator itself and not a power spectrum emulator.

4.4.1.3 Cross-Correlation Coefficient

In Fourier space, every mode can be written as $\delta(\vec{k}) = Ae^{i\theta}$, where A and θ are the mode amplitude and phase, respectively. When using the power spectrum, we are effectively comparing how well the amplitude of the modes from the network and the simulation agree. However, that statistic neglects the correlations in mode phases, which are very important in the non-linear regime. To quantify the correlations between the mode phases, we use the cross-correlation coefficient, r, defined as

$$r(k) = \frac{P_{\rm pred\times target}(k)}{\sqrt{P_{\rm pred}(k)P_{\rm target}(k)}}, \tag{4.3}$$

where the numerator is the cross-power spectrum between the predictions and the target and the denominator contains the auto-power spectrum of the prediction and the target. Values of r close to 1 indicate a very good correlation in mode phases. In the bottom-left panel of Fig. 4.3, we show the cross-correlation coefficient averaged over the testing set for the different cases considered. We find that NECOLA achieves the highest accuracy, being within 1% down to $k = 1\ h{\rm Mpc}^{-1}$.

4.4.1.4 Bispectrum

The third statistic that we consider to quantify the agreement between the full simulations and the network predictions is the bispectrum, defined as

$$\langle \delta_{\mathbf{k_1}} \delta_{\mathbf{k_2}} \delta_{\mathbf{k_3}} \rangle \equiv \delta_D(\mathbf{k_{123}}) B(\mathbf{k_1}, \mathbf{k_2}, \mathbf{k_3}), \tag{4.4}$$

where $\delta(\mathbf{k})$ the overdensity in the Fourier space and $\mathbf{k_{123}} \equiv \mathbf{k}_1 + \mathbf{k}_2 + \mathbf{k}_3$.

Differently to the power spectrum, the bispectrum quantifies the correlation between triplets of modes in closed triangles. For Gaussian density fields, this quantity is zero, and therefore, its amplitude and shape capture information about the non-Gaussianities in a given field. In the top-right panel of Fig. 4.3, we show the bispectrum for $k_1 = 0.15\ h\mathrm{Mpc}^{-1}$ and $k_2 = 0.25\ h\mathrm{Mpc}^{-1}$ as a function of the angle between k_1 and k_2, θ. On this scale, we cannot see large differences, besides the fact that the Zel'dovich approximation underestimates the amplitude of the bispectrum, as expected. In the bottom-right panel of Fig. 4.3, we show the ratio between the different bispectra to the bispectrum of the N-body simulation. We find that both neural networks give very accurate results, although NECOLA is slightly more accurate.

The above values of k_1 and k_2 are chosen in order to probe the nonlinear scales of the Universe at which the non-Gaussian signatures in the mass distribution (induced by

non-linear gravitational instability) are imprinted. The model has been evaluated at other values of k_1 and k_2 as well, and performs equally well.

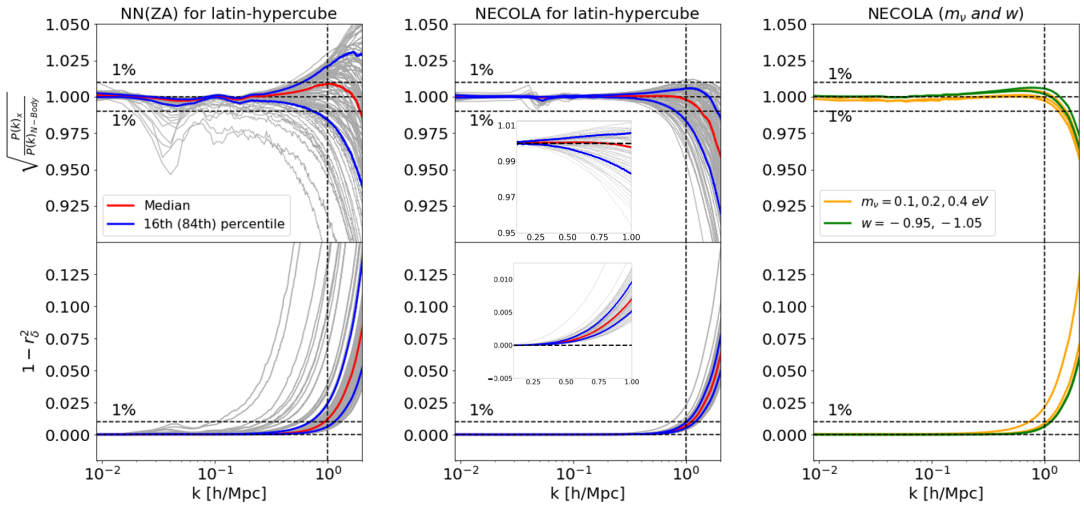

Figure 4.4: We test the NN(ZA) and NECOLA models, which are trained on simulations with a fixed cosmology, on models with very different values of the cosmological parameters. The left and middle panels show the results when using 100 simulations of the Quijote latin-hypercube (that vary $\Omega_{\rm m}$, $\Omega_{\rm b}$, h, n_s, and σ_8), while the right panel displays the results for cosmologies with massive neutrinos and a dark energy equation of state different to -1. The red lines represent the median while the blue lines represent the 16th (and 84th) percentile of the predictions. As can be seen, NECOLA not only performs better than NN(ZA), but it is surprisingly accurate all the way down to $k \sim 1$ hMpc^{-1}. Besides, it also works for models with massive neutrinos and $w \neq -1$. The curve with the largest difference in the neutrino cross-correlation coefficient corresponds to a model with $M_\nu = 0.4$ eV. Figure taken from [6] under a CC BY license.

4.4.2 Model Extrapolation

We now explore how our model extrapolates to cosmologies different from the one used to train the model.

We first test the extrapolation properties of the model on the parameters $\Omega_{\rm m}$, $\Omega_{\rm b}$, h, n_s, and σ_8 by using 100 simulations of the Quijote latin-hypercube set. We emphasize that for the simulations in this set, the values of these 5 cosmological parameters are varied at the same time, together with the value of the initial random seed. For each of these simulations, we run its COLA counterpart and input it to the network, which corrects the positions of the particles.

For each cosmology, we compute the power spectrum of the output of NECOLA and of the full N-body Quijote simulation. In Fig. 4.4, we show in the middle panel the transfer function together with the cross-correlation coefficient. As can be seen, NECOLA is able to correct the output of the COLA simulations in all cases with surprising accuracy: below $\simeq 1\%$ down to $k = 1\ h{\rm Mpc}^{-1}$.

Next, we repeat the same exercise but using NN(ZA) and show the results in the left panel of Fig. 4.4. As can be seen, the network trained on COLA snapshots exhibits much stronger extrapolation features than the one trained on Zel'dovich displacements.

We now investigate if NECOLA is also able to correct COLA outputs for simulations with massive neutrinos. We emphasize that no simulations used for training the model contain massive neutrinos. For this, we made use of simulations from the M_ν^+, M_ν^{++}, and M_ν^{+++} Quijote sets, corresponding to cosmologies with sums of the neutrino masses equal to 0.1 eV, 0.2 eV, and 0.4 eV. In these simulations, we have

both dark matter and neutrino particles. From each set, we take 10 simulations and run their COLA counterpart. Next, we input to NECOLA the displacement vectors of the dark matter particles of the COLA simulation, and NECOLA outputs the corrected positions of the dark matter particles for these massive neutrino models.

Table 4.1
Computational cost associated to running a full N-body simulation, a COLA simulation, NN(ZA) and NECOLA. Note that in case of NECOLA and NN(ZA), we report the GPU wall time.

Simulation	**N-body (QUIJOTE)**	**Fast (COLA)**	**NECOLA (PyTorch-GPU)**	**NN (ZA)**
CPU-/GPU-sec	10^6	10^4	125	59

In the right panel of Fig. 4.4, we show the results of this calculation with yellow lines. As can be seen, NECOLA is able to correct the positions of the dark matter particles such that their power spectrum and cross-correlation coefficient agree with the full N-body calculation below 1% down to $k = 1\ h\mathrm{Mpc}^{-1}$. We note that although our network only works with the cold dark matter field, assuming a linear neutrino field correlated with the initial Gaussian field will, for most of the cases, give very accurate predictions for the total matter field [111]. On the other hand, the cold dark matter field is the one responsible for the abundance and clustering of dark matter halos and galaxies [112, 113]. In Giusarma et al. [114], the authors proposed a deep learning-based Convolutional Neural Network (U-Net) model to generate simulations with massive neutrinos from standard ΛCDM simulations without neutrinos. Their model was able to reproduce the 3-dimensional spatial distribution of matter upto

scales of 0.7 hMpc^{-1} (see Figure 4 of Giusarma et al. [114]), thus emulating the effect of massive neutrinos on the large-scale structure. It is interesting to note that NECOLA gives more accurate results than the model by Giusarma et al. [114] at all scales, capturing the effects of non-linear evolution.

Lastly, we study the performance of our model for cosmologies with values of the dark energy equation of state, w, different from -1. For this, we made use of the 10 simulations of the w^+ and w^- Quijote sets, that have a value of w equal to-0.95 and -1.05, respectively. For each of these simulations, we run their COLA counterpart and compute the displacement vectors. We then input those into the network that returns the corrected positions of the dark matter particles. In the right panel of Fig. 4.4, we show with green lines the results of computing the transfer function and cross-correlation coefficient between the output of the network and the full N-body simulations. As can be seen, in this case as well, NECOLA is able to correct the output of the cosmologies that it has never seen before.

4.4.3 Computational cost

A typical N-body simulation takes roughly 500 CPU hours to run, or $\sim 10^6$ CPU seconds, while a single COLA simulation takes around 3 CPU hours or $\sim 10^4$ CPU seconds. We run our CNN model on 1 GPU (320 NVIDIA P100-16GB) using PyTorch

[115] and it takes $\sim$ 125 GPU seconds to run. A runtime comparison of the target, benchmark, and our model is shown in Table 4.1. Thus, in practice, the main limitation of our model comes from the computational cost associated with running COLA simulations itself. Despite this, our model allows us to speed up the computational cost by a factor of 100.

4.5 Summary

Providing accurate theoretical predictions is necessary in order to extract the maximum amount of information from upcoming cosmological surveys. The computational cost of running full N-body simulations is currently too expensive to carry out standard analysis such as MCMC. On the other hand, fast simulations can reduce the computational cost by orders of magnitude at the expense of sacrificing accuracy.

In this work, we have shown that we can use neural networks to correct the output of approximate simulations to match full N-body simulations from the Quijote suite. Our model, coined NECOLA, from Neural Enhanced COLA, has been trained on simulations with a fixed value of the cosmological parameters. We have shown that our model is not only able to correct the output of COLA simulations run with the same cosmology as the one used to train the network, but is also able to correct COLA simulations that have very different values of the parameters $\Omega_{\rm m}$, $\Omega_{\rm b}$, h, n_s,

σ_8, M_ν, and w. This surprising feature of our network indicates that the correction from the output of COLA to a full N-body might be universal, i.e. independent of cosmology.

This may have important consequences for perturbation theory studies, that are able to accurately model the linear and perturbative regime but fail on non-linear scales. Our work indicates that a generic, cosmology-independent correction may be feasible, at least in the case of the power spectrum.

Our network can be used as a field-level emulator for covariance estimation, likelihood-free analysis, detecting features in the cosmic web and to explore not only the initial modes amplitudes and phases [116], but also the cosmological parameter space [117].

We note however that further work is needed to claim that our model is precise for statistics other than the power spectrum and cross-correlation function when using it in extrapolation. In future work, we will quantify the accuracy of our network on other summary statistics like bispectrum, halo mass function, etc. Additional work is also needed to incorporate velocities into this framework, that will allow performing studies in redshift-space. Besides, further work is needed to quantify the accuracy of NECOLA at redshifts other than the one used for training, together with the universality of the network under changes of simulation resolution.

Overall, this work opens an interesting direction in the development of fast and generalized field-level emulators needed to maximize the scientific return of upcoming cosmological missions.

The trained models, predictions and statistics extracted from the testing and extrapolation sets are hosted under the public github repository `https://github.com/neeravkaushal/cola-to-nbody.git` and the model has been trained using the map2map code `https://github.com/eelregit/map2map.git`.

4.6 Acknowledgments

The Quijote simulations used in this work are publicly available at `https://github.com/franciscovillaescusa/Quijote-simulations.git`. The COLA simulations have been run using the MG-PICOLA code, publicly available at `https://github.com/HAWinther/MG-PICOLA-PUBLIC.git`. We acknowledge that our work has been performed using the Princeton Research Computing resources at Princeton University which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE). E.G. thanks Michigan Space Grant Consortium for their support. The work of FVN has been supported by the WFIRST program through NNG26PJ30C and NNN12AA01C. FVN and YL are supported by the Simons Foundation. Research reported in this publication was supported in part by

funding provided by the National Aeronautics and Space Administration (NASA), under award number NNX15AJ20H, Michigan Space Grant Consortium (MSGC).

Chapter 5

νGAN: Conditional GAN-based Emulator for Cosmic Web Simulations with Neutrinos

In this Chapter, we develop a Generative Adversarial Network (GAN) that generates two dimensional cosmic webs conditioned on a range of neutrino masses. Our model could generate cosmic webs from a small latent vector space using CNN-based upsampling. We test the accuracy and quality of our generated cosmic webs using several summary statistics and present our preliminary results in this chapter.

5.1 Abstract

The presence of relic neutrinos significantly affects the evolution of density perturbations in the universe. Due to their large free streaming lengths, neutrinos suppress structure formation at small scales. In order to compare the information extracted from large-scale cosmological surveys, rigorous theoretical predictions of the effect of neutrinos on the evolution of universe are needed. A significant tool to obtain these predictions is running dark matter N-body cosmological simulations of the universe in the presence of massive neutrinos which is very computationally expensive. It takes around 700 CPU hours to run a single N-body simulation with a specific massive neutrino [118]. In this work, we propose a deep-learning based generative adversarial network (GAN) model that could emulate our universe with a specified neutrino mass. Our model νGAN generates 2D cosmic webs conditioned on neutrino masses in the range 0.0 eV to 0.8 eV. The generated samples are statistically independent, uncorrelated and indistinguishable from the actual samples. We compare the accuracy of our results visually and on various summary statistics prominent in cosmology. Preliminary results indicate that the generated samples are accurate to within 5% on power spectrum between $k = 0.01$ to $k = 0.5$. This opens up a new avenue for research on a universal cosmology-injected emulators that could be conditioned on a number of cosmological parameters to provide reliable, accurate and fast emulators as substitutes for traditional simulations.

5.2 Introduction

Neutrinos are one of the most abundant particles in the universe with number densities somewhat less than photons. They were relativistic at early times and behaved as radiation. At the present time, unlike photons, neutrinos are non-relativistic and are known to have rest mass. This tells that the relic neutrinos can produce significant effects on the cosmological observables, particularly on the low-redshift evolution of cosmological density perturbations. Neutrinos have large thermal velocities unlike other gravitating massive species like cold dark matter or baryons, and thus neutrinos leave distinctive signatures in many cosmological observables. They strongly affect the background evolution of the universe, as well as the evolution of cosmological perturbations. Current and future state-of-the-art cosmological surveys such as Euclid [119], DESI [120], WFIRST [121], LSST [122], and CMB-S4 [123], are expected to improve the constraints on the cosmological parameters and provide information that could reduce the error in the constraints on neutrino masses and their mass ordering.

In order to compare the results from the measurement of these cosmological variables, rigorous theoretical predictions of the spatial distribution of matter and luminous tracers in the presence of massive neutrinos are required. Analytic tools like perturbation theory [50] can be used to provide such predictions at semi-linear scales. Most of the information, however, resides on nonlinear scales [e.g. 51–70]. In the absence

of an analytical model in cosmology, numerical simulations with massive neutrinos provide the most powerful tool to study these nonlinear scales and compare against the observations. Particularly, the N-body simulations evolve cosmological matter fluctuations under gravity alone, allowing to compare the theory with predictions and generating mock galaxy catalogs, compute covariance matrices, and optimize observational strategies. Various N-body simulations with massive neutrinos have been developed and used over the past couple of years. They have helped in studying the impact of neutrino masses on clustering in fully nonlinear scale in real space [112, 124–127], on clustering and abundance of halo and cosmic voids [113, 128, 129], and on clustering of matter in real-space [130, 131]. The major drawback of cosmological N-body simulations is that they are very computationally expensive to run. A single N-body simulation requires large computational resources and a runtime in days or even weeks. This computational bottleneck limits the amount of information we can extract from observational data and check against the theory. Faster methods of generating cosmological simulations are thus needed that could accelerate this process while maintaining the accuracy and reliability of the predictions.

Over the past decade, deep learning has come out to be one of the most efficient and reliable tool to either generate cosmological simulations or map from less accurate simulations to more accurate ones, or to generate superresolution realizations of the full phase-space matter distribution of the universe from low resolution N-body simulations [6, 97–100, 132]. More recently, a class of deep learning neural networks

called the generative adversarial networks (GANs) have been extensively used to generate various kinds of cosmological maps of the universe like dark matter cosmic webs [132], weak lensing convergence maps [133], and non-tomographic sky convergence maps [134].

In this work, we use deep convolutional GANs to generate 2D cosmic webs of the universe, conditioned on a range of neutrino masses. Deep generative models can effectively learn the complex probability distributions of the data and can generate new, random and statistically independent and identically distributed data samples after training on a set of N-body simulations. These new data samples are uncorrelated to the training examples. We condition our model, called νGAN, on neutrino masses so that after training, we can generate dark matter cosmic webs with arbitrary neutrino masses. After training the network, numerous new samples can be generated within a matter of seconds. Conditioning on neutrino masses, on the other hand, provides overcoming the computational bottleneck of generating variable mass neutrino simulations using traditional methods. We test our results on various summary statistics viz the power spectrum, the transfer function, the pixel intensity histograms, peak statistics and structural similarity tests. We find that our model is accurate to 5% on the power spectrum between $k = 0.01$ to $k = 0.5$.

This paper is organized as follows. Section 5.3 briefly introduces and discusses conditional GAN and the data we used for training our GAN model. Section 5.4 discusses

the training process, model architecture and hyperparameters while section 5.5 discusses the results obtained. Finally, we draw conclusions and other discussions in section 5.6.

5.3 Methods

5.3.1 Conditional GAN

A conditional Generative Adversarial Network, or CGAN [48] for short, is a type of deep learning approach to conditional generative modeling using neural networks like convolutional neural nets (CNNs) or densely connected (linear) layers. The network consists of a generator and a discriminator. The goal of the generator is to create fake data as close to the real data as possible while the goal of the discriminator is to classify the real data as real and fake data as fake. Both the networks work in a game-theoretic approach and achieve convergence through training. Once trained, the generator can create new samples that are indistinguishable to the real samples (See 3.6 or more details).

The model we use in this work conditions the generator (G) and the discriminator (D) both on a parameter y, which in this work would be the neutrino mass.

The algorithm works as follows:

† The input is a random noise vector (z) that can be Gaussian distributed (usually drawn from a unit-normal distribution $N(0, 1)$), uniform distributed or can be some other structured input.

† The generator G, takes the latent vector z and the random variable (the condition) y and parameterized by a neural network, gives the output $G(z, y)$.

† The discriminator D, also parameterized by a neural network, inputs real samples x and fake samples $G(z, y)$ i.e., the samples generated by G, and outputs scores $D(x)$ and $D(G(z, y))$ respectively. Each score represents the belief of discriminator in the sample being real i.e. coming from the distribution of the real data, $p_{data}(x)$. This score, when scaled to $[0, 1]$ can also be loosely interpreted as an implicit likelihood of the data given D i.e. $p(x|D)$.

† The predictions of D are compared to the actual, true labels and a loss is computed ($L(D, G)$).

† This loss is backpropagated, first through D and then through G to update the weights and biases of both.

† Steps 1-5 are repeated over several epochs while iterating through the entire dataset.

For more details on a conditional GAN, see section 3.6.1.

5.3.2 Data

The various steps involved in the generation and preprocessing of data are as follows:

† We run N-body simulations using COLA approximation [84, 85] with MG-PICOLA[1][41] code. The simulations follow the evolution of 1024^3 cold dark matter (CDM) particles in the presence of neutrinos with masses 0.0, 0.1, 0.4, and 0.8 eV from a redshift of $z = 9$ to $z = 0$ in 50 timesteps. The cosmological parameters used for these simulations are $\Omega_M = 0.3175$, $\Omega_B = 0.0490$, $n_s = 0.9624$, $\sigma_8 = 0.8340$, $H = 67.11$, and $m_\nu = (0.0, 0.1, 0.4, 0.8)$. We generate 2 realizations for each neutrino mass and thus 10 realizations in total for all the 5 neutrino masses. Each simulation gives the 3D spatial coordinates of 1024^3 cold dark matter particles.

† Following [132], for each realization, we take this 3D cube of particle positions and divide the positions along x-axis into 1000 equal segments. We then extract 2D slices of particle positions in the y-z plane and choose 500 non-consecutive 2D slices of position coordinates. We repeat the same procedure along the y and the z axes and thus obtain 1500 2D slices for one realization and thus 15000

[1]https://github.com/HAWinther/MG-PICOLA-PUBLIC

for the entire data (10 realizations).

† These 2D slices are then pixelized into 256×256 slices. The value at each pixel here corresponds to its particle count. This effectively makes the data 2D grayscale images of size 256×256. These images are then smoothed with a gaussian filter of standard deviation 1. The only difference is that instead of the pixel values being integers, they are now floating point numbers with a huge range of magnitudes.

† The data is then scaled to $[-1, 1]$ as this scaling has been found to be very effective in improving the performance of the model [132]. The scaling also allows the final activation of the generator to be tanh. The original data (ρ) and the scaled data ($\rho(x)$) are related by the transformation:

$$\rho(x) = \frac{2x}{x + a} - 1, \tag{5.1}$$

where a was chosen to be 4.

This transformation is very similar to the logarithmic function. As the cosmic web of the universe spans a dynamic range of magnitudes between the almost empty cosmic voids and the super-massive galaxy clusters, this transformation enhances the contrast on the network of filaments, galaxy sheets and dark matter halos. a in 5.1 controls the median value of the images and has been fixed to 4 throughout the training of

the network.

5.4 Implementation

We use a Wasserstein deep convolutional generative adversarial network (WGAN) [135] instead of a standard DCGAN. While the discriminator of a conventional GAN outputs a probability value that denotes its confidence in the degree of realness of the sample, the discriminator of a WGAN assigns a score to each sample based on the distance between the real and fake distribution. This distance, which is a measure of the distance between two probability distributions, is called the Wasserstein or Earth Mover's distance (see [136] for a nice review on the difference between GANs and WGANs). The discriminator in case of a WGAN is called a critic.

Like conventional GANs, our WGAN model, νGAN, consists of a generator and a discriminator. The generator consists of one linear layer and six transposed convolutions. The neutrino mass m_ν is concatenated to the latent vector z of size 200 at the start and operated on by a linear layer. This is followed by upsampling by six transposed convolutions of filter sizes 5 and 3 with strides 2 and 1 respectively. Each upsampling operation is followed by a batch normalization and a relu activation, except for the final layer where we use a tanh activation and no batch normalization. Tables 5.1 shows the architecture of the generator.

Table 5.1
Generator architecture of our model.

Layer	Operations	Filter	Dimension
z			bs $\times 200$
$h0$	linear + identity		bs $\times 512$ $\times 16 \times 16$
$h1$	deconv + BatchNorm + ReLU	5×5	bs $\times 256$ $\times 32 \times 32$
$h2$	deconv + BatchNorm + ReLU	5×5	bs $\times 128$ $\times 64 \times 64$
$h3$	deconv + BatchNorm + ReLU	3×3	bs $\times 128$ $\times 64 \times 64$
$h4$	deconv + BatchNorm + ReLU	5×5	bs $\times 64$ $\times 128 \times 128$
$h5$	deconv + BatchNorm + ReLU	3×3	bs $\times 64$ $\times 128 \times 128$
$h6$	deconv + Tanh	5×5	bs $\times 1$ $\times 256 \times 256$

The discriminator consists of one linear layer and four convolutions. The convolutions use a filter of size 5 and stride 2 and double the channels and reduce the feature size by half with each operation. They are followed by batch normalization and a leaky relu activation with parameter 0.2. After the convolutions, the data is squeezed, and the neutrino mass is concatenated to it. Finally, a linear layer is applied which gives the desired output. Table 5.2 shows the discriminator details.

Table 5.2
Discriminator architecture of our model.

Layer	Operations	Filter	Dimension
X			bs $\times 1 \times 256 \times 256$
$h0$	conv + BatchNorm + Leaky ReLU	5×5	bs $\times 64 \times 128 \times 128$
$h1$	conv + BatchNorm + Leaky ReLU	5×5	bs $\times 128 \times 64 \times 64$
$h2$	conv + BatchNorm + Leaky ReLU	5×5	bs $\times 256 \times 32 \times 32$
$h3$	conv + BatchNorm + Leaky ReLU	5×5	bs $\times 512 \times 16 \times 16$
$h4$	linear + identity		bs $\times 1$

WGANs were used to eliminate mode collapse and induce stable training. The networks were trained until convergence was achieved in terms of a stable distance between the generated and real images. The hyperparameters used for training are shown in table 5.3.

Table 5.3
Hyperparameters used for model training.

Hyperparameter	**Value**
Learning rate (G, D)	$(10^{-5}, 10^{-5})$
Batch size	16
Latent vector size	200
Latent vector distribution	Standard Normal
Optimizer	ADAM
Gradient Penalty	1000
β_1, β_2	$0.5, 0.999$
Augmentation	True
Epochs	300

5.5 Results

In this section, we assess the performance of our model in various ways. First, we perform a visual comparison of the synthetic and the actual images. Second, we perform a quantitative assessment of the results using various summary statistics.

5.5.1 Visual Comparison

We compare the results of the N-body samples and νGAN-generated samples visually. Figure 5.1 shows 10 random images of cosmic web from the N-body simulations (top) and 10 random samples generated by νGAN (bottom). It can be seen that νGAN captures the prominent visual artifacts of the data quite well. The structure of filaments and halos are well reproduced. This shows the capability of GANs to reproduce the cosmic web. It should be noted that in this figure, there is no need for the samples in the bottom two panels to be the same as the samples in top two panels.

The parameterization of our model on neutrino masses can be checked by generating the images using the same latent vector space and a different neutrino mass. Figure 5.2 shows the images from N-body simulations (top) and νGAN (bottom) using various neutrino masses. The latent vector in νGAN and the random seed in the simulations are separately fixed, so as to visually assess the images for different neutrino masses under the same fixed conditions. As can be seen, all the images from νGAN in the bottom row of figure 5.2 look the same. The effect of neutrinos on the clustering of matter at small scales is too small to be evident visually from the cosmic web images. Please note that the images in the top row should not be compared to the corresponding images in the bottom row.

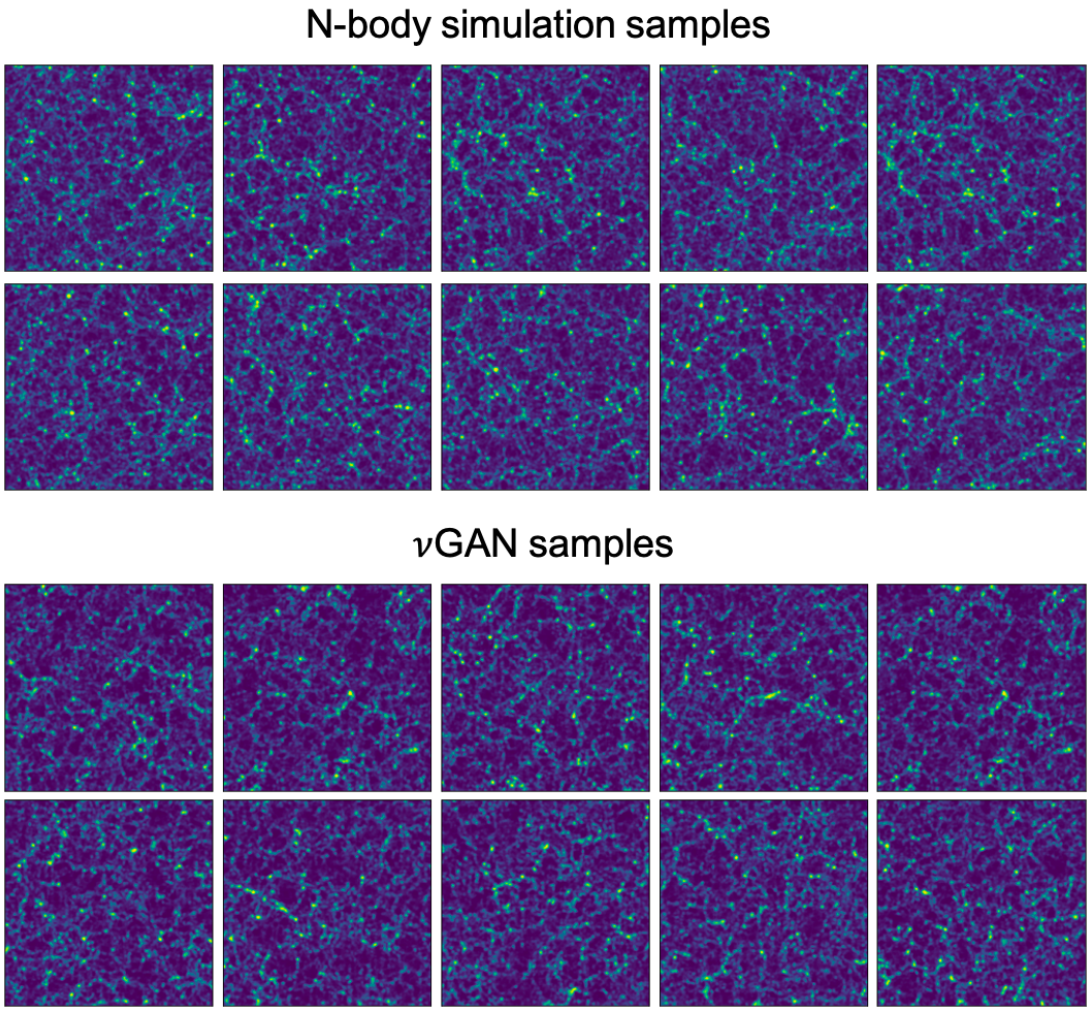

Figure 5.1: The top two panels show 10 cosmic webs from N-body simulations while the images in the bottom two panels are generated by νGAN. Each bright spot in the image denotes the average number of dark matter particles or the density contrast (see 1.1.5) in that pixel location. *Note that the pixel values are scaled to [-1,1] and the top* 10 *images are not to be visually compared to the bottom* 10 *images.*

We now show the comparison of our model and the simulations using the most commonly employed statistics in cosmology called the cosmological summary statistics, and the similarity metrics used in computer vision.

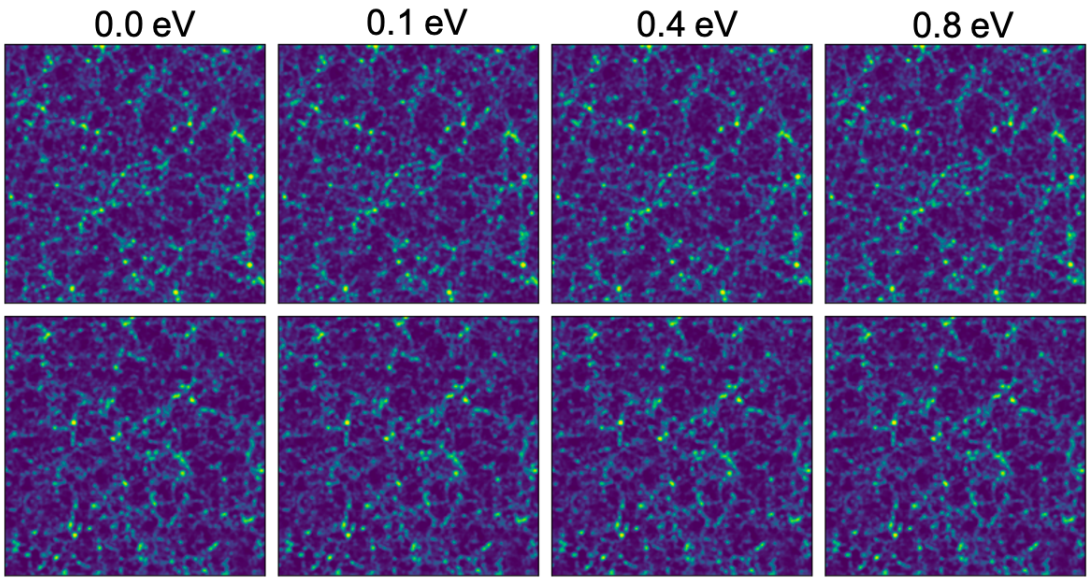

Figure 5.2: The top panel shows the cosmic webs from our simulations while the bottom ones are generated by νGAN. Note that for the top images, the random seed during the simulations was fixed and for the bottom images, the latent vector was fixed. The images in the top row therefore look similar to each other and the same is true for the bottom row. The images in the top row are not comparable to the images in the bottom row.

5.5.2 Power Spectrum and Transfer Function

Figure 5.3 shows the average 2D power spectra and transfer function (see eqn. 4.2 in 4.4.1.2) comparison between the actual N-body simulations (black curve) and the samples from our model (red curve) for various neutrino masses. The upper limit on the neutrino mass from the latest experiments [137] is 0.8 eV, which we use in our analyses. It can be seen that the power spectra are almost overlapping and have very small margin of error, especially at smaller scales. In order to better quantify and visualize the differences between the truth and the predictions in the power spectrum,

we also show the transfer function in the bottom panel. We focus our analysis on angular scales larger than a few Mpc. This is primarily because currently, the N-body simulations do not agree well in their predictions for smaller scales [132, 138]. The model performs really well with roughly a 95% accuracy between $k = 0.01$ and $k = 0.5$ for all neutrino masses. The predictions start to get worse at $k > 0.5$ which induces a noise in transfer function at those scales. This can be attributed to nonlinear processes at small scales of the universe, which makes the predictions worse on smaller scales. Currently, we are working on improving this error margin to much smaller values at nonlinear scales.

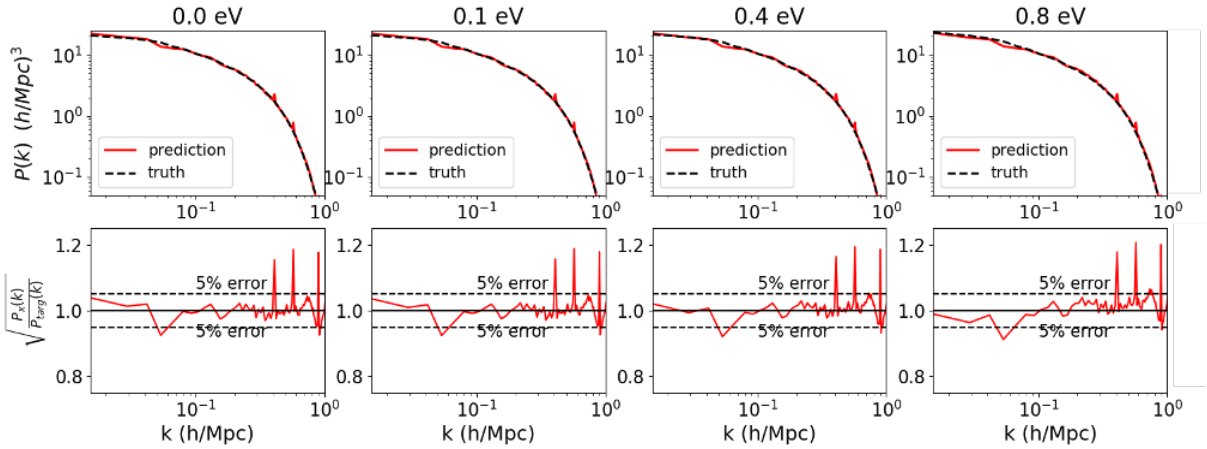

Figure 5.3: Power spectrum and transfer function comparison. The top panel shows the average 2D power spectra of the N-body images (black curves) and the ones generated by νGAN (red curves) for various neutrino masses. The difference in power spectra is very small (within 5%) at linear and mildly nonlinear scales.

5.5.3 Pixel Intensity Histogram

Figure 5.4 shows the distribution of mass map pixels (N_{pixels}) in N-body and νGAN-generated maps. Mass map histogram and peak counts are simple computer vision statistics that compare maps and constrain cosmological models [139, 140]. The pixel intensity histograms in general shows a good agreement with significant differences appearing only for the lowest pixel intensity values. The same has also been detected in the works of [133] and [141]. The small pixel values correspond to the black regions of the image which denote the cosmic voids. The top panel shows the mass histograms and the bottom panel shows the fractional difference in predictions wrt the actual values. The number of pixels for small pixel values (at around -1) are predicted (red curve) to be slightly more than the actual N-body samples (black curve). This might be attributed to the model not learning to reproduce the distribution of structure around the cosmic voids by overestimating the voids.

5.5.4 Pixel Peak Histogram

Although the power spectrum fully characterizes the information embedded in a gaussian field, to extract the information stored in the non-gaussianities in the field, we

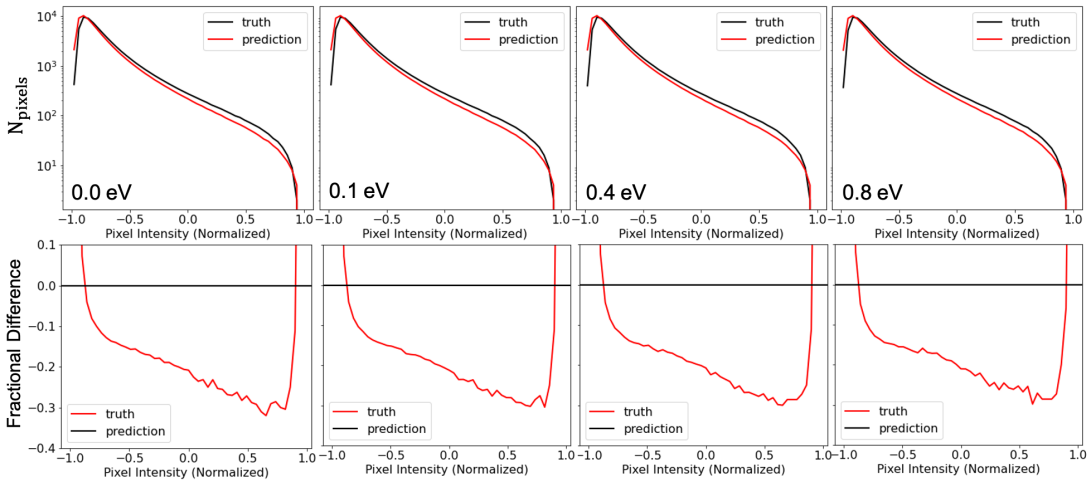

Figure 5.4: A comparison of pixel intensity histogram of the samples generated from the N-body simulations and our model, νGAN. The curves are averaged over 500 samples. The major difference is at lower pixel intensity values.

need higher order statistics (see 1.1.6 for a detailed description). The higher-order statistics are usually very computationally intensive. A popular alternative to power spectrum to analyze the density distribution of the cosmic webs is the "peak statistics". These extract the non-gaussian features present in the data and are commonly employed to analyze weak lensing data [140, 142]. A peak refers to a pixel in the image that has a higher value than all of its immediate 24 neighbors. The peaks are then counted as a function of their height.

We show in figure 5.5 the distribution of mass map peaks (N_{peaks}), which describes the distribution of values at the local maxima of the map. All the pixels greater than their 5×5 patch neighborhood i.e., 24 neighbors are searched and extracted. A histogram of the extracted peak values is then computed. Finally, the median histogram of

500 such images is computed along with their 16% and 84% percentiles from N-body simulations and νGAN. The plots in the top panel show the peak statistics histograms, while the bottom panel plots show the fractional difference in the number of peaks. This shows that the samples from the N-body images and the νGAN images are very close to each other.

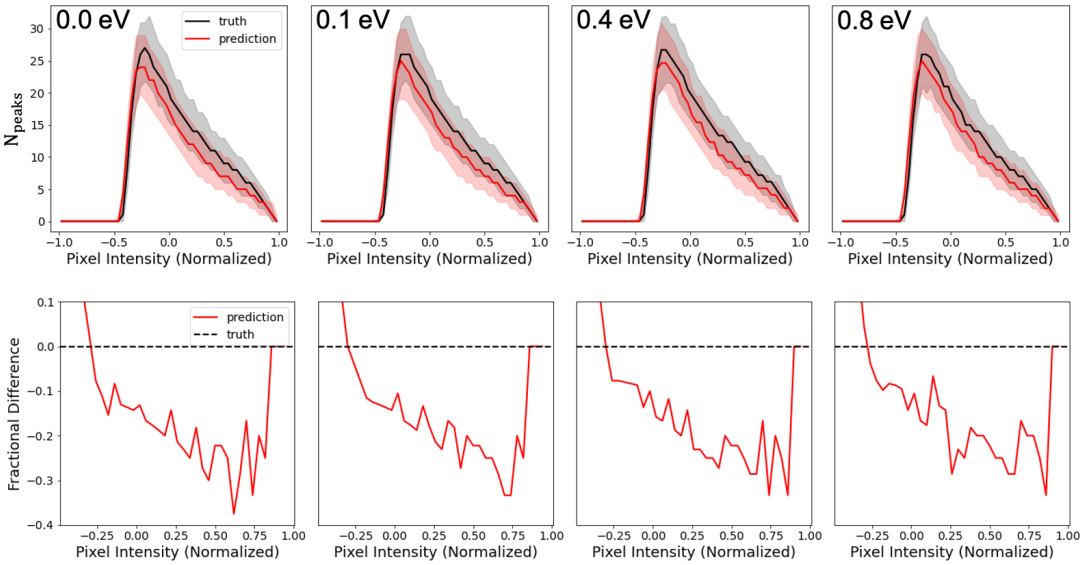

Figure 5.5: A comparison of pixel peaks. The solid lines show the median histogram from 500 samples generated by νGAN and from N-body simulations.The corresponding color shades show the 16%th and 84%th percentile of the distribution. Note that the pixels are scaled to [-1,1].

5.5.5 Multi-Scale Structural Similarity Index (MS-SSIM)

A common problem encountered while working with GANs is that the network is trained in such a poor way that the generator keeps producing a small subset of data

and is not able to learn the variance in the entire distribution of the data. This is known as the problem of **mode collapse**[2]. νGAN gets rid of mode collapse by using a bigger latent vector (z) of size 200. This can be evidenced from the fact that in figure 5.1, each image that νGAN generates is stunningly different from the others, giving a direct evidence that the model does not suffer from mode collapse.

The Multi-scale structural similarity index (MS-SSIM) is a very commonly used image similarity measure used in image analysis studies. It is very useful in detecting whether a model suffers from mode collapse or not. *One of the main reasons of studying this is that the cosmological summary statistics of the truth and predictions can still agree well with each other, even if the model suffers from mode collapse.*

The MS-SSIM between two images returns a score between 0 and 1 where 0 means identical images and 1 means completely different images. The cosmic web images are stochastic and are only similar statistically, we compute the MS-SSIM between an ensemble of 2000 images from N-body simulations and νGAN. This is because, we are more interested in the similarity between a large set of images than between individual images [134]. Following [134], we calculate the significance of the difference in MS-SSIM scores as follows:

[2]An example of mode collapse would be of a GAN trained on a dataset of handwritten digits from 0 to 9. This network would keep producing only a small subset of digits (say 2, 4 and 9) and will never learn the entire dataset if it suffers from model collapse.

$$s_{\rm SSIM} = 2\frac{\langle {\rm SSIM}^{\nu{\rm GAN}}\rangle - \langle {\rm SSIM}^{\rm N\text{-}body}\rangle}{\sigma[{\rm SSIM}^{\nu{\rm GAN}}] + \sigma[{\rm SSIM}^{\rm N\text{-}body}]} \tag{5.2}$$

where $\langle {\rm SSIM}\rangle$ is the mean score and $\sigma[{\rm SSIM}]$ is the standard deviation. The smaller this score is, the more similar are the images generated by νGAN and N-body simulations. We calculate the mean and standard deviation of MS-SSIM score between 500 randomly selected images for each neutrino mass, both for νGAN and the original N-body. Table 5.4 shows the values of SSIM significance scores between N-body and νGAN-generated samples for various neutrino masses. All the scores are less than 1 and are very strong indicators that νGAN preserves the statistics of the actual N-body samples. The samples generated by νGAN thus agree very well with the N-body images.

Table 5.4
MS-SSIM scores for νGAN for each neutrino mass

Neutrino mass (in eV)	**MS-SSIM score**
0.0	0.58
0.1	0.48
0.4	0.41
0.8	0.43

5.6 Conclusions

In this work, we showed how a class of generative models, called generative adversarial networks, can be used to learn the dark matter cosmic web of the universe in the presence of neutrinos. We developed a deep Wasserstein GAN model to mimic the structure formation in the universe. Our model, νGAN conditions on neutrino masses and generate statistically independent and uncorrelated samples of the 2D cosmic web. Using a bigger latent size, νGAN eliminates the problem of mode collapse, which is verified with the Multi-scale structural similarity score between the generated and real samples. The model converges pretty well and the samples generated by it are visually indistinguishable from the actual N-body samples. We also check the accuracy of νGAN's predictions using various cosmological and computer vision summary statistics. The power spectrum, transfer function, pixel intensity histogram, peak statistics, and the MS-SSIM significance score between νGAN-generated samples and the N-body samples agree really well. This shows that νGAN indeed produces novel data instead of just mimicking the training N-body data, which has also been seen in several past works [48, 132–134, 141, 143]. The most important feature of our approach is that our model can generate new samples of the cosmic web in a fraction of a second on modern GPU-based systems. Compared to a traditional N-body simulation, this is a speedup of orders of magnitude. This directly addresses the limitation of the computational bottleneck in generating new simulations of the universe and

reduces the computational burden on the current resources. In the future, the need to efficiently generate vast N-body simulations will increase due to upcoming large-scale cosmological surveys that will generate vast swaths of observational data. Various kinds of new analysis methods based on deep learning [144] or advanced statistics [145] will aim to extract more information from the cosmological data and GANs will play a very pivotal role in providing theoretical predictions to compare against this data.

5.7 Future Works

In the future, we aim to improve the accuracy and reliability of our model. Currently, the model is not performing very well on small nonlinear scales and the largest scales (due to cosmic variance). We will try to improve the predictions of νGAN on these scales by incorporating a different approach of training, wherein the network will be trained on the residual of the matter maps between the fiducial cosmology (massless neutrinos) and the massive neutrino cosmology. This should, in principle, enable the model to ignore the effects of cosmic variance and learn the residual pixels themselves at all scales. We will also use higher-order statistics such as bispectrum, Minkowski functionals, and cross-correlation functions. Finally, it would also be interesting to explore the number of simulations needed to train νGAN for a given precision requirement.

Chapter 6

The CAMELS project: public data release

6.1 Abstract

The Cosmology and Astrophysics with MachinE Learning Simulations (CAMELS) project was developed to combine cosmology with astrophysics through thousands of cosmological hydrodynamic simulations and machine learning. CAMELS contains 4,233 cosmological simulations, 2,049 N-body and 2,184 state-of-the-art hydrodynamic simulations that sample a vast volume in parameter space. In this paper we present the CAMELS public data release, describing the characteristics of the

CAMELS simulations and a variety of data products generated from them, including halo, subhalo, galaxy, and void catalogues, power spectra, bispectra, Lyman-α spectra, probability distribution functions, halo radial profiles, and X-rays photon lists. We also release over one thousand catalogues that contain billions of galaxies from CAMELS-SAM: a large collection of N-body simulations that have been combined with the Santa Cruz Semi-Analytic Model. We release all the data, comprising more than 350 terabytes and containing 143,922 snapshots, millions of halos, galaxies and summary statistics. We provide further technical details on how to access, download, read, and process the data at `https://camels.readthedocs.io`.

6.2 Introduction

Recent advances in deep learning are triggering a revolution across fields, and cosmology and astrophysics are not left behind. Applications include parameter inference [146–153], superresolution [154–156], generation of mock data [157–159], painting hydrodynamic properties on N-body simulations [100, 160–169], improving the halo-galaxy connection [170–174], removing/cleaning astrophysical effects [175–177], emulating non-linear evolution and speeding up numerical simulations [6, 178–180], learning functions to interpolate among simulation properties [181, 182], estimating masses of dark matter halos [183–187] and galaxy clusters [188–194], finding universal relations in subhalo properties [195], generating realistic galaxy images [196],

model selection and classification [e.g. 197], and improving SED fitting techniques [198, 199], among many others (see Stein 200[1] for a comprehensive compilation). At its core, many of these results are based on using neural networks to approximate complex functions that may live in a high dimensional space. These techniques have the potential to revolutionize the way we do cosmology and astrophysics.

From the cosmological side we have now a well established and accepted model: the Λ cold dark matter (ΛCDM) model. This model not only describes the laws and constituents of our Universe, but it is also capable of explaining a large variety of cosmological observables, from the temperature anisotropies of the cosmic microwave background to the spatial distribution of galaxies at low redshift. The model has free parameters characterizing fundamental properties of the Universe such as its geometry, composition, the properties of dark energy, the sum of the neutrino masses, etc. One of the most important tasks in cosmology is to constrain the values of the these parameters with the highest degree of accuracy. In that way, we may be able to provide answers to fundamental questions such as: "What is the nature of dark energy?" and "What are the masses of the neutrinos?"

Many studies have shown that there is a wealth of cosmological information located on mildly to highly non-linear scales that need summary statistics other than the power spectrum to be retrieved [51–70, 201–203]. Extracting the maximum amount

[1] `https://github.com/georgestein/ml-in-cosmology`

of information from these scales presents two main challenges. First, the optimal summary statistics that fully characterizes non-Gaussian density fields is currently unknown. Second, these scales are expected to be affected by astrophysical effects, such as feedback from supernovae and active galactic nuclei (AGN), in a poorly understood way [e.g. 204, 205]. Due to this uncertainty, cosmological analysis are typically carried out avoiding scales that are affected by astrophysical processes.

On the other hand, the cosmological dependence on astrophysical processes such as the formation and evolution of galaxies is typically neglected. Thus, while intrinsically linked, cosmology and galaxy formation tend to progress in parallel with limited interactions. Building bridges between cosmology and galaxy formation will thus benefit the development of both branches and contribute to an unified understanding.

Unfortunately, the interplay of cosmology and astrophysics takes places on many different scales, including non-linear ones. This implies that cosmological hydrodynamic simulations are among the best tools to model and study the interactions between cosmology and astrophysics. However, given the uncertainties in both cosmology and galaxy formation models, it would be desirable to run simulations for different values of the cosmological parameters and also for different astrophysical models. Finally, if the number of simulations is large enough, one can make use of machine learning techniques to extract the maximum amount of information from the simulations while at the same time being able to develop high-dimensional interpolators to explore the

parameter space without having to run additional simulations.

The Cosmology and Astrophysics with MachinE Learning Simulations (CAMELS) project [206] was conceived to combine cosmology and astrophysics through numerical simulations and machine learning. At its core, CAMELS consists of a set of 4,233 cosmological simulations that have different values of the cosmological parameters and different astrophysical models. All these virtual universes can be used as a large dataset to train machine learning algorithms.

The CAMELS project was first introduced and described in detail in Villaescusa-Navarro et al. [206]. The theoretical justification behind some of its main features (e.g. the use of a latin-hypercube covering a big volume in parameter space) was presented in Villaescusa-Navarro et al. [207]. Since then, a number of different works have made use of the CAMELS simulations to carry out a large and diverse variety of tasks:

1. In [195] CAMELS was used to identify a universal relation between subhalo properties using neural networks and symbolic regression.

2. In [208] CAMELS was used to train convolutional neural networks to inpaint masked regions of highly non-linear 2D maps from different physical fields.

3. In [146] CAMELS was used to show that neural networks can extract cosmological information and marginalize over baryonic effects at the field level using

multiple fields simultaneously.

4. In [147] CAMELS was used to show that neural networks can place robust, percent level, constraints on $\Omega_{\rm m}$ and σ_8 from 2D maps containing the total matter mass of hydrodynamic simulations.

5. In [209] the CAMELS Multified Dataset, a collection of hundreds of thousands of 2D maps and 3D grids for 13 different fields was presented and publicly released.

6. In [157] CAMELS was used to train a generative model that can produce diverse neural hydrogen maps by end of reionization (z$\sim$6) as a function of cosmology.

7. In [186], a model based on Graph Neural Networks (GNNs) was trained on the data from the CAMELS simulations to predict the total mass of a dark matter halo given its galactic properties while accounting for astrophysical uncertainties.

8. In [187] the GNN models proposed in [186] and trained on CAMELS data were used to obtain the first constrain on the mass of the Milky Way and Andromeda using artificial intelligence.

9. In [210] CAMELS was used to investigate the potential of auto- and cross-power spectra of the baryon distribution to robustly constrain cosmology and baryonic feedback.

10. In [211] CAMELS was used to reduce the scatter in the Sunyaev-Zeldovich (SZ) flux-mass relation, Y-M, to provide more accurate estimates of cluster masses.

11. In [212] CAMELS was used to study deviations from self-similarity in the Y-M relation due to baryonic feedback processes, and to find an alternative relation which is more robust.

12. In [213] CAMELS was used to demonstrate the strong constraints that next-generation measurements of the y-distortions could provide on feedback models.

13. In Moser et al. [214] CAMELS was used to compute thermal and kinetic SZ profiles. A Fisher analysis was performed to forecast the constraining power of observed SZ profiles on the astrophysical models varied in the simulations.

14. In [215] CAMELS was used to investigate whether the value of the cosmological parameters can be constrained using properties of a single galaxy.

15. In Jo et al. [216] CAMELS has been exploited to infer the full posterior on the combinations of cosmological and astrophysical parameters that reproduce observations such as cosmic star formation history and stellar mass functions using simulation-based inference.

16. Perez et al. [217] created CAMELS-SAM, a third larger 'hump' of CAMELS by combining N-body simulations with the Santa Cruz semi-analytic model of galaxy formation. CAMELS-SAM contains billions of galaxies and represents a

perfect tool to investigate and quantify the amount of cosmological information that can be extracted with galaxy redshift surveys.

In this paper, we describe the characteristics of the CAMELS simulations together with a variety of data products obtained from them, and we publicly release all available data. This paper is accompanied by the online documentation hosted at `https://camels.readthedocs.io`, containing further technical details on how to access, read, and manipulate CAMELS data. We believe that the CAMELS data will trigger new developments and findings in the fields of cosmology and galaxy formation.

This paper is organized as follows. In Sec. 6.3 we briefly describe the simulations of the CAMELS project and their scientific goals. The specifications of the data release are outlined in detail in Sec. 6.4. In Sec. 6.5 we describe how to access and download the data together with the overall data organization. We conclude in Sec. 6.6.

6.3 Simulations

6.3.1 Overview

CAMELS consists of a set of 4,233 cosmological simulations: 2,049 N-body and 2,184 hydrodynamic. All simulations follow the evolution of 256^3 dark matter particles and 256^3 fluid elements (only the hydrodynamic simulations) from $z = 127$ down to $z = 0$ in a periodic box of $(25\ h^{-1}\mathrm{Mpc})^3$ volume. The initial conditions were generated at $z = 127$ using second order perturbation theory (2LPT)[2]. The linear power spectra were computed using CAMB [12]. The mass resolution is approximately $1.27 \times 10^7\ h^{-1}M_\odot$ per baryonic resolution element and the gravitational softening length is approximately 2 kpc. For each simulation we have saved 34 snapshots, from $z = 6$ to $z = 0$. All simulations share the value of these cosmological parameters: $\Omega_\mathrm{b} = 0.049$, $h = 0.6711$, $n_s = 0.9624$, $\sum m_\nu = 0.0$ eV, $w = -1$. However, the value of Ω_m and σ_8 varies from simulation to simulation.

The state-of-the-art hydrodynamic simulations have been run using two different codes, AREPO [218, 219] and GIZMO [220], and they made use of the IllustrisTNG [221, 222] and SIMBA [223] galaxy formation models, respectively. However, the values of four astrophysical parameters vary from simulation to simulation. Two

[2] `https://cosmo.nyu.edu/roman/2LPT/`

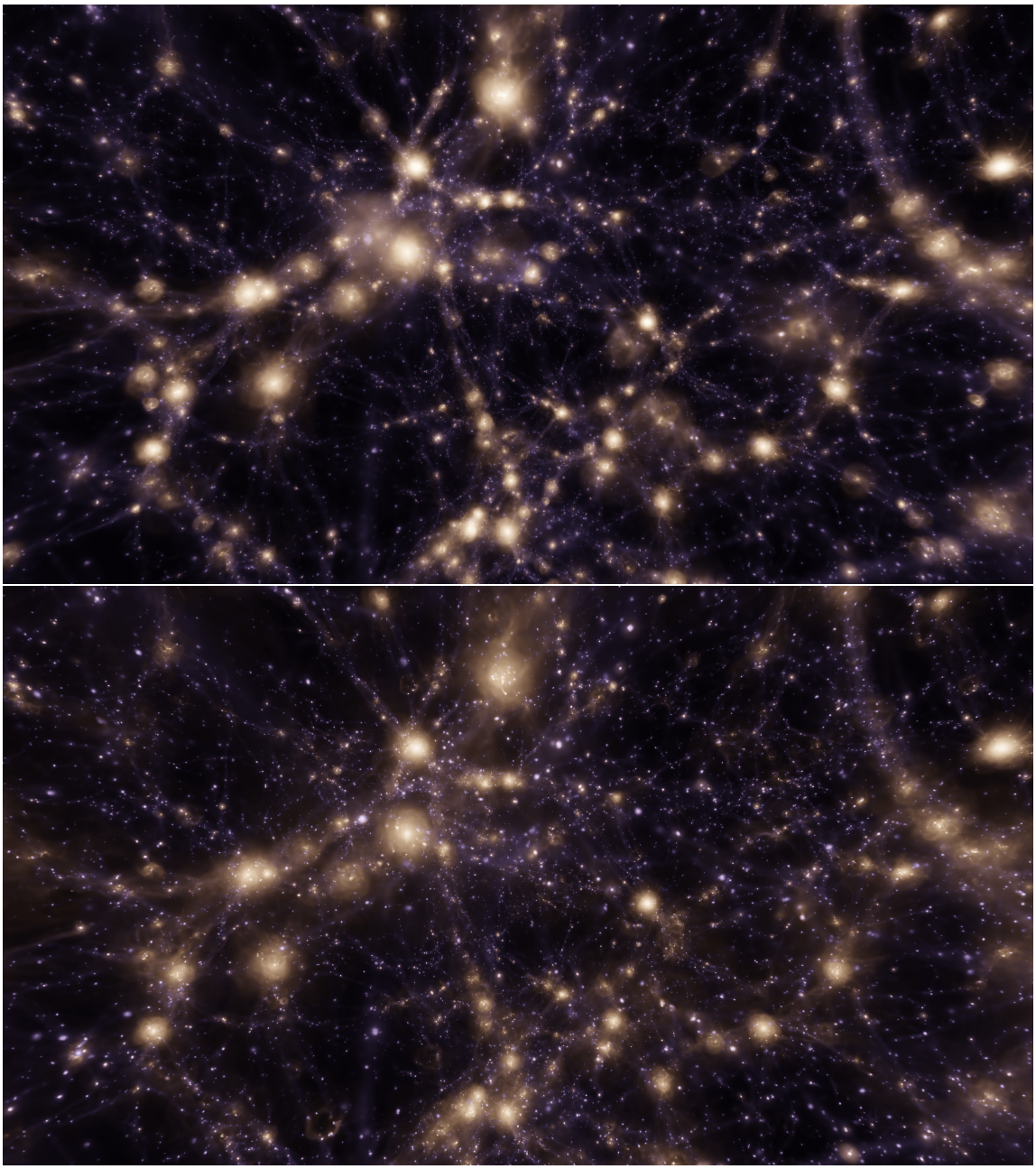

Figure 6.1: We show two images of the gas distribution of two distinct IllustrisTNG simulations. The one on the top displays the results for a simulation with high supernova feedback strength, while the one on the bottom is from a simulation with low supernova feedback. The color represents gas temperature, while its brightness corresponds to the gas density. Finally, we apply an extinction based on gas metallicity. As can be seen, the effect of feedback is very pronounced: it not only affects the gas abundance and temperature on the smallest galaxies but it also changes the gas distribution in the most massive galaxies.

parameters, $A_{\rm SN1}$ and $A_{\rm SN2}$, control the efficiency of supernova feedback, while the other two parameters, $A_{\rm AGN1}$ and $A_{\rm AGN2}$, parametrize the efficiency of feedback from supermassive black holes, as described in more detail below. In Fig. 6.1 we illustrate visually the effect of changing one of the astrophysical parameters in one simulation. As can be seen, while the large-scale structure remains unchanged, changing the efficiency of supernova feedback has a large effect on both small and large galaxies.

For each hydrodynamic simulation, CAMELS includes its N-body counterpart. The N-body simulations have been run with GADGET-III [224]. With the simulation snapshots and initial conditions we also release the GADGET parameter files, CAMB parameters files, and linear power spectra used to run the simulations.

6.3.2 Organization

The CAMELS simulations are divided into three different suites:

† **IllustrisTNG**. All simulations run with the AREPO code and employing the IllustrisTNG model belong to this suite. There are 1,092 IllustrisTNG simulations in CAMELS.

† **SIMBA**. All simulations run with the GIZMO code and employing the SIMBA subgrid model belong to this suite. There are 1,092 SIMBA simulations in

CAMELS.

† **N-body**. All N-body simulations belong to this suite. There are 2,049 N-body simulations in CAMELS.

We provide further details on each suite below. Each simulation suite contains four different sets, depending on the way the values of the cosmological parameters, astrophysical parameters, and initial conditions random phases are organized:

† **LH** stands for *latin-hypercube*. This set contains 1,000 simulations, each with different values of $\Omega_{\rm m}$, σ_8, $A_{\rm SN1}$, $A_{\rm SN2}$, $A_{\rm AGN1}$, $A_{\rm AGN2}$, and the initial conditions random phases. In the case of the N-body suite, this set contains 2,000 simulations varying $\Omega_{\rm m}$, σ_8 and the initial conditions random phases, such that they match those from the IllustrisTNG and SIMBA LH sets.

† **1P** stands for *1 parameter at a time*. This set contains 61 simulations with the same values of the initial conditions random seed. The simulations only differ in the value of a single cosmological or astrophysical parameter at a time, with 11 variations for each, including the set of fiducial values. In the case of the N-body suite, this set contains 21 simulations varying $\Omega_{\rm m}$ and σ_8.

† **CV** stands for *cosmic variance*. This set contains 27 simulations that share the values of the cosmological and astrophysical parameters. The simulations only differ in the value of the initial conditions random seed. There are 27 N-body

counterpart simulations for this set.

† **EX** stands for *extreme*. This set contains 4 simulations that have the same value of the initial conditions random seed and the same value of the cosmological parameters. One of them represents a model with no feedback, while the other two have either extremely large supernova or AGN feedback. The N-body suite only contains 1 simulation.

For further details on the CAMELS simulations we refer the reader to Villaescusa-Navarro et al. [206] and references therein.

6.3.3 Parameters

Both the IllustrisTNG and SIMBA simulation suites model galaxy formation by following Newtonian gravity in an expanding background, hydrodynamics, radiative cooling, star-formation, stellar evolution and feedback, SMBH growth and AGN feedback. IllustrisTNG also follows magnetic fields in the MHD limit and SIMBA follows dust grains. The implementations of gravity and hydrodynamics solvers differ between the codes, as well as the parameterizations of radiative cooling, star-formation and stellar evolution. However, the most consequential differences between the suites are in the implementations of feedback in the form of galactic winds and from AGN, since the physics of these processes are the least theoretically understood as well as

least observationally constrained. Therefore, these are also the parts of the physical modeling which we have chosen to apply variations to, through the parameters mentioned above, $A_{\rm SN1}$, $A_{\rm SN2}$, $A_{\rm AGN1}$, $A_{\rm AGN2}$, as described next.

CAMELS was designed to sample a large volume in parameter space. Thus, the value of both the cosmological and astrophysical parameters is varied within a very broad range:

$$\Omega_{\rm m} \in [0.1, 0.5], \tag{6.1}$$

$$\sigma_8 \in [0.6, 1.0], \tag{6.2}$$

$$A_{\rm SN1} \in [0.25, 4.0], \tag{6.3}$$

$$A_{\rm SN2} \in [0.5, 2.0], \tag{6.4}$$

$$A_{\rm AGN1} \in [0.25, 4.0], \tag{6.5}$$

$$A_{\rm AGN2} \in [0.5, 2.0]. \tag{6.6}$$

In both the LH and 1P sets, the value of $\Omega_{\rm m}$ and σ_8 is sampled linearly, while the value of the astrophysical parameters is varied in logarithmic scale.

In both models, $A_{\rm SN1}$ represents a normalization factor for flux of the galactic wind feedback. In IllustrisTNG it is implemented as a pre-factor for the overall energy

output per unit star-formation [222], while in SIMBA it is implemented as a pre-factor for the mass-loading factor (wind mass outflow rate per unit star-formation rate) relative to that predicted by higher-resolution simulations [225]. In both models, $A_{\rm SN2}$ represents a normalization factor for the speed of the galactic winds. This implies that for a fixed $A_{\rm SN1}$, changes in $A_{\rm SN2}$ in IllustrisTNG affect the wind speed in concert with the mass-loading factor (to keep a fixed energy output), while in SIMBA changes in $A_{\rm SN2}$ affect the wind speed in concert with the wind energy flux (with a fixed mass-loading factor).

In both models, $A_{\rm AGN1}$ represents a normalization factor for the energy output of AGN feedback while $A_{\rm AGN2}$ affects the specific energy of AGN feedback. However, the implementations of AGN feedback are quite significantly different between the suites and so is the effect of those parameters. In IllustrisTNG, $A_{\rm AGN1}$ is implemented as a pre-factor for the overall power injected in the 'kinetic' feedback mode [221], while in SIMBA it is implemented as a pre-factor for the momentum flux of mechanical outflows [226] in the 'quasar' and 'jet' feedback modes. In IllustrisTNG, $A_{\rm AGN2}$ directly parameterizes the burstiness and the temperature of the heated gas during AGN feedback 'bursts', while in SIMBA it controls the speed of continuously-driven AGN jets. We refer the reader to Villaescusa-Navarro et al. [206] for a detailed description of the feedback parameter variations in CAMELS.

It is very important to remark that, in light of the discussion above, while the cosmological parameters in the N-body, IllustrisTNG, and SIMBA suites represent the very same physical effect, the astrophysical parameters in the SIMBA and IllustrisTNG suites do not. The reason is that these parameters characterize similar physical processes but in different subgrid models. Thus, one should not attempt to match these parameters across suites. In other words, when doing, e.g., parameter inference from some observable to the value of the cosmological and astrophysical parameters, and the model is trained on IllustrisTNG simulations, one can attempt to test the model to see if it is able to recover the correct cosmology from SIMBA simulations. On the other hand, one should not try to infer the value of the astrophysical parameters of IllustrisTNG simulations from a model trained on SIMBA simulations.

To illustrate the differences between the IllustrisTNG and SIMBA simulations we have taken all galaxies of all simulations belonging to the LH sets of both suites. For each galaxy we consider 8 different properties and in Fig. 6.2 we show 1D and 2D distributions of them. As can be seen, while the distributions overlap in all cases, there are noticeable differences in all cases.

6.4 Data Description

In this section we describe the different data products we release.

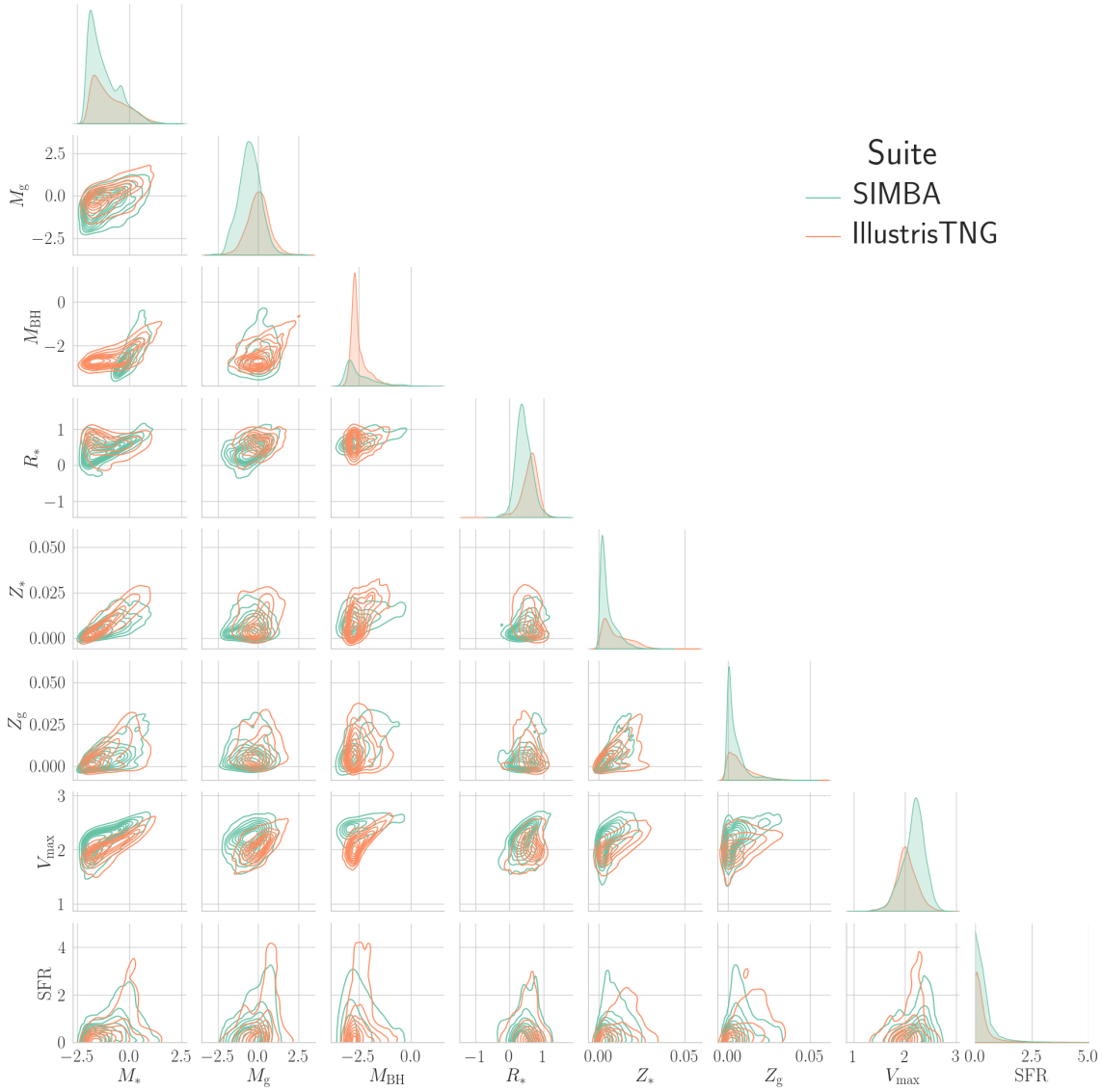

Figure 6.2: In this plot we illustrate the similarities and differences between the IllustrisTNG and SIMBA suites considering eight different properties of the subhalos: 1) stellar mass, M_*, 2) gas mass, M_g, 3) black-hole mass, M_{BH}, 4) stellar half-mass radius, R_*, 5) stellar metallicity, Z_*, 6) gas metallicity, Z_g, 7) maximum circular velocity, V_{max}, and 8) star-formation rate, SFR. We show the 1-dimensional and 2-dimensional distribution of these properties for all galaxies in the LH sets of the IllustrisTNG (orange) and SIMBA (green) suites. Masses are in units of $10^{10}/(M_\odot/h)$, R_* in kpc/h, V_{max} in km/s and SFR in $M_\odot$/yr; the logarithm of each variable is shown except for the metallicity and the SFR.

6.4.1 Snapshots

We release the full snapshots generated by the GADGET-III, AREPO, and GIZMO codes. For each simulation, we have 34 snapshots from $z = 6$ down to $z = 0$ (we provide further details in the online documentation about the redshifts of the snapshots). We also release the initial conditions of each simulation.

All initial condition files and the snapshots of all simulations contain the positions, velocities, and IDs of the particles. The snapshots of the hydrodynamic simulations contain additional fields that store properties of the gas, stars, and black-hole particles. Examples are the masses of the particles, the electron fraction from gas or the age of the star particles. We note that the simulations from the IllustrisTNG and SIMBA suites are not identical in terms of the fields they store. The differences reflect the different subgrid models employed in these two simulations. The structure and contents of the IllustrisTNG snapshots are the same as in the publicly released IllustrisTNG simulation data [227]. Likewise, the SIMBA snapshots are identical in format to that available on the publicly released SIMBA database[3].

The snapshots are stored as `hdf5` files, and we provide details in the online documentation on how to read and manipulate the data from them.

[3] `http://simba.roe.ac.uk`

6.4.2 Halo and subhalo catalogues

We release the halo and subhalo catalogues generated from the CAMELS simulations. The halos and subhalos have been identified using SUBFIND [228, 229], ROCKSTAR [230], and the Amiga Halo Finder [AHF; 231]. The codes have been run on top of all snapshots of all simulations. In total, we release 506,022 catalogues that contain millions of halos, subhalos, and galaxies. We now describe the catalogues in more detail.

6.4.2.1 Subfind

SUBFIND [228, 229] was run on-the-fly for the IllustrisTNG simulations while for the SIMBA and N-body simulations it was run in post-processing. SUBFIND identifies both halos and subhalos, and computes several physical quantities for them in both the N-body and hydrodynamic simulations. We release all SUBFIND catalogues (one per simulation and redshift) for all simulations and all redshifts. The data is stored as `hdf5` files, and we provide details on how to read the data and the information stored in them in the online documentation.

6.4.2.2 Amiga Halo Finder

AHF [231] was run in post-processing for the IllustrisTNG and SIMBA simulations. AHF utilizes isodensity contours to locate halo centers. Halo virial radii are defined to represent spherical overdensity regions with 200 times the critical density. We release the AHF catalogues for all simulations and redshift snapshots, including (1) global halo properties, (2) radial profiles, and (3) particle ID lists to identify the host halo of each particle. We provide further details on the format and how to read these catalogues in the online documentation.

6.4.2.3 Rockstar

In addition to the SUBFIND and AHF halo catalogs, we also release halo catalogs constructed using the ROCKSTAR halo finder [230]. ROCKSTAR identifies dark matter halos based on an adaptive hierarchical refinement of friends-of-friends in six-dimensional phase space plus time. Substructures are identified using successively smaller linking length and particles are assigned to the inner-most substructures, which are defined as halo seeds, based on their phase-space proximity. We release ROCKSTAR halo catalogs of all 34 snapshots from $z = 6$ to 0 for all simulations.

Furthermore, we use CONSISTENT-TREES [232] to generate merger trees from the

ROCKSTAR halo catalogs. We note that CONSISTENT-TREES ensures consistency of halo mass, position, and velocity across time steps. Since all CAMELS simulations have only 34 snapshots, we perform the following exercise to quantify its validity. We have compared the ROCKSTAR + CONSISTENT-TREES outputs at $z = 0$ from two CAMELS simulations that have the same initial conditions but different time resolution (34 versus 200 snapshots). We find good agreement between the outputs for certain proxies of merger history such as peak mass and half-mass assembly time. However, we caution readers when using more detailed properties of the halo merger histories, such as accretion history, which are affected by the lower time sampling. All ROCKSTAR catalogues and CONSISTENT-TREES merger trees occupy 1.2 Terabytes of data.

6.4.2.4 CAESAR

We release full cross-matched galaxy/halo catalogs for each snapshot generated using the YT extension package CAESAR[4]. CAESAR identifies halos based on a 3-D friends-of-friends (FoF) algorithm using a linking length of 0.2 times the mean interparticle spacing, and within each halo it identifies galaxies based on a 6D FoF with a linking length of 0.0056 times the mean interparticle spacing applied only to dense gas (hydrogen number density $n_H < 0.13$ cm^{-3}) and star particles.

[4]caesar.readthedocs.io

CAESAR then calculates a huge number of properties for each halo and galaxy, including physical properties such as masses and sizes for each component, dynamical properties such as velocity dispersions, and photometric properties using the Flexible Stellar Population Synthesis [233, 234] package[5]. There are over 130 bands pre-computed, both apparent and absolute magnitudes, both with and without dust extinction. Extinction is calculated via the line-of-sight dust content to each star along a chosen viewing axis (for IllustrisTNG, a Milky Way-like dust-to-metal ratio is assumed), providing pseudo-radiative transfer that generally agrees with full radiative transfer calculations within 0.1 magnitudes. An extinction law is assumed that is a composite of Milky Way for low galaxy specific star formation rates (sSFR$< 10^{-10}$ yr^{-1}), Calzetti for high (sSFR$> 10^{-9}$ yr^{-1}), and interpolated in between, with a further interpolation in galaxy stellar metallicity to incorporate an SMC law such that at above solar metallicity no SMC law is folded in, while for metallicities below one-tenth solar it is fully SMC (regardless of sSFR).

All this information is stored in a single `hdf5` file for each snapshot, called a CAESAR catalog. Quantities from the catalog can be loaded into CAESAR using simple Python list comprehension, and it is straightforward to access halo information for a given galaxy and vice versa. CAESAR also provides particle membership lists for each galaxy/halo, so that one can compute any user-desired quantity by loading the

[5]See `https://caesar.readthedocs.io/en/latest/catalog.html` for the full list of quantities.

particles from the original snapshot[6]. CAESAR also provides the functionality to compute progenitors and descendants of galaxies and/or halos across different snapshots, though this information has not been pre-computed.

CAESAR catalogs typically are roughly 1% of the size of the corresponding snapshots, so they provide a compact and manageable way to access galaxy and halo data quickly and conveniently. CAESAR also interfaces seamlessly with `yt` for further analysis and visualization. See the online documentation and `caesar.readthedocs.io` for more details.

6.4.3 Void catalogues

We release void catalogs built from the CAMELS simulations with the Void IDentification and Examination toolkit (`VIDE`) [235]. `VIDE`, based on `ZOBOV` [236], has been widely used to find voids both in data—e.g. voids from the SDSS BOSS [237, 238] and eBOSS [239] datasets, or data from DES [240]—and simulations [e.g. 241–244]. Furthermore, `VIDE` has also been applied to hydrodynamic simulations [245, 246], showing its suitability for the CAMELS dataset.

`VIDE` was run on top of CAMELS galaxies, that were defined as subhalos containing more than 20 star particles. Given the size of the CAMELS simulations, and the

[6]CAESAR works most straightforwardly with the PYGADGETREADER package for this; see the CAESAR documentation for examples.

extended size of cosmic voids (that usually span sizes from $5 - 100\ h^{-1}$Mpc), the number of voids for each CAMELS simulation is relatively small. The `VIDE` catalogues store information about the positions, sizes, ellipticities, and member galaxies of each void. In the online documentation we provide further details on how to read and manipulate the `VIDE` catalogues.

6.4.4 Lyman-alpha spectra

We release mock Lyman-α spectra generated using a public, well-tested code exhibited in Bird et al. [247] and used previously for studies of the Lyman-α Forest in Gurvich et al. [248]. The spectra is generated for 5,000 sightlines randomly placed through the simulation box. This spectral data was generated for the IllustrisTNG and SIMBA suites for all simulation sets at all redshifts. The locations of the random sightlines vary across snapshots.

The total absorption along a sightline is the sum of the absorption from all the nearby gas cells. The simulated spectra has a spectral resolution of 1 km/s and any lines with an optical depth of $\tau < 10^{-5}$ are neglected. For further details on the artificial spectra calculation, we direct the reader to Bird et al. [247]. In the online documentation we provide details on how to read and manipulate the Lyman-α spectra.

6.4.5 Summary statistics

We release a large set of summary statistics, containing power spectra, bispectra, and probability distribution functions. This data can be used for a large variety of tasks such as carrying out parameter inference and building emulators.

6.4.5.1 Power spectrum

The power spectrum is the most prominent summary statistic of cosmology. The procedure used to carry out this task is the following. First, the positions and masses of the considered particles are read from the snapshots. Next, the masses of the particles are deposited into a regular grid with 512^3 voxels using the Cloud-in-Cell mass-assignment scheme (MAS). We then Fourier transform that field and correct modes amplitudes to account for the MAS. Finally, the power spectrum is computed by averaging the square of the modes amplitudes

$$P(k_i) = \frac{1}{N_i} \sum_{k \in k_{\rm bin}} |\delta(\mathbf{k})|^2 \,, \tag{6.7}$$

where the k-bins have a width equal to the fundamental frequency, $k_F = 2\pi/L$ (L is the box size), and N_i is the number of modes in the k-bin. The wavenumber

associated with each bin is

$$k_i = \frac{1}{N_i} \sum_{k \in k_{\rm bin}} k \ . \tag{6.8}$$

We have computed the power spectra of the total matter for both the N-body and the hydrodynamic simulations. Besides, for the hydrodynamic simulations we have also computed the power spectra of the gas, dark matter, stars, and black hole components. We have done this for all snapshots of each simulation. We have made use of PYLIANS[7] to carry out the calculation. In total, we release 440,946 power spectra. All power spectra occupy $\simeq 10$ Gigabytes of data.

The above methods are inefficient if we wish to compute the power spectrum at large k, since they require a unwieldy FFT grid. In this regime, alternative methods such as configuration-space power spectrum estimators [249] can be of use, since their computational cost decreases as the minimum scale increases. We provide power spectrum multipoles computed up to $k = 1,000\ h\,{\rm Mpc}^{-1}$ and $\ell_{\rm max} = 4$, using a combination of the above PYLIANS code and the HIPSTER pair-counting approach package [250], switching between the two at $k = 25\ h\,{\rm Mpc}^{-1}$ and convolving the small-scale spectra with a window of size $R_0 = 1\ h^{-1}{\rm Mpc}$ for efficiency. Spectra are computed at $z = 0$ for all matter species listed above, and we include results from each simulation of the LH set from the IllustrisTNG, SIMBA, and N-body suites in both real- and redshift-space, with the latter using three choices of redshift-space axis. In total we compute 44,000 power spectra up to $k = 1,000\ h\,{\rm Mpc}^{-1}$, requiring

[7] `https://pylians3.readthedocs.io`

$\simeq 14,000$ CPU-hours and occupying $\simeq 0.6$ Gigabytes of storage.

For all spectra, we store the value of k in each k-bin, the value of $P(k_i)$, and (for the large-scale spectra) the number of modes in each bin. We provide further details on how to read and manipulate these files in the online documentation.

6.4.5.2 Bispectrum

On large scales, the first non-Gaussian statistic of interest is the bispectrum, encoding the three-point average of the density field. In this release, we provide bispectrum measurements from gas, dark matter and total matter for the 1,000 simulations of the LH set of the IllustrisTNG and SIMBA suites, as well as 1,000 N-body simulations. These are performed at redshift zero, both in real-space and redshift-space (for three choices of line-of-sight). Additional data can be computed upon request.

On large scales, bispectra are computed analogously to §6.4.5.1, first gridding the data with 128^3 voxels using a Triangular-Shaped-Cloud MAS scheme. We then use the Pylians estimator [251], implementing the approach of Watkinson et al. [252], which practically computes the following sum via a series of FFTs:

$$B(k_1, k_2, \mu) = \frac{\sum_{\mathbf{k}_1} \sum_{\mathbf{k}_2} \delta(\mathbf{k}_1)\delta(\mathbf{k}_2)\delta(-\mathbf{k}_1 - \mathbf{k}_2)}{N_T(k_1, k_2, \mu)}. \tag{6.9}$$

The bispectrum is parametrized by two lengths, k_1 and k_2, and an internal angle $\mu \equiv \hat{\mathbf{k}}_1 \cdot \hat{\mathbf{k}}_2$, with $N_T(k_1, k_2, \mu)$ giving the number of triangles per bin. We use 20 k-bins with $\Delta k = 0.25\ h\,\mathrm{Mpc}^{-1} \approx k_F$, and ten linearly spaced μ bins.

The above method becomes prohibitively expensive as $k_{\rm max}$ (and thus the FFT grid) increases. To ameliorate this, we compute the bispectra at large k using the HIPSTER code, as for the small-scale power spectra, here convolving the spectra with a smooth window of scale $R_0 = 2\ h^{-1}$Mpc. This computes the Legendre multipoles of the bispectrum, related to Eq. 6.9 by

$$B(k_1, k_2, \mu) = \sum_{\ell=0}^{\infty} B_\ell(k_1, k_2) L_\ell(\mu), \tag{6.10}$$

for Legendre polynomial $L_\ell(\mu)$, and uses 25 linearly spaced k-bins in the range $[0, 50] h\,\mathrm{Mpc}^{-1}$ for $\ell \leq 5$, subsampling to 10^5 particles for efficiency. These bispectra are computed for the same simulations as before, and will allow information to be extracted from very small scales. In total, 28,000 bispectra are estimated using each method, requiring $\simeq$ 70,000 CPU-hours and $\simeq$ 2.1 Gigabytes of storage.

6.4.5.3 Probability distribution function

We estimate probability distribution functions (PDF) for 13 different physical fields using the 3D grids of the CAMELS Multifield Dataset (CMD) (see Sec. 6.4.8). The

PDFs are calculated for all the fields: 1) gas temperature, 2) gas pressure, 3) neutral hydrogen density, 4) electron number density, 5) gas metallicity, 6) gas density, 7) dark matter density, 8) total mass density, 9) stellar mass density, 10) magnetic fields, 11) ratio between magnesium over iron, 12) gas velocity, and 13) dark matter velocity, for all the grid sizes, i.e., 128, 256 and 512 at redshifts 0.0, 0.5, 1.0, 1.5, and 2.0. The PDFs are calculated as follows. First, the 1,000 3D grids from all simulations in the LH set are read into memory. We then calculate the minimum value across grids and if it equals 0, a small offset is added to all voxels of all grids. The offset, ε, is given by

$$\varepsilon = \frac{\min_{\mathrm{non-zero}}}{10^{20}}, \tag{6.11}$$

where $\min_{\mathrm{non-zero}}$ denotes the non-zero minimum of all the 1,000 grids. Then we log-transform the entire field (to the base 10) and construct a histogram of 500 bins between the minimum and maximum values of the entire field. Finally, we save to disk the number of counts in each bin for each grid in the considered field.

6.4.6 Profiles

We provide three-dimensional spherically-averaged profiles of gas density, thermal pressure, gas mass-weighted temperature, and gas mass-weighted metallicity for the

1P, LH, and CV sets of both the IllustrisTNG and SIMBA suites. We follow Moser et al. [253] in extracting halo information and construction of the profiles. Specifically, we use `illstack_CAMELS`[8] (a CAMELS-specific version of the original, more general code `illstack` used in Moser et al. 253), to generate the three-dimensional profiles, extending radially from $0.01-10$ Mpc in 25 $\log_{10}$ bins. The profiles are stored in `hdf5` format which can be read with the python script provided in the `illstack_CAMELS` repository.

6.4.7 X-rays

We provide mock X-ray photon lists in the form of SIMPUT fits files for all halos above 10^{12} $M_\odot$ across all hydrodynamic CAMELS runs at redshift $z = 0.05$ obtained from the snapshot 032. The SIMPUT files are generated using the pyXSIM package[9] and contain positional coordinates in RA and DEC coordinates and energy in units of keV. These files serve as inputs into other software packages, including SOXS[10] and SIXTE [254] that generate mock observations for specific telescopes using custom instrument profiles. These SIMPUT files can also represent idealized observations by an X-ray telescope, and we also provide a single collated file with projected radial surface brightness (SB) profiles for all halos for the soft X-ray band (0.5-2.0 keV) in

[8] `https://github.com/emilymmoser/illstack_CAMELS`
[9] `http://hea-www.cfa.harvard.edu/~jzuhone/pyxsim/`
[10] `http://hea-www.cfa.harvard.edu/~jzuhone/soxs/`

units of erg s^{-1} kpc^{-2}. This file holds 1.6×10^5 radial profiles across the 2,190 1P, CV, LH, and EX simulations.

6.4.8 CAMELS Multifield Dataset

The CAMELS Multifield Dataset, CMD, is a collection of hundreds of thousands of 2D maps and 3D grids generated from CAMELS data. CMD contains 15,000 2D maps for 13 different fields at $z = 0$, and 15,000 3D grids, at three different spatial resolutions and at five different redshifts. The data was generated by assigning particles positions and properties (e.g. mass and temperature for the temperature field) to either 2D maps or 3D grids. There are many possible machine learning applications of this dataset, e.g.: 1) parameter inference [146, 147], 2) summary or field level emulation, 3) mapping N-body to hydrodynamic simulations, 4) superresolution, and 5) time evolution. In total, CDM represents over 70 Terabytes of data. We refer the reader to [209] and the CMD online documentation[11] for further details on this dataset.

6.4.9 CAMELS-SAM

CAMELS-SAM represents a newer third 'hump' of CAMELS, mimicking its construction and purpose but using larger N-body volumes that are populated with galaxies

[11] `https://camels-multifield-dataset.readthedocs.io`

using the Santa Cruz semi-analytic model (SAM, Somerville et al. 255, 256) of galaxy formation. The N-body simulations are run with AREPO [219], and follow the evolution of 640^3 dark matter particles over a periodic box of $(100\ h^{-1}\ \mathrm{cMpc})^3$ volume from $z = 127$ to $z = 0$. For each simulation we save 100 snapshots. The initial conditions were otherwise generated as described in §6.3, with the same underlying cosmology, and a newly generated latin hypercube varying Ω_m, σ_8, and three SAM parameters. Those parameters were chosen as the ones closest to the astrophysical parameters varied in CAMELS. Two parameters control the amplitude and rate of mass outflow from massive stars out of a galaxy, and the third parameter broadly controlling the strength of the radio jet mode of AGN.

Like CAMELS, CAMELS-SAM has an LH set containing 1,000 simulations. The values of the cosmological and astrophysical parameters in the set are organized in a latin-hypercube. We additionally have 5 simulations in the CV set where the value of the initial random seed varies and the 5 parameters are held fixed to their fiducial values. Finally, a 1P set with 12 simulations exists, where the SC-SAM was run at the smallest and largest value of each SAM parameter for two of the CV simulations.

It is important to emphasize the differences between the original CAMELS and the CAMELS-SAM simulations. First, CAMELS-SAM consists of N-body simulations with a volume 64× larger than the former, while CAMELS contains both N-body and hydrodynamic simulations. Second, CAMELS-SAM stored 100 snapshots while

CAMELS only kept 34. Third, galaxies are modelled in very different ways: in CAMELS they arise from the hydrodynamic simulations while in CAMELS-SAM they are modelled through the Santa Cruz semi-analytic model.

For all CAMELS-SAM simulations, we release:

† The halo and subhalo catalogues from both SUBFIND and ROCKSTAR.

† The merger trees generated from CONSISTENT TREES.

† The galaxy catalogues from the Santa Cruz SAM.

The galaxy catalogues are stored as *.dat* text files with comma-separated values. These files contain information about the halo and galaxies from all snapshots of a given simulation. The exact available properties, their organization and units, and example code to open these files can be found on the CAMELS-SAM online documentation[12]. The total size of these data products is around 35 Terabytes.

The raw data (compressing full N-body snapshots across redshifts) has been stored on tape and its content can be retrieved upon request. We refer the reader to Perez, Genel, et al. (2022) for further details on CAMELS-SAM, as well as a proof-of-concept of its power using clustering summary statistics to constrain cosmology and astrophysics with neural networks.

[12] `https://camels-sam.readthedocs.io`

6.5 Data Access and structure

In this section we describe the different methods to access the data and its structure.

6.5.1 Data Access

We provide access to CAMELS data through four different platforms:

† **Binder**. Binder is a system that allows users to read and manipulate data that is hosted at the Flatiron Institute through either a Jupyter notebook or a unix shell. The system provides access to the entire CAMELS data and allows users to perform calculations that do not require large amounts of CPU power. We note that heavy calculations are not supported by this system, and we recommend the user to download the data locally and work with it accordingly. We provide the link to the Binder environment in the online documentation. All CAMELS data can be accessed, read, and manipulated through Binder.

We provide further technical details on Binder usage in the online documentation.

† **Globus**. Globus[13] is a system designed to transfer large amounts of data in an

[13] `https://www.globus.org`

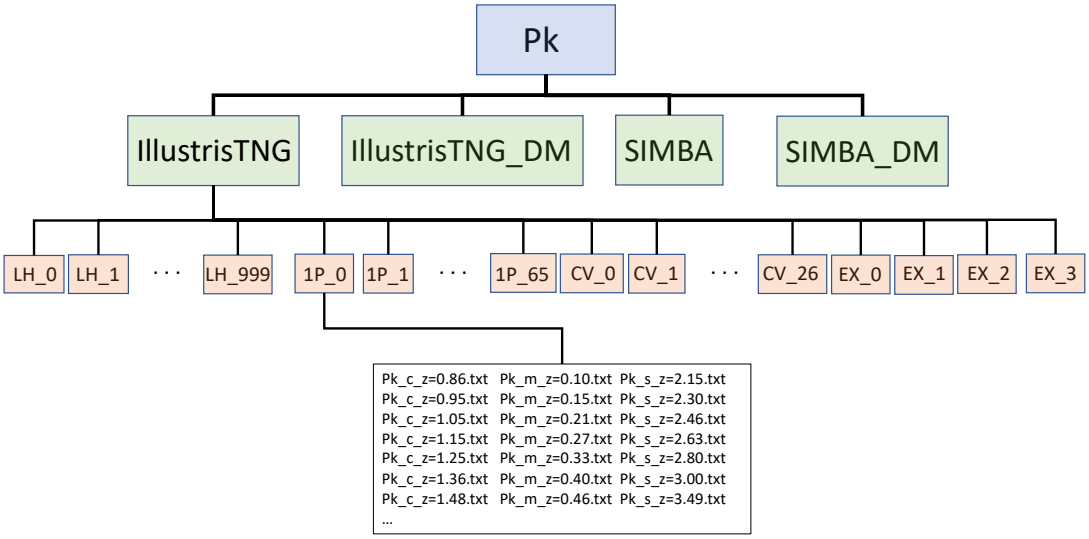

Figure 6.3: This scheme shows the generic structure of CAMELS data. The top level represents the type of data it contains (power spectra in this case). Inside that folder there are typically four folders containing the data for the three different simulation suites: IllustrisTNG, SIMBA, and their N-body counterparts (`IllustrisTNG_DM` and `SIMBA_DM`). Within each of those folders there are numerous folders, containing the data from the different simulations belonging to each suite; i.e. the simulations from the four sets: LH, 1P, CV, and EX. Finally, inside each of those folders the user can find the data products themselves. In this particular case, the power spectra for the different component.

efficient way. All CAMELS data can be transferred through globus. We provide the globus link in the online documentation[14]. Users can transfer the data to either another cluster or directly to their personal computer.

† **Url**. We also provide a uniform resource locator (url) to access the data through a browser. We do not recommend transferring large quantities of data using this procedure, as both the speed and its reliability is much worse than globus.

[14]Since this link may change with time, we make it available in the online documentation, where it can be updated if needed.

On the other hand, to download small amounts of data, such as a particular power spectrum or a halo catalogue, it may be useful. All CAMELS data can be accessed and downloaded through the url. We provide the url link in the online documentation where it will be always updated.

† **FlatHUB**. FlatHUB is a platform that allows users to explore and compare data from different simulations by browsing and filtering the data, making simple preview plots, and downloading sub-samples of the data. We provide access to the SUBFIND halo and subhalo catalogues of the IllustrisTNG and SIMBA suites through this platform. We provide a link to FlatHUB in the online documentation.

6.5.2 Data Structure

The data is organized in different folders that contain similar type of data:

† **Sims**. This folder contains the raw data from the simulations, such as initial conditions, snapshots, and parameter files. This folder contains 205 terabytes of data.

† **FOF_Subfind**. This folder contains the SUBFIND halo and subhalo catalogues described in Sec. 6.4.2.1. This folder contains 4 terabytes of data.

† **AHF**. This folder contains the AHF halo catalogues described in Sec. 6.4.2.2. This folder contains 6 terabytes of data.

† **Rockstar**. This folder contains the ROCKSTAR halo and subhalo catalogues together with the merger trees from CONSISTENT-TREES as described in Sec. 6.4.2.3. This folder contains 1 terabyte of data.

† **Caesar**. This folder contains the CAESAR halo and galaxy catalogues described in Sec. 6.4.2.4. This folder contains around 1 terabyte of data.

† **Pk**. This folder contains the power spectra described in Sec. 6.4.5.1. This folder contains approximately 10 gigabytes of data.

† **Bk**. This folder contains the bispectra measurements described in Sec. 6.4.5.2. This folder contains approximately 2.6 gigabytes of data.

† **CMD**. This folder contains the CAMELS Multifield Dataset. This folder contains 76 terabytes of data.

† **VIDE_Voids**. This folder contains the void catalogues described in Sec. 6.4.3. This folder contains 200 megabytes of data.

† **Lya**. This folder contains the Lyman-α spectra described in Sec. 6.4.4. This folder contains 14 terabytes of data.

† **PDF**. This folder contains the probability distribution function measurements described in Sec. 6.4.5.3. This folder contains more than 1 gigabyte of data.

† **Profiles.** This folder contains the spherically-averaged 3D profiles described in Sec. 6.4.6. This folder contains 48 gigabytes of data.

† **X-rays**. This folder contains the X-rays photon lists described in Sec. 6.4.7. This folder contains over 100 gigabytes of data.

† **SCSAM**. This folder contains all CAMELS-SAM data products described in Sec. 6.4.9. This folder contains more than 50 terabytes.

† **Utils**. This folder contains additional files that can be useful to the user, including a file with the value of the scale factors corresponding to simulation snapshots and files indicating the values of the cosmological and astrophysical parameters of each simulation.

When possible, we have organized the data in the different folders in a self-similar way. We show the generic data structure scheme in Fig. 6.3. The data is first organized into folders that contain: 1) the IllustrisTNG hydrodynamic simulations, 2) the SIMBA hydrodynamic simulations, 3) the N-body counterparts of 1), and 4) the N-body counterparts of 2). Inside each of these folders the user can find many different sub-folders whose name refers to the specific simulation set and realization: e.g. the first simulation of the LH set is denoted as LH_0. Finally, inside each of those folders the user can find the data with the particular characteristics of each data product. We note that these folders may contain data products for a particular CAMELS simulation at all redshifts.

For some data products, e.g. CMD and CAMELS-SAM, the data organization is slightly different to the one outlined above. In those cases, we provide further details in the online documentation.

6.6 Summary

The goal of the CAMELS project is to connect cosmology with astrophysics via thousands of state-of-the-art cosmological hydrodynamic simulations and extract the maximum amount of information from them via machine learning. CAMELS contains 4,233 cosmological simulations, 2,049 N-body simulations and 2,184 state-of-the-art hydrodynamic simulations sampling a vast volume in parameter space using two independent codes that solve hydrodynamic equations and implement subgrid physics in very distinct ways. CAMELS data have already been used for a large variety of tasks, from providing the first constraints on the mass of the Milky Way and Andromeda galaxies using artificial intelligence to showing that neural networks can extract information from vastly different physical fields while marginalizing over astrophysical effects at the field level.

In this paper we have described the characteristics of the CAMELS simulations and a variety of additional data generated from them, including halo, subhalo, galaxy, and void catalogues, power spectra, bispectra, Lyman-α spectra, probability distribution

functions, radial profiles, and X-rays photon lists. We have also described CAMELS-SAM, a collection of more than 1,000 galaxy catalogues created by applying the Santa Cruz Semi-Analytic Model to a set of hundreds of N-body simulations. We have made all this data publicly available, comprising hundreds of terabytes. We provide access to the data through different platforms, including a Binder environment for interactive data manipulation with Jupyter notebooks, a Globus link for efficient transfer of large amounts of data, and the FlatHUB platform for quick exploration of SUBFIND (sub)halo catalogues. We emphasize that the information outlined in this paper may become outdated as additional data products become available over time. However, the online documentation located at `https://camels.readthedocs.io` will always be updated accordingly.

It is also important to be aware of the limitations associated to the CAMELS simulations. First, the volume sampled by each individual simulation is relatively small, $(25\ h^{-1}\mathrm{Mpc})^3$, inhibiting the formation of the most extreme objects in the Universe such as galaxy clusters and large voids. Second, while CAMELS covers a large volume in parameter space, it would be desirable to make it even larger by including other cosmological and astrophysical parameters. Third, CAMELS only contains two distinct suites of hydrodynamic simulations: IllustrisTNG and SIMBA. Ideally, we would like to expand CAMELS to simulations performed with additional codes employing different subgrid models. Fourth, the resolution of CAMELS may not be high enough for some astrophysical problems. Future versions of CAMELS will be

designed to tackle these limitations.

We believe that CAMELS data will become a powerful tool for the community.

ACKNOWLEDGEMENTS

We are indebted to the high-performance computing system administrators at the Flatiron Institute and Princeton University for their invaluable help and work accommodating the CAMELS storage needs. The authors are pleased to acknowledge that the work reported in this paper was partially performed using the Research Computing resources at Princeton University which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and the Office of Information Technology's Research Computing Division. Further technical details on the CAMELS simulations and instructions to download the data can be found in `https://camels.readthedocs.io` and `https://www.camel-simulations.org`. The work of FVN, SG, DAA, SH, OP, AP, KW, WC, ME, US, DS, BB, BW, RS, and GB was supported by the Simons Foundation. DAA was supported in part by NSF grants AST-2009687 and AST-2108944. PVD acknowledges support from the Generalitat Valenciana through the GenT program (CIDEGENT/2018/019, CPI-21-108). BB is grateful for generous support by the David and Lucile Packard Foundation and Alfred P. Sloan Foundation. The Flatiron Institute is supported by the Simons

Foundation.

Chapter 7

A Novel RID Method of Muon Trajectory Reconstruction in Water Cherenkov Detectors

7.1 Abstract

Cosmic rays that strike the top of the Earth's atmosphere generate a shower of secondary particles that move toward the surface with relativistic speeds. Water Cherenkov Detectors (WCDs) on the ground can detect charged muons which are

one of the many particles generated in the shower, with Cherenkov Imaging technique. A large number of these muons travel in WCD tanks near the speed of light in vacuum, faster than the speed of light in water and so trigger isotropic Cherenkov radiation, which is detected by the Photomultiplier tubes (PMTs) placed inside the tanks. When the radial component of the speed of muon toward a PMT drops from superluminal to subluminal, it records Cherenkov light from an optical phenomenon known as Relativistic Image Doubling (RID), which causes two Cherenkov images of the same muon suddenly appear, with both images moving in geometrically opposite directions on the original muon track. The quantities associated with the RID effect can be measured experimentally with a variety of detector types and can be used to find various points on the original trajectory of the muon. In this paper, a detailed study of reconstructing the trajectory of a muon entering a Water Cherenkov Detector, using the RID technique has been presented. It is found that the measurements of standard RID observables enables a complete reconstruction of the trajectory of the muon to a high degree of accuracy with less than 1% error.

7.2 Introduction

When a single high energy particle like a gamma-ray photon strikes the top of the Earth's atmosphere, it produces a cascade of other particles which travel down toward the surface with relativistic speeds [257]. This air shower consisting of a variety

of charged and uncharged particles, also contains a charged muon whose lifetime is roughly 2.2 microseconds in its rest frame of reference [258], but owing to its relativistic speed, this lifetime is dilated in the frame of reference of the Earth. Therefore, the charged muons can reach the surface of the Earth without decaying.

These charged muons enter large tanks in Water Cherenkov Detectors (WCDs) that are made specifically to detect the former. The muons traveling faster than the speed of light in water, trigger Cherenkov radiation [259], which can be observed by a number of detectors inside the tanks.

Recently, an optical phenomenon known as Relativistic Image Doubling (RID) [260–263] has gained much attention. With RID, objects moving superluminally in a medium (faster than the medium speed of light) can appear twice simultaneously to an observer. When the radial speed of the muon toward a detector inside WCD drops from superluminal to subluminal, two bright Cherenkov images of the muon suddenly appear and diverge. This non-classical creation of images has been experimentally observed by Clerici et al. [264] wherein the authors investigated the kinematic effects linked with the superluminal motion of a light source using high temporal resolution imaging techniques and found image pair annihilation and creation when the speed of the source towards the observer dropped from superluminal to subluminal propagation regions. Furthermore, Velten et al. [265] experimentally observed temporal inversion effects using the light-in-flight (LiF) femto-photography and showed that in

the single image visualization of a video of a laser pulse traveling through a bottle of a specific liquid, the events could appear to happen at incorrect timings and can also appear in the wrong temporal order. This could also create effects that could seem to move superluminally. Faccio and Velten [266] provides a review of various time of flight distortions and relativistic effects observed by the light-in-flight photography techniques. The same RID effect has been hypothesized to help explain light curves in gamma-ray bursts [267]. Recently, RID effects have been suggested to be commonly found in images of air showers by Imaging Atmospheric Cherenkov Telescopes [268] and the Cherenkov images of the muon in Water Cherenkov Detectors [7].

In this work, it is shown how this unique and interesting optical phenomenon can be used to completely reconstruct muon tracks in ground-based Water Cherenkov Detectors (WCDs) like those deployed by Auger [269], HAWC [270], Kamiokande [271], and IceCube [272]. It is shown that the trajectory of a muon traveling inside a WCD with a constant velocity, can be completely reconstructed by extracting two points on its trajectory with at least three detectors observing an RID effect, given an estimate of an independent measurement of the muon velocity.

The paper is structured as follows. In Section 2, the conceptual basis for RIDs is reviewed briefly in relation to how they can be detected from inside the WCDs. For a detailed discussion of the mathematical framework behind the concepts, the reader is referred to Nemiroff and Kaushal [268] and Kaushal and Nemiroff [7]. In Section

3, the RID algorithm for the reconstruction of a muon track starting from the top of the tank and ending at the tank floor, is developed mathematically using systems of non-linear equations. In section 4, a simulated trajectory of a muon entering a WCD tank similar to the tanks used in the High-Altitude Water Cherenkov (HAWC) observatory, is reconstructed. It is assumed that a typical WCD is equipped with a PMT that records brightness as a function of time and a digital camera that records brightness as a function of angular position (or a video detector that records both the brightness of the muon track and its angular position with time). In Section 5, different methods of further constraining the particle trajectory using the RID algorithm and the advantages of this technique are discussed.

7.3 RID: A Brief Review

Several RID concepts discussed in this section are followed from Nemiroff and Kaushal [268]. Consider a cosmic ray muon traveling with a speed $v > c_w$, where c_w is the speed of light in water, and entering a WCD tank filled with water up to height H. It enters the top of the tank at time $t = 0$ through point A and leaves the bottom of the tank through point B. The muon is assumed to be at a constant speed v throughout the tank.

As soon as the muon enters the tank, it causes the emission of isotropic Cherenkov

radiation around its track in a cone, which is observed by the detectors placed at the bottom of the tank, as its "Cherenkov image". A detector inside the advanced Cherenkov cone of the muon will observe the phenomenon of Relativistic Image Doubling (RID). From the point of view of such a detector, the muon will first traverse a path starting from its entry point A (See Figure 7.1) in the tank down to a height z_C from the bottom of the tank, where the radial component of its speed toward the detector (v_r) equals the water speed of light (c_w). In this region of the track, the radial speed of the muon toward the detector is *faster* than the speed of the Cherenkov radiation it causes. So, this region of the muon track will be seen by the detector time-backwards i.e. the Cherenkov radiation emitted increasingly earlier along the muon track will reach the detector at increasingly later times. This happens because the emissions of the muon precede the muon itself in this region and therefore, this part of the track will appear to go *up* from height z_C along the track. After the muon has descended down past z_C, its radial speed toward the detector will be *slower* than its Cherenkov light, so this region of the muon track will appear normally to the detector i.e. the muon will appear to travel down towards the exit point monotonically with time. Therefore, the detector first observes the muon at height z_C on its track and *not* at the point of its entry in the tank. After the muon is first seen at z_C, it is simultaneously seen at two locations on its original track, one below and other above z_C.

Note that a detector outside the Cherenkov cone of the muon will not observe an RID

effect because the radial speed of the muon toward this detector is always subluminal, even though the total speed of the muon is always superluminal. For such a detector, the muon will appear to travel from the entry point A to the exit point B classically.

The detector is located at the floor of the tank at D, a distance L from the point of entry A and M from the point of exit B. The path length of the muon in the tank is given by $\frac{H}{\cos\theta}$ where θ is the angle between the muon path and the vertical. The height of the muon from the ground at any time t during its course in the tank is given by z. This is shown in Figure 7.1.

The time taken by the detector to observe the muon since the muon entered the tank is given by t_{total} which can be written as the sum of two times. The first is the time taken by the muon to descend to a height z from the ground, $t_{descend}$ and the second is the time taken by the light to reach from that location at height z to the detector, $t_{radiation}$. The "critical height" where the muon is first seen by the detector is given by z_C and it occurs at a time t_{min}, when t_{total} is a minimum. This can be found by solving $dt_{total}/dz = 0$ for z [268]. For a muon entering the tank from the top and leaving through the bottom, z_C is given by

$$z_C = H - \left(L\cos\alpha - \frac{c_w L\sin\alpha}{\sqrt{v^2 - c_w^2}}\right)\cos\theta, \tag{7.1}$$

where α is the angle between the detector and the muon track through point A.

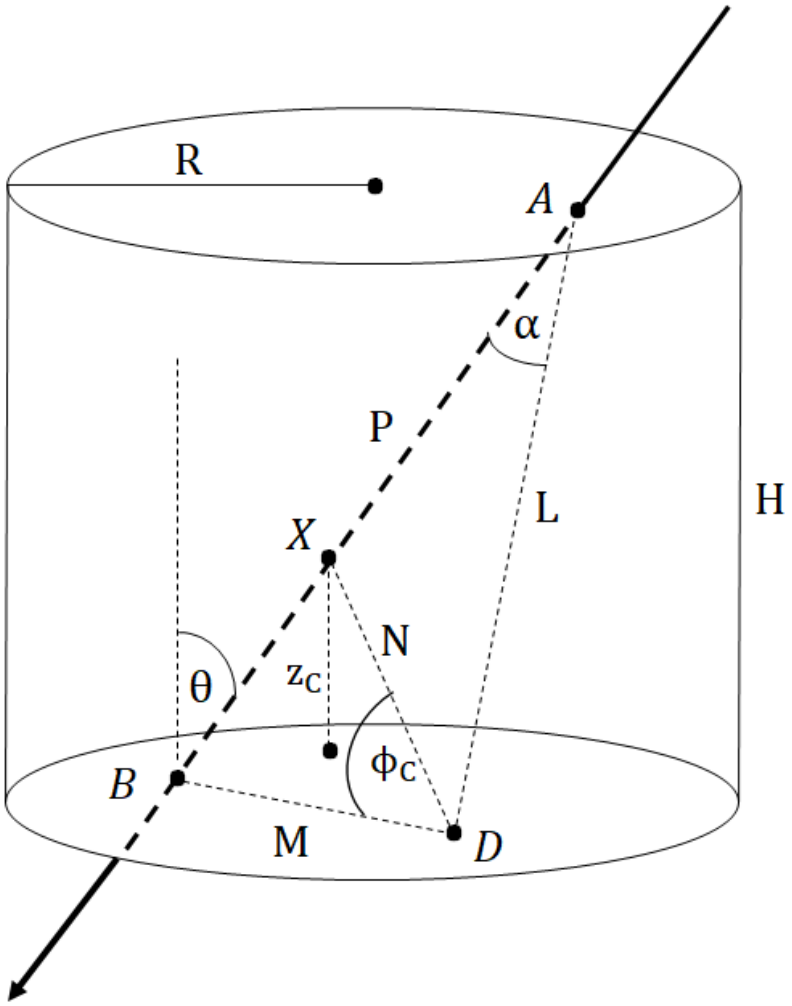

Figure 7.1: A muon enters the top of a WCD tank through A and leaves through the bottom at B. The path length of the muon inside the tank is given by P $(= H/\cos\theta)$ while the distances of the detector D from A and B are given by L and M respectively [7]. Figure taken from [8] under a CC BY license.

The "critical angle", ϕ_C, corresponding to the critical height z_C, where the muon is first seen by the detector, is the angle between the line joining the detector D with the point X of critical height on the muon path (i.e. DX) and the line joining the detector with the exit point B of muon at the WCD floor (i.e. BD). It is given by

$$\phi_C = \arccos\left(\frac{(P^2+L^2+M^2)\cos\theta + 2L\cos\alpha(z_C - P\cos\theta) - 2Pz_C}{2M\cos\theta\sqrt{L^2 - 2L\cos\alpha\left(P - \frac{z_C}{\cos\theta}\right) + \left(P - \frac{z_C}{\cos\theta}\right)^2}}\right). \quad (7.2)$$

where P is the path length of the muon inside the tank (See Figure 7.1).

After t_{min}, two images of the muon are observed simultaneously by the detector at heights $z_\pm$ from the ground and angular locations $\phi_\pm$ wrt the detector-exit point line (i.e. BD). Once the height and time of each image of the pair is known, their apparent brightness can be calculated using their transverse velocities [268].

7.4 Reconstruction of Muon Trajectory: Algorithm

Consider a muon entering the WCD tank from the top through point A with coordinates (x_A, y_A, H) and leaving from the bottom through point B with coordinates $(x_B, y_B, 0)$. The tank contains four detectors at the bottom arranged in a Y-pattern as shown in Figure 7.2. We only require a minimum of three detectors to develop the mathematical formulation for reconstruction of the trajectory. Consider detectors D_1, D_2 and D_3 at locations $(x_{D1}, y_{D1}, 0)$, $(x_{D2}, y_{D2}, 0)$ and $(x_{D3}, y_{D3}, 0)$ respectively.

These detectors are assumed to be inside the advanced Cherenkov cone of the muon and therefore, all three of them will observe an RID effect.

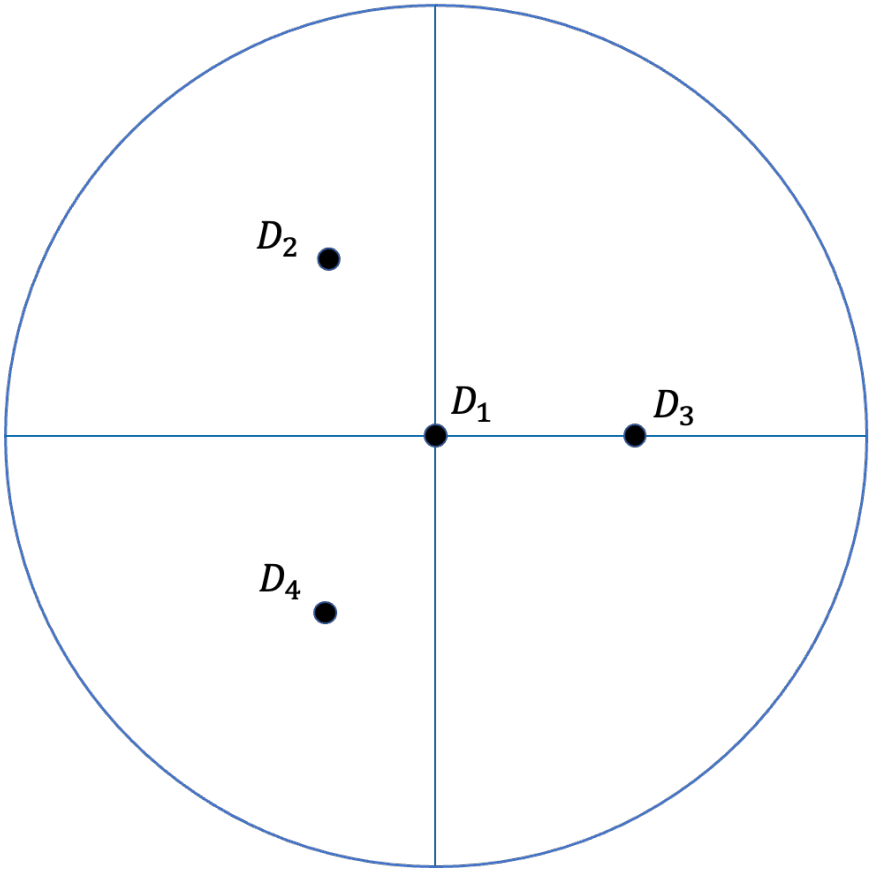

Figure 7.2: A top view of the WCD water tank. The three detectors D_2, D_3 and D_4 are arranged in an equilateral triangle with detector D_1 at the circumcenter of the triangle. Only detectors D_1, D_2 and D_3 will be considered for the reconstruction of the muon trajectory.

A detector here is considered to be a combination of a PMT which records the brightness and time of the muon trajectory and a video detector tracking both the angular position and brightness in time. Instead of a video detector, a digital camera could also be used as a static detector providing a record of brightness and angular location of the muon.

In order to completely reconstruct the trajectory of the particle, at least two points on its track need to be estimated. Consider the triangle ABD between a detector and the entry and exit points of the muon in the tank (Figure 7.1).

A system of 3 non-linear equations can be set up for each detector as follows,

$$\left.\begin{aligned} \sqrt{L^2 + M^2 - 2\ L\ M \cos\phi_i} &= \frac{H}{\cos\theta} \\ \sqrt{M^2 + N^2 - 2\ M\ N \cos\phi_C} &= \frac{z_C}{\cos\theta} \\ \sqrt{L^2 + N^2 - 2\ L\ N \cos(\phi_i - \phi_C)} &= \frac{H - z_C}{\cos\theta} \end{aligned}\right\} \tag{7.3}$$

where the critical height z_C, the critical angle ϕ_C and the angle between the entry and exit point of the muon through the detector, ϕ_i ($= \angle ADB$), are the RID observables that can be measured experimentally. The angle θ can be written in terms of lengths M and N and can be computed from the light curve of the muon and an independent measurement of its velocity. This will be calculated for the example case of a muon entering a HAWC-like WCD in the next section.

This system of equations (7.3) has a unique solution for the parameters L, M and N which are then evaluated for each pair of detectors i.e. $(L_i, M_i, N_i, L_j, M_j, N_j) \quad \forall \ (i,j) = (1,2), (1,3), (2,3)$.

Now, for any single detector system, say $D_1 - D_2$, a system of 4 non-linear equations

with 4 unknown variables (x_A, y_A, x_B, y_B) can be set up to find the coordinates of points A and B, as follows

$$\left.\begin{aligned} (x_A - x_{D1})^2 + (y_A - y_{D1})^2 + H^2 &= L_1^2 \\ (x_B - x_{D1})^2 + (y_B - y_{D1})^2 &= M_1^2 \\ (x_A - x_{D2})^2 + (y_A - y_{D2})^2 + H^2 &= L_2^2 \\ (x_B - x_{D2})^2 + (y_B - y_{D2})^2 &= M_2^2 \end{aligned}\right\} \tag{7.4}$$

Solving the above system of equations for one detector system gives a number of possible values of coordinates of A and B, because there are more than one A and B pairs that can have the same lengths L, M and N. The actual coordinates of A and B and thus the correct muon trajectory can be completely extracted by solving system (7.4) for the other two detector systems i.e. $D_1 - D_3$ and $D_2 - D_3$ as well and the actual coordinates are the ones that are common to all the detector pairs. In general, the more the detectors, the higher the precision on the constraints of the muon trajectory. With 4 detectors, 6 possible detector pairs can be formed that are sufficient to pinpoint the coordinates to a very high degree of precision.

7.5 Reconstruction of Muon Trajectory: Example

In this section, a simulated trajectory will be reconstructed for a muon incident on a HAWC-like WCD tank from the top at A and exiting through B, making some angle θ wrt the vertical. The light curve for the muon in this case is shown in Figure 7.3. The height of the water level in the tank, H, is 4.5 meters and the four detectors are placed at the bottom of the tank in a Y-pattern [273].

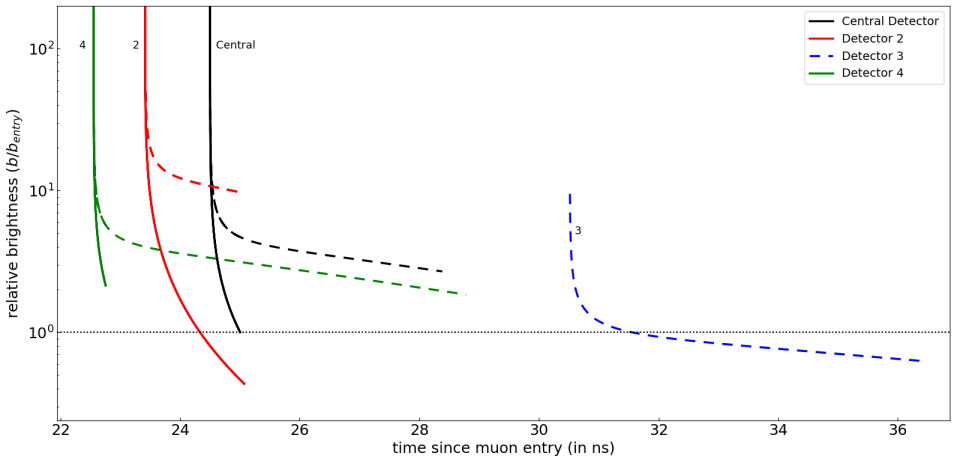

Figure 7.3: Light curve of a muon entering the WCD from the top and leaving through the bottom of the tank. The brightness on the y-axis is normalized wrt the brightness at the entry point as seen by the central detector. The dashed curve represents the Cherenkov image of the muon going towards the exit point B on the ground while the solid curve represents the image going up towards the entry point A. Figure taken from [8] under a CC BY license.

It can be seen from Figure 7.3 that only three of the four detectors will see an RID effect. The dashed curve represents that Cherenkov image of the muon which is going

downwards along the original muon track up to the exit point B while the solid curve represents the Cherenkov image going up towards the entry point A.

For a detector observing an RID, the total time duration of the dashed curve is given by

$$\begin{aligned} \Delta t_{dashed} &= \Big(\frac{AB}{v} + \frac{BD}{c}\Big) - \Big(\frac{AX}{v} + \frac{DX}{c}\Big) \\ &= \frac{z_C}{v\cos\theta} + \frac{M - X}{c} \end{aligned} \tag{7.5}$$

and therefore, the value of $\cos\theta$ is

$$\cos\theta = \frac{c\ z_C}{v\ (c\ \Delta t_{dashed} + X - M)} \tag{7.6}$$

Plots of heights and angular locations of the Cherenkov images of the muon versus the total time since the entry of muon in the tank, corresponding to the light curve shown in Figure 7.3, are shown in Figures 7.4 and 7.5. These plots are generated for the case of a muon incident at an angle using the RID code [274].

A number of parameters that can be derived from these plots and can be experimentally measured, along with their values for the example case considered in this

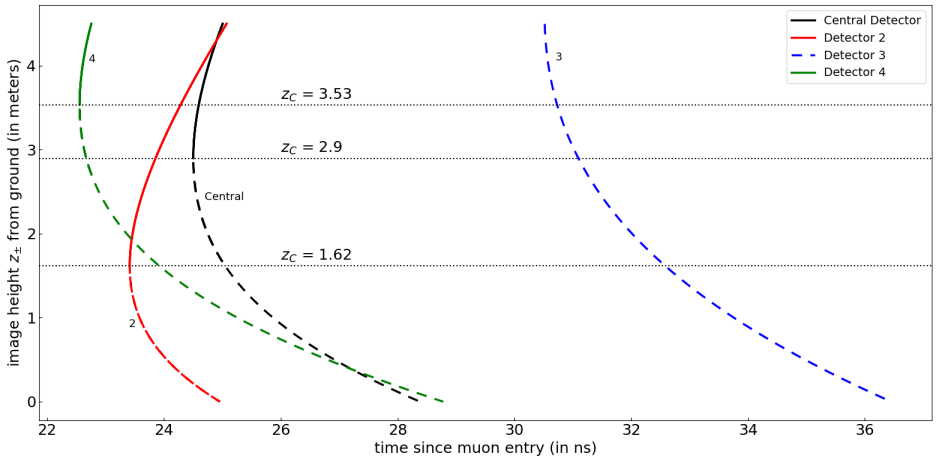

Figure 7.4: A graph of image heights versus the total time elapsed since the entry of muon in the tank. Note that for detector 3, there is no value of z_C inside the tank. Therefore, detector 3 is not inside the Cherenkov cone and will not observe an RID event. Figure taken from [8] under a CC BY license.

section, are listed in Table 7.1. The value of v can be obtained by an independent measurement of the Cherenkov angle of emission of the muon using standard methods employed in various Cherenkov detector systems. All the other parameters except the unknowns can be obtained by the measurement of RID observables.

Solving the system of equations (7.3) yields the values of L, M and N for the three detectors which represent the detector - entry point distance (D_iA), the detector - exit point distance (D_iB) and the detector - critical height distance (D_iX) respectively. These are shown in Table 7.2.

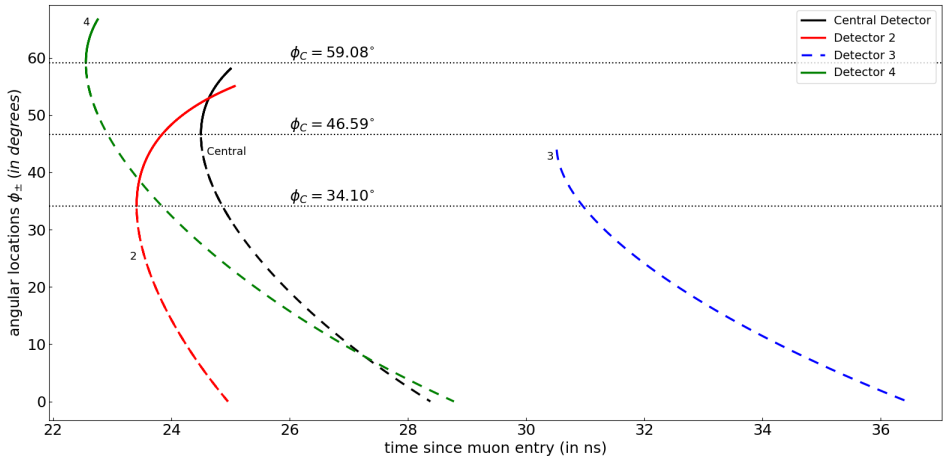

Figure 7.5: A plot of angular locations versus the total time for the Cherenkov images of the muon. Different detectors see the muon for the first time at different critical angles corresponding to different critical heights from the ground. Figure taken from [8] under a CC BY license.

The values of L and M for the three detectors are subsequently put in the system of equations (7.4) to obtain several sets of solutions for the coordinates of A and B of the muon representing the possible trajectories. The correct solutions is the one common to all the three detector pairs. The resulting coordinates of A and B are averaged over the three detector systems and are shown in Table 7.3. The significance of N is discussed in the next section.

It is seen from the table that the estimated values for the coordinates of the entry and exit point of the muon are very close to the actual values generated with simulations and the percentage error in extracting the coordinates is less than 1% in this

Table 7.1

A list of all parameters and their values for the example muon incidence of section 4. The objective is to find the unknown coordinates of the muon entry and exit points. The systems of non-linear equations (7.3) and (7.4) are solved to first obtain a measure of L, M and N for each detector and then the values of unknown parameters depicting the coordinates of A and B.

Parameter	**Value (SI units)**	**Description**
H	4.5	Height of the WCD tank
x_A, y_A, z_A	Unknown, Unknown, H	Coordinates of the muon entry point A
x_B, y_B, z_B	Unknown, Unknown, 0.0	Coordinates of the muon exit point B
x_{D1}, y_{D1}, z_{D1}	0.0, 0.0, 0.0	Coordinates of detector D_1
x_{D2}, y_{D2}, z_{D2}	-0.925, 1.602, 0.0	Coordinates of detector D_2
x_{D3}, y_{D3}, z_{D3}	1.85, 0.0, 0.0	Coordinates of detector D_3
z_{C1}	2.896	Critical height for detector D_1
z_{C2}	1.618	Critical height for detector D_2
z_{C3}	3.530	Critical height for detector D_3
ϕ_{C1}	0.813	Critical angle for detector D_1
ϕ_{C2}	0.595	Critical angle for detector D_2
ϕ_{C3}	1.031	Critical angle for detector D_3
ϕ_{i1}	1.014	Angle between the entry and exit point through detector D_1
ϕ_{i2}	0.961	Angle between the entry and exit point through detector D_2
ϕ_{i3}	1.164	Angle between the entry and exit point through detector D_3
Δt_{dashed_1}	3.88	Time for which the Cherenkov image going downwards from z_{C1} is visible to detector D_1
Δt_{dashed_2}	1.54	Time for which the Cherenkov image going downwards from z_{C2} is visible to detector D_2
Δt_{dashed_3}	6.23	Time for which the Cherenkov image going downwards from z_{C3} is visible to detector D_3
c_{vacuum}	3.0×10^8	Speed of light in vacuum
c_w	$c_{vacuum}/1.33$	Speed of light in water
v	c_{vacuum}	Speed of muon in water

case. Thus, the two points on the muon track are completely extracted with the measurements of observables of a typical RID phenomenon, which enables a complete reconstruction of the trajectory of the muon. The precision of the measurements of RID quantities, in turn, helps in further constraining this trajectory to a higher precision.

Table 7.2

Solution for the system of equations (7.3) for each of the three detectors. L is the distance between the detector and the muon entry point A, M is the distance between the detector and the muon exit point B, and N is the distance between the detector and the point X at critical height z_C where the muon is first observed by that detector. These values are fed to the system of equations (7.4) to extract the coordinates of the muon entry point A and exit point B.

Parameter	**Value**
L_1, M_1, N_1	5.6362, 2.7968, 4.2393
L_2, M_2, N_2	5.6460, 2.0217, 2.9679
L_3, M_3, N_3	5.1384, 2.8906, 4.3154

Table 7.3

The solution for the system of equations (7.4). Columns 2 to 4 are the coordinates of A and B obtained from each possible pair of detectors. Column 5 and 6 contain the mean values and the actual simulated values pertaining to the light curves of the muon trajectory respectively. Finally, the last column contains the percentage errors in results.

Results						
Parameter	$D_1 - D_2$	$D_2 - D_3$	$D_1 - D_3$	Mean	Actual Values	Error
x_A	2.3873	2.3746	2.3848	2.382	2.400	0.75%
y_A	2.4120	2.4244	2.4222	2.420	2.400	0.83%
x_B	0.7820	0.7808	0.7816	0.781	0.787	0.76%
y_B	2.6853	2.6856	2.6859	2.686	2.684	0.07%

The Cherenkov light from the muons is usually seen to be emitted into a forward cone with the direction of motion of the charged particle as the axis of the cone. The opening angle of the Cherenkov cone depends on the index of refraction of the medium. In water where the refractive index is 1.33, the opening angle of Cherenkov radiation is 41°. This angle is also calculated using the RID algorithm and comes out to be 41.22°, which is a fractional error of around 0.5% and demonstrates the

accuracy and reliability of this algorithm.

It is to be noted that in order for the detectors to observe the RID event, the critical height (z_C) for each detector should be less than the height of the water tank as the detector could only observe the events occurring inside the volume of the tank. It follows, therefore,

$$0 < z_{C_i} < H$$

$$\text{or,} \quad \frac{-P_i}{L_i} < \frac{\sin\alpha_i}{\sqrt{q^2-1}} - \cos\alpha_i < 0,$$

where $q = v/c_w > 1$ and i denotes the detector number. For the case where the muon enters the tank from the top and leaves from the bottom or vice-versa, this solves to,

$$\sec\alpha_i < q < \sqrt{\frac{\sec^4(\frac{\alpha_i}{2})\Big(L_i^2 + P^2 - 2L_iP\cos\alpha_i\Big)}{\Big((L_i+P)\tan^2(\frac{\alpha_i}{2}) - L_i + P\Big)^2}}, \quad \text{for} \quad P < L_i\cos\alpha_i, \quad \alpha_i \in (0, \frac{\pi}{2})$$

or,

$$\sec\alpha_i < q < \frac{M_i}{|P - L_i\cos\alpha_i|}, \quad \text{for} \quad P < L_i\cos\alpha_i, \quad \alpha_i \in (0, \frac{\pi}{2}) \tag{7.7}$$

and

$$\sec \alpha_i < q, \quad \text{for} \quad P \geq L_i \cos \alpha_i, \quad \alpha_i \in (0, \frac{\pi}{2}) \tag{7.8}$$

Inequalities 7.7 and 7.8 give the muon velocity constraints for which the detectors will observe the RID event and these constraints also agree with the simulations. Thus, the RID observations can also independently constrain the velocity of the muon and can be used as an additional method to check against the standard muon velocity estimation techniques currently used in the WCDs.

For a detector to observe RID, the distance (x) traveled by the muon after entering the tank should be less than the distance at which the RID event occurs. This distance depends on the interplay of velocity (v) of the muon and the time resolution of the detector. Given that the velocity of the muon entering the WCD tank follows some distribution, it is straightforward to calculate the time resolution that would provide the minimum spatial resolution on the muon path that could be observed by the detectors. Our simulations for the muon velocity used in Table 7.1 show that a detector having a lower time resolution than 10 ns (i.e. timesteps higher than 10 ns) will not be able to observe any RID event. This, though, does not mean that the RID event does not occur for that detector. It simply means that that particular detector will not be able to observe the RID event for lower resolutions, because the muon will

already be outside the tank before the detector measures the associated observable at the next time step. The system error of reconstruction of the muon trajectory is thus of the order of time resolution of the detector, which in our example case is 10 nanoseconds. This time resolution will be higher for muons traveling closer to (and obviously higher than) the speed of light in water than the ones traveling with speeds much higher than the speed of light in water. A detailed discussion on various detector types and their time resolutions is provided in the next section. On the other hand, as evident from inequalities 7.7 and 7.8, there exist upper and lower velocity bounds for particular muon trajectories such that for muons traveling with speeds outside those bounds, the detector will never observe the RID, no matter how high the resolution. This is due to the fact that the RID for these muon trajectories occur at locations that are outside the water tanks, thus inaccessible to be observed by the detectors.

7.6 Discussion

As we have seen, the complete trajectory of the muon can be reconstructed from the experimentally measured RID observables like the critical height z_C, the critical angle ϕ_C, etc. A minimum of three detectors observing RID are required for this reconstruction. To further constrain the muon trajectory to much precise values, the system of equations (7.3) can be solved for at least four detectors observing an RID

followed by solving a non-linear system of twelve equations given by,

$$\left.\begin{aligned} (x_A - x_{Di})^2 + (y_A - y_{Di})^2 + H^2 &= L_i^2 \\ (x_B - x_{Di})^2 + (y_B - y_{Di})^2 &= M_i^2 \\ (x_{Xi} - x_{Di})^2 + (y_{Xi} - y_{Di})^2 + z_{Ci}^2 &= N_i^2 \end{aligned}\right\} \quad \forall \;\; i = 1, 2, 3, 4 \tag{7.9}$$

where i represents a single detector. This system of equations, when solved within some error tolerance, will give 6 points on the path of the muon trajectory.

One of the many parameters involved in the working of the RID technique that can quantify the detectibility of the muon trajectory are θ (the angle between the muon trajectory and the vertical) and ϕ_i (the angle between the entry and exit point of the muon through the detector). The current methods of estimation of the muon trajectory in WCDs [275–278] use a completely different algorithm that also involves an independent estimation of the muon velocity in the WCD tank, which is another parameter in our reconstruction method. One of the assumptions of the RID technique is that this muon velocity is constant during the entire muon track across the WCD.

It should be noted that the RID method fails to work if no detector observes the RID event or if no more than 2 detectors observe the RID event. Though, It has been shown by Kaushal and Nemiroff [7] using two separate simulation algorithms, that of

all the muons entering the WCD, between 85% and 90% will be observed to trigger RID event by at least one detector, thus indicating that it should be very common for HAWC-like WCDs to observe RID events.

As the images of the muon after the RID event fade within a few nanoseconds, it might seem that these light curves are practically immeasurable. However, the increasing frame rate of capturing images, attributed to computer technology and miniaturization, have resulted in imagers that are able to capture sub-nanosecond events [264]. Firstly, there are hybrid pixel detectors which are fast time-stamping cameras sensitive to optical photons, such as MAPS-based PImMS-1 and PImMS-2 with a 12.5 nanoseconds resolution, a CMOS-based TimepixCam with a 10 nanosecond resolution and a Hybrid CMOS-based Tpx3Cam with a 1.6 nanosecond resolution (See Nomerotski [279] for a full review). Secondly, there are numerous Microchannel plate-based photomultiplier tubes (MCP-PMTs) whose time response can be as fast as 100 picoseconds FWHM with a gain of up to 10^7 [280]. To observe small events in close temporal proximity to much larger signals, the response of MCP-PMTs can be gated with an on/off ratio of up to 10^{13} in just 2 nanoseconds. It is worth noting that Hamamatsu PMTs (that are currently employed in HAWC detectors) models R3809U-(50,51,52,53) have the response time of 1.2 nanoseconds and can very easily observe the RID event duration, even in case of the fast dimming phase. These technological innovations raise the possibility of placing video detectors inside WCDs that can resolve RID events in both time and angle. Alternatively, simple digital

cameras may be placed that can resolve the Cerenkov images only in angle, leaving the temporal resolution to the PMTs.

Using the trajectory representing the fast dimming phase (or the muon image that seems to go along the original muon direction) is one of the two trajectories that can be independently used for the reconstruction. Another method to improve the estimation of the trajectory is to use the slow dimming trajectory i.e., the muon light curves for the Cherenkov image going upwards (solid curves in Figure 7.3). The same algorithm when applied to the Cherenkov images going upward, will give more sets of values for the coordinates of A and B, further decreasing the error in the final estimates, enabling a much more precise trajectory. This has not been done in this work as the goal of this work is to just introduce how the RID technique can be used for this reconstruction with the least amount of information that could possibly be extracted from the RID events.

The RID technique have a number of advantages over the traditional reconstruction methods currently employed in WCDs. First, only three detectors observing RID are needed to reconstruct the muon's trajectory unlike numerous PMTs currently employed in various WCD facilities around the world. Additional PMTs will only increase the precision of the reconstruction. Second, RID method can be used as an independent technique to constrain the muon velocities and can be used as a check

against the traditional methods of estimating muon velocities. Third, the RID reconstruction method is a much more generalized algorithm that can be employed in other Cherenkov detection principles (such as the Imaging Atmospheric Cherenkov Telescope (IACT) imaging and the Ring-imaging Cherenkov (RICH) detector systems) that usually employ different techniques of reconstruction after the detection of Cherenkov photons by the PMTs.

There are also some limitations and complexities associated with the RID technique. First, as the detector itself is an extended structure and not a point, its large size might result in the light travel time across its surface being significant when compared to the time taken by the Cherenkov light to reach the detector. Then any light curve that a PMT measures will convolve the size and shape of the PMT, not just the geometry inherent to the muon's path. Second, because RID events are observer-dependent even for the same muon trajectory, the locations of the detectors are very important. Simply adding together the brightness of different images from multiple PMTs, at the times of the brightness measurements, for example, will typically convolute RID effects beyond recognition. However, a careful reconstruction accounting for the timing of separate RID events as seen by different detectors should be possible that could enhance RID detection and better determine the muon's real track inside the WCD [268].

The RID algorithm can independently confirm the information about the muon's

trajectory, including the brightness along its path. When combined with the standard algorithms used at Water Cherenkov Detector systems for the reconstruction of Cherenkov cone, it can greatly constrain the muon trajectory with much smaller errors. This, in turn, can reduce the cost of construction, maintenance and working of a very large number of detectors usually deployed in such systems.

The same RID algorithm can also be used in Imaging Atmospheric Cherenkov Telescopes (IACTs) to reconstruct the trajectory of the secondary charged particles in the air showers. This can greatly reduce the number of telescopes used in the IACT systems and provide much better directional estimates of the shower.

In sum, the RID method is a novel reconstruction algorithm that provides a highly accurate, simple and effective technique to reconstruct the trajectories of muons in WCDs and can greatly reduce the number of detectors used at typical Water Cherenkov Detector systems.

7.7 Future Impacts

Various water Cherenkov detector systems around the world use thousands of Photomultiplier Tubes (11,129 PMTs in Super-Kamiokande detector in Japan, for example)

in order to detect the Cherenkov radiation emitted by muons from the cosmic air-showers entering the water tanks. Cherenkov detection is based on the simple fact that the greater the accuracy of reconstructing the air shower trajectory, the better the pinpointing of the source of the cosmic/gamma rays. This work presents an entirely new technique to reconstruct the same shower trajectory in a much simpler manner. It shows that no more than 3 PMTs observing RID are needed to extract the complete muon trajectory within a percent level accuracy, and increasing this number to 4,5 or 6 PMTs will only increase the accuracy of the reconstruction.delimits the manufacturing and installation of numerous highly expensive PMTs. This work can also serve as a standard theoretical check against the experimentally extracted muon trajectories in Water Cherenkov Detectors.

I believe that the RID reconstruction method may have the potential to change the way present WCD systems construct cosmic-ray trajectories with muons, by providing a very accurate, efficient, reliable and economically viable method to the WCD community. It is based on a very intuitive phenomenon and is aimed towards the next generation of Cherenkov detectors. If implemented, this technique will reduce the number of PMTs currently used in water-based Cherenkov detectors. This can further play a phenomenal role in the reconstruction of air showers as well by constraining the shower trajectory with Imaging Atmospheric Cherenkov Telescopes.

References

[1] Planck Collaboration, N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak, R. Battye, K. Benabed, J. P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J. F. Cardoso, J. Carron, A. Challinor, H. C. Chiang, J. Chluba, L. P. L. Colombo, C. Combet, D. Contreras, B. P. Crill, F. Cuttaia, P. de Bernardis, G. de Zotti, J. Delabrouille, J. M. Delouis, E. Di Valentino, J. M. Diego, O. Doré, M. Douspis, A. Ducout, X. Dupac, S. Dusini, G. Efstathiou, F. Elsner, T. A. Enßlin, H. K. Eriksen, Y. Fantaye, M. Farhang, J. Fergusson, R. Fernandez-Cobos, F. Finelli, F. Forastieri, M. Frailis, E. Franceschi, A. Frolov, S. Galeotta, S. Galli, K. Ganga, R. T. Génova-Santos, M. Gerbino, T. Ghosh, J. González-Nuevo, K. M. Górski, S. Gratton, A. Gruppuso, J. E. Gudmundsson, J. Hamann, W. Hand ley, D. Herranz, E. Hivon, Z. Huang, A. H. Jaffe, W. C. Jones, A. Karakci,

E. Keihänen, R. Keskitalo, K. Kiiveri, J. Kim, T. S. Kisner, L. Knox, N. Krachmalnicoff, M. Kunz, H. Kurki-Suonio, G. Lagache, J. M. Lamarre, A. Lasenby, M. Lattanzi, C. R. Lawrence, M. Le Jeune, P. Lemos, J. Lesgourgues, F. Levrier, A. Lewis, M. Liguori, P. B. Lilje, M. Lilley, V. Lindholm, M. López-Caniego, P. M. Lubin, Y. Z. Ma, J. F. Macías-Pérez, G. Maggio, D. Maino, N. Mandolesi, A. Mangilli, A. Marcos-Caballero, M. Maris, P. G. Martin, M. Martinelli, E. Martínez-González, S. Matarrese, N. Mauri, J. D. McEwen, P. R. Meinhold, A. Melchiorri, A. Mennella, M. Migliaccio, M. Millea, S. Mitra, M. A. Miville-Deschênes, D. Molinari, L. Montier, G. Morgante, A. Moss, P. Natoli, H. U. Nørgaard-Nielsen, L. Pagano, D. Paoletti, B. Partridge, G. Patanchon, H. V. Peiris, F. Perrotta, V. Pettorino, F. Piacentini, L. Polastri, G. Polenta, J. L. Puget, J. P. Rachen, M. Reinecke, M. Remazeilles, A. Renzi, G. Rocha, C. Rosset, G. Roudier, J. A. Rubiño-Martín, B. Ruiz-Granados, L. Salvati, M. Sandri, M. Savelainen, D. Scott, E. P. S. Shellard, C. Sirignano, G. Sirri, L. D. Spencer, R. Sunyaev, A. S. Suur-Uski, J. A. Tauber, D. Tavagnacco, M. Tenti, L. Toffolatti, M. Tomasi, T. Trombetti, L. Valenziano, J. Valiviita, B. Van Tent, L. Vibert, P. Vielva, F. Villa, N. Vittorio, B. D. Wand elt, I. K. Wehus, M. White, S. D. M. White, A. Zacchei, and A. Zonca. Planck 2018 results. VI. Cosmological parameters. *arXiv e-prints*, art. arXiv:1807.06209, Jul 2018.

[2] Kuan-Wei Huang, Tiziana Di Matteo, Aklant K. Bhowmick, Yu Feng, and

Chung-Pei Ma. BLUETIDES simulation: establishing black hole-galaxy relations at high redshift. , 478(4):5063–5073, August 2018. doi: 10.1093/mnras/sty1329.

[3] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Ana Maria Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, Antoine Brochard, Cora Uhlemann, Chi-Ting Chiang, Siyu He, Alice Pisani, Andrej Obuljen, Yu Feng, Emanuele Castorina, Gabriella Contardo, Christina D. Kreisch, Andrina Nicola, Justin Alsing, Roman Scoccimarro, Licia Verde, Matteo Viel, Shirley Ho, Stephane Mallat, Benjamin Wandelt, and David N. Spergel. The quijote simulations. *The Astrophysical Journal Supplement Series*, 250(1):2, aug 2020. doi: 10.3847/1538-4365/ab9d82. URL `https://doi.org/10.3847/1538-4365/ab9d82`.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL `https://arxiv.org/abs/1409.1556`.

[5] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Neerav Kaushal, Francisco Villaescusa-Navarro, Elena Giusarma, Yin Li, Conner Hawry, and Mauricio Reyes. NECOLA: Toward a Universal Field-level Cosmological Emulator. , 930(2):115, May 2022. doi: 10.3847/1538-4357/ac5c4a.

[7] N. Kaushal and R. J. Nemiroff. *ApJ*, 898:53, 2020. doi: 10.3847/1538-4357/ab98fa. URL `https://doi.org/10.3847/1538-4357/ab98fa`.

[8] Neerav Kaushal. A Novel RID Algorithm of Muon Trajectory Reconstruction in Water Cherenkov Detectors. , 936(2):120, September 2022. doi: 10.3847/1538-4357/ac8798.

[9] Planck Collaboration, Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A. J., Barreiro, R. B., Bartolo, N., Basak, S., Battye, R., Benabed, K., Bernard, J.-P., Bersanelli, M., Bielewicz, P., Bock, J. J., Bond, J. R., Borrill, J., Bouchet, F. R., Boulanger, F., Bucher, M., Burigana, C., Butler, R. C., Calabrese, E., Cardoso, J.-F., Carron, J., Challinor, A., Chiang, H. C., Chluba, J., Colombo, L. P. L., Combet, C., Contreras, D., Crill, B. P., Cuttaia, F., de Bernardis, P., de Zotti, G., Delabrouille, J., Delouis, J.-M., Di Valentino, E., Diego, J. M., Doré, O., Douspis, M., Ducout, A., Dupac, X., Dusini, S., Efstathiou, G., Elsner, F., Enßlin, T. A., Eriksen, H. K., Fantaye, Y., Farhang, M., Fergusson, J., Fernandez-Cobos, R., Finelli, F., Forastieri, F., Frailis, M., Fraisse, A. A., Franceschi, E., Frolov, A., Galeotta, S., Galli, S., Ganga, K., Génova-Santos, R. T., Gerbino, M., Ghosh, T., González-Nuevo, J., Górski, K. M., Gratton, S., Gruppuso, A., Gudmundsson, J. E., Hamann,

J., Handley, W., Hansen, F. K., Herranz, D., Hildebrandt, S. R., Hivon, E., Huang, Z., Jaffe, A. H., Jones, W. C., Karakci, A., Keihänen, E., Keskitalo, R., Kiiveri, K., Kim, J., Kisner, T. S., Knox, L., Krachmalnicoff, N., Kunz, M., Kurki-Suonio, H., Lagache, G., Lamarre, J.-M., Lasenby, A., Lattanzi, M., Lawrence, C. R., Le Jeune, M., Lemos, P., Lesgourgues, J., Levrier, F., Lewis, A., Liguori, M., Lilje, P. B., Lilley, M., Lindholm, V., López-Caniego, M., Lubin, P. M., Ma, Y.-Z., Macías-Pérez, J. F., Maggio, G., Maino, D., Mandolesi, N., Mangilli, A., Marcos-Caballero, A., Maris, M., Martin, P. G., Martinelli, M., Martínez-González, E., Matarrese, S., Mauri, N., McEwen, J. D., Meinhold, P. R., Melchiorri, A., Mennella, A., Migliaccio, M., Millea, M., Mitra, S., Miville-Deschênes, M.-A., Molinari, D., Montier, L., Morgante, G., Moss, A., Natoli, P., Nørgaard-Nielsen, H. U., Pagano, L., Paoletti, D., Partridge, B., Patanchon, G., Peiris, H. V., Perrotta, F., Pettorino, V., Piacentini, F., Polastri, L., Polenta, G., Puget, J.-L., Rachen, J. P., Reinecke, M., Remazeilles, M., Renzi, A., Rocha, G., Rosset, C., Roudier, G., Rubiño-Martín, J. A., Ruiz-Granados, B., Salvati, L., Sandri, M., Savelainen, M., Scott, D., Shellard, E. P. S., Sirignano, C., Sirri, G., Spencer, L. D., Sunyaev, R., Suur-Uski, A.-S., Tauber, J. A., Tavagnacco, D., Tenti, M., Toffolatti, L., Tomasi, M., Trombetti, T., Valenziano, L., Valiviita, J., Van Tent, B., Vibert, L., Vielva, P., Villa, F., Vittorio, N., Wandelt, B. D., Wehus, I. K., White, M., White, S. D. M., Zacchei, A., and Zonca, A. Planck 2018 results - vi. cosmological parameters.

A&A, 641:A6, 2020. doi: 10.1051/0004-6361/201833910. URL https://doi.org/10.1051/0004-6361/201833910.

[10] Stephen A. Shectman, Stephen D. Landy, Augustus Oemler, Douglas L. Tucker, Huan Lin, Robert P. Kirshner, and Paul L. Schechter. The Las Campanas Redshift Survey. , 470:172, October 1996. doi: 10.1086/177858.

[11] Edwin Hubble. A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae. *Proceedings of the National Academy of Science*, 15(3):168–173, March 1929. doi: 10.1073/pnas.15.3.168.

[12] A. Lewis, A. Challinor, and A. Lasenby. Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models. , 538:473–476, August 2000. doi: 10.1086/309179.

[13] Alan H. Guth and Kannan Jagannathan. The Inflationary Universe: The Quest for a New Theory of Cosmic Origins. *American Journal of Physics*, 66(1):94–95, January 1998. doi: 10.1119/1.18814.

[14] Alan H. Guth. The Big Bang and Cosmic Inflation. In Gosta Ekspong, editor, *The Oskar Klein Memorial Lectures: 1988-1999*, pages 159–206. 2014. doi: 10.1142/9789814571616_0012.

[15] Alan H. Guth. Eternal Inflation. *Annals of the New York Academy of Sciences*, 950(1):66–82, January 2001. doi: 10.1111/j.1749-6632.2001.tb02128.x.

[16] Alan H. Guth. The big bang and cosmic inflation. In *The Oskar Klein Memorial Lectures, Vol. 2*, pages 27–70. 1994. doi: 10.1142/9789814335911_0003.

[17] Ya. B. Zel'dovich. Gravitational instability: An approximate theory for large density perturbations. , 5:84–89, March 1970.

[18] E. R. Harrison. Fluctuations at the threshold of classical cosmology. *Phys. Rev. D*, 1:2726–2730, May 1970. doi: 10.1103/PhysRevD.1.2726. URL `https://link.aps.org/doi/10.1103/PhysRevD.1.2726`.

[19] G. F. Smoot, C. L. Bennett, A. Kogut, E. L. Wright, J. Aymon, N. W. Boggess, E. S. Cheng, G. de Amici, S. Gulkis, M. G. Hauser, G. Hinshaw, P. D. Jackson, M. Janssen, E. Kaita, T. Kelsall, P. Keegstra, C. Lineweaver, K. Loewenstein, P. Lubin, J. Mather, S. S. Meyer, S. H. Moseley, T. Murdock, L. Rokke, R. F. Silverberg, L. Tenorio, R. Weiss, and D. T. Wilkinson. Structure in the COBE Differential Microwave Radiometer First-Year Maps. , 396:L1, September 1992. doi: 10.1086/186504.

[20] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom, I. M. Hook, A. G. Kim, M. Y. Kim, J. C. Lee, N. J. Nunes, R. Pain, C. R. Pennypacker, R. Quimby, C. Lidman, R. S. Ellis, M. Irwin, R. G. McMahon, P. Ruiz-Lapuente, N. Walton, B. Schaefer, B. J. Boyle, A. V. Filippenko, T. Matheson, A. S. Fruchter, N. Panagia, H. J. M. Newberg, W. J. Couch, and The Supernova Cosmology

Project. Measurements of Ω and Λ from 42 High-Redshift Supernovae. , 517 (2):565–586, June 1999. doi: 10.1086/307221.

[21] V. de Lapparent, M. J. Geller, and J. P. Huchra. A Slice of the Universe. , 302: L1, March 1986. doi: 10.1086/184625.

[22] P. J. E. Peebles. *The large-scale structure of the universe*. 1980.

[23] R. Khatiwada, D. Bowring, A. S. Chou, A. Sonnenschein, W. Wester, D. V. Mitchell, T. Braine, C. Bartram, R. Cervantes, N. Crisosto, N. Du, L. J. Rosenberg, G. Rybka, J. Yang, D. Will, S. Kimes, G. Carosi, N. Woollett, S. Durham, L. D. Duffy, R. Bradley, C. Boutan, M. Jones, B. H. LaRoque, N. S. Oblath, M. S. Taubman, J. Tedeschi, John Clarke, A. Dove, A. Hashim, I. Siddiqi, N. Stevenson, A. Eddins, S. R. O'Kelley, S. Nawaz, A. Agrawal, A. V. Dixit, J. R. Gleason, S. Jois, P. Sikivie, N. S. Sullivan, D. B. Tanner, J. A. Solomon, E. Lentz, E. J. Daw, M. G. Perry, J. H. Buckley, P. M. Harrington, E. A. Henriksen, K. W. Murch, and G. C. Hilton. Axion Dark Matter Experiment: Detailed design and operations. *Review of Scientific Instruments*, 92(12):124502, December 2021. doi: 10.1063/5.0037857.

[24] C. Bartram, T. Braine, R. Cervantes, N. Crisosto, N. Du, G. Leum, L. J. Rosenberg, G. Rybka, J. Yang, D. Bowring, A. S. Chou, R. Khatiwada, A. Sonnenschein, W. Wester, G. Carosi, N. Woollett, L. D. Duffy, M. Goryachev, B. McAllister, M. E. Tobar, C. Boutan, M. Jones, B. H. LaRoque, N. S. Oblath, M. S.

Taubman, John Clarke, A. Dove, A. Eddins, S. R. O'Kelley, S. Nawaz, I. Siddiqi, N. Stevenson, A. Agrawal, A. V. Dixit, J. R. Gleason, S. Jois, P. Sikivie, J. A. Solomon, N. S. Sullivan, D. B. Tanner, E. Lentz, E. J. Daw, M. G. Perry, J. H. Buckley, P. M. Harrington, E. A. Henriksen, K. W. Murch, and ADMX Collaboration. Axion dark matter experiment: Run 1B analysis details. , 103 (3):032002, February 2021. doi: 10.1103/PhysRevD.103.032002.

[25] Kims Collaboration, H. S. Lee, H. Bhang, J. H. Choi, I. S. Hahn, D. He, M. J. Hwang, H. J. Kim, S. C. Kim, S. K. Kim, S. Y. Kim, T. Y. Kim, Y. D. Kim, J. W. Kwak, Y. J. Kwon, J. Lee, J. H. Lee, J. I. Lee, M. J. Lee, J. Li, S. S. Myung, H. Park, H. Y. Yang, and J. J. Zhu. First limit on WIMP cross section with low background CsI(Tℓ) crystal detector. *Physics Letters B*, 633(2-3): 201–208, February 2006. doi: 10.1016/j.physletb.2005.12.035.

[26] S. C. Kim, H. Bhang, J. H. Choi, W. G. Kang, B. H. Kim, H. J. Kim, K. W. Kim, S. K. Kim, Y. D. Kim, J. Lee, J. H. Lee, J. K. Lee, M. J. Lee, S. J. Lee, J. Li, J. Li, X. R. Li, Y. J. Li, S. S. Myung, S. L. Olsen, S. Ryu, I. S. Seong, J. H. So, and Q. Yue. New limits on interactions between weakly interacting massive particles and nucleons obtained with csi(tl) crystal detectors. *Phys. Rev. Lett.*, 108:181301, Apr 2012. doi: 10.1103/PhysRevLett.108.181301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.108.181301`.

[27] P. Agnes, T. Alexander, A. Alton, K. Arisaka, H. O. Back, B. Baldin,

K. Biery, G. Bonfini, M. Bossa, A. Brigatti, J. Brodsky, F. Budano, L. Cadonati, F. Calaprice, N. Canci, A. Candela, H. Cao, M. Cariello, P. Cavalcante, A. Chavarria, A. Chepurnov, A. G. Cocco, L. Crippa, D. D'Angelo, M. D'Incecco, S. Davini, M. De Deo, A. Derbin, A. Devoto, F. Di Eusanio, G. Di Pietro, E. Edkins, A. Empl, A. Fan, G. Fiorillo, K. Fomenko, G. Forster, D. Franco, F. Gabriele, C. Galbiati, A. Goretti, L. Grandi, M. Gromov, M. Y. Guan, Y. Guardincerri, B. Hackett, K. Herner, E. V. Hungerford, Al. Ianni, An. Ianni, C. Jollet, K. Keeter, C. Kendziora, S. Kidner, V. Kobychev, G. Koh, D. Korablev, G. Korga, A. Kurlej, P. X. Li, B. Loer, P. Lombardi, C. Love, L. Ludhova, S. Luitz, Y. Q. Ma, I. Machulin, A. Mandarano, S. Mari, J. Maricic, L. Marini, C. J. Martoff, A. Meregaglia, E. Meroni, P. D. Meyers, R. Milincic, D. Montanari, A. Monte, M. Montuschi, M. E. Monzani, P. Mosteiro, B. Mount, V. Muratova, P. Musico, A. Nelson, S. Odrowski, M. Okounkova, M. Orsini, F. Ortica, L. Pagani, M. Pallavicini, E. Pantic, L. Papp, S. Parmeggiano, R. Parsells, K. Pelczar, N. Pelliccia, S. Perasso, A. Pocar, S. Pordes, D. Pugachev, H. Qian, K. Randle, G. Ranucci, A. Razeto, B. Reinhold, A. Renshaw, A. Romani, B. Rossi, N. Rossi, S. D. Rountree, D. Sablone, P. Saggese, R. Saldanha, W. Sands, S. Sangiorgio, E. Segreto, D. Semenov, E. Shields, M. Skorokhvatov, O. Smirnov, A. Sotnikov, C. Stanford, Y. Suvorov, R. Tartaglia, J. Tatarowicz, G. Testera, A. Tonazzo, E. Unzhakov, R. B. Vogelaar, M. Wada, S. Walker, H. Wang, Y. Wang, A. Watson, S. Westerdale, M. Wojcik, A. Wright,

X. Xiang, J. Xu, C. G. Yang, J. Yoo, S. Zavatarelli, A. Zec, C. Zhu, and G. Zuzel. First results from the DarkSide-500.25emdark matter experiment at Laboratori Nazionali del Gran Sasso. *Physics Letters B*, 743:456–466, April 2015. doi: 10.1016/j.physletb.2015.03.012.

[28] P. Agnes, L. Agostino, I. F. M. Albuquerque, T. Alexander, A. K. Alton, K. Arisaka, H. O. Back, B. Baldin, K. Biery, G. Bonfini, M. Bossa, B. Bottino, A. Brigatti, J. Brodsky, F. Budano, S. Bussino, M. Cadeddu, L. Cadonati, M. Cadoni, F. Calaprice, N. Canci, A. Candela, H. Cao, M. Cariello, M. Carlini, S. Catalanotti, P. Cavalcante, A. Chepurnov, A. G. Cocco, G. Covone, L. Crippa, D. D'Angelo, M. D'Incecco, S. Davini, S. De Cecco, M. De Deo, M. De Vincenzi, A. Derbin, A. Devoto, F. Di Eusanio, G. Di Pietro, E. Edkins, A. Empl, A. Fan, G. Fiorillo, K. Fomenko, G. Forster, D. Franco, F. Gabriele, C. Galbiati, C. Giganti, A. M. Goretti, F. Granato, L. Grandi, M. Gromov, M. Guan, Y. Guardincerri, B. R. Hackett, J. Hall, K. Herner, P. H. Humble, E. V. Hungerford, Al. Ianni, An. Ianni, I. James, C. Jollet, K. Keeter, C. L. Kendziora, V. Kobychev, G. Koh, D. Korablev, G. Korga, A. Kubankin, X. Li, M. Lissia, P. Lombardi, S. Luitz, Y. Ma, I. N. Machulin, A. Mandarano, S. M. Mari, J. Maricic, L. Marini, C. J. Martoff, A. Meregaglia, P. D. Meyers, T. Miletic, R. Milincic, D. Montanari, A. Monte, M. Montuschi, M. Monzani, P. Mosteiro, B. J. Mount, V. N. Muratova, P. Musico, J. Napolitano, A. Nelson, S. Odrowski, M. Orsini, F. Ortica, L. Pagani, M. Pallavicini, E. Pantic,

S. Parmeggiano, K. Pelczar, N. Pelliccia, S. Perasso, A. Pocar, S. Pordes, D. A. Pugachev, H. Qian, K. Randle, G. Ranucci, A. Razeto, B. Reinhold, A. L. Renshaw, A. Romani, B. Rossi, N. Rossi, D. Rountree, D. Sablone, P. Saggese, R. Saldanha, W. Sands, S. Sangiorgio, C. Savarese, E. Segreto, D. A. Semenov, E. Shields, P. N. Singh, M. D. Skorokhvatov, O. Smirnov, A. Sotnikov, C. Stanford, Y. Suvorov, R. Tartaglia, J. Tatarowicz, G. Testera, A. Tonazzo, P. Trinchese, E. V. Unzhakov, A. Vishneva, B. Vogelaar, M. Wada, S. Walker, H. Wang, Y. Wang, A. W. Watson, S. Westerdale, J. Wilhelmi, M. M. Wojcik, X. Xiang, J. Xu, C. Yang, J. Yoo, S. Zavatarelli, A. Zec, W. Zhong, C. Zhu, and G. Zuzel. Results from the first use of low radioactivity argon in a dark matter search. *Phys. Rev. D*, 93:081101, Apr 2016. doi: 10.1103/PhysRevD.93.081101. URL `https://link.aps.org/doi/10.1103/PhysRevD.93.081101`.

[29] Marc Kamionkowski, Licia Verde, and Raul Jimenez. The void abundance with non-gaussian primordial perturbations. , 2009(1):010, January 2009. doi: 10.1088/1475-7516/2009/01/010.

[30] P. J. E. Peebles. The Void Phenomenon. , 557(2):495–504, August 2001. doi: 10.1086/322254.

[31] Sunny Vagnozzi, Elena Giusarma, Olga Mena, Katherine Freese, Martina Gerbino, Shirley Ho, and Massimiliano Lattanzi. Unveiling mml:math xmlns:mml="http://www.w3.org/1998/math/MathML"

display="inline"mml:mi/mml:mi/mml:math secrets with cosmological data: Neutrino masses and mass hierarchy. *Physical Review D*, 96(12), dec 2017. doi: 10.1103/physrevd.96.123503. URL https://doi.org/10.1103%2Fphysrevd.96.123503.

[32] Mark Vogelsberger, Federico Marinacci, Paul Torrey, and Ewald Puchwein. Cosmological simulations of galaxy formation. *Nature Reviews Physics*, 2(1): 42–66, January 2020. doi: 10.1038/s42254-019-0127-2.

[33] J. S. Bagla. Cosmological n-body simulation: Techniques, scope and status. 2004. doi: 10.48550/ARXIV.ASTRO-PH/0411043. URL https://arxiv.org/abs/astro-ph/0411043.

[34] J. Richard Bond, Lev Kofman, and Dmitry Pogosyan. How filaments of galaxies are woven into the cosmic web. , 380(6575):603–606, April 1996. doi: 10.1038/380603a0.

[35] Jörg P. Dietrich, Norbert Werner, Douglas Clowe, Alexis Finoguenov, Tom Kitching, Lance Miller, and Aurora Simionescu. A filament of dark matter between two clusters of galaxies. , 487(7406):202–204, July 2012. doi: 10.1038/nature11224.

[36] Volker Springel. The cosmological simulation code GADGET-2. , 364(4):1105–1134, December 2005. doi: 10.1111/j.1365-2966.2005.09655.x.

[37] Antony Lewis, Anthony Challinor, and Anthony Lasenby. Efficient Computation of Cosmic Microwave Background Anisotropies in Closed Friedmann-Robertson-Walker Models. , 538(2):473–476, August 2000. doi: 10.1086/309179.

[38] Ya. B. Zel'dovich. Gravitational instability: An approximate theory for large density perturbations. , 5:84–89, March 1970.

[39] Svetlin Tassev, Matias Zaldarriaga, and Daniel J Eisenstein. Solving large scale structure in ten easy steps with COLA. *Journal of Cosmology and Astroparticle Physics*, 2013(06):036–036, jun 2013. doi: 10.1088/1475-7516/2013/06/036. URL `https://doi.org/10.1088/1475-7516/2013/06/036`.

[40] C. Howlett, M. Manera, and W. J. Percival. L-PICOLA: A parallel code for fast dark matter simulation. *Astronomy and Computing*, 12:109–126, September 2015. doi: 10.1016/j.ascom.2015.07.003.

[41] Bill S. Wright, Hans A. Winther, and Kazuya Koyama. COLA with massive neutrinos. , 2017(10):054, October 2017. doi: 10.1088/1475-7516/2017/10/054.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition

Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[44] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Conference on Fairness, Accountability, and Transparency*, 2020. doi: 10.1145/3351095.3375709.

[45] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and Analysis of a Large Scale Image Ontology. Vision Sciences Society, 2009.

[46] Russell Reed and Robert J. Marks II. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, 1999*. Bradford Books, 1999.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

[48] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL `https://arxiv.org/abs/1406.2661`.

[49] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv e-prints*, art. arXiv:1411.1784, November 2014.

[50] F. Bernardeau, S. Colombi, E. Gaztañaga, and R. Scoccimarro. Large-scale structure of the Universe and cosmological perturbation theory. , 367(1-3): 1–248, September 2002. doi: 10.1016/S0370-1573(02)00135-7.

[51] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Ana Maria Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, Antoine Brochard, Cora Uhlemann, Chi-Ting Chiang, Siyu He, Alice Pisani, Andrej Obuljen, Yu Feng, Emanuele Castorina, Gabriella Contardo, Christina D. Kreisch, Andrina Nicola, Justin Alsing, Roman Scoccimarro, Licia Verde, Matteo Viel, Shirley Ho, Stephane Mallat, Benjamin Wandelt, and David N. Spergel. The Quijote Simulations. , 250(1): 2, September 2020. doi: 10.3847/1538-4365/ab9d82.

[52] Lado Samushia, Zachary Slepian, and Francisco Villaescusa-Navarro. Information Content of Higher-Order Galaxy Correlation Functions. *arXiv e-prints*, art. arXiv:2102.01696, February 2021.

[53] Davide Gualdi, Hector Gil-Marin, and Licia Verde. Joint analysis of anisotropic power spectrum, bispectrum and trispectrum: application to N-body simulations. *arXiv e-prints*, art. arXiv:2104.03976, April 2021.

[54] Joseph Kuruvilla and Nabila Aghanim. Information content in mean pairwise velocity and mean relative velocity between pairs in a triplet. *arXiv e-prints*, art. arXiv:2102.06709, February 2021.

[55] Adrian E. Bayer, Francisco Villaescusa-Navarro, Elena Massara, Jia Liu, David N. Spergel, Licia Verde, Benjamin Wandelt, Matteo Viel, and Shirley Ho. Detecting neutrino mass by combining matter clustering, halos, and voids, 2021.

[56] Arka Banerjee, Emanuele Castorina, Francisco Villaescusa-Navarro, Travis Court, and Matteo Viel. Weighing neutrinos with the halo environment. , 2020(6):032, June 2020. doi: 10.1088/1475-7516/2020/06/032.

[57] ChangHoon Hahn, Francisco Villaescusa-Navarro, Emanuele Castorina, and Roman Scoccimarro. Constraining M_ν with the bispectrum. Part I. Breaking parameter degeneracies. , 2020(3):040, March 2020. doi: 10.1088/1475-7516/2020/03/040.

[58] Cora Uhlemann, Oliver Friedrich, Francisco Villaescusa-Navarro, Arka Banerjee, and Sandrine Codis. Fisher for complements: extracting cosmology and neutrino mass from the counts-in-cells PDF. , 495(4):4006–4027, July 2020. doi: 10.1093/mnras/staa1155.

[59] Oliver Friedrich, Cora Uhlemann, Francisco Villaescusa-Navarro, Tobias Baldauf, Marc Manera, and Takahiro Nishimichi. Primordial non-Gaussianity without tails - how to measure f_{NL} with the bulk of the density PDF. , 498(1): 464–483, October 2020. doi: 10.1093/mnras/staa2160.

[60] Elena Massara, Francisco Villaescusa-Navarro, Shirley Ho, Neal Dalal, and David N. Spergel. Using the Marked Power Spectrum to Detect the Signature of Neutrinos in Large-Scale Structure. , 126(1):011301, January 2021. doi: 10.1103/PhysRevLett.126.011301.

[61] Ji-Ping Dai, Licia Verde, and Jun-Qing Xia. What can we learn by combining the skew spectrum and the power spectrum? , 2020(8):007, August 2020. doi: 10.1088/1475-7516/2020/08/007.

[62] E. Allys, T. Marchand, J. F. Cardoso, F. Villaescusa-Navarro, S. Ho, and S. Mallat. New interpretable statistics for large-scale structure analysis and generation. , 102(10):103506, November 2020. doi: 10.1103/PhysRevD.102.103506.

[63] Arka Banerjee and Tom Abel. Nearest neighbour distributions: New statistical measures for cosmological clustering. , 500(4):5479–5499, January 2021. doi: 10.1093/mnras/staa3604.

[64] Arka Banerjee and Tom Abel. Cosmological cross-correlations and nearest neighbour distributions. , 504(2):2911–2923, June 2021. doi: 10.1093/mnras/stab961.

[65] Davide Gualdi, Sergi Novell, Héctor Gil-Marín, and Licia Verde. Matter trispectrum: theoretical modelling and comparison to N-body simulations. , 2021(1): 015, January 2021. doi: 10.1088/1475-7516/2021/01/015.

[66] Utkarsh Giri and Kendrick M. Smith. Exploring KSZ velocity reconstruction with N-body simulations and the halo model. *arXiv e-prints*, art. arXiv:2010.07193, October 2020.

[67] Lucia F. de la Bella, Nicolas Tessore, and Sarah Bridle. The unequal-time matter power spectrum: impact on weak lensing observables. *arXiv e-prints*, art. arXiv:2011.06185, November 2020.

[68] ChangHoon Hahn and Francisco Villaescusa-Navarro. Constraining M_ν with the bispectrum. Part II. The information content of the galaxy bispectrum monopole. , 2021(4):029, April 2021. doi: 10.1088/1475-7516/2021/04/029.

[69] Georgios Valogiannis and Cora Dvorkin. Towards an Optimal Estimation of Cosmological Parameters with the Wavelet Scattering Transform. *arXiv e-prints*, art. arXiv:2108.07821, August 2021.

[70] Joseph Kuruvilla. Cosmology with the kinetic Sunyaev-Zeldovich effect: Independent of the optical depth and σ_8. *arXiv e-prints*, art. arXiv:2109.13938, September 2021.

[71] Lehman H. Garrison, Daniel J. Eisenstein, Douglas Ferrer, Jeremy L. Tinker, Philip A. Pinto, and David H. Weinberg. The Abacus Cosmos: A Suite of Cosmological N-body Simulations. *Astrophys. J. Suppl.*, 236(2):43, 2018. doi: 10.3847/1538-4365/aabfd3.

[72] Tomoaki Ishiyama, Francisco Prada, Anatoly A Klypin, Manodeep Sinha, R Benton Metcalf, Eric Jullo, Bruno Altieri, Sofía A Cora, Darren Croton, Sylvain de la Torre, and et al. The uchuu simulations: Data release 1 and dark matter halo concentrations. *Monthly Notices of the Royal Astronomical Society*, 506(3):4210–4231, Jun 2021. ISSN 1365-2966. doi: 10.1093/mnras/stab1755. URL http://dx.doi.org/10.1093/mnras/stab1755.

[73] Raul E Angulo, Matteo Zennaro, Sergio Contreras, Giovanni Aricò, Marcos Pellejero-Ibañez, and Jens Stücker. The bacco simulation project: exploiting the full power of large-scale structure for cosmology. *Monthly Notices of the Royal Astronomical Society*, 507(4):5869–5881, Jul 2021. ISSN 1365-2966. doi: 10.1093/mnras/stab2018. URL http://dx.doi.org/10.1093/mnras/stab2018.

[74] Joseph DeRose, Risa H. Wechsler, Jeremy L. Tinker, Matthew R. Becker, Yao-Yuan Mao, Thomas McClintock, Sean McLaughlin, Eduardo Rozo, and Zhongxu Zhai. The AEMULUS Project. I. Numerical Simulations for Precision Cosmology. , 875(1):69, Apr 2019. doi: 10.3847/1538-4357/ab1085.

[75] Thomas McClintock, Eduardo Rozo, Matthew R. Becker, Joseph DeRose, Yau-Yuan Mao, Sean McLaughlin, Jeremy L. Tinker, Risa H. Wechsler, and Zhongxu Zhai. The Aemulus Project II: Emulating the Halo Mass Function. *Astrophys. J.*, 872(1):53, 2019. doi: 10.3847/1538-4357/aaf568.

[76] Zhongxu Zhai, Jeremy L. Tinker, Matthew R. Becker, Joseph DeRose, Yao-Yuan Mao, Thomas McClintock, Sean McLaughlin, Eduardo Rozo, and Risa H. Wechsler. The Aemulus Project III: Emulation of the Galaxy Correlation Function. *Astrophys. J.*, 874(1):95, 2019. doi: 10.3847/1538-4357/ab0d7b.

[77] Thomas McClintock, Eduardo Rozo, Arka Banerjee, Matthew R. Becker, Joseph DeRose, Sean McLaughlin, Jeremy L. Tinker, Risa H. Wechsler, and Zhongxu Zhai. The aemulus project iv: Emulating halo bias, 2019.

[78] Nina A. Maksimova, Lehman H. Garrison, Daniel J. Eisenstein, Boryana Hadzhiyska, Sownak Bose, and Thomas P. Satterthwaite. ABACUSSUMMIT: A Massive Set of High-Accuracy, High-Resolution N-Body Simulations. , September 2021. doi: 10.1093/mnras/stab2484.

[79] Takahiro Nishimichi, Masahiro Takada, Ryuichi Takahashi, Ken Osato, Masato Shirasaki, Taira Oogi, Hironao Miyatake, Masamune Oguri, Ryoma Murata, Yosuke Kobayashi, and Naoki Yoshida. Dark Quest. I. Fast and Accurate Emulation of Halo Clustering Statistics and Its Application to Galaxy Clustering. , 884(1):29, October 2019. doi: 10.3847/1538-4357/ab3719.

[80] Francisco-Shu Kitaura and Steffen Heß. Cosmological structure formation with augmented Lagrangian perturbation theory. *Monthly Notices of the Royal Astronomical Society: Letters*, 435(1):L78–L82, 08 2013. ISSN 1745-3925. doi: 10.1093/mnrasl/slt101. URL `https://doi.org/10.1093/mnrasl/slt101`.

[81] R. Scoccimarro and R. K. Sheth. PTHALOS: a fast method for generating mock galaxy distributions. , 329:629–640, January 2002. doi: 10.1046/j.1365-8711.2002.04999.x.

[82] Pierluigi Monaco, Tom Theuns, and Giuliano Taffoni. The pinocchio algorithm: pinpointing orbit-crossing collapsed hierarchical objects in a linear density field. *Monthly Notices of the Royal Astronomical Society*, 331(3):587–608, 04 2002. ISSN 0035-8711. doi: 10.1046/j.1365-8711.2002.05162.x. URL `https://doi.org/10.1046/j.1365-8711.2002.05162.x`.

[83] Y. Feng, M.-Y. Chu, and U. Seljak. FastPM: a new scheme for fast simulations of dark matter and halos. *ArXiv e-prints*, March 2016.

[84] Svetlin Tassev, Matias Zaldarriaga, and Daniel J Eisenstein. Solving large scale structure in ten easy steps with COLA. *Journal of Cosmology and Astroparticle Physics*, 2013(06):036–036, jun 2013. doi: 10.1088/1475-7516/2013/06/036. URL `https://doi.org/10.1088`.

[85] Svetlin Tassev, Daniel J. Eisenstein, Benjamin D. Wandelt, and Matias Zaldarriaga. scola: The n-body cola method extended to the spatial domain, 2015.

[86] C.-H. Chuang, F.-S. Kitaura, F. Prada, C. Zhao, and G. Yepes. EZmocks: extending the Zel'dovich approximation to generate mock galaxy catalogues with accurate clustering statistics. , 446:2621–2628, January 2015. doi: 10.1093/mnras/stu2301.

[87] Chirag Modi, Francois Lanusse, and Uros Seljak. FlowPM: Distributed TensorFlow Implementation of the FastPM Cosmological N-body Solver. *arXiv e-prints*, art. arXiv:2010.11847, October 2020.

[88] F.-S. Kitaura, G. Yepes, and F. Prada. Modelling baryon acoustic oscillations with perturbation theory and stochastic halo biasing. , 439:L21–L25, March 2014. doi: 10.1093/mnrasl/slt172.

[89] Peter Coles and Bernard Jones. A lognormal model for the cosmological mass distribution. , 248:1–13, January 1991. doi: 10.1093/mnras/248.1.1.

[90] Aniket Agrawal, Ryu Makiya, Chi-Ting Chiang, Donghui Jeong, Shun Saito, and Eiichiro Komatsu. Generating log-normal mock catalog of galaxies in redshift space. *Journal of Cosmology and Astroparticle Physics*, 2017(10): 003–003, Oct 2017. ISSN 1475-7516. doi: 10.1088/1475-7516/2017/10/003. URL `http://dx.doi.org/10.1088/1475-7516/2017/10/003`.

[91] S. Avila, S. G. Murray, A. Knebe, C. Power, A. S. G. Robotham, and J. Garcia-Bellido. Halogen: a tool for fast generation of mock halo catalogues. *Monthly Notices of the Royal Astronomical Society*, 450(2):1856–1867, Apr 2015. ISSN 1365-2966. doi: 10.1093/mnras/stv711. URL `http://dx.doi.org/10.1093/mnras/stv711`.

[92] Federico Tosone, Mark C Neyrinck, Benjamin R Granett, Luigi Guzzo, and Nicola Vittorio. muscle-ups: improved approximations of the matter field with

the extended Press–Schechter formalism and Lagrangian perturbation theory. *Monthly Notices of the Royal Astronomical Society*, 505(2):2999–3015, 05 2021. ISSN 0035-8711. doi: 10.1093/mnras/stab1517. URL https://doi.org/10.1093/mnras/stab1517.

[93] Martin White, Jeremy L. Tinker, and Cameron K. McBride. Mock galaxy catalogues using the quick particle mesh method. *Monthly Notices of the Royal Astronomical Society*, 437(3):2594–2606, 11 2013. ISSN 0035-8711. doi: 10.1093/mnras/stt2071. URL https://doi.org/10.1093/mnras/stt2071.

[94] Philippe Berger and George Stein. A volumetric deep convolutional neural network for simulation of mock dark matter halo catalogues. *Monthly Notices of the Royal Astronomical Society*, 482(3):2861–2871, Nov 2018. ISSN 1365-2966. doi: 10.1093/mnras/sty2949. URL http://dx.doi.org/10.1093/mnras/sty2949.

[95] J. R. Bond and S. T. Myers. The Peak-Patch Picture of Cosmic Catalogs. I. Algorithms. , 103:1, March 1996. doi: 10.1086/192267.

[96] George Stein, Marcelo A. Alvarez, and J. Richard Bond. The mass-Peak Patch algorithm for fast generation of deep all-sky dark matter halo catalogues and its N-body validation. , 483(2):2236–2250, February 2019. doi: 10.1093/mnras/sty3226.

[97] Yin Li, Yueying Ni, Rupert A. C. Croft, Tiziana Di Matteo, Simeon Bird,

and Yu Feng. Ai-assisted superresolution cosmological simulations. *Proceedings of the National Academy of Sciences*, 118(19), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2022038118. URL https://www.pnas.org/content/118/19/e2022038118.

[98] Yueying Ni, Yin Li, Patrick Lachance, Rupert A C Croft, Tiziana Di Matteo, Simeon Bird, and Yu Feng. Ai-assisted superresolution cosmological simulations – ii. halo substructures, velocities, and higher order statistics. *Monthly Notices of the Royal Astronomical Society*, 507(1):1021–1033, Jul 2021. ISSN 1365-2966. doi: 10.1093/mnras/stab2113. URL http://dx.doi.org/10.1093/mnras/stab2113.

[99] Siyu He, Yin Li, Yu Feng, Shirley Ho, Siamak Ravanbakhsh, Wei Chen, and Barnabás Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Science*, 116(28):13825–13832, July 2019. doi: 10.1073/pnas.1821458116.

[100] Renan Alves de Oliveira, Yin Li, Francisco Villaescusa-Navarro, Shirley Ho, and David N. Spergel. Fast and Accurate Non-Linear Predictions of Universes with Deep Learning. *arXiv e-prints*, art. arXiv:2012.00240, November 2020.

[101] F. Bernardeau, S. Colombi, E. Gaztañaga, and R. Scoccimarro. Large-scale structure of the Universe and cosmological perturbation theory. , 367:1–248, September 2002. doi: 10.1016/S0370-1573(02)00135-7.

[102] Renan Alves de Oliveira, Yin Li, Francisco Villaescusa-Navarro, Shirley Ho, and David N. Spergel. Fast and Accurate Non-Linear Predictions of Universes with Deep Learning. *arXiv e-prints*, art. arXiv:2012.00240, November 2020.

[103] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation, 2016.

[104] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[105] Katrin Heitmann, David Higdon, Martin White, Salman Habib, Brian J. Williams, Earl Lawrence, and Christian Wagner. The Coyote Universe. II. Cosmological Models and Precision Emulation of the Nonlinear Matter Power Spectrum. , 705(1):156–174, November 2009. doi: 10.1088/0004-637X/705/1/156.

[106] Katrin Heitmann, Martin White, Christian Wagner, Salman Habib, and David Higdon. The Coyote Universe. I. Precision Determination of the Nonlinear Matter Power Spectrum. , 715(1):104–121, May 2010. doi: 10.1088/0004-637X/715/1/104.

[107] Earl Lawrence, Katrin Heitmann, Martin White, David Higdon, Christian Wagner, Salman Habib, and Brian Williams. The Coyote Universe. III. Simulation Suite and Precision Emulator for the Nonlinear Matter Power Spectrum. , 713 (2):1322–1331, April 2010. doi: 10.1088/0004-637X/713/2/1322.

[108] Katrin Heitmann, Earl Lawrence, Juliana Kwan, Salman Habib, and David Higdon. The Coyote Universe Extended: Precision Emulation of the Matter Power Spectrum. , 780(1):111, January 2014. doi: 10.1088/0004-637X/780/1/111.

[109] Katrin Heitmann, Derek Bingham, Earl Lawrence, Steven Bergner, Salman Habib, David Higdon, Adrian Pope, Rahul Biswas, Hal Finkel, Nicholas Frontiere, and et al. The mira–titan universe: Precision predictions for dark energy surveys. *The Astrophysical Journal*, 820(2):108, Mar 2016. ISSN 1538-4357. doi: 10.3847/0004-637x/820/2/108. URL http://dx.doi.org/10.3847/0004-637X/820/2/108.

[110] Earl Lawrence, Katrin Heitmann, Juliana Kwan, Amol Upadhye, Derek Bingham, Salman Habib, David Higdon, Adrian Pope, Hal Finkel, and Nicholas Frontiere. The Mira-Titan Universe. II. Matter Power Spectrum Emulation. , 847(1):50, September 2017. doi: 10.3847/1538-4357/aa86a9.

[111] Elena Massara, Francisco Villaescusa-Navarro, and Matteo Viel. The halo model in a massive neutrino cosmology. , 2014(12):053, December 2014. doi: 10.1088/1475-7516/2014/12/053.

[112] F. Villaescusa-Navarro, F. Marulli, M. Viel, E. Branchini, E. Castorina, E. Sefusatti, and S. Saito. Cosmology with massive neutrinos I: towards a realistic

modeling of the relation between matter, haloes and galaxies. , 3:011, March 2014. doi: 10.1088/1475-7516/2014/03/011.

[113] E. Castorina, E. Sefusatti, R. K. Sheth, F. Villaescusa-Navarro, and M. Viel. Cosmology with massive neutrinos II: on the universality of the halo mass function and bias. , 2:049, February 2014. doi: 10.1088/1475-7516/2014/02/049.

[114] Elena Giusarma, Mauricio Reyes Hurtado, Francisco Villaescusa-Navarro, Siyu He, Shirley Ho, and ChangHoon Hahn. Learning neutrino effects in Cosmology with Convolutional Neural Networks. *arXiv e-prints*, art. arXiv:1910.04255, October 2019.

[115] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[116] Jens Jasche and Benjamin D. Wandelt. Methods for Bayesian Power Spectrum

Inference with Galaxy Surveys. , 779(1):15, December 2013. doi: 10.1088/0004-637X/779/1/15.

[117] Jens Jasche and Benjamin D. Wandelt. Bayesian physical reconstruction of initial conditions from large-scale structure surveys. *Monthly Notices of the Royal Astronomical Society*, 432(2):894–913, 04 2013. ISSN 0035-8711. doi: 10.1093/mnras/stt449. URL `https://doi.org/10.1093/mnras/stt449`.

[118] Elena Giusarma, Mauricio Reyes Hurtado, Francisco Villaescusa-Navarro, Siyu He, Shirley Ho, and ChangHoon Hahn. Learning neutrino effects in Cosmology with Convolutional Neural Networks. *arXiv e-prints*, art. arXiv:1910.04255, October 2019.

[119] Euclid, Euclid. `http://sci.esa.int/euclid`.

[120] Dark Energy Spectroscopic Instrument, DESI. `https://www.desi.lbl.gov`.

[121] D. Spergel, N. Gehrels, C. Baltay, D. Bennett, J. Breckinridge, M. Donahue, A. Dressler, B. S. Gaudi, T. Greene, O. Guyon, C. Hirata, J. Kalirai, N. J. Kasdin, B. Macintosh, W. Moos, S. Perlmutter, M. Postman, B. Rauscher, J. Rhodes, Y. Wang, D. Weinberg, D. Benford, M. Hudson, W. S. Jeong, Y. Mellier, W. Traub, T. Yamada, P. Capak, J. Colbert, D. Masters, M. Penny, D. Savransky, D. Stern, N. Zimmerman, R. Barry, L. Bartusek, K. Carpenter, E. Cheng, D. Content, F. Dekens, R. Demers, K. Grady, C. Jackson, G. Kuan, J. Kruk, M. Melton, B. Nemati, B. Parvin, I. Poberezhskiy, C. Peddie, J. Ruffa,

J. K. Wallace, A. Whipple, E. Wollack, and F. Zhao. Wide-Field InfrarRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report. *arXiv e-prints*, art. arXiv:1503.03757, Mar 2015.

[122] The LSST Dark Energy Science Collaboration, Rachel Mandelbaum, Tim Eifler, Renée Hložek, Thomas Collett, Eric Gawiser, Daniel Scolnic, David Alonso, Humna Awan, Rahul Biswas, Jonathan Blazek, Patricia Burchat, Nora Elisa Chisari, Ian Dell'Antonio, Seth Digel, Josh Frieman, Daniel A. Goldstein, Isobel Hook, Željko Ivezić, Steven M. Kahn, Sowmya Kamath, David Kirkby, Thomas Kitching, Elisabeth Krause, Pierre-François Leget, Philip J. Marshall, Joshua Meyers, Hironao Miyatake, Jeffrey A. Newman, Robert Nichol, Eli Rykoff, F. Javier Sanchez, Anže Slosar, Mark Sullivan, and M. A. Troxel. The LSST Dark Energy Science Collaboration (DESC) Science Requirements Document. *arXiv e-prints*, art. arXiv:1809.01669, September 2018.

[123] K. N. Abazajian, P. Adshead, Z. Ahmed, S. W. Allen, D. Alonso, K. S. Arnold, C. Baccigalupi, J. G. Bartlett, N. Battaglia, B. A. Benson, C. A. Bischoff, J. Borrill, V. Buza, E. Calabrese, R. Caldwell, J. E. Carlstrom, C. L. Chang, T. M. Crawford, F.-Y. Cyr-Racine, F. De Bernardis, T. de Haan, S. di Serego Alighieri, J. Dunkley, C. Dvorkin, J. Errard, G. Fabbian, S. Feeney, S. Ferraro, J. P. Filippini, R. Flauger, G. M. Fuller, V. Gluscevic, D. Green, D. Grin, E. Grohs, J. W. Henning, J. C. Hill, R. Hlozek, G. Holder, W. Holzapfel, W. Hu, K. M. Huffenberger, R. Keskitalo, L. Knox, A. Kosowsky, J. Kovac, E. D.

Kovetz, C.-L. Kuo, A. Kusaka, M. Le Jeune, A. T. Lee, M. Lilley, M. Loverde, M. S. Madhavacheril, A. Mantz, D. J. E. Marsh, J. McMahon, P. D. Meerburg, J. Meyers, A. D. Miller, J. B. Munoz, H. N. Nguyen, M. D. Niemack, M. Peloso, J. Peloton, L. Pogosian, C. Pryke, M. Raveri, C. L. Reichardt, G. Rocha, A. Rotti, E. Schaan, M. M. Schmittfull, D. Scott, N. Sehgal, S. Shandera, B. D. Sherwin, T. L. Smith, L. Sorbo, G. D. Starkman, K. T. Story, A. van Engelen, J. D. Vieira, S. Watson, N. Whitehorn, and W. L. Kimmy Wu. CMB-S4 Science Book, First Edition. *ArXiv e-prints*, October 2016.

[124] Simeon Bird, Matteo Viel, and Martin G. Haehnelt. Massive neutrinos and the non-linear matter power spectrum. *Monthly Notices of the Royal Astronomical Society*, 420(3):2551–2561, 02 2012. ISSN 0035-8711. doi: 10.1111/j.1365-2966.2011.20222.x. URL `https://doi.org/10.1111/j.1365-2966.2011.20222.x`.

[125] Julien Lesgourgues and Sergio Pastor. Neutrino mass from Cosmology. *arXiv e-prints*, art. arXiv:1212.6154, December 2012.

[126] F. Villaescusa-Navarro, M. Vogelsberger, M. Viel, and A. Loeb. Neutrino signatures on the high-transmission regions of the Lyman α forest. , 431:3670–3677, Jun 2013. doi: 10.1093/mnras/stt452.

[127] Marco Peloso, Massimo Pietroni, Matteo Viel, and Francisco Villaescusa-Navarro. The effect of massive neutrinos on the bao peak, 2015. URL `https://arxiv.org/abs/1505.07477`.

[128] E. Castorina, C. Carbone, J. Bel, E. Sefusatti, and K. Dolag. DEMNUni: the clustering of large-scale structures in the presence of massive neutrinos. , 7:043, July 2015. doi: 10.1088/1475-7516/2015/07/043.

[129] Elena Massara, Francisco Villaescusa-Navarro, Matteo Viel, and P. M. Sutter. Voids in massive neutrino cosmologies. *JCAP*, 1511(11):018, 2015. doi: 10.1088/1475-7516/2015/11/018.

[130] Francisco Villaescusa-Navarro, Arka Banerjee, Neal Dalal, Emanuele Castorina, Roman Scoccimarro, Raul Angulo, and David N. Spergel. The imprint of neutrinos on clustering in redshift space. *The Astrophysical Journal*, 861(1):53, jul 2018. doi: 10.3847/1538-4357/aac6bf. URL `https://doi.org/10.3847%2F1538-4357%2Faac6bf`.

[131] J. Bel, A. Pezzotta, C. Carbone, E. Sefusatti, and L. Guzzo. Accurate fitting functions for peculiar velocity spectra in standard and massive-neutrino cosmologies. *Astronomy &amp Astrophysics*, 622:A109, feb 2019. doi: 10.1051/0004-6361/201834513. URL `https://doi.org/10.1051%2F0004-6361%2F201834513`.

[132] Andres C. Rodríguez, Tomasz Kacprzak, Aurelien Lucchi, Adam Amara, Raphaël Sgier, Janis Fluri, Thomas Hofmann, and Alexandre Réfrégier. Fast cosmic web simulations with generative adversarial networks. *Computational Astrophysics and Cosmology*, 5(1):4, November 2018. doi: 10.1186/

s40668-018-0026-4.

[133] Andrius Tamosiunas, Hans A. Winther, Kazuya Koyama, David J. Bacon, Robert C. Nichol, and Ben Mawdsley. Investigating cosmological GAN emulators using latent space interpolation. , 506(2):3049–3067, September 2021. doi: 10.1093/mnras/stab1879.

[134] Nathanaël Perraudin, Sandro Marcon, Aurelien Lucchi, and Tomasz Kacprzak. Emulation of cosmological mass maps with conditional generative adversarial networks. *arXiv e-prints*, art. arXiv:2004.08139, April 2020.

[135] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. URL `https://arxiv.org/abs/1701.07875`.

[136] Lilian Weng. From gan to wgan, 2019. URL `https://arxiv.org/abs/1904.08994`.

[137] Katrin Collaboration, M. Aker, A. Beglarian, J. Behrens, A. Berlev, U. Besserer, B. Bieringer, F. Block, S. Bobien, M. Böttcher, B. Bornschein, L. Bornschein, T. Brunst, T. S. Caldwell, R. M. D. Carney, L. La Cascio, S. Chilingaryan, W. Choi, K. Debowski, M. Deffert, M. Descher, D. Díaz Barrero, P. J. Doe, O. Dragoun, G. Drexlin, K. Eitel, E. Ellinger, R. Engel, S. Enomoto, A. Felden, J. A. Formaggio, F. M. Fränkle, G. B. Franklin, F. Friedel, A. Fulst, K. Gauda, W. Gil, F. Glück, R. Grössle, R. Gumbsheimer, V. Gupta, T. Höhn, V. Hannen, N. Haußmann, K. Helbing, S. Hickford, R. Hiller, D. Hillesheimer, D. Hinz,

T. Houdy, A. Huber, A. Jansen, C. Karl, F. Kellerer, J. Kellerer, M. Kleifges, M. Klein, C. Köhler, L. Köllenberger, A. Kopmann, M. Korzeczek, A. Kovalík, B. Krasch, H. Krause, N. Kunka, T. Lasserre, T. L. Le, O. Lebeda, B. Lehnert, A. Lokhov, M. Machatschek, E. Malcherek, M. Mark, A. Marsteller, E. L. Martin, C. Melzer, A. Menshikov, S. Mertens, J. Mostafa, K. Müller, H. Neumann, S. Niemes, P. Oelpmann, D. S. Parno, A. W. P. Poon, J. M. L. Poyato, F. Priester, S. Ramachandran, R. G. H. Robertson, W. Rodejohann, M. Röllig, C. Röttele, C. Rodenbeck, M. Ryšavý, R. Sack, A. Saenz, P. Schäfer, A. Schaller Née Pollithy, L. Schimpf, K. Schlösser, M. Schlösser, L. Schlüter, S. Schneidewind, M. Schrank, B. Schulz, A. Schwemmer, M. Šefčík, V. Sibille, D. Siegmann, M. Slezák, F. Spanier, M. Steidl, M. Sturm, M. Sun, D. Tcherniakhovski, H. H. Telle, L. A. Thorne, T. Thümmler, N. Titov, I. Tkachev, K. Urban, K. Valerius, D. Vénos, A. P. Vizcaya Hernández, C. Weinheimer, S. Welte, J. Wendel, J. F. Wilkerson, J. Wolf, S. Wüstling, J. Wydra, W. Xu, Y. R. Yen, S. Zadoroghny, and G. Zeller. Direct neutrino-mass measurement with sub-electronvolt sensitivity. *Nature Physics*, 18(2):160–166, February 2022. doi: 10.1038/s41567-021-01463-1.

[138] Aurel Schneider, Romain Teyssier, Doug Potter, Joachim Stadel, Julian Onions, Darren S. Reed, Robert E. Smith, Volker Springel, Frazer R. Pearce, and Roman Scoccimarro. Matter power spectrum and the challenge of percent accuracy. *Journal of Cosmology and Astroparticle Physics*, 2016(04):047–047, apr

2016. doi: 10.1088/1475-7516/2016/04/047. URL https://doi.org/10.1088%2F1475-7516%2F2016%2F04%2F047.

[139] M. Gatti, C. Chang, O. Friedrich, B. Jain, D. Bacon, M. Crocce, J. DeRose, I. Ferrero, P. Fosalba, E. Gaztanaga, D. Gruen, I. Harrison, N. Jeffrey, N. MacCrann, T. McClintock, L. Secco, L. Whiteway, T. M. C. Abbott, S. Allam, J. Annis, S. Avila, D. Brooks, E. Buckley-Geer, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, R. Cawthon, L. N. da Costa, J. De Vicente, S. Desai, H. T. Diehl, P. Doel, T. F. Eifler, J. Estrada, S. Everett, A. E. Evrard, J. Frieman, J. García-Bellido, D. W. Gerdes, R. A. Gruendl, J. Gschwend, G. Gutierrez, D. J. James, M. D. Johnson, E. Krause, K. Kuehn, M. Lima, M. A. G. Maia, M. March, J. L. Marshall, P. Melchior, F. Menanteau, R. Miquel, A. Palmese, F. Paz-Chinchón, A. A. Plazas, C. Sánchez, E. Sanchez, V. Scarpine, M. Schubnell, S. Santiago, I. Sevilla-Noarbe, M. Smith, M. Soares-Santos, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, M. A. Troxel, J. Zuntz, J. Zuntz, and DES Collaboration. Dark Energy Survey Year 3 results: cosmology with moments of weak lensing mass maps - validation on simulations. , 498(3):4060–4087, November 2020. doi: 10.1093/mnras/staa2680.

[140] T. Kacprzak, D. Kirk, O. Friedrich, A. Amara, A. Refregier, L. Marian, J. P. Dietrich, E. Suchyta, J. Aleksić, D. Bacon, M. R. Becker, C. Bonnett, S. L. Bridle, C. Chang, T. F. Eifler, W. G. Hartley, E. M. Huff, E. Krause, N. MacCrann, P. Melchior, A. Nicola, S. Samuroff, E. Sheldon, M. A. Troxel, J. Weller,

J. Zuntz, T. M. C. Abbott, F. B. Abdalla, R. Armstrong, A. Benoit-Lévy, G. M. Bernstein, R. A. Bernstein, E. Bertin, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, F. J. Castander, M. Crocce, C. B. D'Andrea, L. N. da Costa, S. Desai, H. T. Diehl, A. E. Evrard, A. Fausti Neto, B. Flaugher, P. Fosalba, J. Frieman, D. W. Gerdes, D. A. Goldstein, D. Gruen, R. A. Gruendl, G. Gutierrez, K. Honscheid, B. Jain, D. J. James, M. Jarvis, K. Kuehn, N. Kuropatkin, O. Lahav, M. Lima, M. March, J. L. Marshall, P. Martini, C. J. Miller, R. Miquel, J. J. Mohr, R. C. Nichol, B. Nord, A. A. Plazas, A. K. Romer, A. Roodman, E. S. Rykoff, E. Sanchez, V. Scarpine, M. Schubnell, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, F. Sobreira, M. E. C. Swanson, G. Tarle, D. Thomas, V. Vikram, A. R. Walker, Y. Zhang, and DES Collaboration. Cosmology constraints from shear peak statistics in Dark Energy Survey Science Verification data. , 463(4):3653–3673, December 2016. doi: 10.1093/mnras/stw2070.

[141] Mustafa Mustafa, Deborah Bard, Wahid Bhimji, Zarija Lukić, Rami Al-Rfou, and Jan M. Kratochvil. CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks. *Computational Astrophysics and Cosmology*, 6(1):1, May 2019. doi: 10.1186/s40668-019-0029-9.

[142] Nicolas Martinet, Peter Schneider, Hendrik Hildebrandt, HuanYuan Shan, Marika Asgari, Jörg P. Dietrich, Joachim Harnois-Dé raps, Thomas Erben,

Aniello Grado, Catherine Heymans, Henk Hoekstra, Dominik Klaes, Konrad Kuijken, Julian Merten, and Reiko Nakajima. KiDS-450: cosmological constraints from weak-lensing peak statistics – II: Inference from shear peaks using n-body simulations. *Monthly Notices of the Royal Astronomical Society*, 474(1):712–730, oct 2017. doi: 10.1093/mnras/stx2793. URL https://doi.org/10.1093%2Fmnras%2Fstx2793.

[143] Richard M. Feder, Philippe Berger, and George Stein. Nonlinear 3d cosmic web simulation with heavy-tailed generative adversarial networks. *Physical Review D*, 102(10), nov 2020. doi: 10.1103/physrevd.102.103504. URL https://doi.org/10.1103%2Fphysrevd.102.103504.

[144] Jorit Schmelzle, Aurelien Lucchi, Tomasz Kacprzak, Adam Amara, Raphael Sgier, Alexandre Réfrégier, and Thomas Hofmann. Cosmological model discrimination with deep learning, 2017. URL https://arxiv.org/abs/1707.05167.

[145] Andrea Petri, Zoltán Haiman, Lam Hui, Morgan May, and Jan M. Kratochvil. Cosmology with minkowski functionals and moments of the weak lensing convergence field. *Phys. Rev. D*, 88:123002, Dec 2013. doi: 10.1103/PhysRevD.88.123002. URL https://link.aps.org/doi/10.1103/PhysRevD.88.123002.

[146] Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, David N. Spergel, Yin Li, Benjamin Wandelt, Andrina Nicola, Leander Thiele, Sultan Hassan, Jose Manuel Zorrilla Matilla, Desika Narayanan, Romeel Dave, and

Mark Vogelsberger. Multifield Cosmology with Artificial Intelligence. *arXiv e-prints*, art. arXiv:2109.09747, September 2021.

[147] Francisco Villaescusa-Navarro, Shy Genel, Daniel Angles-Alcazar, David N. Spergel, Yin Li, Benjamin Wandelt, Leander Thiele, Andrina Nicola, Jose Manuel Zorrilla Matilla, Helen Shao, Sultan Hassan, Desika Narayanan, Romeel Dave, and Mark Vogelsberger. Robust marginalization of baryonic effects for cosmological inference at the field level. *arXiv e-prints*, art. arXiv:2109.10360, September 2021.

[148] Siamak Ravanbakhsh, Junier Oliva, Sebastien Fromenteau, Layne C. Price, Shirley Ho, Jeff Schneider, and Barnabas Poczos. Estimating Cosmological Parameters from the Dark Matter Distribution. *arXiv e-prints*, art. arXiv:1711.02033, Nov 2017.

[149] Alex Cole, Benjamin Kurt Miller, Samuel J. Witte, Maxwell X. Cai, Meiert W. Grootes, Francesco Nattino, and Christoph Weniger. Fast and Credible Likelihood-Free Cosmology with Truncated Marginal Neural Ratio Estimation. *arXiv e-prints*, art. arXiv:2111.08030, November 2021.

[150] Michelle Ntampaka, Daniel J. Eisenstein, Sihan Yuan, and Lehman H. Garrison. A Hybrid Deep Learning Approach to Cosmological Constraints from Galaxy Redshift Surveys. , 889(2):151, February 2020. doi: 10.3847/1538-4357/ab5f5e.

[151] J. E. G. Peek and Blakesley Burkhart. Do Androids Dream of Magnetic Fields? Using Neural Networks to Interpret the Turbulent Interstellar Medium. , 882 (1):L12, September 2019. doi: 10.3847/2041-8213/ab3a9e.

[152] Tumelo Mangena, Sultan Hassan, and Mario G. Santos. Constraining the reionization history using deep learning from 21-cm tomography with the Square Kilometre Array. , 494(1):600–606, May 2020. doi: 10.1093/mnras/staa750.

[153] Sultan Hassan, Sambatra Andrianomena, and Caitlin Doughty. Constraining the astrophysics and cosmology from 21 cm tomography using deep learning with the SKA. , 494(4):5761–5774, June 2020. doi: 10.1093/mnras/staa1151.

[154] Doogesh Kodi Ramanah, Tom Charnock, Francisco Villaescusa-Navarro, and Benjamin D. Wandelt. Super-resolution emulator of cosmological simulations using deep physical models. , 495(4):4227–4236, May 2020. doi: 10.1093/mnras/staa1428.

[155] Yin Li, Yueying Ni, Rupert A. C. Croft, Tiziana Di Matteo, Simeon Bird, and Yu Feng. AI-assisted superresolution cosmological simulations. *Proceedings of the National Academy of Science*, 118(19):2022038118, May 2021. doi: 10.1073/pnas.2022038118.

[156] Yueying Ni, Yin Li, Patrick Lachance, Rupert A. C. Croft, Tiziana Di Matteo, Simeon Bird, and Yu Feng. AI-assisted super-resolution cosmological simulations II: Halo substructures, velocities and higher order statistics. *arXiv*

e-prints, art. arXiv:2105.01016, May 2021.

[157] Sultan Hassan, Francisco Villaescusa-Navarro, Benjamin Wandelt, David N. Spergel, Daniel Anglés-Alcázar, Shy Genel, Miles Cranmer, Greg L. Bryan, Romeel Davé, Rachel S. Somerville, Michael Eickenberg, Desika Narayanan, Shirley Ho, and Sambatra Andrianomena. HIFlow: Generating Diverse HI Maps Conditioned on Cosmology using Normalizing Flow. *arXiv e-prints*, art. arXiv:2110.02983, October 2021.

[158] Dongwon Han, Neelima Sehgal, and Francisco Villaescusa-Navarro. MillimeterDL: Deep Learning Simulations of the Microwave Sky. *arXiv e-prints*, art. arXiv:2105.11444, May 2021.

[159] Juan Zamudio-Fernandez, Atakan Okan, Francisco Villaescusa-Navarro, Seda Bilaloglu, Asena Derin Cengiz, Siyu He, Laurence Perreault Levasseur, and Shirley Ho. HIGAN: Cosmic Neutral Hydrogen with Generative Adversarial Networks. *arXiv e-prints*, art. arXiv:1904.12846, April 2019.

[160] Xinyue Zhang, Yanfang Wang, Wei Zhang, Yueqiu Sun, Siyu He, Gabriella Contardo, Francisco Villaescusa-Navarro, and Shirley Ho. From Dark Matter to Galaxies with Convolutional Networks. *arXiv e-prints*, art. arXiv:1902.05965, February 2019.

[161] Jacky H. T. Yip, Xinyue Zhang, Yanfang Wang, Wei Zhang, Yueqiu Sun, Gabriella Contardo, Francisco Villaescusa-Navarro, Siyu He, Shy Genel, and

Shirley Ho. From Dark Matter to Galaxies with Convolutional Neural Networks. *arXiv e-prints*, art. arXiv:1910.07813, October 2019.

[162] Noah Kasmanoff, Francisco Villaescusa-Navarro, Jeremy Tinker, and Shirley Ho. dm2gal: Mapping Dark Matter to Galaxies with Neural Networks. *arXiv e-prints*, art. arXiv:2012.00186, November 2020.

[163] Digvijay Wadekar and Roman Scoccimarro. The Galaxy Power Spectrum Multipoles Covariance in Perturbation Theory. *arXiv e-prints*, art. arXiv:1910.02914, October 2019.

[164] Leander Thiele, Francisco Villaescusa-Navarro, David N. Spergel, Dylan Nelson, and Annalisa Pillepich. Teaching neural networks to generate Fast Sunyaev Zel'dovich Maps. *arXiv e-prints*, art. arXiv:2007.07267, July 2020.

[165] Yongseok Jo and Ji-hoon Kim. Machine-assisted semi-simulation model (MSSM): estimating galactic baryonic properties from their dark matter using a machine trained on hydrodynamic simulations. , 489(3):3565–3581, November 2019. doi: 10.1093/mnras/stz2304.

[166] Benjamin Horowitz, Max Dornfest, Zarija Lukić, and Peter Harrington. HyPhy: Deep Generative Conditional Posterior Mapping of Hydrodynamical Physics. *arXiv e-prints*, art. arXiv:2106.12675, June 2021.

[167] Peter Harrington, Mustafa Mustafa, Max Dornfest, Benjamin Horowitz, and

Zarija Lukić. Fast, high-fidelity Lyman α forests with convolutional neural networks. *arXiv e-prints*, art. arXiv:2106.12662, June 2021.

[168] M. Bernardini, R. Feldmann, D. Anglés-Alcázar, M. Boylan-Kolchin, J. Bullock, L. Mayer, and J. Stadel. From EMBER to FIRE: predicting high resolution baryon fields from dark matter simulations with deep learning. , 509(1):1323–1341, January 2022. doi: 10.1093/mnras/stab3088.

[169] Ben Moews, Romeel Davé, Sourav Mitra, Sultan Hassan, and Weiguang Cui. Hybrid analytic and machine-learned baryonic property insertion into galactic dark matter haloes. , 504(3):4024–4038, July 2021. doi: 10.1093/mnras/stab1120.

[170] Benjamin P. Moster, Thorsten Naab, Magnus Lindström, and Joseph A. O'Leary. GalaxyNet: connecting galaxies and dark matter haloes with deep neural networks and reinforcement learning in large volumes. , 507(2):2115–2136, October 2021. doi: 10.1093/mnras/stab1449.

[171] Digvijay Wadekar, Francisco Villaescusa-Navarro, Shirley Ho, and Laurence Perreault-Levasseur. HInet: Generating neutral hydrogen from dark matter with neural networks. *arXiv e-prints*, art. arXiv:2007.10340, July 2020.

[172] Xiaoju Xu, Saurabh Kumar, Idit Zehavi, and Sergio Contreras. Predicting halo occupation and galaxy assembly bias with machine learning. , 507(4):4879–4899, November 2021. doi: 10.1093/mnras/stab2464.

[173] Christopher C. Lovell, Stephen M. Wilkins, Peter A. Thomas, Matthieu Schaller, Carlton M. Baugh, Giulio Fabbian, and Yannick Bahé. A machine learning approach to mapping baryons on to dark matter haloes using the EAGLE and C-EAGLE simulations. , November 2021. doi: 10.1093/mnras/stab3221.

[174] Ana Maria Delgado, Digvijay Wadekar, Boryana Hadzhiyska, Sownak Bose, Lars Hernquist, and Shirley Ho. Modeling the galaxy-halo connection with machine learning. *arXiv e-prints*, art. arXiv:2111.02422, November 2021.

[175] T. Lucas Makinen, Lachlan Lancaster, Francisco Villaescusa-Navarro, Peter Melchior, Shirley Ho, Laurence Perreault-Levasseur, and David N. Spergel. deep21: a deep learning method for 21 cm foreground removal. , 2021(4): 081, April 2021. doi: 10.1088/1475-7516/2021/04/081.

[176] Pablo Villanueva-Domingo and Francisco Villaescusa-Navarro. Removing Astrophysics in 21 cm maps with Neural Networks. *arXiv e-prints*, art. arXiv:2006.14305, June 2020.

[177] Samuel Gagnon-Hartman, Yue Cui, Adrian Liu, and Siamak Ravanbakhsh. Recovering the wedge modes lost to 21-cm foregrounds. , 504(4):4716–4729, July 2021. doi: 10.1093/mnras/stab1158.

[178] S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen, and B. Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Science*, 116:13825–13832, July 2019. doi: 10.1073/pnas.1821458116.

[179] Doogesh Kodi Ramanah, Tom Charnock, and Guilhem Lavaux. Painting halos from cosmic density fields of dark matter with physically motivated neural networks. , 100(4):043515, August 2019. doi: 10.1103/PhysRevD.100.043515.

[180] Xiaofeng Dong, Nesar Ramachandra, Salman Habib, Katrin Heitmann, Michael Buehlmann, and Sandeep Madireddy. Physical Benchmarking for AI-Generated Cosmic Web. *arXiv e-prints*, art. arXiv:2112.05681, December 2021.

[181] Elena Giusarma, Mauricio Reyes Hurtado, Francisco Villaescusa-Navarro, Siyu He, Shirley Ho, and ChangHoon Hahn. Learning neutrino effects in Cosmology with Convolutional Neural Networks. *arXiv e-prints*, art. arXiv:1910.04255, October 2019.

[182] Chang Chen, Yin Li, Francisco Villaescusa-Navarro, Shirley Ho, and Anthony Pullen. Learning the Evolution of the Universe in N-body Simulations. *arXiv e-prints*, art. arXiv:2012.05472, December 2020.

[183] Zhong-Yi Man, Ying-Jie Peng, Jing-Jing Shi, Xu Kong, Cheng-Peng Zhang, Jing Dou, and Ke-Xin Guo. The Fundamental Relation between Halo Mass and

Galaxy Group Properties. , 881(1):74, August 2019. doi: 10.3847/1538-4357/ab2ece.

[184] Victor F. Calderon and Andreas A. Berlind. Prediction of galaxy halo masses in SDSS DR7 via a machine learning approach. , 490(2):2367–2379, December 2019. doi: 10.1093/mnras/stz2775.

[185] Luisa Lucie-Smith, Hiranya V. Peiris, Andrew Pontzen, Brian Nord, and Jeyan Thiyagalingam. Deep learning insights into cosmological structure formation. *arXiv e-prints*, art. arXiv:2011.10577, November 2020.

[186] Pablo Villanueva-Domingo, Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, Federico Marinacci, David N. Spergel, Lars Hernquist, Mark Vogelsberger, Romeel Dave, and Desika Narayanan. Inferring halo masses with Graph Neural Networks. *arXiv e-prints*, art. arXiv:2111.08683, November 2021.

[187] Pablo Villanueva-Domingo, Francisco Villaescusa-Navarro, Shy Genel, Daniel Anglés-Alcázar, Lars Hernquist, Federico Marinacci, David N. Spergel, Mark Vogelsberger, and Desika Narayanan. Weighing the Milky Way and Andromeda with Artificial Intelligence. *arXiv e-prints*, art. arXiv:2111.14874, November 2021.

[188] M. Ntampaka et al. A Deep Learning Approach to Galaxy Cluster X-ray Masses. *Astrophys. J.*, 876(1):82, 2019. doi: 10.3847/1538-4357/ab14eb.

[189] Matthew Ho, Markus Michael Rau, Michelle Ntampaka, Arya Farahi, Hy Trac, and Barnabas Poczos. A Robust and Efficient Deep Learning Method for Dynamical Mass Measurements of Galaxy Clusters. *Astrophys. J.*, 887:25, 2 2019. doi: 10.3847/1538-4357/ab4f82.

[190] Doogesh Kodi Ramanah, Radosław Wojtak, Zoe Ansari, Christa Gall, and Jens Hjorth. Dynamical mass inference of galaxy clusters with neural flows. *Mon. Not. Roy. Astron. Soc.*, 499(2):1985–1997, 2020. doi: 10.1093/mnras/staa2886.

[191] Doogesh Kodi Ramanah, Radosław Wojtak, and Nikki Arendse. Simulation-based inference of dynamical galaxy cluster masses with 3D convolutional neural networks. *Mon. Not. Roy. Astron. Soc.*, 501(3):4080–4091, 2021. doi: 10.1093/mnras/staa3922.

[192] Z. Yan, A. J. Mead, L. Van Waerbeke, G. Hinshaw, and I. G. McCarthy. Galaxy cluster mass estimation with deep learning and hydrodynamical simulations. , 499(3):3445–3458, December 2020. doi: 10.1093/mnras/staa3030.

[193] N. Gupta and C. L. Reichardt. Mass Estimation of Galaxy Clusters with Deep Learning. I. Sunyaev-Zel'dovich Effect. , 900(2):110, September 2020. doi: 10.3847/1538-4357/aba694.

[194] Daniel de Andres, Weiguang Cui, Florian Ruppin, Marco De Petris, Gustavo Yepes, Ichraf Lahouli, Gianmarco Aversano, Romain Dupuis, and Mahmoud

Jarraya. Mass Estimation of Planck Galaxy Clusters using Deep Learning. 11 2021.

[195] Helen Shao, Francisco Villaescusa-Navarro, Shy Genel, David N. Spergel, Daniel Angles-Alcazar, Lars Hernquist, Romeel Dave, Desika Narayanan, Gabriella Contardo, and Mark Vogelsberger. Finding universal relations in subhalo properties with artificial intelligence. *arXiv e-prints*, art. arXiv:2109.04484, September 2021.

[196] Levi Fussell and Ben Moews. Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks. , 485(3):3203–3214, May 2019. doi: 10.1093/mnras/stz602.

[197] Sultan Hassan, Adrian Liu, Saul Kohn, and Paul La Plante. Identifying reionization sources from 21 cm maps using Convolutional Neural Networks. , 483 (2):2524–2537, February 2019. doi: 10.1093/mnras/sty3282.

[198] Christopher C Lovell, Viviana Acquaviva, Peter A Thomas, Kartheik G Iyer, Eric Gawiser, and Stephen M Wilkins. Learning the relationship between galaxies spectra and their star formation histories using convolutional neural networks and cosmological simulations. , 490(4):5503–5520, 10 2019. ISSN 0035-8711. doi: 10.1093/mnras/stz2851. URL `https://doi.org/10.1093/mnras/stz2851`.

[199] Sankalp Gilda, Sidney Lower, and Desika Narayanan. MIRKWOOD: Fast and Accurate SED Modeling Using Machine Learning. , 916(1):43, July 2021. doi: 10.3847/1538-4357/ac0058.

[200] George Stein. georgestein/ml-in-cosmology: Machine learning in cosmology, September 2020. URL `https://doi.org/10.5281/zenodo.4024768`.

[201] Krishna Naidoo, Elena Massara, and Ofer Lahav. Cosmology and neutrino mass with the Minimum Spanning Tree. *arXiv e-prints*, art. arXiv:2111.12088, November 2021.

[202] Lucas Porth, Gary M. Bernstein, Robert E. Smith, and Abigail J. Lee. The Information Content of Projected Galaxy Fields. *arXiv e-prints*, art. arXiv:2111.13702, November 2021.

[203] Hector J. Hortua. Constraining cosmological parameters from N-body simulations with Bayesian Neural Networks. *arXiv e-prints*, art. arXiv:2112.11865, December 2021.

[204] R. S. Somerville and R. Davé. Physical Models of Galaxy Formation in a Cosmological Framework. , 53:51–113, August 2015. doi: 10.1146/annurev-astro-082812-140951.

[205] Thorsten Naab and Jeremiah P. Ostriker. Theoretical Challenges in Galaxy Formation. , 55(1):59–109, August 2017. doi: 10.1146/annurev-astro-081913-040019.

[206] Francisco Villaescusa-Navarro, Daniel Anglés-Alcázar, Shy Genel, David N. Spergel, Rachel S. Somerville, Romeel Dave, Annalisa Pillepich, Lars Hernquist, Dylan Nelson, Paul Torrey, Desika Narayanan, Yin Li, Oliver Philcox, Valentina La Torre, Ana Maria Delgado, Shirley Ho, Sultan Hassan, Blakesley Burkhart, Digvijay Wadekar, Nicholas Battaglia, Gabriella Contardo, and Greg L. Bryan. The CAMELS Project: Cosmology and Astrophysics with Machine-learning Simulations. , 915(1):71, July 2021. doi: 10.3847/1538-4357/abf7ba.

[207] Francisco Villaescusa-Navarro, Benjamin D. Wandelt, Daniel Anglés-Alcázar, Shy Genel, Jose Manuel Zorrilla Mantilla, Shirley Ho, and David N. Spergel. Neural networks as optimal estimators to marginalize over baryonic effects. *arXiv e-prints*, art. arXiv:2011.05992, November 2020.

[208] Faizan G. Mohammad, Francisco Villaescusa-Navarro, Shy Genel, Daniel Angles-Alcazar, and Mark Vogelsberger. Inpainting hydrodynamical maps with deep learning. *arXiv e-prints*, art. arXiv:2109.07070, September 2021.

[209] Francisco Villaescusa-Navarro, Shy Genel, Daniel Angles-Alcazar, Leander Thiele, Romeel Dave, Desika Narayanan, Andrina Nicola, Yin Li, Pablo Villanueva-Domingo, Benjamin Wandelt, David N. Spergel, Rachel S. Somerville, Jose Manuel Zorrilla Matilla, Faizan G. Mohammad, Sultan Hassan, Helen Shao, Digvijay Wadekar, Michael Eickenberg, Kaze W. K. Wong, Gabriella Contardo, Yongseok Jo, Emily Moser, Erwin T. Lau, Luis Fernando Machado Poletti Valle, Lucia A. Perez, Daisuke Nagai, Nicholas Battaglia,

and Mark Vogelsberger. The CAMELS Multifield Dataset: Learning the Universe's Fundamental Parameters with Artificial Intelligence. *arXiv e-prints*, art. arXiv:2109.10915, September 2021.

[210] Andrina Nicola, Francisco Villaescusa-Navarro, David N. Spergel, Jo Dunkley, Daniel Anglés-Alcázar, Romeel Davé, Shy Genel, Lars Hernquist, Daisuke Nagai, Rachel S. Somerville, and Benjamin D. Wandelt. Breaking baryon-cosmology degeneracy with the electron density power spectrum. *arXiv e-prints*, art. arXiv:2201.04142, January 2022.

[211] Digvijay Wadekar, Leander Thiele, Francisco Villaescusa-Navarro, J. Colin Hill, David N. Spergel, Miles Cranmer, Nicholas Battaglia, Daniel Anglés-Alcázar, Lars Hernquist, and Shirley Ho. Augmenting astrophysical scaling relations with machine learning : application to reducing the SZ flux-mass scatter. *arXiv e-prints*, art. arXiv:2201.01305, January 2022.

[212] Digvijay Wadekar, Leander Thiele, Francisco Villaescusa-Navarro, Colin Hill, David Spergel, Miles Cranmer, Shirley Ho, et al. *in preparation*, 2022.

[213] Leander Thiele, Digvijay Wadekar, J. Colin Hill, Nicholas Battaglia, Jens Chluba, Francisco Villaescusa-Navarro, Lars Hernquist, Mark Vogelsberger, Daniel Anglés-Alcázar, and Federico Marinacci. Percent-level constraints on baryonic feedback with spectral distortion measurements. *arXiv e-prints*, art. arXiv:2201.01663, January 2022.

[214] Emily Moser, Nicholas Battaglia, Daisuke Nagai, Erwin Lau, Luis Fernando Machado Poletti Valle, Francisco Villaescusa-Navarro, Stefania Amodeo, Daniel Angles-Alcazar, Greg L. Bryan, Romeel Dave, Lars Hernquist, and Mark Vogelsberger. The Circumgalactic Medium from the CAMELS Simulations: Forecasting Constraints on Feedback Processes from Future Sunyaev-Zeldovich Observations. *arXiv e-prints*, art. arXiv:2201.02708, January 2022.

[215] Francisco Villaescusa-Navarro, Jupiter Ding, Shy Genel, Stephanie Tonnesen, Valentina La Torre, David N. Spergel, Romain Teyssier, Yin Li, Caroline Heneka, Pablo Lemos, Daniel Anglés-Alcázar, Daisuke Nagai, and Mark Vogelsberger. Cosmology with one galaxy? *arXiv e-prints*, art. arXiv:2201.02202, January 2022.

[216] Yongseok Jo, Shy Genel, Benjamin Wandelt, Francisco Villaescusa-Navarro, Greg Bryan, Rachel Somerville, Daniel Angles-Alcazar, Ji-hoon Kim, et al. *in preparation*, 2022.

[217] Lucia Perez, Shy Genel, et al. *in preparation*, 2022.

[218] V. Springel. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. , 401:791–851, January 2010. doi: 10.1111/j.1365-2966.2009.15715.x.

[219] Rainer Weinberger, Volker Springel, and Rüdiger Pakmor. The Arepo public code release. *arXiv e-prints*, art. arXiv:1909.04667, September 2019.

[220] P. F. Hopkins. A new class of accurate, mesh-free hydrodynamic simulation methods. , 450:53–110, June 2015. doi: 10.1093/mnras/stv195.

[221] R. Weinberger, V. Springel, L. Hernquist, A. Pillepich, F. Marinacci, R. Pakmor, D. Nelson, S. Genel, M. Vogelsberger, J. Naiman, and P. Torrey. Simulating galaxy formation with black hole driven thermal and kinetic feedback. , 465:3291–3308, March 2017. doi: 10.1093/mnras/stw2944.

[222] A. Pillepich, V. Springel, D. Nelson, S. Genel, J. Naiman, R. Pakmor, L. Hernquist, P. Torrey, M. Vogelsberger, R. Weinberger, and F. Marinacci. Simulating galaxy formation with the IllustrisTNG model. , 473:4077–4106, January 2018. doi: 10.1093/mnras/stx2656.

[223] Romeel Davé, Daniel Anglés-Alcázar, Desika Narayanan, Qi Li, Mika H. Rafieferantsoa, and Sarah Appleby. SIMBA: Cosmological simulations with black hole growth and feedback. , 486(2):2827–2849, June 2019. doi: 10.1093/mnras/stz937.

[224] V. Springel. The cosmological simulation code GADGET-2. , 364:1105–1134, December 2005. doi: 10.1111/j.1365-2966.2005.09655.x.

[225] D. Anglés-Alcázar, C.-A. Faucher-Giguère, D. Kereš, P. F. Hopkins, E. Quataert, and N. Murray. The cosmic baryon cycle and galaxy mass assembly in the FIRE simulations. , 470:4698–4719, October 2017. doi: 10.1093/mnras/stx1517.

[226] D. Anglés-Alcázar, R. Davé, C.-A. Faucher-Giguère, F. Özel, and P. F. Hopkins. Gravitational torque-driven black hole growth and feedback in cosmological simulations. , 464:2840–2853, January 2017. doi: 10.1093/mnras/stw2565.

[227] Dylan Nelson, Volker Springel, Annalisa Pillepich, Vicente Rodriguez-Gomez, Paul Torrey, Shy Genel, Mark Vogelsberger, Ruediger Pakmor, Federico Marinacci, Rainer Weinberger, Luke Kelley, Mark Lovell, Benedikt Diemer, and Lars Hernquist. The IllustrisTNG simulations: public data release. *Computational Astrophysics and Cosmology*, 6(1):2, May 2019. doi: 10.1186/s40668-019-0028-x.

[228] V. Springel, S. D. M. White, G. Tormen, and G. Kauffmann. Populating a cluster of galaxies - I. Results at [formmu2]z=0. , 328:726–750, December 2001. doi: 10.1046/j.1365-8711.2001.04912.x.

[229] K. Dolag, S. Borgani, G. Murante, and V. Springel. Substructures in hydrodynamical cluster simulations. , 399(2):497–514, October 2009. doi: 10.1111/j.1365-2966.2009.15034.x.

[230] Peter S. Behroozi, Risa H. Wechsler, and Hao-Yi Wu. The ROCKSTAR Phase-space Temporal Halo Finder and the Velocity Offsets of Cluster Cores. , 762 (2):109, Jan 2013. doi: 10.1088/0004-637X/762/2/109.

[231] S. R. Knollmann and A. Knebe. AHF: Amiga's Halo Finder. , 182:608–624, June 2009. doi: 10.1088/0067-0049/182/2/608.

[232] Peter S. Behroozi, Risa H. Wechsler, Hao-Yi Wu, Michael T. Busha, Anatoly A. Klypin, and Joel R. Primack. Gravitationally Consistent Halo Catalogs and Merger Trees for Precision Cosmology. , 763(1):18, January 2013. doi: 10.1088/0004-637X/763/1/18.

[233] Charlie Conroy, James E. Gunn, and Martin White. The Propagation of Uncertainties in Stellar Population Synthesis Modeling. I. The Relevance of Uncertain Aspects of Stellar Evolution and the Initial Mass Function to the Derived Physical Properties of Galaxies. , 699(1):486–506, July 2009. doi: 10.1088/0004-637X/699/1/486.

[234] Charlie Conroy and James E. Gunn. The Propagation of Uncertainties in Stellar Population Synthesis Modeling. III. Model Calibration, Comparison, and Evaluation. , 712(2):833–857, April 2010. doi: 10.1088/0004-637X/712/2/833.

[235] P. M. Sutter, G. Lavaux, N. Hamaus, A. Pisani, B. D. Wandelt, M. Warren, F. Villaescusa-Navarro, P. Zivick, Q. Mao, and B. B. Thompson. VIDE: The Void IDentification and Examination toolkit. *Astronomy and Computing*, 9: 1–9, March 2015. doi: 10.1016/j.ascom.2014.10.002.

[236] Mark C. Neyrinck. ZOBOV: a parameter-free void-finding algorithm. *Monthly Notices of the Royal Astronomical Society*, 386:2101–2109, Jun 2008. doi: 10.1111/j.1365-2966.2008.13180.x.

[237] Nico Hamaus, Alice Pisani, Paul M. Sutter, Guilhem Lavaux, Stéphanie Escoffier, Benjamin D. Wandelt, and Jochen Weller. Constraints on Cosmology and Gravity from the Dynamics of Voids. *Phys. Rev. Lett.*, 117(9):091302, 2016. doi: 10.1103/PhysRevLett.117.091302.

[238] Nico Hamaus, Alice Pisani, Jin-Ah Choi, Guilhem Lavaux, Benjamin D. Wandelt, and Jochen Weller. Precision cosmology with voids in the final BOSS data. *Journal of Cosmology and Astroparticle Physics*, 2020(12):023, December 2020. doi: 10.1088/1475-7516/2020/12/023.

[239] Marie Aubert, Marie-Claude Cousinou, Stéphanie Escoffier, Adam J. Hawken, Seshadri Nadathur, Shadab Alam, Julian Bautista, Etienne Burtin, Arnaud de Mattia, Héctor Gil-Marín, Jiamin Hou, Eric Jullo, Richard Neveux, Graziano Rossi, Alex Smith, Amélie Tamone, and Mariana Vargas Magaña. The Completed SDSS-IV Extended Baryon Oscillation Spectroscopic Survey: Growth rate of structure measurement from cosmic voids. *arXiv e-prints*, art. arXiv:2007.09013, July 2020.

[240] G. Pollina, N. Hamaus, K. Paech, K. Dolag, J. Weller, C. Sánchez, E. S. Rykoff, B. Jain, T. M. C. Abbott, S. Allam, S. Avila, R. A. Bernstein, E. Bertin, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, J. De Vicente, D. L. DePoy, S. Desai, H. T. Diehl, P. Doel, A. E. Evrard, B. Flaugher, P. Fosalba, J. Frieman, J. García-Bellido, D. W. Gerdes, T. Giannantonio, D. Gruen,

J. Gschwend, G. Gutierrez, W. G. Hartley, D. L. Hollowood, K. Honscheid, B. Hoyle, D. J. James, T. Jeltema, K. Kuehn, N. Kuropatkin, M. Lima, M. March, J. L. Marshall, P. Melchior, F. Menanteau, R. Miquel, A. A. Plazas, A. K. Romer, E. Sanchez, V. Scarpine, R. Schindler, M. Schubnell, I. Sevilla-Noarbe, M. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, G. Tarle, A. R. Walker, and W. Wester. On the relative bias of void tracers in the Dark Energy Survey. *ArXiv e-prints*, June 2018.

[241] Christina D. Kreisch, Alice Pisani, Carmelita Carbone, Jia Liu, Adam J. Hawken, Elena Massara, David N. Spergel, and Benjamin D. Wandelt. Massive neutrinos leave fingerprints on cosmic voids. , 488(3):4413–4426, September 2019. doi: 10.1093/mnras/stz1944.

[242] Giovanni Verza, Alice Pisani, Carmelita Carbone, Nico Hamaus, and Luigi Guzzo. The void size function in dynamical dark energy cosmologies. *Journal of Cosmology and Astroparticle Physics*, 2019(12):040–040, Dec 2019. ISSN 1475-7516. doi: 10.1088/1475-7516/2019/12/040. URL `http://dx.doi.org/10.1088/1475-7516/2019/12/040`.

[243] Sofia Contarini, Federico Marulli, Lauro Moscardini, Alfonso Veropalumbo, Carlo Giocoli, and Marco Baldi. Cosmic voids in modified gravity models with massive neutrinos. *accepted*, April 2021. doi: 10.1093/mnras/stab1112.

[244] Christina D. Kreisch, Alice Pisani, Francisco Villaescusa-Navarro, David N.

Spergel, Benjamin D. Wandelt, Nico Hamaus, and Adrian E. Bayer. The GIGANTES dataset: precision cosmology from voids in the machine learning era. *arXiv e-prints*, art. arXiv:2107.02304, July 2021.

[245] Mélanie Habouzit, Alice Pisani, Andy Goulding, Yohan Dubois, Rachel S. Somerville, and Jenny E. Greene. Properties of simulated galaxies and supermassive black holes in cosmic voids. , 493(1):899–921, March 2020. doi: 10.1093/mnras/staa219.

[246] Rushy R. Panchal, Alice Pisani, and David N. Spergel. How Do Galaxy Properties Affect Void Statistics? , 901(1):87, September 2020. doi: 10.3847/1538-4357/abadff.

[247] Simeon Bird, Martin Haehnelt, Marcel Neeleman, Shy Genel, Mark Vogelsberger, and Lars Hernquist. Reproducing the kinematics of damped Lyman α systems. , 447(2):1834–1846, February 2015. doi: 10.1093/mnras/stu2542.

[248] Alex Gurvich, Blakesley Burkhart, and Simeon Bird. The Effect of AGN Heating on the Low-redshift Lyα Forest. , 835(2):175, February 2017. doi: 10.3847/1538-4357/835/2/175.

[249] Oliver H. E. Philcox and Daniel J. Eisenstein. Computing the small-scale galaxy power spectrum and bispectrum in configuration space. , 492(1):1214–1242, February 2020. doi: 10.1093/mnras/stz3335.

[250] Oliver H. E. Philcox. A faster Fourier transform? Computing small-scale power spectra and bispectra for cosmological simulations in ğ'a(N^2) time. , 501(3): 4004–4034, March 2021. doi: 10.1093/mnras/staa3882.

[251] Francisco Villaescusa-Navarro. Pylians: Python libraries for the analysis of numerical simulations, November 2018.

[252] Catherine A. Watkinson, Suman Majumdar, Jonathan R. Pritchard, and Rajesh Mondal. A fast estimator for the bispectrum and beyond - a practical method for measuring non-Gaussianity in 21-cm maps. , 472(2):2436–2446, December 2017. doi: 10.1093/mnras/stx2130.

[253] Emily Moser, Stefania Amodeo, Nicholas Battaglia, Marcelo A. Alvarez, Simone Ferraro, and Emmanuel Schaan. The Impacts of Modeling Choices on the Inference of Circumgalactic Medium Properties from Sunyaev-Zeldovich Observations. , 919(1):2, September 2021. doi: 10.3847/1538-4357/ac0cea.

[254] Thomas Dauser, Sebastian Falkner, Maximilian Lorenz, Christian Kirsch, Philippe Peille, Edoardo Cucchetti, Christian Schmid, Thorsten Brand, Mirjam Oertel, Randall Smith, and Jörn Wilms. SIXTE: a generic X-ray instrument simulation toolkit. , 630:A66, Sep 2019. doi: 10.1051/0004-6361/201935978.

[255] Rachel S. Somerville, Philip F. Hopkins, Thomas J. Cox, Brant E. Robertson, and Lars Hernquist. A semi-analytic model for the co-evolution of galaxies,

black holes and active galactic nuclei. , 391(2):481–506, December 2008. doi: 10.1111/j.1365-2966.2008.13805.x.

[256] Rachel S. Somerville, Gergö Popping, and Scott C. Trager. Star formation in semi-analytic galaxy formation models with multiphase gas. , 453(4):4337–4367, November 2015. doi: 10.1093/mnras/stv1877.

[257] B. Rossi. *ZPhy*, 82:151, 1933. doi: 10.1007/BF01341486. URL https://doi.org/10.1007/BF01341486.

[258] M. Tanabashi, K. Hagiwara, K. Hikasa, et al. *PhRvD*, 98:030001, 2018. doi: 10.1103/PhysRevD.98.030001. URL https://doi.org/10.1103/PhysRevD.98.030001.

[259] P. A. Čerenkov. *PhRv*, 52:378, 1937. doi: 10.1103/PhysRev.52.378. URL https://doi.org/10.1103/PhysRev.52.378.

[260] A. Cavaliere, P. Morrison, and L. Sartori. *Sci*, 173:525, 1971. doi: 10.1126/science.173.3996.525. URL https://doi.org/10.1126/science.173.3996.525.

[261] R. J. Nemiroff. *PASA*, 32:e001, 2015. doi: 10.1017/pasa.2014.46. URL https://doi.org/10.1017/pasa.2014.46.

[262] R. J. Nemiroff. *AnP*, 530:1700333, 2018. doi: 10.1002/andp.201700333. URL https://doi.org/10.1002/andp.201700333.

[263] R. Nemiroff. *AAS Meeting*, 233:251.01, 2019.

[264] M. Clerici, G. C. Spalding, R. Warburton, et al. *SciA*, 2:e1501691, 2016. doi: 10.1126/sciadv.1501691. URL https://doi.org/10.1126/sciadv.1501691.

[265] A. Velten, D. Wu, A. Jarabo, et al. *ACM Trans. Graph.*, 32:1, 2013. doi: 10.1145/2461912.2461928. URL https://doi.org/10.1145/2461912.2461928.

[266] D. Faccio and A. Velten. *RPPh*, 81, 2018. doi: 10.1088/1361-6633/aacca1. URL https://doi.org/10.1088/1361-6633/aacca1.

[267] J. Hakkila and R. Nemiroff. *ApJ*, 883:70, 2019. doi: 10.3847/1538-4357/ab3bdf. URL https://doi.org/10.3847/1538-4357/ab3bdf.

[268] R. J. Nemiroff and N. Kaushal. *ApJ*, 889:122, 2020. doi: 10.3847/1538-4357/ab6440. URL https://doi.org/10.3847/1538-4357/ab6440.

[269] A. Aab, P. Abreu, M. Aglietta, et al. *ApJ*, 802:111, 2015. doi: 10.1088/0004-637X/802/2/111. URL https://doi.org/10.1088/0004-637X/802/2/111.

[270] S. BenZvi. *EPJWC*, 105:01003, 2015. doi: 10.1051/epjconf/201510501003. URL https://doi.org/10.1051/epjconf/201510501003.

[271] K. Arisaka, T. Kajita, T. Kifune, et al. In M. Blecher and K. Gotow, editors, *AIP Conf. Proc. 114, Low Energy Tests of Conservation Laws in Particle Physics*, page 54. AIP, 1984.

[272] M. G. Aartsen, M. Ackermann, J. Adams, et al. *JInst*, 12:P03012, 2017. doi: 10.1088/1748-0221/12/03/P03012. URL https://doi.org/10.1088/1748-0221/12/03/P03012.

[273] A. Sandoval. *ICRC (Rio de Janeiro)*, 33:1155, 2013.

[274] N. Kaushal and R. J. Nemiroff. ascl:2005.001, 2020.

[275] P. Huentemeyer, J. A. J. Matthews, and B. Dingus. arXiv:0909.2830, 2009.

[276] V. Joshi. *PhD Thesis*. PhD thesis, Heidelberg Univ., 2019. doi:10.11588/heidok.00026062.

[277] I. G. Wisher. *PhD Thesis*. PhD thesis, Univ. of Wisconsin-Madison, 2016.

[278] B. L. Dingus. In S. Ritz, P. Michelson, and C. A. Meegan, editors, *AIP Conf. Proc. 912, The First GLAST Symp.*, page 438. AIP, 2007.

[279] A. Nomerotski. *NIMPA*, 937:26, 2019. doi: 10.1016/j.nima.2019.05.034. URL https://doi.org/10.1016/j.nima.2019.05.034.

[280] J. S. Milnes, C. J. Horsfield, M. S. Rubery, V. Yu. Glebov, and H. W. Herrmann. *RScI*, 83:10D301, 2012. doi: 10.1063/1.4728313. URL https://doi.org/10.1063/1.4728313.