Application of Machine Learning in Cosmology

by
Chandan Ganjigere Nagarajappa

A thesis submitted to the College of Agriculture, Engineering and Sciences
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Physics
in the
School of Chemistry and Physics

UNIVERSITY OF KWAZULU-NATAL™
INYUVESI YAKWAZULU-NATALI

Westville Campus, Durban, South Africa

Supervised by Professor Yin-Zhe Ma

Abstract

We present a novel approach to estimate the value of primordial non-Gaussianity ($f_{\rm NL}$) parameter directly from the Cosmic Microwave Background (CMB) maps using a convolutional neural network (CNN). While traditional methods rely on complex statistical techniques, this study proposes a simpler approach that employs a neural network to estimate $f_{\rm NL}$. The neural network model is trained on simulated CMB maps with known $f_{\rm NL}$ in range of $[-50, 50]$, and its performance is evaluated using various metrics. The results indicate that the proposed approach can accurately estimate $f_{\rm NL}$ values from CMB maps with a significant reduction in complexity compared to traditional methods. With 500 validation data, the $f_{\rm NL}^{\rm output}$ against $f_{\rm NL}^{\rm input}$ graph can be fitted as $y = ax + b$, where $a = 0.980^{+0.098}_{-0.102}$ and $b = 0.277^{+0.098}_{-0.101}$, indicating the unbiasedness of the primordial non-Gaussianity estimation. The results indicate that the CNN technique can be widely applied to other cosmological parameter estimation directly from CMB images.

We further extend our neural network approach to the estimation of cosmological parameters by using both CMB temperature and polarisation data. Conventional methods for parameter estimation and constraint, such as two-point statistics, likelihood inference, and MCMC methods necessitate substantial computational resources, posing challenges for detailed analyses of large cosmological datasets. In our study, we introduce a customized CNN model tailored specifically for estimating cosmological parameters from CMB maps. Our CNN architecture exhibits sensitivity to variations in underlying cosmological parameters within CMB maps, enabling more precise and computationally efficient parameter estimation. We present our findings in estimating the standard six cosmological parameters, including the Cold Dark Matter (CDM) density ($\Omega_{\rm c}h^2$), optical depth (τ), the scalar amplitude of the primordial curvature perturbations (A_s), Hubble parameter (H_0), the Baryon matter density ($\Omega_{\rm b}h^2$), and the spectral index of the primordial curvature perturbations (n_s). These parameters are estimated independently of each other from the simulated Cosmological Microwave Background maps using CNN. We show the excellent agreement between the inputs and outputs of cosmological parameters, indicating the accurate recovery of parameter values by using the CNN method.

Declaration

I, **Chandan Ganjigere Nagarajappa**, declare that:

1. The research presented in this thesis, with the exception of any explicitly stated exceptions, is entirely original and has not been submitted for any degree or examination at any other university.
2. This thesis does not contain any information, data, graphs, pictures, or other materials belonging to other individuals or researchers unless properly cited and acknowledged.
3. Where other written sources have been quoted, I have either rewritten their words while preserving their meaning or used quotation marks and italics to indicate exact quotes. In all cases, the sources have been referenced appropriately.
4. I have not copied and pasted any text, graphics, or tables from the internet without proper citation and acknowledgment of the source in both the thesis and the References section.

Based on the research described in this thesis;

1. **Publication 1 (Published in Monthly Notices of Royal Astronomical Society)**

   Chandan G Nagarajappa, Yin-Zhe Ma, Constraining primordial non-Gaussianity using neural networks, Monthly Notices of the Royal Astronomical Society, Volume 529, Issue 4, April 2024, Pages 3289–3300, `https://doi.org/10.1093/mnras/stae679`

   Contribution: I executed all the analysis, modeling, and computations and initially drafted the results under the guidance and supervision of Professor Yin-Zhe Ma, who provided ongoing feedback and assistance during the subsequent revisions leading up to submission.

2. **Publication 2** (manuscript in preparation)

   Chandan G Nagarajappa, Yin-Zhe Ma. **Cosmological Parameter Estimation from CMB Maps using Neural Networks**

   Contribution: I executed all the analysis, modeling, and computations and initially drafted the results under the guidance and supervision of Professor Yin-Zhe Ma, who provided ongoing feedback and assistance during the subsequent revisions leading up to submission.

I, Chandan Ganjigere Nagarajappa, the undersigned, hereby affirm that the thesis titled "**Application of Machine Learning in Cosmology**" is an original work solely produced by me and is being submitted to the University of KwaZulu-Natal, Durban, for the award of the degree of Doctor of Philosophy in Science. I declare that this thesis has not been previously submitted for any degree or examination at any other university, and I am currently registered as a student at the School of Chemistry and Physics, Westville campus, UKZN, with registration number 219095826." …

[redacted]

Chandan Ganjigere Nagarajappa, December 22, 2023
Reg. No. : 219095826
College of Agriculture, Engineering and Sciences
School of Chemistry and Physics,
Westville Campus, UKZN
Durban, SA

As the candidate's supervisor, I, Yin-Zhe Ma, have approved this thesis titled **"Application of Machine Learning in Cosmology"** for the submission towards the degree of Doctor of Philosophy (PhD). The research discussed in this dissertation is carried by Chandan Ganjigere Nagarajappa under my supervision in the College of Agriculture, Engineering and Science of the University of KwaZulu-Natal, Durban, from July 2019 until December 2023

Signed.....[redacted].....Date.....12/12/2023.....

Name: Prof. Yin-Zhe Ma

Acknowledgement

As I complete my doctoral journey, I am filled with a deep sense of gratitude and appreciation for the many individuals and institutions that have contributed to my success. This acknowledgement is an expression of my heartfelt thanks to those who have played a significant role in my academic and personal growth.

Firstly, I would like to extend my sincere gratitude to my advisor, Professor Yin-Zhe Ma, for his unwavering support and guidance throughout my PhD program. His expertise, insights, and encouragement have been instrumental in shaping my research and helping me develop as a scholar. I am grateful for his mentorship, which has been both challenging and rewarding. His generous funding, support to attend conferences, has also enabled me to pursue my research interests for which I am deeply appreciative.

I would also like to express my gratitude to the funding agencies "New Insights into Astrophysics and Cosmology with Theoretical Models Confronting Observational Data" of the National Institute for Theoretical and Computational Sciences of South Africa. Partial funding from Big Data for Society and Science (BDSS) and the National Research Foundation (NRF) for and top-up from my advisor when necessitated, which has been critical in enabling me to carry out my research. I would like to express my gratitude to the *Centre for High-Performance Computing (CHPC), Cape Town, South Africa*, for providing me with access to their state-of-the-art computing facilities. Immense gratitude towards SATS school, in particular machine learning course conducted by Prof. Jon Shock and his student Jeremy du Plessis.

I would also like to thank the *University of KwaZulu-Natal, Durban, South Africa* for providing me with the opportunity to pursue my PhD program. The University's commitment to academic excellence and its focus on research have been inspiring, and I am proud to be a part of this community. The University's resources and seminar series in our *Astrophysics Research Centre (ARC)*, have been essential in facilitating my research and professional development. I extend my thanks to our research group NUCAC (NAOC-UKZN Computational Astrophysics Centre), in particular I thank Mthokosizi, Cheng Cheng, Guo-Jian Wang, Elimboto Yohana, Ayodeji Ibitoye, Tony, Wei-ming, Denis Tramonte, with whom I had fruitful discussions during our weekly meetings which helped in my research.

Lastly, I would like to acknowledge the significant role that my family and friends have played in supporting me throughout this journey. Their encouragement, love, and unwavering belief in me have been a source of strength during challenging times. My Father Nagarajappa, my Mother Chandrakala and my sister Sushma for their sacrifices and support have enabled me to focus on my studies and research. I am grateful for their presence in my life and their contribution to my success. I thank the friends I made in Durban who are like my family to me. In particular Doctor Uncle Rana, Kala aunty, Dr. Saumitra Misra, Karpoormath family for their love and support for my entire stay in Durban. I also thank Patchamma Moodley, Kamala moodley, Uncle Raam, Aunt Romila for their care.

I would also like to extend my heartfelt thanks to my friends and colleagues from India, including those at IISc, RRI, and other universities, for their unwavering support and belief in me. Their encouragement, motivation, discussions, advises, and willingness to help at any time have been invaluable throughout my PhD program. I am grateful for the contributions of JK, Swamy, Asha, Venkat who have been a constant source of support and inspiration. I also extend my thanks to Parbati, Rahul, Austin, Nandini, Irla, Sreeja, Rohan, Mari, and Jaggu. Additionally, I would like to thank Mugundhan for his support discussions both online and offline regarding all parts of life and research, Saurabh for his insightful discussions, and Anshu for her genuine care and support. Their presence in my life has been a true blessing. I would like to express my sincere gratitude to Vidya for being a constant source of support, unwavering belief in me and encouragement throughout my journey. My gratitude towards Dr. Moumita Aich, Prof. Matt Hilton, Prof. Subarthi Ray, Prof. Sunil Maharaj who made my stay memorable in Durban, and Prof. Prof. T. A. Abinandanan, and CPR staff at IISc for their support.

In conclusion, I would like to express my heartfelt thanks to all those who have contributed to my academic and personal growth during this PhD program. If there are any individuals who have provided support and encouragement during your journey but were not specifically mentioned, please accept my sincere apologies for any oversight. As I move forward into the next phase of my academic career, I will carry their lessons and insights with me always.

Contents

List of Tables

List of Figures

1. Introduction

The human fascination with the universe and its origins is very ancient. It can be traced back to ancient civilizations such as the Indians, Greeks, Chinese, Babylonians, and Egyptians. These cultures developed mythological explanations for the universe, often involving deities and supernatural forces. As scientific knowledge advanced, so too did our understanding of the universe. The ancient Greeks made significant contributions to astronomy, with philosophers such as Aristotle and Ptolemy developing geocentric models of the solar system. These models placed the Earth at the center of the universe and explained the movements of the planets and stars in terms of circular orbits.

The scientific revolution brought about a significant shift in our understanding of the universe, with the development of the telescope and the laws of motion and universal gravitation by Isaac Newton. This led to the discovery of new celestial objects, such as comets and nebulae, and the development of new theories, such as the nebular hypothesis, which proposed that the solar system formed from a rotating cloud of gas and dust. Heinrich Olber's 19th-century astronomical concept, known as Olber's paradox or the dark sky paradox, presented a significant obstacle to the understanding of an everlasting and stationary universe. The paradox argues that if the universe is infinite and static, then every line of sight in the sky should lead to a bright object, such as a star, due to the vastness of the universe. However, this is not observed, as the night sky appears dark and empty. Perhaps Einstein's discovery of general relativity in early 1915s, the discovery of the expanding universe by Edwin Hubble in 1920s drew attention towards cosmology which marked the beginning of modern cosmology.

The study of the universe continues to fascinate and challenge us, as we strive to unravel its mysteries and understand our place in it. Cosmology is a rapidly evolving field, with new discoveries and insights being made all the time and it all started with a 'Big Bang'. Cosmology has been shaped by various theories, including the Steady State Universe, the Expanding Universe after the Big Bang, Inflationary Models Linde (1982, 2005); Hawking W (1966); Kofman et al. (2002), and Cyclic Models Penrose (1965). Some of these models have been refuted by observable evidence, while others remain unproven. Initially, the scientific community was skeptical of the Big Bang theory, which was proposed by the Belgian Catholic priest and astronomer Georges Lemaître in the 1920s. Lemaître's theory, based on Albert Einstein's theory of general relativity, suggested that the universe began as a singularity, a point of infinite density and temperature, and has been expanding ever since.

In opposition to the Big Bang theory, Herman Bondi proposed the Steady State model in the 1940s. Bondi, along with Thomas Gold and Fred Hoyle, developed this model as an alternative to the Big Bang theory, which was gaining popularity at the time. Bondi presented his ideas in a paper published in the journal Nature in 1948, arguing that the universe was not expanding but rather remained constant in size and density, with new matter being continuously created to replace the matter that was moving away from us due to the expansion.

The Steady State model gained some support in the scientific community in the 1950s and up until the 1960s. However, the dispute between the Steady State Model and an evolutionary model of the universe was settled by the discovery of the Cosmic Microwave Background radiation (CMB) in 1964 by Arno Penzias and Robert Wilson Wilson & Penzias (1965). The redshift of galaxies due to the expansion, which the Steady State Model failed to explain, and the CMB radiation provided strong evidence in support of the Big Bang theory. Following the discovery of the Cosmic Microwave Background (CMB), a series of intriguing questions have emerged in the field of cosmology. The discovery of redshifted wavelengths of primordial photons at the far end of the electromagnetic spectrum is attributed to the expansion of the universe as these photons travel through space. This observation reinforces the hypothesis that the universe began as a hot and dense state during the Big Bang and has been expanding since then, marking the inception of physical cosmology. Subsequent investigations have aimed to address various cosmological queries, such as how the highly structured universe we observe today could have originated from a remarkably smooth universe as indicated by the Cosmic Microwave Background (CMB) that formed a few thousand years after the Big Bang. The studies also seek to understand the forces driving the expansion of the cosmic structure and the possibility of a rapid expansion, known as inflation, that occurred just a fraction of a second after the Big Bang. These processes are being examined independently and collectively to explain the structure formation and the evolution of the universe. The growth of large-scale struc-

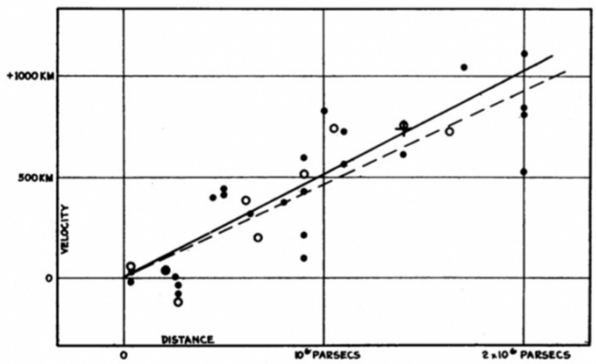

FIGURE 1

Velocity-Distance Relation among Extra-Galactic Nebulae.

Figure 1.1: Edwin Hubble's plot of the Velocity-Distance relationship for galaxies (Credit : Hubble (1929)).

ture in the universe is linked to the linear perturbation theory Bernardeau et al. (2002); Mukhanov et al. (1992).

1.1 The Big Bang and the Expanding universe

Hubble's discovery of the linear relationship between a galaxy's redshift and its distance, now known as Hubble-Lemaitre's law, was a major breakthrough in our understanding of the universe. This law, which was first published in a paper in the Proceedings of the National Academy of Sciences in 1929. The decpicted figure (1.1) is a original plot by Hubble.

Hubble's measurements showed that the farther away a galaxy was, the faster it appeared to be moving away from us. From a plot of redshift (z) versus distance (r) shown in (1.1), he found the famous linear relation now known as Hubble's law. It is mathematically expressed as:

$$z = \frac{H_0}{c} r, \tag{1.1.1}$$

where H_0 is a constant now called Hubble's constant. Hubble applied classical non-relativistic relation for the Doppler shift $z = v/c$ as the values observed in redhift of the galaxies where small where v is the radial velocity of the light galaxy, by this implementation Hubble's law can be re-written as:

$$v = H_0 r, \tag{1.1.2}$$

Where H_0 is the Hubble constant with units of $\mathrm{kms^{-1}Mpc^{-1}}$. Initially, due to the underestimation of the distances to the galaxies, Hubble's constant was thought to be around 500 $kms^{-1}Mpc^{-1}$, but later several measurements estimates the value of Hubble constant to be between 65-75 $kms^{-1}Mpc^{-1}$, where 'Mpc' is a megaparsec. $1Mpc$ is approximately equal to 3.26×10^6 light-years.

The immediate implication of the Hubble's law is that if we travel back in time to reverse the cosmic expansion, the galaxies would have been infinitely close to each other at a specific point. This time in the past is determined by the inverse of the Hubble parameter H_0."

$$t_0 = \frac{r}{v} = \frac{r}{H_0 r} = H_0^{-1}. \tag{1.1.3}$$

The concept of Hubble time, denoted by the inverse of the Hubble constant $\mathrm{H_0^{-1}}$, is significant in cosmology. With a Hubble constant of $\mathrm{H_0 = 70 \pm 7kms^{-1}Mpc^{-1}}$, the Hubble time is approximately $\mathrm{H_0^{-1} = 14.0 \pm 1.5}$ Gyr. The latest determination of the Hubble constant, which is $69.8 \pm 0.8\mathrm{kms^{-1}Mpc^{-1}}$, is derived from the calibration of the Tip of the Red Giant Branch (TRGB) applied to Type Ia supernovae (SNeIa) measurements obtained by NASA/ESA Hubble Space Telescope in a study conducted by Freedman et al. (2019); Freedman et al. (2001). This value is slightly lower than the previously reported measurement of $74.03 \pm 1.42\mathrm{kms^{-1}Mpc^{-1}}$ obtained by observing pulsating stars called Cepheid variables in a nearby satellite galaxy known as the Large Magellanic Cloud through the Hubble SH0ES (Supernovae H_0 for the Equation of State) project led by Riess et al. (2019). Despite the deviations in the Hubble value discussed in Chen et al. (2017), it is largely agreed that if the relative velocities of galaxies have remained constant in the past, then all the galaxies in the universe were compactly packed into a very small volume approximately 14 billion years ago. This Hubble time is consistent with the computed ages of known stars which are very old in our universe. The discovery of galactic redshifts provides a compelling argument for the Big Bang model, which proposes that the universe expanded from a highly dense and compact state to its current low-density configuration. This evolutionary process is a natural consequence of the observed redshifts of distant galaxies. The Hubble law not only provides the Hubble time but also offers a natural distance scale, known as the Hubble distance. This distance corresponds to the farthest point that a photon can travel within the age of the universe, which is approximately equal to the speed of light divided by the Hubble constant (c/H_0). With current estimates, this distance is approximately 4300 Mpc (megaparsecs) with a margin of error of around 400 Mpc.

1.2 The cosmological principle

The observations and theoretical developments in cosmology points to smooth universe which is homogeneous and expanding, inflation and the existence of dark matter and perhaps dark energy. The smooth universe is characterized by one parameter, its geometry. There are three possibilities: flat, open and closed universes.

1.2.1 Composition of the Universe

Cosmologists believe that the universe is made up of several components, including matter, dark matter, and dark energy. While the matter that makes up the objects we see around us is composed of familiar elementary particles such as protons, neutrons, and electrons, the majority of the universe is actually made up of two mysterious components: dark matter and dark energy.

Dark matter, the first suggestion put forth by Zwicky in 1933, is a type of matter that does not interact with light or other forms of electromagnetic radiation, making it invisible to telescopes. Although the dark matter cannot be directly observed, its influence on visible matter can be inferred through its gravitational effects. Scientists estimate that dark matter makes up roughly about 27% of the total mass-energy of the universe.

Dark energy, on the other hand, is a hypothetical form of energy that permeates all of space and is suspected to be the main culprit for the *accelerating expansion* Riess et al. (1998); Perlmutter et al. (1999) of the universe which we observe today. Its exact nature is still not fully understood, but some theories suggest that it may be related to the vacuum fluctuations of spacetime itself. Dark energy is estimated to make up around 68%of the total mass-energy of the universe. So while the matter that makes up our daily lives is composed of familiar elementary particles, the majority of the universe is made up of these mysterious components that are still being studied and understood by cosmologists. The properties of the most cosmological important particles are summarized below in a table Ryden (2017)

Particle properties			
particle	symbol	rest energy (MeV)	charge
proton	p	938.3	+1
neutron	n	939.9	0
electron	e^-	0.511	-1
neutrino	γ_e,γ_μ,γ_τ	-	0
photon	γ	0	0
dark matter	-	-	0

Table 1.1: Properties of the particles composing the universe

The materials which we interact with everyday in our life is made up of protons, neutrons and electrons. Baryons, which are particles made up of three quarks, include both protons and neutrons. A proton consists of two up quarks, each with a charge of +2/3, and one down quark with a charge of $-1/3$, resulting in a net positive charge of +1. On the other hand, a neutron has one up quark and two down quarks, making it electrically neutral. The neutron marginally has a greater rest energy than the proton, with $\mathrm{m_nc^2} = 939.6$ MeV compared to $\mathrm{m_pc^2} = 938.3$ MeV for the proton. While free neutrons are unstable and decay into protons with a decay time of $\emptyset_\mathrm{n} = 940\mathrm{s}$, protons have been found to be stable with a decay time greater than the Hubble time in experiments. Neutrons can be kept from decaying when bound inside atomic nuclei with one or more protons.

Leptons, including electrons, are a type of elementary particle that do not contain quarks in their composition. The mass of an electron is much smaller than that of a neutron or proton, with a rest energy of 0.511 MeV. Electrons have an electric charge that is equal in magnitude to that of a proton but opposite in sign. In the universe, the number of electrons is equal to the number of protons on large scales, resulting in electrical neutrality. However, since the mass of the protons is significantly larger than that of electrons, by a factor of 1836 to 1, the mass density of electrons is only a small perturbation to the mass density of protons and neutrons. As a result, the component of the universe made up of ions, atoms, and molecules is generally referred to as baryonic matter, as only the baryons (protons and neutrons) contribute significantly to the mass density.

The majority of baryonic matter in the universe, approximately three-fourths, is currently in the form of ordinary hydrogen, which is the simplest of all elements. The remaining baryonic matter is primarily in the form of helium, which is the next simplest element. This is known as the cosmic abundance of elements, and it can be described, to lowest order, as a mix of three parts hydrogen to one part helium, with only minor contamination by heavier elements. This means that despite the abundance of heavier elements on our planet, they make up only a small fraction of the baryonic component in the universe as a whole.

1.2.2 Cosmic Microwave Background

The Cosmic Microwave Background (CMB), initially discovered by Arno Penzias and Robert Wilson in 1965 using a microwave antenna at Bell Labs, has since been confirmed to be an isotropic background of microwave radiation with a blackbody spectrum of approximately 2.7K, as revealed by the Cosmic Background Explorer (COBE) satellite Smoot et al. (1992); Fixsen et al. (1996). Subsequent missions such as the Wilkinson Microwave Anisotropy Probe (WMAP Bennett et al. (2013); Komatsu et al. (2009); Spergel et al. (2007)) and the Planck satellite have provided more detailed measurements and insights into the early universe, including the small fluctuations in density that serve as the seeds for the formation of large-scale structures. These discoveries have greatly advanced our understanding of the Big Bang theory and the evolution of the universe, provided strong evidence for the expanding universe proposed by Lemaître.

The observations reveal that the Cosmic Microwave Background is exquisitely well fitted by the equation Ryden (2017) of blackbody spectrum,

$$\epsilon(f)df = \frac{8\pi h}{c^3}\frac{f^3 df}{exp(hf/kT) - 1}, \tag{1.2.1}$$

The peak in the blackbody function occurs at $hf \ = 2.82kT$. Integrating over all frequencies, the equation (1.2.1) yields a total density for blackbody radiation at

$$\epsilon_\gamma = \alpha T^4, \tag{1.2.2}$$

where,

$$\alpha = \frac{\pi^2}{15}\frac{k^4}{\hbar^3 c^3} = 7.56 \times 10^{-6} J m^{-3} K^{-4}. \tag{1.2.3}$$

The number density of photons in blackbody radiation can be computed from the equation (1.2.1) as

$$n_\gamma = \beta T^3, \tag{1.2.4}$$

Where,

$$\beta = \frac{2.404}{\pi^2}\frac{k^3}{\hbar^3 c^3} = 2.03 \times 10^7 m^{-3} K^{-3}. \tag{1.2.5}$$

Dividing (1.2.2) by (1.2.4) gives a mean photon energy of $E_{mean} = hf_{mean} \ = 2.70kT$ which is close to the peak in the spectrum.

After studying the cosmic microwave background (CMB) for over two decades, cosmologists discovered that the early universe was not completely smooth as previously thought. The initial surveys didn't detect any irregularities in the CMB, but this led

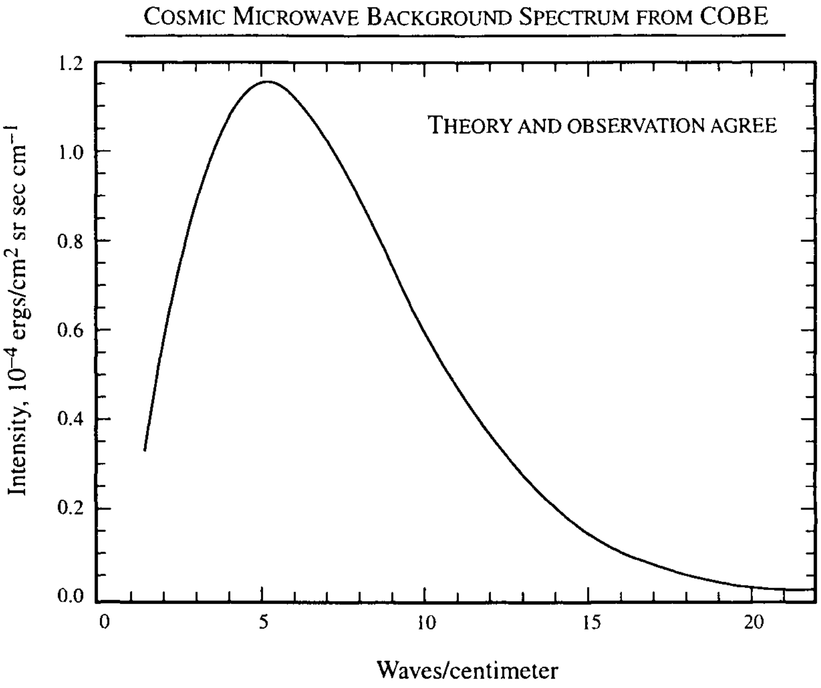

Figure 1.2: Intensity of cosmic microwave radiation as a function of wavenumber from *Far InfraRed Absolute Spectrophotometer* (FIRAS Mather et al. (1994))

to believe that the Big Bang was smooth. Several later studies show small disturbances in the cosmic plasma, which suggests that the universe wasn't completely uniform in its early stages [Allen et al. (1987), Komatsu et al. (2005), Liguori et al. (2010), Mather et al. (1999)]. To understand these perturbations, we need to look beyond the Standard Model.

The cosmic microwave background radiation (CMB) that fills our universe today can be explained as a relic of a time when the universe was sufficiently hot and dense to be opaque. As the universe expanded over time, it became transparent, and the CMB is now observed as a faint, nearly uniform microwave glow filling all of space. The temperature of the CMB is around 2.725 Kelvin White et al. (1994), which is a factor of 1100 lower than the temperature of around 3000 Kelvin at the time when the universe became transparent. This decrease in temperature is a direct consequence of the expansion of the universe, as the wavelengths of electromagnetic radiation are stretched as space expands, resulting in a lower observed frequency and temperature. The cosmic microwave background (CMB) provides us with a unique window into the universe when it was only 300,000 years old. The photons in the CMB were last scattered off electrons at a redshift of approximately 1100. Later on, advanced experiments with higher sensitivity has measured the temperatures with the order of 1 in 10^4 that is $\delta T/T = 10^{-4}$ Wilson & Silk (1981); Peebles & Yu (1970)

The Power spectrum

There have been over 50 experiments to measure the cosmic microwave background (CMB) using various techniques. Telescopes on Earth typically focus on measuring CMB in small patches on the sky due to the interference caused by the Earth's atmosphere. These experiments are often referred to as ground-based experiments. In contrast, balloon experiments and satellites are able to measure the CMB over larger areas of the sky, including the entire sky (or as much as we can see; we cannot see through the galaxy). Balloon experiments, such as BOOMERanG and ARCADE, fly high in the atmosphere to minimize interference from the Earth's atmosphere. Satellite experiments, such as COBE (Cosmic Background Explorer), WMAP (Wilkinson Microwave Anisotropy Probe), and Planck, are able to map out the CMB over larger areas of the sky due to their higher altitudes and broader fields of view. The spatial clustering of the galaxies can be measured by two-point correlation and the power spectrum Neyman & Scott (1952); Zehavi et al. (2005).

These CMB experiments measure the temperature of CMB photons $T(\hat{n})$ as functions of direction on the sky.

$$T(\hat{n}) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} a_{lm} Y_{lm}(\hat{n}) \tag{1.2.6}$$

where $\hat{n}$ is the direction on the sky, a_{lm} are the spherical harmonics coefficients, and Y_{lm} are the spherical harmonics themselves.

By removing the mean temperature over all directions we get

$$\frac{\Delta T}{T} = \Theta(\vec{x}, \hat{p}, t) \tag{1.2.7}$$

Where 'Θ' is the perturbation, $\vec{x}$ is our position, $\hat{p}$ is the direction on the sky the photons hits the detectors and 't' is the time elapsed.

This perturbation (1.2.7) can be expanded using spherical harmonics as,

$$\Theta(\vec{x},\hat{p},t) = \sum_{l=1}^{\infty}\sum_{m=-l}^{l} a_{lm}(\vec{x},t)Y_{lm}(\hat{p}) \tag{1.2.8}$$

The experiments try to accurately measure a_{lm}'s. Expanding $\Theta(\vec{x},\hat{p},t)$ above implies that

$$a_{lm}(\vec{x},t) = \int d\Omega_{\hat{p}} Y^{*}_{lm}\Theta(\vec{x},\hat{p},t), \tag{1.2.9}$$

Expressing 'Θ' in terms of its Fourier components gives,

$$a_{lm}(\vec{x},t) = \int \frac{d^3k}{(2\pi)^3} e^{i\vec{k}.\vec{x}} \int d\Omega_{\hat{p}} Y^{*}_{lm}(\hat{p})\Theta(\vec{x},\hat{p},t) \tag{1.2.10}$$

Inflation predicts that the initial perturbations are very close to being a Gaussian random. Consequently the a_{lm}s will also be a Gaussian random field with mean 0 and some variance C_l,

$$\langle a_{lm}a^{*}_{l'm'}\rangle = \delta_{ll'}\delta_{mm'}C_l \tag{1.2.11}$$

Where $\langle\rangle$ denotes the average over many different realizations of our universes. All the statistical information about it is contained in its power spectrum, C_l. This is because the power spectrum describes the distribution of power in the CMB as a function of angular scale (represented by the multipole moment l). In practice, there may be some deviations from pure Gaussianity depending on the model of inflation. However, these deviations are typically small. According to standard inflationary models with a single inflation potential, a nearly perfect Gaussian spectrum of primordial fluctuations is predicted by many cosmologists Guth & Pi (1982); Mukhanov et al. (1992); Starobinsky (1982); Bardeen et al. (1983). The primordial fluctuations in cosmology, as described by the statistics of the two-point correlation function in real space Georgakakis et al. (2000); Maddox et al. (1990) and power spectrum in Fourier space, provide a comprehensive understanding of Gaussian random fields. This approach has been extensively utilized in the analysis of Large Scale Structure (LSS) observations and Cosmic Microwave Background

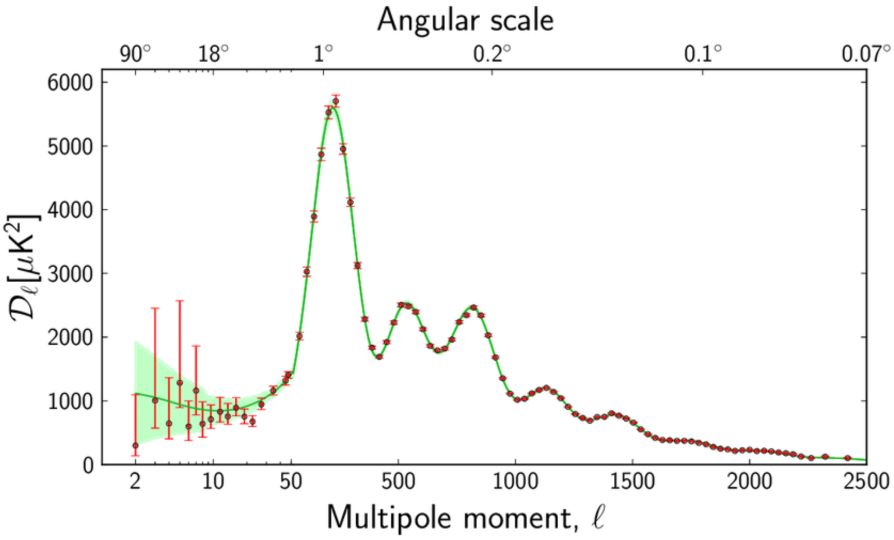

Figure 1.3: *The CMB angular power-spectrum as measured by Planck, the green shaded region and the large red error bars shows the uncertainty for low 'l'*[1]

(CMB) measurements, offering a complete and coherent description of the early universe's distribution and evolution.

For a fixed l, each a_{lm} (the CMB temperature anisotropy coefficients) has the same variance. Moreover, for each value of l, observe that there $2l+1$ possible values of m (the angular momentum quantum number). After measuring the a_{lm}s in an experiment, we can estimate the power spectrum by averaging over these values. This process is known as pixelation or spherical harmonic transform. The power spectrum C_l is given by Challinor & Peiris (2009); Knox (1995)

$$\hat{C}_l = \frac{1}{2l+1} \sum_{m=-l}^{l} |a_{lm}|^2 \tag{1.2.12}$$

The theory is expressed in terms of an ensemble average (average of multiple universes), whereas we can only measure in the single universe that we have. This is where the statistical assumptions come into play. According to this assumption, as long as we have a sufficient number of modes, the mean of these will be equivalent to the ensemble average. As a result, for large l values, we have enough statistical power to accurately estimate C_l, while for very low l values, we may encounter cosmic variance as we have

---

[1] Credit : NASA / WMAP Science Team https://cmb.winterscoming.no/theory_observables.php

very few m's to estimate C_l because we can only observe and measure single universe which we have today. This uncertainty is shown in the green shaded region and large red error bars in the figure (1.3) Hinshaw et al. (2009).

1.2.3 Cosmological parameter estimation

The observational data from various surveys helps us to determine the values of the fundamental parameters that describe the large-scale structure and evolution of the universe. These parameters include the density of matter, dark energy, the expansion rate of the universe, and the geometry of space-time. By comparing theoretical models with observational data, one can constrain the values of these parameters and test different cosmological theories.

In cosmology, when estimating cosmological parameters using the CMB, we are not directly measuring them but rather making estimates based on assumptions about the initial power spectrum Kolb & Turner (1990); Dodelson (2003). This is because the CMB provides information about the initial fluctuations in the universe, which we cannot directly measure. In contrast, when making physical measurements, we are using prior knowledge based on the laws of physics. While other data sets provide experimental information about cosmological parameters, the CMB is particularly valuable because it contains information about the initial conditions of the universe. Ultimately, we would ideally measure cosmological parameters through other means and then use the CMB to determine the initial fluctuations, which could provide insights into the physics of inflation and high energies close to the Planck scale.

There are several techniques involved in cosmological parameter estimation, which can be broadly categorized into two main approaches:

1. Bayesian and Likelihood-based methods
2. Markov chain Monte Carlo (MCMC) methods

Bayes' and Likelihood estimation

These methods involve constructing a likelihood function that describes the probability of observing the data given a set of parameters. The maximum likelihood estimate is then found by maximizing this function. Bayesian methods are a popular variant of likelihood-based methods, which incorporate prior information about the parameters to obtain a posterior distribution that reflects the uncertainty in the estimates.

To estimate cosmological parameters from CMB data, a fundamental concept from probability theory called Bayes' theorem is utilized. In a probability space, given two sets 'A' and 'B' with an intersection $(A \cap B)$, the probability of a specific event belonging to set A also being a part of set B is

$$\frac{P[A|B]}{P[A]} =: P[B|A]. \tag{1.2.13}$$

$P[B|A]$ denotes the conditional probability of 'B' given 'A'. Exchanging 'A' and 'B' we obtain the conditional probability of A given B. Hence,

$$P[A|B] = P[B|A]P[A] = P[A|B]P[B]. \tag{1.2.14}$$

If we fix a cosmological model 'm' that is described by a set of parameters $(\Lambda_1, ..., \Lambda_M) = \Lambda$. In the conducted experiment, a set of measurements $(\mathrm{d}_1, ..., \mathrm{d}_\mathrm{N})$ along with their corresponding errors $(\sigma_1, ..., \sigma_N)$ were obtained. Subsequently, the experiment determined the CMB temperatures as $(n_1, ..., n_N)$. For the given model parameters Λ, we can calculate the predicted outcome of the experiment in terms of expectation value and variance $(d_i(m, \Lambda), \sigma_i(m, \Lambda))$ for each of the data points. Our assumption here is the model is statisical and predicts expectation values for the measurements $d_i(m, \Lambda)$ and their variances $\sigma_i(m, \Lambda)$.

The data are C_l's, the distribution of the data points are Gaussian, the probability of measuring 'd_i' in a model 'm' with parameter values 'Λ' taking into account the measurement of uncertainty 'σ_i' is given by,

$$P[d_i|m, \lambda] = \frac{1}{\sqrt{2\pi(\sigma_i(m, \lambda)^2 + \sigma_i^2)}} exp\left(-\frac{(d_i - d_i(m, \lambda))^2}{2(\sigma_i(m, \lambda)^2 + \sigma_i^2)}\right). \tag{1.2.15}$$

If the measurements 'd_i' are independent and the directions 'n_i' are much farther apart than the beam width and by neglecting correlations, if the data are the a_{lm}, the joint probability of measuring $(d_1, ..., d_N)$ with errors $(\sigma_1, ..., \sigma_N)$ is the product of the individual probabilities,

$$P[\{d_i, \sigma_i\} |n, \lambda] \equiv \mathcal{L}(\{d_i, \sigma_i\} |n, \lambda) = \prod_{i=1}^{N} \left[\frac{exp\left(-\frac{(d_i - d_i(m,\lambda))^2}{2(\sigma_i(m,\lambda)^2 + \sigma_i^2)}\right)}{\sqrt{2\pi(\sigma_i(m,\lambda)^2 + \sigma_i^2)}} \right] \tag{1.2.16}$$

When the measurements are not independent but Gaussian with correlation function,

$$\langle d_i d_j \rangle = C_{ij} \tag{1.2.17}$$

Then the equation (1.2.16) becomes,

$$\mathcal{L}(\{d_i, \sigma_i\} |n, \lambda) = \frac{1}{\sqrt{detC(2\pi)^N}} exp\left(-\frac{d_i C_{ij}^{-1} d_j}{2}\right), \tag{1.2.18}$$

The above function (1.2.18) is called the likelihood function. It gives us the likelihood of the data $\{d_i, \sigma_i\}$ given the model 'm' with parameters 'λ'.

If the posterior probability distribution is not Gaussian, the previously mentioned approximations become inaccurate, and more intricate methods such as Markov Chain Monte Carlo (MCMC) are required to accurately evaluate the posterior distribution and model probabilities.

Markov Chain Monte Carlo (MCMC)

An MCMC involves simulating a Markov chain that explores the parameter space and converges to the posterior distribution. MCMC methods are particularly useful for complex models with many parameters, as they can efficiently explore the high-dimensional parameter space and provide accurate estimates of the posterior distributions.

Consider the scenario as given in the equation (1.2.14), to compute the expectation value and variance of derived parameters,

$$\langle h \rangle = \int h(\lambda) p(\lambda) d^M \lambda, \tag{1.2.19}$$

and

$$\langle (h(\lambda) - (\langle h(\lambda) \rangle)^2 \rangle = \int (h(\lambda) - \langle h \rangle)^2 p(\lambda) d^M \lambda; \tag{1.2.20}$$

Assuming fair sampling of parameter space, $S = (\lambda^1, \lambda^2, ..., \lambda^R$, where number of sampling points $\lambda \in S$ in a small volume 'V' of parameter space is proportional to $\int_V p(\lambda) d^M \lambda$, then we could approximate

$$\langle f \rangle = \int f(\lambda) p(\lambda) d^M \lambda \simeq \frac{1}{R} \sum_{i=1}^{R} f(\lambda^{(i)}). \tag{1.2.21}$$

The marginalized probability densities at $\mu = (\lambda_{i_1}, ..., \lambda_{i_K})$ would be proportional to the sum of all the points for which the parameters of interest are in a given infinitesimal volume around μ, that is,

$$p_K(\mu) = \int (\prod_{j \neq i_r} d\lambda_j) p(\lambda_1, ..., \lambda_M) \simeq \frac{1}{R} \sum_{i=1}^{R} \Theta_{V_\mu}(\lambda^{(i)}). \tag{1.2.22}$$

where V_μ is the volume in parameter space where the parameter values λ_{i_1} to λ_{i_K} are very close to the value μ while all the other parameters are arbitrary. Θ_{V_μ} is Heaviside function on the volume. The marginalized probability distribution of the parameters μ would simply result from projecting the entire probability distribution onto these parameters. In particular, the probability of finding the parameter λ_1 within a small interval I is proportional to the number of points that satisfy the condition $\lambda_1 \in I$ while the remaining parameters are unrestricted.

The identified issues can be effectively resolved by identifying a suitable representative sample from the parameter space. In high-dimensional problems, Markov chain Monte Carlo (MCMC) methods are particularly beneficial due to their effectiveness.

The computation complexity in MCMC and Likelihood estimation is enormous (Lewis & Bridle (2011)). There are several groups using MCMC to estimate parameters reporting over 1000s of hours of CPU usage for semi-realistic simulation. For a semi-realistic simulation similar to Planck LFI 70 GHz, the computational cost of producing one single sample is about ~ 20 CPU-hours and that the typical Markov chain correlation length is ~ 100 samples. The net effective cost per independent sample is ~ 2000 CPU-hours, in comparison with all low-level processing costs of 812 CPU-hours (Eskilt et al. (2023)).

1.3 Geometry, the Metric and Relativity

In this section, we briefly study the geometry While a deep understanding of general relativity is not necessary to learn most of cosmology, having a passing knowledge of the theory is crucial. Some key concepts that are essential for understanding cosmology include the metric, geodesics and Einstein equations which relates the curvature of space-time to the distribution of matter and energy in the universe. By applying these equations to the Robertson-Walker metric we relate the parameters in the metric to the density in the universe.

1.3.1 Curvature

In Newton's view, the universe was Euclidean and space is unchanging. In this concept, mass(m) dictates gravity how to exert a force (F) given by the equation

$$F = -G\frac{Mm}{r^2}, \tag{1.3.1}$$

Force tells mass how to accelerate $F = ma$.

The Einstein concept is, mass-energy tells space-time how to curve. Curved space-time tells mass-energy how to move, curvature is a property of space-time itself.

Consider a triangle on a Euclidean plane constructed by connecting three points with geodesics as shown in the figure (1.4)[2], the sides of the triangle forms angles α, β, γ, they obey the relation,

$$\alpha + \beta + \gamma = \pi, \tag{1.3.2}$$

---

[2] https://en.wikipedia.org/wiki/Shape_of_the_universe

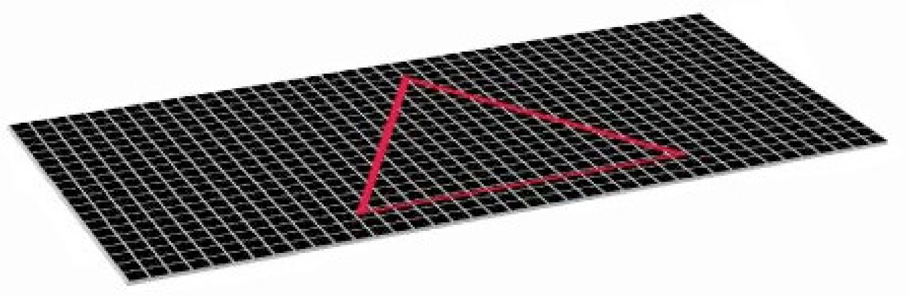

Figure 1.4: A flat two-dimensional space.

Expressing the above equation in a Cartesian coordinate system where the distance (ds) between the two points (x, y) and $(x + dx, y + dy)$ is given by

$$ds^2 = dx^2 + dy^2, \tag{1.3.3}$$

The equation (1.3.3) is represented in a polar coordinate system where the distance between two points (r, θ) and $(r + dr, \theta + d\theta)$ is written as,

$$ds^2 = dr^2 + r^2 d\theta^2. \tag{1.3.4}$$

Both the equations (1.3.3) and (1.3.4) represents the flat space.

Consider a two-dimensional surface of constant negative curvature, with radius of curvature R. If a triangle is constructed on this surface by connecting three points with geodesics, the angles at its vertices α, β, γ obey the relation,

$$\alpha + \beta + \gamma = \pi - A/R^2. \tag{1.3.5}$$

Such surfaces are visualized in the figure (1.5)[3]

Now consider another simple two-dimensional space, on the surface of a sphere which is positively curved as shown in the figure (1.6)[4]. In this case, the geodesic is part of a great circle. The sum of the angles formed by the triangle is given by,

$$\alpha + \beta + \gamma = \pi + A/R^2. \tag{1.3.6}$$

---

[3] https://en.wikipedia.org/wiki/Shape_of_the_universe

[4] https://en.wikipedia.org/wiki/Shape_of_the_universe

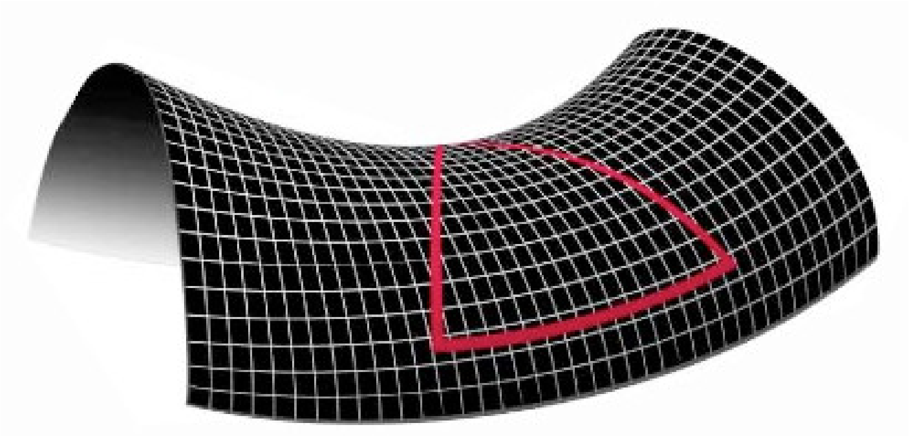

Figure 1.5: A negatively curved two-dimensional space.

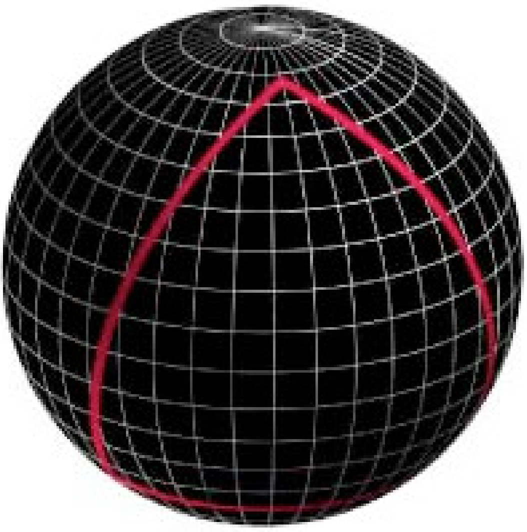

Figure 1.6: A positively curved two-dimensional space.

Where ‘A’ is the area of the triangle, ‘R’ is the radius of the sphere. The triangle constructed on these surfaces doesn’t follow Euclidean geometry. The sum of the three angles of the triangle is greater than 180 degrees are called “positively curved” spaces.

A polar coordinate on the surface of the sphere where ‘r’ is the distance from the North pole and θ is the azimuthal angle measured relative to the prime meridian, then the distance ‘ds’ is given by,

$$ds^2 = dr^2 + R^2 \sin^2(r/R)d\theta^2. \tag{1.3.7}$$

The farthest distance between two points is πR, the surface of the sphere has a finite area $4\pi R^2$. The relation given by the equations (1.3.3,1.3.6 and 1.3.5) are known as **the metrics**, the provide the distance ‘ds’ between the two points in a space

The number ‘K’ is called the curvature constant dictates if space is flat, positively curved or negatively curved depending on the value of K. For a flat space value of $K = 0$, for a positively curved space $K = +1$ and $K = -1$ for a negatively curved space.

We can extend the (1.3.6) to 3-dimensional, isotropic, homogeneous sapce, for all flat-space $K = 0$, positively curved $K = 1$, negatively curved $K = -1$ in a compact form,

$$ds^2 = dr^2 + S_k(r^2)d\Omega^2, \tag{1.3.8}$$

where

$$d\Omega^2 \equiv d\theta^2 + \sin^2\theta d\phi^2, \tag{1.3.9}$$

$$S_k(r) = \begin{cases} R\sin(r/R) & (k = +1) \\ r & (k = 0) \\ R\sinh(r/R) & (k = -1). \end{cases} \tag{1.3.10}$$

Within the limit $r << R$, $S_k \approx r$ independent of the value of k. For a flat space or a negatively curved space, S_k increases monotonically with r, with $S_k \to \infty$ as $r \to \infty$.

Contrary to flat or negatively curved space, in a positively curved space S_k increases to maximum when $S_k = R$ at $r/R = \pi/2$, hence the finite volume.

1.3.2 Robertson-Walker Metric

Einstein's relativity teaches that space-time together comprise a 4-dimensional space. We can compute the 4-dimensional distance between two events in space-time.

Consider two events occurring at $A(t, r, \theta, \phi)$ and at $B(t+dt, r+dr, \theta+d\theta, \phi+d\phi)$. According to special relativity theory, the space-time separation between these two events is,

$$ds^2 = -c^2dt^2 + dr^2 + r^2d\Omega^2. \tag{1.3.11}$$

The metric given in the (1.3.11) is called the *Minkowski metric*, and the space-time defined by Minkowski metric is called Minkowski space-time. A photon traveling through a spacetime is considered a four-dimensional geodesic when photon's trajectory obeys the relation

$$ds^2 = 0 = -c^2dt^2 + dr^2 + r^2d\Omega^2. \tag{1.3.12}$$

If a photo moves radially away or towards the centre then the θ, ϕ remains unchanged,

$$c^2dt^2 = dr^2, \tag{1.3.13}$$

$$dr/dt = \pm c. \tag{1.3.14}$$

The Minkowski metric in the equation (1.3.11) applies when space-time is not curved by the gravity due to mass and energy. This is a special case but in the presence of gravity the case is different. The physicists Arthur Walker and Howard Robertson wanted to investigate whether the geometry of the universe could change over time, given certain assumptions about its homogeneity and isotropy when the space is allowed to expand or contract as a function of time. They independently arrived at the equation now called Robertson-Walker metric. In its general form the RW metric is written as,

$$ds^2 = -c^2dt^2 + a(t)^2 \left[\frac{dx^2}{1 - kx^2/R_0^2} + x^2 d\Omega^2 \right]. \tag{1.3.15}$$

The Robertson-Walker metric can also be written in the form,

$$ds^2 = -c^2dt^2 + a(t)^2 \left[dr^2 + S_k(r)^2 d\Omega^2 \right]. \tag{1.3.16}$$

Where S_k is given by (1.3.10) for three different types of curvature Kiefer & Polarski (2009); Weinberg (1972).

The time variable t in the Robertson-Walker metric is the proper time also known as the cosmic time, which is the time measured by an observer who sees the uniformly expanding universe around him. The spatial variables (r, θ, ϕ) are called the *comoving coordinates* of a point in space. For a perfectly homogeneous, isotropic space comoving coordinates of any point remain constant with time.

While the Robertson-Walker metric provides a useful approximation for describing the large-scale structure of the universe, it does not accurately represent the behavior of matter on smaller scales. On these scales, the universe is ”lumpy” and inhomogeneous, with dense structures held together by various forces such as electromagnetism or gravity. These structures do not expand uniformly like the overall universe, but rather evolve according to their own dynamics. It's only on very large scales, beyond the size of galaxy clusters, that the expansion of the universe can be accurately described by a single scale factor. This insight into the behavior of matter on different scales has important implications for our understanding of the universe's evolution. The expansion of the universe can be homogeneous and isotropic when observed at the larger scales $100Mpc$, where a single scale factor can accurately describe it.

1.3.3 The Friedmann Equation

Nikolai Ivanovich Lobachevski was a Russian mathematician who is considered one of the founders of non-Euclidean geometry. In 1829, he proposed a new type of geometry that did not conform to the traditional Euclidean geometry, which assumes that space is flat and has no curvature. It wasn't until Albert Einstein's theory of general relativity in 1915 that the concept of curved space was linked to its mass-energy content.

Einstein's theory of general relativity proposed that gravity was not a force, as previously thought, but rather a curvature in the fabric of space-time caused by the presence of mass and energy Coles & Lucchin (2002); Peacock (1998); Bernstein (1988). This curvature is described by Einstein's field equation, which relates the geometry of spacetime to its mass-energy content. The Poisson equation holds significance as the primary relationship connecting mass and gravitational force in Newtonian physics. It represents the Newtonian limit of Einstein's field equation and can be expressed as,

$$\nabla^2\Phi = 4\pi G\rho. \tag{1.3.17}$$

where 'Φ' is a gravitation potential at give point is space and the mass density is ρ at that point. The Friedmann equation is a specific form of Einstein's field equations derived in 1922 by Alexander Alexandrovich Friedmann, that applies to homogeneous and isotropic universe. It relates the scale factor '$a(t)$', which describes the expansion or contraction of the universe over time, to the curvature of the universe (described by the parameter 'K'), the radius of the universe 'R_0' at a given time, and the energy density '$\epsilon(t)$' and pressure of the universe '$P(t)$'.

The general scheme to obtain Friendmann equations is from the Einstein's field equations, which is adopted from Mo et al. (2010a). We can arrive at the Friedmann's equation using Newtonian dynamics without bringing relativity into play Liddle & Lyth (2000).

Consider a homogeneous sphere containing matter with total mass 'M_s' constant with time. The sphere is expanding or contracting isotropically, as a result its radius is a function of time $R_s(t)$. Let's bring a test mass 'm' on the surface of the sphere. According to Newton's law of gravity, the force 'F' experienced by the test mass is

$$F = -\frac{GM_s m}{R_s(t)^2}. \tag{1.3.18}$$

We know that $F = ma = m * dv/dt = m * d^2x/dt^2$ from Newton's second law, by using this and replacing for x by the radius R_s, we get,

$$\frac{d^2R_s}{dt^2} = -\frac{GM_s}{R_s(t)^2}. \tag{1.3.19}$$

Integrating and multiplying by dR_s/dt on the both side gives,

$$\frac{1}{2}\left(\frac{dR_s}{dt}\right)^2 = \frac{GM_s}{R_s(t)} + C, \tag{1.3.20}$$

where 'C' is a constant of integration. This is the energy conservation equation. The equation (1.3.20) gives sum of the kinetic energy per unit mass and the gravitational potential energy per unit mass is constant for a given matter at the surface of a sphere.

The kinetic energy per unit mass is given by,

$$E_{kin} = \frac{1}{2}\left(\frac{dR_s}{dt}\right)^2. \tag{1.3.21}$$

The gravitational potential energy per unit mass is given by,

$$E_{pot} = -\frac{GM_s}{R_s(t)}. \tag{1.3.22}$$

The mass within the defined sphere of radius R_s is constant, so we can write,

$$M_s = \frac{4\pi}{3}\rho(t)R_s(t)^3. \tag{1.3.23}$$

Consider a scale factor $a(t)$ which gives the rate at which universe is expanding and is a function of time. The distance between two points is a product of a scale factor and the comoving distance (r_s).

$$R_s(t) = a(t)r_s. \tag{1.3.24}$$

The energy conservation equation can be written in terms of density of mass and scale factor as,

$$\frac{1}{2}r_s^2\dot{a}^2 = \frac{4\pi}{3}Gr_s^2\rho(t)a(t)^2 + C. \tag{1.3.25}$$

Which (1.3.25) can be re-written after dividing on the both side by $r_s^2a^2/2$ in the form,

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\rho(t) + \frac{2C}{r_s^2}\frac{1}{a(t)^2}. \tag{1.3.26}$$

The equation (1.3.26) gives the **Friedmann equation** in its Newtonian form, where relativistic effects are ignored. For $\dot{a} < 0$ is contracting sphere, whereas $\dot{a} > 0$ is expanding sphere which is what we see today.

The expanding universe falls into one of the three cases.

1. $C > 0$ the right hand side of the equation (1.3.26) is always positive, so $\dot{a}^2$ is always positive, implying that the expansion of the universe is indefinite.

2. $C < 0$, the right hand side of the equation (1.3.26) starts out positive but at a maximum scale factor
$$a_{max} = -\frac{GM_s}{Ur_s}, \tag{1.3.27}$$
the right hand side will equal to zero, and expansion stops, as $\ddot{a}$ is negative, so the universe contracts after expanding.

3. $C = 0$, in this case $\dot{a} \to 0$ as $t \to \inf$ and $\rho \to 0$, where expansion slows down and halts completely.

The Newtonian form of Friedmann's equation, which is a simplified version of the full equation in general relativity, neglects the curvature of space-time and assumes a flat (Euclidean) geometry,

By taking relativistic effects into account the Friedmann equation is written as,

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3c^2}\epsilon(t) - \frac{kc^2}{R_0^2}\frac{1}{a(t)^2}. \tag{1.3.28}$$

To obtain the Friedmann equation with relativistic effect, Einstein's method emphasized in assessing the gravitational impact of a particle. According to Einstein, it is crucial to consider not only its 'mass m' but also its energy 'E'. This concept, known as the equivalence principle, asserts that a particle's gravitational force is proportional to its total energy-mass $(E + m)$, rather than solely its 'mass m', which is represented by the formula:

$$E = (m^2c^4 + p^2c^2)^{1/2}. \tag{1.3.29}$$

where 'p' is the momentum of the particle observed. In the context of cosmology, peculiar motion refers to the additional motion that an object(particle, galaxy) has beyond the overall expansion or contraction of the universe.The velocity associated with this peculiar motion is called the peculiar velocity. For a non-relativistic massive particle, its peculiar velocity is much less that 'c', and its momentum 'p' is given by mv, so its total energy is,

$$E_{non-rel} \approx mc^2\left(1 + \frac{v^2}{c^2}\right) \approx mc^2 + \frac{1}{2}mv^2. \tag{1.3.30}$$

In the early universe, when it was hot and dense, particles such as photons and neutrinos also had significant energies due to their high temperatures. This energy contributed to the total energy density of the universe, which was not solely determined by the mass-energy density of matter. As the universe expanded and cooled, these particles became less energetic and their contribution to the energy density decreased, but they still play a significant role in the evolution of the universe. The energy of photons and other relativistic mass less particles are given by,

$$E_{rel} = pc = hf, \tag{1.3.31}$$

Photons, being massless particles, do not contribute significantly to the stress-energy tensor due to their lack of mass, but they still carry energy and momentum that contributes to the curvature of space-time.

In a universe whose expansion is described by the scale factor a(t), there is a linear relation between recession speed $v(t)$ and proper distance $d(t)$ given by Hubble-Leimatre's law, that is,

$$v(t) = H(t)d(t), \tag{1.3.32}$$

where the Hubble constant H_0 is the present-day value of the Hubble parameter where $t = t_0$, $H(t)$, which is related to scale-factor by $H(t) \equiv \dot{a}/a$. Using this we can re-write

Friedmann equation (1.3.28) in the form,

$$H(t)^2 = \frac{8\pi G}{3c^2}\epsilon(t) - \frac{kc^2}{R_0^2 a(t)^2}. \tag{1.3.33}$$

The Friedmann equation (1.3.33) relates the current expansion rate $H(t)$, the current density of energy (ϵ), and the curvature of space-time (k). By combining these equations and observational data, cosmologists can constrain the values of the cosmological parameters. This process is known as cosmological parameter estimation.

The Hubble parameter can give an insight into the *critical density.* For flat space ($k = 0$), the equation (1.3.33) can be re-written as

$$\epsilon(t) \equiv \frac{3c^2}{8\pi G}H(t)^2 \tag{1.3.34}$$

The value of the spatial curvature parameter (k) can be determined by comparing the observed value of the spatial geometry parameter, denoted by $\epsilon(t)$, to the critical density calculated using the Hubble constant and the Friedmann equations. Multiple observations have provided estimates for the Hubble constant within a 10 % uncertainty, which allows for the calculations of the critical density.

$$\epsilon_{c,0} = \frac{3c^2}{8\pi G}H_0^2 = (8.3) \times 10^{-10} Jm^{-3} = 5200 MeVm^{-3} \tag{1.3.35}$$

Its equivalent mass density is,

$$\rho_{c,0} \equiv \epsilon_{c,0}/c^2 = (9.2) \times 10^{-27} kgm^{-3} = (1.4) \times 10^{11} M_\odot Mpc^{-3} \tag{1.3.36}$$

The observations reveal that when averaged over large scales, such as 100 megaparsecs (Mpc) or more, the mean density of the universe is found to be very close to the critical density. This is known as the *cosmological coincidence problem*, as it suggests that the universe is finely tuned to have just the right amount of matter to allow for a flat geometry and slowing of the expansion rate.

1.4 The (ΛCDM) Model

In cosmology, the Lambda-CDM model (ΛCDM) is a theoretical framework that describes the large-scale structure and evolution of the universe. It is based on the combination of two key components: dark matter (CDM, or Cold Dark Matter) and dark energy '(Λ)'. The evolution of the scale factor 'a' with cosmic time 't' is crucial in understanding the history of the universe, as it reveals how the universe expands over time. General relativity provides the connection between this evolution and the energy in the universe.

In early universe, the scale factor a increases rapidly with respect to time, following the trend given by $a(t) \propto t^{1/2}$. This is because at these times, radiation dominates the energy density in the universe. Radiation, which includes photons and other high-energy relativistic particles, has a very low mass density but a high energy density Faber & Gallagher (1979). As a result, it exerts a strong repulsive force that drives the expansion of the universe.

As the universe ages, the dependence of 'a' on 't' changes. At later times, non-relativistic matter, such as atoms and molecules, becomes more dominant than radiation. The energy density of non-relativistic matter is much lower than that of radiation, but it has a much higher mass density. This leads to a slower rate of expansion '$(a(t) \propto t^{2/3})$' as the gravitational attraction between non-relativistic matter slows down the expansion.

One of the earliest pieces of evidence for the existence of dark matter came from the observation of galaxy rotation curves. In the 1970s, astronomers discovered that the outer regions of galaxies rotate much faster than expected based on the visible mass within them. This phenomenon, known as the "flat rotation curve," suggested that there must be additional, invisible mass present in these galaxies. Another piece of evidence for dark matter comes from large-scale structure formation. Computer simulations have shown that the distribution of galaxies in the universe cannot be explained by visible matter alone. The simulations require the presence of a significant amount of non-luminous matter to account for the observed clustering of galaxies. Furthermore, observations of galaxy clusters have provided strong evidence for dark matter. Clusters are large structures made up of many galaxies held together by gravity. The mass of a cluster can be determined by measuring the gravitational lensing it produces on background galaxies.

These measurements have shown that there is much more mass in clusters than can be accounted for by visible matter alone.

Zwicky's discovery of the high velocities of galaxies in the Coma cluster led him to propose the existence of dark matter in the 1930s Zwicky (1933), translated in 2017 Andernach & Zwicky (2017). He observed that the observed velocities of galaxies were too high to be explained by the visible matter in the cluster, and concluded that there must be an additional, invisible form of matter that was exerting a gravitational pull on the galaxies de Swart et al. (2017). This mysterious type of matter, which Zwicky called "dunkle Materie" or "dark matter," was able to hold the galaxies together and prevent them from flying off into space. Zwicky's hypothesis was groundbreaking, as it suggested that the majority of matter in the universe was invisible and unknown, and opened up a new field of research in astrophysics. Further studies by astronomers such as Vera Rubin and Kent Ford in the 1970s Rubin & Ford (1970) provided additional evidence for the existence of dark matter. By measuring the rotation speeds of stars in galaxies such as M33 and Andromeda (M31), they observed that the speeds were higher than expected based on the visible matter within the galaxy. This suggested that there was an additional, invisible form of matter that was exerting a gravitational pull on the stars. Similar observations were made by other astronomers, including Freeman using NGC 300 and M33, who found that the velocities of galaxies in clusters were too high to be explained by the visible matter alone. These observations provided strong support for the hypothesis that dark matter exists and plays a significant role in the structure and evolution of galaxies and clusters Trimble (1987). Today, dark matter is still a major area of study in cosmology, as scientists continue to search for evidence of this elusive substance.

The perplexing cosmological issues, such as the lack of detectable mass surrounding galaxies and clusters of galaxies, as the gravity of normal visible matter is not strong enough to form and hold together the complex structures, as well as the observed accelerated expansion of the universe prompted the development of the two models sCDM (standard Cold Dark Matter) and ΛCDM (Lambda Cold Dark Matter) model of cosmology around the turn of the millennium.

The simplified version of the Cold Dark Matter (CDM) model is sCDM model,

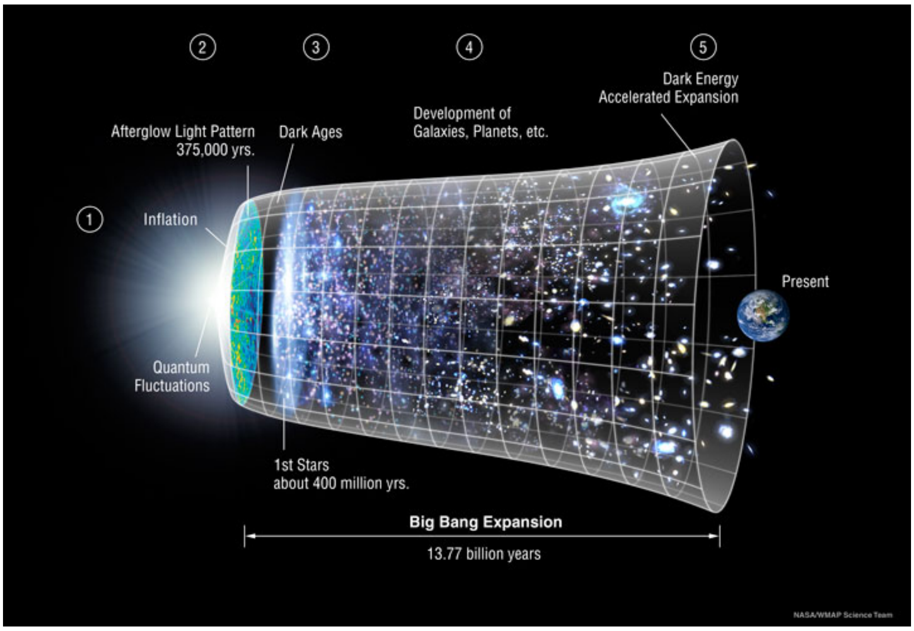

Figure 1.7: *ΛCDM Model of the Universe* [5]

which proposes that dark matter is made up of particles that do not interact with light or other forms of electromagnetic radiation. In sCDM, the universe is assumed to be flat, homogeneous, and isotropic, with a density parameter of unity $\Omega = 1$. The model also assumes that the universe is filled with cold dark matter (CDM), radiation (γ), and that these components dominate the energy density of the universe at different times. ΛCDM (Lambda-Cold Dark Matter) is an extension of sCDM that includes a cosmological constant (Λ), which is a repulsive force that counteracts gravity. This model assumes that the universe is flat, with a density parameter of unity ($\Omega = 1$). The model also assumes that the universe is filled with cold dark matter (CDM), baryonic matter (BM), radiation (γ), and a cosmological constant (Λ). The relative contributions of these components to the energy density of the universe change over time, with CDM dominating at early times, followed by radiation, and then baryonic matter and dark energy becoming dominant at later times.

Both sCDM and ΛCDM models have been successful in explaining many observed features of the universe, such as galaxy formation and evolution, large-scale structure formation, and cosmic microwave background radiation. However, they also have

---

[5] Image Credit: NASA/ LAMBDA Archive / WMAP Science Team https://lambda.gsfc.nasa.gov/education/graphic_history/univ_evol.html

some limitations and unresolved issues, such as the nature of dark matter and dark energy, the origin of large-scale structure formation, and the nature of inflation but ΛCDM is currently considered the more favored model based on a variety of astronomical observations. The ΛCDM model explains the accelerated expansion of Universe. It also predicts small fluctuations in the cosmic microwave background radiation that are in good agreement with observations, while sCDM predicts larger fluctuations that are not observed.

An emerging alternative to dark matter theories is the Modified Newtonian Dynamics (MOND) proposed by Milgrom in 1983 Milgrom (1983) which suggests that different laws of gravity can explain the discrepancy between the matter contents of the universe and the dynamics of galaxies. Famaey and McGaugh Famaey & McGaugh (2012) have also supported this theory. In regions with weak gravitational pull, such as in the outskirts of galaxies or clusters, MOND asserts that Newton's law of gravity becomes irregular, necessitating modifications to explain the phenomenon without the need for dark matter. Recently, a new approach has been suggested that modifies the laws of gravity by assuming the Vainshtein screening mechanism proposed by Platscher Platscher et al. (2018).

Observations from supernovae, cosmic microwave background radiation, and large-scale structure formation all provide strong evidence for the existence of dark energy and support the ΛCDM model.

1.4.1 Benchmark Model

In our previous discussion, we pointed out that a universe composed solely of matter and radiation encounters various issues, which are clearly observed in the astronomical observations. It is called the Benchmark model because it represents a well-established and widely accepted cosmological scenario that is based on a variety of astronomical observations and theoretical considerations. As per the observations, the Benchmark Model, adopts spatially flat universe with three main components:radiation, matter, and a cosmological constant Λ. Radiation refers to the electromagnetic radiation that fills the universe, including the photons (cosmic microwave background radiation), neutrinos.

Matter refers to all types of matter in the universe, including dark matter and baryonic matter (such as protons, neutrons, and electrons). The cosmological constant Λ is a theoretical entity that represents the energy density of empty space and is thought to be responsible for the observed accelerated expansion of the universe.

The current standard model for the universe, known as the ΛCDM model, involves six key parameters Spergel et al. (2007) that have been found to accurately describe the universe. These parameters are,

1. Hubble parameter: This is the rate at which the universe is expanding, measured in kilometers per second per megaparsec ($km/s/Mpc$).
2. Cosmological constant energy-density: This is a form of dark energy Mannheim (2006) that is thought to be responsible for the accelerating expansion of the universe. Its density is denoted by Ω_Λ.
3. Baryons density: This is the density of ordinary matter, such as protons, neutrons, and electrons, in the universe. Its density is denoted by $\Omega_b h^2$.
4. Linear amplitude of density fluctuations σ_8: This is a measure of the initial fluctuations in the density of matter that led to the formation of galaxies Doroshkevich et al. (1978) and other large-scale structures.
5. Index of the power spectrum of primordial density fluctuations: This describes the shape of the initial density fluctuations and is denoted by n_s.
6. Optical depth to last scattering: This describes the amount of time it took for light from the Big Bang to travel through the universe before reaching us today. Its value is denoted by τ

In the Benchmark Model, the majority of the energy density in the universe is not contributed by radiation or matter, but by a cosmological constant. The value of the cosmological constant density parameter $\Omega_\Lambda = 1 - \Omega_{m,0} - \Omega_{r,0} \approx 0.70$. The transition from radiation to matter dominance occurred at a scale factor of approximately $a_{rm} = \Omega_{r,0}/\Omega_{m,0} = 2.8 \times 10^{-4}$, which corresponds to a time of around 4.7×10^4 years. The transition from matter to lambda dominance occurred at a scale factor of approximately

$a_{m\Lambda} = (\Omega_{m,0}/\Omega_{\Lambda,0})^{1/3}$ =0.75, which corresponds to a time of around 9.8 Gyr. The current age of the universe, in the Benchmark Model, is $t_0 = 13.5$ Gyr.

These parameters have been determined through a variety of observational and theoretical methods, including measurements of cosmic microwave background radiation, large-scale structure surveys, and supernova observations. The success of the ΛCDM model lies in accurately describing these observations provides strong support for its validity as a description of the universe's evolution over time.

1.4.2 The Horizon, Flatness problem and Inflation

The horizon problem in the Big-Bang cosmology refers to the fact that the universe appears to be uniform and homogeneous on large scales, despite the fact that light has a finite speed and could not have traveled far enough in the early universe to explain this uniformity. In their groundbreaking works, physicist Stephen Hawking and astronomer Roger Penrose, through their seminal papers on singularities and the inception of universe Hawking W (1966); Penrose (1965), brought forth the horizon issue as a significant obstacle for conventional Big Bang cosmology. This challenge has since been tackled by the influential theory of cosmic inflation, proposed by Alan Guth in the early 1980s Guth (1981).

In the standard Big Bang model, the universe is expanding at a rate that was faster in the past. This means that regions of the universe that are currently far apart were once much closer together. In fact, at very early times, the entire universe was smaller than a single atom today.

The problem arises because light could not have traveled across the entire universe during this early period of rapid expansion. This implies that there should be large-scale differences in temperature and density between different regions of the universe, which are not observed.

During a time interval '*dt*', light can travel a distance of '*cdt*', where '*c*' is the speed of light. This distance is equivalent to a comoving distance,

$$d\xi = c\frac{dt}{a}, \tag{1.4.1}$$

where 'a' is the scale factor. Let's consider a scenario where the comoving distance that light travels from the Big-Bang to a redshift 'z', equivalent to a physical time 't', is equal to the horizon distance 'H',

$$r_{H,comoving}(z) = \int_0^t \frac{cdt}{a(t)}, \tag{1.4.2}$$

as, $\dot{a} = da/dt \implies dt = da/\dot{a} = da/(aH)$,

this gives us,

$$r_{H,comoving}(z) = \int_0^t \frac{cda}{a^2 H(a)} = \int_0^{(1+z)^{-1}} \frac{cda}{a^2 H(a)}. \tag{1.4.3}$$

In a specific cosmic epoch where $z_{eq} >> z >> 0$, the expansion rate 'H' was primarily influenced by the energy density associated with that regime, which was dominated by pressure less matter energy. Consequently, the integral in equation (1.4.3) will have a significant contribution from times or values of a when matter played a dominant role in the Universe. So,

$$H(a) \approx H_0 \sqrt{\Omega_m} a^{-3/2} \tag{1.4.4}$$

Substituting this for the equation (1.4.3),

$$r_{H,comvoing}(z) \approx 2\frac{c}{H_0}\frac{1}{\sqrt{(1+z)\Omega_m}}; z_{eq} >> z >> 0. \tag{1.4.5}$$

Similarly, for the expansion rate 'H' is radiation-dominated at earlier times corresponding to $z >> z_{eq}$, and hence during the time when the universe is dominated by radiation Schneider (2014),

$$H(a) \approx H_0 \frac{\sqrt{\Omega_r}}{a^2}, \tag{1.4.6}$$

Substituting the same for 1.4.3 gives us,

$$r_{H,comoving}(z) \approx \frac{c}{H_0\sqrt{\Omega_r}}\frac{1}{(1+z)}; z >> z_{eq}. \tag{1.4.7}$$

During the very earlier times for larger 'z', the comoving horizon distance was smaller compared to that of recent times.

Consider a proper length $r_{H,pl}$ corresponding to the comoving distance rH, *comoving* at recombination ($z = z_{rec}$), the proper distance is related to comoving distance by,

$$r_{H,pl}(z_{rec}) = ar_{H,comoving}(z_{rec}) = 2\frac{c}{H_0}\Omega_m^{-1/2}(1+z_{rec})^{-3/2}. \tag{1.4.8}$$

Consider a source of radius 'R' which covers a solid angle Ω, the angular diameter distance is given by

$$D_A(z) = \sqrt{\frac{R^2\pi}{\Omega}} \tag{1.4.9}$$

For a model of universe without cosmological constant $\Lambda = 0$,

$$D_A(z) = \frac{c}{H_0}\frac{2}{\Omega_m^2(1+z)^2}\left[\Omega_m z + (\Omega_m - 2)\left(\sqrt{1+\Omega_m} - 1\right)\right]. \tag{1.4.10}$$

In the equation (1.4.9), D_A is taken for the last scattering surface of CMB (recombination), the horizon length will correspond to the angular size on the sky given by,

$$\theta_{H,rec} = \frac{r_{H,pl}(z_{rec})}{D_A(z_{rec})}. \tag{1.4.11}$$

In the case of $\Omega_\Lambda = 0$, for $z >> 1$,

$$D_A(z) \approx \frac{c}{H_0}\frac{2}{\Omega_m z}, \tag{1.4.12}$$

at $z_{rec} \approx 1000$, $\Omega_\Lambda = 0$, we have

$$\theta_{H,rec} \approx= \sqrt{\frac{\Omega_m}{z_{rec}}} = \sqrt{\frac{\Omega_m}{1000}} \simeq \sqrt{\Omega_m}2^0 \tag{1.4.13}$$

This implies that, prior to recombination, a significant portion of the currently observable universe may not have been in communication with other regions as the angle subtended by the horizon length on the sky is about 1^0. Notably, the spatial separation between two regions that were in contact before recombination is much smaller than the size of the universe at recombination. During the time when CMB photons last interacted with matter, disconnected regions have never been able to exchange information, such as temperature data, yet surprisingly, measurements of CMB temperature fluctuations from these areas today exhibit similar fluctuations with a magnitude of approximately $\Delta T/T \sim 10^{-5}$. This scenario is known as the horizon problem.

Inflation provides a solution to this problem by proposing that the universe underwent a brief period of exponential expansion in its early stages. During this period, the size of the universe grew exponentially, allowing regions that were once far apart to come into contact with each other. This process erased any initial differences in temperature and density, resulting in the uniformity we observe today.

Another problem addressed by inflation is the flatness problem East et al. (2016); Guth (1981); Linde (1982); Hawking W (1966). The geometry of the universe is described by a quantity called the spatial curvature, which can be positive, negative, or zero. In the standard Big-Bang model, the curvature is initially very large and decreases over time as the universe expands. However, observations suggest that the curvature is currently very close to zero, which is highly unlikely if it was initially very large.

Inflation provides a solution to this problem by proposing that the universe underwent a brief period of exponential expansion during which it became nearly flat. This process erased any initial curvature, resulting in a nearly flat universe today. Thus in inflationary model, for flat universe, we have,

$$\Omega_\Lambda = \frac{\Lambda}{3H^2} = 1 \tag{1.4.14}$$

In summary, inflation provides a solution to two major problems in the standard Big-Bang model: the horizon problem and the flatness problem. While inflation is still a theoretical concept, it has gained significant support from observations and continues to be an active area of research in cosmology today.

1.5 Baryonic Acoustic Oscillations (BAOs)

BAO (Baryon Acoustic Oscillations) refer to the patterns of fluctuations in the large-scale structure of the universe caused by the oscillations of subatomic particles, known as baryons, during the early stages of the universe's expansion. These oscillations left an imprint on the distribution of matter, which can be observed today as a characteristic distance between galaxy clusters and other large-scale structures. By measuring these distances, astronomers can determine the expansion rate of the universe and provide insights into its evolution over time. BAO measurements Chang et al. (2008) are used in conjunction with other cosmological observations, such as CMB (Cosmic Microwave Background) and supernovae, to test and refine our understanding of the universe's expansion history.

The consistent observation of the comoving baryonic acoustic oscillations (BAOs) length scale spans multiple epochs. The measured angular scale at different redshifts Mao (2012) provides a geometric measure of the expansion history Beutler et al. (2011). During photon decoupling, the relieved pressure allowed for a shell of baryonic matter to form at a fixed radius, which is commonly referred to as the baryon acoustic oscillations (BAOs), where oscillations in BAOs refer to fluctuations in the density of visible baryonic matter, or normal matter, within the Universe. BAOs arise from the same physics that produces the striking peak structure in the cosmic microwave background (CMB) and have a characteristic wavelength determined by the sound horizon at recombination, acting as a '*standard ruler*.' BAOs can contribute to our understanding of the accelerated expansion of the universe by comparing observations of the sound horizon today (using galaxy clustering) to that of the sound horizon at recombination (using the CMB). The Hydrogen-I Intensity Mapping(IM) survey Peterson et al. (2009); Chang et al. (2010) can be utilized in conjunction with BAOs measurements Alam et al. (2017); Chang et al. (2008) in the galaxy power spectrum to investigate large-scale structures in the universe. Specifically, measuring BAOs distance scales can be used to probe the geometry of the universe and constrain its matter content, such as dark energy. IM technique can therefore be employed to study BAOs and deliver bright tomographic maps of neutral hydrogen emission distribution present in the universe today over large angular scales.

1.6 The Halo Model

The Halo model is an analytical model to describe the (dark matter) mass distribution on small, non-linear scales. The fundamental concept behind the halo model is the assumption that the intricate structure of the cosmic web can be fully explained by adding up its constituent parts: dark matter, gas, and galaxies, all distributed within halos. This simplification allows us to tackle the intricacy of the cosmic web by focusing on understanding how to describe the profile, mass distribution, and bias of these halos in relation to the underlying (linear-theory) matter distribution. The challenge lies in accurately populating halos with gas and galaxies or subhalos Mo et al. (2010b). This “mapping information” can be inspired by or extracted from numerical simulations, leading to a versatile analytical model that captures the statistical properties of the cosmic web. This aids in describing the distribution of dark matter density, considering it as a fundamental building blocks. Under the hypothesis that all dark matter is distributed across decoupled halos, which subsequently collapse to form matter bounded gravitational regions amidst the Hubble expansion, it serves as a foundation for understanding dark matter’s behavior in such contexts Cooray & Sheth (2002); Wechsler & Tinker (2018).

$$\rho(r|M) = Mu(r|M) \tag{1.6.1}$$

where $u(r|M)$ is the normalized density profile, r is the distance, M is the halo mass.

$$\int d^3\vec{x}u(\vec{x}|M) = 1 \tag{1.6.2}$$

Imagine a space divided into many small volumes, ΔV_i, which are so small that none of them contain more than one halo center as shown in the fig.(1.8). Let N_i be the occupation number of dark matter haloes in the cell i

Then we have,

$$N_i = 0, 1 \tag{1.6.3}$$

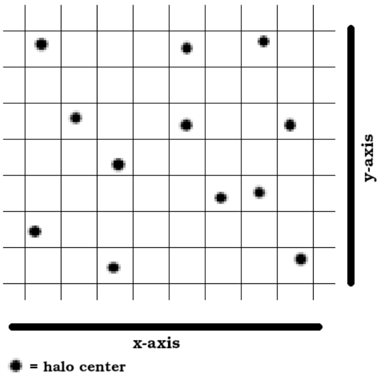

Figure 1.8: halo-grid

so,

$$N_i = N_i^2 = N_i^3 = N_i^N \tag{1.6.4}$$

This allows us to write the matter density field as a summation:

$$\rho(\vec{x}) = \sum_i N_i M_i u(\vec{x} - \vec{x}_i | M_i) \tag{1.6.5}$$

Using two-point statistics, we can write the density field in terms of the halo building blocks,

$$\xi_{mm}(r) \equiv \langle \delta(\vec{x})\delta(\vec{x}+\vec{r})\rangle = \frac{1}{\bar{\rho}^2}\langle \rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle - 1 \tag{1.6.6}$$

where $\delta(\vec{x})$ is the matter field over density.

$$\langle \rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle = \left\langle \sum_i N_i M_i u(\vec{x}_1 - \vec{x}_i | M_i) \cdot \sum_j N_j M_j u(\vec{x}_2 - \vec{x}_j | M_j) \right\rangle \tag{1.6.7}$$

$$\langle \rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle = \sum_i \sum_j \langle N_i N_j M_i M_j u(\vec{x}_1 - \vec{x}_i | M_i) u(\vec{x}_2 - \vec{x}_j | M_j)\rangle \tag{1.6.8}$$

Splitting the above equation into two parts : 1-halo term where $i = j$ and 2-halo term where $i \neq j$, the one-halo term basically correlates pairs of particles (galaxies) resulting from the same halo, while the two-halo term, as its name suggests, relates pairs of particles belonging to two different halos.

For the 1-halo term, $N_i^2 = N_i$, we get,

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{1h} = \sum_i \left\langle N_i M^2 u(\vec{x}_1 - \vec{x}_i|M_i) u(\vec{x}_2 - \vec{x}_i|M_i)\right\rangle \tag{1.6.9}$$

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{1h} = \int dM M^2 n(M) \Delta V_i u(\vec{x}_1 - \vec{x}_i|M) u(\vec{x}_2 - \vec{x}_i|M) \tag{1.6.10}$$

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{1h} = \int dM M^2 n(M) \int d^3\vec{y} u(\vec{x}_1 - \vec{y}|M) u(\vec{x}_2 - \vec{y}|M) \tag{1.6.11}$$

Where the integral,

$$\int d^3\vec{y} u(\vec{x}_1 - \vec{y}|M) u(\vec{x}_2 - \vec{y}|M) \tag{1.6.12}$$

is known as ‘convolutional integral’.

For the 2-halo term, we get,

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{2h} = \sum_i \sum_{j\neq i} \langle N_i N_j M_i M_j u(\vec{x}_1 - \vec{x}_i|M_i) u(\vec{x}_2 - \vec{x}_j|M_j)\rangle \tag{1.6.13}$$

Just as we followed for 1-halo, applying here for 2-halo, we arrive at

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{2h} = \sum_i \sum_{j\neq i} \int dM_1 M_1 n(M_1) \int dM_2 M_2 n(M_2) \Delta V_i \Delta_j \times$$

$$u(\vec{x}_1 - \vec{x}_i|M_1) u(\vec{x}_2 - \vec{x}_j|M_2) \tag{1.6.14}$$

The above equation cannot correctly describe the two-halo term in its current state, it would only hold if dark matter haloes are Poisson distributed but the dark matter haloes themselves are clustered, i.e., have a non-zero two point correlation function.

Thus the clustering of dark matter haloes is characterized by halo-halo correlation function:

$$\xi_{hh}(r|M_1, M_2) = b(M_1)b(M_2)\xi^{lin}_{mm}(r) \tag{1.6.15}$$

where $b(M)$ is the halo bias function. The above description of the halo-halo correlation function is only valid on large (linear) scales! On small scales non-linearities and halo exclusion become important.

$$\langle\rho(\vec{x})\rho(\vec{x}+\vec{r})\rangle_{2h} = \sum_i \sum_{j\neq i} \langle N_i N_j M_i M_j u(\vec{x}_1 - \vec{x}_i|M_i) u(\vec{x}_2 - \vec{x}_j|M_j)\rangle \tag{1.6.16}$$

$$= \sum_i \sum_{j\neq i} \int dM_1 M_1 n(M_1) \int dM_2 M_2 n(M_2) \Delta V_i \Delta_j \times$$

$$[1 + \xi_{hh}(\vec{x}_i - \vec{x}_j|M_1, M_2)] u(\vec{x}_1 - \vec{x}_i|M_i) u(\vec{x}_2 - \vec{x}_j|M_j) \tag{1.6.17}$$

$$= \bar{\rho}^2 + \int dM_1 M_1 n(M_1 \int dM_2 M_2 n(M_2) \times$$

$$\int d^3\vec{y}_1 \int d^3\vec{y}_2 u(\vec{x}_1 - \vec{y}_1|M_1) u(\vec{x}_2 - \vec{y}_2|M_2) \xi_{hh}(\vec{y}_1 - \vec{y}_2|M_1, M_2) \tag{1.6.18}$$

$$= \bar{\rho}^2 + \int dM_1 M_1 b(M_1 n(M_1 \int dM_2 M_2 b(M_2) n(M_2) \times$$

$$\int d^3\vec{y}_1 \int d^3\vec{y}_2 u(\vec{x}_1 - \vec{y}_1|M_1) u(\vec{x}_2 - \vec{y}_2|M_2) \xi^{lin}_{hh}(\vec{y}_1 - \vec{y}_2) \tag{1.6.19}$$

Summarizing the Halo model,

$$\xi(r) = \xi^{1h}(r) + \xi^{2h}(r) \tag{1.6.20}$$

$$\xi^{1h}(r) = \frac{1}{\bar{\rho}^2} \int dM M^2 n(M) \int d^3\vec{y} u(\vec{x} - \vec{y}|M) u(\vec{x} + \vec{r} - \vec{y}|M) \tag{1.6.21}$$

$$\xi^{2h}(r) = \frac{1}{\bar{\rho}^2}\int dM_1 M_1 b(M_1) n(M_1) \int dM_2 M_2 b(M_2) n(M_2) \times$$
$$\int d^3\vec{y}_1 \int d^3\vec{y}_2 u(\vec{x}-\vec{y}_1|M_1) u(\vec{x}+\vec{r}-\vec{y}_2|M_2) \xi^{lin}_{hh}(\vec{y}_1-\vec{y}_2) \tag{1.6.22}$$

The Halo model ingredients are:

1. The Halo density profiles $\rho(r|M) = Mu(r|M)$
2. The halo mass function, $n(M)$
3. The halo bias function, $b(M)$
4. The linear correlation function of matter $\xi^{lin}_{mm}(r)$

1.7 Redshift space distortion

Redshift space distortion is a phenomenon that occurs in the large-scale structure of the universe due to the relative motion of galaxies. As galaxies move through space, their light is shifted towards the red end of the spectrum, a phenomenon known as redshift. This effect is caused by the Doppler effect, which states that moving objects appear to have a shift in their spectral lines.

In redshift space, the apparent positions of galaxies are distorted due to this relative motion. This distortion is known as redshift space distortion (RSD). RSD can be used to study the large-scale structure of the universe and to measure the expansion rate of the universe.

The distortion in RSD occurs because of a peculiar velocity of a galaxy along the line-of-sight Alexander et al. (2012); Postman et al. (1998). This means that galaxies that are actually close together may appear to be farther apart due to their relative motion. Conversely, galaxies that are far apart may appear to be closer together due to their relative motion. The clustering of the galaxies was studied using DEEP2 Galaxy Redshift Survey Newman et al. (2013), SDSS Redshit Survey Zehavi et al. (2011), 2dF Galaxy Redshift survey Colless et al. (2001) and Chandra Deep Field Alexander et al. (2012).

RSD can be used to measure the growth rate of structures in the universe, which is a key parameter in understanding the nature of dark energy and dark matter. By analyzing the distortions in RSD, scientists can also test theories of gravity and cosmology.

In summary, redshift space distortion is a result of the relative motion of galaxies and causes apparent distortions in their positions in space. These distortions can be used to study the large-scale structure of the universe and to test theories of gravity and cosmology.

By following the outline given in (Battye et al. (2013)), to compute the 2-d angular power spectrum of the **HI** by first obtaining the 3-d quantity $\delta\tilde{T}(\vec{r}(z)\hat{n}, z)$, which can be achieved by replacing the **HI** density ρ_{HI} with its linear density field $\delta_{\rho_{HI}}$ in equation (8) of Battye et al. (2013), by adopting different approach to calculate **HI** power spectrum, we consider the real-space brightness temperature fluctuation density contrast as a function of redshift z and position/direction $\hat{\mathbf{n}}$ on the sky,

$$\Delta_{T_b}(z,\hat{n}) = \delta_n - \frac{1}{\mathcal{H}}\left[\hat{n}\cdot(\hat{n}\cdot\vec{\nabla})\vec{v}\right] + \left(\frac{dln(a^3\bar{n}_{HI})}{d\eta} - \frac{\dot{\mathcal{H}}}{\mathcal{H}} - 2\mathcal{H}\right)\delta_\eta + \frac{1}{\mathcal{H}}\Phi + \Psi \tag{1.7.1}$$

where,

$$\hat{n}\cdot(\hat{n}\cdot\hat{\nabla})\vec{v} = \sum_{i,j}\hat{n}_i\cdot(\hat{n}_j\cdot\nabla_j)v_i. \tag{1.7.2}$$

In the equation (1.7.1) the first two terms are respectively, the density and the redshit-space distortion (RSD), the third term results from evaluating the zero-order brightness temperature at the perturbed time corresponding to the observed redshift, whereas the fourth term is the integrated Sachs-Wolfe (ISW) effect, and the last term (arises from $|d\lambda/dz|$) is a result of conversion between radio distance in gas frame ($d\lambda$) with increments in redshift (z).

For the redshift-space distortion term (second term), we first perform some transformation by defining the velocity field in both real and Fourier space,

$$\vec{v}(\vec{r}) = \int\frac{d^3\vec{k}}{(2\pi)^{3/2}}\tilde{v}(\vec{k})e^{i\vec{k}\cdot\vec{r}}. \tag{1.7.3}$$

In this case, the dimension of $\vec{v}(\vec{r})$ is $[LT^{-1}]$, and thus the dimension of $\tilde{v}\vec{k}$ is $[L^3T^{-1}]$. We want to preserve the dimension, so, we define the scalar $\tilde{v}(\vec{k})$ to have the same dimension as $\vec{v}(\vec{k})$ according to Hall et al. (2013),

$$\vec{v}(\vec{k}) = -i\hat{\mathbf{k}}\tilde{v}(\vec{k}). \tag{1.7.4}$$

the Fourier transform of the RSD term is

$$\Delta^{(2)}_{T_b}(z, \hat{n}) = -\frac{1}{\mathcal{H}} \int \frac{d^3\vec{(k)}}{(2\pi)^{3/2}} \left(\tilde{v}(\vec{k})(\hat{k}\cdot\hat{n})^2 k\right) e^{i\vec{k}\cdot\vec{r}} \tag{1.7.5}$$

In summary, redshift space distortion is a result of the relative motion of galaxies and causes apparent distortions in their positions in space. These distortions can be used to study the large-scale structure of the universe and to test theories of gravity and cosmology.

2. Constraining non-Gaussianity $f_{\mathrm{NL}}^{\mathrm{local}}$ using Convolutional Neural Networks

In this chapter, we discuss our work to estimate non-Gaussianity $f_{\mathrm{NL}}^{\mathrm{local}}$ parameter from CMB maps using machine learning, particularly using type of neural network called Convolutional Neural Network (CNN).

2.1 Neural Networks

Neural networks, a branch of machine learning, are the foundation of deep learning techniques. Comprised of layers consisting of input, hidden, and output nodes, each node is linked to others and assigned a weight and threshold. When the output of a node surpasses a predetermined threshold, it becomes activated, transmitting data to the succeeding layer. Otherwise, no information is forwarded to the subsequent layer.

There are several types of neural networks, each with its own unique structure and application. A few examples of the many types of neural networks that exist are Feedforward Neural Network, Recurrent Neural Network, Convolutional Neural Network, Autoencoders, Reinforcement learning and so on. Each type of neural network has its own unique strengths and weaknesses depending on the specific application at hand. In our work we mainly focus on Convolutional Neural Network (CNN).

Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of artificial neural network that is designed to process grid-like data, such as images, videos, and volumes. CNNs are composed of multiple layers, including convolution layers, pooling layers, and fully connected layers. Though CNNs were originally developed for processing image data later they have been adapted for use with other types of data as well as CNNs are particularly effective at extracting spatial features from 2D, 3D input data.

Convolutional Neural Network have three main types of layers, which are,

1. Convolutional layer
2. Pooling layer
3. Fully-Connected (FC) layer

The exact behavior of each layer can vary depending on the specific architecture and hyperparameters used. However, in general, the initial convolutional layers learn to extract simple features such as edges and textures, while later layers learn to combine these features into more complex representations that can help the network to learn spatial relations and patterns.

2.2 Introduction

The Cosmic Microwave Background (CMB) radiation is the snapshot of the photon temperature variation across the whole sky at the time of recombination (around 380,000 years after the Big Bang), which characterises crucial information to understand the early Universe (Scott & Smoot, 2010; Dodelson & Schmidt, 2020). The CMB temperature and polarization anisotropies provide measurable quantities that can be compared with theories of the early universe and constrain cosmological parameters (Komatsu et al., 2009; Hinshaw et al., 2013; Planck Collaboration et al., 2020a). One of the key parameters of interest is the level of primordial non-Gaussianity in the primordial fluctuations, which is captured by non-linear coupling parameter f_{NL} (Komatsu & Spergel, 2001; Planck Collaboration et al., 2016, 2020b) for the deviation to Bardeen curvature perturbation Φ (Φ_{H} in Bardeen 1980), i.e.

$$\Phi(\mathbf{x}) = \Phi_{\mathrm{L}}(\mathbf{x}) + f_{\mathrm{NL}} \left(\Phi_{\mathrm{L}}^2(\mathbf{x}) - \langle \Phi_{\mathrm{L}}^2(\mathbf{x}) \rangle \right), \tag{2.2.1}$$

where Φ_{L} are the Gaussian linear perturbations with zero mean.

There have been several statistical methods developed to estimate the non-Gaussianity parameter f_{NL} from the CMB maps since the era of the Wilkinson Microwave

Anisotropy Probe (*WMAP*). One of the most commonly used methods is bispectrum analysis, which measures the correlations between three different Fourier modes (k_1, k_2, k_3) in the CMB temperature and polarization maps and compares with the theoretical models. *WMAP* used its five-year data and provided an estimate of the local shape of non-Gaussianity $f_{\rm NL}^{\rm local}$ [1] as $-9 < f_{\rm NL}^{\rm local} < 111$ at 95% confidence level (C.L.) (Komatsu et al., 2009). The ESA's *Planck* satellite measured the full-sky CMB fluctuations with higher resolution, and its nominal mission gives $f_{\rm NL}^{\rm local} = 2.7 \pm 5.8$ (Planck Collaboration et al., 2014) and full-mission gives -0.9 ± 5.1 (1σ C.L., Planck Collaboration et al. 2020a,b). In addition, the South Pole Telescope (ground-based CMB experiment) has also measured the $f_{\rm NL}$ using the bispectra method, and gives $f_{\rm NL}^{\rm local} = 420 \pm 350$ (Fergusson et al., 2017). Future surveys such as Simons Observatory (Ade et al., 2019) and CMB Stage-4 (Abazajian et al., 2022) will constrain the $f_{\rm NL}^{\rm local}$ to higher precision.

In addition to the bispectrum analysis of the CMB, there are several other methods to constrain the primordial non-Gaussianity. For example, the Minkowski functionals analysis characterises the topology of the CMB maps using topological descriptors known as Minkowski functionals, and applied the topological estimators to the observed CMB maps (Hikage et al., 2006; Planck Collaboration et al., 2016, 2020b). The peak counting method involves counting the number of peaks in the CMB maps and uses this information to constrain $f_{\rm NL}$ (Marian et al., 2011); and the wavelet bispectrum method decomposes the CMB maps into different wavelet scales and uses the bispectrum of each scale to estimate $f_{\rm NL}$ (Curto et al., 2011; Planck Collaboration et al., 2014). The apart from using CMB maps, there are many studies using large-scale structure data to constrain primordial non-Gaussianity, such as using Baryon Oscillation Spectroscopic Survey's (BOSS) DR16 data (Barreira, 2022; Cabass et al., 2022a,b), using peculiar velocity field scale-dependent bias (Ma et al., 2013) and using 21-cm intensity mapping (Li & Ma, 2017).

While the aforementioned methods have been successful in estimating $f_{\rm NL}$ from

---

[1] The local non-Gaussianity corresponds to the k-space configuration as $k_3 \ll k_2 \approx k_1$, which means the correlations of long-wavelength modes with short-wavelength fluctuations. This $f_{\rm NL}^{\rm local}$ is small and undetectable for single-field slow-roll inflation models (Salopek & Bond, 1990; Falk et al., 1993; Gangui et al., 1994; Maldacena, 2003), but can be large and detectable for multi-field, curvaton scenario (Linde & Mukhanov, 1997; Lyth et al., 2003) or violent, non-linear reheating (Enqvist et al., 2005; Jokinen & Mazumdar, 2006; Chambers & Rajantie, 2008).

CMB maps, they can be computationally expensive and time-consuming, and may require careful modeling of the instrumental effects and other sources of uncertainty in the data. In particular, we want to explore the possibility of using neutral network method of machine learning to directly recognise the non-Gaussianity pattern in the CMB map. In recent years, machine learning techniques, particularly neural networks, have emerged as a promising technique to deal with large cosmological datasets (see, e.g. George & Huerta 2018; Hezaveh et al. 2017; Wang et al. 2022, 2023). Neural networks are able to learn complex patterns and correlations in large datasets, and can provide accurate predictions with relatively low computational cost and time. Furthermore, neural networks can be trained on simulated data, which allows for efficient exploration of the parameter space and testing of different models. Neural networks have the potential to surpass traditional observables like the power spectrum in performance. They can also excel in comparison to other previously established methods, such as those based on non-Gaussian information extraction, like lensing peaks (Gupta et al. 2018.)

In this work, we will exploit the scenario of using neural networks to estimate the primordial non-Gaussianity in the CMB data. We will develop a customized neural network architecture and training process that can handle CMB maps with different levels of primordial non-Gaussianities. We will also test the robustness to the instrumental effects and high-dimensional data representation.

This chapter is organised as follows. In Sec. 3.1, we will review the method we use to generate CMB maps with different levels of local-shape of primordial non-Gaussianities ($f_{\rm NL}$). In Sec. 2.4, we describe the neural networks, loss function and training process. In Sec. 2.5 we present the results of training and model evaluation. The conclusion is presented in the last section.

2.3 Methodology

2.3.1 Training and Evaluation procedure

This paper follows a structured flowchart shown in Fig. 2.1. We first simulate the pure Gaussian CMB maps, and add non-Gaussian CMB maps with different $f_{\rm NL}^{\rm local}$ values to the Gaussian ones to synthesize the simulated maps. We then convolve them with *Planck* beam function and add *Planck* pixel noise. We name them as "Synthesized Maps". These synthesized maps form the training data sets used to train the Convolutional Neural Network (CNN) model. Once the network is fully trained, the CNN model is tested on synthetic data to assess its performance. Subsequently, the trained model is employed to estimate $f_{\rm NL}$ values on another set of simulated *Planck* maps with different $f_{\rm NL}^{\rm local}$ values. This comprehensive approach ensures the model's effectiveness in handling real CMB data and contributes to the understanding of CMB analysis and estimation of $f_{\rm NL}$.

2.3.2 Simulating CMB Gaussian and non-Gaussian Maps

We use CAMB to calculate the temperature angular power spectrum of the CMB, by using the *Wilkinson Microwave Anisotropy Probe* (*WMAP*) five-year data with the then BAO and Type-Ia supernovae best-fitting parameter values: $\Omega_\Lambda = 0.721$, $\Omega_{\rm c}h^2 = 0.1143$, $\Omega_{\rm b}h^2 = 0.02256$, $h = 0.701$, $n_{\rm s} = 0.96$, $\tau = 0.084$ and $\Delta^2_{\mathfrak{R}}(k_* = 0.002\,{\rm Mpc}^{-1}) = 2.457 \times 10^{-9}$ (Komatsu et al., 2009; Dunkley et al., 2009). The reason we use this set of parameters is that we will utilise the non-Gaussian maps generated from the "Optimized Quadrature Scheme" by Elsner & Wandelt (2009), which takes the values in the above set. We then simulate 1000 pure CMB maps from the outputed theoretical angular power spectrum of CMB (C_ℓ) out to $\ell_{\rm max} = 1024$. For the non-Gaussian component of the map ($a^{\rm nG}_{\ell m}$), we utilise the Elsner & Wandelt "Optimized Quadrature Scheme" non-Gaussian maps (Elsner & Wandelt, 2009), and download 1000 of them from their website[2].

We then combine them to compute the synthesized maps with different values of $f_{\rm NL}^{\rm local}$:

$$a_{\ell m} = a^{\rm G}_{\ell m} + f_{\rm NL}^{\rm local} a^{\rm nG}_{\ell m}. \tag{2.3.1}$$

In Fig. 2.2, we show the Gaussian and non-Gaussian maps with different levels of $f_{\rm NL}^{\rm local}$ values. One see clearly that above the level of $f_{\rm NL}^{\rm local} \gtrsim \mathcal{O}(10^3)$ the non-Gaussianity of the map becomes visible. But so far, $a_{\ell m}$ in Eq. (2.3.1) is a theoretical simulation which does not include observational effects. We further convolve the $a_{\ell m}$ in Eq. (2.3.1) with Gaussian beam function and add the *Planck* instrumental noise to mimic the observational map.

---

[2] `http://planck.mpa-garching.mpg.de/cmb/fnl-simulations/`

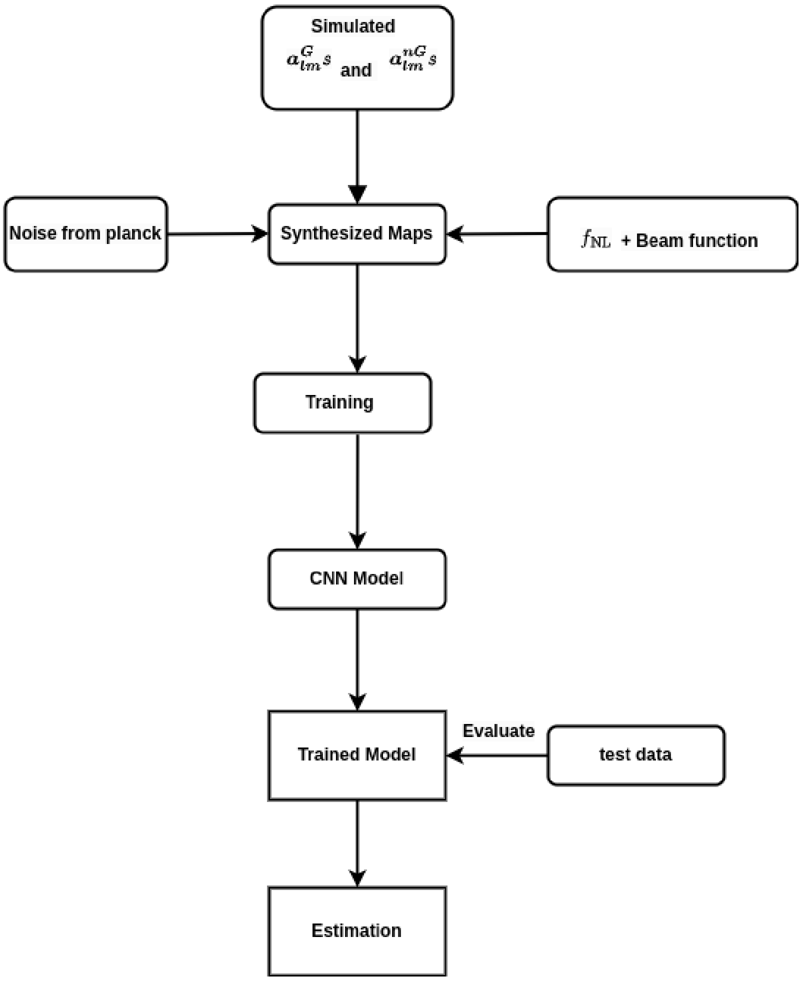

Figure 2.1: *Flowchart depicting the process of generating maps and feeding them into a neural network model, as well as training the model and estimating on test data. The flowchart shows the different stages of the process, including preprocessing, data augmentation, training, and testing, and the different inputs and outputs at each stage. The synthesized maps are the combination of simulated* $a^{\rm G}_{\ell m}$ *and their corresponding* $a^{\rm nG}_{\ell m}$ *with varying* $f_{\rm NL}$ *values, as described by Eq. (2.3.1)*

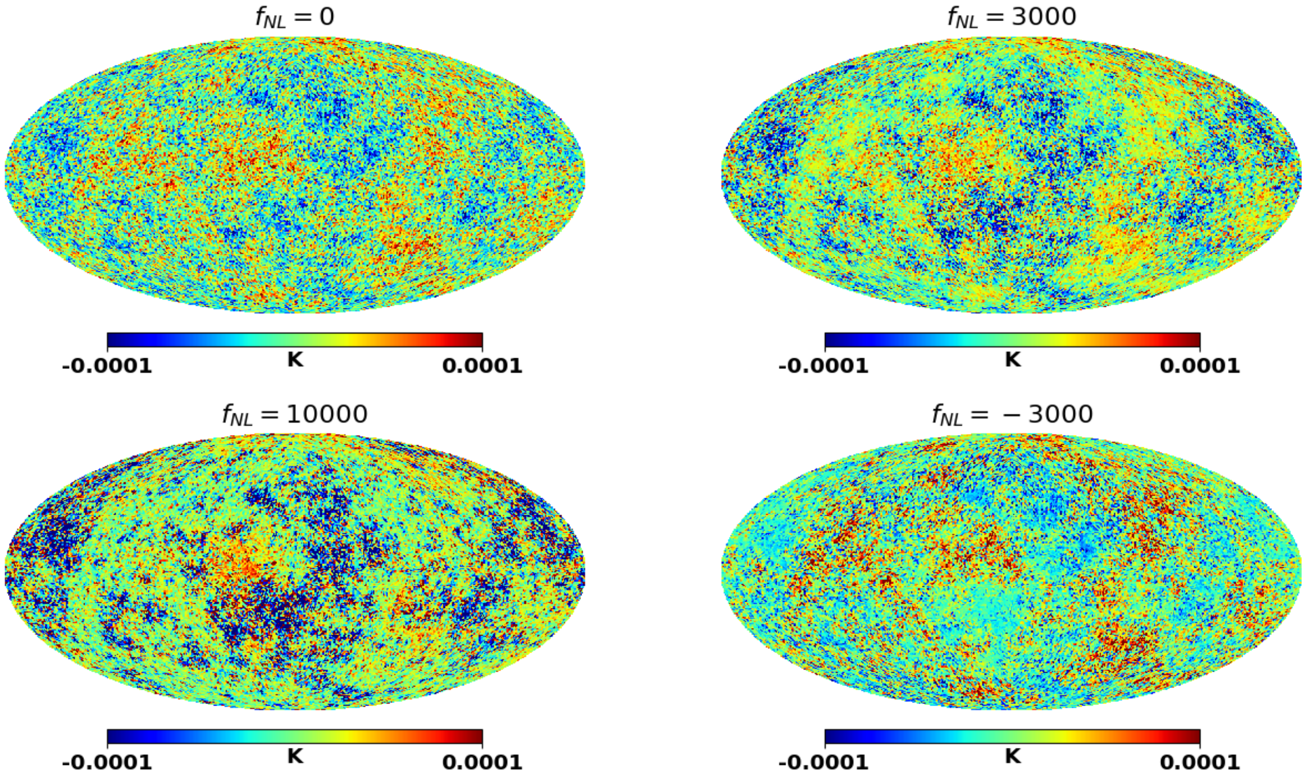

Figure 2.2: *Comparison of CMB maps with different levels of primordial non-Gaussianities (calculated via Eq. (2.3.4)). The different panels show the CMB maps with increasing levels of non-Gaussianity. It becomes obvious that above the level of* $f_{\rm NL}^{\rm local} \gtrsim \mathcal{O}(1000)$ *the non-Gaussianity becomes visible in the CMB maps.*

For beam convolution, we just simply multiply Eq. (2.3.1) with a Gaussian kernel

$$a_{\ell m} \to a_{\ell m} e^{-\ell^2 \sigma_{\rm b}^2/2}, \tag{2.3.2}$$

where $\sigma_{\rm b} = \theta_{\rm FWHM}/\sqrt{8 \ln 2}$, and $\theta_{\rm FWHM} = 5\,{\rm arcmin}$ is the *Planck* Full-Width-Half-Maximum beam size.

2.3.3 Noise

We now simulate the noise of the CMB map. To mimic the true *Planck* mission data, we download the *Planck* maps and use the "Half-Ring Half-Difference (HRHD)" method to estimate the noise power spectrum. Basically, we calculate

$$\mathrm{HD} = \frac{\delta_1 - \delta_2}{2}, \tag{2.3.3}$$

where δ_1 and δ_2 are the splits of the data from *Planck* detectors into two halves. Therefore the difference in Eq. (2.3.3) makes the signal cancel out but keeps the noise fluctuations. We do this practice for both *Planck* SMICA and SEVEM maps, and the resultant HD maps are shown in the upper panels of Fig. 2.3. One can see that the noise across the whole sky is almost homogeneous except for the ecliptic pole region, which is due to more scannings on these two regions.

We then calculate the noise power spectrum (N_ℓ) from the HD maps, and the results are shown in the lower panel of Fig. 2.3. One can see that the noise power spectrum of SMICA is slightly higher than SEVEM map, which might be due to the difference in the component separation methods. We use the resultant noise power spectrum N_ℓ to simulate the noise map ($n_{\ell m}$) and add to the convolved CMB map to resemble the total (simulated) CMB map

$$a^{\rm tot}_{\ell m} = a_{\ell m} e^{-\ell^2 \sigma_{\rm b}^2/2} + n_{\ell m}, \tag{2.3.4}$$

which ensures the training set is as close to the true *Planck* map as possible. In Fig. 2.4, we show the comparison between theoretical CMB power spectrum $C_\ell^{\rm TT}$ (blue solid line), convolved spectrum $C_\ell^{\rm TT} B_\ell^2$ (black solid line) and the estimated power spectrum from simulated maps (yellow dots). One can see that the power spectrum from the simulated

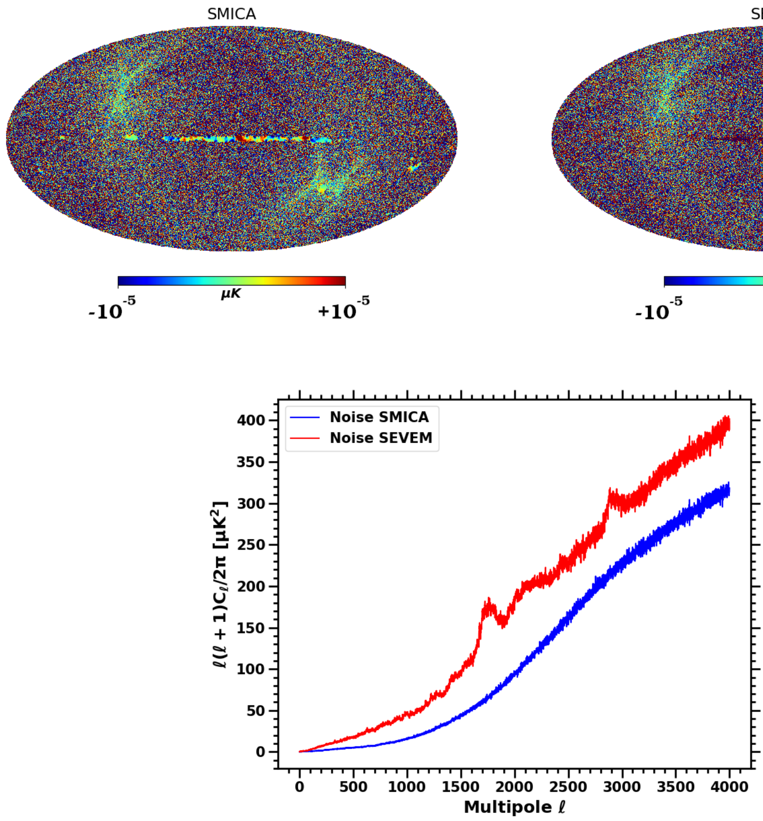

Figure 2.3: *Top Left* and *Top Right* panels are the noise maps of *Planck* SMICA and SEVEM CMB maps, which are obtained by using the "Half-Ring Half-Difference (HRHD)" method. *Bottom* panel compares the noise power spectra of SMICA and SEVEM maps.

map is very close to the convolved CMB power spectrum (black line), which is because on scales of $\ell \leq 1000$, the noise level is not very high compared to the intrinsic CMB level. We nonetheless include the noise simulation for the accuracy of the study.

2.4 Convolutional Neural Network

2.4.1 Single Neuron

A typical single neuron used in machine learning is a fundamental building block of artificial neural networks. It represents the basic computational unit that processes inputs and produces an output as shown in Fig. 2.5. The neuron receives inputs from various

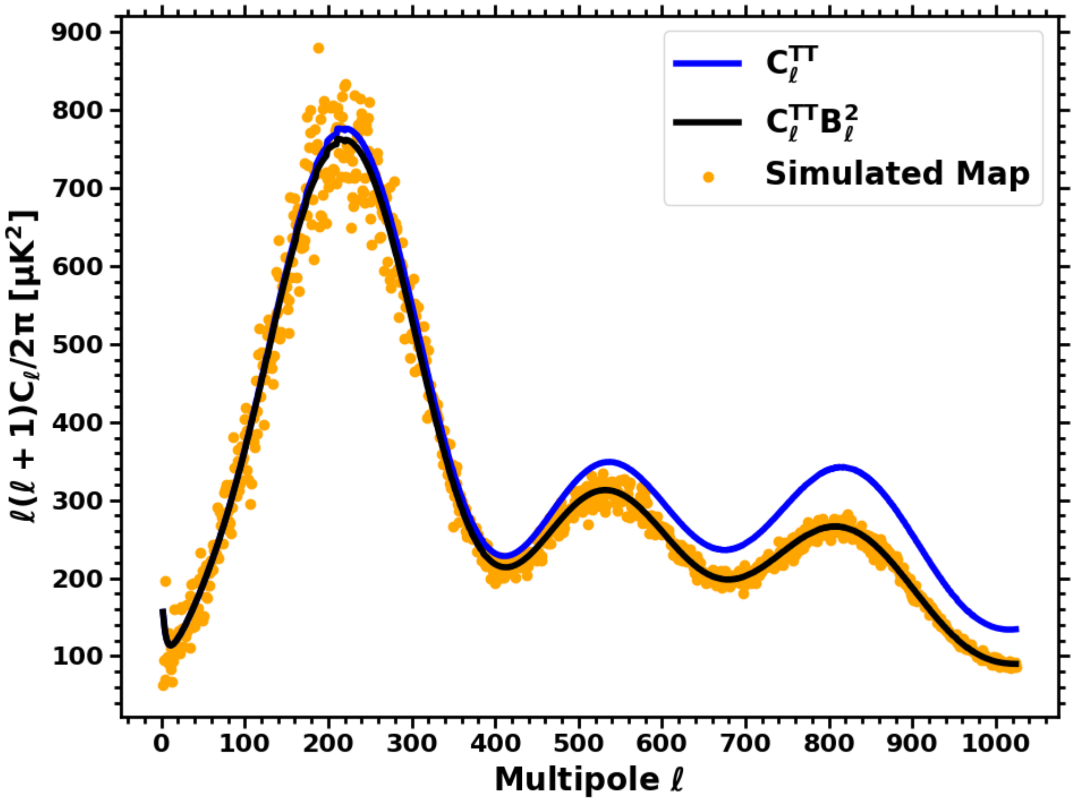

Figure 2.4: *Comparison between theoretical CMB power spectrum* C_ℓ^{TT}*, convolved spectrum* $C_\ell^{\mathrm{TT}} B_\ell^2$ *and the estimated power spectrum from simulated maps.*

sources, each multiplied by corresponding weights. These inputs are then summed together, including a bias term. The weighted sum is passed through an activation function to introduce non-linearity and determine the neuron's output.

The mathematical representation of a single neuron in a neural network is given by,

$$p_i = \sum_{i=1}^{n} w_i x_i + b. \tag{2.4.1}$$

The neuron takes input signals $(x_1, x_2, ..., x_n)$, and computes a weighted sum of these inputs, denoted by the symbol p_i. The weights used in the summation are denoted by $(w_1, w_2, ..., w_n)$, and the bias term is denoted by b.

The input signals $(x_1, x_2, ..., x_n)$ can represent any number of features or attributes that describe the input data, such as pixel values in an image. The weights $(w_1, w_2, ..., w_n)$ determine the strength of the connection between each input signal and the neuron. Alternatively, one can think of the weights as representing the importance of each input feature for the task at hand.

The bias term, b, is an additional parameter that is added to the weighted sum. It can be regarded as the representation of the neuron's willingness to fire even when all input signals are zero. In this sense, the bias term provides a measure of the neuron's overall activity level.

Once the weighted sum p is computed, it is typically passed through an activation function, such as the sigmoid function or ReLU (Rectified Linear Unit) function, to introduce non-linearity into the model and allow the neuron to model complex mappings between the input and output.

2.4.2 Neural Network

A typical neural network is composed of multiple layers of interconnected neurons depicted in Fig. 2.5, where each neuron performs a specific computation. These neurons are organized into different layers, including input layer, hidden layers, and output layer. A typical neural network is shown in the Fig. 2.6.

The input layer of a neural network receives the raw input data (deep green dots on the left side of Fig. 2.6). Each neuron in the input layer represents a specific feature

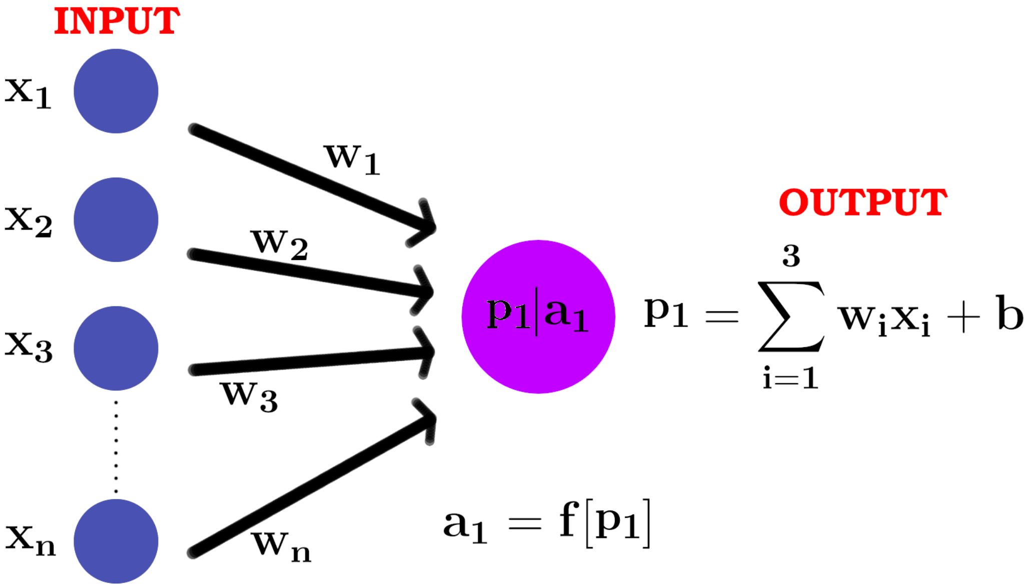

Figure 2.5: *A single neuron with an input and its corresponding output. The input is multiplied by a weight, which is then added to a bias term. The resulting value is passed through an activation function, which produces the output of the neuron.*

or attribute of the input data. The number of neurons in the input layer corresponds to the dimensionality of the input data.

Hidden layers are intermediate layers between the input and output layers (shallow green dots in the middle of Fig. 2.6). They are responsible for learning and representing complex patterns and relationships within the data. Each neuron in the hidden layers receives inputs from the previous layer, applies a weighted sum of these inputs, and passes the result through an activation function. The number of hidden layers and the number of neurons in each hidden layer can vary depending on the complexity of the problem and the desired model capacity.

The output layer represents the final prediction or output of the neural network (yellow dots on the right side of Fig. 2.6). It consists of neurons that produce the desired output based on the learned representations from the hidden layers. The number of neurons in the output layer depends on the specific task. In our work, the number of output neurons corresponds to the single output neuron, because it is a regression task.

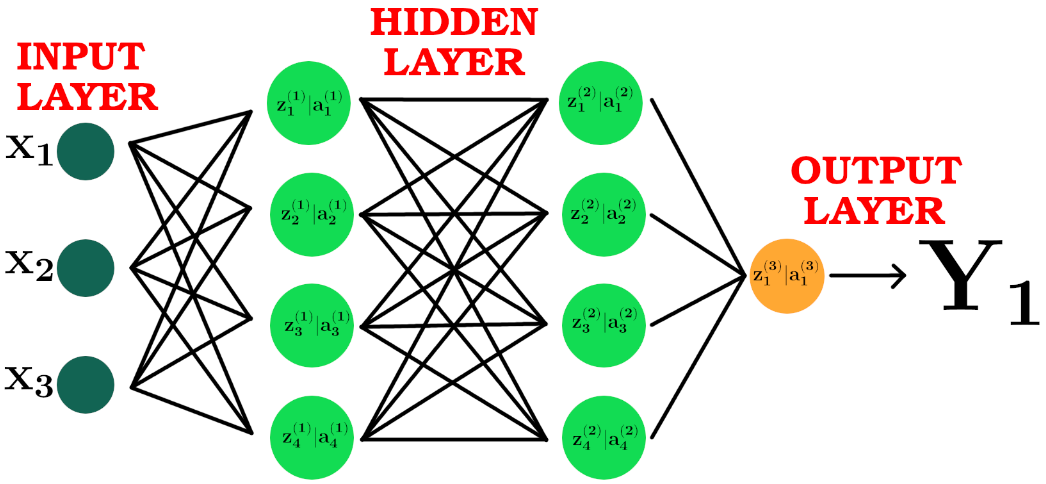

Figure 2.6: *The image depicts a neural network with an input layer, multiple hidden layers, and an output layer. Each circle represents a neuron, and the lines connecting them represent the connections between neurons. The input layer takes in the input data, which is then processed through the hidden layers using various activation functions. Finally, the output layer produces the final result, which could be a classification or prediction based on the input data. The image demonstrates the complexity of neural networks and how they can be used for tasks such as image recognition, natural language processing, and more.*

2.4.3 Convolutional Neural Network (CNN)

For our regression work, we use CNN model to carry out the regression task. Our input data is 2D array of CMB temperature maps. The model is using the Keras deep learning library's Sequential API which is designed to make it easy to create neural networks consisting of multiple layers. This specific neural network model includes convolutional layers, pooling layers, and fully connected layers, which are typical for image related problems.

A CNN is a deep learning architecture composed of multiple layers of interconnected neurons, including the typical single neurons described earlier. A CNN is specialized type of neural network that is specifically designed for processing structured grid-like data, such as images or sequences, and has become highly effective in various computer vision tasks.

CNNs utilize specialized layers, namely convolutional layers and pooling layers, which are not typically found in traditional neural networks. Convolutional layers apply convolutional operations to the input data, using learnable filters or kernels to extract local features. These filters slide over the input, computing element-wise multiplications and summations. Pooling layers then downsample the feature maps obtained from the convolutional layers, reducing their spatial dimensions while preserving important features. These layers enable CNNs to capture spatial hierarchies and translation-invariant representations. The earlier layers of a CNN learn low-level features, such as edges or corners, while deeper layers learn higher-level features, such as textures or object shapes. This feature learning capability allows CNNs to effectively extract and represent complex patterns in the CMB data, making them well-suited for our regression task. The following Fig. 2.7 shows a general CNN structure.

Our CNN model used for regression consists of 10 layers including the input layer, convolutional layers, dense layers, and the output layer. The breakdown of the layers are:

- Input Layer: The model expects input images with a shape of $(256, 192, 3)$. This means the input array must have a height of 256, width of 192, and three channels. The input layer with appropriate shape is represented in rectangular red box in

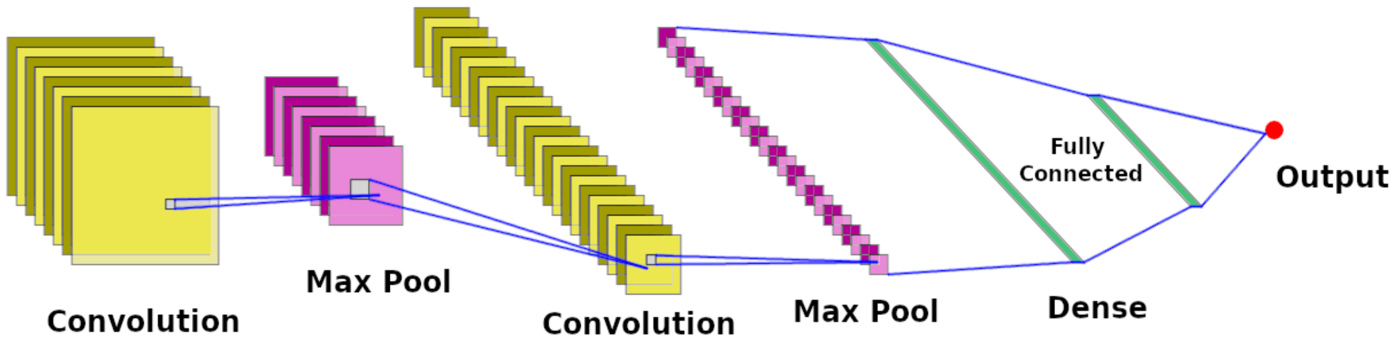

Figure 2.7: *This is a diagram of a convolutional neural network (CNN) structure. It includes input data, convolutional layers, activation functions, pooling layers, fully connected layers, and an output layer. The input data is passed through the convolutional layers, which extract features from the input data. Activation functions introduce non-linearity to the network, and pooling layers downsample the output of the convolutional layers. The fully connected layers take the output of the pooling layers and produce a final output. The output layer is responsible for producing the final prediction based on the input data.*

Fig. 2.7.

- Convolutional Layers:
    1. The first convolutional layer has 32 filters of size $(3, 3)$ and uses the ReLU activation function. It takes the input image shape. The convolutional layer is part a of hidden layer which is represented in Fig. 2.7 in yelow rectangular boxes.
    2. A max-pooling layer with a pool size of $(2, 2)$ follows each convolutional layer. It is also a part of a hidden layer. This layer downsamples the feature maps obtained from the previous layer. A max pool layer is represented in pink rectangular box immediately after convolutional layer represented in Fig. 2.7. It is possible to have a convolutional neural network (CNN) without max pooling layers. While max pooling is a common technique used in CNNs to reduce the spatial dimensions of feature maps and prevent overfitting, it is not an essential component of a CNN.
- Fully Connected Layers:
    1. After the convolutional layers, a flatten layer is added to convert the 2D feature maps into a 1D vector.
    2. Then, a series of dense also known as fully connected layer (green shaded region in Fig. 2.7) layers follow, with 256, 128, and 64 neurons, respectively. Each dense layer uses the ReLU activation function to introduce non-linearity and capture complex relationships in the data.
    3. The last dense layer has a single neuron with a linear activation function, which makes it suitable for regression tasks.
- Model Output: The final layer of the model (represented as red dot in the Fig. 2.7) consists of a single neuron with a linear activation function. This indicates that the model is designed for regression tasks, where it aims to predict a continuous numerical value.

The given CNN model has approximately $1,479,721$ trainable parameters for the input of shape $(256, 192, 3)$.

When the model is compiled using the method, we specify the loss function, optimizer, and evaluation metrics we will use for the training. The optimization algorithm sends error signals, usually in the form of gradients, back through the network to fine-tune the weights of the nodes in the network, which is the essence of backpropagation. The compiling method sets up the backpropagation algorithm to be used during the training process.

Let L be the loss function, p be the output of a particular layer, x be the input to that layer, w be the weights of that layer, b be the bias of that layer, and δ be the error signal at that layer. The back propagation algorithm can be summarized as follows:

1. First, compute the error signal δ for the output layer,

$$\delta = \frac{\partial x}{\partial p}. \tag{2.4.2}$$

2. Then, compute the error signal δ for the previous layer using the current layer's error signal,

$$\tilde{\delta} = w^T \cdot \delta \cdot g'(x). \tag{2.4.3}$$

   where $g'(x)$ is the derivative of the activation function used in the previous layer.

3. The gradient of the loss function with respect to the weights and biases of the current layer are

$$\begin{aligned} \frac{\partial L}{\partial w} &= \delta \cdot x \\ .\frac{\partial L}{\partial b} &= \delta. \end{aligned} \tag{2.4.4}$$

4. Then, update the weights and biases of the current layer using an optimization

algorithm, such as stochastic gradient descent:

$$\begin{aligned} w_{\text{new}} &= w - \eta \cdot \frac{\partial L}{\partial w} \\ b_{\text{new}} &= b - \eta \cdot \frac{\partial L}{\partial b}, \end{aligned} \tag{2.4.5}$$

where η is the learning rate, which is a hyperparameter that determines the step size of the optimizer while updating the weights of the neural network during training. Typically, η is a constant throughout the training process but can be changed either manually or automatically during the training process. Therefore, η is a critical parameter that needs to be tuned to achieve the best performance of the model.

The above steps are repeated for all previous layers in the neural network. The above equations are then used iteratively for multiple epochs until the loss function converges and the neural network is trained. In machine learning, an epoch signifies a complete iteration through the entire dataset during the training phase. Put simply, it involves presenting all the training examples to the model once. The back propagation algorithm is used in our CNN model to adjust the weights of the neural network to minimize the specified loss function and improve the model's performance on the evaluation metrics.

2.4.4 Loss function

In machine learning, a loss function is used to measure the difference between the predicted outputs of a model and the actual values. The goal of the model is to minimize this loss function during training, which improves the accuracy of predictions.

The choice of loss function depends on the type of problem being solved. For regression problems, where the output is a continuous value, the mean squared error (MSE), the mean absolute error (MAE), or the Huber loss is commonly used.

Below are the two commonly used loss functions

- Mean Square Error (MSE).

$$l(y_i, y_o) = \frac{1}{n}\left(\sum (y_i - y_o)\right)^2. \tag{2.4.6}$$

- Mean Absolute Error (MAE)

$$l(y_i, y_o) = \frac{1}{n} \sum |y_i - y_o| \,. \tag{2.4.7}$$

where $\mathrm{y_i}$ is the input and $\mathrm{y_o}$ is the output.

These loss functions are used during training to optimize the parameters of the neural network in order to minimize the difference between the predicted and actual outputs.

2.4.5 Data Preparation

z-scaling

After obtaining the simulated $a_{\ell m}$ maps through Eq. (2.3.4), we convert the $a_{\ell m}$s into a full-sky map with $n_{\mathrm{side}} = 64$, resulting in a float array of size 49,152. The histogram of the array values is very close to Gaussian distribution with values in order of $\mathcal{O}(10^{-5})$. Because of the small values in the array which might be insensitive to CNN recognition, we employ the z-scaling, which preserves the Gaussian shape of the array while amplifying the value of each data point. This ensures that the CNN can effectively recognize and learn from the changes in the values, even though they are initially very small. As a result, the z-scaling technique allows for better sensitivity and interpretation of the data by the CNN.

We apply the z-scaling as follows:

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{2.4.8}$$

z_i is the standardized value (z-score) of the ith data point, x_i is the original value of the data point, μ is the mean of the data distribution in the given array which is z-scaled, and σ is the standard deviation of the data distribution in the given array. The mean value of the array for one of the sample map before scaling was 4.822×10^{-8} and the standard deviation was 3.91×10^{-5}. After applying z-scaling, the mean value of the array shown in the Fig. 2.8 has been shifted very close to zero, $\mu = -2.02 \times 10^{-17}$ and the standard deviation has been scaled to $\sigma = 1$. This transformation preserves the shape of the distribution, meaning that the relative positions and spread of the data points

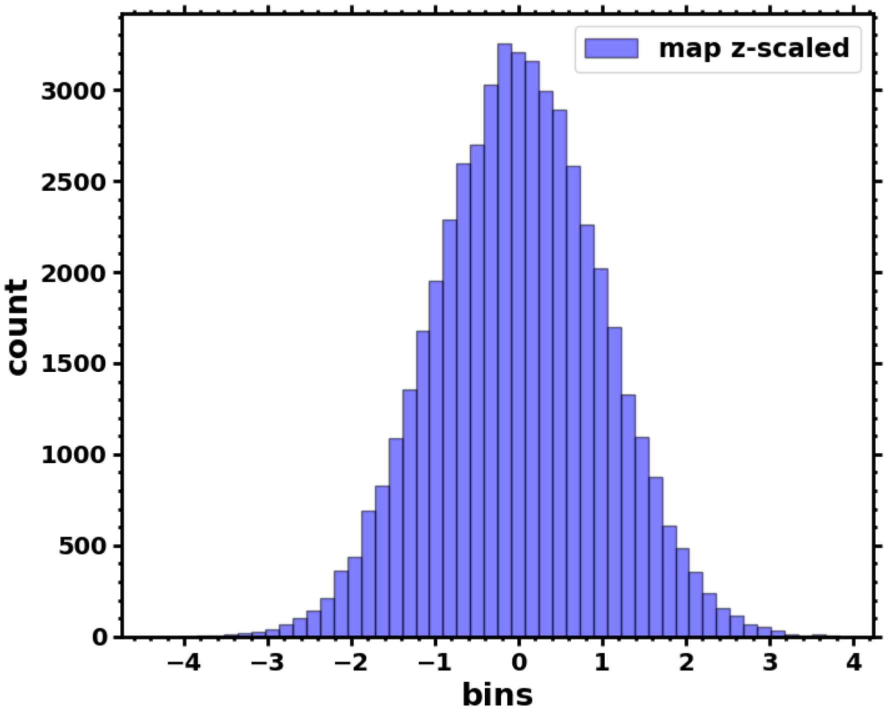

Figure 2.8: *This histogram plot shows the distribution of array values after applying z-scaling with the mean and standard deviation corresponding to each individual map. The x-axis represents the bin range, while the y-axis shows the frequency of values in that bin. The plot reveals the spread and shape of the data distribution for each map array. The plot shows a roughly symmetric distribution with a peak around zero, indicating that the mean has been shifted to zero. This type of preprocessing is often applied to input data in machine learning models to improve training performance and accuracy.*

remain unchanged, but they are now measured in standard deviation units instead of their original units.

This process helps to amplify the values and make them more discernible to the CNN, while maintaining the Gaussian distribution of the data. The z-scaling is applied to the each map in the training dataset separately. This approach ensures that the array values are more sensitive to the CNN model. By applying the scaling independently to each map, we prevent any data leakage between different maps, which could potentially compromise the performance of the model on the unseen data. By treating each map independently during the scaling process, we maintain the integrity of the training dataset and ensure that the CNN learns the patterns and variations specific to each map. This approach helps the model generalize better to the new and unseen dataset, as it has not been exposed to any information from other maps.

The histogram in the Fig. 2.8 illustrates the distribution of values in the array. This scaling process ensures that the CNN can effectively capture the subtle variations and patterns within the data, enabling better interpretation and learning.

2D Conversion

Because the power of the CNN lies in its ability to capture shapes, features, and depths in images, to leverage this capability, we utilize the healpix module to project the one-dimensional array into a two-dimensional shape.

The array after z-scaling is still a 1D array, which is then converted into 2D map. For a map of $n_{\text{side}} = 64$, the array has $49,152$ elements. We transform the effective map into a two-dimensional array of size $(256, 192)$, where the values from the z-scaled array are preserved without any clipping or truncation of data points. This approach ensures that no information is lost compared to conventional methods that convert the map into a mollweide projection or a PNG image with a white background. Such conventional methods can introduce systematic errors and distortions to the data, compromising its accuracy.

By directly converting the map into a two-dimensional array while preserving the z-scaled values, we maintain the integrity of the data, allowing the CNN to effectively learn and extract meaningful features for improved analysis and prediction. This process gives us the map of size (256, 192).

Array stacking

The 2D conversion of the map results in an array of size (256, 192), which is suitable for a CNN to learn features and shapes within the data. However, it does not capture the depth information. In Fig. 2.2, we can observe that different values of f_{NL} uniformly change the values of the array, but the underlying features remain consistent. The features of the map are altered when different realizations of the simulated maps are combined linearly. This would have been good to estimate f_{NL} within the same realization because features will not change for the same realization of the CMB map, but this does not suit our application because we have different realisations of CMB.

By adopting this approach, we enable the CNN model to capture the intricate details of the features, as well as the varying levels of intensity or depth within the data. This multi-channel representation enhances the CNN's ability to extract meaningful patterns and to better the predictions based on the combined information from all three channels.

2.4.6 Training

The dataset of 10,000 maps was generated by combining Gaussian and non-Gaussian maps provided by Elsner & Wandelt (2009), which represent different realisations. We then multiply a range of $f_{\mathrm{NL}}^{\mathrm{local}}$ values in between -50 to 50 to the $a_{\ell m}^{\mathrm{nG}}$ maps and add them to the Gaussian CMB maps (Eq. (2.3.1)).

During the training process, the model's parameters are adjusted to minimize the difference between the predicted and actual values of the $f_{\mathrm{NL}}^{\mathrm{local}}$ maps in the training dataset. This is done using a technique called backpropagation, which is a type of gradient descent algorithm. The validation dataset is used to monitor the model's performance during training and prevent overfitting, which occurs when the model fits too closely to the training data and performs poorly on new, unseen data. Thus, we split the initial 10000 maps into training, validation and the test dataset.

By doing the splitting, and test sets, we can ensure that the model is trained on a diverse set of inputs and can make accurate predictions for unseen data. This is an important step in developing a reliable and accurate model for predicting $f_{\rm NL}^{\rm local}$ values from CMB maps. The dataset of 10,000 maps is split into 9500 for training and 500 for testing, here to evaluate the model's true performance 500 maps were set aside as a test dataset which are unseen during the training and validation phases. While training, 10% of the training data was assigned for validation dataset that is 950 and the remaining 8,550 maps were used for as the training dataset.

During training, two loss functions were considered: mean square error (MSE) and mean absolute error (MAE). After assessing the model's performance, mean absolute error was selected as the preferred loss function, yielding slightly better results. Extensive hyperparameter tuning was performed to optimize the model, including batch size (256) and learning rate (0.001). Early stopping and model checkpoints were employed to prevent overfitting and save the best-performing weights. The training process was initially set to run for 1,000 epochs, but concluded at 909 epochs when no significant improvement in validation loss was observed. The weights from the epoch with the best performance were saved for subsequent evaluation on the test dataset.

This meticulous training approach ensures that the model is optimized and capable of accurately estimating $f_{\rm NL}$ on previously unseen data. Here, unseen data refers to the data that is not used in training the CNN like test dataset.

2.5 Results

In this section, we present the results obtained from our trained model.

2.5.1 Training and Validation loss curve

One important factor that indicates a well-trained model is the behavior of the training and validation loss curves throughout the learning process. These curves provide valuable insights into the model's performance, including close scrutiny of either underfitting or overfitting behaviors that require further adjustments.

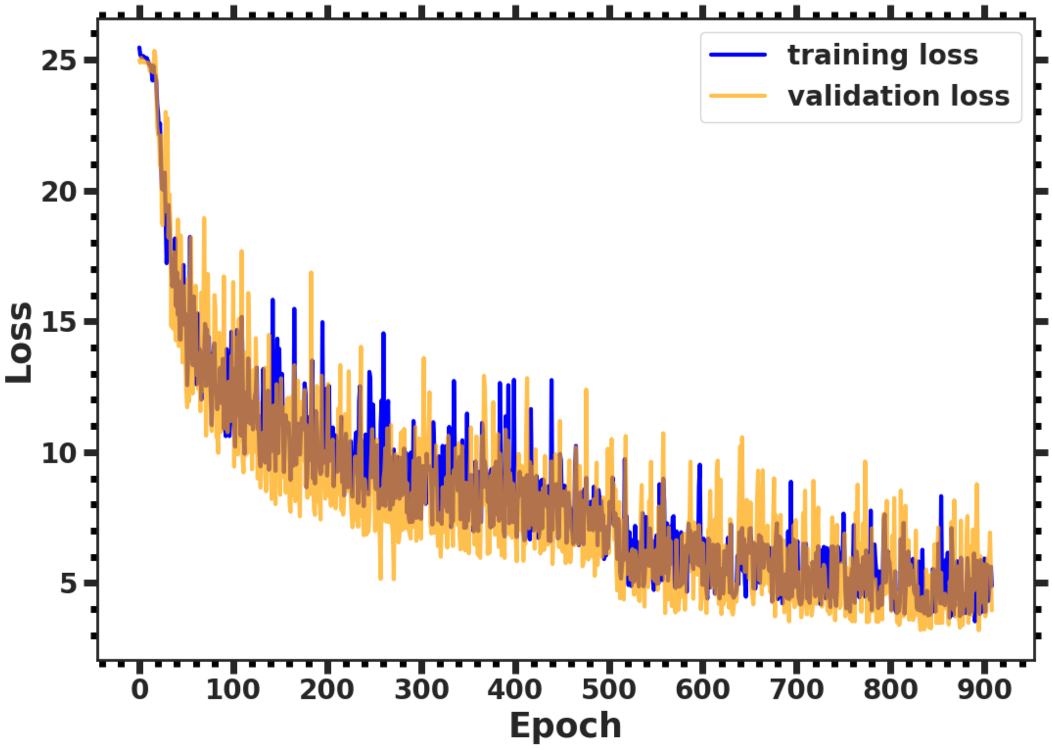

Figure 2.9: *The image shows the training loss and validation loss for n = 9500 samples. The loss function used is* `MAE` *without scaling the target variables. The training loss reaches minimum value* 3.5337 *at the epoch* 890 *whereas the validation loss reaches minimum value* 3.1913 *at the epoch* 894

In our study, we utilized the mean absolute error (MAE) as the chosen loss function. Initially, both the training and validation loss started at values above 25. As the model began training, the loss function gradually decreased, following a smooth curve as illustrated in Fig. 2.9. Over the course of 909 epochs, the validation loss and training loss exhibited a steady decrease until the early stopping mechanism was triggered. The initial epoch was set to 1000, and the best validation loss of 3.2 was achieved at epoch 895. Importantly, the loss curves demonstrate a desirable pattern, showing that our model is not suffering from overfitting or underfitting.

The observed behavior of the loss curves provides evidence of the effectiveness of our training process. The gradual decrease in loss signifies that the model is learning and adapting to the data, while the absence of significant fluctuations suggests stability in the training procedure. The achieved validation loss of 3.2 demonstrates the model's ability to generalize well to the unseen data. These results highlight the successful training of our model, indicating its potential for accurate predictions and reliable performance on the unseen data.

2.5.2 Model Evaluation

Estimation on synthetic data

In the analysis, the performance of the trained model is evaluated by estimating the input $f_{\rm NL}$ values on both the training dataset and the unseen test dataset. The model checkpoint saves the weights of the epoch with the least validation loss for further analysis.

To assess the model's performance, we plot the output $f_{\rm NL}^{\rm local}$ values against the input values in Fig. 2.10. The blue dots represent the estimation of $f_{\rm NL}$ on the training data consisting of $8,550$ $f_{\rm NL}^{\rm local}$ values, while the yellow dots represent the estimation on the validation data containing 950 $f_{\rm NL}^{\rm local}$ split during the training. The y-axis corresponds to the output $f_{\rm NL}$ values predicted by the model, while the red line represents the ground truth where $y = x$.

In Fig. 2.10, we can see that the $f_{\rm NL}^{\rm local}$ outputs against inputs graph exhibits a strong correlation, demonstrating a reasonable level of accuracy.

Tests of unbiasedness

We further test the unbiasedness of the test data for the $n = 500$ outputed $f_{\rm NL}^{\rm local}$ against input values. We utilise the minimal χ^2-fitting method, which is a straightforward way to find the best-fitting line through a series of scattering points. Considering a straightline with slope a and interception b, we model the linear regression as $y = ax + b$, then we formulate the χ^2 as follows

$$\chi^2 = \sum_{i=1}^{n=500} \left(f_{\rm NL}^{\rm out} - y(f_{\rm NL}^{\rm in}) \right)^2 . \tag{2.5.1}$$

Then we run it on a regular 2-D grid of parameter space (a, b), with flat prior ranges as $a = (-10, 10)$, $b = (-10, 10)$ and each side of sampling 10000. We show the

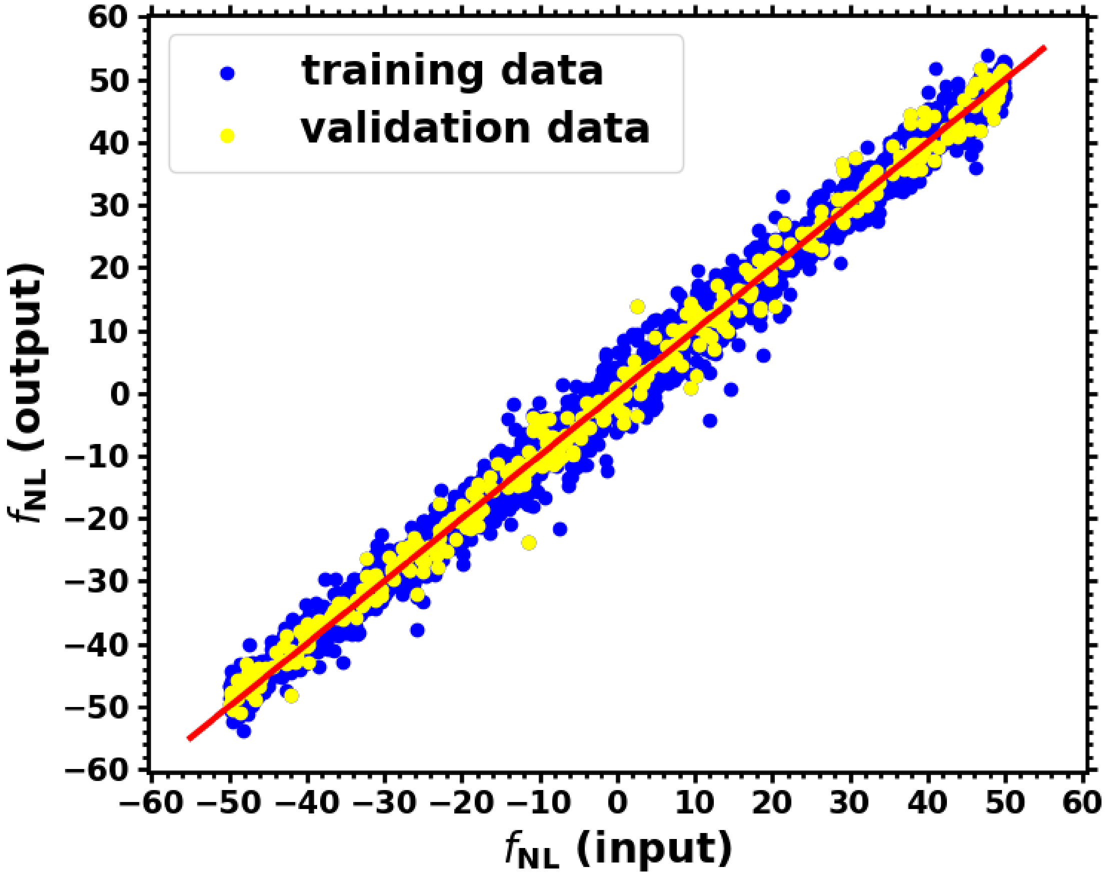

Figure 2.10: *Scatter plot showing the comparison of estimated* f_{NL} *values by a CNN model on training and test dataset, with the solid line indicating the ideal case where estimated values would match the true values. The blue dots are training data* $(n = 8550)$ *and yellow dots* $(n = 950)$ *are validation data, the deviation of from the ideal line in red indicates the level of error in the model's predictions.*

results of the constraint in Fig. 2.11. One can see that the best-fitting slope parameter a and interception b are

$$a = 0.980^{+0.098}_{-0.102}, \quad b = 0.277^{+0.098}_{-0.101}, \tag{2.5.2}$$

which recovers the complete unbiasedness $(a, b) = (1, 0)$ in 2σ C.L. One can also see the 2-D joint constraint in the corner plot in Fig. 2.12, which shows that the unbiased linear regression $(a, b) = (1, 0)$ locates within the 95% C.L. contour. The R^2 score for the test data is 0.96, indicating a high level of accuracy in the estimation provided by the CNN model.

2.6 Conclusion

In this work, we developed a deep learning model that could accurately identify the non-Gaussianity parameter ($f_{\rm NL}$), directly from cosmic microwave background (CMB) maps. To achieve this, we trained a convolutional neural network (CNN) on a synthetic CMB map dataset that contained maps with different values of $f_{\rm NL}$. The CNN model consisted of several layers, including convolutional, pooling, and fully connected layers, which were optimized using the Adam optimizer and the binary MAE (mean absolute error) loss function.

During training, we used a batch size of 256 and a learning rate of 0.001, which were selected based on their impact on the model's performance. We also employed various early stopping, to prevent overfitting to the training data.

The plot of loss function versus epoch for both training data and validation data is a smooth curve which tells us the model is converging and not overfitting to the training data. In our case, the increase in validation loss is minimal and does not affect the overall performance of the model. This suggests that the model has generalized well to new data and is not overfitting to the training set. We can see that the loss for both the training and validation data decreases gradually during the initial epochs, indicating that the model is learning quickly and efficiently. The fact that the loss for both datasets converges to around a similar value also suggests that the model has learned to distinguish between maps with different values of $f_{\rm NL}$ on both the training and validation sets. The

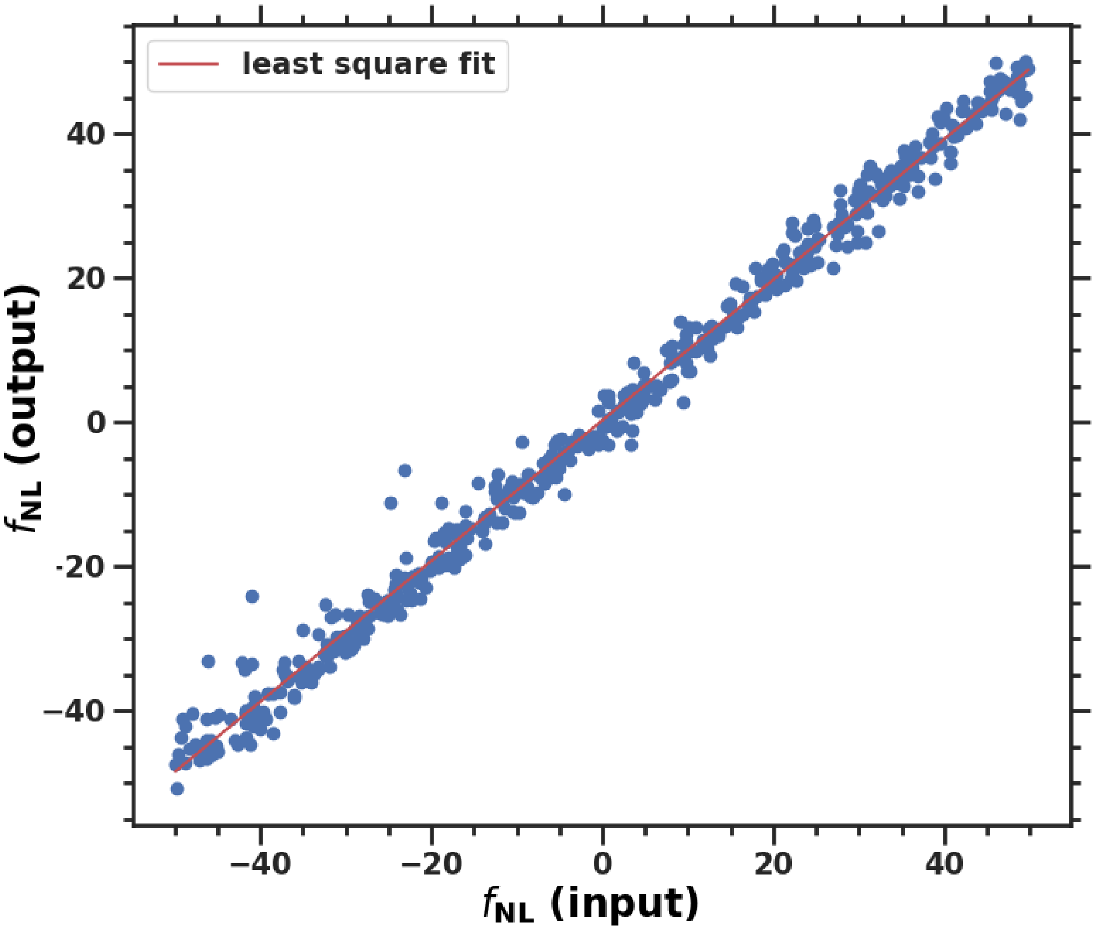

Figure 2.11: *This graph shows the relationship between the input* f_{NL} *and the predicted* f_{NL} *values from a machine learning model. The blue dots* ($n = 500$) *represent the input* f_{NL} *values, while the red line represents the least square fit of the input* f_{NL} *and predicted* f_{NL}*. The intercept of the line is* 0.274*, and the slope is* 0.975*, indicating a strong positive correlation between the two variables.*

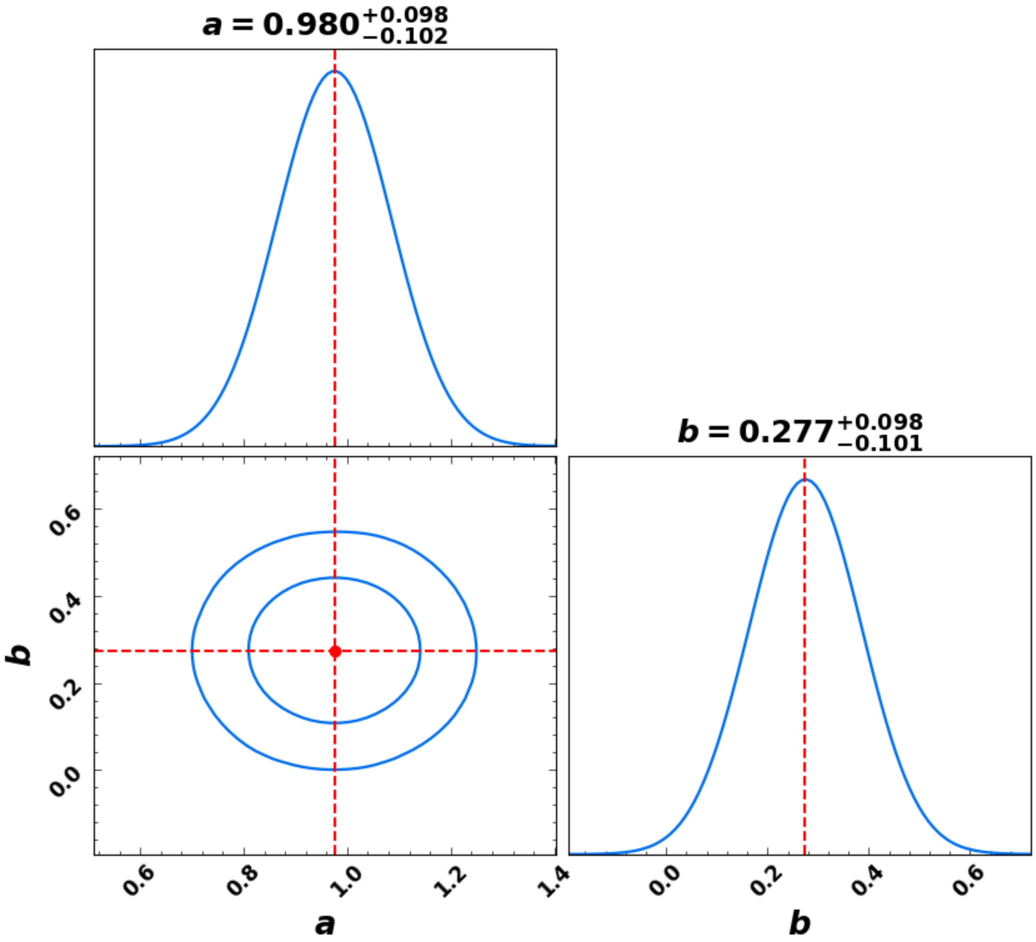

Figure 2.12: *Constraints on slope parameter a and interception b by using the test data. The two 2-D contours are the* 68% *and* 95% *confidence level of the parameters* (a, b)*, and the diagonal plots are the marginalised distribution of the parameters.*

smooth curve of the loss function versus epoch provides further evidence that our CNN model is well-designed and trained,

To evaluate the performance of our CNN model, we used two metrics: prediction accuracy on both the input map used during training and on a separate test map that was not seen during training. We found that the model can achieve high accuracy on both the training and test datasets, indicating that it has learned to distinguish between maps with different values of f_{NL}.

Overall, our results suggest that a well-designed and trained CNN model can be effective in identifying f_{NL} from synthetic CMB map images directly. However, we acknowledge that further testing with more diverse and realistic data with varying cosmological parameters are necessary to fully assess the model's performance and generalizability. Nonetheless, our results are promising and demonstrate the potential of deep learning techniques for cosmological parameter estimation from the CMB map directly.

3. Cosmological Parameter Estimation using CNNs

In extracting information from both the early and present universe, the temperature and polarization anisotropies within the cosmic microwave background (CMB) radiation serve as valuable sources. Originating from the last scattering surface (LSS) at approximately $2970K$, this polarized CMB radiation is observed in the current universe as a blackbody radiation with a temperature range of $2.725 \pm 0.002K$ (Fixsen et al. (1996); Mather et al. (1999)). Initially detected by Penzias and Wilson in 1965 Wilson & Penzias (1965), the CMB was subsequently measured for its tiny temperature anisotropy by the COBE satellite (Smoot et al. (1992)). Following this, the WMAP satellite (Hinshaw et al. (2009); Bennett et al. (2013)) improved measurements by estimating the CMB angular power spectra for temperature and polarization anisotropies. Notably, recent advancements in satellite-based experiments, such as Planck (Planck Collaboration et al. (2020a,c)), further refined the accuracy of temperature and polarization anisotropy measurements within the CMB.

The latest findings from ground-based experiments such as ACT (Sievers et al. (2013); Choi et al. (2020); Aiola et al. (2020) and SPT (Hou et al. (2014); Benson et al. (2014); Dutcher et al. (2021)) have unveiled their refined analyses of the CMB observations. In the contemporary era, a primary objective within the CMB community revolves around achieving higher precision in measuring the CMB polarization anisotropies. This focused effort aims to furnish the essential information necessary for a more accurate comprehension of the early universe.

The traditional methods to estimate or constrain the parameters are using two-point statistics, likelihood inference method, MCMC method which are briefly explained in 2, but such methods demand very high computational requirements for analyzing large-scale cosmological datasets and can be extremely demanding, requiring significant computing power and resources. This can limit the ability of researchers to perform detailed analyses of large datasets, particularly for smaller institutions or organizations with limited computational resources.

The need for constraining cosmological parameters arises from the fact that the universe is vast and complex, and our understanding of its fundamental properties is still incomplete. By measuring and quantifying the values of these parameters, we can gain insights into the nature of the universe and test our current theories of cosmology.

Some of the key reasons for constraining cosmological parameters include:

1. By measuring the values of cosmological parameters will help us in predictions of different cosmological models and determine which ones are most consistent with the observed data. This can help us to refine our understanding of the universe and identify areas where further research is needed.
2. Understanding dark matter and dark energy which are the most mysterious components of the universe, it is estimated that they make up a significant portion of its total mass-energy content. By constraining the parameters that describe their properties, we can gain insights into their nature and behavior, which could have important implications for our understanding of the universe as a whole.
3. Predicting future cosmic evolution by understanding how cosmological parameters have evolved over time, we can make predictions about how they will continue to evolve in the future.
4. The development of new technologies and observational techniques is essential for advancing our understanding of cosmology. By constraining cosmological parameters, we can identify areas where new technologies are needed to improve our measurements and observations, which can drive innovation and technological development in related fields.

Constraining cosmological parameters is a critical step in advancing our understanding of the universe and testing our current theories of cosmology. It requires a multidisciplinary approach that combines observations from a variety of sources, as well as sophisticated statistical techniques and computational resources. The computational requirements for analyzing large-scale cosmological datasets can be extremely demanding, requiring significant computing power and resources. This can limit the ability of researchers to perform detailed analyses of large datasets, particularly for smaller institutions or organizations with limited computational resources. Hence, we adopted neural networks, particularly Convolutional Neural Networks (CNN) to carry out the task of estimating parameters from CMB maps.

CNNs are a powerful and robust machine learning technique for image and signal processing tasks, it is an upcoming practical solution for overcoming computational challenges in cosmological parameter estimation. The process often involve developing customized neural network architectures that are specifically designed to handle the unique challenges of cosmological parameter estimation, rather than relying on off-the-shelf CNNs.

In our work, we develop a custom CNN model specifically for estimating cosmological parameters from CMB maps. Our model demonstrates sensitivity to the variations in underlying cosmological parameters in the CMB maps, allowing for more accurate and efficient parameter estimation.

3.1 Methodology

3.1.1 Simulated CMB Maps generation

Utilizing CAMB [1], we compute the temperature angular power spectrum of the Cosmic Microwave Background (CMB) based on the planck-2018 but as we are estimating each parameter independently. We generate six different dataset varying each parameter independently within the range as prescribed below in the Table (3.1).

These are the cosmological parameters we focus on estimating using CNN.

Beam function

We perform beam convolution on the CMB temperature maps before feeding them into the convolutional neural networks (CNNs) for parameter estimation. This step is necessary

---

[1] CAMB Lewis et al. (2000) (Code for Anisotropies in the Microwave Background), a cosmology code for calculating CMB, lensing, galaxy count, dark-age 21cm power spectra, matter power spectra and transfer functions.

parameter	range
H	$(64.5, 74.5)$
$\Omega_b h^2$	$(0.015, 0.030)$
$\Omega_c h^2$	$(0.1, 0.2)$
$10^{-9} A_s$	$(1.5, 3.5)$
n_s	$(0.85, 1.265)$
τ	$(0.01, 0.11)$

Table 3.1: *Range of the parameters : H is the Hubble parameter, $\Omega_b h^2$ is the baryon matter density, $\Omega_c h^2$ is the cold dark matter (CDM) density, n_s is the spectral index of the primordial curvature perturbations, A_s is the scalar amplitude of the primordial curvature perturbations, τ is the optical depth.*

because the CMB signals are not directly observed but are measured after being convolved with the instrument's beam pattern. The beam convolution introduces additional noise and distortion to the CMB signal, which can affect the accuracy of parameter estimation. By applying beam convolution during the training process, CNNs can learn to account for this additional noise and distortion, leading to more accurate parameter estimates.

For beam convolution, we just simply multiply Eq. (2.3.1) with a Gaussian kernel,

$$a_{\ell m} \to a_{\ell m} e^{-\ell^2 \sigma_{\mathrm{b}}^2/2}, \tag{3.1.1}$$

where $\sigma_{\mathrm{b}} = \theta_{\mathrm{FWHM}}/\sqrt{8 \ln 2}$, and $\theta_{\mathrm{FWHM}} = 5\,\mathrm{arcmin}$ is the *Planck* Full-Width-Half-Maximum beam size.

Noise

The noise properties of the Planck satellite mission data are estimated using a method called the Half-Ring Half-Difference(HRHD). This involves splitting the data from each of the Planck detectors into two halves, corresponding to two different scans of the sky and taking the difference between the two halves, which is independent of any astrophysical signal. The HRHD method is used here to estimate the noise properties of the Planck data to remove the instrumental noise using the following equation,

$$HD = \frac{\delta_1 - \delta_2}{2}. \tag{3.1.2}$$

where δ_1 and δ_2 are the half-ring maps released by *Planck* mission.

The SMICA Planck maps are obtained to prepare our training data. The HRHD method mentioned above is used to estimate the noise properties and to get the resulting noise map. Once the CMB maps are obtained, the 'C_l' values are computed using HEALPY [2]. First, 'a_{lm}' values are obtained using HEALPY'S subroutine `healpy.map2alm()`, further 'C_ls' are obtained from another subroutine `hp.alm2cl()`. The a_{lm}s are computed for $l \in \{2, 1535\}$. Once C_ls are obtained, multiple noise profile maps are generated with different realizations using subroutine `healpy.synfast(`C_l`)` for `nside = 64`.

3.1.2 Simulated CMB + *planck* noise Maps

The final piece of training data is to ensure that the training dataset closely resembles the reality of the Planck CMB data. Consequently, the generated noise maps are combined with the CMB maps using the equation (3.1.3) below,

$$C_l^{tot} = B_l^2 C_l^{CMB} + N_l. \tag{3.1.3}$$

Where C_l values are obtained from simulated CMB map, N_l is the Noise from the Planck map.

By simulating and incorporating various noise maps with multiple realizations by using different random seeds, the training dataset encompasses a range of noise realizations, the training dataset captures the combined effect of the CMB signal and the instrumental noise. This approach allows for a more accurate representation of the true

---

[2] HEALPY is a Python package to handle pixelated data on the sphere. It is based on the Hierarchical Equal Area isoLatitude Pixelization (HEALPix) scheme and bundles the HEALPIX, C++ library Zonca et al. (2019); Górski et al. (2005).

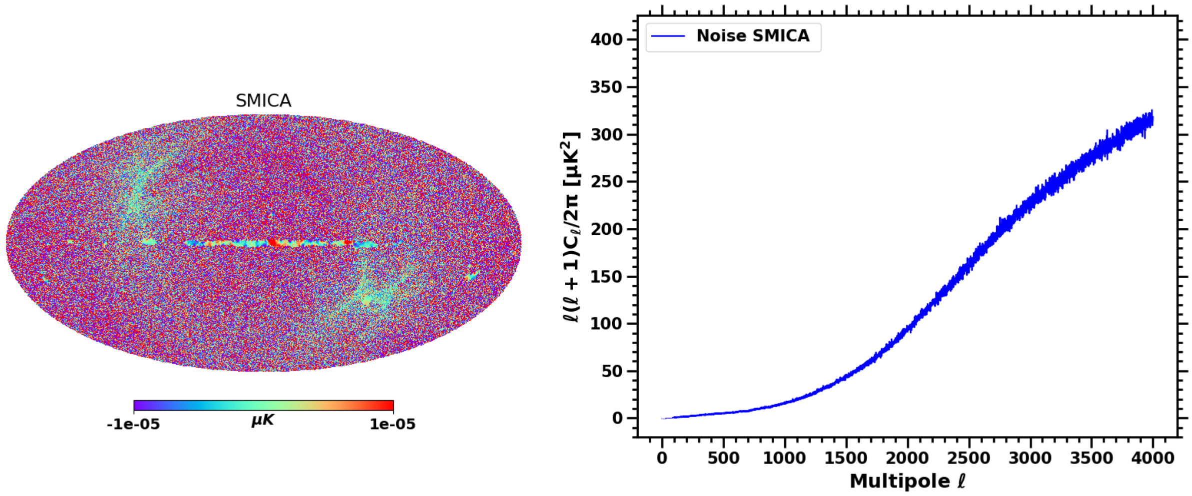

Figure 3.1: *The provided image left depicts a noise map of the Cosmic Microwave Background (CMB), which has been obtained by taking half the difference between the foreground removal technique SMICA. The left plot compares the noise power spectra calculated using two different methods SMICA - for analyzing the Cosmic Microwave Background radiation data from the Planck survey, data release 2018.*

observations made by the Planck satellite. We calculate the noise power spectrum of SMICA from the maps which is consistent with the expectations.

3.1.3 Convolutional Neural Network

Single Neuron

In the section (2), we explain in detail how a single neuron functions and its pictorial representation in (2.5)

Recall the mathematical representation of a single neuron in a neural network 2.4.1,

$$p_i = \sum_{i=1}^{n} w_i x_i + b. \tag{3.1.4}$$

The neural network architecture we employ, known as Convolutional Neural Networks (CNNs), is illustrated in Fig. 3.2 is made up of numerous neurons expressed in the Eq. 3.1.4. This type of neural network is particularly well-suited for processing visual data, such as images, due to its ability to extract features from the input through a series of convolutional and pooling layers which are hidden layers in Fig. 3.2. The network consists of multiple layers, including convolutional layers that apply filters to the input image to extract features, pooling layers that downsample the feature maps to reduce computational complexity, and fully connected layers that combine the extracted features to make predictions. The CNN structure we use has been shown to be highly effective in processing CMB maps.

3.1.4 Customized Convolutional Neural Network (CNN) Model

- Input Layer: The model expects input images with a shape of (256, 192, 3) represented in blue dots in Fig 3.2. This means the input array must have a height of 256, width of 192, and three channels.
- Convolutional layers:
  - Four Conv2D layers with increasing number of filters $(32, 64, 128, 256)$ and ReLU activation functions. Each layer is followed by a MaxPooling2D layer to downsample the spatial dimensions.
- Flatten layers:
  - The Flatten layer transforms the output from the convolutional part of the model into a one-dimensional array, preparing it for input into the dense layers.
- Dense layers:
  - Two Dense layers with 256 and 128 neurons, respectively, and ReLU activation functions.
  - The output layer consists of a single neuron with *linear activation function* for our regression task.
- Regularization and Dropout:
  - ‘L1’ and ‘L2’ regularization (with a coefficient of 0.001) have been applied to the kernel weights of each convolutional and dense layer.

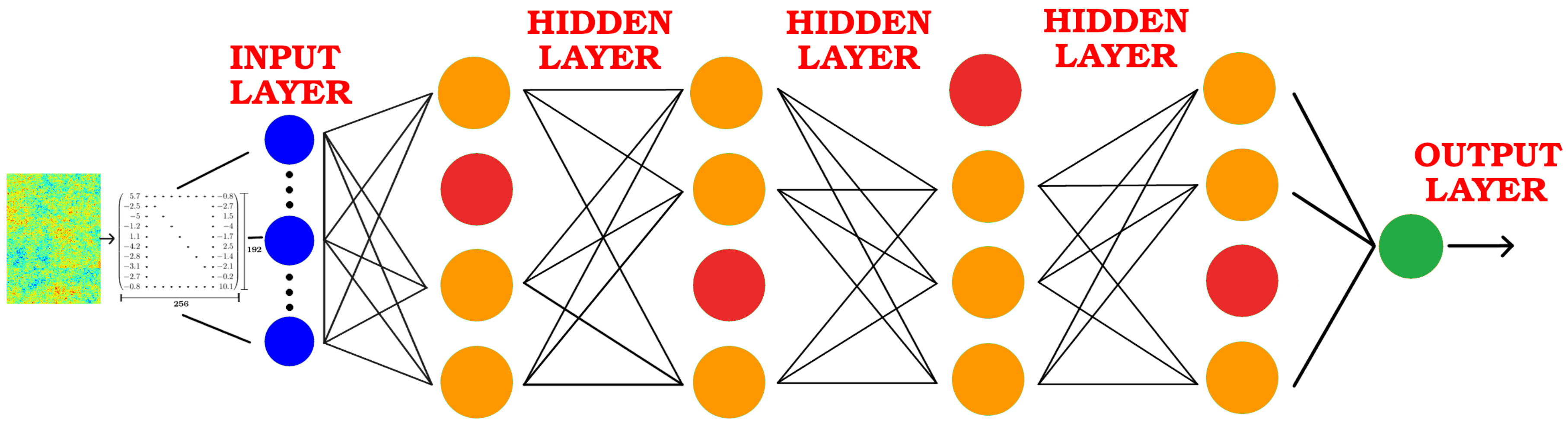

Figure 3.2: *The image depicts a neural network with an input layer, multiple hidden layers, and an output layer. Each circle represents a neuron, and the lines connecting them represent the connections between neurons. The red colored neurons are dropped out neuron to counter over-fitting. The input layer takes in the input data, which is then processed through the hidden layers using various activation functions. Finally, the output layer produces the final result, which could be a classification or prediction based on the input data. The image demonstrates the complexity of neural networks and how they can be used for tasks such as image recognition, natural language processing, and more.*

  - The Dropout rates of 0.25 as needed for each Dropout layer to control the fraction of units to drop. The red dots in Fig 3.2 respresents the dropped out neuron which are not connected to any other neuron in the network. Experimentation with different Dropout rates can help in achieving better performance and avoiding over-fitting in your model.

- Compilation:
  - The model is compiled using the 'adam' optimizer and 'mean absolute error' loss function, suitable for regression tasks.

  Output Layer:

  - The output layer (a green dot in Fig 3.2) is typically a single fully connected layer that takes the output from the final convolutional layer as input. The output layer has a single neuron that predicts a continuous numerical value. The activation function used in the output layer is linear for our task.

Loss function

In our training, we use MEAN ABSOLUTE ERROR as a loss function. It measures the average magnitude of errors between predicted values and true values.

Mean Absolute Error (MAE) is calculated as the average of the absolute differences between predicted and actual values across all data points in the dataset. Mathematically, for a dataset with 'n' samples:

$$loss_{mae} = \frac{1}{n}\sum_{i=1}^{n}|y_{\{true,i\}} - y_{\{pred,i\}}| \tag{3.1.5}$$

where $y_{\{true,i\}}$ is the true target value of the i_{th} sample, $y_{\{pred,i\}}$ is the predicted value of the i_{th} sample, '$|\cdot|$' denotes the absolute values.

MAE is less sensitive to outliers compared to other loss functions like Mean Squared Error (MSE) because it treats all errors linearly without squaring them making it ideal for the data with outliers but the larger errors have a proportional impact on the total loss, as MAE considers the absolute magnitude of errors.

L1, L2 and Dropout

`L1 Regularization` and `L2 Regularization` also known as *Lasso regression* and *Ridge regression* respectively, along with `Dropout`, are regularization techniques used in machine learning to prevent over-fitting and enhance the generalization capability of models.

`L1 regularization` adds a penalty term to the loss function proportional to the absolute value of the weights' coefficients, mathematically they are given by

$$L_1 = l + \lambda \sum_{i=1}^{n} |w_i| \tag{3.1.6}$$

`L2 regularization` adds a penalty to the loss function proportional to the squared magnitued of the weights' coefficients. The regularization is mathematically represented as,

$$L_2 = l + \lambda \sum_{i=1}^{n} |w_i^2| \tag{3.1.7}$$

Where 'l' is the loss function adopted during the training, 'λ' is the parameter strength that determines the impact of the regularization term on the overall loss, 'w_i' represents individual weights of the model, where 'i' ranges from '1' to 'n', the total number of weights in the model.

Dropout is a regularization technique used during training where a random fraction of neurons (units) are dropped out or ignored during each iteration. Dropout is a regularization technique used during training where a random fraction of neurons (units) are dropped out or ignored during each iteration. At each training iteration, a neuron has a probability 'p' of being retained (not dropped out) where 'p' is a hyperparameter typically set between 0.2 to 0.5. Dropout is used to prevent overfitting, which occurs when a model lears to fit the training data too well, capturing the unwanted noise and patterns that are specific to the training data but are not generalized to new unseen data.

3.1.5 Data Preparation

Z-scaling

Following the derivation of the final 'a_{lm}' maps using the beam function and the introduction of noise into the simulated CMB maps, we transform them into a map representation using the HEALPY subroutine *hp.alm2map*().

In our scenario, the `nside` value is 64, generating a float array of size 49152. The histogram showcasing the array values displays a notably Gaussian distribution. However, these values are exceedingly minute, approximately on the scale of 10^{-5}. Given the subtle disparities among the array values, the CNN may potentially lack sensitivity to such minute fluctuations.

To mitigate this concern, we implement z-scaling, a technique that retains the Gaussian nature of the array while magnifying the magnitude of each data point. This approach ensures that the CNN can adeptly discern and learn from the variations in the values, despite their initial minuteness. Consequently, the application of the z-scaling technique augments the sensitivity and interpretability of the data by the CNN.

Z-scaling is applied as follows:

$$z_i = \frac{x_i - \mu}{\sigma}. \tag{3.1.8}$$

where:

- z_i is the standardized value (z-score) of the ith data point.
- x_i is the original value of the data point
- μ is the mean of the data distribution in the given array which is z-scaled.
- σ is the standard deviation of the data distribution in the given array.

The z-scaling is independently applied to each input map in our dataset. This procedure serves to enhance the discernibility of values for the CNN while preserving the Gaussian distribution pattern of the each input map.

By implementing scaling separately for each map, we prevent any potential data crossover between different maps, safeguarding the model's performance on unseen data. Treating each map individually during the scaling process upholds the integrity of the training dataset, allowing the CNN to grasp distinct patterns and variations specific to each map. This strategy facilitates better generalization of the model to novel datasets, as it remains uninfluenced by information from other maps.

The histogram in the fig (2.8) demonstrates the distribution of array values, showcasing the scaled values that maintain the array's original shape.

2D Conversion.

Upon the application of z-scaling, the resultant map retains its one-dimensional array format of size 49152, aligning with the chosen `nside` parameter set at '64'. However, the strength of a CNN lies in its capability to apprehend shapes, complex patterns, intricate details, and depth within images. To harness this potential, we make use of the healpix module, facilitating the transformation of the one-dimensional array into a two-dimensional structure.

Leveraging a specific HEALPYfunction, we derive the spatial coordinates of each pixel within the image for the given array of the map. This transformation effectively reshapes the map into a two-dimensional array of dimensions (256, 192).

By directly converting the map into a two-dimensional rectangular array while safeguarding the z-scaled values, further '*padding*' is not necessary as the input shape is rectangular array. This process furnishes us with an each map of dimensions (256, 192) which is ideal for the input layer. For each parameter to estimate 10^4 maps are generated with numerous realization of CMB map as well as noise maps.

Array stacking

The transformation of the map into a (256, 192) array enhances the CNN's ability to detect shapes and features; however, it lacks depth information due to its one-channel structure. This absence of depth information becomes critical as each map $(m_1, m_2, ..., m_n)$ can yield diverse array values for identical underlying parameters. Consequently, the values in the CMB map array can be influenced not only by the underlying parameters but also by different realizations, leading to increased complexity. This complexity arises from the existence of multiple array values for each the map corresponding to the same underlying parameter values. To address this challenge, a 3-channel feature of CNN is employed. This is achieved by replicating the initial array of (256, 192) three times, resulting in an array of shape (256, 192, 3) for each map. This transformation enables the incorporation of depth-related information, allowing the model to capture the nuances across multiple realizations for a more comprehensive analysis.

Crucially, this process retains the integrity of z-scaled values without any clipping or data point truncation. This methodology ensures a preservation of information that might otherwise be compromised in conventional approaches, such as converting the map into a mollweide projection a white background or a PNG image which employs data truncation. These conventional methods have the potential to introduce systematic errors and distortions to the dataset, impacting its accuracy.

3.1.6 Target scaling

Target or output scaling is a process which refers to the process of scaling or normalizing the target variables or output values in the training dataset. Unlike feature scaling, which normalizes input features to a certain range, target scaling involves adjusting the target values themselves. The primary goal of target scaling is to preprocess and transform the target values to a specific range or format that can benefit the training process of a neural network, it ensures consistency and facilitate convergence during training. Improving the stability and speed of learning by ensuring that the model's learning rate remains consistent across different target value magnitudes. Normalizing the targets can prevent the model from prioritizing certain features based solely on their scale.

Methods for target scaling can vary based on the nature of the problem and the desired outcome. The usual scaling employed are "`Min-Max scaling, Standardization, Log Transformation, custom scaling`".

Each parameter is custom *Min-Max* scaled from either 1 to 100 or 1 to 1000 depending on the parameter estimated. The six parameters are estimated and for each parameter dataset is generated separately as well as the training is carried out separately.

The formula for scaling the target values to a specific range is done using

$$X_{scaled} = a + \left(\frac{(X - X_{min}) \times (b - a)}{X_{max} - X_{min}} \right) \tag{3.1.9}$$

Where,

1. X is the original value of the target variable.

2. X_{min} is the minimum value of the target variable.
3. X_{max} is the maximum value of the target variable.
4. a is the lower bound of the desired range (1 in this case).
5. b is the upper bound of the desired range (100 or 1000).

This formula linearly scales the target values from the original range to the specified range of 1 to 1000, maintaining the relative relationships between the values while fitting them within the desired bounds. The differences between the scaled values will also be more pronounced in the {1 to 1000} range compared to the {0 to 1} or {1,100}.

3.1.7 Training

The custom CNN model used is discussed in the section (3.1.4), the same model is used to estimate all six parameters but independtly of each other. This means for each parameter the dataset is generated independtly and fed to the CNN independently for estimation. Hence, there are six dataset and six trained model for each parameter.

Parameter $\mathbf{H_0}$

In the process of estimating the parameter 'H_0', the decision was made to not to scale the target values, as the model displayed improved performance without this adjustment. The training dataset comprises 10,000 maps generated with randomly varying 'H_0' values spanning the range of $\{64.5, 75.5\}$ in a uniform distribution.

Out of the total 10000 maps, 500 are reserved for testing the trained model. Within the remaining set of 9500 maps, a 10% subset is designated as the validation set during the training process.

The model utilizes a `batchsize` of 256, employing the `Mean Absolute Error` as its loss function. Additionally, `L1` and `L2` regularization are both configured at 0.01, incorporating a dropout rate of 0.2. The model is trained over a maximum of 1000 epochs, implementing an early stopping mechanism if no improvement in the model.

Parameter $\Omega_b h^2$

During the estimation of the parameter '$\Omega_b h^2$', a decision was made to scale the target values within the range of $\{1, 100\}$. This adjustment was implemented after initial modeling attempts without scaling produced poor estimations, and subsequent modeling with scaling showed improved performance. The training dataset consists of 10000 maps, each generated with randomly varying values of $\Omega_b h^2$' uniformly distributed within the range of $\{0.015, 0.030\}$.

From the total set of 10000 maps, 500 have been set aside for testing the trained model. Among the remaining 9500 maps, a subset comprising 10% has been allocated as the validation set for training purposes.

The model utilizes a `batchsize` of 256 and employs the `Mean Absolute Error` as its chosen loss function. Additionally, both `L1` and `L2` regularization are set at 0.01, while a dropout rate of 0.2 is incorporated. Training is conducted over a maximum of 1000 epochs, with an early stopping mechanism in place if no improvement in the model.

Parameter $\Omega_c h^2$

During the estimation of the parameter '$\Omega_c h^2$', the model initially attempted to scale the target values within the range of $\{0, 1\}$; however, this scaling resulted in diminished performance. Consequently, the decision was made to scale within the range of $\{1, 100\}$, which improved the performance significantly. By randomly varying '$\Omega_c h^2$' values uniformly distributed within the range of $\{0.10, 0.20\}$ the training dataset of 10,000 maps are generated among which 500 are reserved for testing the trained model. Within the remaining set of 9,500 maps, a 10% subset is designated as the validation set during the training process.

The model utilizes a `batchsize` of 256, employing the `Mean Absolute Error` as its loss function. Additionally, `L1` and `L2` regularization are both configured at 0.01, incorporating a dropout rate of 0.2. The model is trained over a maximum of 1000 epochs, implementing an early stopping mechanism if no improvement in the model.

Parameter $\mathbf{A_s}$

Estimating the parameter 'A_s' initially posed considerable challenges. Given its order of 10^{-9}, the minute differences between neighboring parameters presented a significant hurdle. Scaling attempts within various ranges, including $\{0, 1\}$, $\{1, 100\}$, and $\{1, 500\}$, all resulted in notably poor performance. After several trial and error iterations, scaling within the range of $\{1, 1000\}$ proved to be remarkably effective but it did introduce a little over-fitting which was tackled by increasing dropout to 0.25, additionally, `L1` regularization is set at 0.01 and `L2` regularization is set at 0.05.

The training data consists of 10000 maps featuring 'A_s' values randomly varied within the range of $\{1.5, 2.5\} \times 10^{-9}$, where 500 maps are set aside for testing the trained model, the 10% of remaining 9500 is reserved for validation set during the training process. The training is conducted over a maximum of 1000 epochs with an early stopping mechanism in place if there is no improvement in the model

Parameter $\mathbf{n_s}$

In the pursuit of estimating 'n_s', the scaling approach within $\{0, 1\}$ and not scaling the targets adversely affected the model's performance. Therefore, the model proceeded with scaling the target values within the range of $\{1, 100\}$ which showed siginificant improvement in the performance, using a training dataset consisting of 10000 maps featuring 'n_s' values randomly varied within the range of $\{0.85, 0.1265\}$. As usual, 500 maps have been set aside for testing the trained model, the subset comprising 10% of remaining 9500 maps are set for validation set during the training process

This model also utilizes a `batchsize` of 256 and employs the `Mean Absolute Error` as its chosen loss function and scaling the target values within the range of $\{1, 100\}$ had a siginificant improvement in performance. Training is conducted over a maximum of 1000 epochs, with an early stopping mechanism in place if no improvement in the model.

Parameter $\boldsymbol{\tau}$

The estimation of 'τ' involved an attempt to scale the target values within the range of $\{1, 1000\}$, which initially improved the model's performance. Subsequently, however, it

parameter	range	scaling	L_1	L_2	Dropout
H	$(64.5, 74.5)$	none	0.01	0.01	0.2
$\Omega_b h^2$	$(0.015, 0.030)$	{1,100}	0.01	0.01	0.2
$\Omega_c h^2$	$(0.1, 0.2)$	{1,1000}	0.01	0.01	0.2
$10^{-9} A_s$	$(1.5, 3.5)$	{1,1000}	0.01	0.05	0.25
n_s	$(0.85, 1.265)$	{1,100}	0.01	0.01	0.2
τ	$(0.01, 0.11)$	{1,1000}	0.01	0.01	0.2

Table 3.2: *Range of the hyperparameters used for each parameter during training*

led to over-fitting issues. Thus, the decision was made to scale down within the range $\{1, 100\}$, resulting in a very good performance with dataset comprising 10000 maps with 'τ' values uniformly distributed within the range of $\{0.01, 0.11\}$, out of which 500 maps are set aside for testing the model after the training as usual, the 10% of remaining 9500 is reserved for validation set during the training process. With the `batchsize` of 256 and the `Mean Absolute Error` as loss function the training is conducted over a maximum of 1000 epochs with an early stopping mechanism in place if there is no improvement in the model.

The table (4.1) summarizes the range of each parameter, scaling and regularization strength employed during the training.

3.2 Results

In this section, we discuss the outcomes derived from training and estimating the parameters using customized CNN model described in the (3.1.4) using the loss curves, estimation on the simulated data and confidense ellipse. Assessing the performance of a regression model is not simple a task. A model may exhibit over-fitting or under-fitting characteristics, even when its predictions are highly accurate. Thus examining training and validation losses gives us an insight into the performance of the model. In this section we discuss the performance of the model using loss curves, estimation on the training/validation data, test of unbiasedness on the unseen data and the confidence ellipse for each parameter.

The loss plots discussed below for each parameter shows a gradual decreasing trend in both the losses indicating that the model is learning and adjusting its weights accordingly to fit the training data better. The training loss stabilizes to a low value suggesting that the model has effectively learned the patterns in the training data and might perform well on new, unseen data. Validation loss also converges around similar value as training loss indicating that the model is generalizing well. Any sign of over-fitting or under-fitting could be seen if there is a significant gap between the training loss and validation loss suggesting poor performance. Ideally, the validation loss should follow a similar decreasing trend to the training loss which is the case we see for all the six parameters. Comparing the loss magnitudes of training and validation loss show that they converge around a similar value and the difference between them is minimal.

The model's performance is evaluated by plotting the true values against the predicted values for each parameter. The estimation is visualized for both the training and validation sets. The plots align well with the expected outcomes, consistent with the observed loss curves for each parameter. The divergence from the red line, where '$x = y$', indicates a range from broad to narrow spread across each parameter. This pattern aligns with the volatility observed and the gaps between the loss curves for each parameter. These findings suggest that the model is showcasing its capacity to estimate each parameter with commendable accuracy.

We test the performance our model using the minimal χ^2-fitting method, it is a simple technique used to determine the best-fitting line through a set of scattering points. This method involves finding the values of the slope (a) and intercept (b) of the line that result in the smallest possible value of the chi-squared statistic. The chi-squared statistic is a measure of how well the model fits the data, and a smaller value indicates a better fit.

To apply this method, the input (x) and output (y) can be modeled as a straight line with a slope (a) and intercept (b). We then write this linear regression equation as $y = ax + b$. The χ^2 is formulated as,

$$\chi^2 = \sum_{i=1}^{n=500} \left(y^{\text{out}} - y^{\text{in}}\right)^2 . \tag{3.2.1}$$

From the above equation, we constrain the result using χ^2 statistic then run it on a 2-D grid of parameter space with flat prior ranges for a and b, each with a range of $(-10, 10)$. We sample 10,000 points on this grid. The results of the constraint are shown in least square fit plots for each parameter, where the best-fitting slope parameter a and interception b are found. This point recovers the complete unbiasness with $(a, b) = (1, 0)$ within $2 - \sigma$ C.L. We also present the 2-D joint constraint using a corner plots for each parameter, which demonstrates that the unbiased linear regression (a, b) = (1, 0) falls within the 95% C.L. contour.

Hubble constant $\mathbf{H_0}$

The figure (3.3) shows the `training vs validation loss` for Hubble constant. The training loss achieves its minimum value of 1.5338 by the 485^{th} epoch, while the validation loss reaches its minimum of 1.6624 at epoch 468^{th} where losses exhibit smooth curve with consistent gradual decrease. The absolute difference between these losses amounts to 0.1286, suggesting convergence around comparable values.

Further we evaluate the model's performance on the training data and the validation data. The simulated data consists of 10000 maps which are split into three parts. The test data consists of 500 maps which are unseen by the model during the training process. The remaining 9500 maps are used for learning with 10% reserved for validation $(n = 950)$ data and the rest $(n = 8550)$ for training the model.

The scatter plot in the Fig. 3.4 shows the comparison of prediction by the model on the training data represented in blue dots and validation data represented in yellow dots, with the solid line $x = y$ indicating the ideal case where estimated values would match the true values for the Hubble parameter H.

The Fig. 3.5 illustrates the correspondence between the input variable, denoted as 'H', and the predicted 'H' values derived from a machine learning model on the unseen test data for the Hubble parameter. The blue dots portray the actual input 'H' values, while the red line represents the least squares regression fit between the input 'H' and the

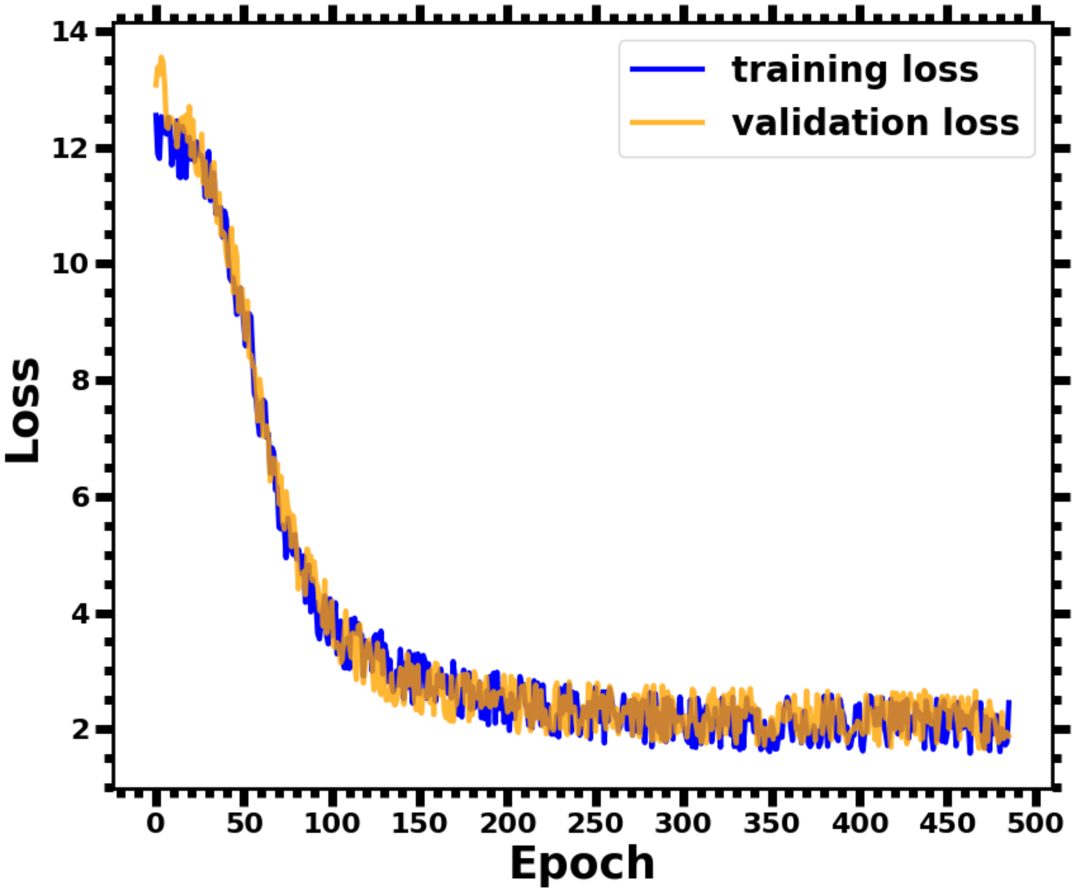

Figure 3.3: ***Parameter* $\mathbf{H_0}$**: *The loss curves exhibit smoothness and a consistent, gradual decrease. The training loss achieves its minimum value of* 1.5338 *by the* 485^{th} *epoch, while the validation loss reaches its minimum of* 1.6624 *at epoch* 468^{th}*. Notably, the absolute difference between these losses amounts to* 0.1286*, suggesting convergence around comparable values. The model's estimation of the parameter* H_0 *demonstrates exceptional performance. Note that the parameter is unscaled before the training.*

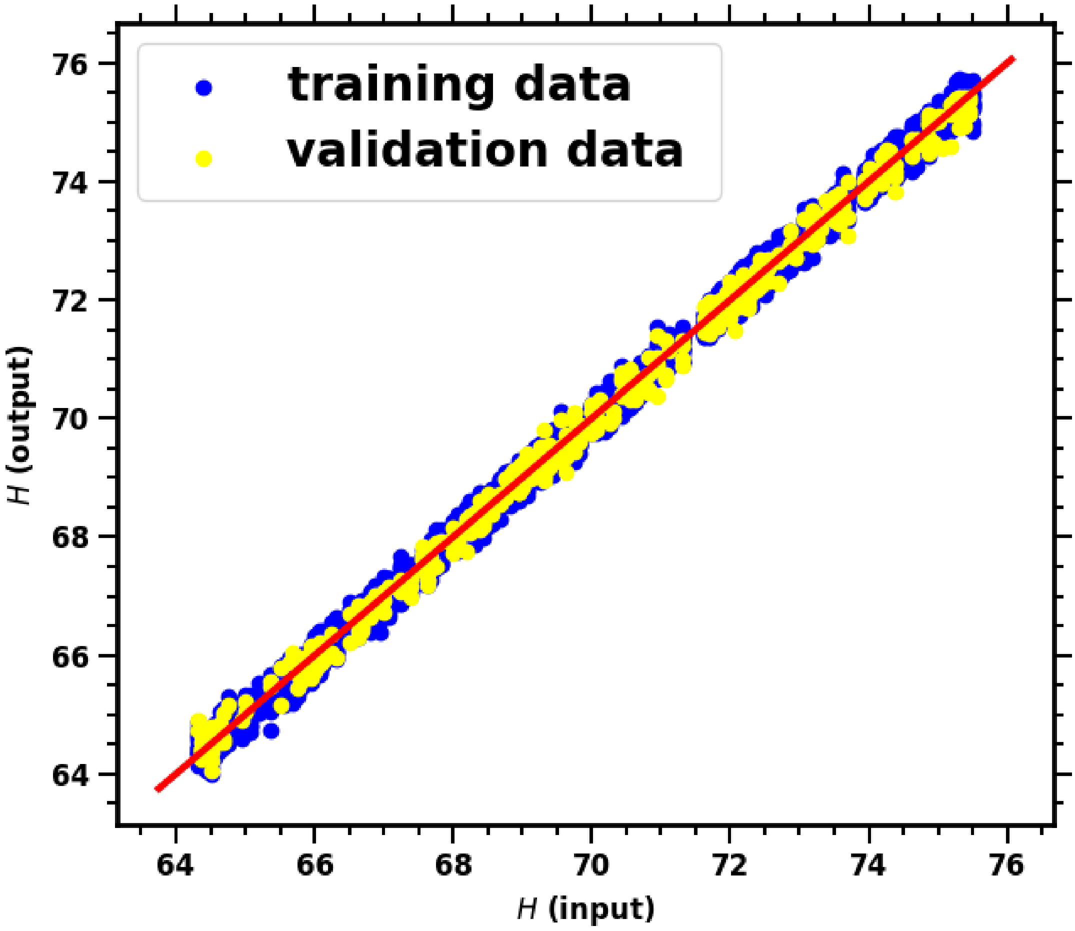

Figure 3.4: *Scatter plot showing the comparison of estimated 'H' values by a CNN model on training and test dataset. The training data ($n = 8550$) in blue dots and validation data ($n = 950$) in yellow dots, the ideal line is $x = y$ in red.*

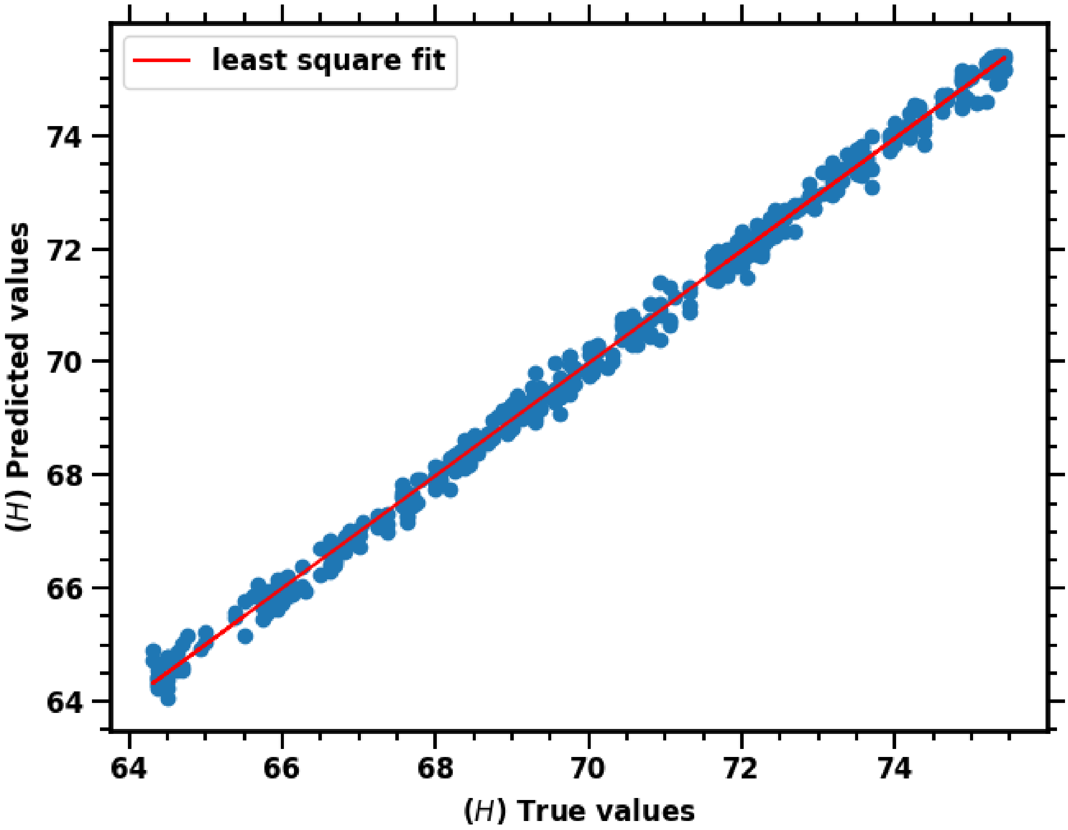

Figure 3.5: *Least square fit plot for Hubble parameter on the test dataset containing* $n = 500$ *maps, represented in bluedots. The least square fit line is represented in red with the intercept '*0.3844*', slope is '*0.9941*'. This strong positive correlation between the two variables is evident from these values, implying that as the input '*H*' increases, there is a corresponding rise in the predicted '*H*' values.*

predicted 'H' for the testdata set. The intercept of the line, calculated as '0.3844', signifies the value of 'H' when the predicted 'H' is zero, while the slope, computed as '0.9941' for the test dataset containing 500 maps, indicates the rate of change between the input 'H' and the predicted 'H'. These values indicate a robust positive correlation between the two variables, affirming that increases in the input 'H' correspond to corresponding rises in the predicted 'H' values.

The Fig. 3.6 shows the results of the constraint for Hubble parameter (H), he best-fitting slope parameter a and interception b are

$$a = 0.992^{+0.100}_{-0.098}, \quad b = 0.384^{+0.100}_{-0.099}, \tag{3.2.2}$$

Baryon matter density Parameter $\Omega_b h^2$

The loss curves for baryon density parameter shown in the Fig. 3.7 also exhibits a smooth and gradual decrease as epoch increases indicating convergence of both the losses around similar value. The minimum training loss achieved is 3.1073 at the epoch 734 whereas the minimum validation loss is 3.0643 at the epoch 736. The absolute difference between the training loss and validation loss is 0.0429, suggesting convergence around comparable values.

The model's evaluation is done on the training and validation data. The visualized scatter plot in the Fig. 3.8 depicts the prediction of the model on the training dataset of size $n = 8550$ in blue dots and the validation dataset of size $n = 950$ in yellow dots, red line representing the ground truth $x = y$.

The graph in Fig. 3.9 depicts the association between the input variable '$\Omega_b h^2$' and the corresponding predicted values from a machine learning model on an unseen test dataset, specifically focusing on the baryon matter density parameter. The blue dots symbolize the actual input values, while the red line represents the least squares regression line fitted to the input and predicted values for the test dataset. Notably, the line's intercept, calculated as 0.0014, represents the value of '$\Omega_b h^2$' when the predicted '$\Omega_b h^2$' is zero. Additionally, the slope of the line, measured as 0.9329, indicates the rate of change between the input '$\Omega_b h^2$' and the predicted values. These figures collectively suggest a robust positive correlation between the two variables, signifying that variations in the input '$\Omega_b h^2$' correspond to proportional changes in the predicted '$\Omega_b h^2$' values.

The Fig. 3.10 shows the results of the constraint for baryon matter density parameter ($\Omega_b h^2$), the best-fitting slope parameter a and interception b are,

$$a = 0.996^{+0.103}_{-0.097}, \quad b = 0.004^{+0.098}_{-0.102}, \tag{3.2.3}$$

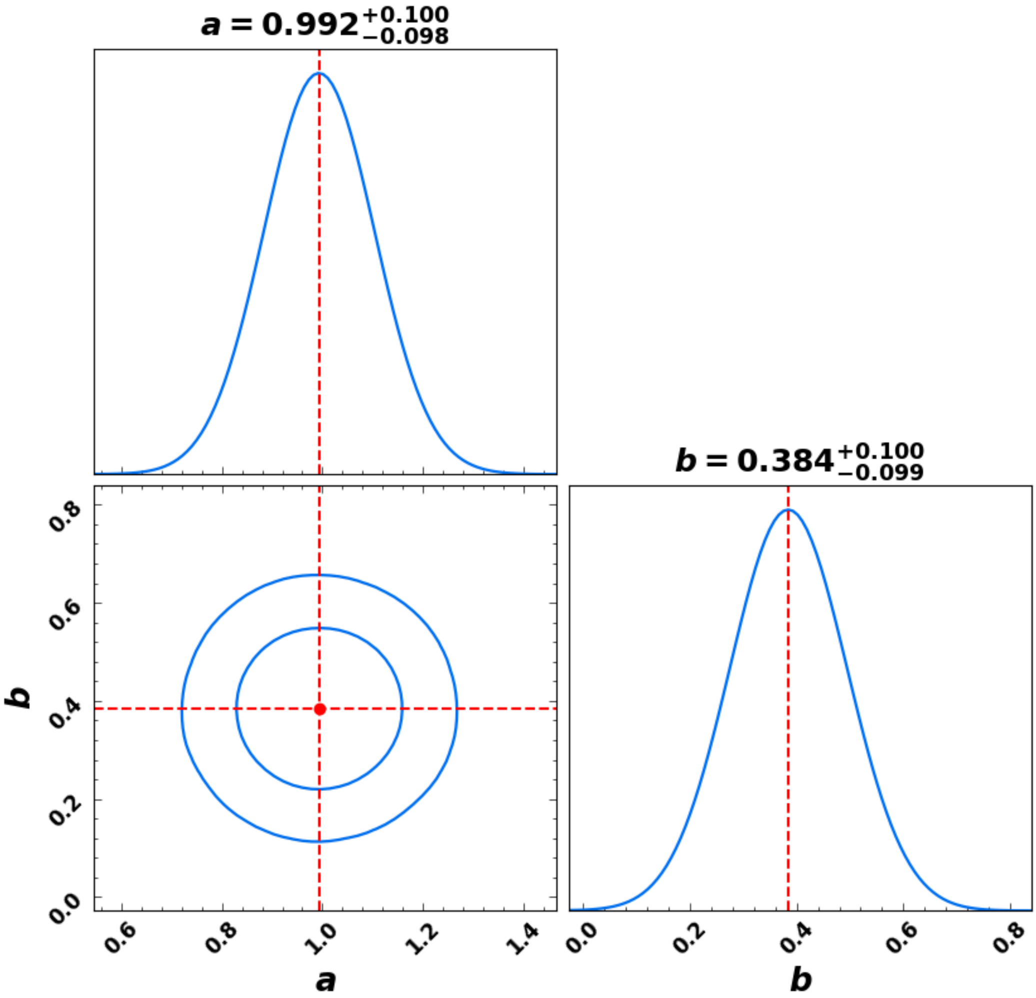

Figure 3.6: *Parameter H_0: Constraints on slope parameter a and interception b by using the test data. The two 2-D contours are the 68% and 95% confidence level of the parameters (a, b), and the diagonal plots are the marginalised distribution of the parameters.*

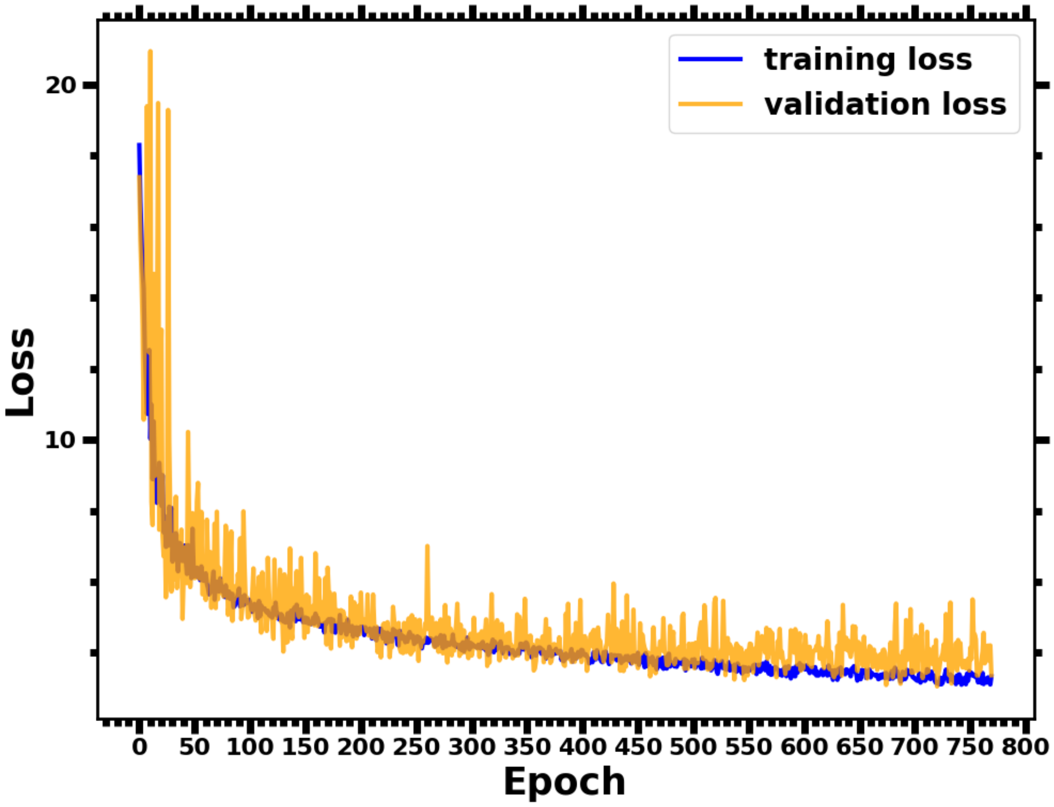

Figure 3.7: ***Parameter* $\mathbf{\Omega_b h^2}$**: *The minimum training loss is* 3.1073 *at the epoch* 734 *and the minimum validation loss is* 3.0643 *at the epoch* 736*. The absolute difference between the training loss and validation loss is* 0.0429*. The model demonstrates excellent performance in estimating the baryon matter density* ($\Omega_b h^2$)*. Note that the parameter is scaled between* $\{1, 100\}$ *before training*

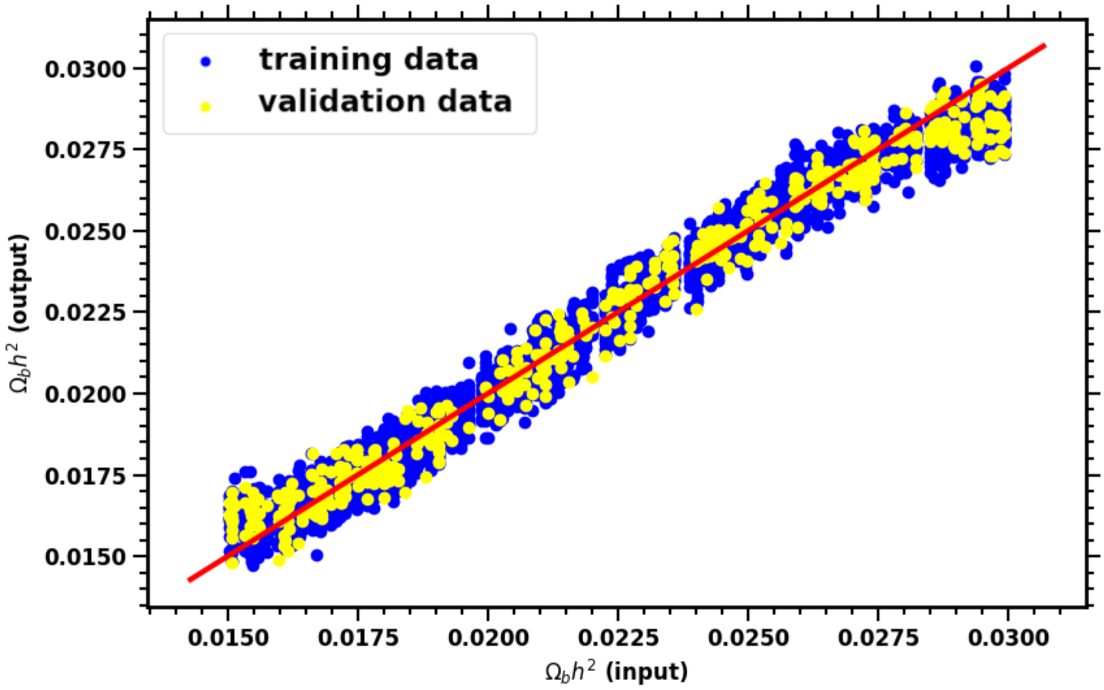

Figure 3.8: *Scatter plot showing the comparison of estimated '$\Omega_b h^2$' values by a CNN model on training and test dataset, with the solid line indicating the ideal case where estimated values would match the true values. The deviation of the blue (training data* $n = 8550$*) and yellow dots (validation data* $n = 950$*) from this ideal line indicates the level of error in the model's predictions."*

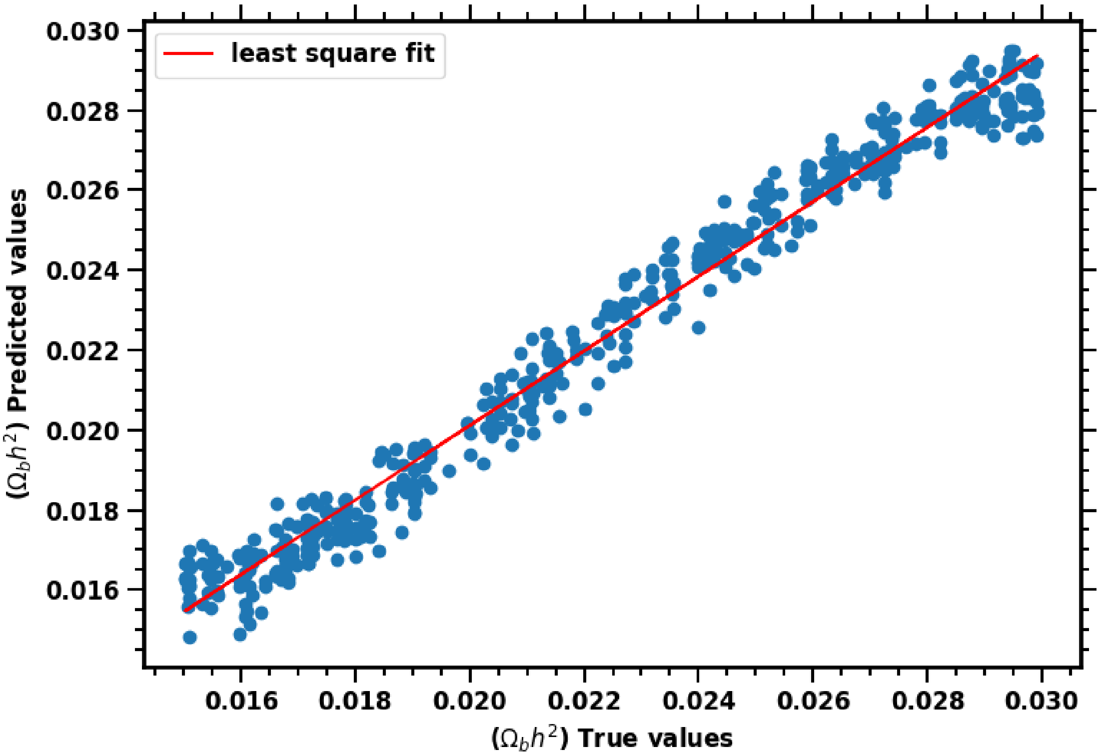

Figure 3.9: *Least square fit plot for baryon matter density parameter* $\Omega_b h^2$ *on the test dataset containing* $n = 500$ *maps, represented in bluedots. The least square fit line is represented in red with the intercept '*0.0014*', slope is '*0.9329*'. The observed correlation between the variables is robustly positive, as indicated by these values, which suggest that an increase in the input '*$\Omega_b h^2$*' is directly associated with a corresponding rise in the predicted '*$\Omega_b h^2$*' values.*

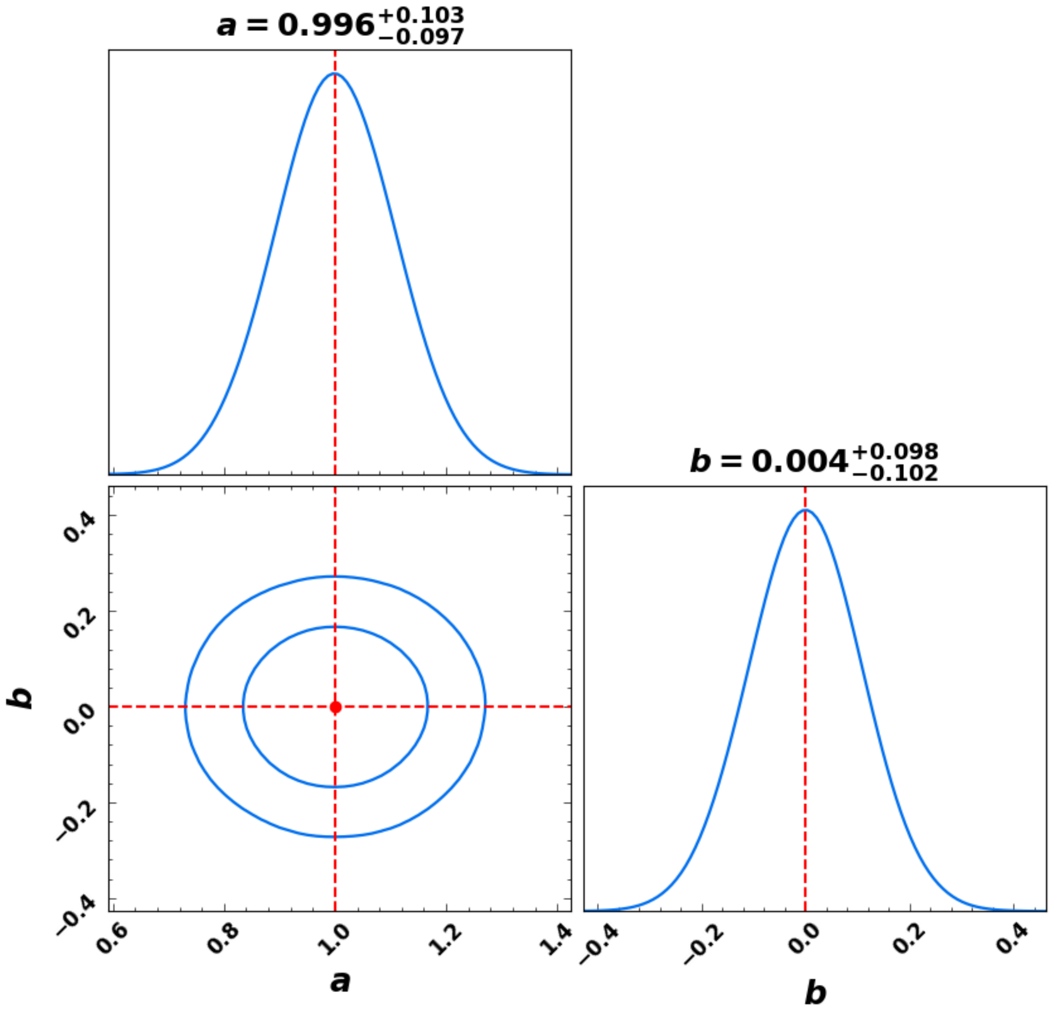

Figure 3.10: *Parameter $\Omega_b h^2$: Constraints on slope parameter a and interception b by using the test data. The two 2-D contours are the 68% and 95% confidence level of the parameters (a, b), and the diagonal plots are the marginalised distribution of the parameters.*

Cold dark matter density Parameter $\Omega_c h^2$

In the initial stages the validation loss is high which reduces down as the epoch increases. The model achieves a good performace with minimum value of training loss is 10.4885 at the epoch 439 and the minimum validation loss is 14.3275 at the epoch 471. The figure 3.11 shows a very slight under-fitting as validation loss tends to diverge from training loss at the end of the training but yet with an acceptable difference of 3.8390 between training and validation loss. There is an evidence of convergence and the plot shows a small gap between training and validation losses, which suggests a degree of under-fitting but within an acceptable range. The model's estimation of the CDM density ($\Omega_c h^2$) parameter exhibits good performance. Note that the parameter is scaled between $\{1, 1000\}$ before training.

The training set consists of $n = 8550$ maps represented in the scatter plot Fig. 3.12 and the validation set consist of $n = 950$ maps, with ground truth $x = y$ line in red. The deviation of the blue (training data) and yellow dots (validation data) from this ideal line indicates the level of error in the model's predictions.

Fig. 3.13 shows the relationship between the input and the predicted CDM density parameter '$\Omega_c h^2$' values from a machine learning model on the test dataset. The blue dots represent the input values, while the red line represents the least square fit of the input and predicted values '$\Omega_c h^2$'. The intercept of the line is 0.0004, and the slope is 0.9956, indicating a strong positive correlation between the two variables. The number of maps in test dataset is 500.

Fig. 3.14 shows the results of the Cold dark matter density parameter ($\Omega_c h^2$), the best-fitting slope parameter a and interception b are

$$a = 0.997^{+0.101}_{-0.099}, \quad b = 0.002^{+0.099}_{-0.101}, \tag{3.2.4}$$

Scalar Amplitude $\mathrm{A_s}$

The loss curves shown in the Fig. 3.15 exhibit smoothness and a consistent, gradual decrease. The training loss achieves its minimum value of 8.9905 by the 432^{nd} epoch, while

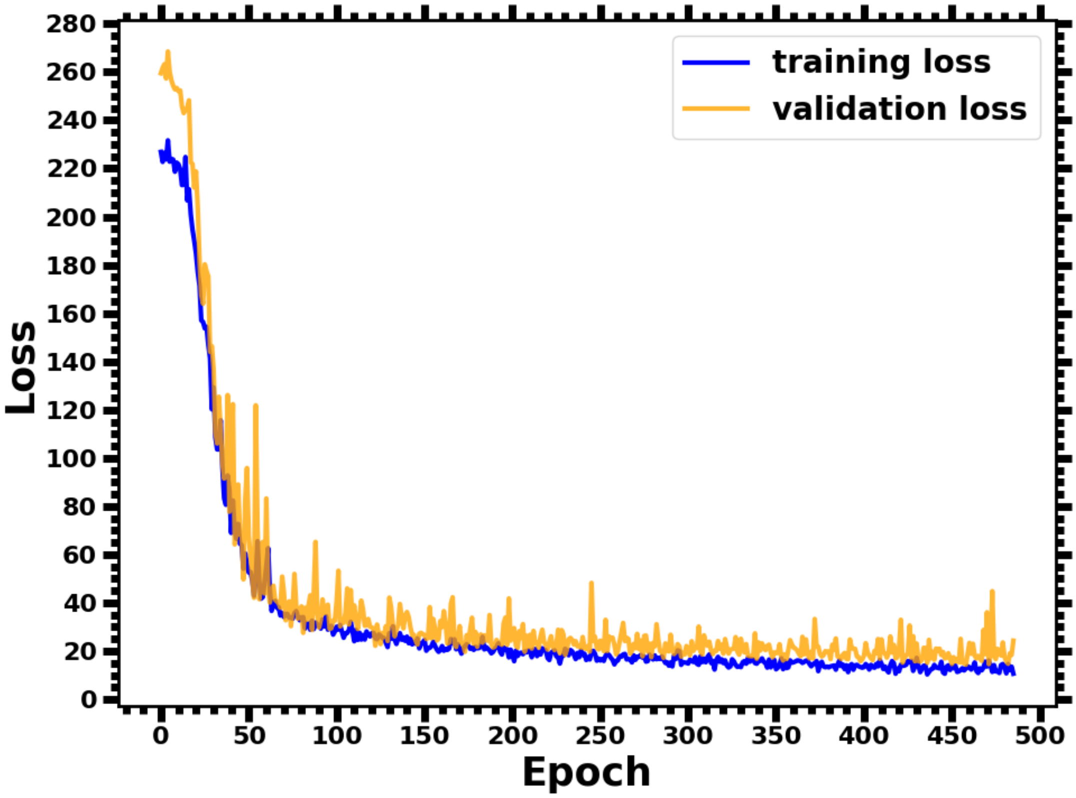

Figure 3.11: ***Parameter*** $\mathbf{\Omega_c h^2}$*: The loss curves display smoothness and a consistent, gradual decrease. The training loss reaches its minimum value of training loss is* 10.4885 *at the epoch* 439 *and the minimum validation loss is* 14.3275 *at the epoch* 471*. With an absolute difference of* 3.8390 *between these losses*

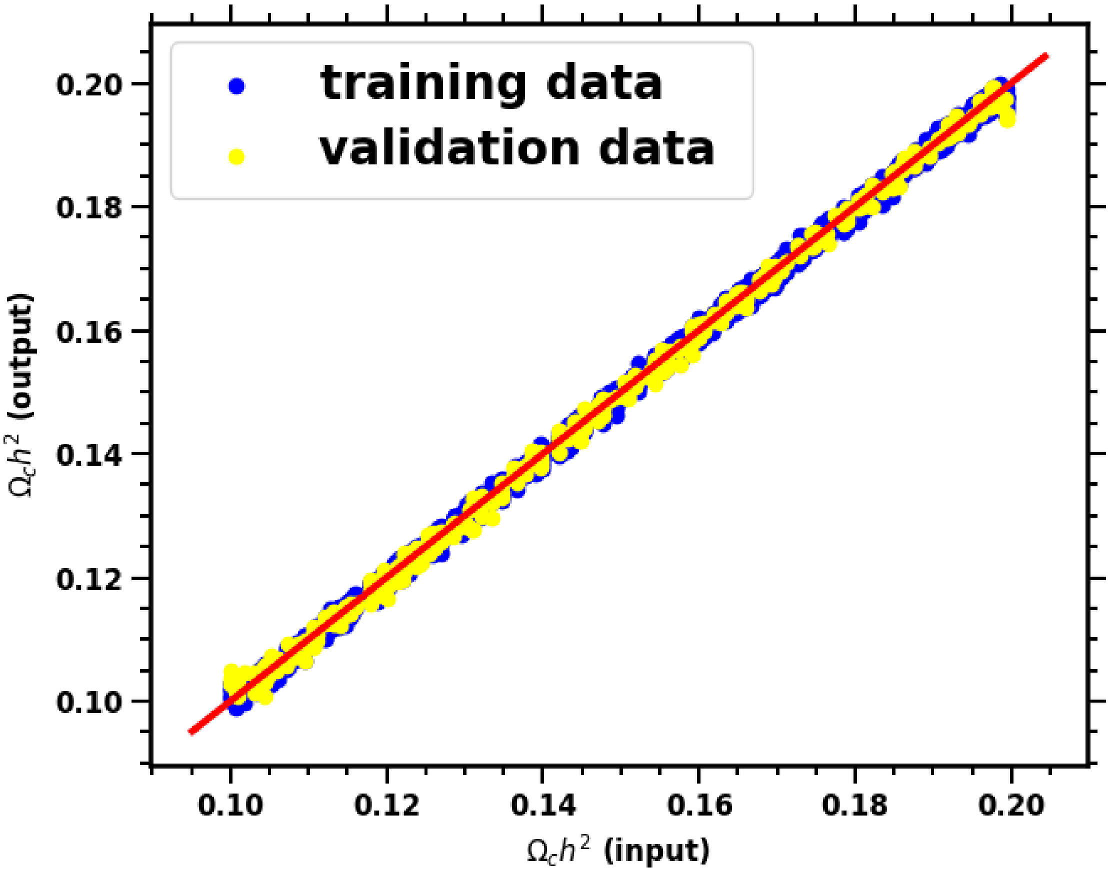

Figure 3.12: *Scatter plot showing the comparison of estimated '$\Omega_c h^2$' values by a CNN model on training ($n = 8550$) in blue dots and validation dataset ($n = 950$) in yellow dots, with the solid line indicating the ideal case where estimated values would match the true values.*

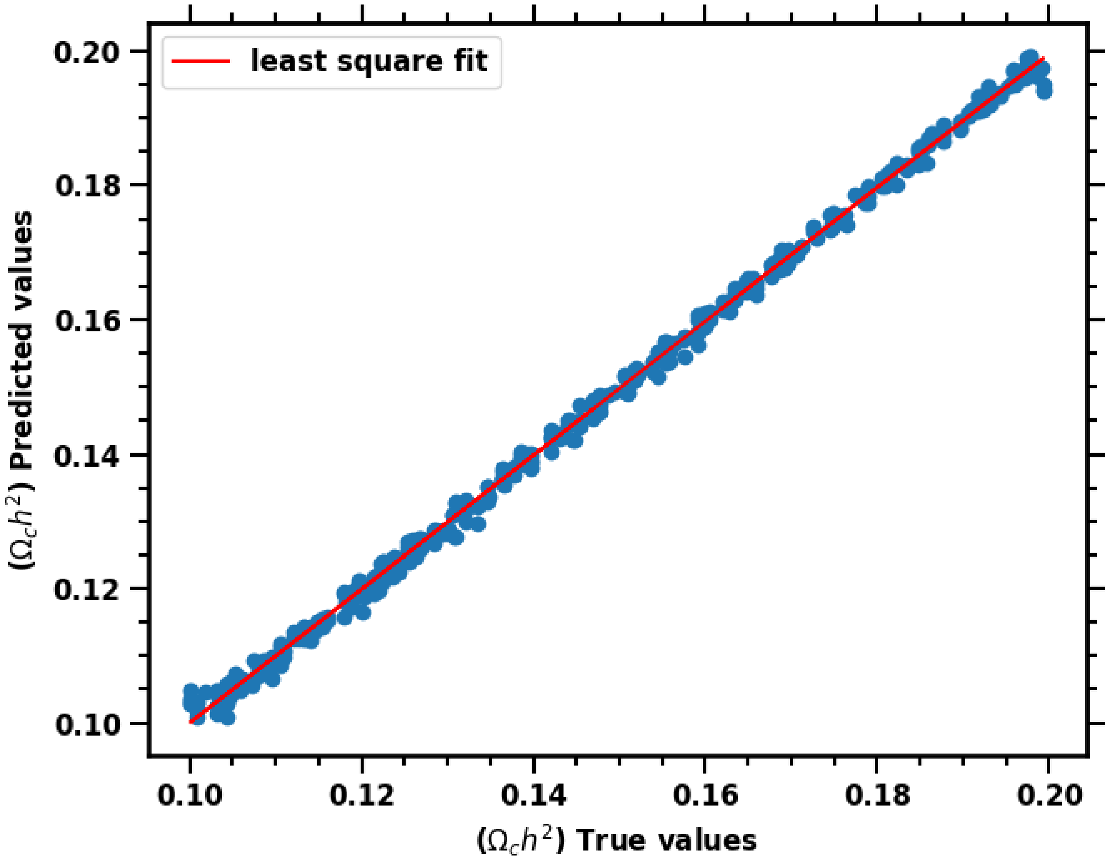

Figure 3.13: *Least square fit plot for cold dark matter density parameter* $\Omega_c h^2$ *on the test dataset containing* $n = 500$ *maps, represented in bluedots. The least square fit line is represented in red with the intercept* 0.0004 *and the slope is* 0.9956 *indicating a strong positive correlation between the two variables.*

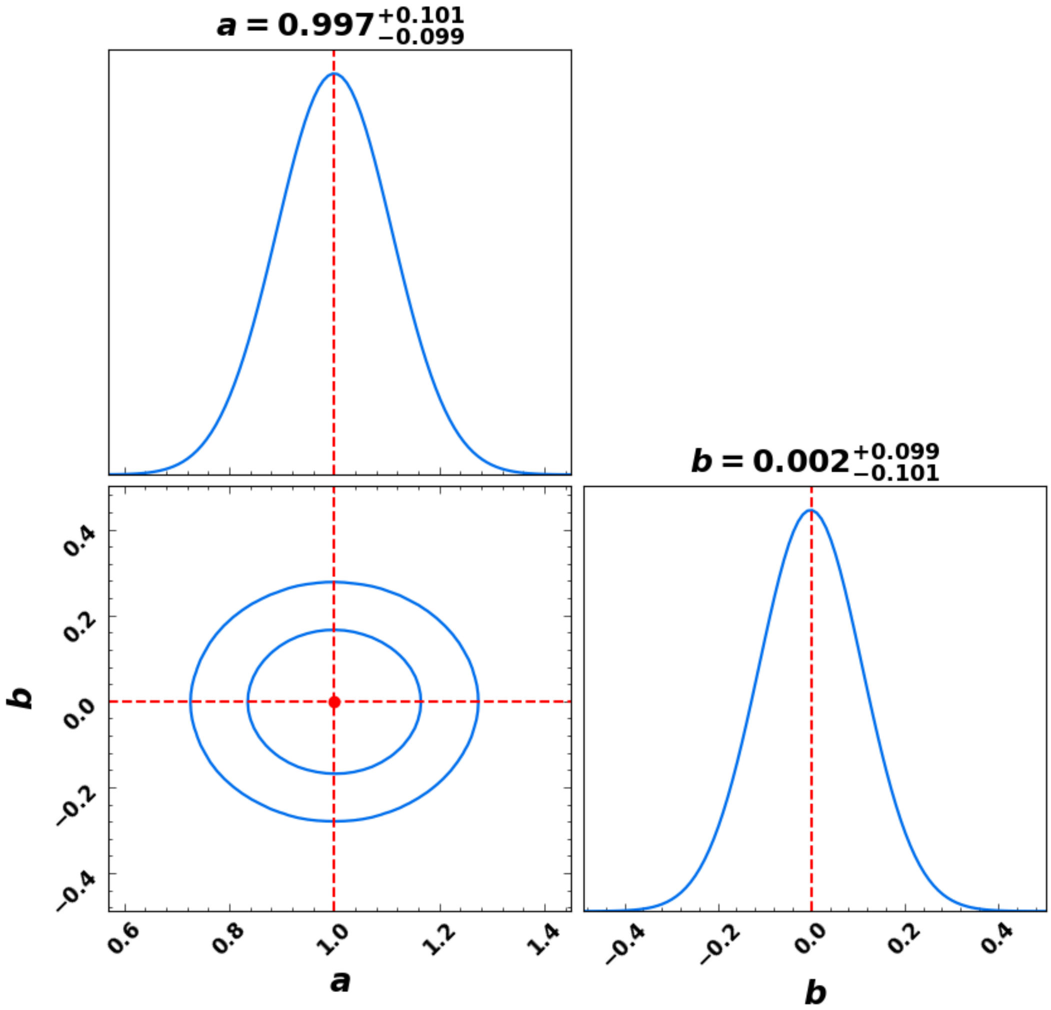

Figure 3.14: *Parameter $\Omega_c h^2$: Constraints on slope parameter a and interception b by using the test data. The two 2-D contours are the 68% and 95% confidence level of the parameters (a, b), and the diagonal plots are the marginalised distribution of the parameters.*

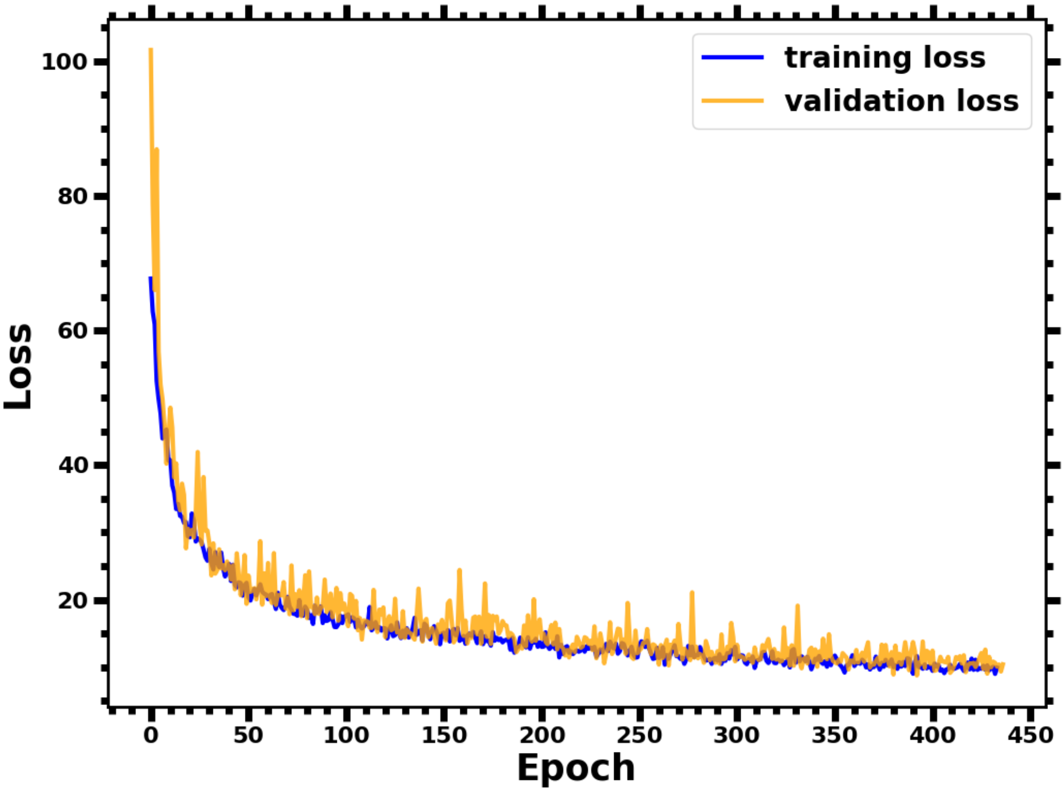

Figure 3.15: ***Parameter* $\mathbf{A_s}$**: *The loss curves display smoothness and a consistent, gradual decrease. The training loss reaches its minimum value of training loss is* 8.9905 *at the epoch* 432 *and the minimum validation loss is* 8.7651 *at the epoch* 392. *With an absolute difference of* 0.2253 *between these losses*

the validation loss reaches its minimum of 8.7651 at epoch 392^{nd}. Notably, the absolute difference between these losses amounts to 0.2253, suggesting convergence around comparable values. The model's estimation of the scalar amplitude of the primordial curvature pertubations (A_s) demonstrates exceptional performance. Note that the parameter is scaled between $\{1, 1000\}$ before training.

Scatter plot in Fig. 3.16 showing the comparison of estimated 'As' values by a CNN model on training and validation dataset, with the solid line indicating the ideal case where estimated values would match the true values. The deviation of the blue (training data $n = 8550$) and yellow dots (validation data $n = 950$) from this ideal line indicates the level of error in the model's predictions.

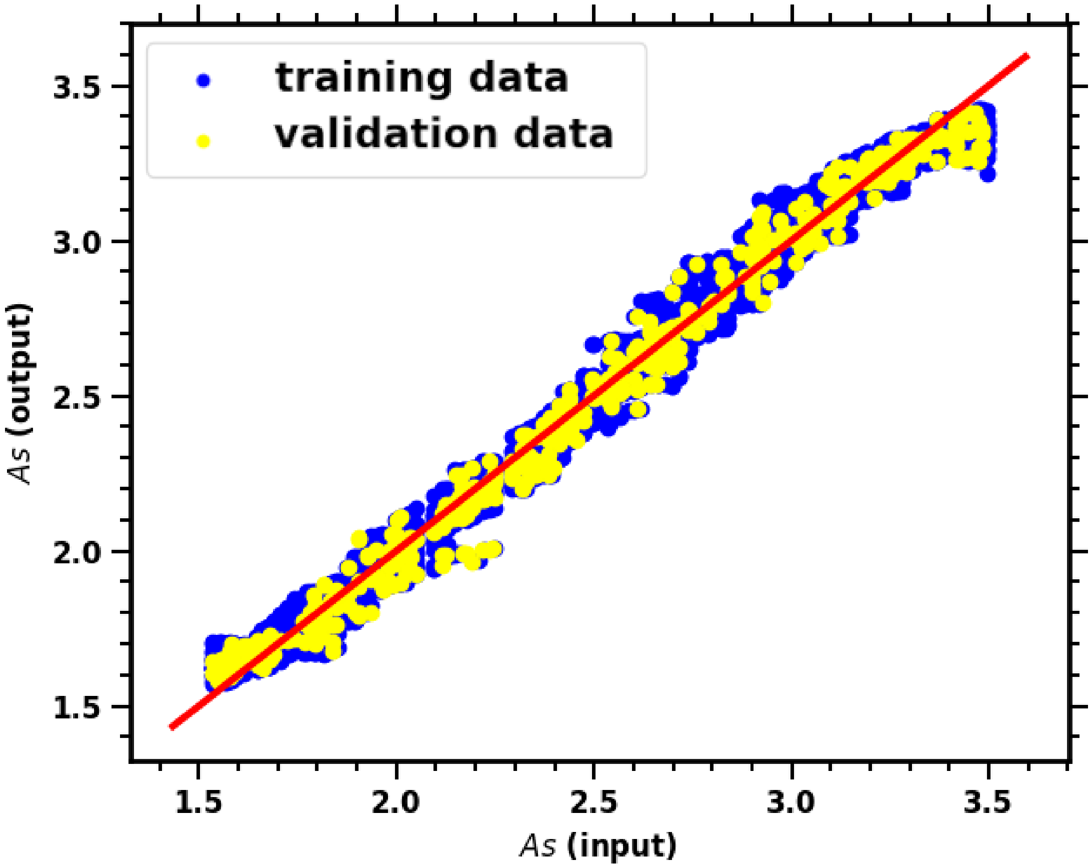

Figure 3.16: *Scatter plot showing the comparison of estimated 'A_s' values by a CNN model on training ($n = 8550$) in blue dots and validation dataset ($n = 950$) in yellow dots, with the solid line indicating the ideal case where estimated values would match the true values.*

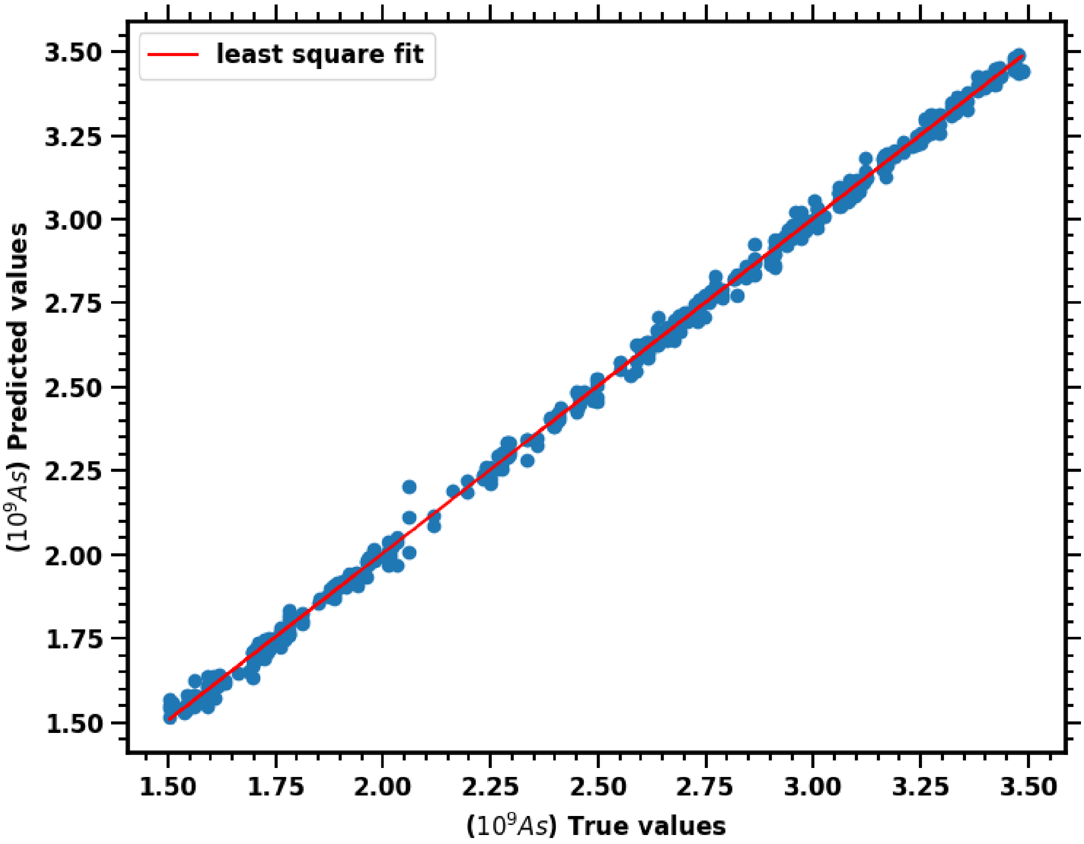

Figure 3.17: *Least square fit for the scalar amplitude* A_s*, slope is* 0.9890*, intercept is* 0.0155 *for the test dataset of size* $n = 500$ *represented in blue dots, red line represents the least square fit.*

The graph in the Fig. 3.17 shows the relationship between the input values and the predicted values of the scalar amplitude of the primordial curvature perturbations (A_s) from a machine learning model on the test dataset containing 500 maps. The blue dots represent the input values, while the red line represents the least square fit of the input and predicted values 'As'. The intercept of the line is 0.0155, and the slope is 0.9890, indicating a strong positive correlation between the two variables.

The Fig. 3.18 shows the results of the constraint for the scalar amplitude (A_s), the best-fitting slope parameter a and interception b are

$$a = 0.998^{+0.101}_{-0.100}, \quad b = 0.0.007^{+0.100}_{-0.101}, \tag{3.2.5}$$

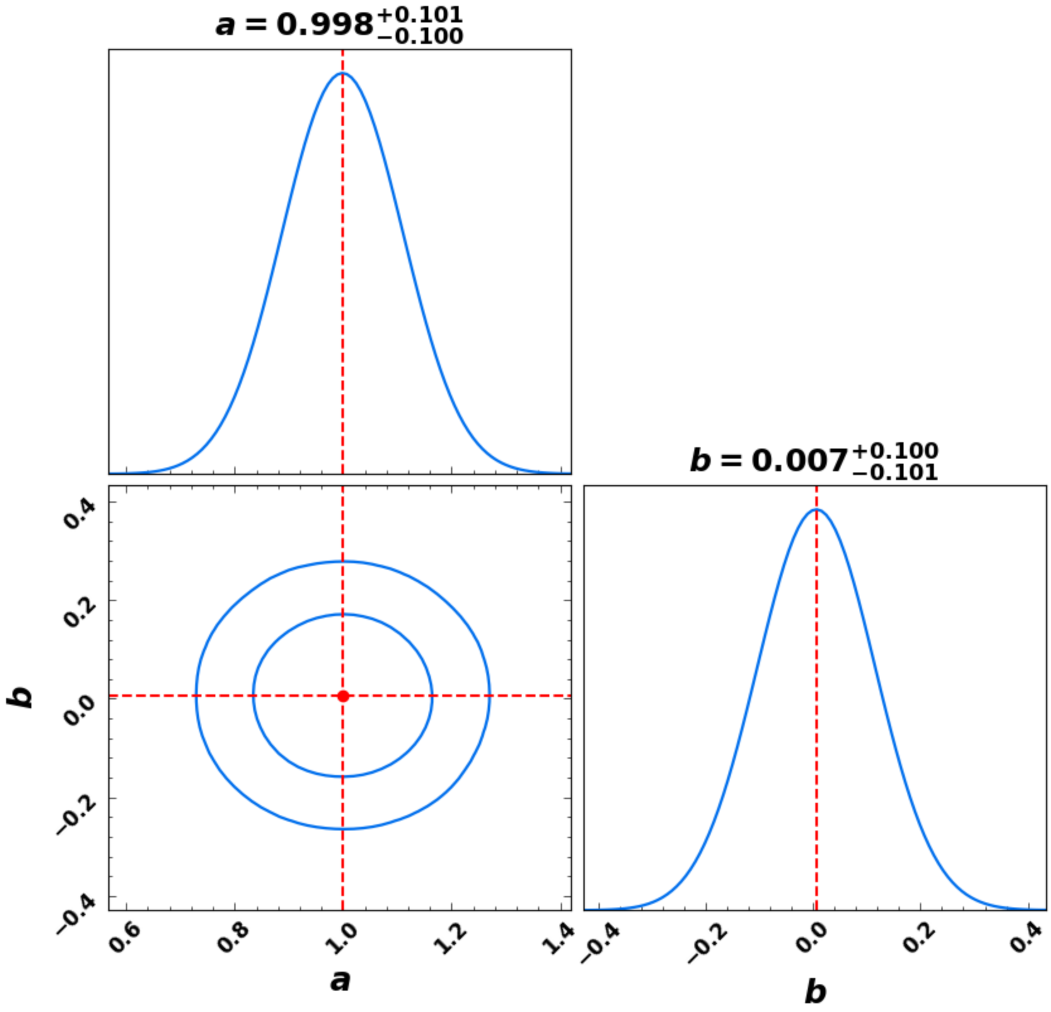

Figure 3.18: *Parameter* A_s*: Constraints on slope parameter* a *and interception* b *by using the test data. The two 2-D contours are the* 68% *and* 95% *confidence level of the parameters* (a, b)*, and the diagonal plots are the marginalised distribution of the parameters.*

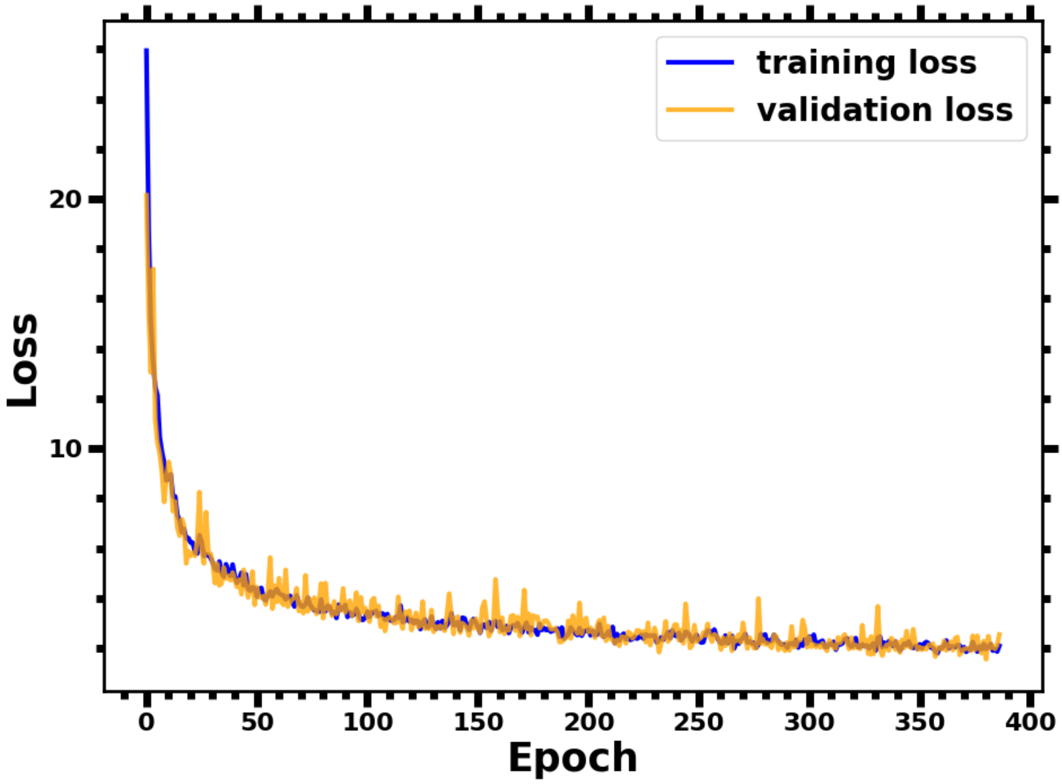

Figure 3.19: ***Parameter* $\mathbf{n_s}$**: *The loss curves show smoothness and a consistent, gradual decline. At the* 348^{th} *epoch, the training loss reaches its minimum value of* 1.9405*, while the validation loss hits its minimum of* 1.8214 *at the* 380^{th} *epoch.*

Spectral Index $\mathbf{n_s}$

The loss curves for the estimation of the spectral index of the primordial curvature perturbations (n_s) displays exceptional performance shown in Fig 3.19, with the training loss reaching its minimum value of 1.9405 at 348^{th} epoch, while the validation loss hits its minimum of 1.8214 at the 380^{th} epoch. Remarkably, the absolute difference between these losses is 0.1190, suggesting convergence around comparable values. The model's estimation of the spectral index of the primordial curvature perturbations (n_s) displays exceptional performance. It's important to note that the parameter is scaled between $1, 100$ before training.

The training set consists of $n = 8550$ maps represented in the scatter plot Fig. 3.20 and the validation set consist of $n = 950$ maps, with ground truth $x = y$ line in red.

Scatter plot shows the comparison of estimated 'n_s' values by a CNN model on training and test dataset. The deviation of the blue (training data) and yellow dots (validation data) from this ideal line indicates the level of error in the model's predictions.

The Fig. 3.21 shows the relationship between the input values and the predicted values from a machine learning model for the spectral index of the primordial curvature perturbations (n_s) on the test dataset of $n = 500$. The blue dots represent the input values, while the red line represents the least square fit of the input values and predicted values 'n_s'. The intercept of the line is 0.0002, and the slope is 1.0004, indicating a strong positive correlation between the two variables.

The Fig. 3.22 shows the results of the constraint for spectral index (n_s), the best-fitting slope parameter a and interception b are

$$a = 1.000^{+0.100}_{-0.100}, \quad b = 0.002^{+0.099}_{-0.101}, \tag{3.2.6}$$

The Optical depth τ

The loss curves exhibit smoothness with a consistent, gradual decrease. The training loss achieves its minimum value of 20.55 by the 548^{th} epoch, while the validation loss reaches its minimum of 15.0219 at the 530^{th} epoch. Notably, the absolute difference between these losses amounts to 5.5280, indicating convergence around comparable values. The model's estimation of the optical depth (τ) demonstrates exceptional performance. It's essential to note that the parameter is scaled between $1, 1000$ before training.

Scatter plot showing the comparison of estimated optical depth ('τ') by a CNN model on training and validation dataset, with the solid line indicating the ideal case where estimated values would match the true values. The deviation of the blue (training data $n = 8550$) and yellow dots (validation data $n = 950$) from this ideal line indicates the level of error in the model's predictions. There is a strong correlation with input and output values.

The Fig. 3.25 shows the relationship between the input and the predicted values for the optical depth (τ) from a machine learning model on the test dataset of $n = 500$.

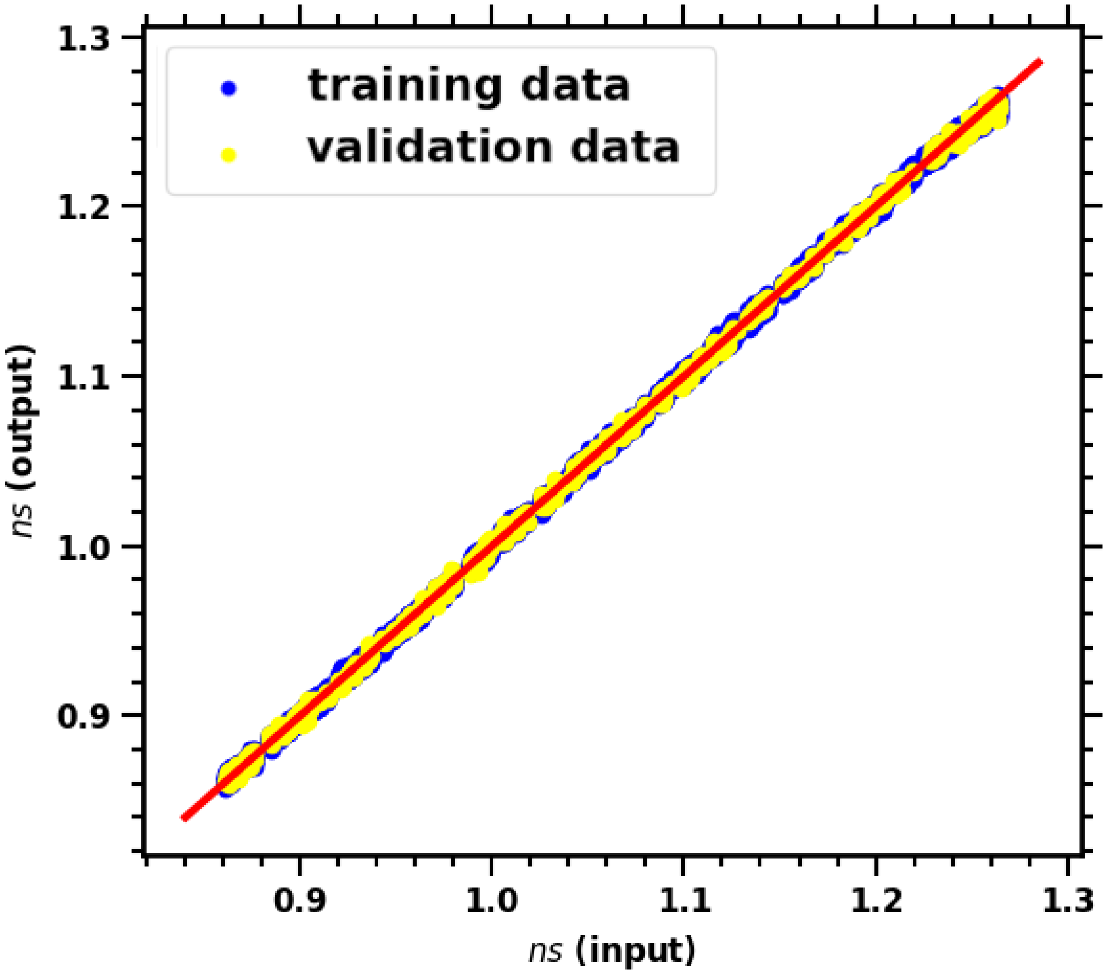

Figure 3.20: *Scatter plot showing the comparison of estimated 'n_s' values by a CNN model on training ($n = 8550$) in blue dots and validation dataset ($n = 950$) in yellow dots, with the solid line indicating the ideal case where estimated values would match the true values.*

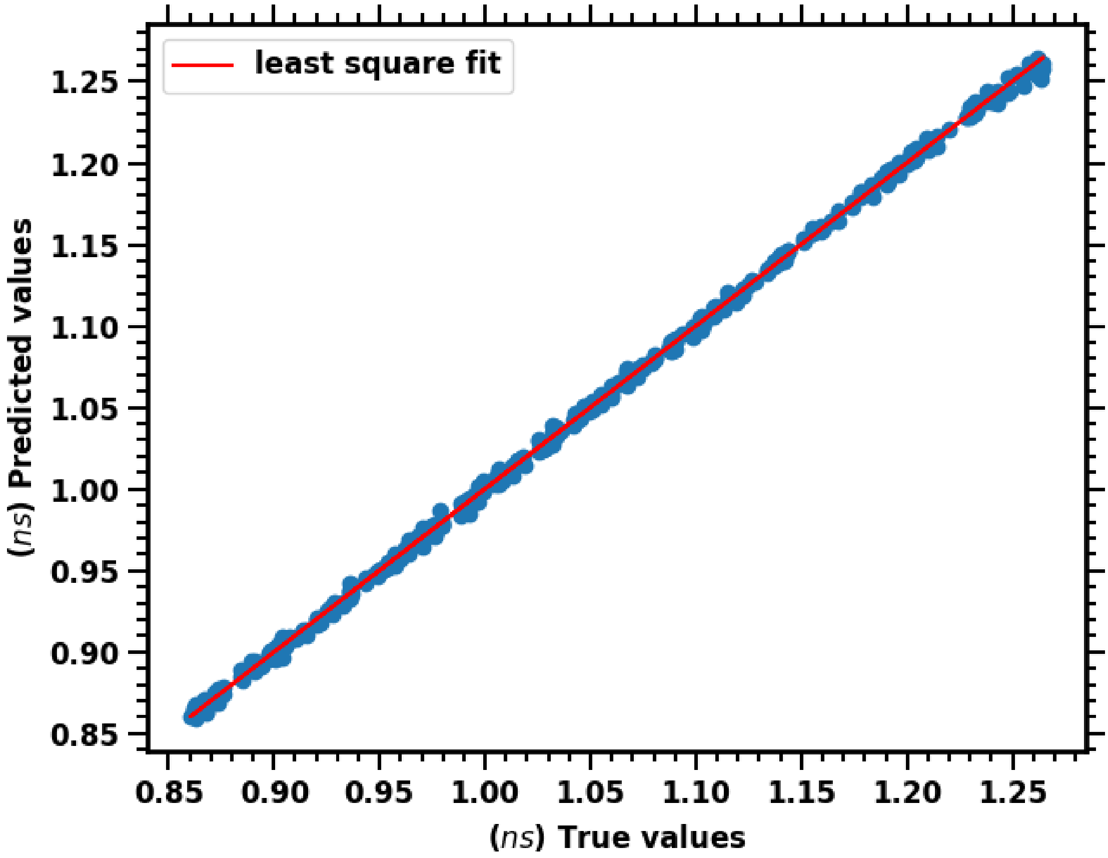

Figure 3.21: *The blue dots represent the input values* $n = 500$, *while the red line represents the least square fit of the input values '*n_s*' and predicted values '*n_s*'. The intercept of the line is* 0.0002, *slope is* 1.0004.

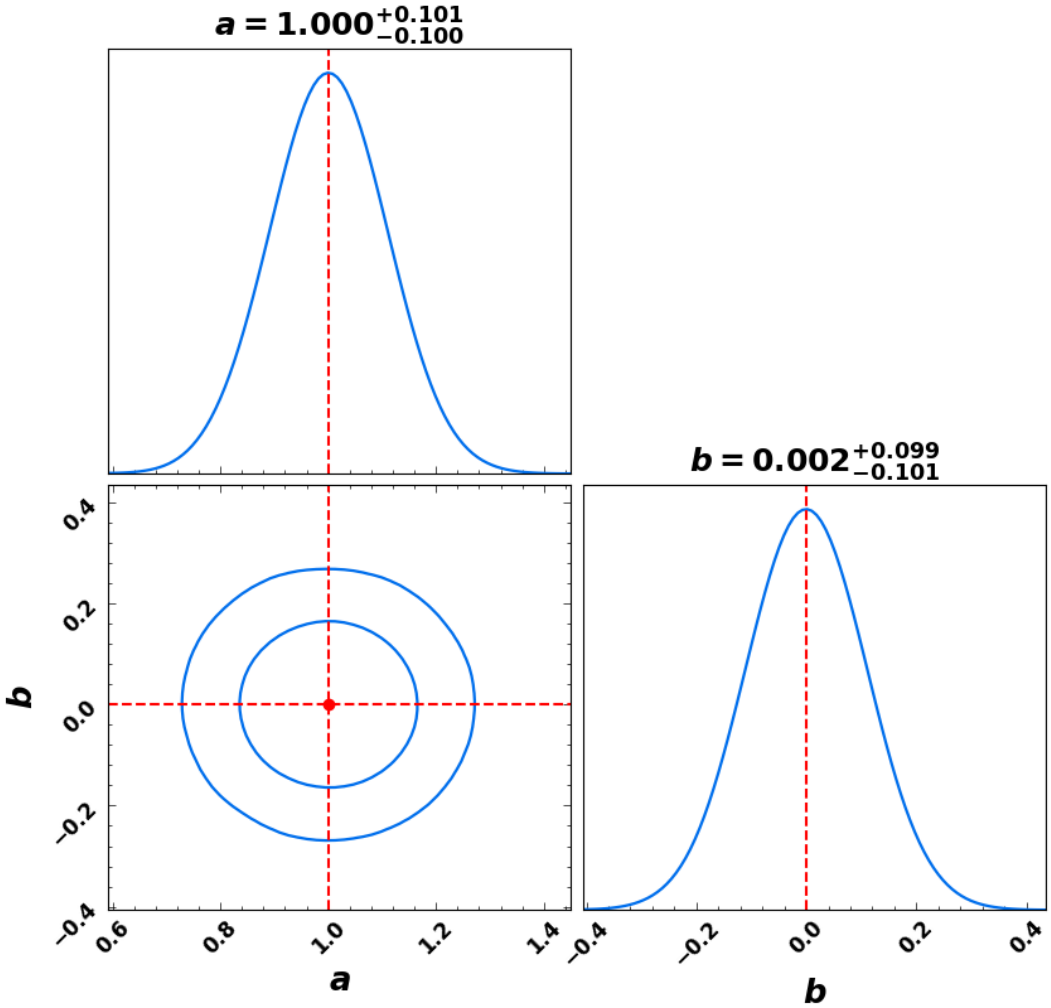

Figure 3.22: *Parameter n_s: Constraints on slope parameter a and interception b by using the test data. The two 2-D contours are the 68% and 95% confidence level of the parameters (a, b), and the diagonal plots are the marginalised distribution of the parameters.*

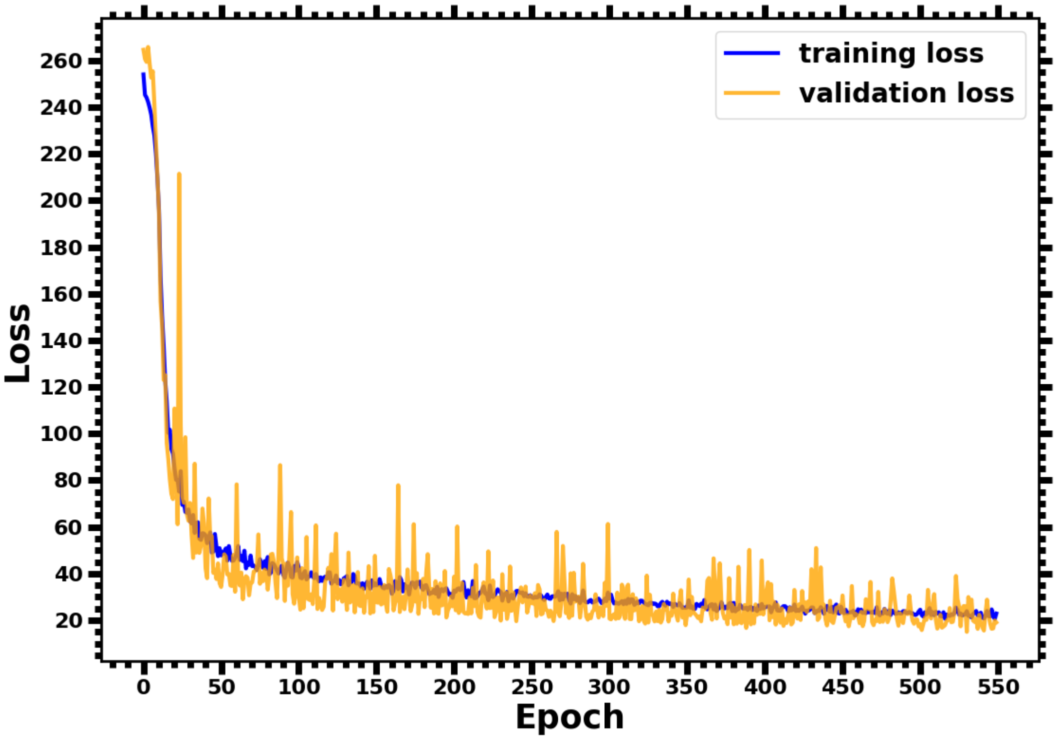

Figure 3.23: ***Parameter*** $\boldsymbol{\tau}$*: The training loss achieves its minimum value of* 20.55 *by the* 548*th* *epoch, while the validation loss reaches its minimum of* 15.0219 *at the* 530*th* *epoch. The absolute difference between these losses amounts to* 5.5280

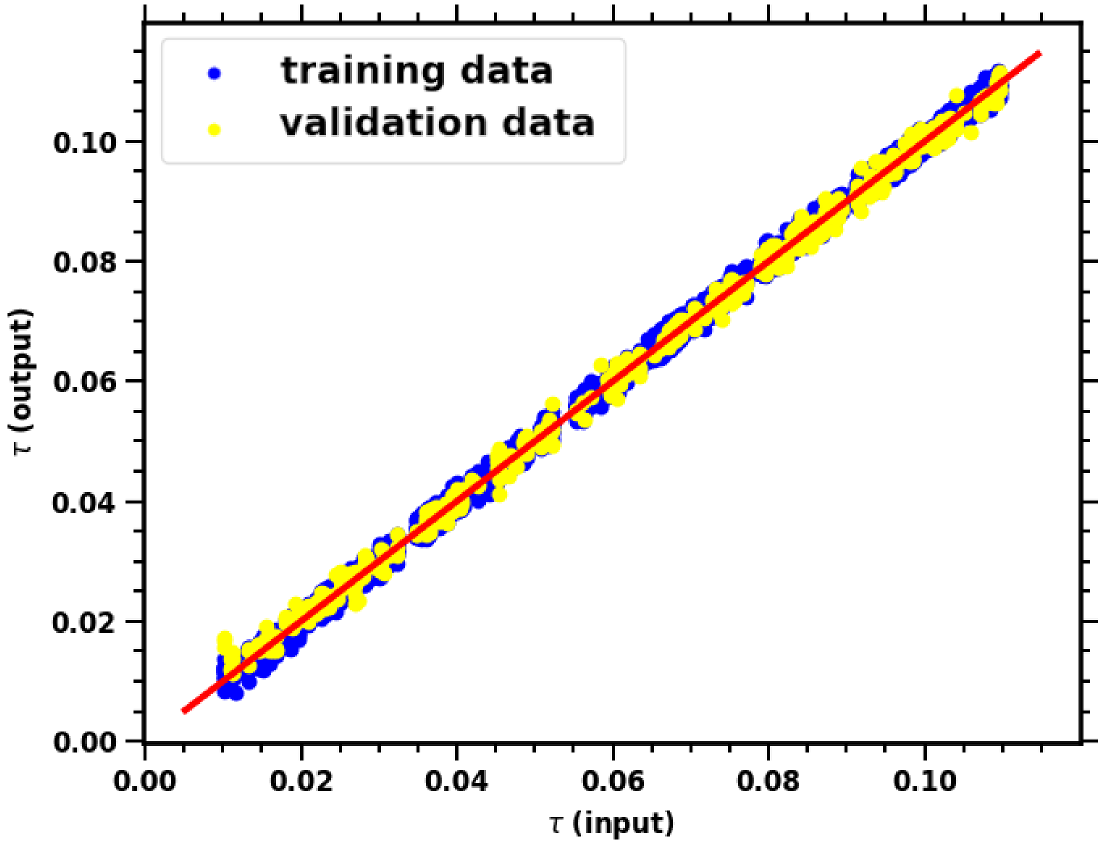
● training data
● validation data
0.00
0.02
0.04
0.06
0.08
0.10
τ (output)
τ (input)

Figure 3.24:

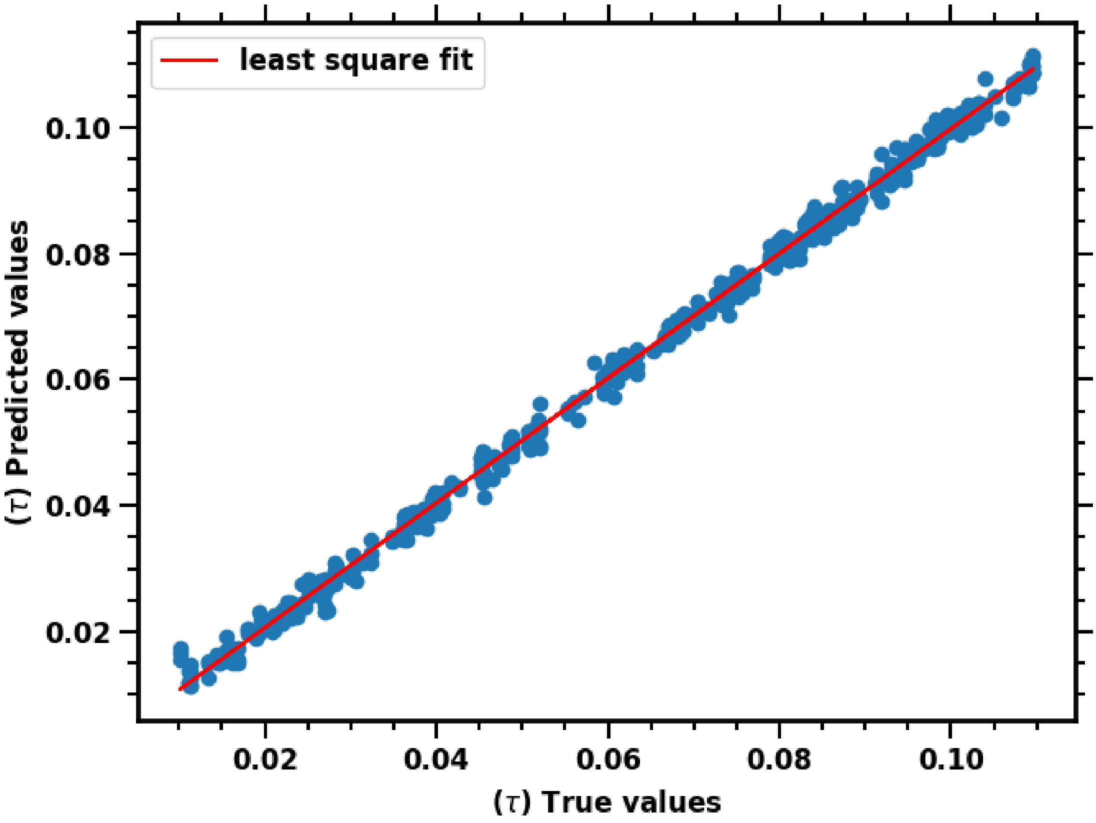

Figure 3.25: *Least square fit for the parameter Optical depth* τ*, slope is* 0.989 *and the intercept is* 0.0008*, the test dataset size* $n = 500$.

The blue dots represent the input values, while the red line represents the least square fit of the input values and predicted values 'τ'. The intercept of the line is 0.0008, and the slope is 0.9890, indicating a strong positive correlation between the two variables.

The Fig. 3.26 shows the results of the constraint for optical depth (τ), the best-fitting slope parameter a and interception b are

$$a = 0.989^{+0.100}_{-0.100}, \quad b = 0.003^{+0.099}_{-0.102}, \tag{3.2.7}$$

In summary, we have employed multiple techniques to evaluate the performance of our regression model. Firstly, we monitored the training and validation losses during the training process to ensure that the model was learning and generalizing effectively. This visualization helped us identify any potential overfitting or underfitting issues and adjust

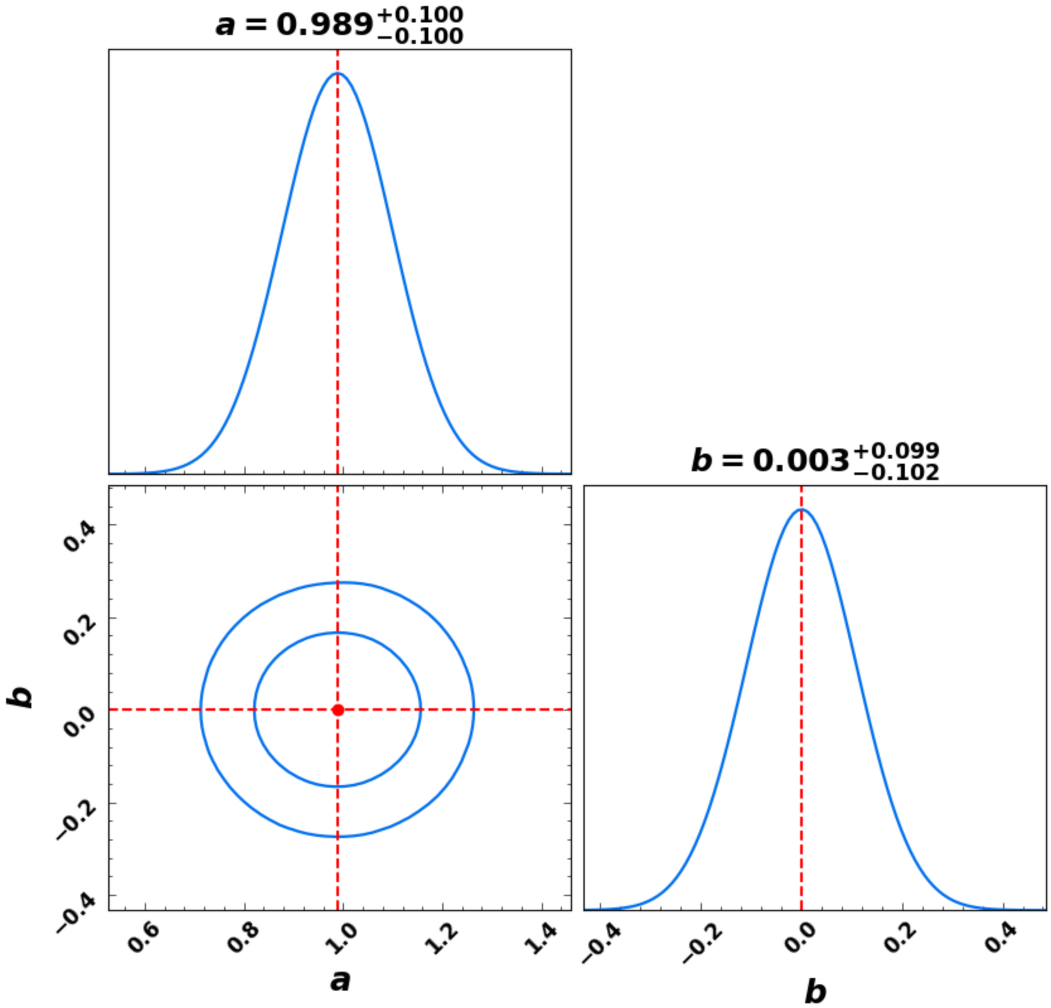

Figure 3.26: *Parameter* τ*: Constraints on slope parameter* a *and interception* b *by using the test data. The two 2-D contours are the* 68% *and* 95% *confidence level of the parameters* (a, b)*, and the diagonal plots are the marginalised distribution of the parameters.*

our hyperparameters accordingly. Hyperparameters are specific configuration settings or variables in machine learning and artificial intelligence models that are not learned from the data during the training process. Hyperparameters are typically set before the training begins and have a significant impact on the model's performance and behavior. Examples of hyperparameters include learning rate, number of hidden layers, and regularization strength. These parameters are crucial for fine-tuning the model and achieving optimal results, as they help control the model's complexity and generalization ability.

Furthermore, we calculated several evaluation metrics such as χ^2, R^2 to compare the predicted values with the actual values. These metrics provided us with a quantitative measure of the model's accuracy and correlation with the target variable.

Overall, by using a combination of these techniques, we were able to gain a comprehensive understanding of our regression model's performance and identify areas for improvement.

4. Conclusions and Future Work

4.1 Conclusion

This thesis embarked with a comprehensive review in Chapter 1, encompassing an array of theoretical cosmological subjects. It traversed through pivotal concepts like the Big Bang theory, exploring the expansion of the universe, and the foundational cosmological principles governing its homogeneous and isotropic nature. Noteworthy discussions revolved around the profound significance of cosmic microwave background (CMB) and its relevance in furthering scientific inquiry. The chapter delved into topics such as the halo model, baryonic acoustic oscillations, and redshift space distortion, offering a succinct yet comprehensive overview, elucidating their theoretical processes and deriving associated equations. Emphasizing the importance of a thorough understanding, the section navigated through fundamental mathematical and physical underpinnings crucial for contemporary cosmological studies. These included detailed derivations, unveiling the statistical and mathematical frameworks indispensable for constraining cosmological parameters. Chapter 1 also underscored the significance of constraining non-Gaussianity parameters and estimating cosmological parameters linked to the Lambda Cold Dark Matter (CDM) model of the universe, specifically from relic CMB radiation.

Chapter 2 provides a concise overview of the non-Gaussianity parameter denoted by $f_{\rm NL}^{\rm local}$ and its significance in the context of Cosmic Microwave Background (CMB) studies. Additionally, this chapter introduces basics of neural network and CNN, and our innovative approach aimed at estimating the $f_{\rm NL}^{\rm local}$ parameter from CMB data, leveraging the machine learning technique of Convolutional Neural Networks (CNNs). It details the generation of the requisite dataset, explains the CNN architecture employed for training, and delves into the intricate aspects of data preprocessing and augmentation methods pertinent to the model. We present the outcomes of our research, highlighting the Convolutional Neural Network's (CNN) remarkable efficacy in accurately estimating the $f_{\rm NL}^{\rm local}$ parameter across our training, validation, and test dataset. The intercept of the least-square fit is 0.274 and slope is 0.975 for the test dataset shows the strong positive correlation between input and predicted values. The loss function curves are smooth and the gradual decrease indicates the ability to learn and estimate the $f_{\rm NL}^{\rm local}$ parameter to the new data. The CNN exhibits an impressive **R-squared score** of **0.96**, underscoring its exceptional and consistent performance across the data. Notably, the process of generating $10,000$ maps necessitated approximately `4-6` CPU hours, while the CNN training and estimation phases consumed an estimated duration of `50-70 minutes` in a GPU node. For all our computation related to this work was carried out using CHPC server.

Chapter 3 delves into the significance of cosmological parameters and the imperative need to constrain them using Cosmic Microwave Background (CMB) maps. It outlines the limitations encountered when using conventional techniques for parameter constraints, establishing the rationale for adopting machine learning methods. The methodology details the process of generating requisite data for training each parameter independently using our customized Convolutional Neural Network (CNN), a modified version of the model introduced in Chapter (2). The independent estimation of six cosmological parameters $(H_0, \Omega_{\rm b}h^2, \Omega_{\rm c}h^2, A_{\rm s}, n_{\rm s}, \tau)$ demonstrates the CNN's exemplary performance in parameter estimation. The gradual decrease in loss curves throughout the learning process underscores the model's proficiency. A comprehensive summary of the actual values and R-squared scores for each parameter is presented in a tabular format for easy reference.

parameter	range	scaling	training loss	validation loss	R^2 score
H	$(64.5, 74.5)$	none	1.5338	1.6624	0.951
$\Omega_{\rm b}h^2$	$(0.015, 0.030)$	{1,100}	3.1073	3.0643	0.903
$\Omega_{\rm c}h^2$	$(0.1, 0.2)$	{1,1000}	10.4885	14.3275	0.968
$10^9 A_{\rm s}$	$(1.5, 3.5)$	{1,1000}	8.9905	8.7651	0.972
$n_{\rm s}$	$(0.85, 1.265)$	{1,100}	1.9405	1.8214	0.999
τ	$(0.01, 0.11)$	{1,1000}	20.55	15.0219	0.974

Table 4.1: summary of the parameter estimation.

In summary, the table (3.1.6) presents a list of hyperparameters for each trained and estimated cosmological parameter, along with their dataset range. The table also features the corresponding target scaling range for each estimated cosmological parameter. It also features the Min-Max scaling range applied prior to model training, the lowest training loss attained during an epoch, the minimal validation loss, and the r-square score. The r-square score ranges from 0 (worst performance) to 1 (best performance).

The process of generating 10000 maps necessitated approximately `4-6` CPU hours, while the CNN training and estimation phases consumed an estimated duration of `90-120 minutes` in a GPU node at CHPC. For all our computation related to this work was carried out using CHPC server.

4.2 Future Work

We've demonstrated the neural network's sensitivity to changes in the non-Gaussian parameter within CMB maps in the Chapter (2). To optimize the neural network's training comprehensively, our goal involves generating both Gaussian and non-Gaussian maps across various values of the cosmological constant. Presently, due to the absence of a method for generating non-Gaussian maps, we've utilized available maps limited to a specific set of cosmological constants from (Elsner & Wandelt (2009)). Our future focus is to create our own non-Gaussian maps, expand the dataset's diversity, and extend its application to the real Planck survey.

In other words, the current parameter estimation method involves separately estimating the values of each of the six parameters using individual datasets that vary only one parameter at a time. However, our future goal is to develop a method that estimates all six parameters simultaneously using a single dataset where all parameters are varied simultaneously. This approach, which has the potential to provide more accurate and reliable results as it considers the interdependence and correlation between the parameters. The resulting CMB maps obtained where all the parameters are varied simultaneously is a unified dataset that will be used for training, validation, and testing in the model. Our preliminary groundwork in this direction has shown promising results. However, due to the intricate nature of the data, achieving this goal requires a substantially larger training sample, demanding more memory resources. We are committed to overcoming these challenges and conducting simultaneous estimations for all six parameters, ultimately applying this approach to the actual *Planck* survey.

The application of CNNs to cosmology goes beyond just image classification or regression and can be extended to various other tasks. One such application is the recognition of radio frequency interference (RFI) in radio astronomy data. RFI is a major problem in radio astronomy as it can significantly degrade the quality of data obtained from telescopes. It arises from various sources, such as terrestrial communication systems, aircraft, and weather balloons, and can interfere with the weak signals from distant celestial objects. Traditional methods for RFI mitigation involve filtering techniques, which may not be effective for complex RFI signals. One approach to applying CNNs to time series data is to use a variant called a convolutional recurrent neural network (CRNN). This combines the convolutional layers of a CNN with the recurrent layers of a recurrent neural network (RNN) to create a model that can process sequential data. Another approach is to use a variant called a temporal convolutional network (TCN). This is a deep learning architecture that is specifically designed for processing time series data. It uses dilated convolutions to capture long-range dependencies in the data and has shown promising results in various applications, including speech recognition and stock price prediction. CNNs, on the other hand, have shown promising results in RFI detection and mitigation. The idea is to train a CNN to distinguish between RFI signals and astronomical signals using labeled data. The CNN learns to extract features from the data that are characteristic of RFI or astronomical signals, which can then be used to classify new, unseen data.

Using CNNs to gravitational lensing problems. Gravitational lensing is a phenomenon where the gravity of massive objects, such as galaxies and clusters of galaxies, bends the path of light from more distant objects behind them. This can result in distorted images of the background objects, which can provide valuable insights into the distribution of dark matter in the universe. CNNs can be used to detect and classify gravitational lenses in large-scale surveys. This involves training a CNN on a dataset of simulated and observed gravitational lens images and using it to classify new images as either lenses or non-lenses. This can help us identify new lenses and better understand their properties. Another application is to analyze gravitational lensing data, we need to extract features from the images and make predictions about the properties of the lensing objects and the underlying dark matter distribution. Neural networks, particularly CNNs, are well-suited for this task because they can learn to extract complex features from images and make predictions based on those features. One application of CNNs in gravitational lensing analysis is to predict the mass distribution of lensing objects from their observed images. This involves training a CNN on a dataset of simulated gravitational lensing images with known mass distributions and using it to predict the mass distributions of new, unseen images. This can help us understand the properties of lensing objects and their relationship to dark matter.

Bibliography

Abazajian K., et al., 2022, arXiv e-prints, p. arXiv:2203.08024

Ade P., et al., 2019, Journal of Cosmology and Astroparticle Physics, 2019, 056

Aiola S., et al., 2020, Journal of Cosmology and Astroparticle Physics, 2020, 047–047

Alam S., Ata M., Bailey S., 2017, Monthly Notices of the Royal Astronomical Society, 470, 2617–2652

Alexander D., Stott J., Hickox R. C., 2012, Monthly Notices of the Royal Astronomical Society, 421, 0035–8711

Allen T. J., Grinstein B., Wise M. B., 1987, Physics Letters B, 197, 66

Andernach H., Zwicky F., 2017, English and Spanish Translation of Zwicky's (1933) The Redshift of Extragalactic Nebulae (`arXiv:1711.01693`)

Bardeen J. M., 1980, Physical Review D, 22, 1882

Bardeen J. M., Steinhardt P. J., Turner M. S., 1983, Phys. Rev. D, 28, 679

Barreira A., 2022, Journal of Cosmology and Astroparticle Physics, 2022, 013

Battye R. A., Browne I. W. A., Dickinson C., Heron G., Maffei B., Pourtsidou A., 2013, Monthly Notices of the Royal Astronomical Society, 434, 1239

Bennett C., Larson D., Weiland J., 2013, The Astrophysical Journal Supplement Series, 208

Benson B. A., et al., 2014, in Holland W. S., Zmuidzinas J., eds, Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VII. SPIE, doi:10.1117/12.2057305, `http://dx.doi.org/10.1117/12.2057305`

Bernardeau F., Colombi S., Gaztañaga E., Scoccimarro R., 2002, Physics Reports, 367, 1

Bernstein J., 1988, Kinetic Theory in the Expanding Universe. Cambridge Monographs on Mathematical Physics, Cambridge University Press, doi:10.1017/CBO9780511564185

Beutler F., Blake C., Colless M., 2011, Monthly Notices of the Royal Astronomical Society, 416, 3017–3032

Cabass G., Ivanov M. M., Philcox O. H. E., Simonović M., Zaldarriaga M., 2022a, Physical Review D, 106, 043506

Cabass G., Ivanov M. M., Philcox O. H. E., Simonović M., Zaldarriaga M., 2022b, Physical Review Letters, 129, 021301

Challinor A., Peiris H., 2009, in Novello M., Perez S., eds, American Institute of Physics Conference Series, volume 1132 of American Institute of Physics Conference Series. pp 86–140,, doi:10.1063/1.3151849.

Chambers A., Rajantie A., 2008, Physical Review Letters, 100, 041302

Chang T.-C., Pen U.-L., Peterson J. B., McDonald P., 2008, Physical Review Letters, 100

Chang T.-C., Pen U.-L., Bandura K., Peterson J., 2010, Nature, 466, 463–465

Chen Y., Kumar S., Ratra B., 2017, The Astrophysical Journal, 835, 86

Choi S. K., et al., 2020, Journal of Cosmology and Astroparticle Physics, 2020, 045–045

Coles P., Lucchin F., 2002, Cosmology: The Origin and Evolution of Cosmic Structure, Second Edition

Colless M., Dalton G., Maddox S., 2001, Monthly Notices of the Royal Astronomical Society, 328, 1039–1063

Cooray A., Sheth R., 2002, Physics Reports, 372, 1–129,

Curto A., Martínez-González E., Barreiro R. B., 2011, Monthly Notices of the Royal Astronomical Society, 412, 1038

Dodelson S., 2003, Modern cosmology

Dodelson S., Schmidt F., 2020, Modern Cosmology, doi:10.1016/C2017-0-01943-2.

Doroshkevich A., Shandarin S., Saar E., 1978, Monthly Notices of the Royal Astronomical Society, 184, 643–660

Dunkley J., et al., 2009, The Astrophysical Journal Supplement Series, 180, 306

Dutcher D., et al., 2021, Physical Review D, 104

East W. E., Kleban M., Linde A., Senatore L., 2016, Journal of Cosmology and Astroparticle Physics, 2016, 010–010

Elsner F., Wandelt B. D., 2009, The Astrophysical Journal Supplement Series, 184, 264

Enqvist K., Jokinen A., Mazumdar A., Multamäki T., Väihkönen A., 2005, Physical Review Letters, 94, 161301

Eskilt J. R., et al., 2023, Astronomy and Astrophysics, 678, A169

Faber S. M., Gallagher J. S., 1979, Annual Review of Astronomy and Astrophysics, 17, 135

Falk T., Rangarajan R., Srednicki M., 1993, Astrophysical Journal Letters, 403, L1

Famaey B., McGaugh S. S., 2012, Living Reviews in Relativity, 15

Fergusson J. R., Liguori M., S. S. E. P., van Tent B. J. W., 2017, Journal of Cosmology and Astroparticle

Fixsen D., Cheng E., Gales J., 1996, The Astrophysical Journal, 473

Freedman W. L., et al., 2001, The Astrophysical Journal, 553, 47

Freedman W. L., et al., 2019, The Astrophysical Journal, 882, 34

Gangui A., Lucchin F., Matarrese S., Mollerach S., 1994, The Astrophysical Journal, 430, 447

Georgakakis A., Forbes D. A., Norris R. P., 2000, Monthly Notices of the Royal Astronomical Society, 318, 124–138

George D., Huerta E. A., 2018, Physics Letters B, 778, 64

Górski K. M., Hivon E., Banday A. J., Wandelt B. D., Hansen F. K., Reinecke M., Bartelmann M., 2005, The Astrophysical Journal, 622, 759

Gupta A., Matilla J. M. Z., Hsu D., Haiman Z., 2018, Phys. Rev. D, 97, 103515

Guth A. H., 1981, Phys. Rev. D, 23, 347

Guth A. H., Pi S.-Y., 1982, Phys. Rev. Lett., 49, 1110

Hall A., Bonvin C., Challinor A., 2013, Phys. Rev. D, 87, 064026

Hawking W S., 1966, SINGULARITIES and the GEOMETRY of SPACE-TIME. Cambridge, doi:10.1142/9789812384935˙0010, `https://www.worldscientific.com/doi/abs/10.1142/9789812384935_0010`

Hezaveh Y. D., Perreault Levasseur L., Marshall P. J., 2017, Nature, 548, 555

Hikage C., Komatsu E., Matsubara T., 2006, The Astrophysical Journal, 653, 11

Hinshaw G., Weiland J., Hill R., 2009, The Astrophysical Journal Supplement Series, 180, 225–245

Hinshaw G., et al., 2013, The Astrophysical Journal Supplement Series, 208, 19

Hou Z., et al., 2014, The Astrophysical Journal, 782, 74

Hubble E., 1929, Proceedings of the National Academy of Sciences, 15, 168

Jokinen A., Mazumdar A., 2006, Journal of Cosmology and Astroparticle Physics, 2006, 003

Kiefer C., Polarski D., 2009, Why do cosmological perturbations look classical to us? (`arXiv:0810.0087`)

Knox L., 1995, 52.4307. URL, 52, 4307–4318

Kofman L., Linde A., Mukhanov V., 2002, Journal of High Energy Physics, 2002, 057

Kolb E. W., Turner M. S., 1990, Front. Phys, 69, 1–547

Komatsu E., Spergel D. N., 2001, Physical Review D, 63, 063002

Komatsu E., Spergel D. N., Wandelt B. D., 2005, The Astrophysical Journal, 634, 14

Komatsu E., et al., 2009, The Astrophysical Journal Supplement Series, 180, 330

Lewis A., Bridle S., 2011, CosmoMC: Cosmological MonteCarlo, Astrophysics Source Code Library, record ascl:1106.025 (ascl:1106.025)

Lewis A., Challinor A., Lasenby A., 2000, The Astrophysical Journal, 538, 473

Li Y.-C., Ma Y.-Z., 2017, Physical Review D, 96, 063525

Liddle A. R., Lyth D. H., 2000, Cosmological Inflation and Large-Scale Structure

Liguori M., Sefusatti E., Fergusson J. R., Shellard E. P. S., 2010, Advances in Astronomy, 2010, 980523

Linde A., 1982, Physics Letters B, 108, 389

Linde A., 2005, Particle Physics and Inflationary Cosmology (`arXiv:hep-th/0503203`)

Linde A., Mukhanov V., 1997, Physical Review D, 56, R535

Lyth D. H., Ungarelli C., Wands D., 2003, Physical Review D, 67, 023503

Ma Y.-Z., Taylor J. E., Scott D., 2013, Monthly Notices of the Royal Astronomical Society, 436, 2029

Maddox S., Efstathiou G., Sutherland W., Loveday J., Mannheim P. D., 1990, Monthly Notices of the Royal Astronomical Society, 242, 340–445

Maldacena J., 2003, Journal of High Energy Physics, 2003, 013

Mannheim P. D., 2006, Progress in Particle and Nuclear Physics, 56, 340

Mao X.-C., 2012, The Astrophysical Journal, 752, 10 1088 0004–637 752 2 80

Marian L., Hilbert S., Smith R. E., Schneider P., Desjacques V., 2011, Astrophysical Journal Letters, 728, L13

Mather J. C., et al., 1994, The Astrophysical Journal, 420, 439

Mather J. C., Fixsen D. J., Shafer R. A., Mosier C., Wilkinson D. T., 1999, The Astrophysical Journal, 512, 511

Milgrom M., 1983, The Astrophysical Journal, 270, 365

Mo H., van den Bosch F. C., White S., 2010a, Galaxy Formation and Evolution

Mo H., Bosch F., White S., 2010b, Galaxy Formation and Evolution

Mukhanov V., Feldman H., Brandenberger R., 1992, Physics Reports, 215, 203

Newman J. A., et al., 2013, The Astrophysical Journal Supplement Series, 208, 5

Neyman J., Scott E. L., 1952, The Astrophysical Journal, 116, 144

Peacock J. A., 1998, Cosmological Physics. Cambridge University Press, doi:10.1017/CBO9780511804533

Peebles P., Yu J., 1970, The Astrophysical Journal, 162

Penrose R., 1965, Phys. Rev. Lett., 14, 57

Perlmutter S., et al., 1999, The Astrophysical Journal, 517, 565

Peterson J. B., Aleksan R., Ansari R., 2009, in , Vol. 2010, In astro2010: The Astronomy and Astrophysics Decadal Survey. pp 234,

Planck Collaboration et al., 2014, A&A, 571, A24

Planck Collaboration et al., 2016, Astronomy and Astrophysics, 594, A17

Planck Collaboration et al., 2020a, A&A, 641, A6

Planck Collaboration et al., 2020b, Astronomy and Astrophysics, 641, A9

Planck Collaboration et al., 2020c, Astronomy and Astrophysics, 641, A9

Platscher M., Smirnov J., Meyer S., Bartelmann M., 2018, Journal of Cosmology and Astroparticle Physics, 2018, 009–009

Postman M., Lauer T. R., Szapudi I., Oegerle W., 1998, The Astrophysical Journal, 506, 33–44

Riess A. G., et al., 1998, The Astronomical Journal, 116, 1009

Riess A. G., Casertano S., Yuan W., Macri L. M., Scolnic D., 2019, The Astrophysical Journal, 876, 85

Rubin V. C., Ford W. Kent J., 1970, The Astrophysical Journal, 159, 379

Ryden B., 2017, Introduction to Cosmology. Cambridge University Press, `https://books.google.co.in/books?id=07WSDQAAQBAJ`

Salopek D. S., Bond J. R., 1990, Physical Review D, 42, 3936

Schneider P., 2014, Extragalactic Astronomy and Cosmology, doi:https://doi.org/10.1007/978-3-642-54083-7.

Scott D., Smoot G. F., 2010, arXiv e-prints, p. arXiv:1005.0555

Sievers J. L., et al., 2013, Journal of Cosmology and Astroparticle Physics, 2013, 060–060

Smoot G., Bennett C., Kogut A., 1992, Astrophysical Journal Letters, 396:L1–L5

Spergel D., Bean R., Dore O., 2007, The Astrophysical Journal Supplement Series, 170, 377–408

Starobinsky A., 1982, Physics Letters B, 117, 175

Trimble V., 1987, Annual Review of Astronomy and Astrophysics, 25, 425

Wang G.-J., Cheng C., Ma Y.-Z., Xia J.-Q., 2022, The Astrophysical Journal Supplement Series, 262, 24

Wang G.-J., Cheng C., Ma Y.-Z., Xia J.-Q., Abebe A., Beesham A., 2023, The Astrophysical Journal Supplement Series, 268, 7

Wechsler R. H., Tinker J. L., 2018, Annual Review of Astronomy and Astrophysics, 56, 435–487

Weinberg S., 1972, Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity

White M., Scott D., Silk J., 1994, Annual Review of Astronomy and Astrophysics, 32, 319

Wilson R., Penzias A., 1965, The Astronomical Journal, 70

Wilson M., Silk J., 1981, The Astrophysical Journal, 243, 14–25

Zehavi I., Zheng Z., Weinberg D. H., 2005, The Astrophysical Journal, 630, 1–27

Zehavi I., Zheng Z., Weinberg D. H., 2011, The Astrophysical Journal, 736, 59,

Zonca A., Singer L., Lenz D., Reinecke M., Rosset C., Hivon E., Gorski K., 2019, Journal of Open Source Software, 4, 1298

Zwicky F., 1933, Helvetica Physica Acta, 6, 110

de Swart J. G., Bertone G., van Dongen J., 2017, Nature Astronomy, 1, 0059