Emulation of HI 21-cm Signal from the Epoch of Reionization using Vision Transformers

M.Sc. Thesis

By

PRASAD RAJESH POSTURE

Supervisor: Dr. Suman Majumdar

DEPARTMENT OF ASTRONOMY, ASTROPHYSICS & SPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 20, 2025

Emulation of HI 21-cm Signal from the Epoch of Reionization using Vision Transformers

A THESIS

Submitted in partial fulfillment of the
requirements for the awards of the degree
of
Master of Science

by

PRASAD RAJESH POSTURE

DEPARTMENT OF ASTRONOMY, ASTROPHYSICS & SPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May 20, 2025

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Emulation of HI 21-cm Signal from the Epoch of Reionization using Vision Transformers** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE** and submitted in the **DEPARTMENT OF ASTRONOMY, ASTROPHYSICS AND SPACE ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from **June 2024** to **May 2025** under the supervision of **Dr. Suman Majumdar, Associate Professor, Department of Astronomy, Astrophysics and Space Engineering.**

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

12/05/2025

Signature of the student with date
PRASAD RAJESH POSTURE

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Suman Majumdar

12/05/2025
Signature of the Supervisor of
M.Sc. thesis (with date)
Dr. SUMAN MAJUMDAR

PRASAD RAJESH POSTURE has successfully given his/her M.Sc. Oral Examination held on **06/05/2025**.

Suman Majumdar

Signature(s) of Supervisor(s) of MSc thesis
Date: 12/05/2025

Manoneeta Chakraborty

Programme Coordinator, M.Sc.
Date: 19/05/2025

Manoneeta Chakraborty

Convenor, DPGC
Date: 19/05/2025

S. Das

HoD, DAASE
Date:

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Suman Majumdar, for his invaluable guidance, support, and encouragement throughout this thesis. His expertise and insights have been instrumental in shaping this work and have greatly influenced my academic growth.

I am deeply thankful to my labmates for their insightful discussions and for creating a collaborative and motivating atmosphere. I would like to especially thank my PhD seniors Yashraj, Leon, Mohit, Chandrashekhar, and Shiriny, who treated me like a younger brother and generously guided me throughout this journey. I am also grateful to my classmates and fellow labmates Parth and Vishrut, with whom I shared not just scientific discussions but also many fun and memorable moments in and out of the lab. To my classmates Vijay, Annie, Amar, Gitaj, Anand, Anushka, Ashutosh, Daisy, Harry, Aryan, Devesh, and Navanit, thank you for being part of this journey and for all the camaraderie and support along the way. I would also like to acknowledge PhD seniors Keshav, Aromal, Jibin, Biki, and Chandan for their help, encouragement, and guidance at various stages of my work.

Finally, I am forever grateful to my family and loved ones for their unwavering patience, encouragement, and belief in me throughout this journey. Your support has been my anchor. And last but not least, I would like to thank myself for not giving up, for showing up every day, and for keeping on going.

- Prasad Rajesh Posture

Dedicated to
Aai ani Baba

ABSTRACT

The Epoch of Reionization (EoR) refers to the era when the first luminous sources started to emit the UV radiation and ionized the neutral hydrogen (HI) present in the intergalactic medium (IGM). This era corresponds to one of the major phase transition of the Universe. Studying the redshifted 21-cm signal coming from the neutral hydrogen in the IGM can provide the solid answers to essential questions about EoR such as exact duration of these epochs, formation and properties of the ionizing sources, the morphology of ionized regions and its evolution.

Several ongoing radio experiments are working on the detection of the redshifted 21-cm signal, including LOFAR, HERA, MWA, and uGMRT, and the next generation radio interferometers, such as the SKA, will focus on the era of Cosmic Dawn and Epoch of Reionization (CD-EoR). We need to build forward models of the redshifted 21-cm signal that can interpret the observations made by these radio experiments. The simulations that are generally used for forward modeling are computationally expensive when it comes to rerunning them for a large CD-EoR parameter space. This creates a need to develop an emulator that is computationally cheap and can be employed for exploration of the EoR parameter space. While most existing approaches focus on emulating summary statistics of the 21-cm signal, such methods lose information compared to emulating the 21-cm signal itself.

Emulation of the EoR 21-cm field suffers from the fundamental problem of modeling features or fields which have a large dynamic range (e.g. 4-5 orders of magnitude variation). The previous efforts of emulation of the 21-cm signal faced limitations because they didn't take all length scales into account due to the bias of the model. This led to the underprediction of large and small-scale features depending on the model architecture. Furthermore, these models use fixed astrophysical parameters; hence, they cannot be used for any kind of inference exercise. Taking the drawbacks of these previous model architectures into account, we developed an emulator using Vision Transformers (ViTs) that takes in the entire 3D dark matter density and halo field as inputs, along with the three EoR parameters to predict the hydrogen neutral fraction field. The multi-head self-attention mechanism of the transformer makes it easier to capture the long-range dependencies, making it a perfect candidate for this task. The trained and validated model can quickly and accurately produce the neutral fraction fields for a given set of EoR parameters.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Epoch of Reionization

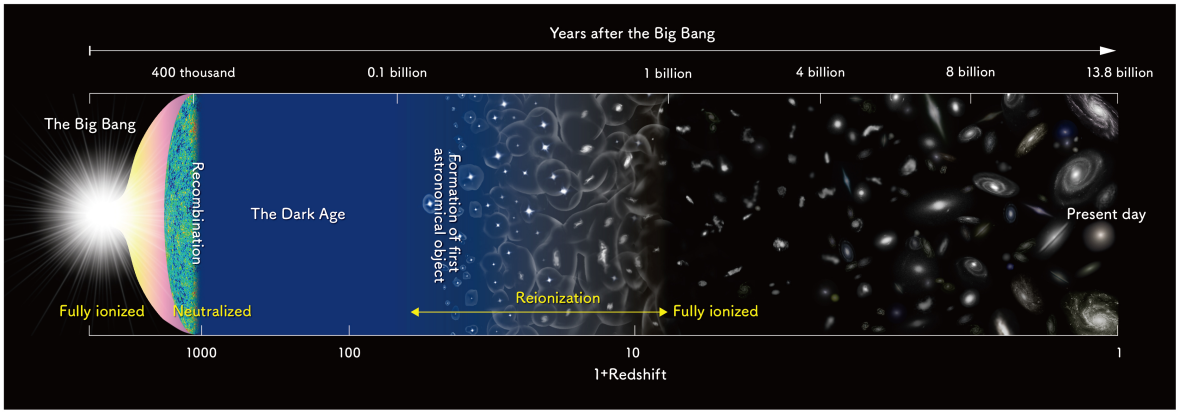


Figure 1.1: A timeline showing the evolution of the Universe. (Figure Credits: NAOJ)

Understanding the cosmic history using precise observations is the main aim of the modern cosmology. In the quest to understand modern cosmology, we expect there can be different periods of interest. One of them is the cosmic dark ages ($30 < z < 200$), which refers to the period when there were no luminous sources. The only signal observable from this era corresponds to the 21-cm line transition of the neutral hydrogen atom. After the first luminous sources were formed they emitted UV radiations that ionized the intergalactic medium (IGM) causing a significant change in the 21-cm signal. This period is known as the Epoch of Reionization (EoR). This epoch marks a major change in the history of the Universe, since the universe went from neutral to ionized [1].

The first stars that formed were made up of pristine gases such as hydrogen and helium. They emitted highly energetic photons into the IGM. The photons having energy equal to the ionization energy of hydrogen and helium got absorbed in the IGM. The photons with higher energies got redshifted to lower energies as the Universe expanded, and ionized the hydrogen which was further away [2]. The first stars produced elements with high metallicity in their core through stellar nucleosynthesis and after they ended with supernova, these heavy elements got

scattered in the IGM [3]. The medium with higher metallicity supported more efficient cooling through atomic and molecular transitions. This led to an increased star formation rate hence rapid reionization [4, 5]. The radiation emitted by these stars increased the temperature of the IGM, which prevented the hydrogen from recombining, but this thermal feedback prevented small-scale structure formation because more gas was needed for efficient cooling [2, 5]. Over the billion years, the ionized regions kept on growing till the Universe was completely ionized. The only places where you could find the neutral hydrogen post-EoR were the high-density regions such as galaxies, Interstellar Medium (ISM), Damped Lyman-α Absorbers (DLAs) [2, 6]. The observations suggest that the process of reionization started before z $\approx$ 7.5-8 and ended before z $\approx$ 6 [7, 8, 9].

1.2 Various Probes of EoR

The EoR marks the major phase transition of the Universe and provides crucial information about the first stars and galaxies in the universe. The observational evidences that help to study the EoR is listed in this section.

1.2.1 CMB Anisotropies

When the matter decoupled from the radiation, the radiation was scattered in all directions, which we see today as Cosmic Microwave Background Radiation (CMBR). These photons didn't have enough energy to interact with the atoms, but they got scattered from the electrons produced during reionization process. The measured quantity from the CMBR observations is the optical depth due to Thompson scattering off free electrons. Since the number density of electrons (n_e) increased due to reionization, the optical depth increased [10]. This scattering effects the angular power spectrum of the CMBR (C_l) where the spherical harmonic index l indicates roughly the angular size probed, $\theta \sim 1/l$. For no reionization case, $n_e = 0$, therefore $\tau = 0$ and if reionization is introduced the value of τ is non-zero. The temperature power spectrum shows that there is damping at low angular scales and also the polarization power spectrum shows damping at high angular scales. One cannot constrain τ from the temperature power spectrum since it is degenerated with amplitude of the dark matter power spectrum σ_8. However, the polarization power spectrum only depends on τ, since the reionization produces a polarization signal at large angular scales, which cannot be compensated by varying any other values. The current constraints provided by this on reionization history imply that the reionization started before redshift $z \approx 7.5 - 8$ [7].

1.2.2 Lyman-α Emitters (LAEs)

The Lyman-α transition in hydrogen refers to the electron transition from the ground state to the first excited state, which requires a photon with wavelength 1216 Å. The Lyman-α emitting galaxies offer an independent probe of reionization, as they get scattered by the neutral hydrogen [11, 12]. If the IGM contains a significant amount of neutral hydrogen fraction, the Lyman-α line flux emitted by the source situated in it, gets strongly attenuated [13]. However, during the reionization process, the presence of ionized regions around the galaxies enables Lyα photons to escape. Thus, directly correlating the LAE visibility to the state of the local IGM. Observations from JADES (JWST Advanced Deep Extragalactic Survey [14]) show declining fraction of Lyα-emitting galaxies at z>5.5, with constraints on neutral fraction of hydrogen as $x_{HI} \approx 0.64^{+0.13}_{-0.21}$ at $z \approx 7$ [15].

1.2.3 Lyman- α Forest

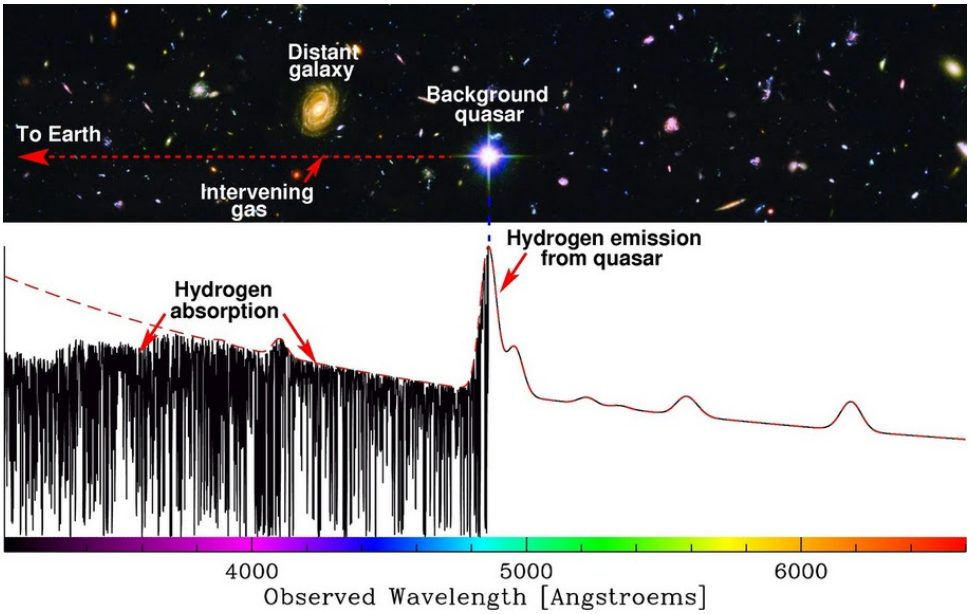


Figure 1.2: Quasar spectra with Lyman-α Forest. (Figure Credits: Bill Keel's Website*)

Quasars ,also known as quasi-stellar objects, are point-like bright sources emitting high-energy photons and hence visible from very high redshifts. All quasars show the prominent emission line as shown by a peak in Figure 1.2. which occurs at the rest frame of the Lyman-α frequency. All the absorption lines left to the prominent peak correspond to the Lyman-α transition of the neutral hydrogen. If you see the top panel of the Figure 1.2, we have radiations coming from the background quasar towards us that pass through the intervening gas containing hydrogen, this gas absorbs the photons which has energy equal to the energy required for Lyman-α transition, the high energy photons get redshifted as universe expands and their energy decreases to the

*https://www.futura-sciences.com/sciences/definitions/physique-foret-lyman-alpha-9973/

required energy and it excites the hydrogen atom which are further towards us giving the forest like features in the spectra [9].

The strength of the absorption spectra depends upon the amount of hydrogen present in the intervening medium, therefore in the absence of neutral hydrogen, the spectra would be flat without any absorption. While considering all the above reasons the quasar spectra carry a great amount of information about the density structure of the Universe in a given direction. Given million quasars along all the lines of sight, in principle, we should be able to map the distribution of neutral hydrogen in the Universe.

1.2.4 21-cm Line

Hydrogen is the most abundant element in the Universe making up to nearly 75% of the gas mass. It simple structure exhibits many wonderful phenomena, one of which is the 21-cm line transition [16]. The 21-cm line of neutral hydrogen arises from the hyperfine splitting of the ground state due to the magnetic moment interaction of the proton and electron The lower energy state is called the singlet state, denoted using subscript 0 and the higher energy state is called the triplet state, denoted using subscript 1. If n_i (where i=0,1) represents the number density of hydrogen atoms in these two hyperfine states, we can write their ratio as

$$\frac{n_1}{n_0} = \frac{g_1}{g_0} exp(-\frac{T_*}{T_S})$$

where $g_1/g_0 = 3$ denoting the ratio of the statistical degeneracy factors of the two levels, $T_* = hc/k\lambda_{21-cm} \approx 0.0682K$ and T_s is the spin temperature which is used for quantifying the relative population of the two splitted energy states [1]. The energy difference between these two splitted ground states i.e. the singlet and triplet state corresponds to the frequency 1420.4 MHz corresponding to the wavelength of 21.11 cm. The above given were the rest frame frequency and wavelength, the observed frequency can be given as,

$$\nu_{observed} = \frac{1420.4}{(1+z)} \text{ MHz}$$

and the observed wavelength can be given as,

$$\lambda_{observed} = 21.11(1+z) \text{ cm}$$

By observing the 21-cm line during the first billion years after Big Bang which now falls in the frequency range 30-200 MHz, we can get information about the first stars, galaxies, and quasars from the ionizing bubbles around them, also information the properties of the intergalactic medium.

1.3 21-cm Signal

1.3.1 21-cm Brightness Temperature

The intensity of the 21-cm line as seen from Earth is measured by the 21-cm brightness temperature. The reason is, that we are dealing with very low frequencies hence we are in the Rayleigh-Jeans regime. This enables us to quantify the specific intensity I_ν in terms of brightness temperature T_b using the Rayleign-Jeans approximation. We have CMBR coming through the neutral hydrogen, having optical depth $\tau << 1$ implying that the medium optically thin. Let T_γ be the temperature associated with it, then using the radiative transfer equations, treating CMBR as background and spin temperature T_S associated with neutral hydrogen as source, and using Rayleigh-Jeans approximation, one can write the excess brightness temperature relative to CMBR which redshifted to the observer, we have,

$$\delta T_b(\hat{n}, z) = \frac{(T_s - T_\gamma)\tau}{1+z} \tag{1.1}$$

Taking the effects of underlying hydrogen density distribution and redshift space distortion into account for calculation of the optical depth, the above expression reduces to Equation 1.2 given in [1].

$$\delta T_b(\hat{n}, z) = 27\, x_{\rm HI}\, (1+\delta_b) \left(\frac{\Omega_b h^2}{0.023}\right) \left(\frac{0.15}{\Omega_m h^2}\right)^{1/2} \left(\frac{1+z}{10}\right)^{1/2} \left(\frac{T_s - T_\gamma}{T_s}\right) \left[\frac{\partial_r v_r}{(1+z)H(z)}\right] \text{mK} \tag{1.2}$$

where, x_{HI} denotes the neutral hydrogen fraction, and δ_b denotes baryonic overdensity both these parameters vary with redshift and line of sight. Then we have Ω_b and Ω_m giving the proportion of baryons and matter density of the Universe. The h denotes the Hubble parameter. The term at the end denotes the effect of peculiar velocities along the line of sight and cosmological expansion causing the redshift space distortion.

However, our simulation deals with recent redshifts ranging from 6 to 13, where the $T_\gamma << T_S$, enabling us to neglect the effects of CMBR temperature and we don't take redshift space distortion (RSD) into account to keep things a bit simple, hence the expression reduces to 1.3,

$$\delta T_b(\hat{n}, z) = 27\, x_{\rm HI}\, (1+\delta_b) \left(\frac{\Omega_b h^2}{0.023}\right) \left(\frac{0.15}{\Omega_m h^2}\right)^{1/2} \left(\frac{1+z}{10}\right)^{1/2} \text{mK} \tag{1.3}$$

1.3.2 Modeling the 21-cm Signal

In this work, we are modeling the 21-cm signal using a semi-numerical simulation called `ReionYuga` [†] [17, 18, 19]. It is based on the excursion set formalism [20] and involves post-processing the N-body outputs [21, 19] to get the neutral fraction field. The simulation takes

[†] https://github.com/rajeshmondal18/ReionYuga

the dark matter density and halo field along with three EoR parameters, namely minimum halo mass ($M_{h,min}$), number of ionizing photons produced per baryon (N_{ion}) and mean free path of the ionizing photons (R_{mfp}), calculates the hydrogen and photon density field from it then uses excursion set formalism to check whether a grid cell is ionized or not.

1.4 Aim of the Thesis

The on goning radio experiments, such as LOFAR, HERA, MWA, uGMRT and upcoming experiments such as SKA will particularly focus on the detection of the cosmological 21-cm signal from the era of the CD-EoR. We need forward models of this signal to interpret the observations made by these telescopes. The reionization simulations that are generally used for this task are computationally expensive when it comes to exploring the large, complex EoR parameter space while doing Bayesian inference.

A go-to approach is to emulate the Fourier statistic (power spectrum, bispectrum, etc.) [22, 23, 24] derived from the fields obtained via simulations and compare it with the Fourier statistic of the observed field. Since the 21-cm signal is highly non-Gaussian, the power spectrum alone cannot provide all the information, you will need an N-point correlation function to capture this non-Gaussianity. Hence, instead of relying on the Fourier statistics and compressing the information, we can directly use the fields produced by the simulation itself for comparing them with observed fields and hence preserving all the information; this is known as field-level inference. Since these simulations are computationally expensive, we need to develop an emulator which will be trained on a few thousand input and output pairs corresponding to a given set of reinoization parameters and later can be used as a forward model for field-level inference.

There are two prominent models available in the literature that take gas density and source field as inputs and give neutral fraction fields or their proxies, such as time of reionization as output. These approaches align with our approach of using dark matter and halo density fields as inputs to predict the neutral fraction fields. These models are CRADLE [25] and PINION [26]. The main drawbacks of these models are, they have fixed the astrophysical parameters and varied the input fields by changing the redshift (PINION) or by changing the initial random seed (CRADLE). However, for doing the field-level inference, we need variation in the astrophysical parameters. Moreover, these models didn't take all scales into account due to their biases. This led to the underprediction of large (PINION) and small-scale (CRADLE) features depending upon the model architecture.

To address these problems, we are developing an emulator using Vision Transformers, which will give us neutral fraction fields quickly and accurately. The model is designed to take the same two 3D inputs of dark density and halo field, and 3 EoR parameters, same as the simulation discussed in the previous section, and it predicts the neutral fraction field corresponding to those

parameters but at a much faster rate, making it suitable for exploration of parameter space. As discussed earlier, we are using Equation 1.3 for 21-cm signal, hence we only need the baryonic overdensity and neutral hydrogen fraction at a given redshift, because the rest of the quantities are constants. With the assumption that the baryonic overdensity follows the underlying dark matter density, here we can use the dark matter density itself. The only other variable, i.e., the neutral fraction field, will be predicted by the emulator, enabling us to calculate the 21-cm brightness temperature using Equation 1.3.

This emulator will help us bypass the simulations while doing the field-level inference [27] and will enable us to explore the reionization parameter space more efficiently. Furthermore, this emulator has potential applications in forward modeling for upcoming radio interferometers, such as the Square Kilometer Array (SKA) [28]. Since SKA is going to produce tomographic maps, we can use this emulator for field-level inference.

CHAPTER 2

Simulating 21-cm Maps of EoR

The training dataset for our emulator was generated through a multi-step simulation process. We first ran an N-body simulation [21] to model the underlying dark matter distribution. We then applied a Friends-of-Friends (FoF) halo-finding algorithm [19] to identify collapsed structures called halos. Finally, we used the semi-numerical simulation, `ReionYuga`* [19, 17, 18] which follows the excursion set formalism [20]. The code simulation produces x_{HI} maps using different combinations of the three EoR parameters, which serve as the primary data for training the emulator. A detailed discussion of each of these simulations and the three EoR parameters is given in the following sections:

2.1 N-Body Dark Matter Simulation

We use Particle-Mesh (PM) N-body simulation to generate the dark matter density fields at a given redshift. PM methods approximate gravitational interactions more efficiently by representing gravitational forces on a fixed grid or mesh. In this method, the grids are populated with dark matter particles, and the gravitational potential is solved for this density grid in Fourier space. The mass resolution is based on the number of particles and simulation volume. The power spectrum obtained from the transfer function given in [29] is normalized with σ_8 and used for generating the initial density perturbations. Then the memory is allocated to save the particle positions and velocities along with $\rho(r)$ and Δk representing initial density and Fourier transform of initial density fluctuations respectively. The Δk is generated on the grid, and initial positions and velocities are determined using the Zeldovich approximation. The velocities and positions were then updated using the leapfrog integration method causing the density to evolve. In the end, we get the snapshot of the distribution of dark matter particles and their velocities at the given redshift as shown in the Figure 2.1 (a).

We ran the simulation for 3072^3 grids at redshift $z = 7$ with grid resolution of 0.07 cMpc giving us a simulation volume of $(215.04\,\text{cMpc})^3$. We filled the set of grids with 1536^3 dark matter particles giving us particle-mass resolution of $\approx 10^8\,\text{M}_\odot$. The cosmological parameters

*https://github.com/rajeshmondal18/ReionYuga

used for this simulation are as $\Omega_m = 0.3183$, $\Omega_\Lambda = 0.6817$, $\Omega_b = 0.049021423$, $h = 0.6704$, $\sigma_8 = 0.8347$, and $n_s = 0.9619$ consistent with Plank+WP [30]. Here onwards we are dropping the cMpc notation for Mpc.

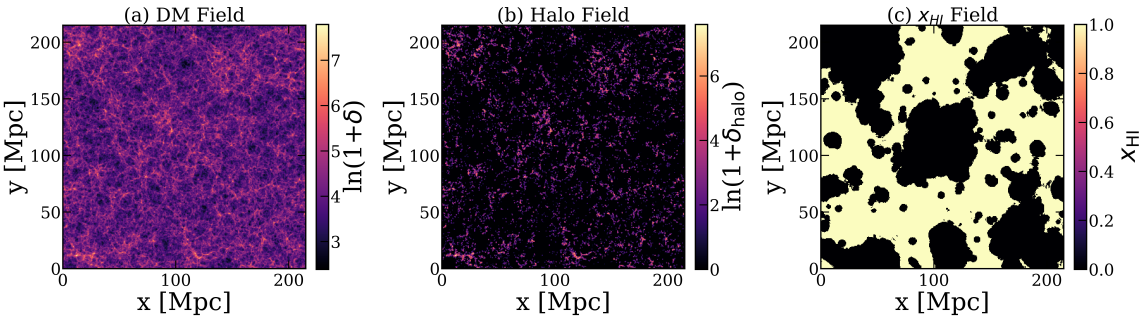


Figure 2.1: Plots of different fields obtained from the simulations: dark matter density, halo field, and neutral hydrogen fraction $x_{\rm HI}$. The grid resolution is reduced from 3072^3 to 384^3, making the grid separation 0.56 Mpc, while generating the maps from the simulation outputs.

2.2 Halo Identification using Friends-of-Friends (FoF)

A dark matter halo is a region of gravitationally bound dark matter assumed to host galaxies and galaxy clusters. The Friends-of-Friends algorithm is used to identify the collapsed halos using the output of the N-body simulation. This is done by using a parameter called linking length (l). For our simulation the value of linking length is 0.2 times the mean inter-particle separation in the simulation. Any two particles having separation less than l are linked together to form a group. Continuing this, a single particle is directly linked to all its neighboring particles within the distance l, and indirectly linked to other particles which are linked to its neighbors, hence it is called the friends-of-friends algorithm. If a particle doesn't have any other particles within the radius l it is considered as part of its own group. All such small groups that lie below the threshold for forming the halo are discarded. In our simulation, the threshold is 10 particles in a group for considering it as a halo. Hence the resulting halo-mass resolution is $\sim 10^9\,{\rm M}_\odot$. The position of the center-of-mass of these halos and their velocities are saved as a halo catalog. The Figure 2.1 (b) shows the halo field for redshift 7.

2.3 Semi-numerical Simulation of 21-cm Signal: ReionYuga

We use a semi-numerical simulation called `ReionYuga` for generating the neutral hydrogen maps. It is based on the excursion set formalism [20]. Since we are using the dark matter only N-body simulation, the density of baryons is not modeled hence the key assumption that goes into this semi-numerical simulation is that the baryonic overdensity perfectly follows the dark

matter overdensity. Additionally, we assume that the ionizing radiation sources are localized within the halos.

2.3.1 Reionization Parameter in Simulation

The following three EoR parameters are varied in the simulation leading to the different reinoization histories which will be the training data for our emulator.

2.3.1.1 Minimum Halo Mass ($\mathbf{M}_{h,min}$)

The minimum halo mass is the mass a halo should have for the gas to cool sufficiently and support star formation. Halos with their masses below this value do not support star formation and hence do not contribute the ionizing radiations. Setting this value lower results in more halos contributing to the reionzation process leading to faster reionization. It is varied within the range $10 \times 10^8\,\mathrm{M}_\odot$ to $800 \times 10^8\,\mathrm{M}_\odot$ and used in the units of 10^8 while feeding it to the network.

2.3.1.2 Ionizing Photon Emitting Efficiency ($\mathbf{N}_{ion}$)

The number of ionizing photons (N_γ) emitted by a source is proportional to the mass of the halo (M_h) hosting that source. Therefore,

$$N_\gamma(M_h \geq M_{h,min}) = N_{ion}\frac{\Omega_b M_h}{\Omega_m m_p} \tag{2.1}$$

where N_{ion} is a dimensionless constant representing the ionizing photon emitting efficiency. It combines several key factors influencing reionization: the number of ionizing photons produced per baryon in stars, the fraction of UV photons that escape into the IGM, and the star formation efficiency.The larger value of this parameter gives larger ionized regions. It is varied within the range 10 to 200.

2.3.1.3 Maximal Distance Traveled by the Ionizing Photons ($\mathbf{R}_{mfp}$)

The mean free path of ionizing photons is the maximum distance traveled by the photons before they get absorbed by the dense pockets of hydrogen gas. It limits the size of the ionizing bubble. However, a larger mean free path doesn't ensure larger ionized regions if the N_{ion} value is smaller. This parameter is varied within the range 1.12 Mpc to 40.32 Mpc. In the simulation, it is used as the maximum smoothing radius for the hydrogen density field, which is derived from the dark matter density field, and for the ionizing photon density field, which is obtained from the halo distribution using Equation 2.1.

We obtained 7,203 data points by varying combinations of these three parameters, 80% of which were used for training and 20% were used for validation.

2.3.2 Excursion Set Formalism

We implement the excursion set formalism by smoothing both the hydrogen density field and the ionizing photon density field using a spherical top-hat window function with radius R. The minimum value of R, denoted $R_{\rm min}$, corresponds to the grid resolution, while the maximum value, $R_{\rm max}$, is set by the mean free path of ionizing photons, which is varied in the simulation (i.e., $R_{\rm max} = R_{\rm mfp}$). For each grid cell, we begin with the smallest smoothing scale $R_{\rm min}$. At each step, we compute the average number density of ionizing photons, $\langle n_\gamma \rangle$, and compare it to the average hydrogen number density, $\langle n_H \rangle$, within a sphere of radius R centered on the grid cell. If the condition $\langle n_\gamma \rangle \geq \langle n_H \rangle$ is satisfied, the cell is marked as ionized. If not, the radius is incremented by a small step ΔR, and the check is repeated. This process continues until the maximum radius $R_{\rm max}$ is reached. Suppose the ionization criterion is still not met at any smoothing scale. In that case, the cell is considered partially ionized, and its ionization fraction is set as $x_{\rm HII} = n_\gamma / n_H$. Repeating this procedure for all cells in the grid yields the final ionization map, which can be converted into a neutral hydrogen fraction field, as illustrated in Figure 2.1(c).

The ReionYuga code takes around 5-8 minutes for a single realization with 384^3 resolution, hence rerunning it for a large number of times for different combinations of EoR parameters while doing the Bayesian inference would be a computationally expensive process. Hence, we aim to develop a neural network based emulator for quick generation of these fields.

CHAPTER 3

Emulation Efforts of 21-cm Maps with Neural Networks

There are two prominent model architectures in the literature that make use of gas density and source field as inputs, which aligns with our approach of using dark matter density and halo field as inputs. These two models are given as PINION (Physics Informed Neural networks for reIONization) by D.Korber et. al (2023) [26] and as CRADLE (Cosmic Reionization and Deep LEarning) by J. Chardin et. al. (2019) [25]. Both of these approaches use convolutional layers at their core for emulation.

3.1 PINION

The PINION model architecture uses the traditional convolutional neural network [31], which is made up of convolutional layers for extracting feature maps, followed by fully connected layers that takes the flattened features from the last convolution layers, as shown in the Figure 3.2. Here they have used gas density (Figure 3.1a) and source fields (Figure 3.1b) corresponding to different redshifts and mapped them to the ionization fraction (Figure 3.1d) fields obtained from C^2-Ray simulation [32]. The input fields exhibit minimal variation with redshift (Figure 3.1a, 3.1b). In contrast, the fields representing the evolution of the photoionization rate (Figure 3.1c) and the ionized fraction (Figure 3.1d) show significant changes as redshift changes. To take this change into account, the source field is smoothed with the radius equal to the mean free path of ionizing photons calculated by using an analytical formula [8] given by equation (3.1)

$$\lambda_{\nu\mathrm{HI}}(z) \approx \frac{c}{H(z)} \times 0.1 \left(\frac{1+z}{4}\right)^{-2.55} \tag{3.1}$$

Then this smoothed source field is fed along with the gas density and source field to the model as shown in the Figure 3.2. Instead of using the full gas density and source field data cubes as input, the approach involves extracting smaller subcubes of size 7^3. Each subcube corresponds to a physical volume of $(16.67\,\mathrm{Mpc})^3$. The feature maps from these subcubes are flattened, and

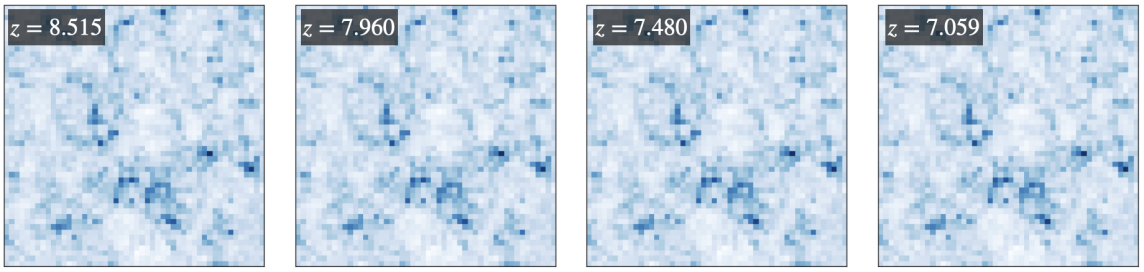
the redshift is provided as an additional input. The model then predicts the ionization fraction at the center of each corresponding subcube. Moreover, this process is repeated to reconstruct the entire 3D ionization fraction cube. The pros of this architecture are its flexibility and scalability; no matter how big your simulation box is you can predict the corresponding ionization fraction field using this model. It uses physics informed approach in which the output of the model is compared to the true data to calculate the data loss $\mathcal{L}_{data}$ and the physics constraint to obtain the ODE loss $\mathcal{L}_{ODE}$ during the training process to adjust the model weights. The model is trained under three different scenarios to evaluate whether incorporating physics-informed inputs improves its performance. Additionally, these scenarios help assess the model's ability to make accurate predictions when trained on a limited amount of data. Based on this, the three scenarios are following: (i) NP (No Physics): only data loss is considered for training the model, (ii) PFD (Physics and Full Data): the training uses $\mathcal{L}_{total} = \mathcal{L}_{data} + \mathcal{L}_{ODE}$ with 46 redshift snapshots, and (iii) LD (Low Data): this also uses total loss but only uses 5 redshift snapshots for training.

By virtue of its model architecture, it is able to predict the small scale variations over all redshifts for all scenarios with great accuracy, as shown in the Figure 3.3. However, since the entire field is not fed to the architecture at the same time, it doesn't learn any large-scale dependencies and hence underpredicts the large-scale variations, particularly at lower redshifts as shown in the Figure 3.3 . Moreover, the approximations done to obtain equation (3.1) comes from the poorly constrained observation of quasar spectra for $2 < z < 4.5$ [8] hence using this formula for smoothing fields obtained from redshift $6 < z < 12$ may introduce unwanted features in the output. This model uses fixed astrophysical parameters, hence it cannot be used for field-level inference.

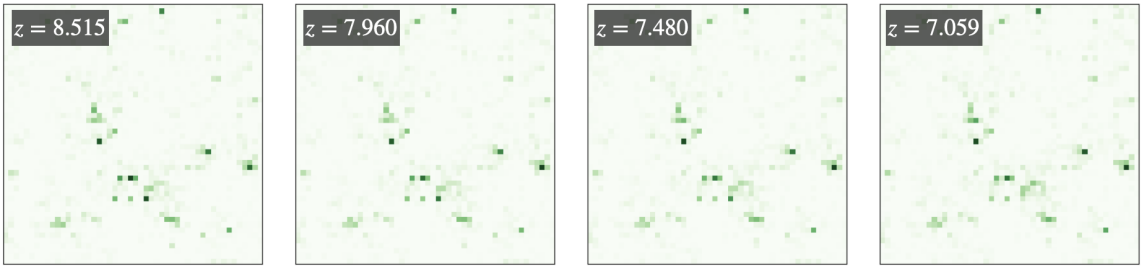
3.2 CRADLE

The CRADLE follows autoencoder-style convolutional neural network architecture for mapping different realizations of gas density and source fields to the field that gives the time of reionization (t_{reion}) (Figure 3.4). These 3D input and output fields are obtained using EMMA [33]. The architecture consists of 2 major parts: the encoder and the decoder. In the encoder arm the input fields are progressively downsampled while extracting the feature maps using convolutional and max pooling layers and then these features are combined and upsampled using transpose convolution in the decoder arm (Figure 3.5).

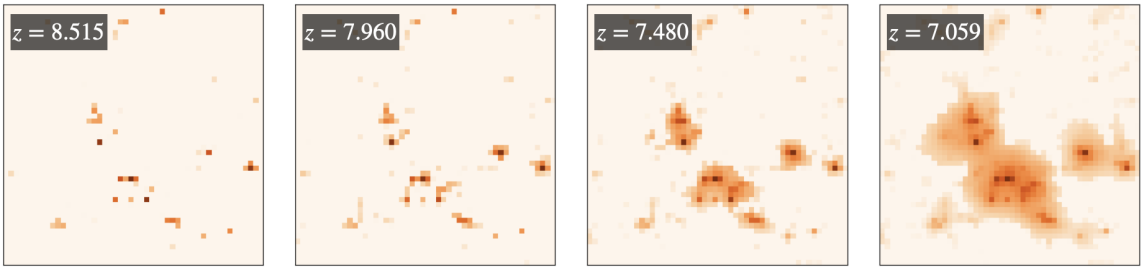
To reduce the computational cost, instead of using the entire 3D fields as inputs and predicting the corresponding output, they sliced these fields into 2D maps. They fed them independently to the model to predict the corresponding 2D output slices of the field. Then these slices are rejoined to form the 3D output field showing the time of reionization. However, in this approach, the influence of sources across different slices wouldn't be considered, and hence



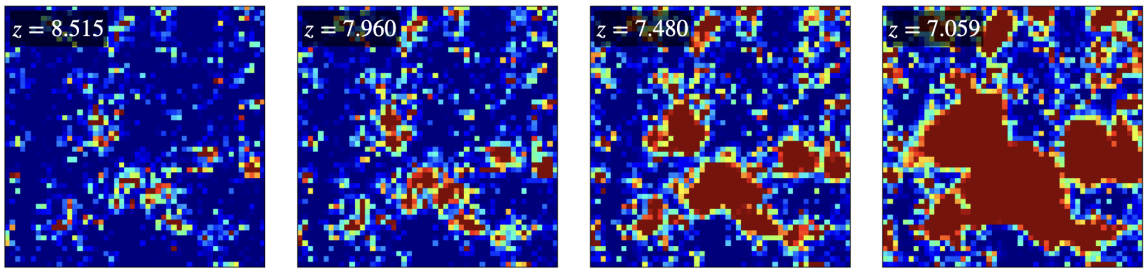
(a) Evolution of the gas density: light blue are under-dense regions and dark blue are over-dense regions.



(b) Evolution of the mass of sources: light green are regions with few particles, and dark green are the sources.



(c) Evolution of the photoionization rate: light orange are regions which receive less ionizing photons and dark orange are the ionizing sources.



(d) Evolution of the ionized fraction: red are ionized regions and dark blue are neutral regions.

Figure 3.1: The plots compare the evolution of different fields with redshift. (Figure Credits: D. Korber et al. [26])

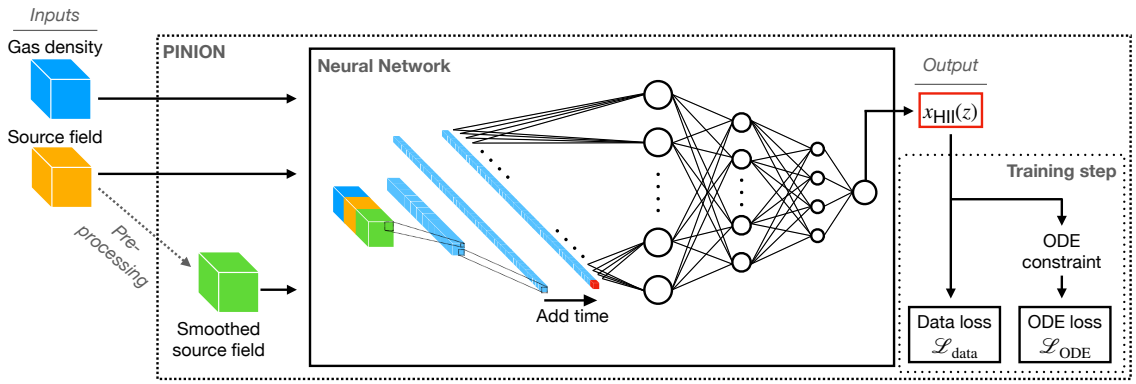


Figure 3.2: Architecture of the PINION Model. (Figure Credits: D. Korber et al. [26])

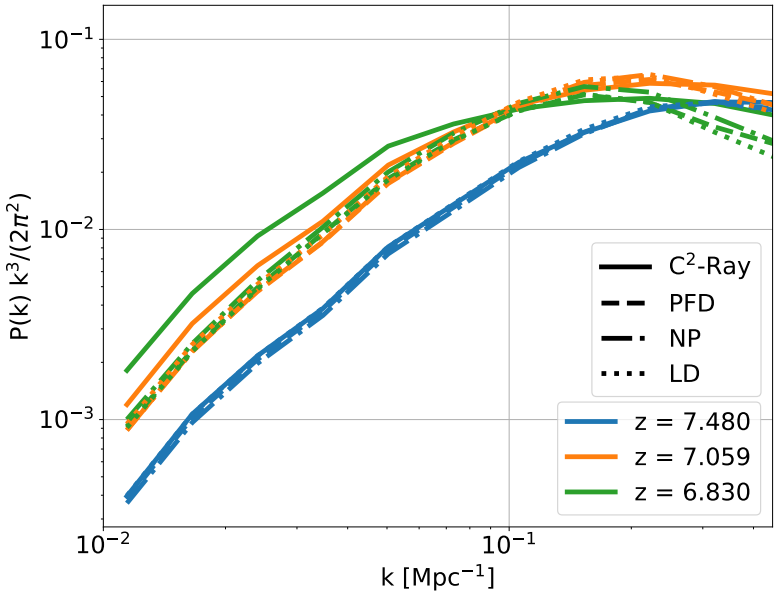


Figure 3.3: Dimensionless power spectrum of xHI. The solid lines correspond to C^2-Ray and the other lines correspond to different methods used for emulation. (Figure Credits: D. Korber et al. [26])

to tackle this, they smoothed the source field and gas density with a Gaussian kernel of size $\sigma = 30$, corresponding to the length scale of 3.75 cMpc/h and fed these smoothed field to the model instead of the original ones. Because of the smoothing, the small-scale features are lost in the input and hence underpredicted in the output, as shown in the power spectrum (Figure 3.6). The underprediction arises due to the smoothing of inputs, which prevents the model from accurately identifying the locations where reionization begins. Moreover, this model also uses fixed astrophysical parameters, making it unsuitable for the field-level inference.

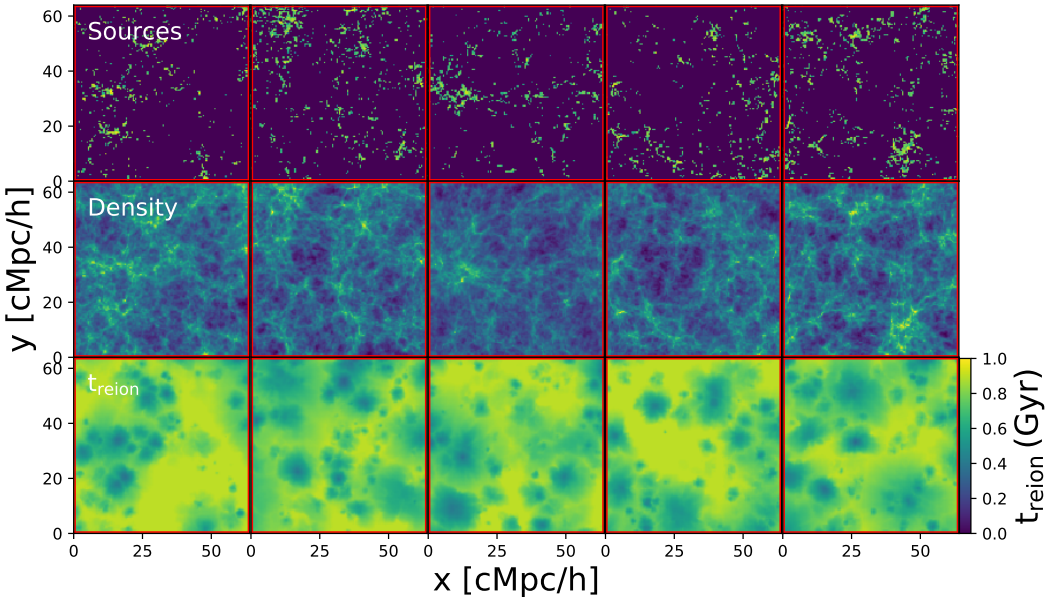


Figure 3.4: Examples of fields used for training the neural network. (Figure Credits: J. Chardin et al. [25])

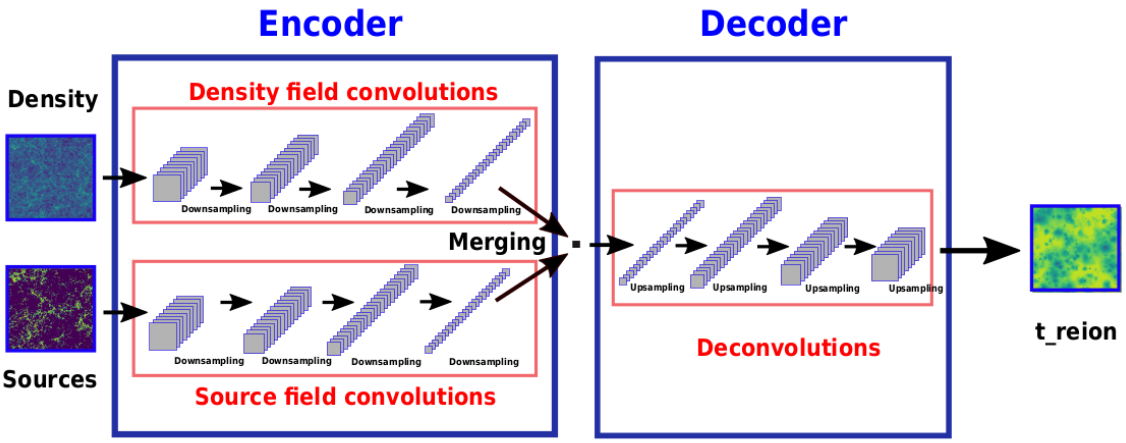


Figure 3.5: Architecture of the CRADLE Model. (Figure Credits: J. Chardin et al. [25])

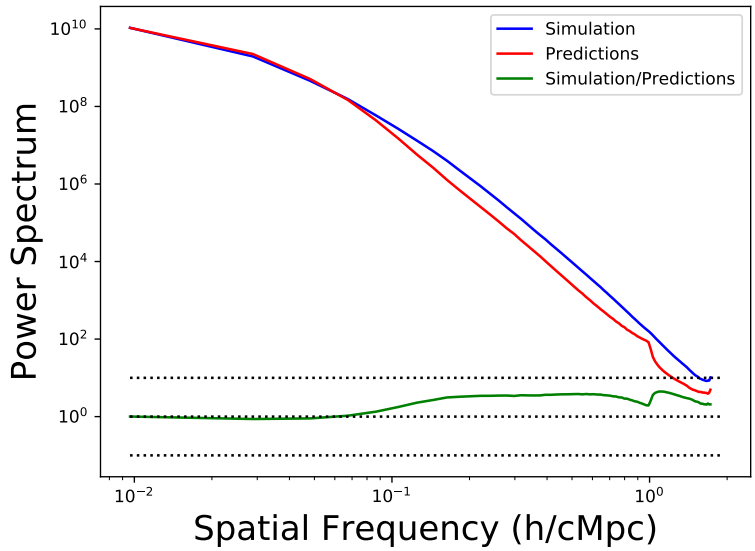


Figure 3.6: Power spectrum of the t_{reion} fields. (Figure Credits: J. Chardin et al. [25])

To address the drawbacks of the above-mentioned approaches, we need an alternative model architecture that takes the entire 3D cubes of the dark matter density and halo field as inputs and gives the 21-cm brightness temperature field as output. Consequently, this model should also consider both the large and small-scale dependencies.Unlike the previously mentioned approaches that map different input realizations to their corresponding outputs, our method uses the same input fields across multiple outputs. As a result, the outputs are primarily conditioned on the three EoR parameters. Therefore, the model architecture must include encoding layers capable of capturing the sensitivity to variations in these parameters. Since this model will have variation in astrophysical parameters, it is a perfect candidate for field-level inference of the reionization model parameters, given the observed 21-cm brightness temperature field.

CHAPTER 4

EMULATION OF EOR 21-CM MAPS USING COSMOVIT, COSMOUIT AND COSMOUNET

In the earlier chapter, we discussed the previous efforts of emulating 21-cm maps and their drawbacks. The aim of this thesis is to propose an improved emulation formalism for 21-cm maps.

4.1 Fundamentals of Vision Transformers

In this work, we will mainly use Vision Transformers (ViTs). The self-attention mechanism of the ViTs help it to capture long-range dependencies very accurately, which is one of the primary requirements of our emulator. There are several ways to build such a model using building blocks of deep learning algorithms. In this section, we discuss those fundamentals while using a toy field as an example.

4.1.1 Patchifying

The transformers were originally designed for natural language processing (NLP) tasks [34], where the words are vectorized and used for further processing, where the similarities and dependence between each word are calculated as an attention score. When this newly introduced architecture was adapted for computer vision tasks, the pixel values of an image were to be used as vectors for calculating their dependence on each other, but for $n \times n$ image, we get n^2 vectors which would be difficult to process if image size is larger, hence instead of taking individual pixels for calculation we break down the image into smaller patches and flatten them into 1D arrays (Figure 4.1) and treat these arrays as vectors for further calculations. [35]. In our case, we have a 3D input field, which is broken down into small 3D subcubes and flattened.

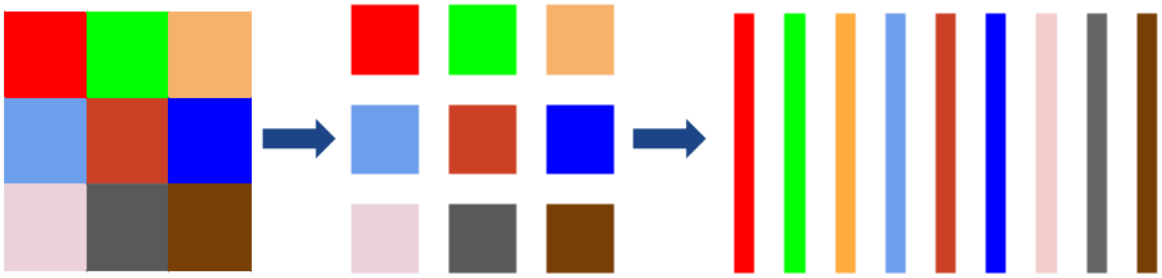


Figure 4.1: Patchifying.

4.1.2 Tokenization

The 1D arrays we get after patchifying the input field are projected into n-dimensional vector space, using the linear projection using feed-forward layers as shown in Figure 4.2. Here n corresponds to the size of each 1D array. We will see why this process is required when we will discuss the calculation of the attention score.

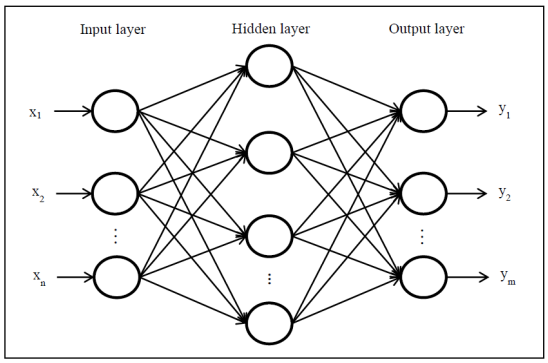


Figure 4.2: Feed-forward layers used for tokenization. (Figure Credits: [36])

4.1.3 Positional Embedding

The transformers are permutation equivariant [37], meaning they don't take the order of tokens explicitly into account while doing the calculations, hence even if you switched the order, the output, i.e., the attention score, remains the same. However, for NLP tasks, the order tokens represent the order of words and hence encapsulate the meaning of the sentence, therefore, it is necessary to take the position of words into account while calculating the attention score. When it comes to computer vision tasks, positional embedding is needed to consider the position of the pixel in a particular patch and the position of the patch in the field. So, this information of the position is added using positional embedding. The positional embedding can be constant, hence given directly while feeding it to the transformer block, or it could be learnable, hence learned during the training process. An example of constant positional embedding is sinusoidal positional embedding [34].

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

where,

- pos: The position in the sequence (from 0 to $l-1$).

- l: The sequence length i.e. number of tokens

- d: The total embedding dimension i.e. total number of elements in an array.

- i: The dimension index (from 0 to $d-1$).

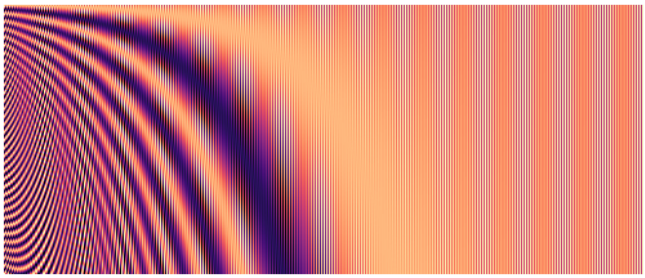


Figure 4.3: Sinusoidal Positional Embedding.

4.1.4 Attention Score

Attention score tells the network how much attention should be given to the input token based on its dependence on the other tokens. This dependence is calculated using a dot product between different tokens which are projected in the n-dimensional vector space. If the vectors align with each other, their normalized dot product would be closer to 1. If they are opposite in this vector space their normalized dot product will be closer to -1, and if they are perpendicular to each other, showing that they are not related the dot product is 0. Figure 4.4 shows a simplified example of how the tokens would look if their embedding dimension were 3.

To calculate the attention score, 3 linear projections of each token are done, called key (**K**), value (**V**), and query (**Q**). In NLP tasks, query represents a vector for which we are seeking information from other vectors, keys are vectors representing potential matches, and value represents information about each key. In simple language, we want to check given a query vector, which key vector is similar to it i.e., aligns with it in the higher dimensional vector space. This alignment is quantified using the dot product between these two vectors, as mentioned earlier. We normalized the dot product using the square root of the embedding dimension (d_k), then we apply the softmax function [38], and multiply it with the value vector which is obtained from the same token as the query vector used for calculation, and this gives the attention score between two vectors for the vector represent using value. Equation 4.1 gives the expression for the attention score.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{4.1}$$

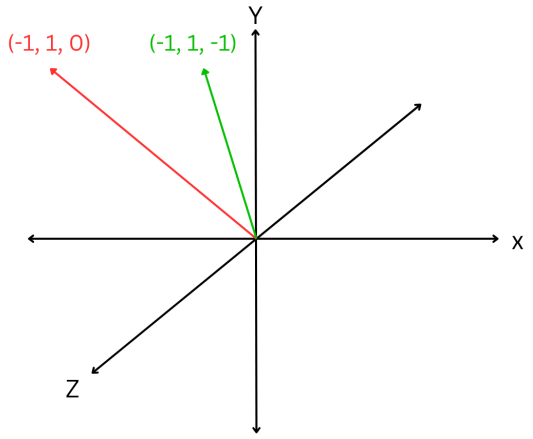


Figure 4.4: A simplified example of attention score calculation using 3D vectors.

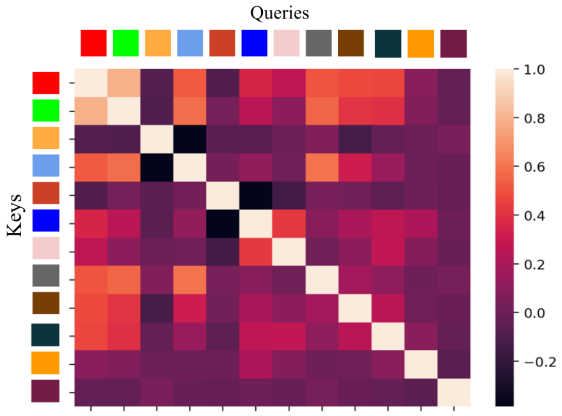


Figure 4.5: Visualization of the dot product of queries and keys.

4.1.5 Convolution Layers

A convolution layer is used for downsampling the input fields. In the convolution process, we have a 3D input field n^3, if a filter of size f^3 traverses over it while doing the convolution operation where the field is averaged out while treating the values of given by filters as weights. The stride decides how much the filter will move further. The padding refers to adding extra cells around the field, this is done to overcome the problem of values in the center being used more times while performing the convolution operation. We can assign as many number of filters as we want, and each filter produces a unique feature map. The formula given in Equation 4.2 gives the size of the output given the input size, filter size, padding, and stride. The values of the filter are learned from the data itself during the training.

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding}}{\text{Stride}} \right\rfloor + 1 \tag{4.2}$$

Vanishing gradients occur when gradients shrink exponentially as they are backpropagated through many layers, especially in very deep networks. This leads to extremely small updates for earlier layers, causing the training to stagnate.

4.1.6 Transpose Convolution Layers

As the name suggests, transpose convolution layers perform the reverse operation of convolution layers. Instead of summarizing a field of input values into a single output value (downsampling), they redistribute a single input value over a larger field to achieve upsampling. For this reason, they are often referred to as deconvolution. These layers take the feature maps produced by convolution layers and upsample them to create new, larger feature maps. The process involves "spreading out" the input values spatially and filling in the gaps, which can be controlled by parameters like kernel size, stride, and padding, which are similar to those of the convolutional layers. Additionally, we have output padding used for making the dimension of the output as desired. The Equation 4.3 gives the size of the output given the parameter values of the transpose convolution.

$$\text{Output Size} = \lfloor (\text{Input Size} - 1) \cdot \text{Stride} - 2 \cdot \text{Input Padding} + \text{Filter Size} + \text{Output Padding} \rfloor \tag{4.3}$$

4.1.7 Residual Connections

First introduced in [39], residual connections are shortcuts used in neural network architecture where the input skip one or more layers and is directly added to the output. When we have models consisting of multiple layers, also known as deep networks, the gradients decay exponentially as they are backpropagated through these many layers. This leads to extremely small updates in the weights during the training process, hence it is slowed; this is called the gradient vanishing problem. The residual connections provide a direct path for the gradient while skipping layers, which leads to significant updates in the weights.

4.1.8 Performance Metrics

There are two key metrics used for quantifying the performance of the models one is mean square error (MSE), other is coefficient of determination (R^2).

4.1.8.1 Mean Square Error (MSE)

It is the mean of the squared difference between true output y_i and predicted output $\hat{y}_i$ (See Equation 4.4).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.4}$$

If we have an image of size m × m, then the i ranges from 1 to n = m × m. We use this MSE for loss calculation which propagates through the neural network for optimizing the weights. Its differentiability makes it an ideal candidate for gradient calculations, but it is sensitive to

outliers and scale-dependent, requiring the outputs that vary over long ranges to be normalized during the training process.

4.1.8.2 Coefficient of Determination (R^2 Score)

It quantifies how well the model captures the variability in the data. R^2 closer to 1 indicates the model explains the variability very well. If it's closer to zero, then the model predicts no better than just predicting the mean of true outputs. If it is negative, then model performance is worse. If we divide and multiply the negative part of the Equation 4.5 by n, the numerator represents the mean of the squared residuals, aka mean square error, and the denominator represents the variance of the true output.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.5}$$

where,

- y_i: True output values
- $\hat{y}_i$: Predicted output values
- $\bar{y}$: Mean of the true values

4.1.9 Optimizers

The size of the training data we are using is larger than the memory of the GPU; hence, feeding the entire data at the same time is not possible. Therefore, we feed the data to the network in batches and calculate and propagate gradients for each batch, this is called mini-batch gradient descent. The problem with this approach is the gradient is calculated considering a small portion of the entire data hence the local behavior of the gradient is not incorporated in such a scenario, this leads to spurious oscillations in the loss function as we move from one batch to another, indicating that the weights are converged to the global minima following a zig-zag path in optimization surface, hence it takes a large number of epochs for the model to reach the global minima to mitigate this we use optimizers [40]. Here we are using Adam optimizer, which has combined properties of Momentum (Stochastic Gradient Descent with Momentum) and RMSprop (Root Mean Squared Propagation) optimizer [41, 42].

4.1.9.1 Gradient-Descent

Given a loss function $L(\mathbf{w}, \mathbf{b})$, where $\mathbf{w}$ are the weights and $\mathbf{b}$ are the biases, with η being the learning rate, the rules for updating the weights and biases through simple gradient descent are given as follows,

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b}) \tag{4.6}$$

$$\mathbf{b} \leftarrow \mathbf{b} - \eta \nabla_{\mathbf{b}} L(\mathbf{w}, \mathbf{b}) \tag{4.7}$$

4.1.9.2 Momentum

Using the momentum optimizer, instead of adjusting the weights based on the gradient of the loss function like we do for the simple gradient descent, we take the moving average of the gradients (Equations 4.8, 4.10) and use that for updating the weights (Equations 4.9, 4.11).

$$\mathbf{v}_{new,w} = \beta \mathbf{v}_{prev,w} + (1 - \beta) \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b}) \tag{4.8}$$

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \mathbf{v}_{new,w} \tag{4.9}$$

Similarly for biases,

$$\mathbf{v}_{new,b} = \beta \mathbf{v}_{prev,b} + (1 - \beta) \nabla_{\mathbf{b}} L(\mathbf{w}, \mathbf{b}) \tag{4.10}$$

$$\mathbf{b} \leftarrow \mathbf{b} - \eta \mathbf{v}_{new,b} \tag{4.11}$$

The hyper-parameter β controls the decay rate of the moving average.

4.1.9.3 RMSprop

Using the RMSprop optimizer, we scale the gradients by a running average of their squared values. This running average is calculated element-wise for each parameter (Equation 4.12, 4.14) and helps normalize the magnitude of the gradients (Equation 4.13, 4.15) , mitigating the effects of exploding or vanishing gradients.

$$\mathbf{s}_{new,w} = \beta \mathbf{s}_{prev,w} + (1 - \beta)(\nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b}))^2 \tag{4.12}$$

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{\sqrt{\mathbf{s}_{new,w} + \epsilon}} \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{b}) \tag{4.13}$$

Similarly for biases,

$$\mathbf{s}_{new,b} = \beta \mathbf{s}_{prev,b} + (1 - \beta)(\nabla_{\mathbf{b}} L(\mathbf{w}, \mathbf{b}))^2 \tag{4.14}$$

$$\mathbf{b} \leftarrow \mathbf{b} - \frac{\eta}{\sqrt{\mathbf{s}_{new,b} + \epsilon}} \nabla_{\mathbf{b}} L(\mathbf{w}, \mathbf{b}) \tag{4.15}$$

The hyperparameter β controls the decay rate of the moving average, and ϵ is a small constant added to avoid division by zero.

4.1.9.4 Adam

As said earlier the Adam has combined properties of both Momentum and RMSprop [41]. The expressions for this optimizer for the updation of weights and biases are given as follows

$$\mathbf{w} \leftarrow \mathbf{w} - \frac{\eta}{\sqrt{\mathbf{s}_{new,w} + \epsilon}} \mathbf{v}_{new,w} \tag{4.16}$$

$$\mathbf{b} \leftarrow \mathbf{b} - \frac{\eta}{\sqrt{\mathbf{s}_{new,b} + \epsilon}} \mathbf{v}_{new,b} \tag{4.17}$$

We have explored multiple model architectures using the combination of the above components, out of which the three prominent model architectures are discussed in the next section. We analyzed model performance through MSE, R^2 score, and visual comparison of fields, and made changes in the next model based on it.

4.2 CosmoViT

We adapted this model architecture from [43]. It was originally designed for image segmentation tasks. Since the neutral fraction fields we are trying to emulate have values consisting of 0s and 1s, except for the boundaries of the ionizing bubbles, this model architecture seemed suitable for the emulation task. While the original implementation used TensorFlow and a single 2D image as input, we re-implemented the architecture in PyTorch and extended it to accommodate two 3D inputs, while integrating three EoR parameters along the way. We are calling it CosmoViT, short for Cosmological Vision Transformers.

4.2.1 Model Architecture of CosmoViT

The model architecture for the CosmoViT can be described using the flowchart given in 4.6. We start by breaking down the 3D cube into subcubes called patches, then flattening them into 1D arrays while doing a linear projection called tokenization. Since transformers are permutation equivariant we have to feed the information about the positions of the pixels using positional embedding. We integrate the 3 EoR parameters into these tokens. Then these tokens are processed within the transformer block where the attention score between different tokens is calculated and added to the original tokens. We reshape these tokens back into the 3 patches and upsample them while treating them as feature maps produced by the transformer encoder block. For upsampling, we use transpose convolution and residual connections alternately to map them to output fields. See table 4.1 for the summary of the CosmoViT (base model) architecture.

4.2.2 Reduction in the Resolution of Fields

The high dimensionality of the original data presented a significant computational challenge. It required a memory of over 125GB for initializing the model itself. To manage memory constraints and computational costs, we reduced the resolution from 384^3 to 48^3 making the grid separation 4.48 cMpc from 0.56 cMpc. We use `BlockReduce`* method offered by `scikit-image`†. The `BlockReduce` method has `block_size` as an argument which decides how many blocks are to be considered along each axis for reduction, in this case, the block size was 8 along each axis, meaning 512 points were combined to a single point in the new field. Another argument is `func` for the reduction function, in our case we used `np.mean`, which takes the average of all the elements in the given block.

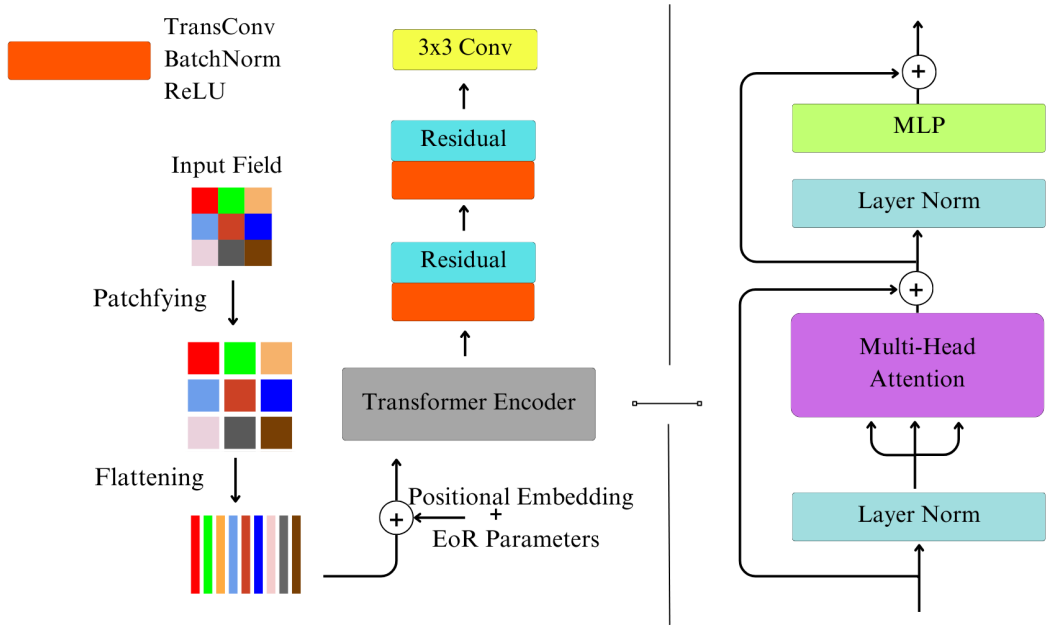


Figure 4.6: Model Architecture of CosmoViT. (Figure Inspiration: Y. Gündüç et al. [43])

4.2.3 Results of CosmoViT

This model was trained for 100 epochs with 48 batch size which took about 13 GPU hours on NVIDIA RTX-A4000-16GB GPU. Figure 4.8 shows the variation in MSE and R^2 score along with the number of epochs for the CosmoViT model. It is evident from the plot that though the loss converges for training and validation data it is still high and the value of R^2 is significantly low even after a large number of epochs. Additionally, the comparison between true and predicted fields (Figure 4.7) shows that the fields predicted by the CosmoViT model all look the same, which implies that the model converged to a generic field to give the relatively

*https://scikit-image.org/docs/stable/api/skimage.measure.html#skimage.measure.block_reduce
†https://scikit-image.org/

Component	No. of Feature Maps	Filter Size	Activation Function
Patch Generation	-	Patch Size: 8	-
Positional Embedding	Num Patches: 216	Flattened Dim: 512	-
Patch Encoder 1	512	-	Linear
Patch Encoder 2	512	-	Linear
Parameter Encoder	512	-	-
Transformer Block 1	4 Attention Heads, FF Dim: 256	-	ReLU
Transformer Block 2	4 Attention Heads, FF Dim: 256	-	ReLU
Transformer Block 3	4 Attention Heads, FF Dim: 256	-	ReLU
Transformer Block 4	4 Attention Heads, FF Dim: 256	-	ReLU
ConvTranspose Layer 1	256	3 × 3	Leaky ReLU
Residual Block 1	256	3 × 3	ReLU
ConvTranspose Layer 2	128	3 × 3	Leaky ReLU
Residual Block 2	128	3 × 3	ReLU
ConvTranspose Layer 3	64	3 × 3	Leaky ReLU
Residual Block 3	64	3 × 3	ReLU
ConvTranspose Layer 4	32	9 × 9	Leaky ReLU
Residual Block 4	32	3 × 3	ReLU
ConvTranspose Layer 5	16	9 × 9	Leaky ReLU
Residual Block 5	16	3 × 3	ReLU
Final Conv Layer	1	3 × 3	ReLU

Table 4.1: Summary of the CosmoViT (base model) architecture.

smaller MSE loss rather than giving an output that is parameter-specific.

The reason behind such model performance could be that the model was designed for image translation tasks, i.e., it would take detailed input images and map them to binarised images, and since the input fields were varying along with the output fields, the model was conditioned over the input fields themselves. However, in our case, the input fields were kept fixed, and the three EoR parameters were integrated in the tokens to introduce the variation, but the model, being only sensitive to input fields, predicted the same output field for all combinations of parameters.

4.3 CosmoUiT

The main flaw of the previous model was that it was producing the same outputs for all combinations of EoR parameters, leading to a significantly high MSE. This problem persisted even after doing the hyperparameter tuning, hence we needed a new model architecture that would produce parameter-specific outputs. In the previous model, we were treating the patches as feature maps and upsampling them using transpose convolution with residual connections, but the feature maps are produced after tracing the entire field with a filter, and they are clearly not just segments of the field, hence, this was considered in the new model architecture. This

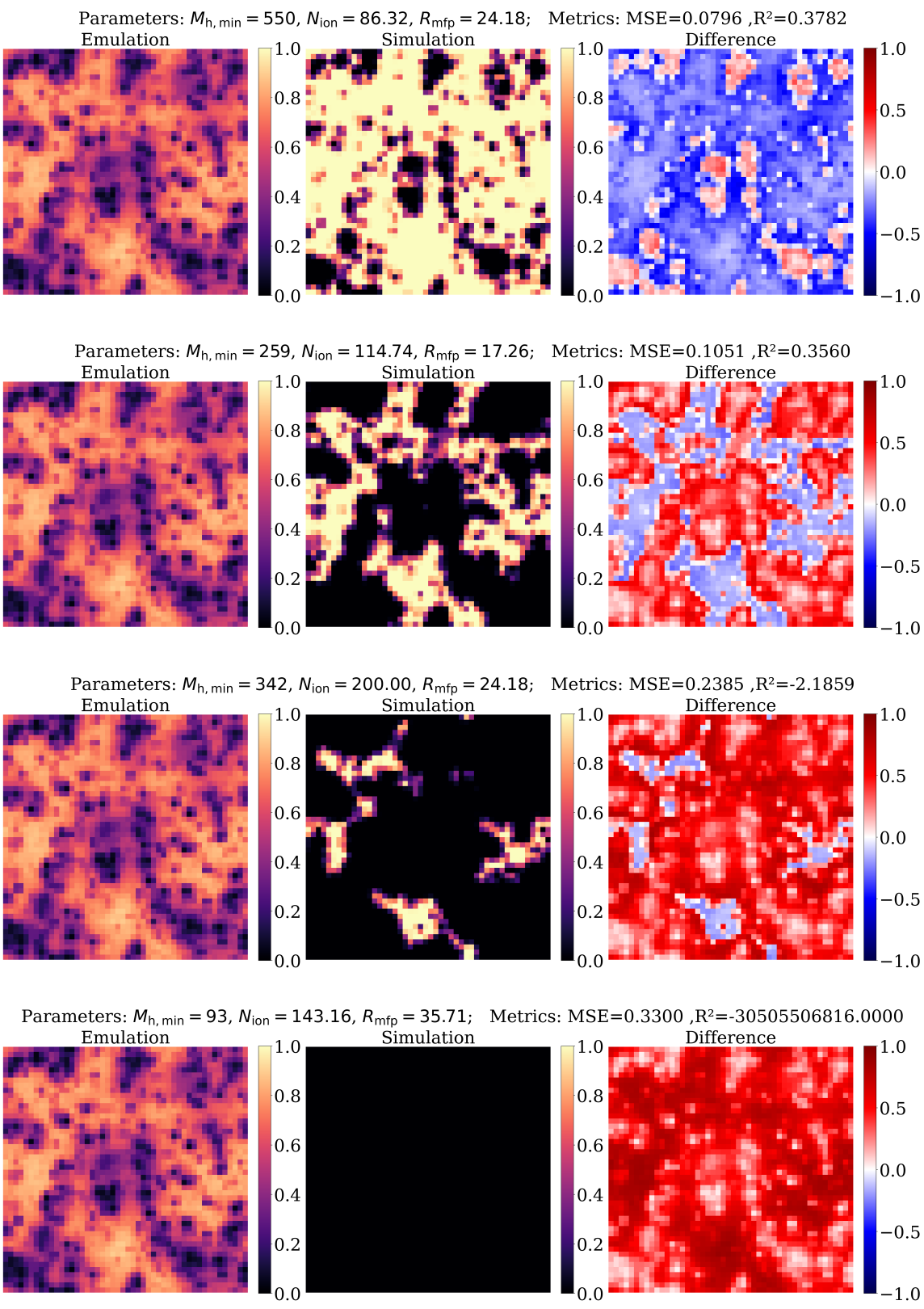


Figure 4.7: *Title:* EoR Parameters and Metric Scores . *First column:* Emulation Output. *Second column:* Simulation Output. *Third column:* Difference.

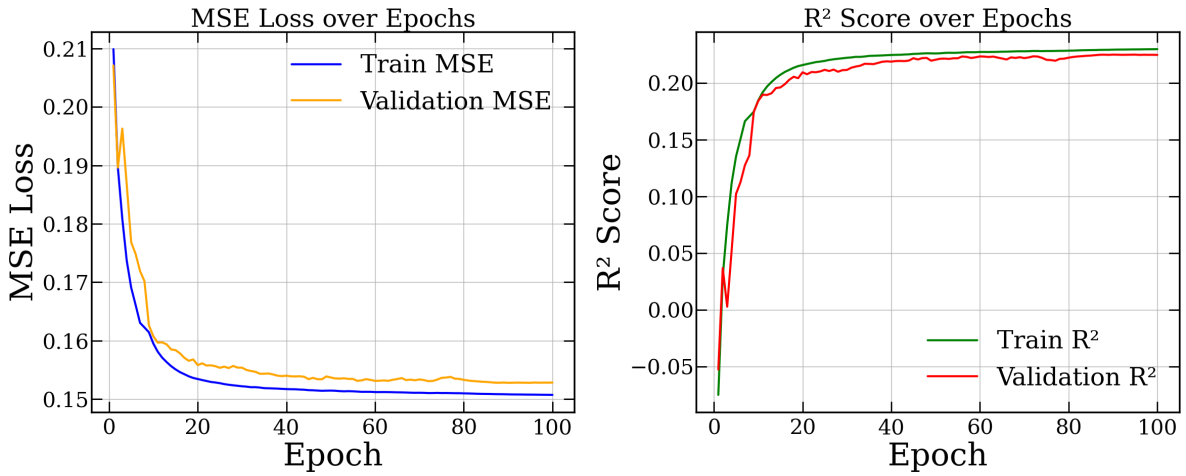


Figure 4.8: The plot shows variations of MSE and R^2 for CosmoViT for training and validation data over epochs

model architecture makes use of the combined architecture of Vision Transformers and UNet, hence, it is called CosmoUiT, short for Cosmological UNet integrated Transformer. This model is expected to show the combined ability of the Vision Transformers to capture the long-range variations and the ability of CNN to capture the small-scale variations, making it perfect for this task.

4.3.1 Model Architecture of CosmoUiT

The model architecture is described in the flowchart given in Figure 4.9. We start by breaking the 3D cube into small subcubes, we flatten them into 1D array and do a linear projection to form tokens the three EoR parameters are then concatenated to these tokens, then they are fed to the transformer encoder block where the attention score between each token is calculated and added to the original tokens. The output we get from the transformer is converted back into the patches and these patches are joined back to form the field. Now this reconstructed field has information about the variations happening at the other parts of the field and also information about the three EoR parameters. We repeated the same procedure for the dark matter density and halo field. These reconstructed fields are then fed to the UNet architecture described in Figure 4.12. The UNet architecture is made up of three parts encoder arm, the decoder arm, and joining them, we have skip connections. The encoder arm progressively downsamples the inputs using convolution and pooling layers, producing feature maps. The lowest stage is called bottleneck, because one cannot extract feature maps below this level. At this level, we concatenate the three EoR parameters to the feature maps. Now, the decoder performs upsampling using transpose convolution and combines these features with corresponding features from the encoder using skip connections, where this combination again goes through the convolution layer to reduce the number of feature maps. The skip connection helps the network for precise localization of output features, hence the spatial information is preserved while the input is mapped to the output. See the model summary given in the table 4.2 for a more detailed description of each

layer.

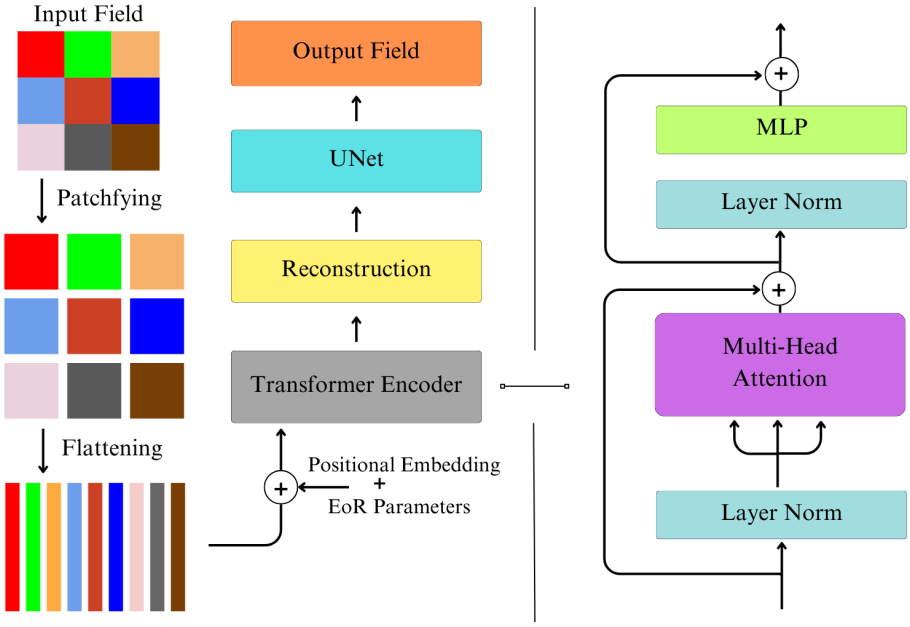


Figure 4.9: Model Architecture of CosmoUiT.

4.3.2 Results of CosmoUiT

We trained this model for 100 epochs for 10 batch size which took about 23 GPU hours on an NVIDIA RTX-A4000-16GB GPU. Figure 4.11 shows the behavior of the MSE loss and R^2 score over the epochs. The validation loss rapidly falls and stays almost constant with slight fluctuations over the next few epochs; on the other hand, the training loss falls steadily. Figure 4.10 shows the comparison between the true and predicted fields, and it is visible that the outputs are parameter-specific, unlike those of CosmoViT's outputs, though the boundaries are a bit fuzzy.

4.4 CosmoUNet

The UNet architecture has been previously used for the image translation task [25], and it is good at capturing the large-scale dependencies through the hierarchical feature extraction process in the encoder arm, while the small-scale dependencies are reintroduced in the decoder arm through skip connections to form the final output. Therefore, the major criticism of the CosmoUiT model is whether the transformer layer provides any additional information at all, or is just the UNet that does all the necessary work.

Component	No. of Feature Maps	Filter Size	Activation Function
Patch Embedding	128	8x8x8	-
Projection Layer	128	-	-
Position Embedding	128	-	-
Parameter Embedding	128	-	-
Fully Connected Layer	128	-	-
Transformer Encoder Layers	128	-	ReLU
- Self-Attention	8 Heads	-	-
- Feedforward	256 Units	-	ReLU
- Layer Normalization	128	-	-
UNet3D Encoder			
DoubleConv Layer (enc1)	32 Filters	3x3x3	ReLU
DoubleConv Layer (enc2)	64 Filters	3x3x3	ReLU
DoubleConv Layer (enc3)	128 Filters	3x3x3	ReLU
DoubleConv Layer (enc4)	256 Filters	3x3x3	ReLU
MaxPool3D (Pooling Layer)	-	2x2x2	-
Bottleneck			
DoubleConv (with Parameters)	512 Filters	3x3x3	ReLU
UNet3D Decoder			
ConvTranspose3D (upconv4)	256 Filters	2x2x2	-
DoubleConv (dec4)	256 Filters	3x3x3	ReLU
ConvTranspose3D (upconv3)	128 Filters	2x2x2	-
DoubleConv (dec3)	128 Filters	3x3x3	ReLU
ConvTranspose3D (upconv2)	64 Filters	2x2x2	-
DoubleConv (dec2)	64 Filters	3x3x3	ReLU
ConvTranspose3D (upconv1)	32 Filters	2x2x2	-
DoubleConv (dec1)	32 Filters	3x3x3	ReLU
Final Conv Layer	1 Filter	3x3x3	-

Table 4.2: Summary of the CosmoUiT (base model) architecture.

4.4.1 Model Architecture of CosmoUNet

To find significant evidence for the above claim, we removed the transformer encoder block from the base model architecture of CosmoUiT and used only the UNet part, calling it CosmoUNet to stay consistent with the naming scheme of models. Now, the dark matter density and halo field are directly fed to the architecture through the encoder arm, and the output is obtained through the decoder arm. The architecture remains same as shown in Figure 4.12 with model summary given in the table 4.3.

The self-attention mechanism of the transformer makes the CosmoUiT to learn the dependence of different patches on each other and also on the three EoR parameters, hence it is able to produce parameter-specific outputs with such good accuracy.

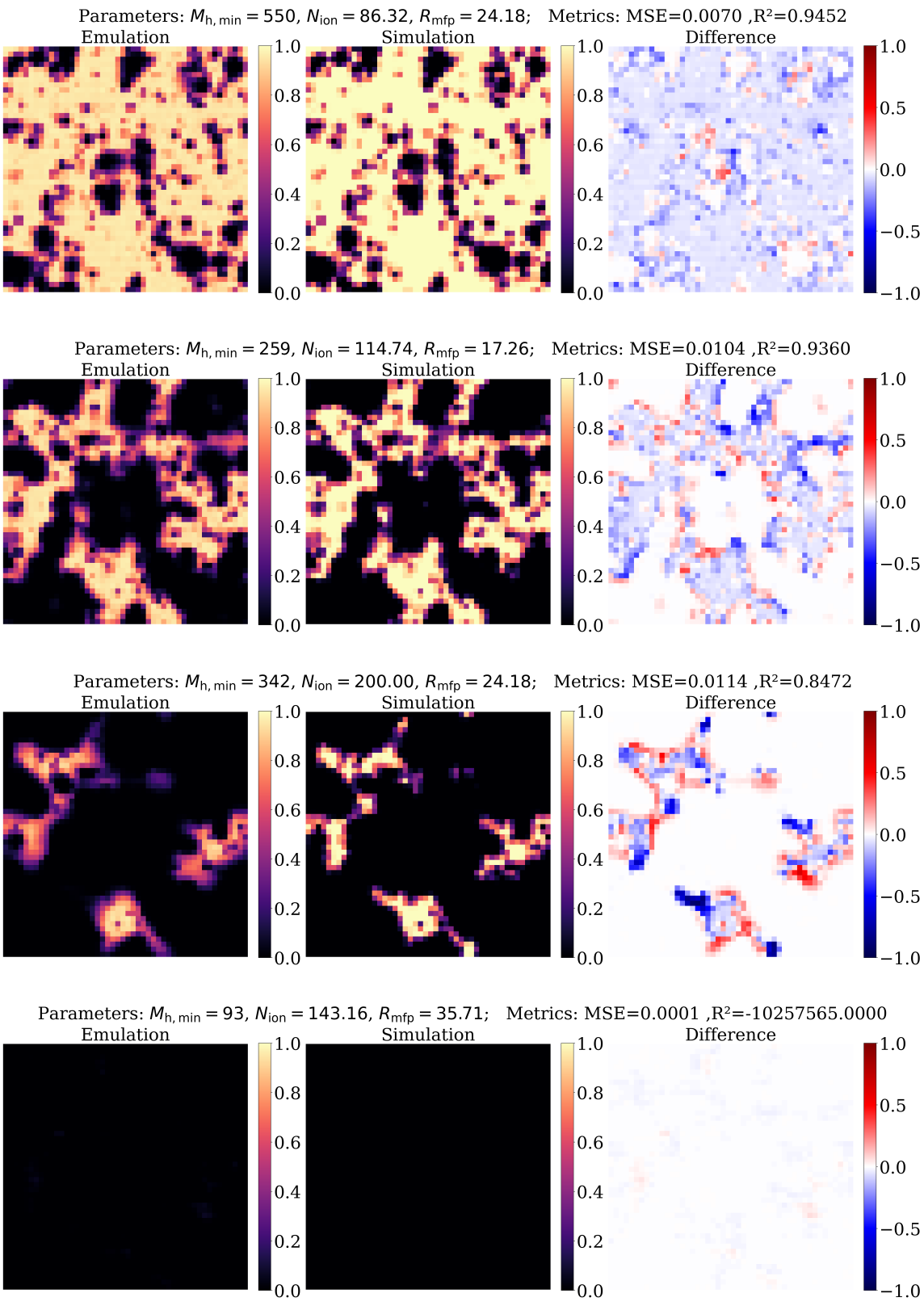


Figure 4.10: *Title:* EoR Parameters and Metric Scores . *First column:* Emulation Output. *Second column:* Simulation Output. *Third column:* Difference.

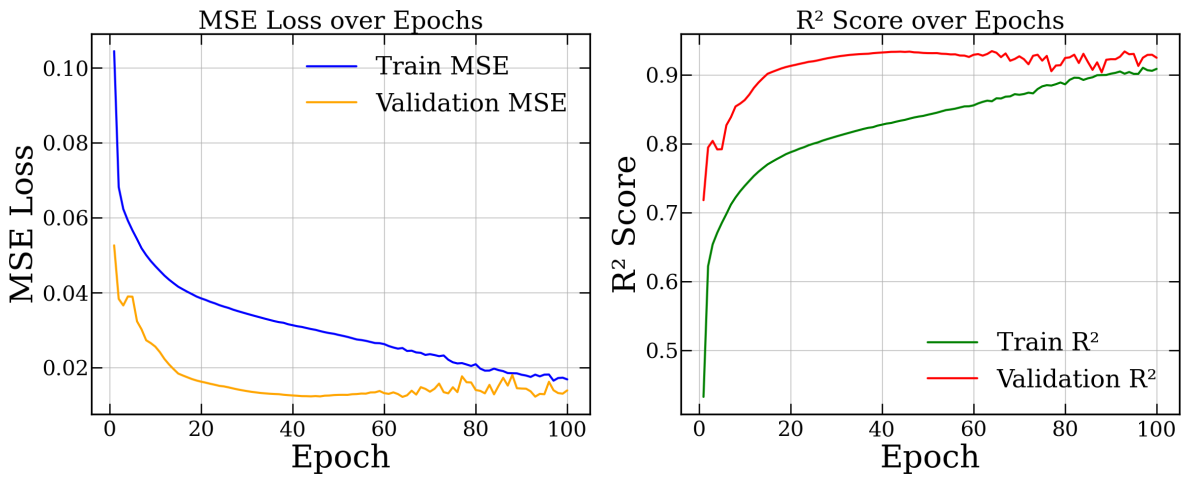


Figure 4.11: The plot shows variations of MSE and R^2 for CosmoUiT for training and validation data over epochs.

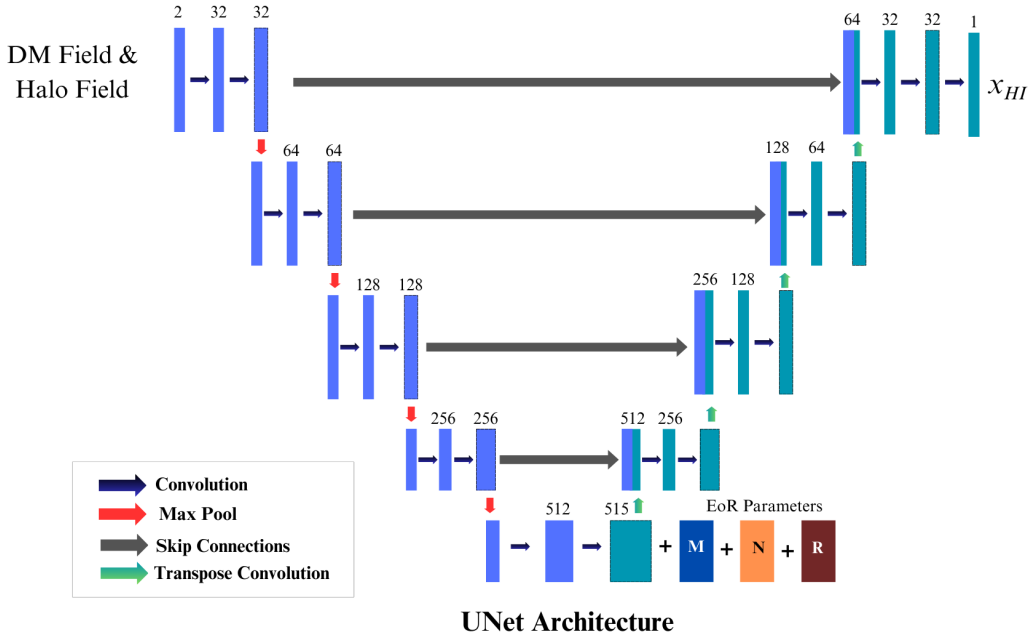


Figure 4.12: Model Architecture of CosmoUNet.

4.4.2 Results of CosmoUNet

We trained the CosmoUNet for 100 epochs with 10 batch-size which took about 23 GPU hours (on NVIDIA RTX-A4000-16GB GPU), which is the same duration as that of the CosmoUiT model. From the Figure 4.14 is evident that though the training loss improves over the epoch the validation loss is almost constant for all the epochs with oscillations after it is crossed by the training loss, implying that the model can perform well on the seen data but struggles to make predictions for the unseen data. This is a classic example of overfitting where the model memorizes the data rather than learning the pattern. It can be observed in its predictions show in Figure 4.13. Since the input fields are fixed and the variation in the output comes from mainly the three EoR parameters, the model becomes biased towards the fixed inputs and struggles to produce parameter-specific outputs; however, adding transformer encoder layer in front of

Component	No. of Feature Maps	Filter Size	Activation Function
Encoder			
DoubleConv Layer (enc1)	32 Filters	3x3x3	ReLU
DoubleConv Layer (enc2)	64 Filters	3x3x3	ReLU
DoubleConv Layer (enc3)	128 Filters	3x3x3	ReLU
DoubleConv Layer (enc4)	256 Filters	3x3x3	ReLU
MaxPool3D (Pooling Layer)	-	2x2x2	-
Bottleneck			
DoubleConv (with Parameters)	512 Filters	3x3x3	ReLU
Decoder			
ConvTranspose3D (upconv4)	256 Filters	2x2x2	-
DoubleConv (dec4)	256 Filters	3x3x3	ReLU
ConvTranspose3D (upconv3)	128 Filters	2x2x2	-
DoubleConv (dec3)	128 Filters	3x3x3	ReLU
ConvTranspose3D (upconv2)	64 Filters	2x2x2	-
DoubleConv (dec2)	64 Filters	3x3x3	ReLU
ConvTranspose3D (upconv1)	32 Filters	2x2x2	-
DoubleConv (dec1)	32 Filters	3x3x3	ReLU
Final Conv Layer	1 Filter	3x3x3	-

Table 4.3: Summary of the CosmoUNet (base model) architecture.

it helps us to encode the information about three parameters beforehand, hence enabling it to produce parameter specific outputs.

4.5 Tuned CosmoUiT

Since the CosmoUiT was the best-performing base model architecture, we chose it for further analysis and improvements.

4.5.1 Model Architecture of Tuned CosmoUiT

We did hyperparameter tuning for the previous model, where we increased the number of nodes in the fully connected layers by four times for all linear projections, which include tokenization of patches, parameter embedding, and feed-forward layers of the transformer encoder block. For the UNet part, after mapping the first image to 16 feature maps, we increased the number of feature maps twice at each step while decreasing the feature map size by half. We reversed the process for the decoder arm, i.e., we halved the number of feature maps while increasing their size by twice. Additionally, instead of using double convolution for combining the features from the encoder arm with the corresponding features of the decoder arm, we used single convolution. All these changes improved the model's performance while making it lighter than

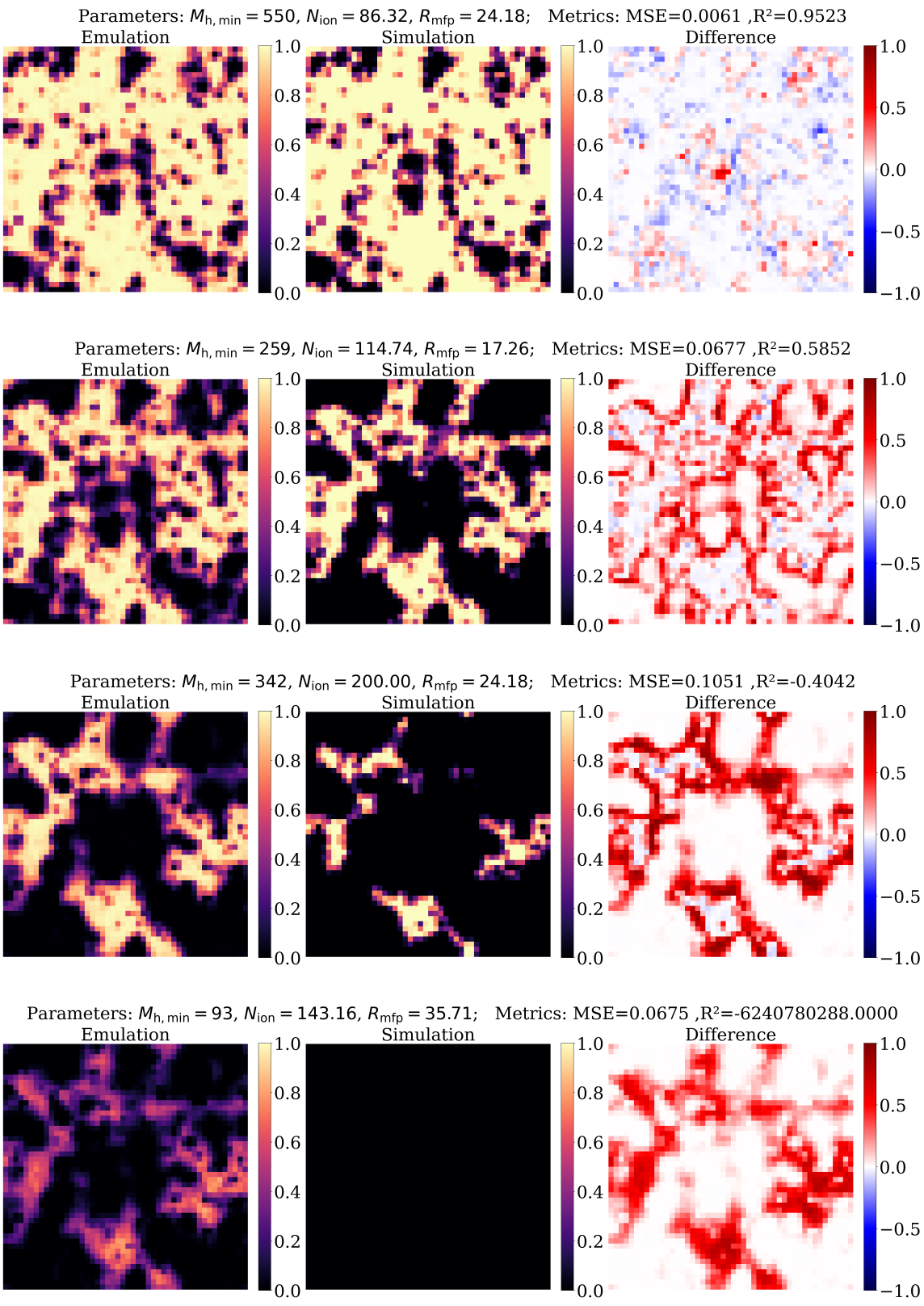


Figure 4.13: *Title:* EoR Parameters and Metric Scores . *First column:* Emulation Output. *Second column:* Simulation Output. *Third column:* Difference.

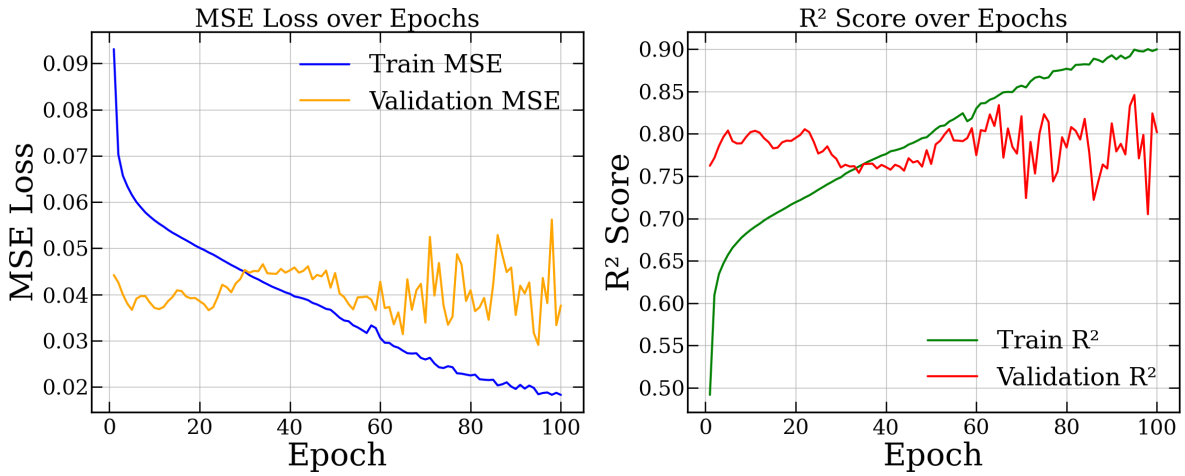


Figure 4.14: The plot shows variations of MSE and R^2 for CosmoUNet for training and validation data over epochs.

the base CosmoUiT model. This also enabled us for using a much larger batch size hence speeding up the training process. The model summary of the tuned model is given in the table 4.4.

4.5.2 Results of Tuned CosmoUiT

Since this model was lighter, we could increase the batch size for training. We trained it using a while loop with stopping criteria that the training should stop if the validation loss does not improve after 10 additional epochs from the last improvement. The model was trained for 722 epochs with a batch size of 96 for 17.8 GPU hours (on NVIDIA RTX-A4000-16GB GPU), which was significantly lesser than the previous models, given the training epochs of 7 times higher. The comparison between true and predicted output shown in Figure 4.15 shows that the boundaries are a bit sharper than those of the base CosmoUiT. Further analysis has shown that this model still struggles at the boundaries of the parameter space (Section: 4.6.2) and gives poor results for unseen random seeds (Section: 4.7), hence there is still scope for improvement.

4.6 Performance Comparison

4.6.1 Distribution of MSE and R^2 Score

After the training is complete, we make predictions on the entire parameter space and compare them with the true output to calculate the MSE and R^2 score. This is done for the three base models and one tuned model, and the distribution of MSEs and R^2 score is shown as boxplots in the Figures 4.18 - 4.20. Figure 4.17 compares the boxplot with the histogram. The yellow line represents the second quartile, also known as the median, with the left edge of the box representing the first quartile and the right edge representing the third quartile. The difference

Component	No. of Feature Maps	Filter Size	Activation Function
Patch Embedding	512	8x8x8	-
Projection Layer	512	-	-
Position Embedding	512	-	-
Parameter Embedding	512	-	-
Fully Connected Layer	512	5120	-
Transformer Encoder Layers	512	-	ReLU
- Self-Attention	8 Heads	-	-
- Feedforward	1024 Units	-	ReLU
- Layer Normalization	512	-	-
UNet3D Encoder			
DoubleConv Layer (enc1)	16 Filters	3x3x3	ReLU
DoubleConv Layer (enc2)	32 Filters	3x3x3	ReLU
DoubleConv Layer (enc3)	64 Filters	3x3x3	ReLU
DoubleConv Layer (enc4)	128 Filters	3x3x3	ReLU
MaxPool3D (Pooling Layer)	-	2x2x2	-
Bottleneck			
DoubleConv (with Parameters)	128 Filters	3x3x3	ReLU
UNet3D Decoder			
ConvTranspose3D (upconv4)	64 Filters	2x2x2	-
SingleConv (dec4)	64 Filters	3x3x3	ReLU
ConvTranspose3D (upconv3)	32 Filters	2x2x2	-
SingleConv (dec3)	32 Filters	3x3x3	ReLU
ConvTranspose3D (upconv2)	16 Filters	2x2x2	-
SingleConv (dec2)	16 Filters	3x3x3	ReLU
ConvTranspose3D (upconv1)	16 Filters	2x2x2	-
SingleConv (dec1)	1 Filters	3x3x3	ReLU

Table 4.4: Summary of the CosmoUiT (tuned model) architecture.

between the third and first quartile is called the interquartile range (IQR), and these whiskers are $1.5 \times$ IQR. All those points that are outside these whiskers are called outliers. The goal here is to have an MSE distribution that is constrained towards the lower values without the significant number of outliers. Same goes for the R^2 score, except here the distribution should be constrained towards the higher values around 1.

The distribution of MSEs for different models (Figure 4.18) shows that, for CosmoViT model the error is spread over a larger range followed by the CosmoUNet. It is also evident that tuning the model makes the error more constrained towards the lower values. The distribution of R^2 score for different as shown in the Figure 4.19, shows the spread over larger ranges for all the models, but this doesn't imply that all the models make predictions that are far off of the true outputs but if you see the formula for R^2 score given in the Equation 4.5, if the field is nearly ionized or neutral then the mean value of the field becomes closer to the true values of the

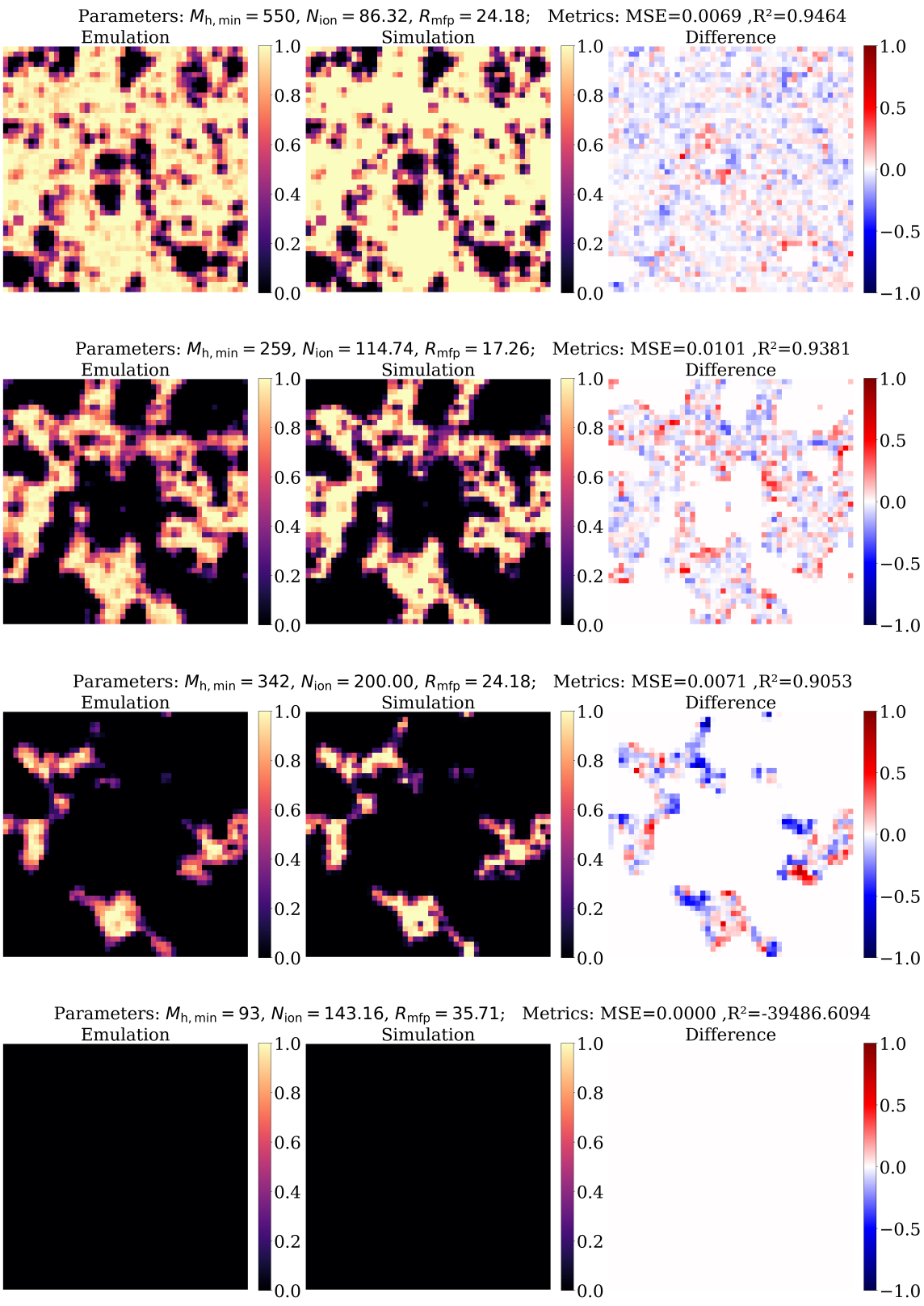


Figure 4.15: *Title:* EoR Parameters and Metric Scores . *First column:* Emulation Output. *Second column:* Simulation Output. *Third column:* Difference.

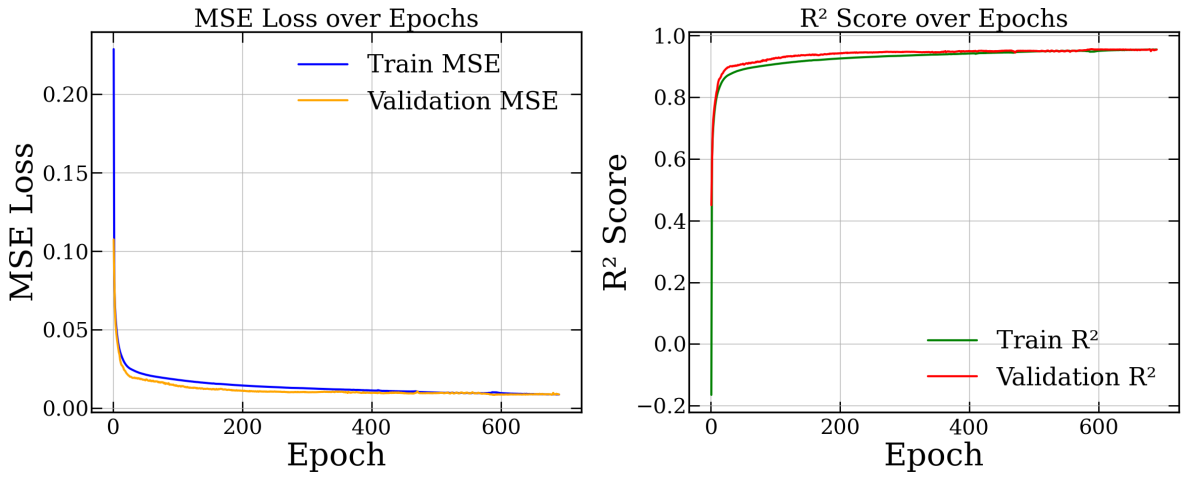


Figure 4.16: The plot shows variations of MSE and R^2 for tuned CosmoUiT for training and validation data over epochs.

field hence giving us very small value, closer to zero in the denominator and if the prediction is slightly off of the true value the value in the denominator is non-zero hence the negative part of the R^2 score shoots up giving us such high values. If we replace all the negative values of R^2 with zero the boxplot looks like Figure 4.20, implying that the tuned CosmoUiT model has better over all R^2 score.

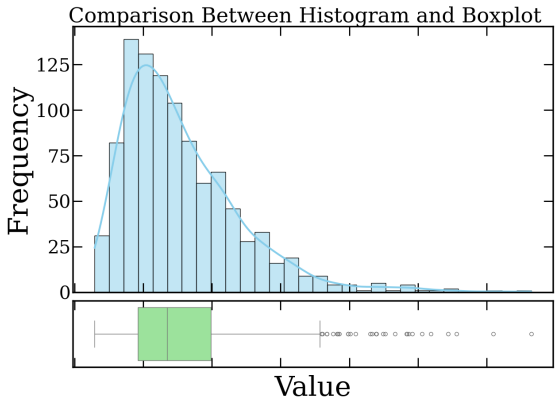


Figure 4.17: Comparison between histogram and boxplot.

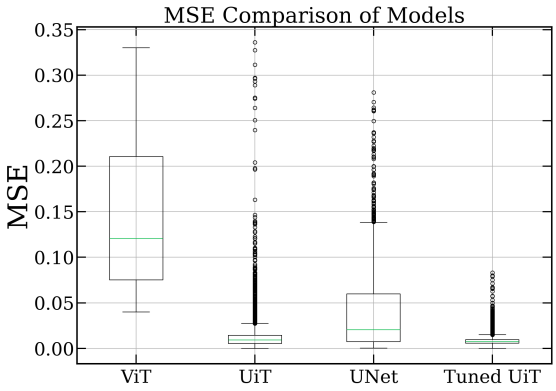


Figure 4.18: Comparison between MSEs of different model architectures.

4.6.2 Variation in MSE and R^2 Score with Reionization Parameters

Figure 4.21 - 4.22 discuss the variation of x_{HI}, MSE and R^2 score with EoR parameters after making predictions on the entire parameter space. From the Figure 4.21, it is evident that the fields with minimum $M_{h,min}$, maximum N_{ion} and maximum R_{mfp} show minimum x_{HI} and vice-versa. In Figure 4.24 we can see that there are two edges that corresponding to higher MSE . The vertical edge corresponds to $N_{ion} = 200$ which is the upper limit of that parameter range and $M_{h,min} = 10 \times 10^8\,M_\odot$ which is a lower limit of that parameter range. The horizontal edge corresponds to $M_{h,min} = 10 \times 10^8\,M_\odot$ and $R_{mfp} = 1.12\,\text{Mpc}$ which is a lower limit of

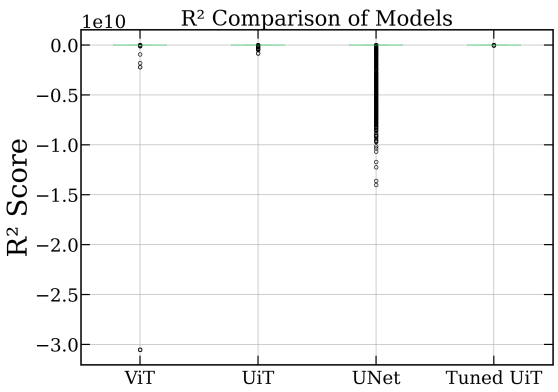


Figure 4.19: Comparison between R^2s of different model architectures.

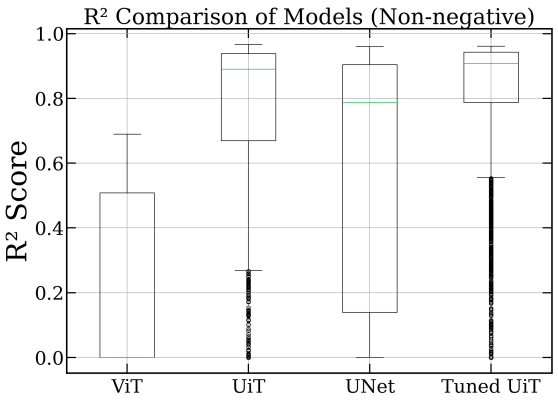


Figure 4.20: Comparison between R^2s (non-negative) of different model architectures.

that parameter range. Figure 4.25 and 4.26 show variation of R^2 along with the three EoR parameters, in here the negative values of R^2 are replaced with zero for making it easier to visualize them along with the other significant (non-negative) values. In Figure 4.25 , the right corner that corresponds to zero value of R^2 but the same corner in MSE's variation in Figure 4.23 corresponds to very small MSE, the same disparity can also be seen in the two planes of the Figure 4.26, first one corresponds to $M_{h,min} = 10 \times 10^8\, M_\odot$ and the second one corresponds to $R_{mfp} = 1.12$ Mpc , however, the comparison of these regions with the neutral fraction cube in Figure 4.21 and 4.22 shows that, all these regions belong to the either very low neutral fraction (the corner 4.25 and left plane 4.26) or to the very high neutral fraction (the lower plane 4.26), because at this point the negative part of R^2 score shoots up as discussed earlier.

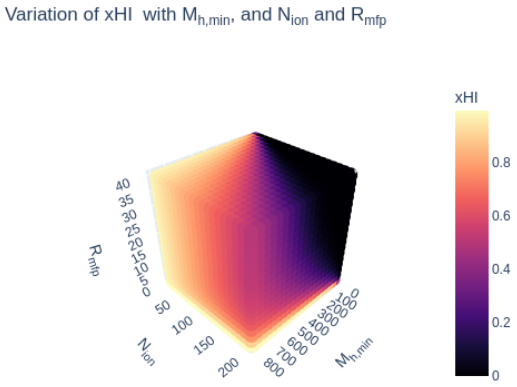


Figure 4.21: Variation of x_{HI} with EoR parameters (front view).

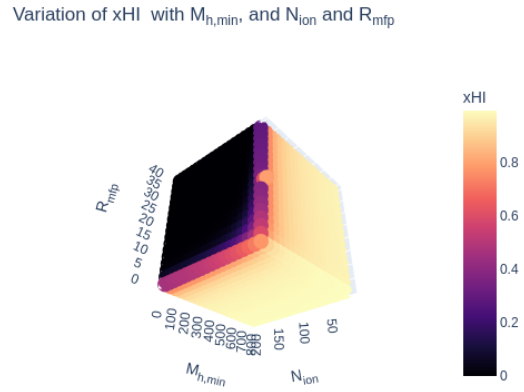


Figure 4.22: Variation of x_{HI} with EoR parameters (back view).

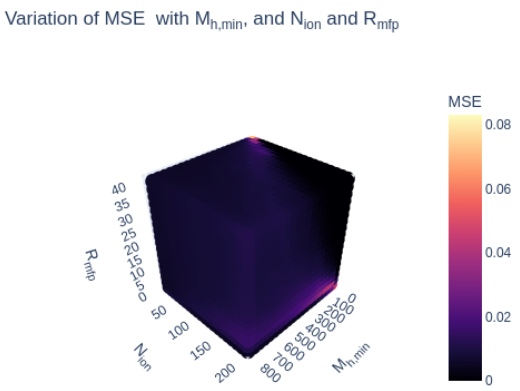


Figure 4.23: Variation of MSE with EoR parameters (front view).

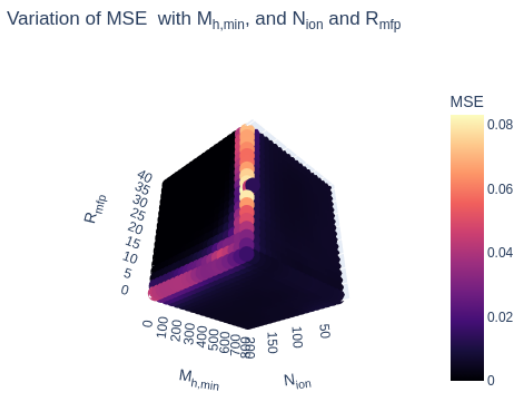


Figure 4.24: Variation of MSE with EoR parameters (back view).

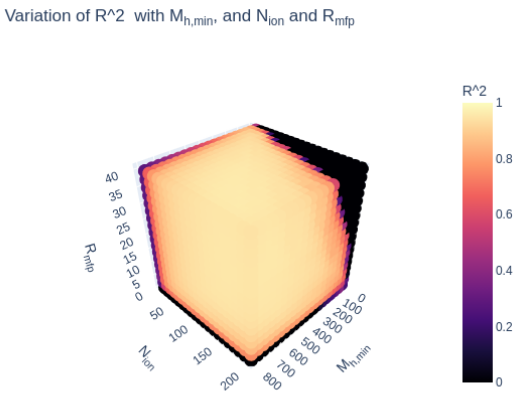


Figure 4.25: Variation of R^2 score with EoR parameters (front view).

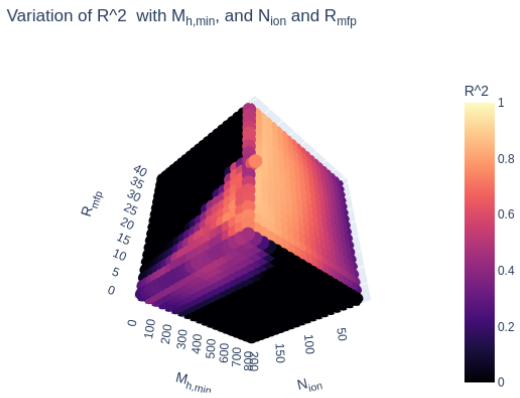


Figure 4.26: Variation of R^2 score with EoR parameters (back view).

4.7 Model Generalization Test: CosmoUiT48

A good generalization is a crucial quality in the case of the emulation tasks. This corresponds to the ability of the of the model to make predictions on the unseen data. To test our model's performance for the same, we tried making predictions on the inputs generated using an entirely different random seed than the one it was trained for. Since it has not seen different input fields it should have been biased towards the particular realization used. However, using data augmentation techniques such as rotation, one can train the model to capture the effect of variation in the fields to some extent while providing just one realization of each of the inputs.

4.7.1 Prediction for Unseen Random Seed

We made predictions using CosmoUiT48 (CosmoUiT for 48^3 grid resolution). The Figures 4.27 - 4.30 show emulation and simulation outputs with the difference between the two. The MSE for these predictions is very high, and the R^2 score is negative, implying very poor model performance. From the difference of the fields, it is visible that the morphological features are not captured properly.

4.7.2 Analysis of Generalization Ability

The key reason behind inaccurate predictions by CosmoUiT48 was the difference between the distributions of input fields obtained from seen and unseen random seeds after downsampling. To quantify the similarity between the two distribution,s we used L_2 distance (Euclidean distance). The L_2 distance measures the root-mean-square difference between two probability density functions (PDFs). Mathematically,

$$D_{L2}(p, q) = \left(\int (p(x) - q(x))^2 dx \right)^{1/2} \tag{4.18}$$

Where,

- $p(x)$ and $q(x)$ are the two PDFs.
- dx small element for integration over x.

However, in our case, the distribution is discrete, therefore, it can be written as,

$$D_{L2} \approx \left(\sum_i (p_i - q_i)^2 \, \Delta x \right)^{1/2} \tag{4.19}$$

Where,

- p_i, q_i = values of the PDFs at bin is.
- Δx = bin width.

The smaller value of the L_2 distance implies similar distributions. We normalize the histograms and do bin-wise comparison to calculate the L_2 distances between the distribution of fields generated via seen and unseen random seeds.

The Figures 4.31 - 4.34 the histograms with both axes in log scale for dark matter and halo distribution for different resolutions. The title contains the value of L_2 distance for each pair. As we downsample the field from 384^3 to 48^3 the L_2 distance between the two fields increases. The distributions of input fields, especially the halo field differ vastly for low resolution compared to the high resolution scenario, hence, the CosmoUiT48 model struggles to make accurate predictions for unseen random seeds.

Predictions made by CosmoUiT48 for unseen data

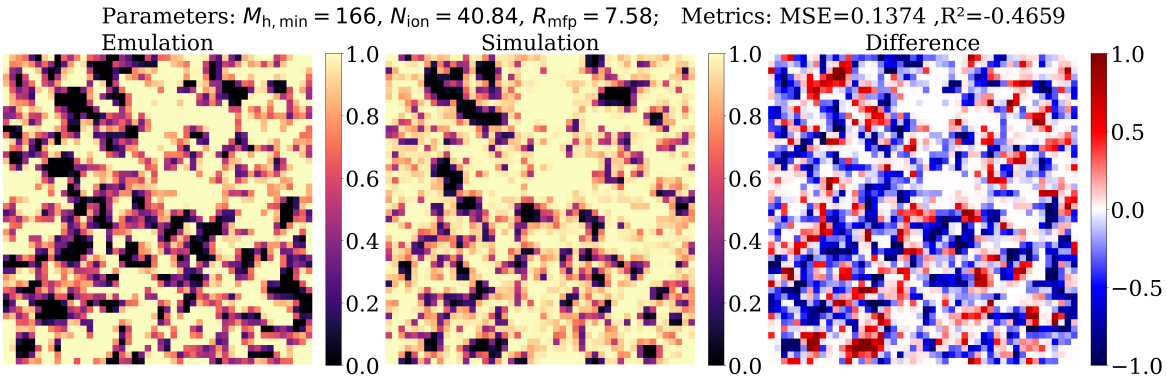


Figure 4.27: Comparison between x_{HI} fields for parameter set 1.

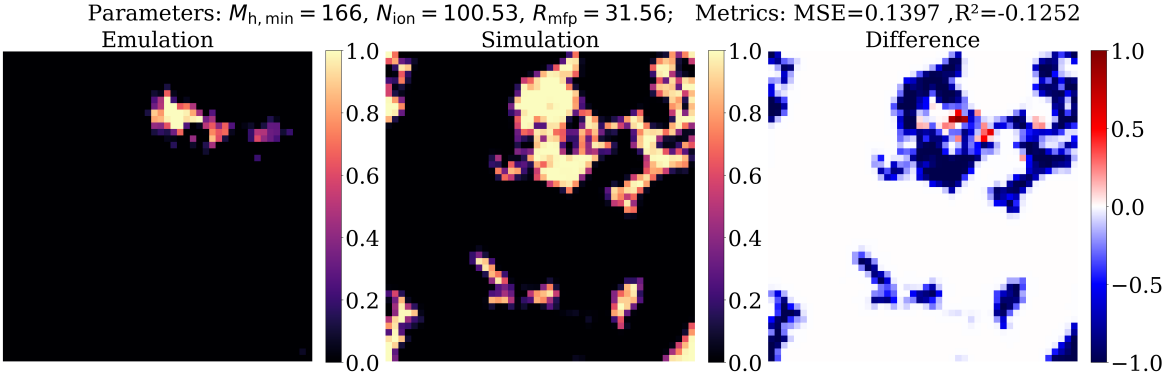


Figure 4.28: Comparison between x_{HI} fields for parameter set 2.

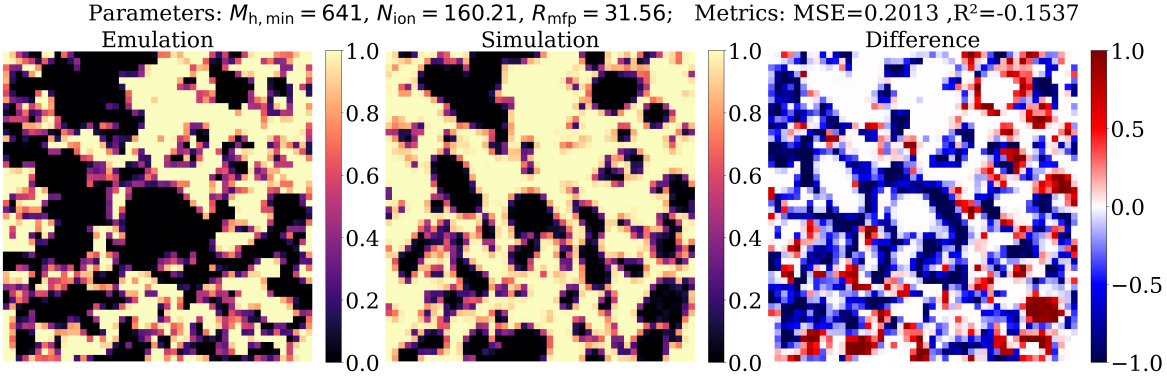


Figure 4.29: Comparison between x_{HI} fields for parameter set 3.

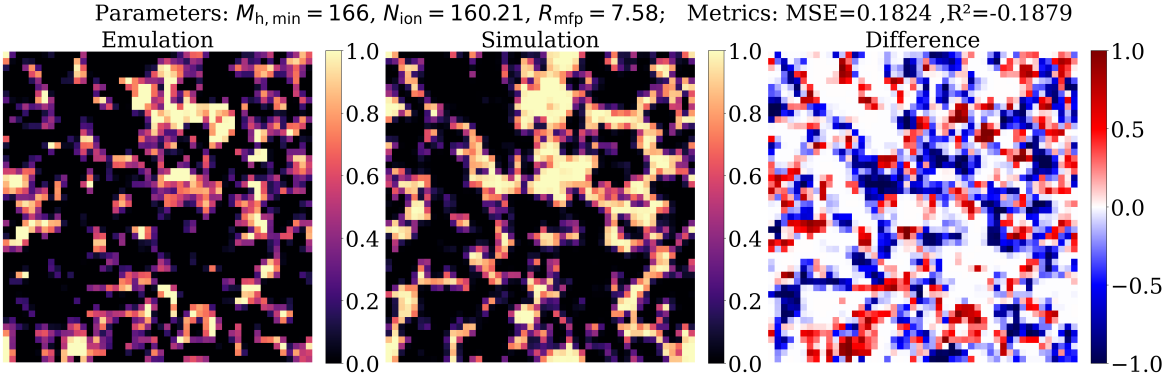


Figure 4.30: Comparison between x_{HI} fields for parameter set 4.

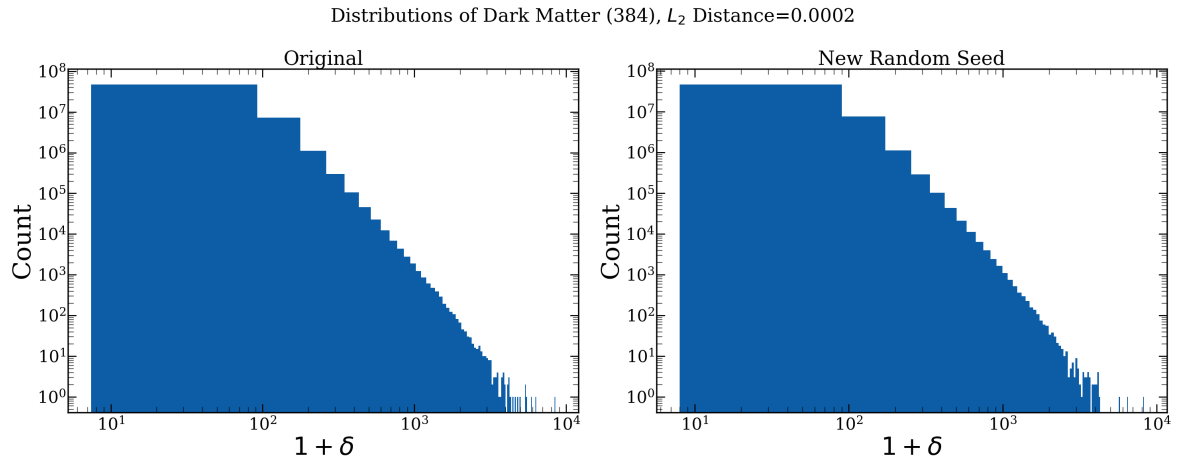


Figure 4.31: Comparison between distribution of dark matter for seen and unseen random seed (384^3 resolution).

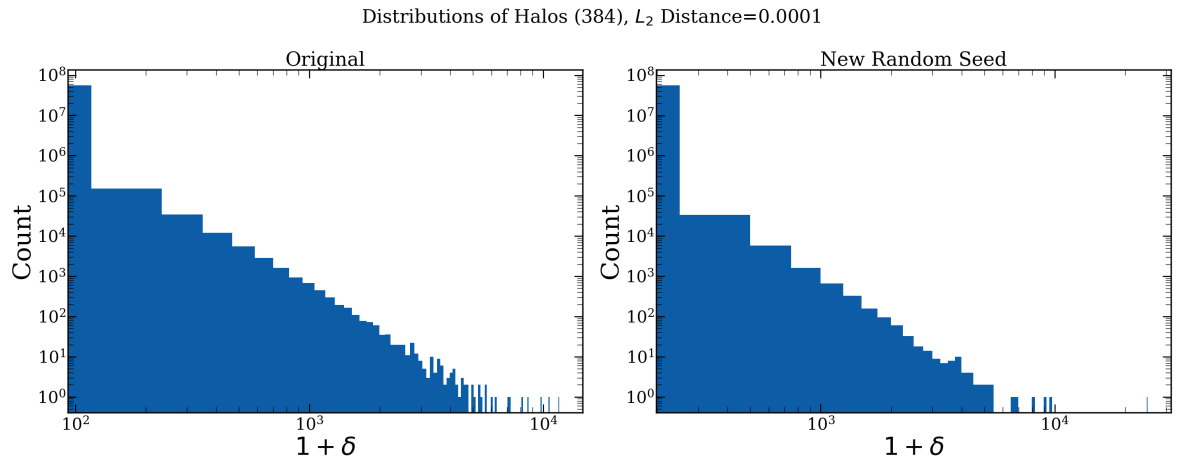


Figure 4.32: Comparison between distribution of halo for seen and unseen random seed (384^3 resolution).

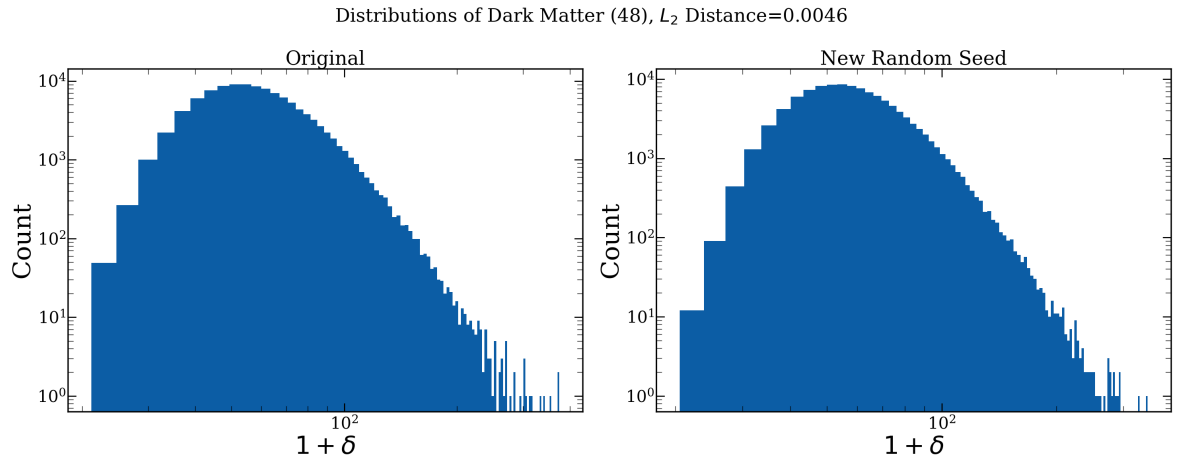


Figure 4.33: Comparison between distribution of dark matter for seen and unseen random seed (48^3 resolution).

4.8 CosmoUiT96

The previously tuned model faced problems at the lower limits of the parameter space and could not make accurate predictions there. It also struggled with generalization for unseen random seeds. The reason behind this is the low resolution of the fields. The lower boundary of the

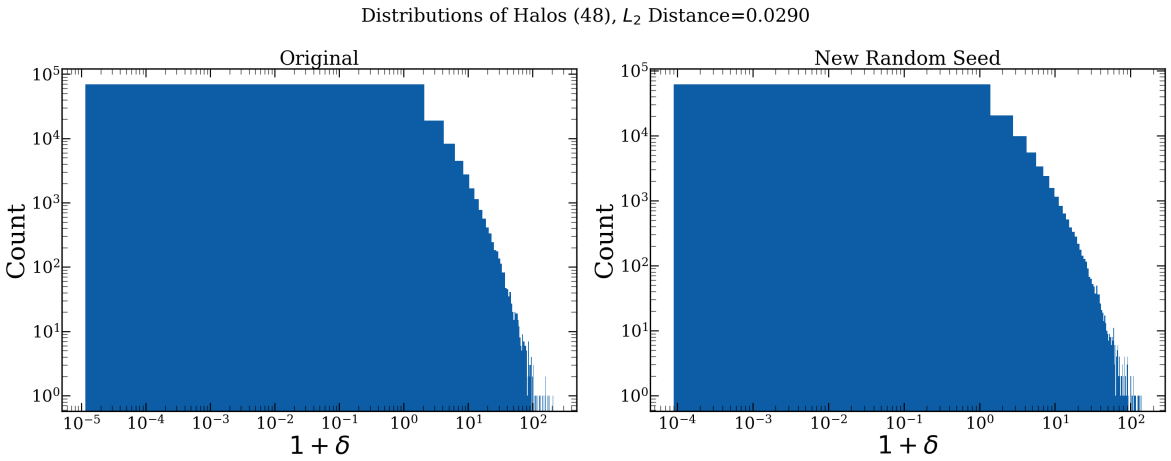


Figure 4.34: Comparison between distribution of halo for seen and unseen random seed (48^3 resolution).

mean free path is 1.12 Mpc; however, our grid size was 4.48 Mpc, Therefore, the effect of this parameter is not significantly visible for these fields for lower values, causing problems when training and predicting. The same problem is faced in the case of the lower boundary of the minimum halo mass parameters. When reducing the fields from 384^3 to 48^3, the range of halo masses was scaled down, as a large number of grid points were zero for this field. As a result, the smallest values of minimum halo mass got reduced further. Given these issues due to low resolution, in this work we increased the resolution from 48^3 to 96^3, resulting in a grid size of 2.24 Mpc. One could argue that the grid size remains larger than the minimum possible mean free path; however, because we now average over fewer pixels (216) compared to the previous method (512), the effect of lower boundaries becomes more apparent in the resulting fields.

4.8.1 Methods for Reduction in Resolution

The initial data generated via multi-step simulation processes had a grid resolution of 384^3. We used the `BlockReduce` method offered by `scikit_image` for reducing the resolution to 48^3 in our earlier work. In this work, we will use Gaussian Smoothing and Resampling to reduce the data from 384^3 to 96^3 and compare it with the Block Reduce method.

4.8.1.1 Block Reduce

As discussed in Section 4.2.2, the main arguments of the `BlockReduce` method are the input field that is to be reduced and the number of blocks to be combined over an operation such as sum, average, median, minimum, maximum, etc. Earlier, we combined 8 blocks along each axis via averaging, resulting in a single block from 512 blocks. In this work, we are combining 4 blocks along each axis via averaging, which results in a single block for every 216 blocks (Figures: 4.36, 4.37, 4.38).

4.8.1.2 Gaussian Smoothing and Resampling

This method was first given in [44] and used for downsampling high-resolution simulation outputs while preserving the finer structures. This method involves smoothing the field with a Gaussian kernel of $\sigma = 0.5$ and choosing every second pixel from the resulting field as shown in the Figure 4.35. It reduces the resolution by a factor of two along each axis. We repeated this process twice to reduce the resolution by a factor of 4^3 (Figures: 4.36, 4.37, 4.38).

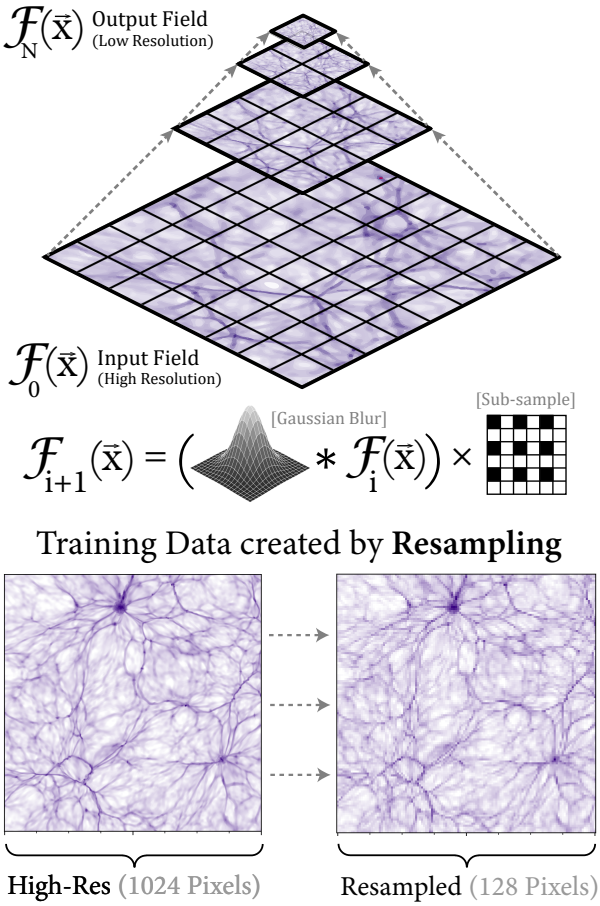


Figure 4.35: Schematic Representation of Gaussian Smoothing (GS) and Resampling Method. (Figure Credits: C. Jacobus et al. [44])

4.8.1.3 Comparison

The Figure 4.39 shows the comparison of power spectrums for higher resolution (384^3) and reduced resolution (96^3) using block reduce and Gaussian smoothing + resampling. It can be seen that the GS Resampling method works slightly better at higher k modes (smaller length scales) for halo and dark matter fields. This method is a combination of two methods: (1) subsampling and (2) local mean. Both of this methods have their own limitations, and combining them to form Gaussian smoothing + resampling helps us to overcome these limitations to some extent. To reduce the field by factor of n along each axis the subsampling method takes every n^{th} pixel (starting anywhere from 0 to n-1) out of every n^3 pixels and the local mean method

does smoothing of the field and takes mean of every n^3 pixels [45]. The main problem with subsampling is aliasing. This method keeps every n^{th} value and discards the rest, but those skipped values can contain important details or patterns. When these are lost, their influence doesn't just disappear, it gets misrepresented at the wrong spatial scales, a phenomenon known as aliasing. As a result, while the overall range of values (like highs and lows) might be preserved, the underlying structure of the data becomes distorted. This is especially problematic in cosmology, where small-scale patterns carry crucial information. The local mean method works by replacing each group of values, such as every n^3 values, with their average. This smooths out sharp or noisy details, which helps to reduce aliasing when downsampling. However, this smoothing blurs out rare or extreme features, like peaks and edges, causing the field to appear overly averaged. As a result, while the resized field looks cleaner, its range of values becomes compressed, and the probability distribution shifts toward the mean. This leads to a biased representation where uncommonly high or low values are underrepresented, making the PDF less accurate. The Gaussian smoothing + resampling method works better because it strikes a balance between preserving spatial structure and avoiding aliasing. Instead of aggressively averaging or blindly skipping values, it applies gentle Gaussian filtering before each resampling step. This filtering suppresses high-frequency components that would otherwise cause aliasing, while still maintaining the essential structure and variability in the data. By performing this process progressively, typically in multiple steps rather than a single downscale, it gradually reduces resolution without distorting important patterns. As a result, both the power spectrum and the probability distribution are better preserved, making this method more faithful to the original data. With these considerations in mind, we adopt the Gaussian smoothing and resampling method for all subsequent data used in this work.

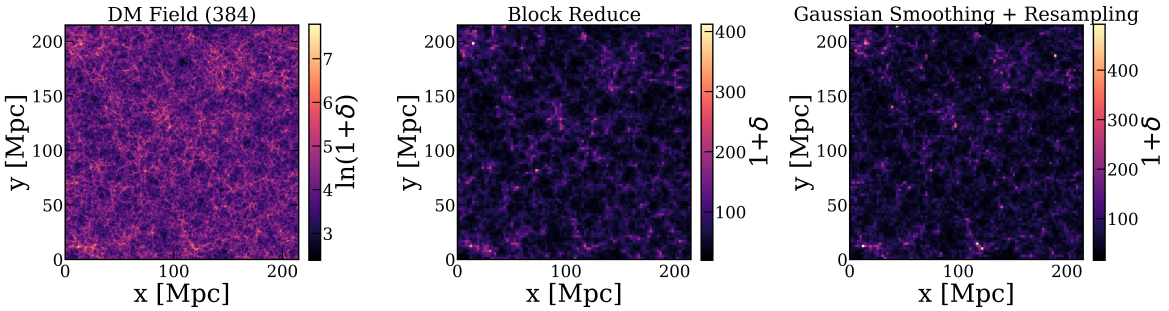


Figure 4.36: Comparison between reduction methods for dark matter field.

4.8.2 Model Architecture of CosmoUiT96

We have made few changes in this architecture to take the increase in resolution into account. The higher resolution requires increased model depth to extract lower-level features, which also increase the computational cost. We try to counter it by reducing the model complexity (i.e., removing unnecessary convolution operations) and by using aggressive variation (i.e. rapidly

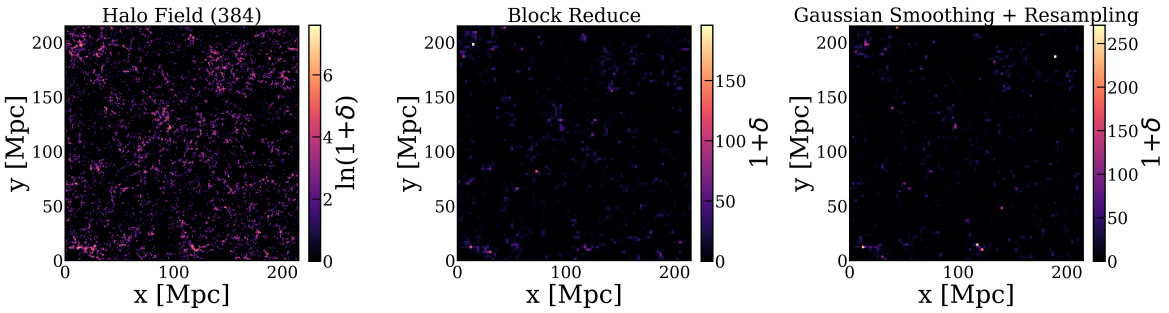


Figure 4.37: Comparison between reduction methods for halo field.

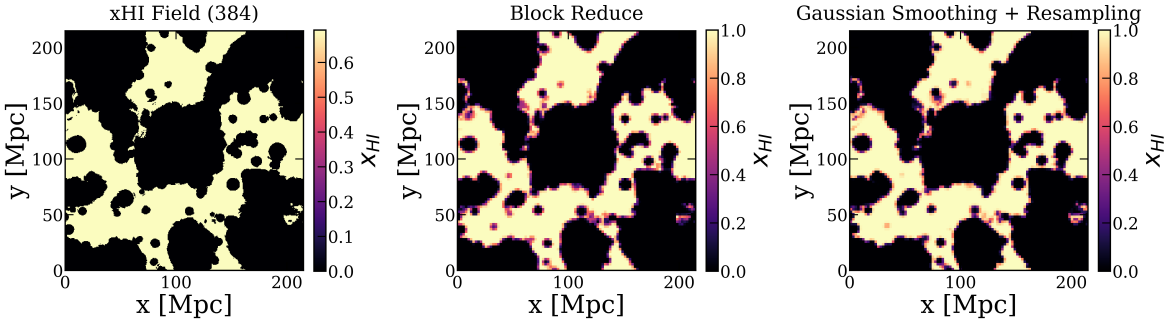


Figure 4.38: Comparison between reduction methods for x_{HI} field.

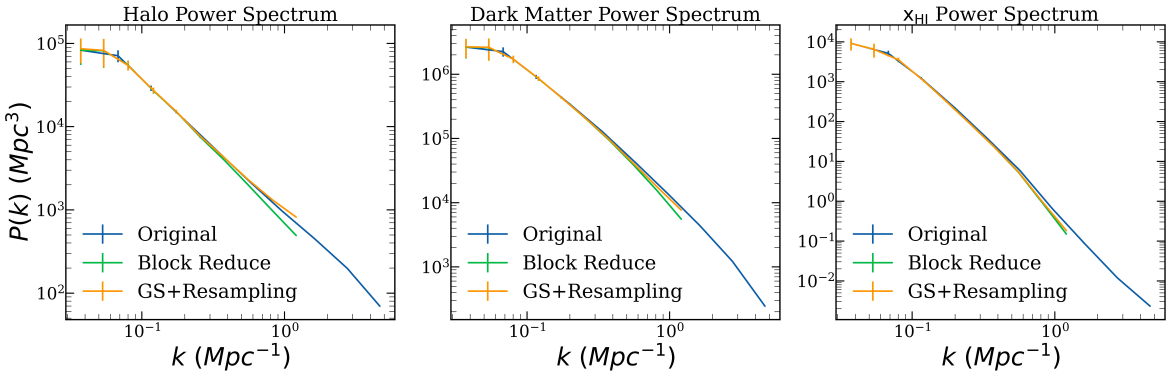


Figure 4.39: Comparison between power spectrums for different methods of reduction.

changing the sizes and numbers of feature maps) while extracting and reconstructing feature maps. In line with these adjustments, the increased input resolution also results in a longer sequence length, growing from 216 to 1728. The encoder layer consist of multiple layers in the following order: (1) 3D Convolution, (2) Dropout (10%), (3) Batch Normalization, (4) LeakyReLU, (5) Max Pooling. The decoder layer consists of a transpose convolution and skip connections. The skip connections, as mentioned earlier, are used for combining the feature maps from the encoder layer at the same level. They consist of the following layers: (1) 3D convolution, (2) Dropout (10%), (3) Batch Normalization, (4) LeakyReLU.

Component	No. of Feature Maps	Filter Size	Activation Function
Patch Embedding	512	8x8x8	-
Projection Layer	512	-	-
Position Embedding	512	-	-
Parameter Embedding	512	-	ReLU
Fully Connected Layer	512	5120 (Mapped)	-
Transformer Encoder Layers	512	-	ReLU
- Self-Attention	8 Heads	-	-
- Feedforward	512 Units	-	ReLU
- Layer Normalization	512	-	-
UNet3D Encoder			
Encoder Layer 1	32 Filters	3x3x3	LeakyReLU
Encoder Layer 2	64 Filters	3x3x3	LeakyReLU
Encoder Layer 3	128 Filters	3x3x3	LeakyReLU
Encoder Layer 4	256 Filters	3x3x3	LeakyReLU
Encoder Layer 5	512 Filters	3x3x3	LeakyReLU
Bottleneck + Parameters	512+3		
UNet3D Skip+Decoder			
Skip+Decoder Layer 5	256 Filters	2x2x2	LeakyReLU
Skip+Decoder Layer 4	128 Filters	2x2x2	LeakyReLU
Skip+Decoder Layer 3	64 Filters	2x2x2	LeakyReLU
Skip+Decoder Layer 2	32 Filters	2x2x2	LeakyReLU
Decoder Layer 1	1 Filters	2x2x2	LeakyReLU

Table 4.5: Summary of the CosmoUiT96 architecture.

4.8.3 Results of CosmoUiT96

We trained this model for 50 epochs, which took around 100 GPU hours on NVIDIA A100-SXM4-40GB GPU. The MSE is 0.012 and the R^2 Score is 0.94. The Figures 4.40 - 4.43 show the comparison between the simulation and the emulation along with the difference between them two. If you observe carefully the primary source of errors is the boundary between the ionized and neutral region. Since there is an abrupt change in the values, the model is not able to capture it predicts gradual change instead, this is called fuzzy boundary problem. It is more prominent in regions with neutral fraction below 0.5. To mitigate this, we need estimate the uncertainties in model's prediction. If we had multiple realizations which are slightly different from each other due to stochastic processes for a given combination of reionization parameters we could quantify the uncertainty as a standard deviation in model's outputs. This is discussed in more details in Section 4.9.

The Figures 4.48 and 4.49 show the variation of MSE with the EoR parameters. It seems that the errors at the edge corresponding to the fixed lower bound of R_{mfp} and $M_{h,min}$, and varying values of N_{ion} have increased. The other edge corresponding to lower bound of $M_{h,min}$

Predictions made by CosmoUiT96 for combinations of EoR parameters

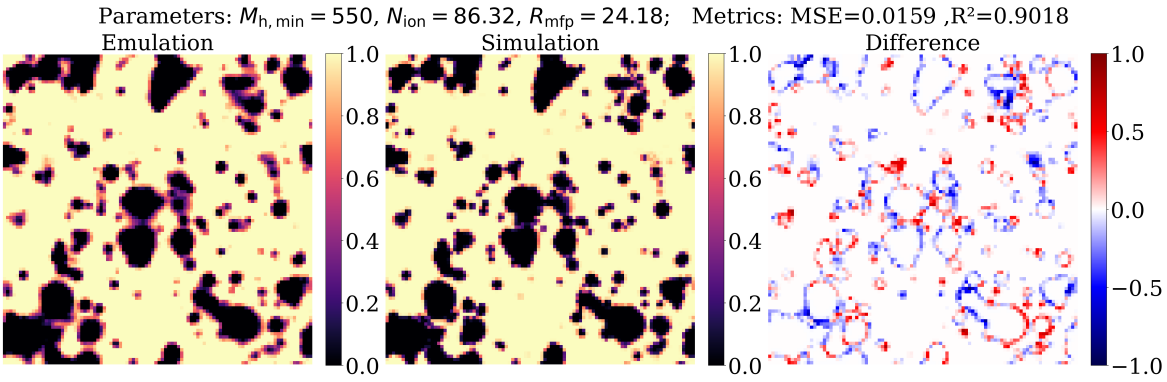


Figure 4.40: Comparison between x_{HI} fields for parameter set 1.

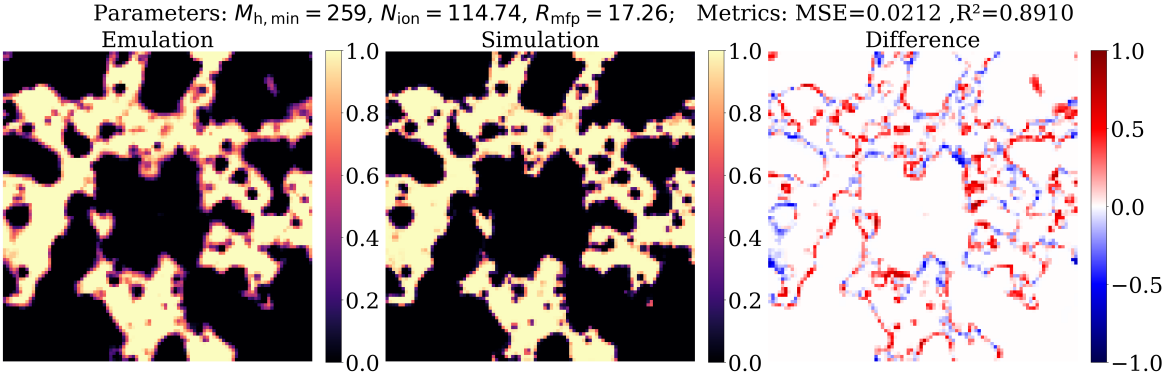


Figure 4.41: Comparison between x_{HI} fields for parameter set 2.

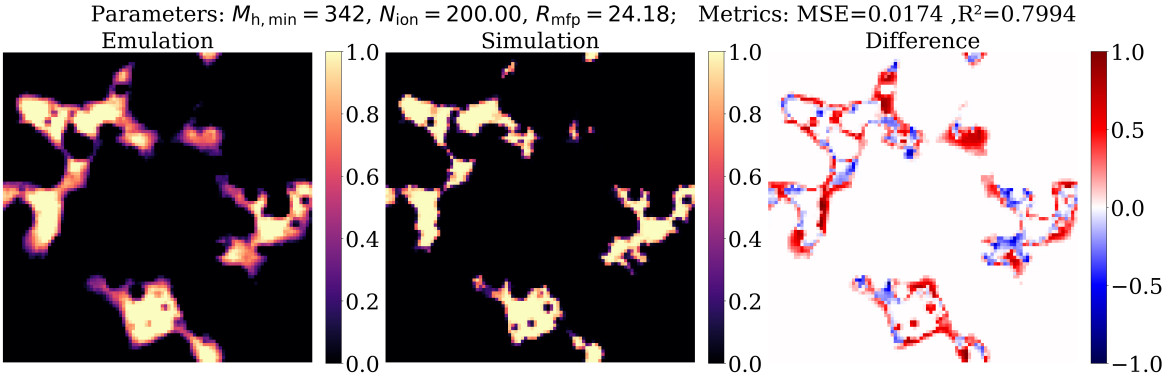


Figure 4.42: Comparison between x_{HI} fields for parameter set 3.

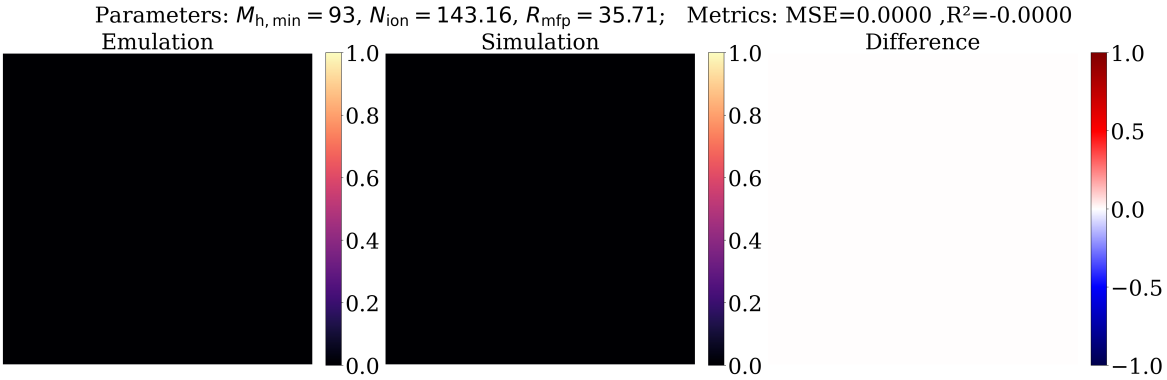


Figure 4.43: Comparison between x_{HI} fields for parameter set 4.

21-cm brightness temperature fields (in mK) calculated using x_{HI} for different combinations of EoR parameters using Equation 1.3

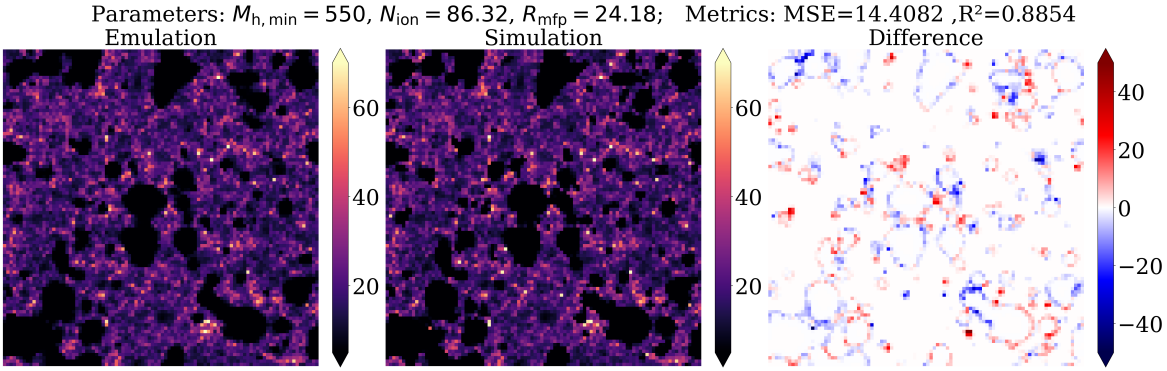


Figure 4.44: Comparison between 21-cm fields for parameter set 1.

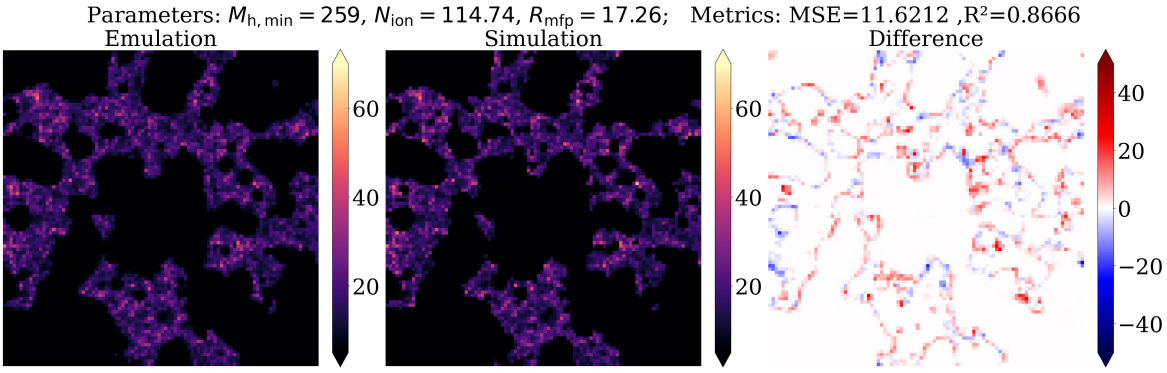


Figure 4.45: Comparison between 21-cm fields for parameter set 2.

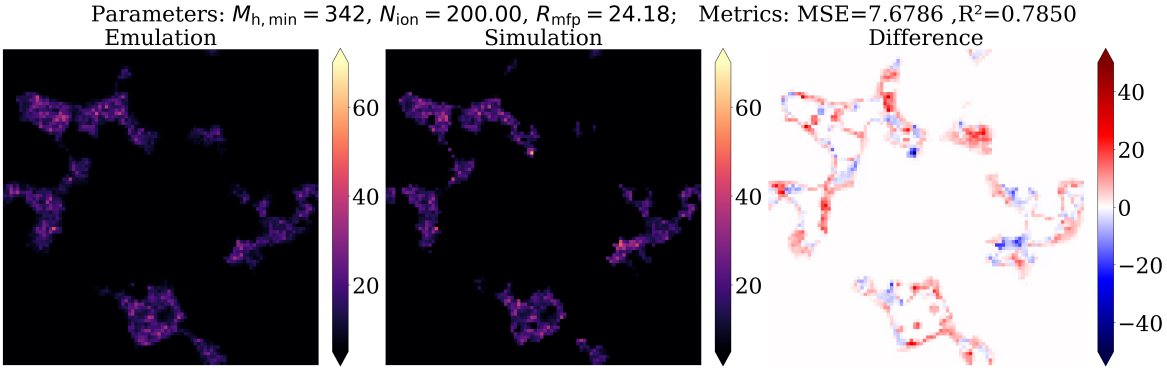


Figure 4.46: Comparison between 21-cm fields for parameter set 3.

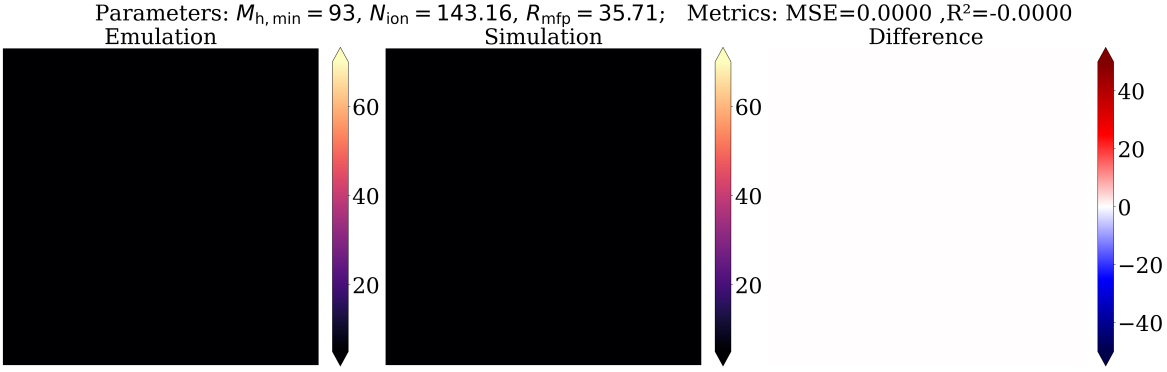


Figure 4.47: Comparison between 21-cm fields for parameter set 4.

and N_{ion} with varying R_{mfp} shows decrease in the errors. The key thing to notice here is that the R^2 Score of the plane corresponding to the lower value R_{mfp} shows significant improvement with no values less than or equal to zero.

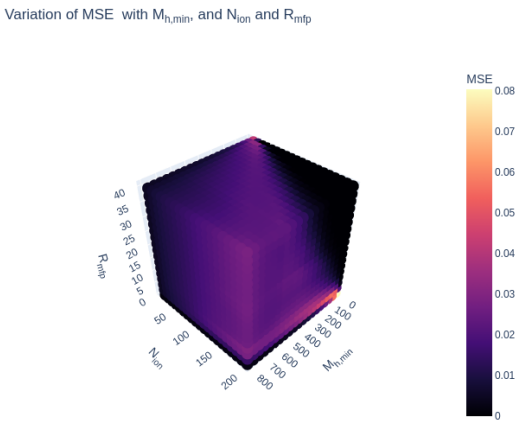


Figure 4.48: Variation of MSE with EoR parameters (front view).

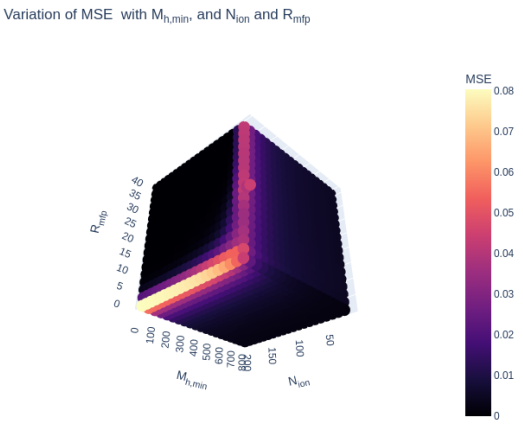


Figure 4.49: Variation of MSE with EoR parameters (back view).

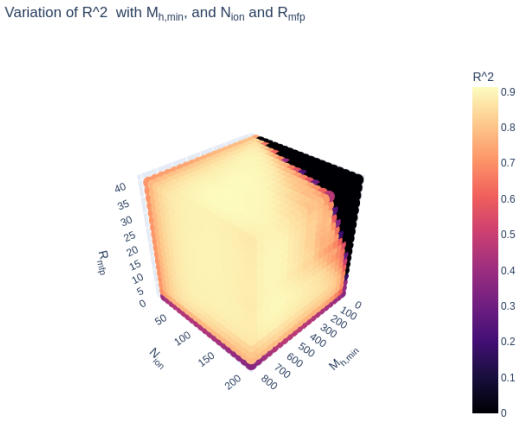


Figure 4.50: Variation of R^2 Score with EoR parameters (front view).

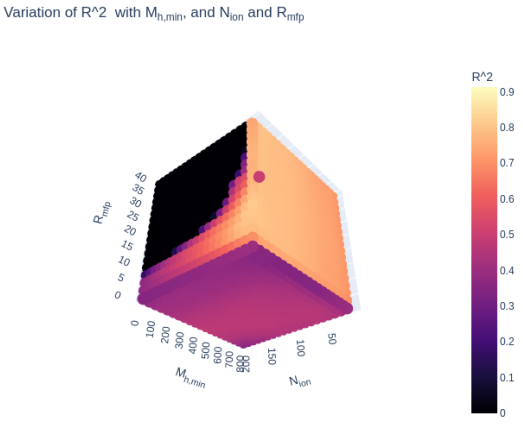


Figure 4.51: Variation of R^2 Score with EoR parameters (back view).

4.9 Uncertainty Estimation

The deep learning based emulators are statistical approximations hence, there are errors associated with the predictions. If these errors are not taken into account while doing the Bayesian inference they may lead to biased estimates of the parameters. To address this, we employ different methods to generate an ensemble of model predictions that are slightly different from each other. We then take their pixel-wise mean as prediction output 4.20, pixel-wise standard

deviation as uncertainty 4.22, and pixel-wise comparison between ground truth and each model prediction gives us the root mean square error (RMSE) 4.21. The formulas for the same are given as follows:

$$\text{Mean Prediction} = \frac{1}{N}\sum_{i=1}^{N} y_{i,pred} \tag{4.20}$$

where

- N is the number of data points (here pixels),
- $y_{i,pred}$ is the predicted value.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{i,true} - y_{i,pred})^2} \tag{4.21}$$

where

- N is the number of data points (here pixels),
- $y_{i,true}$ is the true value,
- $y_{i,pred}$ is the predicted value.

$$\text{Uncertainty} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{i,pred} - \bar{y}_{i,pred})^2} \tag{4.22}$$

where

- N is the number of data points (here pixels),
- $\bar{y}_{i,pred}$ is the mean predicted value,
- $y_{i,pred}$ is the predicted value.

If the mean of the predictions is closer to the ground truth i.e. $\bar{y}_{i,pred} \approx y_{i,true} \rightarrow$ RMSE $\approx$ Uncertainty. Implying that the uncertainty in model's prediction is captured accurately.

4.9.1 Monte-Carlo Dropout

The Monte-Carlo (MC) Dropout method was proposed an alternative uncertainty estimation using Bayesian models in [46, 47]. The Bayesian models are computationally expensive, on the other hand MC dropout method can be implemented by simply adding dropout layers in the neural network without additional computational cost. The dropout layers are used during

training using `model.train()` for regularization i.e. to mitigate the problem of overfitting. The dropout rate decides the fraction of nodes that will be randomly switched off during training. We turn off the dropout layers by using `model.eval()` while evaluating the model over validation or test set. However, if we keep the dropout layers on by removing the `model.eval()` and take ensemble of prediction for same combination of parameters, each prediction will be slightly different, due to different nodes being turned off. This is supposed to capture the stochasticity in model operations and give us uncertainty in its prediction. We took ensemble of 100 such predictions and took their pixel-wise mean and standard deviation for mean prediction and uncertainty. We compared the predictions with ground truth to get the RMSE as discussed earlier. The uncertainty is enhanced by factor of 5 to make it visible in the plot. The scatter plot in Figure 4.55, shows the correlation between RMSE and uncertainty. The dotted line shows the ideal scenario where the RMSE and uncertainty is equal, however, the more realistic scenario would be for them to have a high correlation. The value of the Pearson's correlation coefficient for this method is 0.68, which is not significant. Additionally, RMSE is always greater than the uncertainty, implying that the RMSE never falls below the uncertainty limits of the models, which is an undesirable outcome.

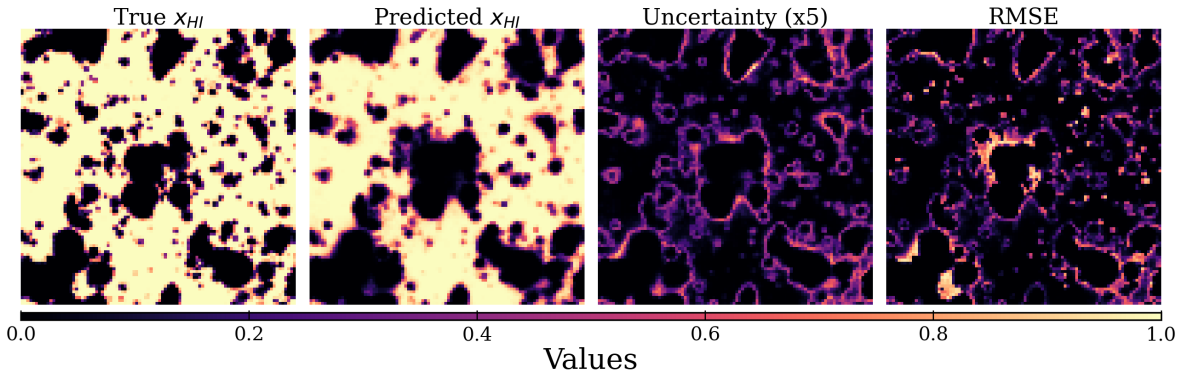


Figure 4.52: Results obtained using Monte-Carlo Dropout Method.

4.9.2 Gaussian Noise Injection

This technique was discussed in [44, 45] to get the uncertainty of ML-based model for Lyman-α flux field and more recently discussed in [48]. In this method, the noise is added to the feature maps whose variance is learned through data. This introduces controlled stochasticity that allows for the generation of the variations. Injecting noise at multiple scales enables the model to better capture spatial variability, overcoming the limitations of single-point noise injection. This noise injection makes it possible to get an ensemble of outputs from which the uncertainty in prediction can be estimated as discussed earlier. We took ensemble of 100 predictions to get the RMSE and uncertainty. The uncertainty is enhanced by a factor of 50 to make it visible on the plot since very low than RMSE, as it can be seen from the scatter plot in Figure 4.56. The low

uncertainty arises because the learned variance is very small, resulting in minimal differences between the original point estimates and those sampled from the narrow distribution. The Pearson's correlation coefficient for this is 0.61.

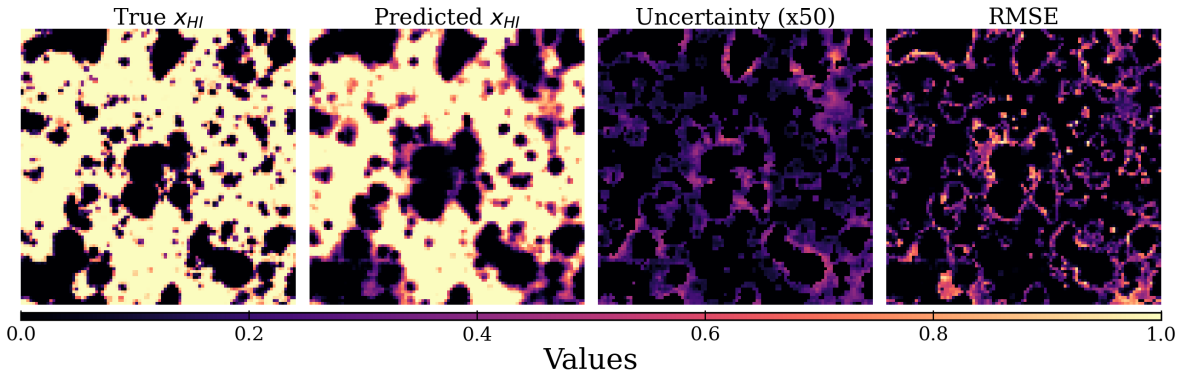


Figure 4.53: Results obtained using Gaussian Noise Injection Method.

4.9.3 Data Augmentation

The data augmentation, in terms of computer vision tasks, refers to artificially increasing the image data (2D or 3D) size by image manipulation techniques, such as zooming, shifting along an axis, flipping axes, and rotation along an axis [49, 50]. This method was used in [51] to estimate the uncertainty associated with the SegUNet model. The SegUNet model was used for segmentation of the noisy 21-cm maps into neutral and ionized regions. Here, we followed a similar approach where we utilized the 24 possible orientations of a cube. We transform the cube into these 24 orientations, make predictions, and do an inverse transformation. Afterwards, we calculate the prediction mean, uncertainty, and RMSE by comparing it with the ground truth as before. The scatter plot in Figure 4.57 shows that the uncertainty is low compared to the RMSE, hence enhanced by a factor of 2 to be visible in Figure 4.52. The Pearson's correlation coefficient for this is 0.83, which is significantly higher than the previous methods used.

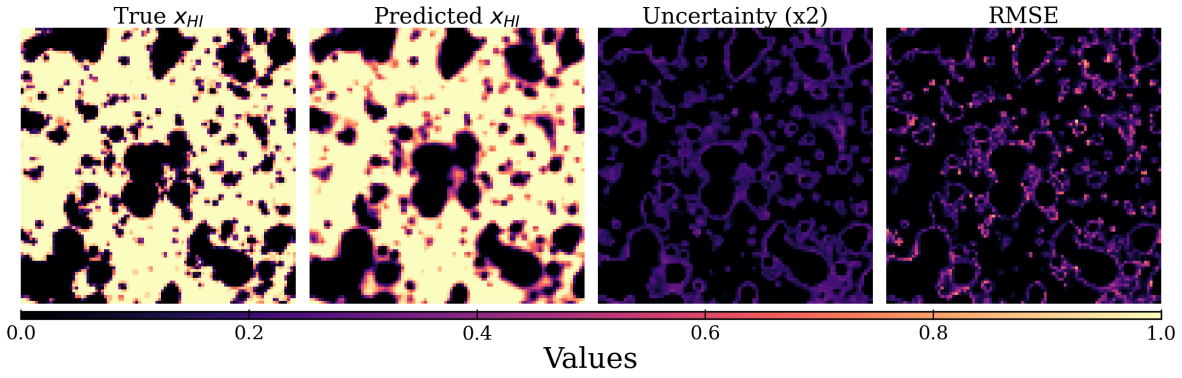


Figure 4.54: Results obtained using Data Augmentation Method.

Correlation plot of uncertainty and RMSE

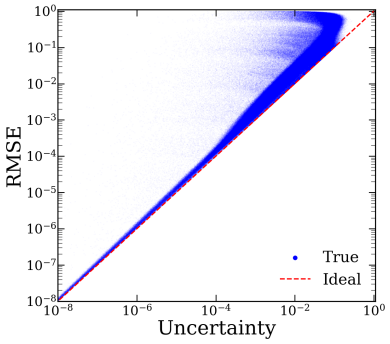


Figure 4.55: Monte-Carlo Dropout Method, PC:0.68.

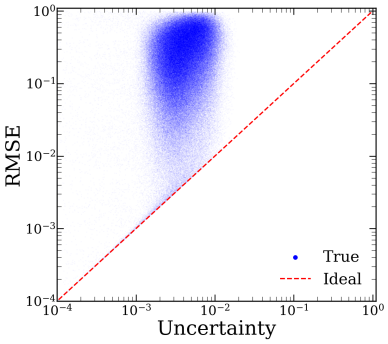


Figure 4.56: Gaussian Noise Injections Method, PC:0.61.

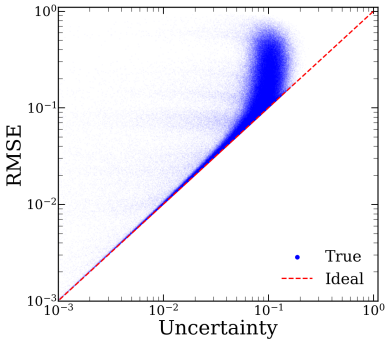


Figure 4.57: Data Augmentation Method, PC:0.83.

We are still exploring new methods of quantifying the uncertainty which will also helps us overcome the fuzzy boundary problem. Once quantified, these uncertainties can be propagated through the inference pipeline for more robust estimates of the reinoization parameters.

4.10 Model Generalization Test: CosmoUiT96

In Section 4.7, we saw that the model trained on low resolution struggles to make prediction on data generated via different random seed. Hence, we calculated the L_2 distance between the PDFs of the input fields. We observed that, when we downsampled the field from 384^3 to 48^3, the L_2 distance between resulting PDFs increased implying lesser similarity in the distribution after downsampling. Hence, we increased the resolution of the data and trained new model to learn the mapping between inputs and outputs.

4.10.1 Prediction for Unseen Random Seed

After training it on higher resolution, we repeated the same analysis for CosmoUiT96. The Figures 4.58 - 4.61 show emulation and simulation outputs with the difference between the two. This model gives better results than the CosmoUiT48. Though, the metrics do not seem significant, the model is able to capture the morphological features accurately without explicitly being trained for such variation. It suffices to say that we will get better results with lesser computational cost if we use this pretrained model for fine tuning over varying input fields generated via different random seeds [52].

4.10.2 Analysis of Generalization Ability

We compared the L_2 distances between the high resolution 384^3 and downsampled field 96^3. The Figures 4.62 - 4.63 the histograms with their titles containing the value of L_2 distance. Although the L_2 distance has increased but it is significantly lower than that for the fields with

Predictions made by CosmoUiT96 for unseen data

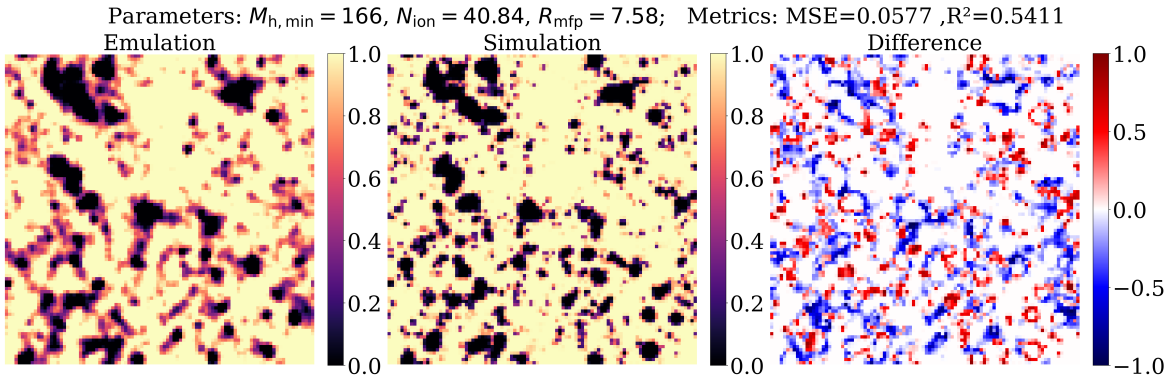


Figure 4.58: Comparison between x_{HI} fields for parameter set 1.

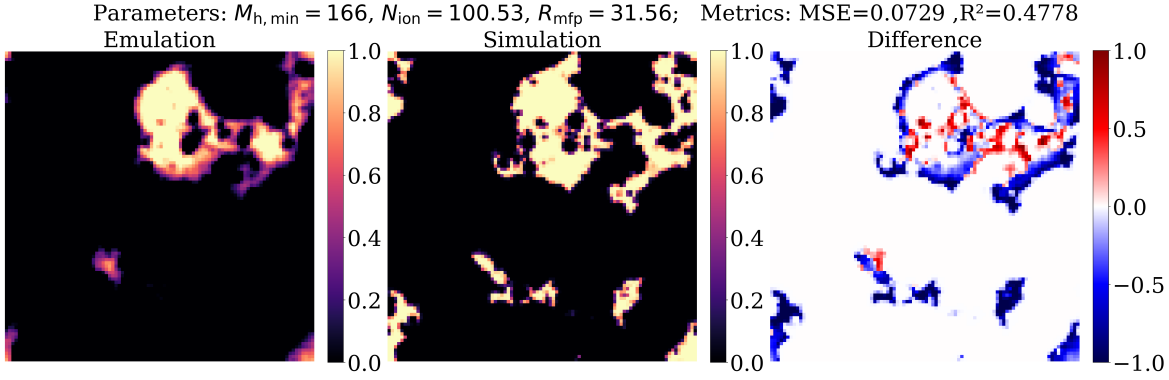


Figure 4.59: Comparison between x_{HI} fields for parameter set 2.

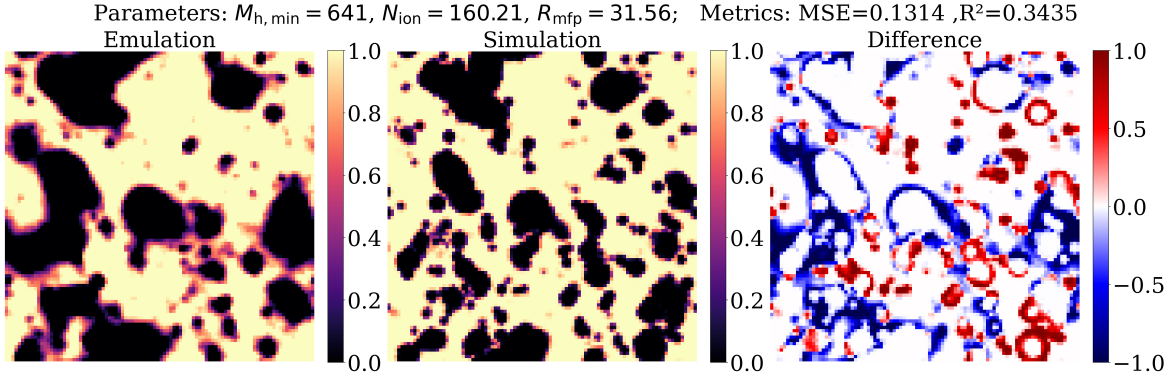


Figure 4.60: Comparison between x_{HI} fields for parameter set 3.

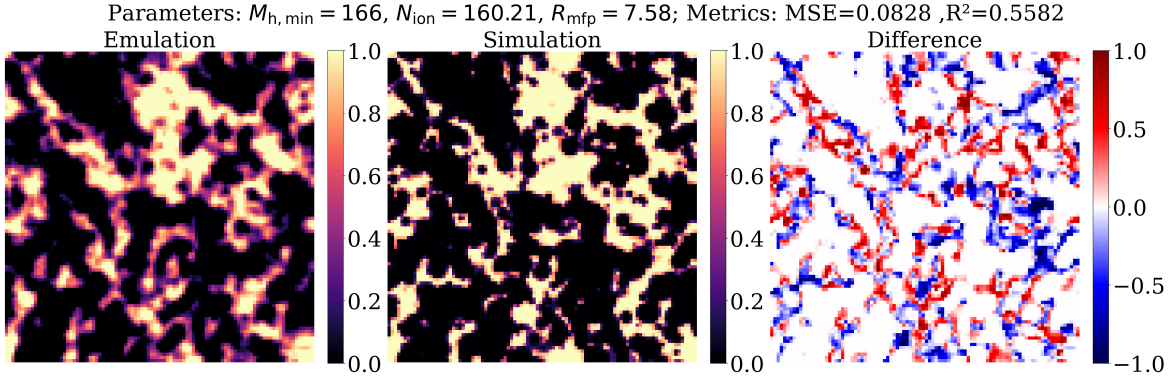


Figure 4.61: Comparison between x_{HI} fields for parameter set 4.

48^3 grid resolution. While CosmoUiT96 makes decent predictions one can go for even higher resolution where the L_2 distance is even smaller, however, this could be a computationally expensive process. Therefore, the ideal choice is to train the model with inputs generated via different random seeds. This will also help us to take the cosmic variance into account while doing the field-level inference.

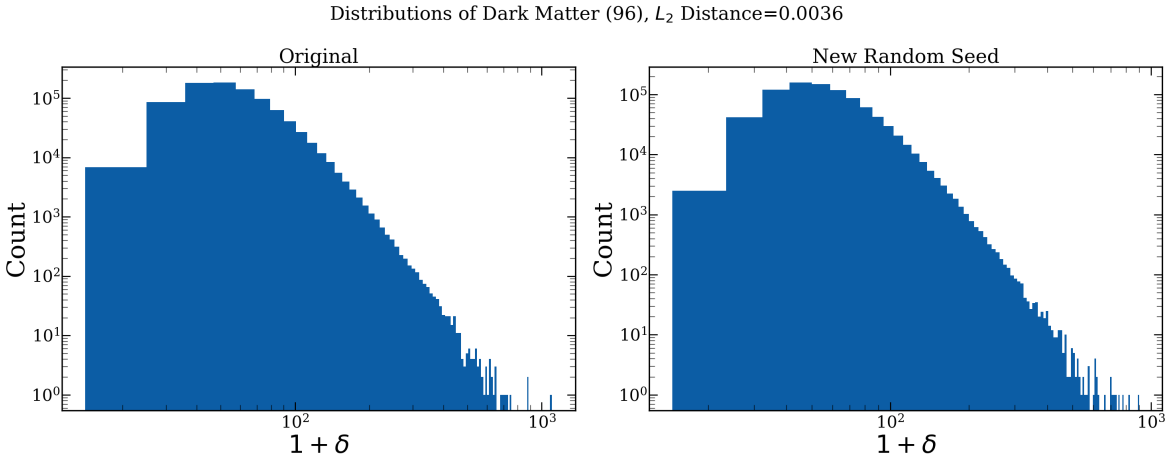


Figure 4.62: Comparison between distribution of dark matter for seen and unseen random seed (96^3 resolution).

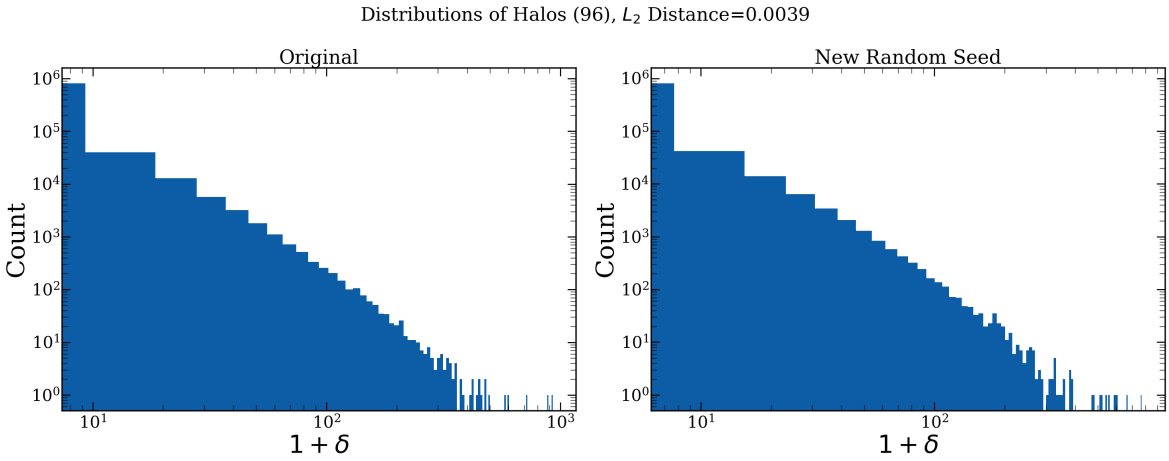


Figure 4.63: Comparison between distribution of halo for seen and unseen random seed (96^3 resolution).

CHAPTER 5

Summary and Future Plans

5.1 Summary

The primary objective of this thesis was to develop a deep learning-based emulator capable of bypassing the simulations required to model the 21-cm signal from EoR. These simulations are computationally expensive when it comes to re-running them while exploring the large EoR model parameter space for Bayesian inference. To address this, we explored a range of architectural strategies using the building blocks of deep learning algorithms. The core components of these architectures are convolution, deconvolution and transformer encoder layers. Our aim was to design a novel architecture that could learn the complex mapping between the underlying dark matter/halo density fields with the three EoR parameters, and the resulting 21-cm brightness temperature fields. While coming up with a suitable architecture, we also had to do extensive experimentation with data representation, parameter conditioning, and computational trade-offs. During the project, we iteratively assessed model performance to strike a balance between accuracy, generalization, and computational efficiency. This iterative process ultimately led to the development of an effective and scalable emulator.

In order to capture the large scale dependencies, the first model we built contained transformer encoder layers to encode this information into the field. Later this field was mapped to the output field using residual connections and transpose convolution. In this architecture, we treated the patches as feature maps and added the three reionization parameters while upsampling the field. We called this model CosmoViT. The main drawback of the model was, that its outputs were not parameter-specific and it gave a generic field for all combinations of EoR parameters with significantly high MSE. The model, originally designed for image translation tasks, relies on variation in the input fields to learn meaningful mappings. However, in this case, the inputs remained fixed across different outputs, which limited the model's ability to effectively condition on the three additional parameters that influence the output field. We also saw that the high resolution requires extensive computational resources, which led us to decrease the field resolution during the initial stage of model development.

Given the inability of CosmoViT to produce the parameter-specific output we designed a

new model architecture. This architecture combines the ability of transformer encoder layers to capture large-scale dependencies with the ability of convolutional layers in modeling small-scale structures. To implement this, we integrate two model architectures: Vision Transformers, which encode the influence of the three reionization parameters within the input fields before mapping them to the output, and a UNet, which performs hierarchical feature extraction and reconstructs the output fields. This model gave us parameter-specific outputs. We called it CosmoUiT (UNet integrated Vision Transformer). We tuned the base CosmoUiT model by changing the number of input nodes of linear projection, depth of UNet architecture, and number of convolutional layers, which improved the model performance and gave us better MSE and R^2 score. It occupied less space on the GPU, which in turn provided space for a larger batch size. The CosmoUNet model was developed to test whether the transformer encoder layer contributes any additional information to the predictions. To ensure a fair comparison, the model specifications were kept the same as those of the base CosmoUiT architecture. This includes the number of layers in the linear projections, the size and number of feature maps, and the activation functions. The only difference was the removal of the transformer encoder layer. The fields were directly fed to the UNet for mapping them to the neutral fraction fields output. The results show that without the transformer encoder layer, the only UNet model tends to overfit and fails to provide parameter-specific outputs.

The CosmoUiT was a well-performing model, however, it struggled at the boundaries of the parameter space and showed poor generalization for unseen random seeds due to low resolution. Hence, we increased the resolution from 48^3 to 96^3. This came at the expense of increased computational cost. We tried to reduce the complexity by removing unnecessary layers. This model gave excellent predictions and the only sources of errors were the boundaries separating the neutral and ionized regions. We tried taking these errors into account via uncertainty estimation in model predictions. There are three methods we tried: (1) Monte-Carlo Dropout, (2) Gaussian Noise Injection and (3) Data Augmentation, out of which the third one performed better than the rest. We made predictions on inputs generated via entirely different random seeds than what the model is trained for. The CosmoUiT48 model struggled while CosmoUiT96 made decent predictions. The reason behind the former's inability to make an accurate prediction was the significant difference between the distribution of seen and unseen input fields after downsampling. We expect to get better results after training the model for such variations.

This thesis presents a robust deep learning framework for emulating HI 21-cm brightness temperature fields during the EoR. With its ability to rapidly produce parameter-specific outputs, the model demonstrates strong potential for field-level inference for upcoming SKA surveys.

5.2 Future Plans

The initial model successfully demonstrated the potential to emulate complex EoR fields, laying the groundwork for future enhancements. The following improvements can be considered as the future scope of this work:

5.2.1 Cosmic Variance

The current model takes the fixed dark matter and halo fields as inputs at a fixed redshift $z = 7$. We have shown in Section 4.10, that the model makes decent predictions for unseen random seeds without explicitly training for such variation. However, to take the cosmic variance into account while doing the field-level inference, we have to use different realizations of dark matter and halo field as input, and hence our next goal will be to train the model for such variation. It is expected that using pretrained model for further training will drastically reduce the computational expenses.

5.2.2 Uncertainty Estimation

The typical deep learning model gives point estimates of their predictions; however, since these models are statistical approximations, they have error associated with them. To address this we already tried three methods: (1) Monte Carlo Dropout (MC Dropout), (2) Gaussian Noise Injections, and (3) Data Augmentation. Out of which the third one gave us a decent correlation between RMSE and uncertainty. However, with that method, the RMSE consistently exceeds the model's uncertainty bounds, which is problematic as it indicates that the prediction error is not contained within the expected uncertainty limits. To address this, we will incorporate Bayesian layers into our model to enable uncertainty quantification. Unlike standard layers that learn fixed weight values, Bayesian layers learn distributions over weights, allowing us to generate predictions that include associated uncertainties. This uncertainty can be propagated through the inference pipeline for more robust estimates of parameters.

5.2.3 Lightcone Effect

The current emulator only works well for a fixed redshift. One can extend this approach to emulate the entire lightcone by training it using an autoregressive approach. In this approach, the model receives the input fields along with a slice of the predicted output from the earlier redshifts to inform predictions at subsequent redshifts. By iteratively feeding forward these outputs, the emulator can progressively generate the full lightcone across the desired redshift range. Using vision transformer for this task will help us preserve the correlation between each slice. By incorporating redshift evolution into the autogregressive framework, we will ensure

that the emulator generates lightcones that are consistent with both the local and large-scale structures in the universe at various epochs.

5.2.4 Redshift Space Distortion

In reality, the observed 21-cm signal is affected by peculiar velocities of the hydrogen clouds, which distort the signal in redshift space. This Redshift Space Distortion (RSD) introduces anisotropies along the line of sight and must be accounted for to make realistic predictions. To incorporate this effect, we plan to extend the emulator to predict the 21-cm brightness temperature field in redshift space. This will involve modifying the training data to include RSD-corrected brightness temperature maps, allowing the model to learn these anisotropic patterns. Incorporating RSD is essential for making the emulator compatible with observational data and enhancing its utility for field-level inference.

5.2.5 Observational Effects

Apart from the astrophysics of IGM and cosmology, the 21-cm signal is also affected by foregrounds, system noise and telescopic effects. Since this effect is significant we have to include it in our inference pipeline. However, instead of adding all the observational effects in the training data itself and trying to reproduce them, we will only add the effects that are irremovable or partially removable, such as foreground leakage into EoR window, thermal noise, ionospheric effects, etc. The other effects, such as spectrally smooth foregrounds, beam effects, calibration errors, etc. that can be modeled and directly subtracted from the observations.

5.2.6 Field-Level Inference

Once the emulator has been trained and validated, the next step will be to use it for field-level inference to constrain the reionization model parameters. This involves comparing the emulator's predictions against mock observations, which are simulated lightcone that incorporate all relevant observational effects. With the above mentioned capabilities, this emulator aims to play a crucial role in efficient field-level inference for upcoming SKA surveys.

References

[1] Jonathan R Pritchard and Abraham Loeb. "21 cm cosmology in the 21st century". In: *Reports on Progress in Physics* 75.8 (July 2012), p. 086901. ISSN: 1361-6633. DOI: 10.1088/0034-4885/75/8/086901. URL: http://dx.doi.org/10.1088/0034-4885/75/8/086901.

[2] Rennan Barkana and Abraham Loeb. "In the beginning: the first sources of light and the reionization of the universe". In: *Physics Reports* 349.2 (July 2001), pp. 125–238. ISSN: 0370-1573. DOI: 10.1016/s0370-1573(01)00019-9. URL: http://dx.doi.org/10.1016/S0370-1573(01)00019-9.

[3] Ken'ichi Nomoto, Chiaki Kobayashi, and Nozomu Tominaga. "Nucleosynthesis of the Elements in Stars and Chemical Enrichment of Galaxies". In: *Annual Review of Astronomy and Astrophysics* 51 (Sept. 2012). DOI: 10.1146/annurev-astro-082812-140956.

[4] Shmuel Bialy and Amiel Sternberg. "Thermal Phases of the Neutral Atomic Interstellar Medium from Solar Metallicity to Primordial Gas". In: *The Astrophysical Journal* 881.2 (Aug. 2019), p. 160. ISSN: 1538-4357. DOI: 10.3847/1538-4357/ab2fd1. URL: http://dx.doi.org/10.3847/1538-4357/ab2fd1.

[5] Minsung Ko et al. *Understanding Stellar Mass-Metallicity and Size Relations in Simulated Ultra-Faint Dwarf Galaxies*. 2024. arXiv: 2411.14683 [astro-ph.GA]. URL: https://arxiv.org/abs/2411.14683.

[6] Arthur M. Wolfe, Eric Gawiser, and Jason X. Prochaska. "DAMPED LY SYSTEMS". In: *Annual Review of Astronomy and Astrophysics* 43.1 (Sept. 2005), pp. 861–918. ISSN: 1545-4282. DOI: 10.1146/annurev.astro.42.053102.133950. URL: http://dx.doi.org/10.1146/annurev.astro.42.053102.133950.

[7] R. Adam et al. "Planckintermediate results: XLVII.Planckconstraints on reionization history". In: *Astronomy amp; Astrophysics* 596 (Dec. 2016), A108. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201628897. URL: http://dx.doi.org/10.1051/0004-6361/201628897.

[8] T. Roy Choudhury. *Analytical Models of the Intergalactic Medium and Reionization*. 2009. arXiv: 0904.4596 [astro-ph.CO]. URL: https://arxiv.org/abs/0904.4596.

[9] J. Liske, J. K. Webb, and R. F. Carswell. "Large-scale structure in the Lyman-alpha forest: a new technique". In: *Monthly Notices of the Royal Astronomical Society* 301.3 (Dec. 1998), pp. 787–796. ISSN: 1365-2966. DOI: 10.1046/j.1365-8711.1998.02048.x. URL: http://dx.doi.org/10.1046/j.1365-8711.1998.02048.x.

[10] L. M. Griffiths, D. Barbosa, and A. R. Liddle. "Cosmic microwave background constraints on the epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 308.3 (Sept. 1999), pp. 854–862. ISSN: 1365-2966. DOI: 10.1046/j.1365-8711.1999.02777.x. URL: http://dx.doi.org/10.1046/j.1365-8711.1999.02777.x.

[11] Sangeeta Malhotra and James E. Rhoads. “Luminosity Functions of Ly Emitters at Redshifts z = 6.5 and z = 5.7: Evidence against Reionization at z 6.5”. In: *The Astrophysical Journal* 617.1 (Nov. 2004), pp. L5–L8. ISSN: 1538-4357. DOI: 10.1086/427182. URL: http://dx.doi.org/10.1086/427182.

[12] Mark Dijkstra. “Ly Emitting Galaxies as a Probe of Reionisation”. In: *Publications of the Astronomical Society of Australia* 31 (2014). ISSN: 1448-6083. DOI: 10.1017/pasa.2014.33. URL: http://dx.doi.org/10.1017/pasa.2014.33.

[13] Jordi Miralda-Escude. “Reionization of the Intergalactic Medium and the Damping Wing of the Gunn-Peterson Trough”. In: *The Astrophysical Journal* 501.1 (July 1998), pp. 15–22. ISSN: 1538-4357. DOI: 10.1086/305799. URL: http://dx.doi.org/10.1086/305799.

[14] Daniel J. Eisenstein et al. *Overview of the JWST Advanced Deep Extragalactic Survey (JADES)*. 2023. arXiv: 2306.02465 [astro-ph.GA]. URL: https://arxiv.org/abs/2306.02465.

[15] Gareth C Jones et al. “JADES: measuring reionization properties using Lyman-alpha emission”. In: *Monthly Notices of the Royal Astronomical Society* 536.3 (Nov. 2024), pp. 2355–2380. ISSN: 0035-8711. DOI: 10.1093/mnras/stae2670. eprint: https://academic.oup.com/mnras/article-pdf/536/3/2355/60927018/stae2670.pdf. URL: https://doi.org/10.1093/mnras/stae2670.

[16] H. C. Van De Hulst. “Origin of the Radio Waves from Space”. In: *Classics in Radio Astronomy*. Dordrecht: Springer Netherlands, 1982, pp. 302–316. ISBN: 978-94-009-7752-5. DOI: 10.1007/978-94-009-7752-5_34. URL: https://doi.org/10.1007/978-94-009-7752-5_34.

[17] T. Roy Choudhury, M. G. Haehnelt, and J. Regan. *Inside-out or Outside-in: The topology of reionization in the photon-starved regime suggested by Lyman-alpha forest data*. 2008. DOI: https://doi.org/10.1111/j.1365-2966.2008.14383.x. arXiv: 0806.1524 [astro-ph]. URL: https://arxiv.org/abs/0806.1524.

[18] Suman Majumdar et al. “On the use of seminumerical simulations in predicting the 21-cm signal from the epoch of reionization”. In: *Monthly Notices of the Royal Astronomical Society* 443.4 (Aug. 2014), pp. 2843–2861. ISSN: 0035-8711. DOI: 10.1093/mnras/stu1342. URL: http://dx.doi.org/10.1093/mnras/stu1342.

[19] Rajesh Mondal et al. “The effect of non-Gaussianity on error predictions for the Epoch of Reionization (EoR) 21-cm power spectrum”. In: *Monthly Notices of the Royal Astronomical Society: Letters* 449.1 (Feb. 2015), pp. L41–L45. ISSN: 1745-3925. DOI: 10.1093/mnrasl/slv015. URL: http://dx.doi.org/10.1093/mnrasl/slv015.

[20] Steven R. Furlanetto, Matias Zaldarriaga, and Lars Hernquist. “The Growth of HII Regions During Reionization”. In: *The Astrophysical Journal* 613.1 (Sept. 2004), pp. 1–15. ISSN: 1538-4357. DOI: 10.1086/423025. URL: http://dx.doi.org/10.1086/423025.

[21] Somnath Bharadwaj and Pennathur Sridharan Srikant. “HI fluctuations at large redshifts: III — Simulating the signal expected at GMRT”. In: *Journal of Astrophysics and Astronomy* 25.1–2 (Mar. 2004), pp. 67–80. ISSN: 0973-7758. DOI: 10.1007/bf02702289. URL: http://dx.doi.org/10.1007/BF02702289.

[22] C J Schmit and J R Pritchard. “Emulation of reionization simulations for Bayesian inference of astrophysics parameters using neural networks”. In: *Monthly Notices of the Royal Astronomical Society* 475.1 (Dec. 2017), pp. 1213–1223. ISSN: 1365-2966. DOI: 10.1093/mnras/stx3292. URL: http://dx.doi.org/10.1093/mnras/stx3292.

[23] Daniela Breitman et al. “<scp>21cmemu</scp>: an emulator of <scp>21cmfast</scp> summary observables”. In: *Monthly Notices of the Royal Astronomical Society* 527.4 (Dec. 2023), pp. 9833–9852. ISSN: 1365-2966. DOI: 10.1093/mnras/stad3849. URL: http://dx.doi.org/10.1093/mnras/stad3849.

[24] Sudipta Sikder et al. “Emulation of the cosmic dawn 21-cm power spectrum and classification of excess radio models using an artificial neural network”. In: *Monthly Notices of the Royal Astronomical Society* 527.4 (Nov. 2023), pp. 9977–9998. ISSN: 0035-8711. DOI: 10.1093/mnras/stad3699. eprint: https://academic.oup.com/mnras/article-pdf/527/4/9977/54919557/stad3699.pdf. URL: https://doi.org/10.1093/mnras/stad3699.

[25] Jonathan Chardin et al. “A deep learning model to emulate simulations of cosmic reionization”. In: *Monthly Notices of the Royal Astronomical Society* 490.1 (Sept. 2019), pp. 1055–1065. ISSN: 1365-2966. DOI: 10.1093/mnras/stz2605. URL: http://dx.doi.org/10.1093/mnras/stz2605.

[26] Damien Korber et al. “PINION: physics-informed neural network for accelerating radiative transfer simulations for cosmic reionization”. In: *Monthly Notices of the Royal Astronomical Society* 521.1 (Feb. 2023), pp. 902–915. ISSN: 1365-2966. DOI: 10.1093/mnras/stad615. URL: http://dx.doi.org/10.1093/mnras/stad615.

[27] Adrian E. Bayer, Uros Seljak, and Chirag Modi. *Field-Level Inference with Microcanonical Langevin Monte Carlo*. 2023. arXiv: 2307.09504 [astro-ph.CO]. URL: https://arxiv.org/abs/2307.09504.

[28] Leon Koopmans et al. “The Cosmic Dawn and Epoch of Reionisation with SKA”. In: *Proceedings of Advancing Astrophysics with the Square Kilometre Array — PoS(AASKA14)*. AASKA14. Sissa Medialab, May 2015. DOI: 10.22323/1.215.0001. URL: http://dx.doi.org/10.22323/1.215.0001.

[29] Daniel J. Eisenstein and Wayne Hu. “Baryonic Features in the Matter Transfer Function”. In: *The Astrophysical Journal* 496.2 (Apr. 1998), pp. 605–614. ISSN: 1538-4357. DOI: 10.1086/305424. URL: http://dx.doi.org/10.1086/305424.

[30] P. A. R. Ade et al. “Planck2013 results. XVI. Cosmological parameters”. In: *Astronomy amp; Astrophysics* 571 (Oct. 2014), A16. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201321591. URL: http://dx.doi.org/10.1051/0004-6361/201321591.

[31] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE]. URL: https://arxiv.org/abs/1511.08458.

[32] P. Hirling et al. “pyC2Ray: A flexible and GPU-accelerated radiative transfer framework for simulating the cosmic epoch of reionization”. In: *Astronomy and Computing* 48 (July 2024), p. 100861. ISSN: 2213-1337. DOI: 10.1016/j.ascom.2024.100861. URL: http://dx.doi.org/10.1016/j.ascom.2024.100861.

[33] Dominique Aubert, Nicolas Deparis, and Pierre Ocvirk. “EMMA: an adaptive mesh refinement cosmological simulation code with radiative transfer”. In: *Monthly Notices of the Royal Astronomical Society* 454.1 (Sept. 2015), pp. 1012–1037. ISSN: 1365-2966. DOI: 10.1093/mnras/stv1896. URL: http://dx.doi.org/10.1093/mnras/stv1896.

[34] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[35] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[36] Sajad Ahmadian and Ali Reza Khanteymoori. “Training back propagation neural networks using asexual reproduction optimization”. In: *2015 7th Conference on Information and Knowledge Technology (IKT)*. 2015, pp. 1–6. DOI: 10.1109/IKT.2015.7288738.

[37] Hengyuan Xu et al. *Permutation Equivariance of Transformers and Its Applications*. 2024. arXiv: 2304.07735 [cs.CR]. URL: https://arxiv.org/abs/2304.07735.

[38] Hemanth Saratchandran et al. *Rethinking Softmax: Self-Attention with Polynomial Activations*. 2024. arXiv: 2410.18613 [cs.LG]. URL: https://arxiv.org/abs/2410.18613.

[39] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[40] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG]. URL: https://arxiv.org/abs/1609.04747.

[41] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[42] Dami Choi et al. *On Empirical Comparisons of Optimizers for Deep Learning*. 2020. arXiv: 1910.05446 [cs.LG]. URL: https://arxiv.org/abs/1910.05446.

[43] Yiğit Gündüç. *Tensor-to-Image: Image-to-Image Translation with Vision Transformers*. 2021. arXiv: 2110.08037 [cs.CV]. URL: https://arxiv.org/abs/2110.08037.

[44] Cooper Jacobus et al. *A Gigaparsec-Scale Hydrodynamic Volume Reconstructed with Deep Learning*. 2024. arXiv: 2411.16920 [astro-ph.CO]. URL: https://arxiv.org/abs/2411.16920.

[45] Cooper Jacobus, Peter Harrington, and Zarija Lukić. *Reconstructing Lyman-α Fields from Low-Resolution Hydrodynamical Simulations with Deep Learning*. 2023. arXiv: 2308.02637 [astro-ph.CO]. URL: https://arxiv.org/abs/2308.02637.

[46] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML]. URL: https://arxiv.org/abs/1506.02142.

[47] Ronald Seoh. "Qualitative Analysis of Monte Carlo Dropout". In: *arXiv e-prints*, arXiv:2007.01720 (July 2020), arXiv:2007.01720. DOI: 10.48550/arXiv.2007.01720. arXiv: 2007.01720 [stat.ML].

[48] Xueqiong Yuan, Jipeng Li, and Ercan Engin Kuruoglu. *Uncertainty Quantification With Noise Injection in Neural Networks: A Bayesian Perspective*. 2025. arXiv: 2501.12314 [stat.ML]. URL: https://arxiv.org/abs/2501.12314.

[49] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

[50] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV]. URL: https://arxiv.org/abs/1512.00567.

[51] Michele Bianco et al. "Deep learning approach for identification of HII regions during reionization in 21-cm observations". In: *Monthly Notices of the Royal Astronomical Society* 505.3 (May 2021), pp. 3982–3997. ISSN: 1365-2966. DOI: 10.1093/mnras/stab1518. URL: http://dx.doi.org/10.1093/mnras/stab1518.

[52] Nishant Jain et al. *Efficiently Robustify Pre-trained Models*. 2023. arXiv: 2309.07499 [cs.CV]. URL: https://arxiv.org/abs/2309.07499.