UNIVERSITY *of the*
WESTERN CAPE

UNIVERSITY OF THE WESTERN CAPE

MASTERS THESIS

Accelerating Reionization Simulations Using Machine Learning

Author:
Mosima Portia Masipa

Supervisor:
Dr. Sultan Hassan
Prof. Mario G. Santos

A thesis submitted in fulfillment of the requirements

for the degree of Masters of Science

in the

Center for Radio Cosmology

Department of Physics and Astronomy

Declaration of Authorship

I, Mosima Portia Masipa, declare that this thesis titled, “Accelerating Reionization Simulations Using Machine Learning” and the work presented in it is my own. I confirm that:

- A paper titled, “ Emulating Radiation Transport on Cosmological Scales Using a Denoising Unet", based on the results presented in Chapter 4, Section 4.2 has been published as a conference paper at the International Conference on Learning Representations (ICLR) 2023.
- This work has been done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 11 July 2023

"*Perfection is not attainable, but if we chase perfection we can catch excellence"*

Vince Lombardi

Abstract

Epoch of Reionization (EoR) refers to the time in the history of the universe when the appearance of the first luminous sources reionized the intergalactic medium (IGM). The EoR carries a wealth of information regarding structure formation and evolution. Ongoing and planned 21cm experiments such as the Hydrogen Epoch of Reionization Array (HERA) and the Square Kilometre Array (SKA) are expected to generate huge amounts of high dimensional datasets, and hence a new generation of efficient simulations and tools are required in order to maximize their scientific return. While Convolutional neural networks (CNNs) achieve the state-of-the-art performance to extract information from large scale fields, generating large training datasets and fully exploring the cosmological and astrophysical parameter space require fast simulations. Semi-numerical simulations are the leading candidates to evolve reionization due to their simplicity and efficiency but they are too slow to enable field level inference. In this thesis, we assess the viability of several generative models and techniques to accelerate a semi-numerical model of reionization, SimFast21. In particular, we focus on generating the ionization fields (highly non-linear fields) *directly* from the initial density fields (highly smoothed fields) without using the ionizing sources locations, and hence emulating the radiative transfer process on cosmological scales. We show that a probabilistic (denoising) U-Net outperforms other deterministic approaches using either a standard autoencoder or a basic U-Net, and achieve high accuracy with a factor of 1,000 faster than the target simulation. This work represents a step forward towards efficiently generating large scale ionization maps, and hence maximizing the scientific return of future reionization surveys.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr. Sultan Hassan and Prof. Mario G. Santos for their continued support, patience, and motivation. I would also like to thank the National Astrophysics and Space Science Programme (NASSP) co-directors, Prof Saalih Allie and Prof Sarah Blyth. Many thanks to the National Research Foundation (NRF), NASSP, and the Centre for Radio Cosmology (CRC) for their financial support.

I would also like to thank the organizers of the Machine Learning Summer School 2022 at the Flatiron Institute (Simons Foundation) for the experience that contributed significantly to my research. To my mentors; Sultan Hassan, Kyunghyun Cho, and Gabriella Contardo, I cannot thank you enough. I am very grateful for the time you took out of your busy schedules to help me with my coding and other aspects of my research. I am eternally grateful.

To my family, where do I even begin? To my mother, I am thankful for your prayers, motivation, and those late-night calls when things were not working out. To my siblings, Mathapelo, Sello, and Kamogelo thank you for being constant reminders of what is important. To my sister Shibu, thank you for believing in me and supporting my academic journey, I owe a lot to you. My deepest gratitude goes to Dr. Zwidofhelangani Khangale for reading my thesis and providing constructive criticism and motivating me throughout. To my friends, housemates, and everyone else who was with me throughout this journey, **NDO LIVHUWA**.

Contents

List of Figures

List of Tables

List of Abbreviations

3D	3-Dimension
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BSD	Bubble Size Distribution
CMB	Cosmic Microwave Background
CNN	Convolutional Neural Network
CRADLE	Cosmological Reionization And Deep LEarning
CRC	Centre for Radio Cosmology
EGMRT	Expanded Giant Metrewave Radio Telescope
EoR	Epoch of Reionization
ESI	Echellette Spectrograph and Imager
FoF	Friends of Friends
GMRT	Giant Metrewave Radio Telescope
GP	Gunn-Peterson
GPU	Graphics Processing Unit
HERA	Hydrogen Epoch of Reionization Array
HI	Neutral Hydrogen
HMM	Hidden Markov Model
IGM	Intergalactic Medium
JWST	James Webb Space Telescope
kSZ	Kinetic Sunyaev-Zel'dovich
Lyα	Lyman alpha
Lyα	Lyman beta

MAE	Mean Absolute Error
mfp	mean free path
MRO	Murchison Radio-astronomy Observatory
MSE	Mean Squared Error
NASSP	National Astrophysics and Space Science Programme
NRF	National Research Foundation
NSF	National Science Foundation
PAPER	Precision Array for Probing the Epoch of Reionization
PCA	Principle Component Analysis
PS	Power Spectrum
ReLU	Rectified Linear Unit
RFI	Radio Frequency Interference
RMSE	Root Mean Squared Error
S/N	Signal-to-noise
SARAO	South African Radio Astronomy Observatory
SKA	Square Kilometre Array
SVM	Support Vector Machines
VAE	Variational Autoencoder
ΛCDM	Lambda Cold Dark Matter

1 Introduction

Studying the history and evolution of the Universe on large scale has been the core focus of Cosmology. The Big Bang Theory is the currently accepted model that explains the origin and evolution of the Universe (e.g. Spergel et al., 2003; Komatsu et al., 2011b; Popolo and Le Delliou, 2016). The flat Lambda Cold Dark Matter (ΛCDM), which is the parameterization of the Big Bang model, proposes a combination of dark matter (~ 25%), dark energy (~70%), and a small fraction of baryonic matter (e.g. Spergel et al., 2003; Năstase, 2019; Turner, 2021). Several observations support the model including the expansion of the Universe (Hubble, 1929; Pierre and Pain, 2012; Gron, 2018); the abundance of light elements (H, He, Li; e.g. Fields, Molaro, and Sarkar, 2014; Coc, Uzan, and Vangioni, 2014); the Cosmic Microwave Background (CMB, Penzias and Wilson, 1965; Komatsu et al., 2011a; Ade et al., 2015), and the formation of galaxies and large-scale structures (Blumenthal et al., 1984; Coil, 2013; De Zotti et al., 2019).

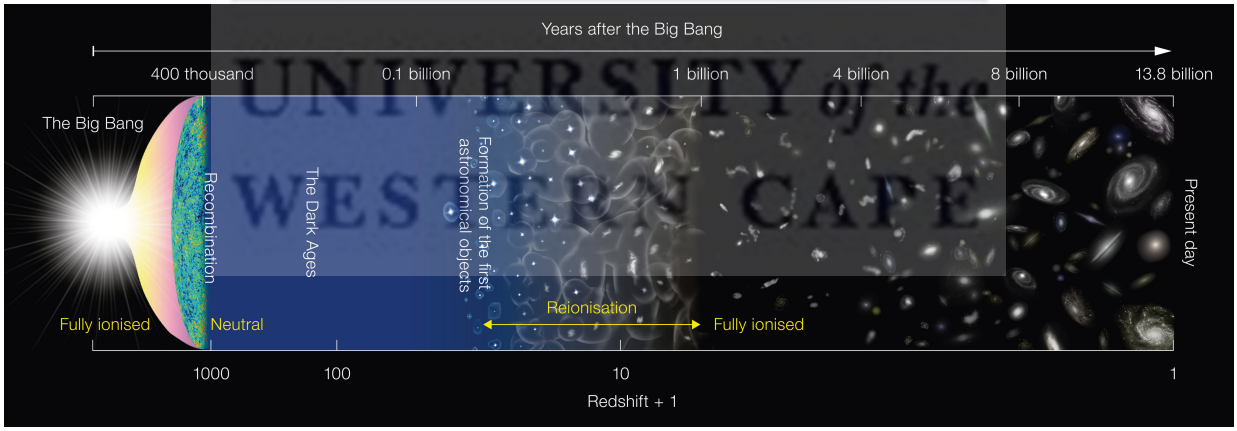

FIGURE 1.1: History of the Universe: The image shows the Universe's timeline, from the Big Bang until the present-day Universe. Credit: European Southern Observatory (2016).

Current data indicates that the Universe formed about formed 13.8 billion years ago. The Big Bang model suggests that it originated from an infinitely dense and hot ball that inflated and expanded; thus, the Universe's evolution started immediately after the Big Bang. After this explosion, the Universe was extremely hot. As it continued to expand, it cooled down, allowing

for recombination. The recombination era happened when the protons and electrons combined to form neutral atoms. Recombination led to the Dark Ages, a period where there were no sources of light. The first sources of light formed during the cosmic dawn and they ionized the intergalactic medium (IGM). The reionization process of the IGM is called the Epoch of Reionization (EoR), the time in the history of the Universe that is explored in this thesis. As the Universe continued to evolve, galaxies, galaxies clusters, and other astrophysical objects observed in the Universe today formed. Figure 1.1 shows the evolution of the Universe, from the Big Bang to the present Universe.

The history of the Universe has two major phase transitions of hydrogen; emission of the CMB and reionization. Based on previous studies, we currently have a good understanding of the CMB. However, reionization remains poorly constrained. CMB, which is the radiation that remained after matter decoupled in the early Universe, gives insight into the idea that the large-scale structure that we observe in the present Universe was a result of small-amplitude density fluctuations (Bennett et al., 1996; Bernardis et al., 2000; Barkana and Loeb, 2001). As the Universe expands, dense regions gravitationally collapsed to form galaxies and clusters. Throughout cosmic time, The Universe originated from a simple state of matter that can be simply described with linear physics and evolved to a more complex and highly non-linear state. Therefore, studying the earliest stages of the Universe, including the EoR, provides a deeper insight on the present-day Universe.

In Astro2010 (National Research Council, 2010) and Astro2020 (Savin et al., 2019; National Academies of Sciences, Engineering, and Medicine (2021)), the National Science Foundation (NSF) decadal committee recommended that studying the early galaxies, stars, and blackholes be a priority. The recommendation made studying EoR one of the leading topics in Astrophysics and Cosmology. There are several key fundamental questions regarding the nature of EoR we want to answer. These include but are not limited to: When did reionization begin and end (EoR timing)? Which sources contributed most to reionization (nature of sources)? What are the size and distribution of the ionized bubbles (EoR morphology)?

1.1 Reionizing the Intergalactic medium

When the first luminous objects appeared/formed, they emitted ultraviolet radiation. This process lead to the IGM being ionized. IGM refers to matter such as dust and hot hydrogen gas that exists between galaxies. Reionizing the IGM happens in three distinctive steps/stages (Gnedin, 2000; Ferrara and Pandolfi, 2014). The initial stage, referred to as the “pre-overlap" phase, happens when the individual ionizing sources turn on and ionize their surroundings. The IGM becomes a 2-phased medium, where one region that is highly ionized is separated from the neutral region by ionization fronts. The second stage, which is termed the “overlap", phase happens rapidly. This is when the neighboring HII regions begin to overlap. It is during this stage that the low-density IGM becomes highly ionized. Some of the neutral gas remains in highly dense structures. The final stage is the “post-overlap" stage. The highly dense regions from the neutral overlapping stage are gradually ionized as galaxy formation continues.

1.2 Constraints on reionization

Constraining the EoR is a challenge in Astrophysics and Cosmology. There are several observational constraints on reionization. For example, CMB surveys are targeting the early Universe and they are used as an indirect probe to reionization while high redshift quasars are used to probe the late Universe and can help to constrain the end of reionization. In this section, we discuss several observational constraints on reionization such as the Gunn-Peterson (GP) optical depth (Gunn and Peterson, 1965; Giallongo et al., 1994; Becker et al., 2001), HI fraction (e.g. McGreer, Mesinger, and D’Odorico, 2015; Mason et al., 2019), temperature, and polarisation anisotropies from the CMB (e.g. Bennett et al., 2012; Planck Collaboration et al., 2018), kinetic Sunyaev-Zel’dovich effect (kSZ, Zeldovich and Sunyaev, 1969; Sunyaev and Zeldovich, 1980), and the 21cm signal (e.g. Greig et al., 2021; Tiwari et al., 2021).

Most of the spectroscopic observations of high redshift quasars suggest that reionization was complete at a redshift of about 6 (e.g. Fan et al., 2006), with some studies questioning the validity of this conclusion (e.g. Mesinger, 2010). The Lyα absorption in the spectra of high-redshift quasars caused by the neutral hydrogen gives rise to the Gunn-Peterson trough, and this provides a marker to the end of reionization (e.g. Bégin, Liu, and Gorce, 2021). In Fan et al. (2006), a sample of about 19 quasars from the Sloan Digital Sky Survey (SDSS) that were at a

redshift range of $5.74 < z < 6.42$ was used to trace the IGM properties in order to constrain the EoR. They used the evolution of the Gunn-Peterson (GP) optical depth as one of their methods to study the properties of the IGM. Figure 1.2 shows the evolution of the optical depth (y-axis) as a function of redshift (x-axis) using combined measurements from the Lyman alpha (Lyα), beta (β), and gamma (γ) results. They found that the evolution of the ionized state of the IGM accelerated at z>5.7, and the GP optical depth changes from $\tau_{GP}^{eff} \sim (1+z)^{4.3}$ to $(1+z)^{>\sim 11}$.

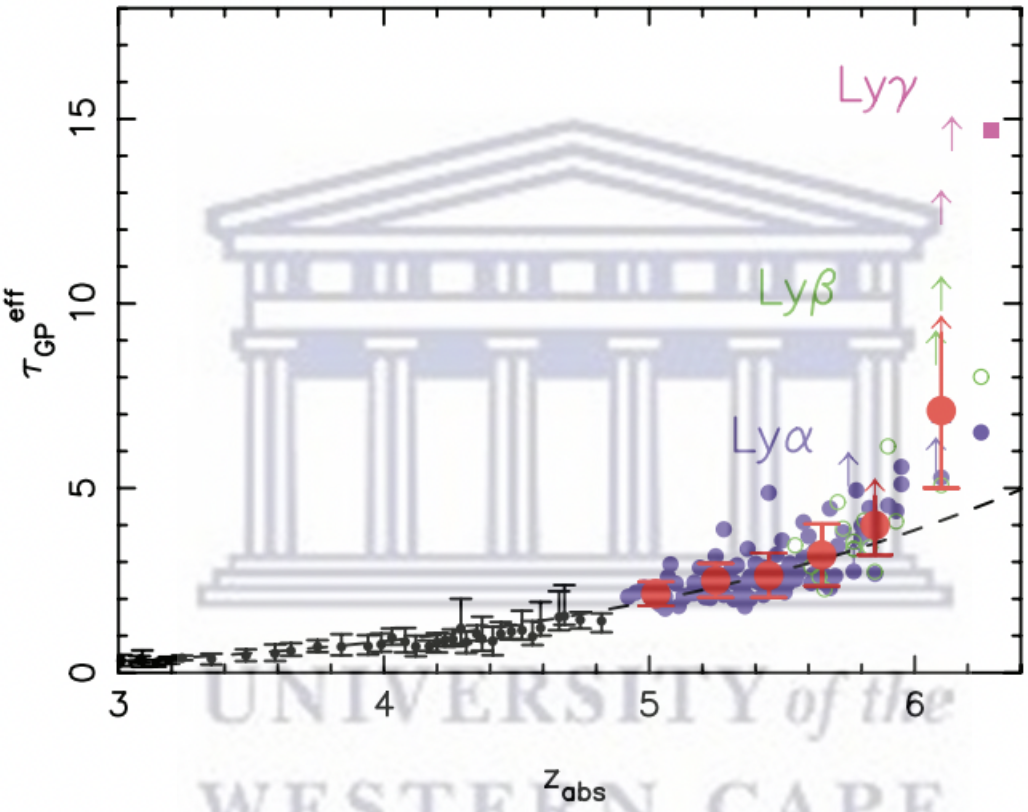

FIGURE 1.2: Evolution of the optical depth as a function of redshift. Credit: Fan et al. (2006).

Results from McGreer, Mesinger, and Fan (2011), who studied model-independent constraints on the HI fraction at redshifts of $z \sim 5-6$ are shown in Figure 1.3. The first work studies 13 spectra of high redshift quasars from the Keck Echellette Spectrograph and Imager (ESI) to place direct upper limits on the HI fraction at the mentioned redshifts. They have used the statistics of covering a fraction of dark pixels in the Lyα and Lyβ forest of these high redshifts quasars. They placed the stringiest constraints on the neutral hydrogen fraction of $x_{HI} \leq 0.5$ at z=7.1. With the subsequent work (McGreer, Mesinger, and D'Odorico, 2015), a sample of 22 quasars was used to study IGM properties, and they found that the IGM becomes highly ionized

as the first stars and galaxies formed and that reionization was almost complete at z~ 6. The significant amount of neutral hydrogen found at z=5.5 suggests reionization may have been a more extended process.

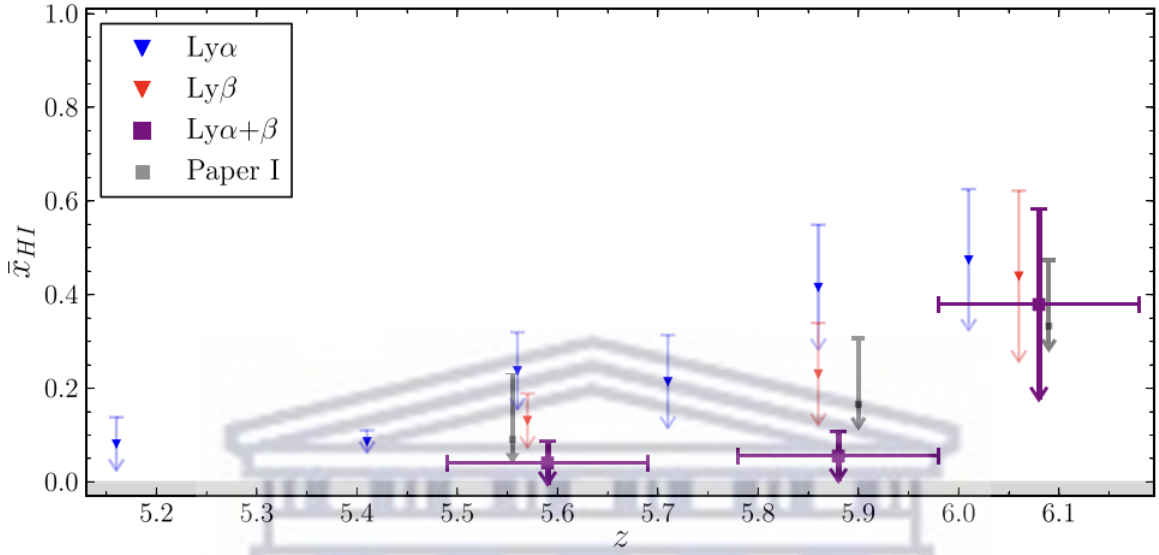

FIGURE 1.3: Evolution of HI as a function of redshift. Credit: McGreer, Mesinger, and D'Odorico (2015).

The Cosmic Microwave Background gives us information on the history of reionization through three processes, kSZ effect, polarisation anisotropies, and temperature anisotropies. These best constraints from CMB surveys come from the Wilkinson Microwave Anisotropy Probe (WMAP) and Planck Satellite. Both surveys provide the timing and nature of reionization using CMB data. These surveys use temperature and polarisation anisotropies to infer the optical depth. This is the optical depth of free electrons to the CMB, and the presence of free electrons increases scattering with CMB photons, affecting intensity and polarization anisotropies. WMAP had several data releases over the years (eg. (WMAP1, Bennett et al., 2003); WMAP third-year data release (WMAP3, Hinshaw et al., 2006; Spergel et al., 2007), WMAP fifth-year data release (WMAP5, Komatsu et al., 2008), WMAP seven-year data release (WMAP7, Komatsu et al., 2010), and WMAP nine-year data release (WMAP9, Bennett et al., 2012). With more follow-up and refinement, the optical depth has changed from the initial measurements of $\tau = 0.17 \pm 0.04$ to the more recent value of $\tau = 0.089 \pm 0.014$. This high value indicated that reionization happened earlier in the history of the Universe and most of the hydrogen in the Universe was ionized by 550 million years after Big Bang. This work concluded that the first galaxies might

have not been sources responsible for ionization and that quasars might have played a larger role in ionizing the IGM.

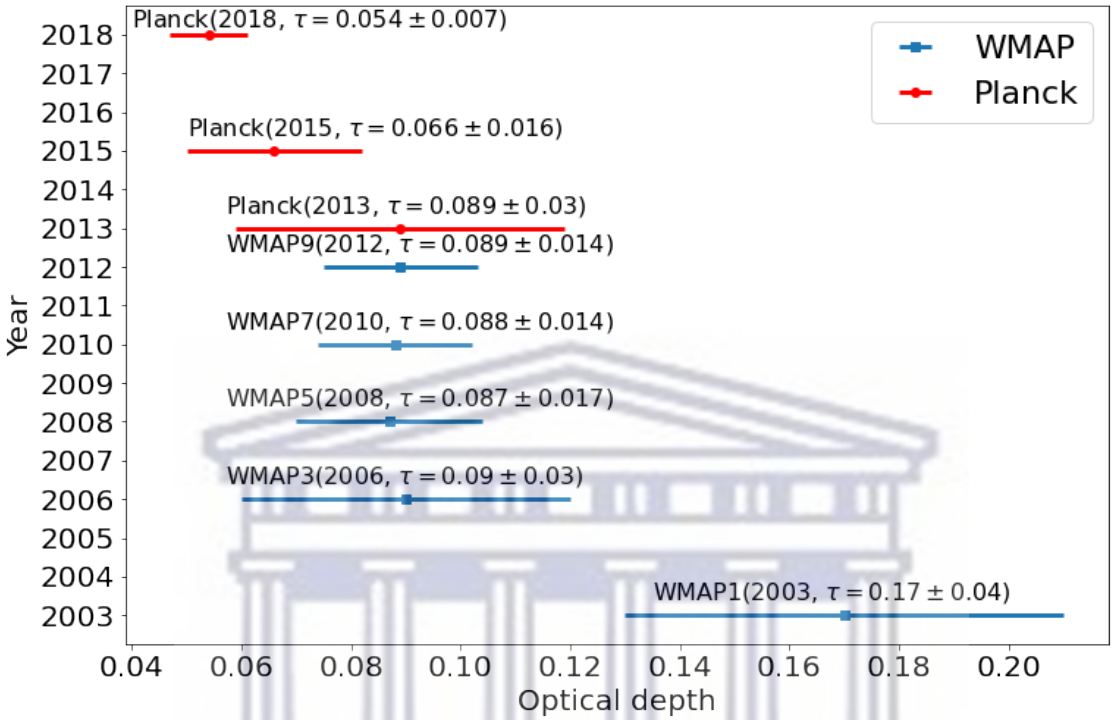

FIGURE 1.4: Optical depth comparison among different studies using WMAP and Planck data.

The Planck collaboration reported a lower value of the optical depth than WMAP. The measurements from Planck Collaboration are consistent with the WMAP measurements. The optical depth measurements from the Plank collaboration were published in several papers (Planck Collaboration et al., 2013; Planck Collaboration et al., 2015; Planck Collaboration et al. (2018)). With more follow-up and improvement in their analysis, the optical depth changed from $\tau = 0.089 \pm 0.03$ to $\tau = 0.054 \pm 0.007$. The low value of τ indicated that reionization happened later. They also concluded that reionization was a quick process that lasted a few million years and it was driven by the earliest galaxies. Figure 1.4 shows the optical depth comparison from different data releases over the years for WMAP and Planck Collaboration. The decrease in the error bars for individual measurements shows that in both studies, there was an improvement in the data, their analysis method, and the instrumentation.

Another probe that can be used to constrain reionization is kSZ from CMB observations. This

probe is sensitive to reionization as it arises from interactions of CMB photons with energetic free electrons produced during EoR (Mesinger, McQuinn, and Spergel, 2011; Park et al., 2013; Battaglia et al., 2013; Choudhury, Mukherjee, and Paul, 2020).

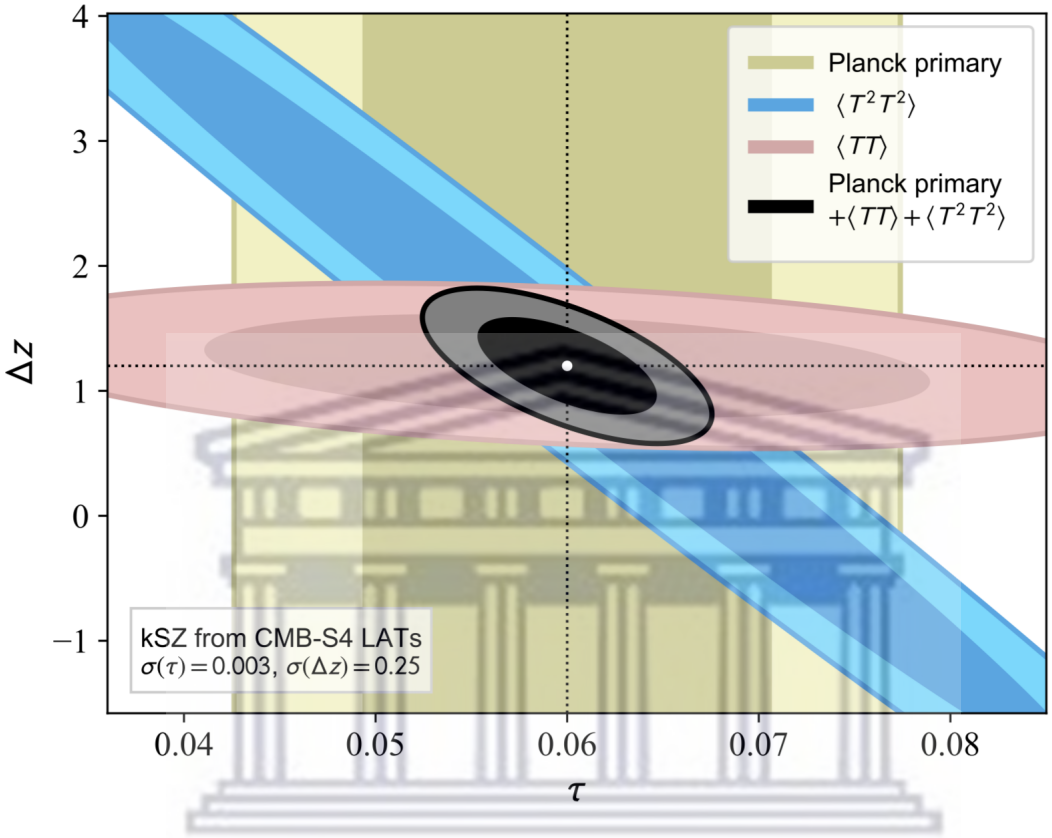

FIGURE 1.5: Constraining reionization using the 2-point function, four-point function, and the Planck primary CMB constraints on τ, yields tighter constraints on τ and Δz. Credit: Alvarez et al. (2020).

EoR represents a strong contaminant to the CMB-derived cosmological constraints. This is due to the degeneracy between the Thompson scattering optical depth, τ, and the amplitude of scalar perturbations, A_s. To mitigate this degeneracy, Alvarez et al. (2020) have used the kSZ effect. The kSZ power spectrum (PS) has a strong dependency on physical reionization parameters, the optical depth (τ), and the duration of reionization (Δz). Using the kSZ two-point function, reconstructed four-point function, and the Planck primary CMB constraints on τ, the degeneracy between τ and Δz can be broken.

In Figure 1.5, the two-point function, shown in pink, weakly constrains the optical depth (τ) but tightly constrains the duration of reionization (Δz). However, the reconstructed four-point function, shown in blue, weakly constrains the duration of reionization (Δz) but tightly constrains the optical depth (τ). Both τ and Δz are tightly constrained when all the probes are

combined; the black oval shows this. The tight constraints on both τ and Δz are $\sigma(\tau) = 0.003$ and $\sigma(\Delta z) = 0.25$, respectively.

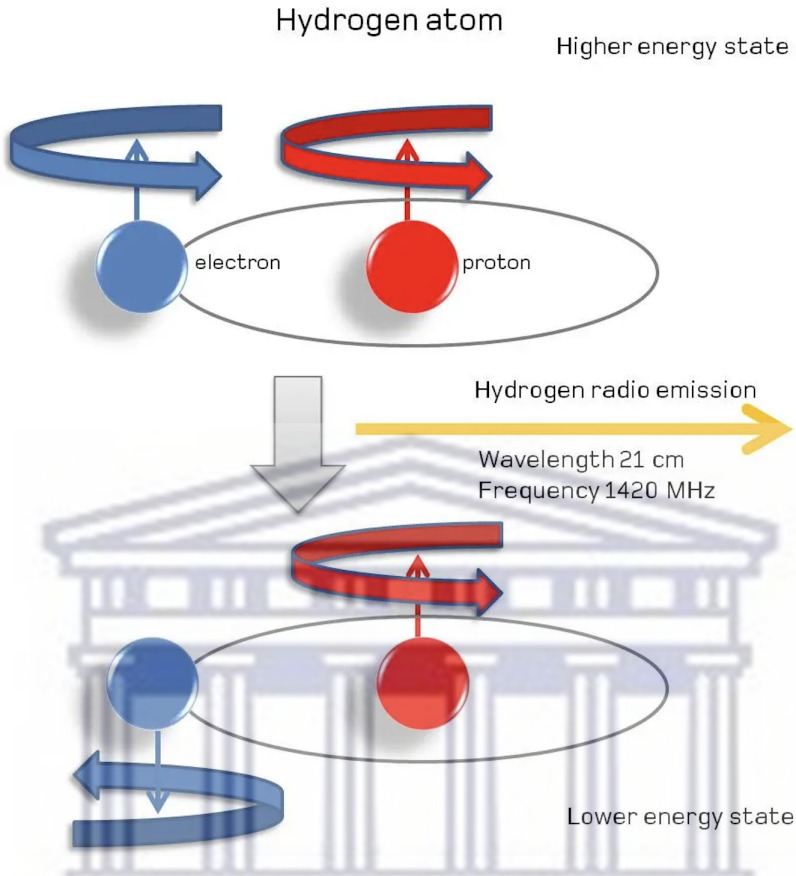

FIGURE 1.6: The 21cm hyperfine line transition of HI. Credit: Siegel (2022)

Strong constraints are expected to come from the 21cm experiments. The 21cm line from neutral hydrogen (HI) can probe the distribution of neutral hydrogen in the Universe and is, therefore, a great way to probe the EoR (Furlanetto, Oh, and Briggs, 2006). This line corresponds to the spin-flip transition of the neutral hydrogen hyperfine line split at the ground level, due to the magnetic dipole. When the spins of the electron and proton are flipped (facing opposite directions), the hydrogen atom is in a slightly lower energy state than when they are aligned. The low energy corresponds to a wavelength of 21cm and the transition is highly forbidden. Figure 1.6 shows the hyperfine line transition of HI.

The observable of this 21cm line is its brightness temperature and this is given by the equation below adapted from Furlanetto, 2006.

$$\delta T_b =\sim 27 x_{HI}(1+\delta_m)\left(\frac{T_s - T_\gamma}{T_s}\right)\left(\frac{1+z}{10}\frac{0.15}{\Omega_m h^2}\right)^2\left(\frac{\Omega_b h^2}{0.023}\right)[\text{mK}], \tag{1.1}$$

where δ is the matter density contrast, x_{HI} is the HI fraction, Ω_m and Ω_b are the matter and baryon densities in the units of critical densities, h is the Hubble constant and T_s is the spin temperature. The 21cm PS can be related to the brightness temperature and it is given by the following equation from Villaescusa-Navarro et al. (2018).

$$P_{21}(k,\mu) = \bar{T}_b^2 \left[(b_{HI} + f\mu^2)^2 + P_m(k) + P_{SN} \right]. \tag{1.2}$$

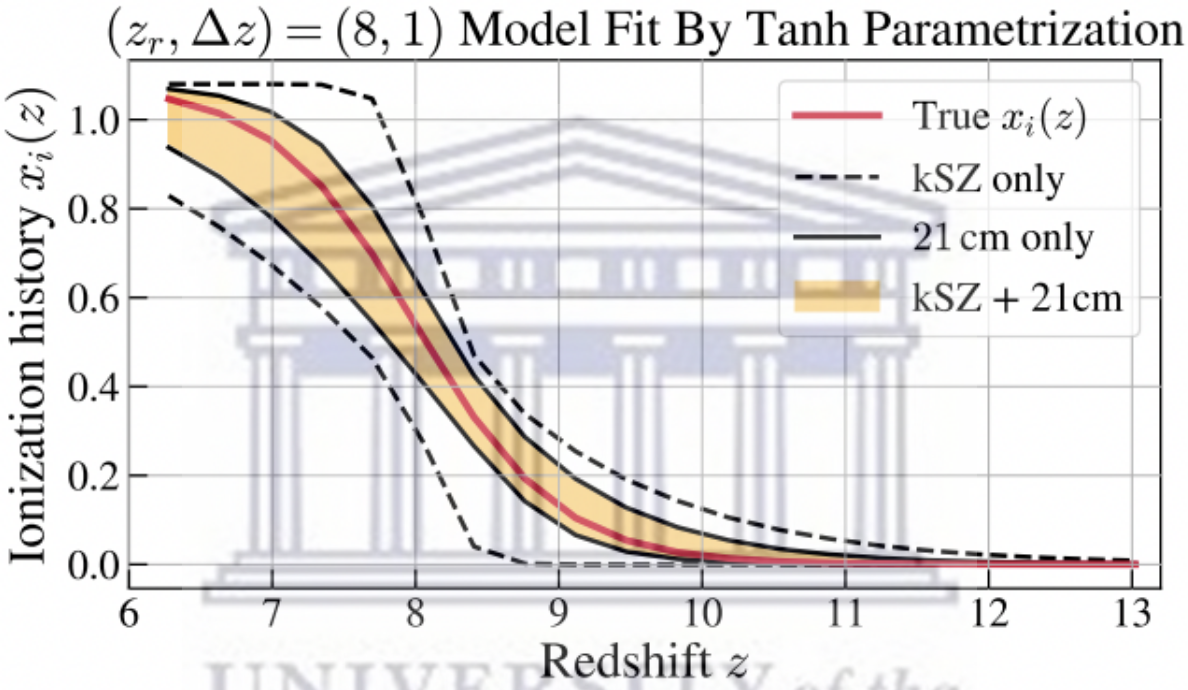

FIGURE 1.7: Reionization history as a function of redshift. Credit: Bégin, Liu, and Gorce (2021).

In Equation 1.2, $\bar{T}_b$ is the mean brightness temperature, b_{HI} is the HI bias, f is the linear growth rate, $\mu = k_z/k$ where k_z is the projection of k along the line of sight, P_m is the linear matter PS, and P_{SN} is the HI shot noise.

History of reionization can be constrained using either the 21cm cosmological signal or the kSZ effect. Combining the 21cm global signal and the kSZ effect gives more precise measurements to constrain the history of reionization than when these probes are used individually (Bégin, Liu, and Gorce, 2021). The two probes complement each other since the 21cm signal is more sensitive to rapidly evolving ionization histories, and the kSZ is sensitive to extended reionization histories. Figure 1.7 shows the ionization history as a function of redshift when

these probes are used together. The true model was compiled using a Bayesian approach, assuming the model is given by an asymmetric model fit by Tanh parametrization.

Given the importance of the 21cm to probe reionization, in the following section we will review some of the 21cm experiments.

1.3 Experiments targeting the 21cm signal

There are several experiments targeting the 21cm cosmological signal. Here we discuss some of the leading facilities and their contribution to EoR studies. These experiments include the Giant Metrewave Radio Telescope (GMRT, Swarup et al., 1996; Paciga et al., 2013); the Murchison Widefield Array (MWA; Tingay et al., 2013; Bowman et al., 2013); LOw-Frequency Array (LOFAR; Haarlem et al., 2013); Hydrogen Epoch of Reionization Array (HERA, DeBoer et al., 2017), and the Square Kilometre Array (SKA; Mellema et al., 2013; Koopmans et al., 2015)

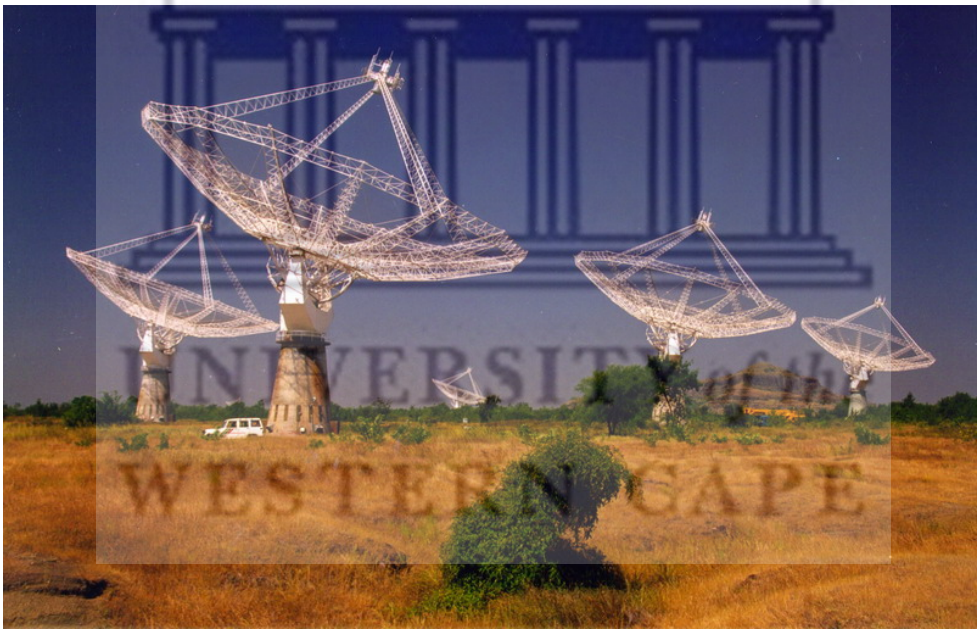

FIGURE 1.8: Several GMRT parabolic dishes. Credit: National Center for Radio Astrophysics-Tata Institute of Fundamental Research (NCRA-TIFR) (1999).

GMRT is a radio telescope with 30 parabolic dishes that are 45 metres in diameter. This telescope is situated in Khodad village, near Pune in India. Only 12 of the 30 dishes serve as the central array while the remaining 18 are placed along three 14km long arms. Figure 1.8 shows several of the GMRT dishes. It observes six frequency bands, namely 38, 153, 233, 327, 611, and 1420 MHz. The primary goal of this telescope is to search for the redshifted 21cm line from HI primordial clouds. Recently, they have developed a way to improve the sensitivity

and resolution of this telescope. The new proposed project, the Expanded Giant Metrewave Radio Telescope (EGMRT; Patra et al., 2019), plans to add 30 new antennas that are at a short distance of about 2.5km from the array centre to the existing 30 dishes of the GMRT, with 26 more at long distances ($\sim 5 - 25km$ from the array centre). The 2σ upper limits from GMRT were published in Paciga et al. (2013), at $(248\ m\mathrm{K})^2$ for $k = 0.50h\mathrm{Mpc}^{-1}$ and $z \approx 8.6$.

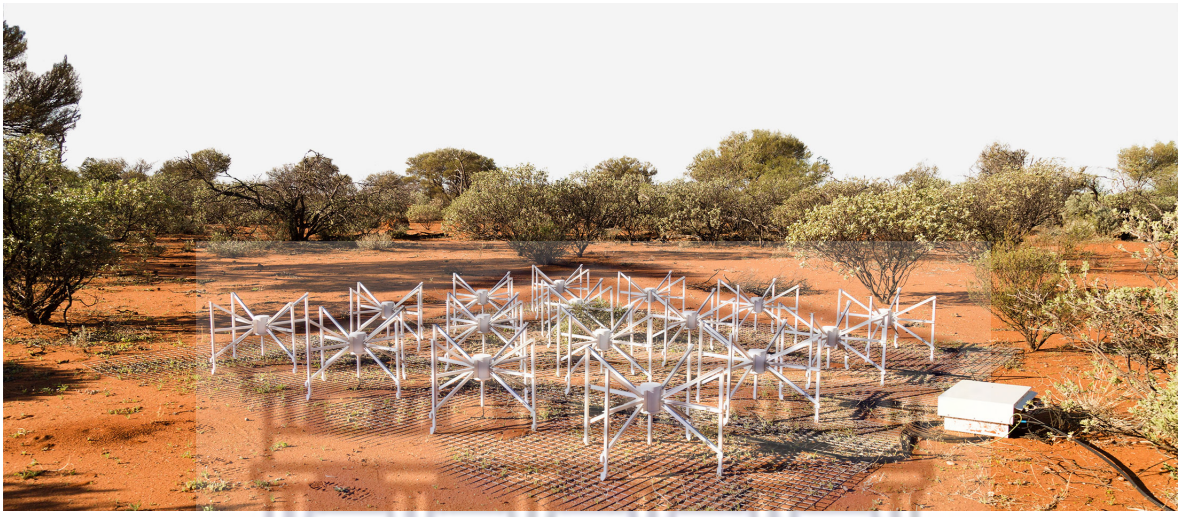

FIGURE 1.9: The Murchison Widefield Array. Credit: Murchison Widefield Array (MWA) (2009)

MWA is a radio interferometer at Western Australia's Murchison Radio-astronomy Observatory (MRO), which began operations in 2013. The location of MWA is ideal for the science it performs; the area has low levels of human-made radio frequency interference (RFI). A scientific goal of this radio telescope is to search for the redshifted 21cm emission from the EoR. It has an arc minute angular resolution which is sufficient to probe the typical ionized bubbles during EoR (Bowman et al., 2013). It is a precursor instrument to the SKA in the low frequency and observes in the 80-300 MHz range. Figure 1.9 shows a layout for the array, which has spider-like antennas. Using the 21-hour Phase I MWA data, Barry et al. (2019) improved on the upper limits placed on the PS reported in Beardsley et al. (2016). The authors reported a noise-dominated upper limit of $\Delta^2 \leq 3.9 \times 10^3$ mK^2 at z=7 and $k \approx 0.2\ h$ Mpc^{-1} while in Li et al. (2019) placed 2σ upper limits on the PS of $\Delta^2 \leq 2.39 \times 10^3$ mK^2 at z=6.5 and $k \approx 0.59\ h$ Mpc^{-1} using 40-hour data from Phase II of MWA.

LOFAR is a radio interferometer with 24 core stations that are distributed over an area of about 2km in diameter. There are 14 remote interferometer stations that are distributed all over the Netherlands, providing a baseline length of $\sim 100km$. Figure 1.10 shows the layout of the core

area of this array. The first upper limits on the PS from LOFAR were published in Patil et al. (2017) using 13-hour data. The authors reported a 2σ upper limit of $\Delta^2_{21} < (79.6)^2$ mK^2 at $z = 10.1$ and $k = 0.053\ h$ cMpc^{-1}. The deepest upper limits for LOFAR were reported by Mertens et al. (2020) using 141 hours of data collected over 12 nights of observations from LOFAR cycles 0,1 and 2. The improved upper limits were $\Delta^2_{21} < (72.8)^2$ mK^2 at a redshift $z \approx 9.1$ and $k = 0.075\ h$ cMPc^{-1}.

FIGURE 1.10: The layout for the core area of the LOw-Frequency Array. Credit: Haarlem et al. (2013).

HERA is a facility that is dedicated to measuring the 21cm line emission from the IGM. It is a 350-element interferometer that consists of 14m parabolic dishes, situated in Carnarvon, a small town in the Northern Cape of South Africa. This radio interferometer telescope observes at a 50-250 MHz frequency range. The scientific goal for HERA is to characterize the evolution of the 21cm PS, which will help constrain the timeline and morphology of the EoR. HERA is a precursor instrument to SKA and the measurements from HERA can be used in conjunction with semi-analytical models to constrain the ionization history. Figure 1.11 shows 19 HERA 14m dishes. The first results from HERA phase I (Abdurashidova et al., 2022) placed upper limits of $\Delta^2_{21} \leq (30.76)^2$ mK^2 at $k = 0.192\ h$ Mpc^{-1} and z=7.9 on the PS with 95% confidence using 18 nights of observational data. Recent and most sensitive upper limits on the 21cm PS

from HERA were published in (The HERA Collaboration et al., 2022). These upper limits were of $\Delta^2(k = 0.34h\text{Mpc}^{-1}) \leq 457\text{mK}^2$ at z=7.9 and $\Delta^2(k = 0.36h\text{Mpc}^{-1}) \leq 3,496\text{mK}^2$ at z=10.4 at 95% confidence .

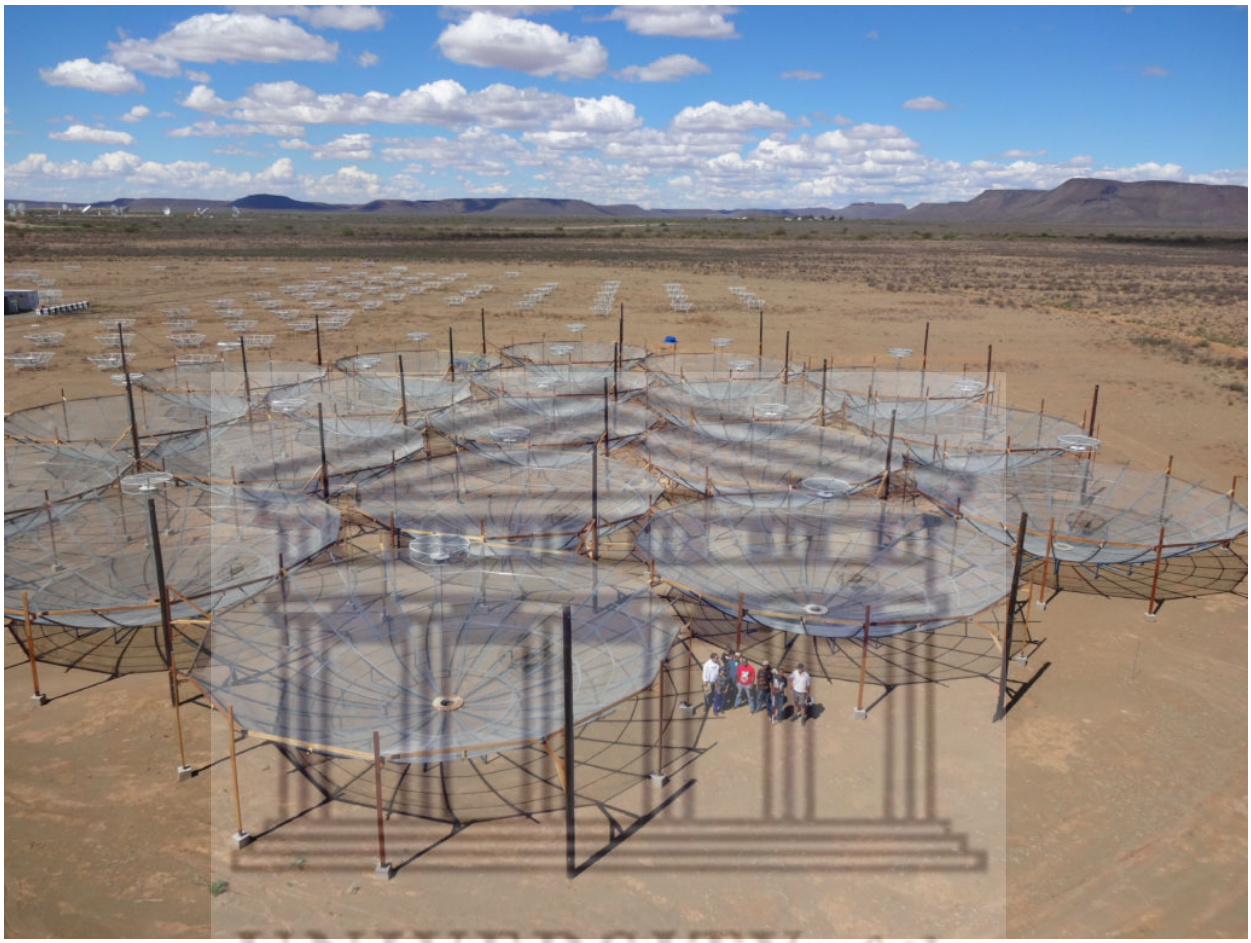

FIGURE 1.11: HERA antennas in Carnarvon. Credit: South African Radio Astronomy observatory (SARAO) (2016).

SKA is one of the largest radio astronomy facilities with its main dishes and antennas situated in South Africa and Australia, respectively. The primary scientific goal of SKA is to unveil the cosmic dawn and the EoR. It has a collecting area of one square kilometre. This radio telescope operates between 50MHz - 14GHz and has improved sensitivity that is $\sim$ 100 times than other telescopes. The lowest frequency of this telescope is built at the MRO in Australia; it is known as the SKA-low. It has 131 072 log-periodic antennas that are grouped into 512 stations of 256 antennas each. A total of 296 of these stations are configured into the core area, while the remaining 216 are along the spiral arms to give this telescope a maximum baseline of 65 km. The SKA-mid antennas are situated in the Karoo desert, Carnarvon, South Africa. It has 133 Gregorian antennas with a 15m diameter. It observes at a frequency range of 0.35-13.8GHz.

This will incorporate the 64 antennas of MeerKAT, which are 13.5 m in diameter. Figure 1.12 shows the artist's impression of this SKA interferometer, with the parabolic dishes in South Africa shown on the left side of the figure and the antennas from Australia shown on the right.

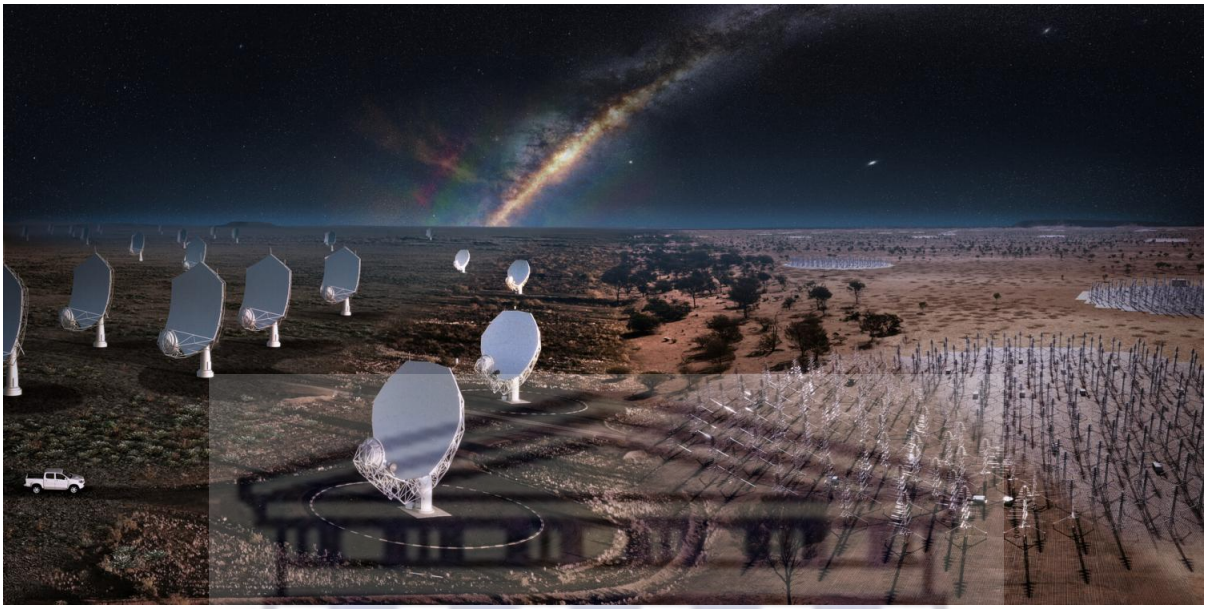

FIGURE 1.12: The overview of the Square Kilometre Observatory(SKAO). Credit: SkAO (2022).

1.4 Machine learning applications on 21cm

Several 21cm studies with machine learning have been explored, with a variety of studies focusing on emulators, parameter estimators, and source identification algorithms. Here, we review some of the previous works done on 21cm with machine learning with both PS and 21cm maps from reionization simulations.

Machine Learning models can be classified as either discriminative or generative models (e.g. Ng and Jordan, 2001). Discriminative models target the posterior distribution and focus on predicting the labels of the data while the generative models model the likelihood of data which captures more information. Discriminative models find the probability directly by assuming some functional form $P(Y|X)$, then estimate parameters of $P(Y|X)$ using the training data. This model learns a boundary between classes in a dataset and is not designed to generate new data. The model is more robust to outliers since the outliers have a moderate impact on the decision boundary. A generative model learns the data distribution. It estimates both the prior $P(Y)$ and likelihood probability $P(X|Y)$ using the training data. Generative models use Bayes' Theorem to calculate the posterior probability $P(Y|X)$. As the name suggests, the models are

able to generate new data points but they are not robust to outliers. Figure 1.13 shows how the discriminative and generative models model data points.

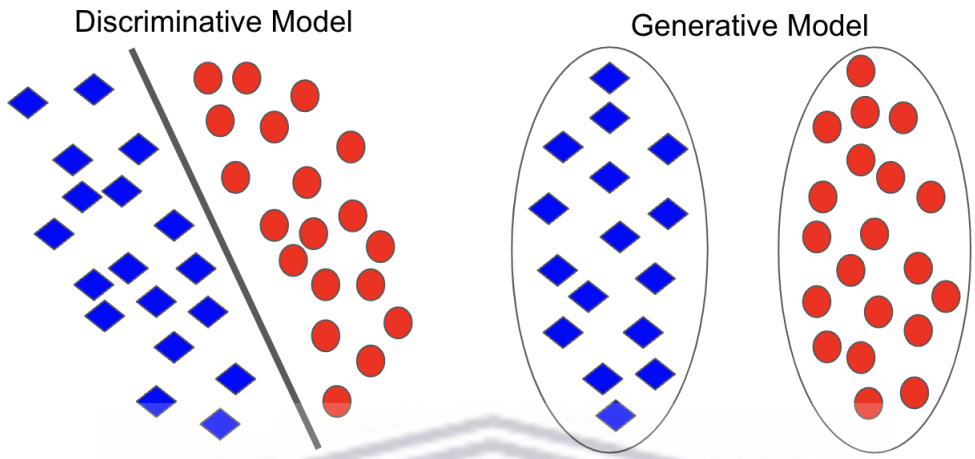

FIGURE 1.13: The difference between a discriminative model and a generative model in terms of how they model data points.

1.4.1 Discriminative models

Using the 21cm PS for parameter recovery

Several studies have focused on using the 21cm PS to get the astrophysical and cosmological parameters by applying machine learning. For example, Shimabukuro and Semelin (2018) used an artificial neural network (ANN) to recover astrophysical parameters from the 21cm PS. The parameters that were recovered from the PS were the ionizing efficiency (ζ), the minimum viral temperature of halos producing ionizing photons (T_{vir}), and the mean free path (mfp) of ionizing photons (R_{mfp}). The PS range explored in this work was $0.06\text{Mpc}^{-1} \leq \text{k} \leq 1.4\text{Mpc}^{-1}$. The authors have found a significant reduction in the error when different redshifts and noise are included during training, achieving RMSE values of 0.172, 0.168, and 0.019 for R_{mfp}, ζ and $log(T_{vir})$, respectively.

Using large scale fields

Although the PS is a great statistical tool to summarize the EoR signal, it has certain limitations. One of the limitations is that when performing parameter recovery using PS, there is some loss of information due to the fact that the 21cm tomographic signal is highly non-Gaussian. Using the 21cm maps to perform parameter inference is a more accurate alternative to the PS. There

have been several studies performed using the 21cm maps for parameter recovery. Gillet et al., 2019 have used a convolutional neural network (CNN) to extract four parameters from the 21cm maps that describe the first galaxies. The parameters recovered were ζ, T_{vir}, L_x/SFR and E_0, where L_x/SFR represents typical soft-band X-ray luminosity to star formation rate and E_0 the minimum X-ray energy capable of escaping the galaxy into the IGM. They used the coefficient of determination (R^2) to quantify the performance of the model. The authors found the R^2 values of T_{vir}, L_x/SFR, ζ, and E_0 to be 0.997, 0.987, 0.955, and 0.728, respectively.

Another work that demonstrates the power of using CNNs for parameter extraction is the work done in Hassan, Andrianomena, and Doughty (2020). In their work, they estimated both the astrophysical and cosmological parameters from 21cm maps at different redshifts. The astrophysical parameters recovered were the photon escape fraction (f_{esc}), the ionizing emissivity power dependence on halo mass (C_{ion}), and the ionizing emissivity redshift evolution index (D_{ion}), and the cosmological parameters recovered were the matter density parameter (Ω_m), the dimensionless Hubble constant (h) and the matter fluctuation amplitude (σ_8). To quantify the results the authors used R^2. They achieve high accuracy, $R^2 > 0.92$, and at lower redshifts and low neutral hydrogen fraction values the value increases to $R^2 > 0.99$.
Additionally, Mangena, Hassan, and Santos (2020), used CNNs to extract the neutral hydrogen fraction directly from 21cm maps. Their network was able to extract the neutral hydrogen fraction with an accuracy of 99 percent, determined by R^2, and this accuracy decreased by 1 percent in the presence of instrumental effects following the SKA design.

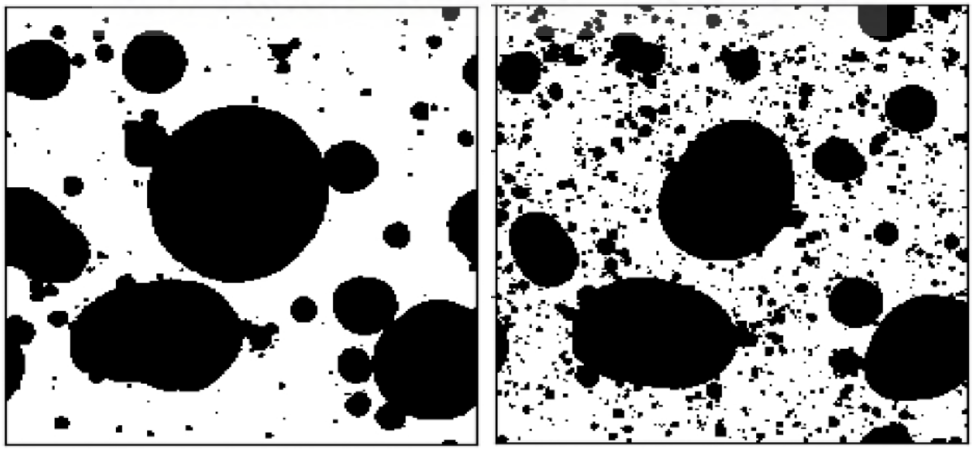

FIGURE 1.14: On the left, the figure shows the bubble size and distribution when AGN is the ionizing source and the figure on the right shows the size and bubble distribution when a galaxy is an ionizing source Hassan et al. (2018).

Finding the sources that are responsible for EoR and the properties that enable them to reionize the IGM are amongst the fundamental questions in astrophysics and cosmology. Distinguishing the 21cm maps from different sources was explored in Hassan et al. (2018) by using a CNN model on data generated using SimFast21, a simulation that we will describe in Chapter 3. Figure 1.14 shows the ionization maps from the AGN and star-forming galaxy when the average ionization fraction is at almost 50%. The field ionized by AGN has fewer small-scale bubbles compared to the one ionized by galaxies.

1.4.2 Generative models

Generating fields from other fields has been explored in Chardin et al. (2019) using an autoencoder, Cosmological Reionization And Deep LEarning (CRADLE), a CNN based model. The data for the training and testing of the model was simulated from EMMA (a cosmological simulation code aimed at investigating the reionization epoch), a code that uses graphics processing units (GPUs) to accelerate hydrodynamic and radiative transfer calculations. This code evolves the radiative transfer using a moment-based method. The autoencoder takes in the stellar particles number counts (sources) and density fields as the inputs into the encoder, and it predicts the reionization times based on these two inputs. These two are merged to form the latent space, from which t_{reion} is predicted. The reionization maps are constructed by marking cells with the cosmological time at which it crosses the reionization fraction of $x_{HII} \geq 0.5$.

Using the testing set on the model, they found out that at large scales, the model makes good predictions, while at small scales, the network does a poor job of predicting the exact shape for the edges of the t_{reion} bubbles. It has been also discovered that the small t_{reion} values are too high compared to the original data. It was concluded that this may be because some small t_{reion} bubbles are missed, or some bubbles not there in the original data were created during the reconstruction process. Figure 1.15 compares the t_{reion} of the original data and the one reconstructed from the model. To quantify the results of the model, they used R^2 and it was found to be > 0.8 in each case. This leaves room for improvement, but the model does a good job of emulating the simulation. This emulator can be used to rapidly obtain the evolving HII regions associated with hydro-only simulations.

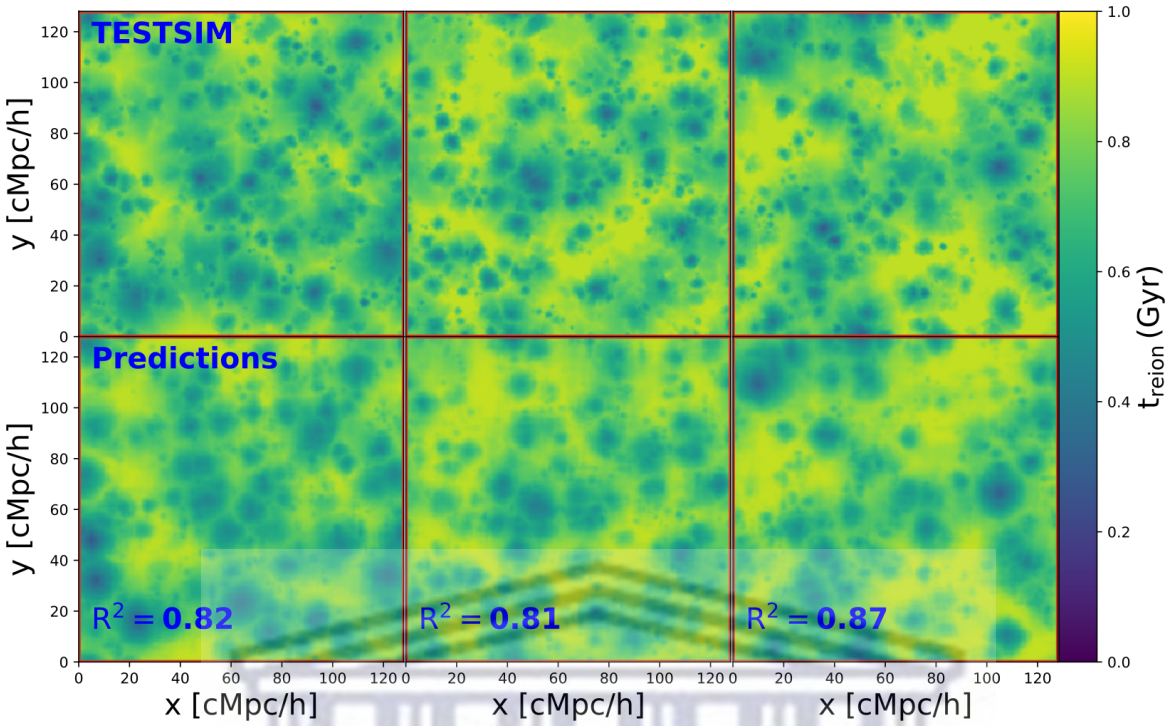

FIGURE 1.15: The results from the Chardin et al. (2019) paper showing the timescales from the simulation at the top row and the predictions from the autoencoder in the bottom row.

1.5 Aims and Objectives

The main focus of this work is to use machine learning techniques to accelerate a semi-numerical simulation, namely SimFast21. These simulations employ an approximated radiative transfer scheme, which represents the most expensive step as compared to generating the density, velocity, and source fields. We here focus on generating the ionization field (the radiative transfer output) directly from the initial density field without using sources' location nor their velocities. We consider several Machine Learning probabilistic and deterministic methods to achieve this task. Several outcomes are expected including :

- Efficient generation of large-scale high-dimensional and highly non-linear fields with a low computational cost.
- Emulating the behavior of radiative transfer on cosmological scales, and full exploration of the high-dimensional prior of the astrophysical and cosmological parameters.
- Enabling efficient high-dimensional posterior distribution studies from future 21cm surveys.

1.6 Overview

The work covered in this thesis is presented in five chapters. The current chapter has focused on introducing our study, with a literature review. Chapter 2 discusses machine learning models and how to build successful CNNs while Chapter 3 focuses on the semi-numerical code (SimFast21) that we used to generate the density and ionization fields to form the dataset used to train our models. This Chapter also presents the methods and approach used to build and train the Machine Learning models. Chapter 4 presents the results from the best-performing models alongside discussion and the final chapter, Chapter 5 provides a conclusion of our main findings and plans for future works.

2 Basics of machine learning

This chapter presents the basics of machine learning with different machine learning methods and their applications. It introduces deep learning which is vital in this work. The chapter explains how to build, tune and train neural networks. It also discusses briefly the generative models we intend to use to accelerate reionization simulations. We here briefly review the basics of machine learning and we refer the interested reader to (Deng, 2014; Malik, 2019; Pant, 2019; Sah, 2020; Alzubaidi, Zhang, and Humaidi, 2021; Tai, 2021; DeLua, 2021; Bewtra, 2022) for a comprehensive review. This chapter is organized as follows: Section 2.1 presents the basics of machine learning and Section 2.2 provides an overview of machine learning methods and gives a brief introduction to deep learning. Section 2.2 also details the fundamentals of machine learning/deep learning models such as activation functions, loss functions, optimizers, and regularization techniques.

2.1 Introduction to Machine Learning

Machine learning (ML) has advanced the field of astronomy in the past decade. We have seen a dramatic increase in the use of ML techniques to analyze data taken by telescopes. Ongoing and planned surveys will produce large amounts of data which requires the development of new techniques. Traditional methods are too slow and they can not support huge volumes of data, as such the new methods to be developed have to be fast and computationally effective.

Artificial Intelligence (AI), is a technique that allows machines to think, learn, understand, and behave like humans. ML is a subset of AI that is concerned with building algorithms that are able to learn over time from historical data; this model can then make a prediction when given new data. We use this approach to solve problems that are too complex for human minds to solve. Deep Learning (DL), is a subset of machine learning that uses neural networks with three or more layers, to learn patterns in a data set and make predictions. Figure 2.1 shows the relationship between AI, ML, and DL.

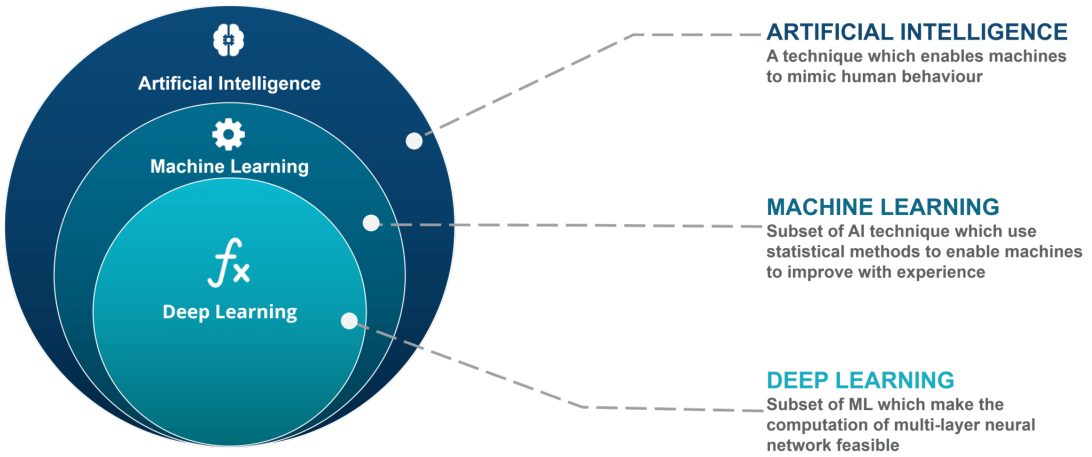

FIGURE 2.1: The relationship between Artificial Intelligence, Machine Learning, and Deep Learning. Credit: Atul (2022)

2.2 Machine learning methods

ML algorithms can either be supervised, unsupervised, or semi-supervised. Figure 2.2 below shows some applications of machine learning methods. Supervised machine learning algorithms are suitable for classification or regression, unsupervised algorithms for clustering, association, and dimensionality reduction, while semi-supervised algorithms are suitable for anomaly detection.

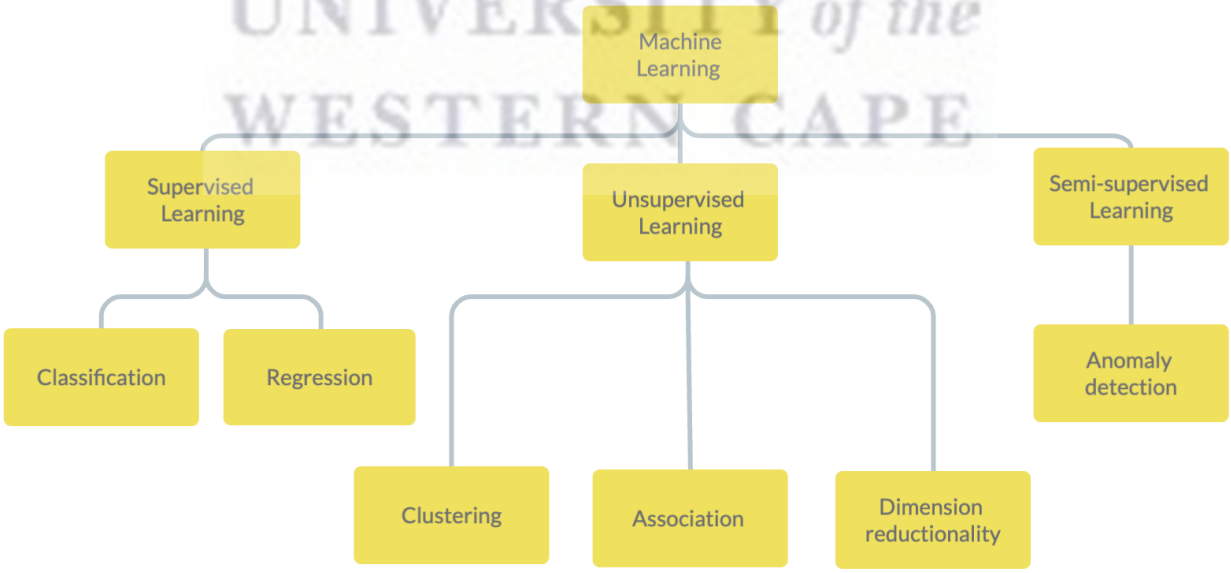

FIGURE 2.2: A tree diagram that shows the different methods used in machine learning and what each method is used for.

Supervised learning is a learning method that can only be applied if all data are labeled to perform classification or regression (e.g. Kotsiantis, 2007; Cunningham, Cord, and Delany, 2008). Examples of classification include identifying whether an input image is a dog or a cat, and classifying whether or not an email is spam. The standard classification algorithms include linear classifiers, Support Vector Machines (SVMs, Cortes and Vapnik, 1995; Zhang and Zhao, 2014; Jones and Singal, 2017), decision trees (Quinlan, 1986; Ball et al., 2006; Franco-Árcega, Flores-Flores, and Gabbasov, 2013), and random forests (Breiman, 2001; Gao, Zhang, and Zhao, 2009). This method can also be used for regression by predicting house or car prices. Standard regression algorithms are linear, logistic, and polynomial.

Unsupervised learning is a learning method that can be applied when there is unlabelled data to perform clustering, association, and finding a lower dimensional representation of the data (Becker and Plumbley, 1996; Cios et al., 2007). This method can find similarities and differences in the dataset. As seen in Figure 2.2, this method can be used for association, dimension reduction, and clustering. Association uses rules to find the relation between variables in a dataset. A dimension reduction algorithm is used when the number of features in a dataset is too high. These features are reduced to a reasonable size without compromising the original information. Principle Component Analysis (PCA, Singh, Gulati, and Gupta, 1998; He and Zhao, 2019; Nemec, 2022) is the most typical reduction method widely used in machine learning. In clustering, data is grouped based on similarities (Jain, Murty, and Flynn, 1999). A popular clustering algorithm is the K-means (Arthur and Vassilvitskii, 2007; Na, Xumin, and Yong, 2010; Jin and Han, 2011) which assigns similar points into groups. Other methods of unsupervised learning include neural nets and probabilistic clustering methods.

Semi-supervised learning is a learning method that can be applied when a dataset has labeled and unlabelled datasets. A smaller labeled dataset is used to guide classification and feature extraction from a larger unlabelled dataset (Zhu and Lafferty, 2005; Chapelle, Chi, and Zien, 2006). This method is used to determine, for instance, anomalies in a dataset. In this case, the algorithm alternates between supervised learning for a subset of labeled data and unsupervised learning for the other subset of unlabeled data.

All ML algorithms follow the same workflow. This workflow has five major stages that describe the algorithm, namely data collection, data pre-processing, model selection, training and testing

the model, and evaluating the model. The key to building a good ML algorithm starts with balancing and pre-processing datasets. A common practice is that the dataset is split into different subsets for training and testing. The next step is to choose the right model. Choosing an ML method depends on several factors, including the presence of labels, data structure, and the nature of the problem. The trained model is then tested on the testing (unseen) dataset to verify its predictive power to generalize on out-of-distribution samples. Depending on the nature of the problem, an accuracy/performance metric will be selected to evaluate the trained model. Generally, complex datasets require complex models. DL models have been very successful in generalizing over complex datasets. The data used in this thesis is very complex (highly non-linear, and high dimensional) and it requires DL models. In the next section, we discuss DL and its components.

2.2.1 Deep Learning

Deep learning is a subset of ML that uses multilayer neural networks that can mimic a human brain and solve complex problems. They have three layers (input layer, (multiple) hidden layers, and output layer) that make use of neural networks to solve complex problems. There are different architectures in DL including but not limited to Multi-Layer Perceptron (MLP, Almeida, 1997; Popescu et al., 2009), Radial Basis Networks (Orr, 1996), Recurrent Neural Networks (RNNs, Rumelhart, Hinton, and Williams, 1985), Convolutional Neural Networks (CNNs, LeCun et al., 1989), and Generative Adversarial Networks (GANs, Goodfellow et al., 2014). Neural networks are the functional unit of DL and essentially imitate how a human brain works to solve problems. They transform an input into a meaningful output. These models have various layers of information-processing stages in hierarchical architecture that are exploited for pattern classification and representational learning. DL models have been used in several fields, and their functionalities are growing with time.

Neurons make up a neural network architecture. Figure 2.3 shows the structure of a neuron, where **x** represents the input data with different structure (e.g. signal or image) and features that the model will learn from, **w** represents learnable weights to process the input. These neurons construct the relationship between the input data and the output. Bias, **b**, is added as an additional learnable variable to add more degrees of freedom to assist training and reduce

over-fitting. The linear combination ($\sum$), binds the weights and the inputs together to find their sum, while the activation function, (ψ), introduces non-linearity. This can be described as a linear combination as follows:

$$Y_k = \psi\left(\sum(w * x_i) + \theta\right). \tag{2.1}$$

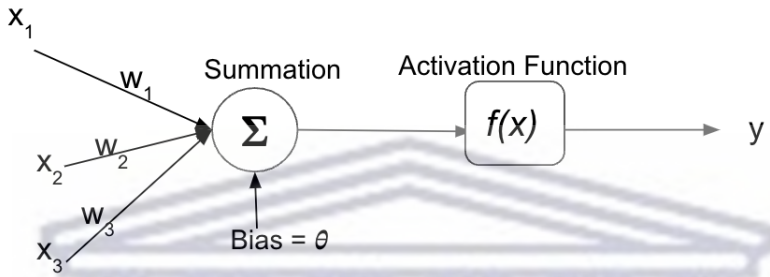

FIGURE 2.3: The structure of an artificial neuron showing the different parameters that define the equation of a neuron.

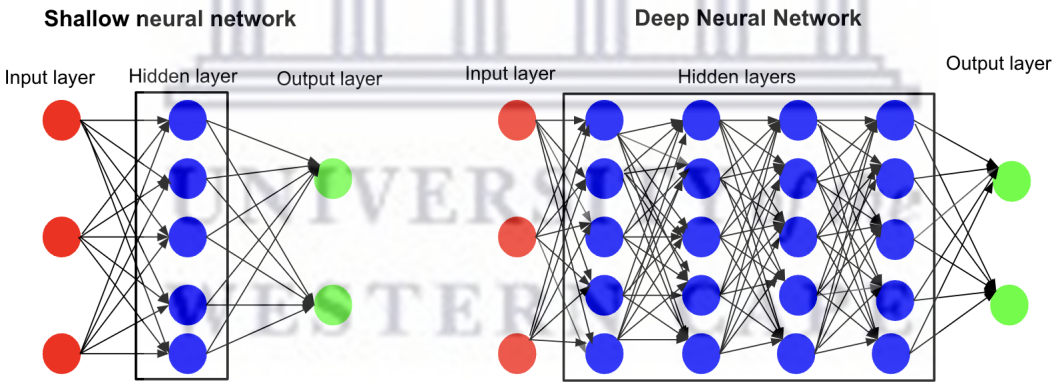

FIGURE 2.4: The difference between shallow and deep neural networks. The middle layer, commonly called the hidden layer, is where learning takes place.

Traditional machine learning techniques have been using simple architectures that contain a single layer of non-linear feature transformation. These shallow models, for instance, include the Hidden Markov Models (HMMs, Handel, 2008), Conditional Random Fields (CRFs, Sutton and McCallum, 2011), and Support Vector Machines (SVM, Busuttil, Abela, and Pace, 2004). These models have one common aspect: a simple architecture consisting of only one layer responsible for transforming the input data into a problem-specific feature space. HMMs, CRFs, and SVM have been efficient in solving simple and well-defined problems and they are

not efficient for real-world applications on big data. Deep models were introduced to address the shortfalls of shallow models. Figure 2.4 shows a comparison between a shallow and a deep neural network where the main difference is the number of hidden layers present in each network. A shallow neural network has one hidden layer while a deep neural network has multiple hidden layers (two or more).

2.2.2 Activation Functions

There are many activation functions used in neural networks. These functions introduce non-linearity into a neural network so that the network can identify and learn complex patterns in a dataset. DL architectures use activation functions so that they can be able to perform computations that are distinctive between the hidden and output layers of these architectures. Activation functions are very efficient when used with their default hyperparameters. In addition to adding non-linearity, the activation function used in a neural network depends on the network output. For example, if the output or target range is from zero to one, sigmoid or softmax will be the best activation functions to use at the last layer. The activation function then decides whether a specific neuron should be activated following simple mathematical operations. The common activation functions used in deep learning that will be discussed here are Sigmoid, Softmax, Tanh, Rectified Linear Unit (ReLU), and the Leaky ReLU (for an extensive review see Neal, 1992; Karlik and Vehbi, 2011; Nwankpa et al., 2018; Jain, 2019; Szandała, 2020; Dubey, Singh, and Chaudhuri, 2021). These activation functions can be applied to any problem. The most common activation function in the intermediate layers is ReLU/LeakyReLU. For the output layers, it depends mainly on the task. For instance, in regression tasks, usually, no activation is required. In classification tasks, the desired output should be in the range between 0 and 1 to quantify the probability of the object being classified correctly, as a result, softmax or sigmoid might be more appropriate to use. Tanh is usually used in generative models where the dynamic range is large, especially in denoising probabilistic approaches.

- **Sigmoid**

  Sigmoid activation function transforms values between 0 and 1 and is used in feed-forward neural networks and most DL architectures' output layers. It has an S shape. This activation function would be suitable to use when performing binary classification.

Although sigmoid is continuously differentiable, the activation function has two major drawbacks; it has a vanishing gradients problem and it is not zero-centered.

Figure 2.5 shows that the gradient turns to zero with large positive and negative values and the function approaches zero and one at very large negative and positive values of x, respectively. Since this function is a complex trigonometric function, it is computationally expensive. The sigmoid function is described using the following equation:

$$\text{sigmiod}(x) = \frac{1}{1 + e^{-x}}. \tag{2.2}$$

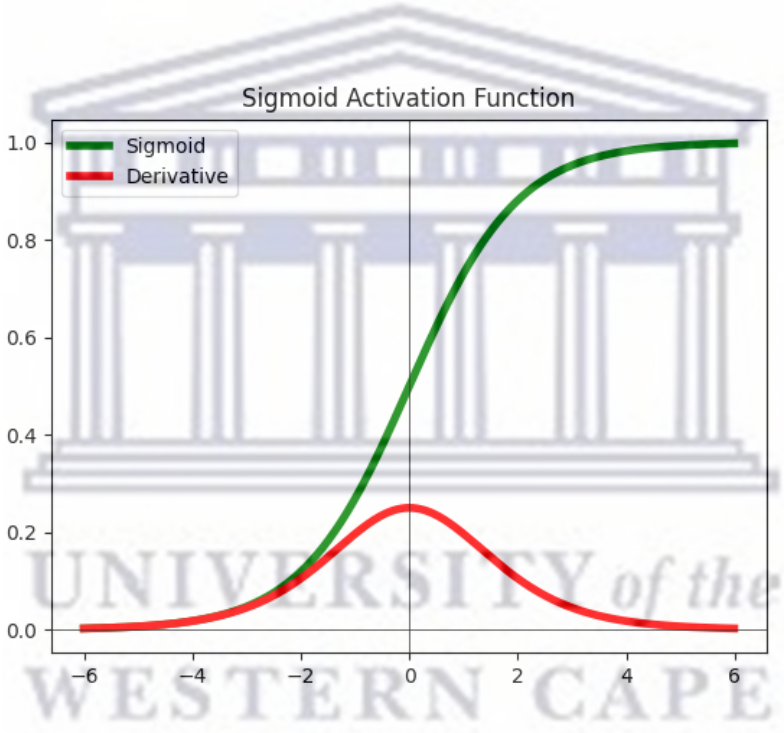

FIGURE 2.5: Sigmoid activation function and its derivative.

- **Softmax**

  Softmax is a generalized form of the sigmoid function. This function is usually used in the final layer of classification models because it predicts probability scores during classification tasks. The sum of the probabilities for the softmax activation function is one. It works for multi-class classification problems. Softmax takes multiple inputs and assigns a probability to individual outputs. For instance, an output of 0.7 indicates an accuracy of 70%. The function is suitable for logistic regression or neural networks. The

equation that defines the softmax is:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{2.3}$$

where z_i is the input vector, e^{z_i} is the standard exponential function for the input vector, K is the number of classes in the multi-class classifier and z^{z_j} is the standard exponential function for output vector. Figure 2.6 shows the softmax activation function and its derivative.

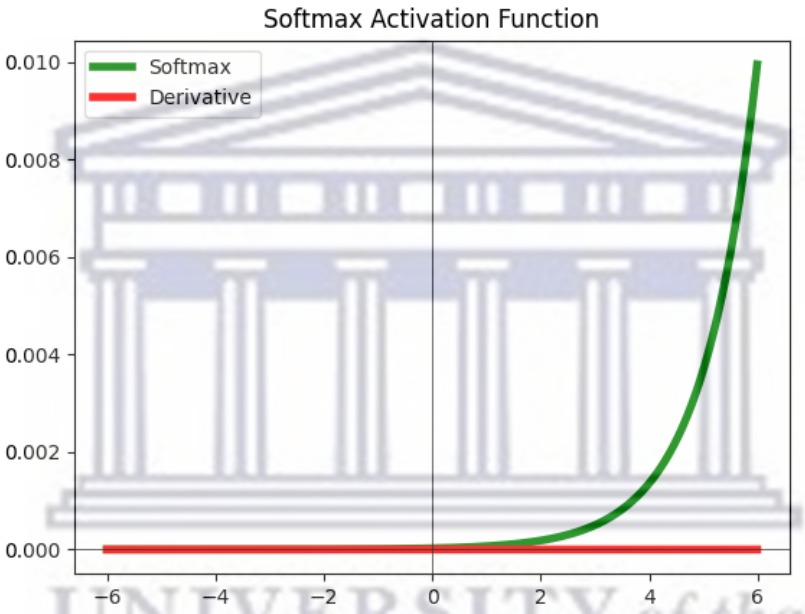

FIGURE 2.6: Softmax activation function and its derivative.

- **Tanh**

  Tanh, a hyperbolic tangent function is an activation that produces values between -1 and 1. The function performs better in multi-layer neural networks. It has the same s shape as the sigmoid function except it is zero-centered. Although it is zero-centered, the tanh function suffers from vanishing gradient problems. The function is defined by Equation 2.4, where x is the input and tanh (x) is the output.

  From Figure 2.7, it can be seen that the function is zero-centered and, at the same time, has a zero gradient for large positive and negative values. Tanh has been used in tasks such as natural language processing (Dauphim et al., 2017), speech recognition (Maas,

Hannun, and Ng, 2013), and generative models.

$$\text{Tanh}(x) = \frac{e^x - e^- x}{e^x + e^{-x}}. \tag{2.4}$$

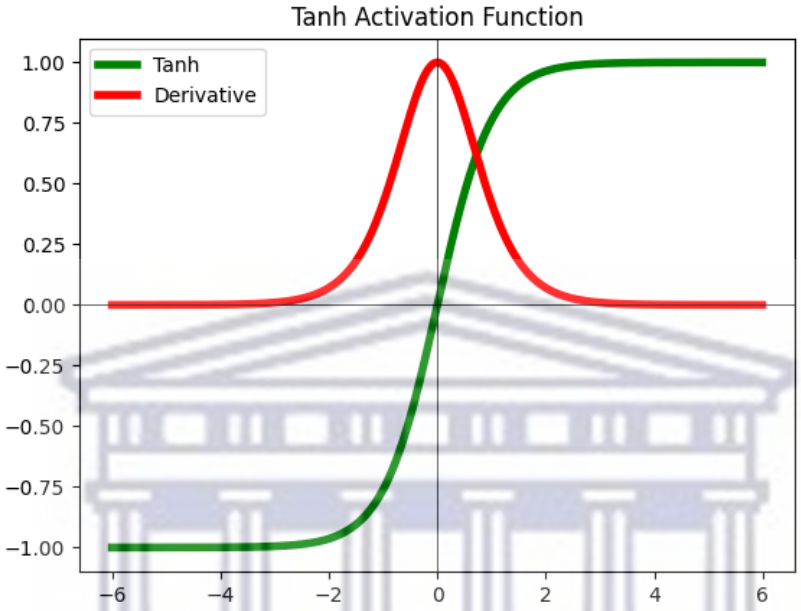

FIGURE 2.7: Tanh activation function and its derivative.

- **ReLU**

The most common activation used in neural networks is ReLU since it is not computationally expensive and does not suffer from vanishing gradient problem. What makes the function computationally efficient is that it does not activate all the neurons at once. ReLU is not zero-centered and it suffers from the "dying" ReLU problem. This means, as seen from Figure 2.8, this function outputs a value of zero for all negative inputs. The “dying" ReLU refers to when some nodes die to halt learning. The following equation defines the ReLU activation function:

$$\text{ReLU}(x) = max(0, x)\,, \tag{2.5}$$

where x is the input and ReLU (x) is the output. The ReLU keeps only the important (positive) and removes the less important (negative) features.

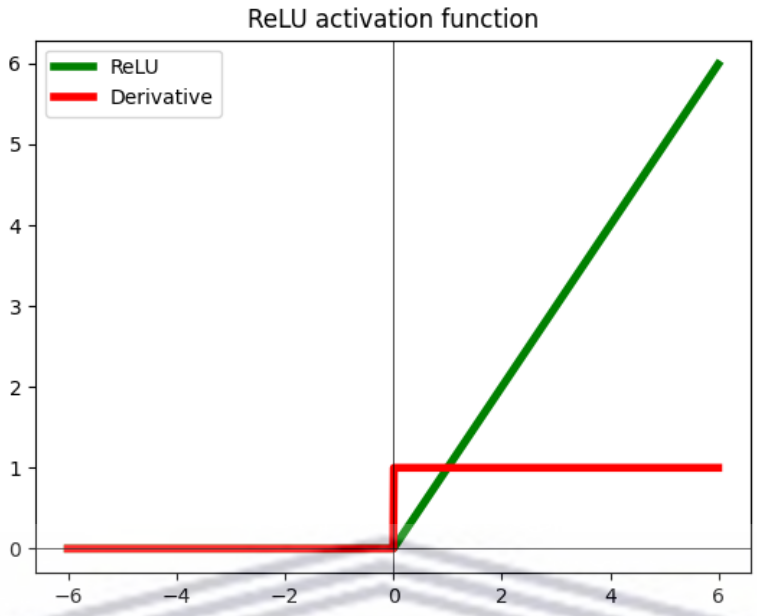

FIGURE 2.8: ReLU activation function and its derivative.

- **Leaky ReLU**

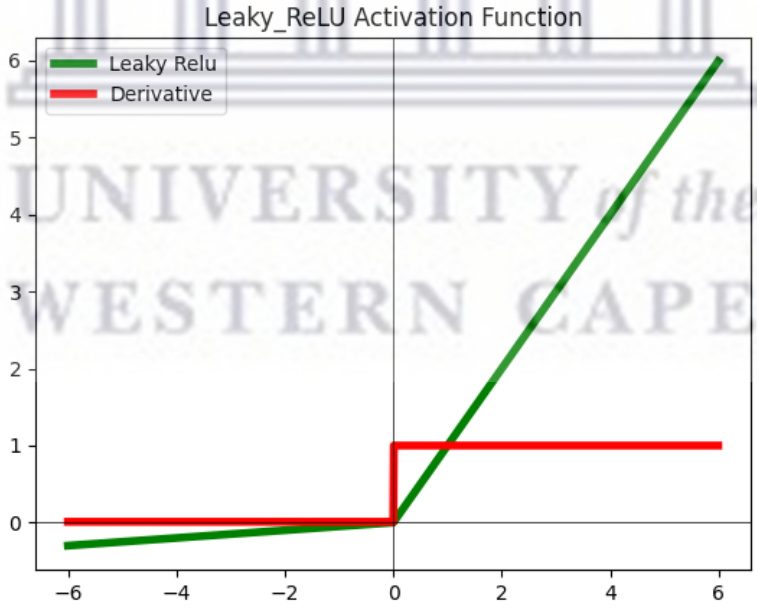

FIGURE 2.9: LeakyReLU activation function and its derivative.

The problem of "dying" ReLU can be solved by using Leaky ReLU. LeakyReLU is the modified version of the ReLU function where a negative slope is introduced to the ReLU function, as seen from the plot in Figure 2.9. The gradient will not be zero, it can be

seen from the ReLU function (2.8) that values that are less than zero have a zero gradient during training but for the LeakyReLU the values of the gradient are non-zero (2.9). The difference between the ReLU and Leaky ReLU can be seen in Equations (2.5) and (2.6), respectively. The hyperparameter α is usually set to be closer to zero. The default value of this hyperparameter is 0.01. If α is set to 1, Leaky ReLU becomes a linear activation function.

$$\text{LeakyReLU}(x) = max(\alpha x, x). \tag{2.6}$$

2.2.3 Loss functions

Loss functions are used to quantify error or the distance between the model predictions and the target. At the beginning of training, the loss value is high since the predictions are significantly different from the target. As training progresses, the loss decreases. The loss evolution of a neural network is expected to converge to minima (whether local or global minima). An optimal loss value is zero indicating a perfect match between the prediction and the target. Different loss functions are used for different tasks, e.g. classification or regression, and the loss function used depends on the error type (For review on loss function see Wang et al., 2022; Ciampiconi et al., 2023).

Regression loss functions

Common regression loss functions include the Mean Squared Error (MSE), Mean Absolute Error (MAE), and log-cosh error. In all the following equations defining different loss functions, y refers to the true value, $\hat{y}$ is the predicted value, $\sum$ is the summation over all the samples and N is the number of the samples in the dataset.

- The MSE is a famous loss function for prediction problems. This loss function is sensitive to outliers. The function has one global minimum since it is a convex function. MSE is differentiable and it can be used with gradient-based optimization algorithms such as stochastic gradient descent (SGD). When the model is too complex or the dataset is too small, the model using MSE as a loss function might overfit. It is described by the equation below:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{2.7}$$

- The Mean Absolute Error (MAE) is the average of the absolute differences between the actual and the predicted value. Unlike the MSE, it is insensitive to outliers because it measures the average magnitudes of the errors without considering their direction. MAE is not differentiable but it can still be used with gradient-based optimization algorithms such as SGD. The function that describes this loss is given as:

$$\text{MAE} = \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{N}. \tag{2.8}$$

- Log-cosh error, which is the logarithmic of a hyperbolic cosine function that is much smoother than the MSE. It is less sensitive to outliers and it is suitable for training neural networks using optimization algorithms such as SGD. Log-cosh is smooth, differentiable, and has a continuous gradient. Similar to the MSE, it has a single global minimum. This loss function measures the difference between the predicted and the true values in terms of the hyperbolic cosine of their differences. This is described by the following Equation:

$$\text{Log} - \text{Cosh} = \sum_{i=1}^{N} log(cosh(y_i - \hat{y}_i)). \tag{2.9}$$

Classification loss functions

Widely used classification loss functions include Cross-Entropy/Log, Hinge loss, and Kullback-Leibler Divergence.

- Cross-Entropy loss function measures the variance between the predicted and the true probability distribution. It can be used in conjunction with the softmax activation function for multi-class classification problems. This loss function is differentiable, convex with a single global minimum and it can be optimized using SGD. Cross-entropy is able to handle imbalanced datasets. When the classification problem has two labels, the Binary Cross Entropy loss function is used:

$$L = -\frac{1}{m} \sum_{i-1}^{m} (y_i.(log(\hat{y}_i)) + (1 - y_i).log(1 - \hat{y}_i)), \tag{2.10}$$

and for multi-class classification:

$$L = -\frac{1}{n}\sum_{i}^{n}\sum_{j}^{m} y_{ij} log(\hat{y}_{ij}) \tag{2.11}$$

This function is derived from a regular likelihood function and the addition of logarithms, which is why it is referred to as a log loss function sometimes

- The Hinge loss was primarily developed for SVMs. This loss function penalizes predictions that are wrong or have low confidence. Hinge loss is used for binary classification problems. It is a convex function with a single global minimum and it can be used with optimization algorithms such as SGD. A convex function is a function where the value of the slope increases along with the increase in the value of x. While this function is not differentiable, it is more robust to outliers and noise. This loss function is given by the equation below:

$$L = max(0, (1 - y)\hat{y}). \tag{2.12}$$

- Kullback-Leibler (KL) Divergence loss functions measure the difference between two probabilistic distributions; the true and predicted probabilistic distributions. This loss function is usually used in generative models and density estimators. KL divergence can be used in both regression and classification problems. The equation below describes the KL divergence loss function:

$$D_{KL} = \sum_{i} P(i) log \frac{P(i)}{Q(i)}, \tag{2.13}$$

where P is the true probabilistic distribution and Q is the predicted probabilistic distribution.

2.2.4 Optimizers

Optimizers are used to minimize loss functions to achieve learning. The optimizers generally compute all partial derivatives of the loss with a repeat to the network parameters to find the direction at which the loss is minimal. These partial derivatives are added to the parameters during the updates, and the speed by which the parameters are directed toward the loss minimum

is controlled by the learning rate, a free hyper-parameter that is usually very small to prevent exploding gradients. The learning rate, as the name suggests, is used to control the rate at which the network learns. The training speed is related to the learning rate. Low learning rate values would result in slow training which might require using larger training epochs, whereas large learning rates might lead to instability of the model and large oscillation around the minimum. An epoch refers to the number of times the whole training dataset goes through the network during training. A good learning rate is found by running several trials to observe the best loss evolution over iteration to achieve convergence. This subsection discusses widely used optimizers such as gradient descent, Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (Adam).

Gradient Descent

Gradient descent is one of the most common optimization algorithms used to find the optimal parameters for a ML/DL model. In gradient descent, the parameters are adjusted iteratively to minimize a given function to its local or global minimum. This type of algorithm update parameters in the direction of the steepest descent of the loss function. There are three different types of gradient descent methods, namely SGD, batch gradient descent, and mini-batch gradient descent (e.g. Robbins and Monro, 1951; Ruder, 2016). Batch is the number of training examples used at each iteration during forward or backpropagation.

- SGD updates the parameters of the model simultaneously. The main advantage of this optimization algorithm is that it updates the parameters frequently, and requires less memory to compute. It can be applied to datasets that are large since parameters are updated simultaneously. The drawback of this algorithm is that it can result in a noisy gradient which might result in the error increasing instead of the opposite, it has a high variance and it is sensitive to learning rates.

- Batch gradient descent is an optimization algorithm where gradients are computed for the entire dataset. The convergence of batch gradient descent is very stable and more accurate although it takes time to converge. This optimization calculates the gradients of the whole dataset to perform a single update, and hence it might be slow for large

datasets. It is sensitive to outliers, requires more computation and memory, and can fall into local minima.

- Mini-batch gradient descent combines the concepts used in SGD and batch gradient descent. The training dataset is split into small datasets called mini-batches and parameters are updated based on these mini-batches. The splitting into mini-batches is usually a small number in order of 10. The algorithm leads to a more stable convergence, calculating the gradients is more efficient and it requires less memory.

Root Mean Square Propagation (RMSProp)

RMSProp (Tieleman and Hinton, 2012; Dauphin et al., 2015) is a gradient-based optimization technique where the learning rate for each parameter is updated based on the magnitude of recent gradients. It is a stochastic technique for mini-batch learning. Major advantages of this optimization algorithm include the use of adaptive learning rates, robustness against noisy gradients, and low memory requirement. Disadvantages include sensitivity to hyperparameters used and the slow to converge.

Adaptive Moment Estimation (Adam)

Adam (Kingma and Ba, 2014) is an optimization algorithm that is the combination of SGD and RMSProp. This optimizer starts with a set of initial parameters and they are iteratively adapted using a loss function. The advantages of this algorithm include fast convergence, robustness to noise, the use of adaptive learning rates, and low memory requirement. When the model is large it requires large memory to compute. Adam is sensitive to hyperparameters.

2.2.5 Regularization Techniques

A model with a large number of parameters and a small dataset may overfit. Regularization techniques are used to combat overfitting and underfitting in DL models. Here, we discuss two popular regularization techniques used in many DL architectures, namely, Batch Normalization and Dropout.

- Dropout is often used in complex DL networks to prevent complex co-adaptations on the training data. Dropout is a Regularization technique where hidden/visible units in

a neural network are dropped randomly (Hinton et al., 2012; Srivastava et al., 2014). The random drop is determined by the probability equation $q = 1 - p$, where p is the probability of retaining the units. For example, if q is 0.2, 20% of the units will be dropped from the network. They are temporarily disabled, along with their inputs and outputs from the connection. An ideal dropout rate in hidden layers is 0.5 while for the input layer is 0.8. Figure 2.10 shows the difference between neural networks without and with dropout.

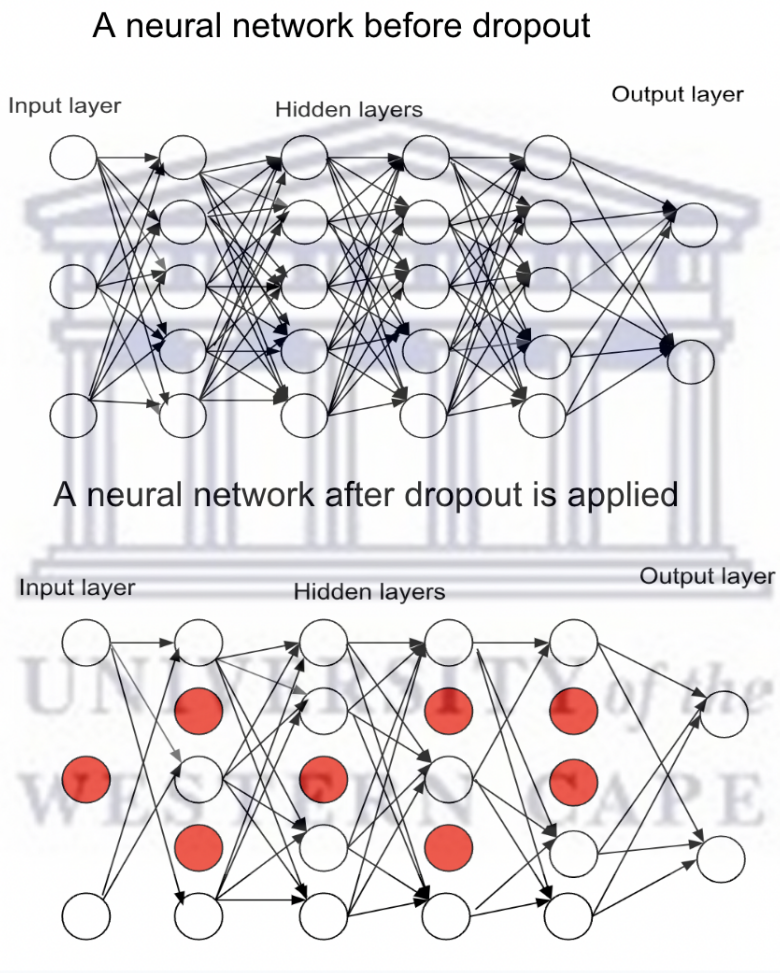

FIGURE 2.10: The difference between a neural network without dropout (top figure) and a neural network after dropout is applied (bottom figure). The red units are randomly disabled.

- During training, the distribution of each layer's inputs changes due to the change in parameters in the previous layers, this phenomenon is called internal covariance shift. Internal covariance shift slows down training and Batch Normalization (BN, Ioffe and Szegedy, 2015) has been introduced to solve this problem. If BN is not used and a higher learning rate is, this could lead to vanishing or exploding gradients in a model and the

network can fall into a local minima. BN allows the use of higher learning rates, model trains faster, and achieves higher accuracy. The BN transform algorithm, which is applied to activation x over a mini-batch, B, is given by the following equations:

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{2.14}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2, \tag{2.15}$$

$$\hat{x} \leftarrow \frac{1}{m}\sum_{i=1}^{m}\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{2.16}$$

$$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i), \tag{2.17}$$

where μ_B (Equation 2.14) is the mean of the mini-batch, σ_B^2 (Equation 2.15) is the variance calculated from the mean of the mini-batch, the x values are normalized using Equation 2.16 to get the outputs defined by Equation 2.17. ϵ, a constant, is added to the mini-batch variance for numerical stability. Normalization is applied to each activation independently and a pair of parameters $\gamma^{(k)}, \beta^{(k)}$ are used to scale and shift the normalized value to give the output, Equation 2.18 shows this relation:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}. \tag{2.18}$$

2.2.6 Overfitting and Underfitting

The loss evolution is set by training. There are three possible outcomes for the loss evolution; underfitting, overfitting, and a good fit.

Overfitting happens when a network learns the training dataset and it is able to model this dataset perfectly so that it learns the noise (Sarle, 1995; Gaurang et al., 2011; Allamy, 2014; Pothuganti, 2018) or the random error and not the underlying relationship between the data points in the dataset. The network is not able to generalize over a new dataset (testing dataset). An underfitting model performs poorly on both the training and the testing dataset while one that is overfitting may do well on the training set and performs badly on the training set.

There are several reasons that will lead the model to overfit or underfit. One reason might be due to a small dataset. There are a couple of ways to prevent a model from overfitting or underfitting (Haider, 2014; Pothuganti, 2018), these include early stopping, network reduction, data expansion, and regularization. Early stopping ensures that the model is stopped/ halted when no improvement is observed over some number of iterations that are found by experiments.

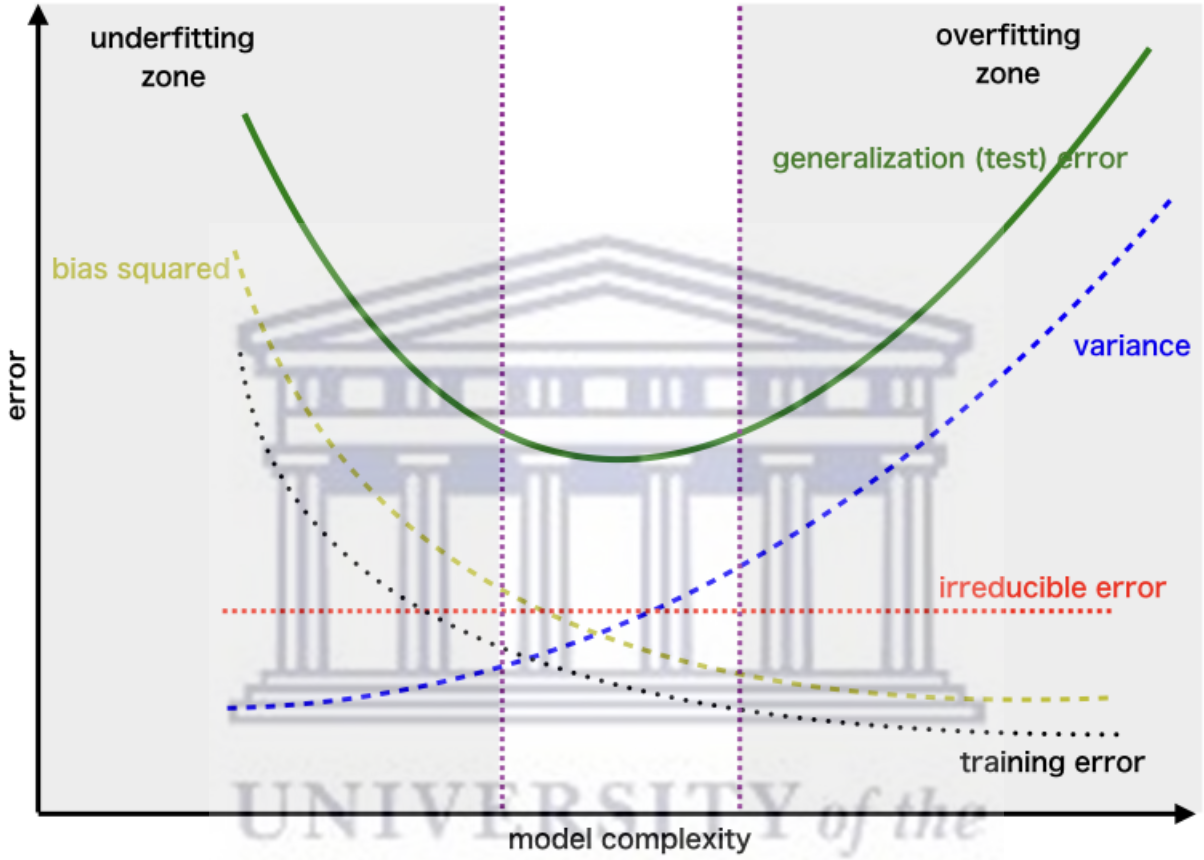

FIGURE 2.11: Bias-variance trade-off. The plot shows the areas where a model is prone to underfitting or overfitting. Credit: Papachristoudis (2019)

Understanding prediction errors is very important in any ML algorithm. These prediction errors, also known as reducible errors are bias and variance and they can be reduced to improve model accuracy. Bias refers to the difference between the average prediction of a model and the correct/true value that the model is trying to predict and the variance is the variability of the model prediction for a given data point that indicates the spread of the data (Geurts, 2002, Chan, 2020). There is a trade-off that exists between bias and variance, the bias-variance trade-off. Bias-variance trade-off shows the relationship between a ML model's ability to fit the training data and its ability to generalize on unseen data. A model that is too simple and has fewer parameters is prone to underfitting and this happens when there is high bias and low variance.

When a model has a large number of parameters it will have high variance and low bias and this leads to overfitting. Ideally, when training a ML model the target is to have low bias and low variance.

The bias and variance of a model are influenced by parameters such as complexity, noise, learning algorithm, and learning sample size. Figure 2.11 shows the relationship between bias and variance, where a model is likely to overfit or underfit, and the progression of the loss evolution. The x-axis represents the complexity of the model, from simple (left) to complex (right) while the y-axis represents the error, from low (bottom) to high (top). Bias is a decreasing function of complexity while variance is an increasing function. As the complexity of the model increases, the bias decreases and this leads the model to generalize better on the training data. The variance increases as model complexity increases, leading to poor generalization of the testing data. The total error of the model is given by the sum of bias squared, variance, and irreducible error. The aim of this bias-variance trade-off is to find a spot where the total error is minimized by finding the right level of model complexity that minimizes both bias and variance. This spot is marked by parallel purple dotted lines in the figure.

2.2.7 Building a Neural Network

A neural network can be built using either dense layers or convolutional layers. There are several components to consider for building neural networks, which can be summarized as follows:

- Input layer- takes in the input features or raw data.
- Hidden layers to transform the input layer into an intermediate representation.
- Activation function to introduce non-linearity in the network.
- Size and type of weights and biases.
- Output layer that gives the final output of the neural network.
- Loss function to quantify the error
- Optimization algorithm to optimize parameters (weights and biases)
- Regularization techniques to prevent overfitting (Dropout or BN)

Most of these components have been discussed in detail in previous subsections and in this subsection we will focus on how to build a neural network using dense layers and convolutional layers, with more focus on the latter as most of the models explored in this thesis use convolutional layers.

A dense layer, which is sometimes referred to as a fully connected layer, is a layer where a neuron in a layer receives input from all other neurons in the previous layer. It is commonly used in Artificial Neural Networks (ANNs) or in the final layer of a neural network. A dense layer has several hyperparameters such as units, activation function, bias initializer, and kernel regulariser. The units define the size of the output from the dense layer, the activation function is used to transform the input values of the neuron by introducing non-linearity, the bias initializer initializes the bias vector if a bias vector is applied, and the kernel regulariser regularises the kernel weight matrix.

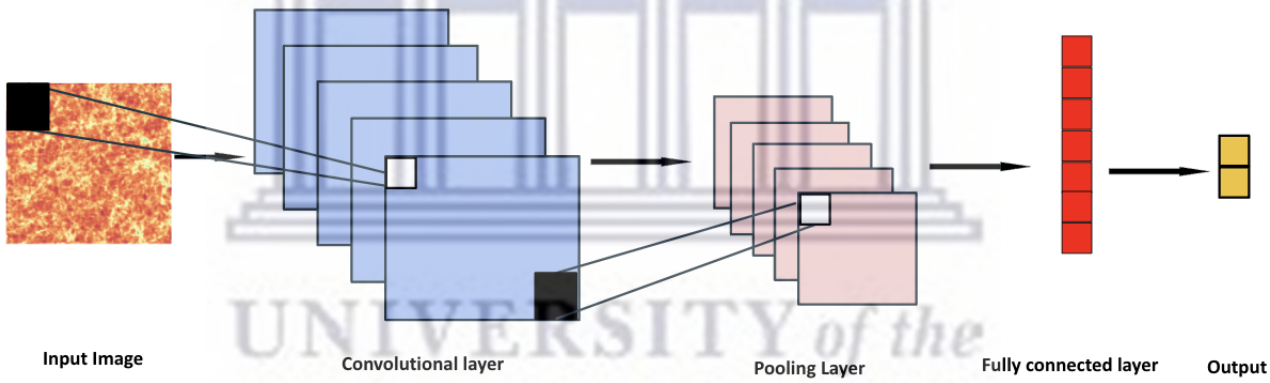

FIGURE 2.12: A mechanism for building a CNN.

A convolutional layer performs convolution to the input data and it was designed to work with high dimensional data. It takes five arguments: the filter, filter size, padding, stride, and an activation function. Usually, an activation function is applied after the Regularization such as BN or dropout. The filter, also known as the feature detector, is important for high dimensional data when the local features are important. The filter size produces the size of this filter matrix. Padding is used to add layers to preserve the input size/dimensions and the stride parameter determines the number of pixels the filter to scan over the image. The stride parameter affects the spatial dimensions of the output feature map. When the parameter is set to one (no stride), the filter will move one pixel in the horizontal and vertical directions and when the stride is set

to be greater than one, the filter will skip pixels during the convolution process. The activation decides which neurons are activated.

A convolutional neural network is constructed using multiple convolutional layers, pooling layers, and a dense (fully connected) layer. Figure 2.12 shows a simple architecture for a CNN with all the important layers indicated. CNN transforms an image for feature extraction. The image is convoluted with a kernel of a specific size. The pooling layer is used for dimensionality reduction to keep more prominent features. There are 2 types of pooling mechanisms that can be used, max pooling and average pooling. Max pooling takes the maximum value in a matrix while the average pooling takes the average value. Max pooling is used in most cases. Figure 2.13 shows the difference between the two pooling mechanisms.

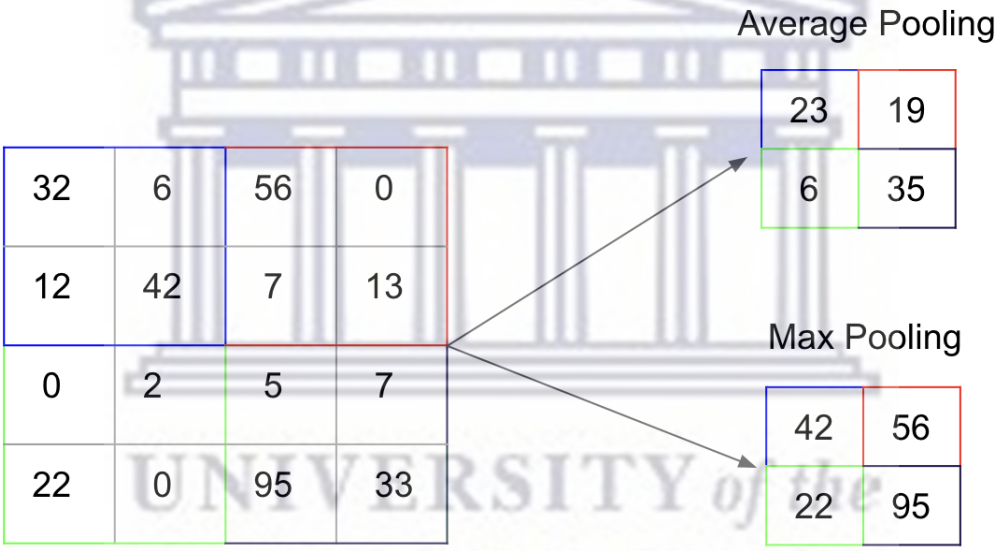

FIGURE 2.13: The difference between the pooling mechanisms; max pooling and average pooling.

The previous section has laid a foundation for neural networks by discussing how they are built, regularised, and optimized. Since generative models are the core of this thesis, in the next section we discuss some of the explored variations of generative models.

2.2.8 Generative models

The generative models discussed here are autoencoders and U-Nets and they are defined as such because they implicitly learn the distribution of the data. Here, we provide a review of these different models and show specific examples of their applications in Chapter 3.

Autoencoders

Autoencoder (Rumelhart, Hinton, and Williams, 1986; Baldi, 2012; Bank et al., 2020) is a type of deep learning model used for dimension reduction and learning efficient encoding of a given dataset and can be conditioned on labels. They can be used to reconstruct the input. Autoencoders are trained to learn the encoding of the data. Figure 2.14 shows the structure of a basic autoencoder. The three major parts of the autoencoder are the encoder, latent space, and the decoder. The purpose of the encoder layer is to act as a feature extractor, it will extract the most useful information in an image and store it as a vector called the latent space. The latent space, sometimes referred to as the bottleneck layer, is the code, this is the compressed vector space of the learning representation. The decoder is the last layer, it attempts to reconstruct the input image from the latent space.

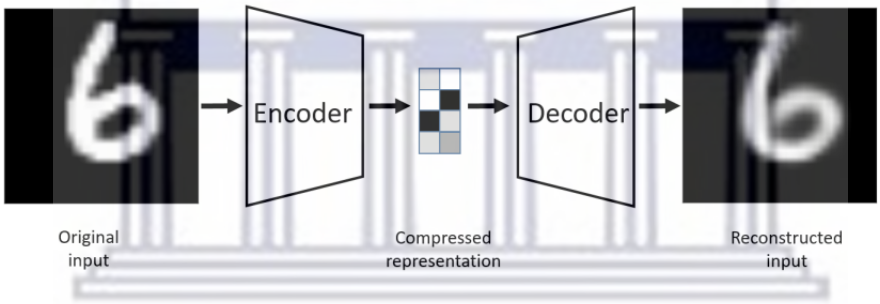

FIGURE 2.14: An example of an autoencoder showing the input, encoder, compressed/latent representation, decoder, and output Bank et al. (2020).

In summary, the autoencoder is a compression technique to find a lower representation of the dataset, from which the representation can be used to recover the original information or data. An optimal latent representation allows for a successful reconstruction of the data with maximum accuracy.

There are different variations of autoencoders. The main ones include denoising, sparse, contractive, under-complete, dense, convolutional, and Variational Autoencoder (VAE). These networks are mainly used for building generative models. The main goal of all versions of the autoencoders is to perform reconstruction with the least amount of information loss. Here we discuss the standard autoencoder and VAE.

- A standard autoencoder can be constructed using either dense or convolutional layers. A dense autoencoder uses dense layers in the encoder and decoder and this type of network

is sometimes referred to as a fully connected network. It requires a one-dimensional dataset as an input. A convolutional autoencoder is a model where information is passed through a series of convolutional layers for feature extraction. These are suitable for high-dimensional datasets.

- The difference between a standard autoencoder and a VAE is that the latent space for the VAE is constructed from a normal distribution. Data points are sampled randomly from a probability distribution in VAEs. The loss function used for VAE is the combination of the reconstruction loss and the divergence loss. The divergence loss is the Kullback-Leibler (KL) divergence which measures how the probability distributions differ from each other and the reconstruction loss can be any of the regression losses. There is a parameter Lambda (λ) that is used to control the regularization term in the loss function. The regularization, in turn, makes sure that the encoded latent space distribution matches the desired probability (Gaussian) distribution. The equation below summarizes the loss of a VAE.

$$L(x, \hat{x}, z_\mu, z_\sigma) = L_{\text{reconstruction}}(x, \hat{x}) + L_{\text{divergence}}(z_\mu, z_\sigma). \tag{2.19}$$

U-Net

A U-Net is a U-shaped neural network that consists of a downsampling path and an upsampling path which are symmetric. It was first used to process biomedical images (Ronneberger, Fischer, and Brox, 2015). This convolutional network is capable of performing image segmentation and assigning class labels pixel by pixel. One advantage of a U-Net is that it does not require a very large dataset to train, it uses data augmentation. A U-Net is a generic architecture and it can be used as a generative or a deterministic model. The deterministic nature of a U-Net will be explored in detail in Chapter 3.

Figure 2.15 shows the architecture for the first U-Net where the downsampling path is shown on the left (1st half of the network) and the upsampling path (second half of the network) on the right. The Downsampling path resembles a typical CNN and it has a double convolution followed by ReLU and maxpooling. The number of features at each downsampling step is doubled. The upsampling path consists of upsampling of feature maps, with double up-convolution that reduces the number of feature channels in half. There are skip connections

(concatenation) of the cropped feature maps from the downsampling path corresponding to layers in the upsampling path.

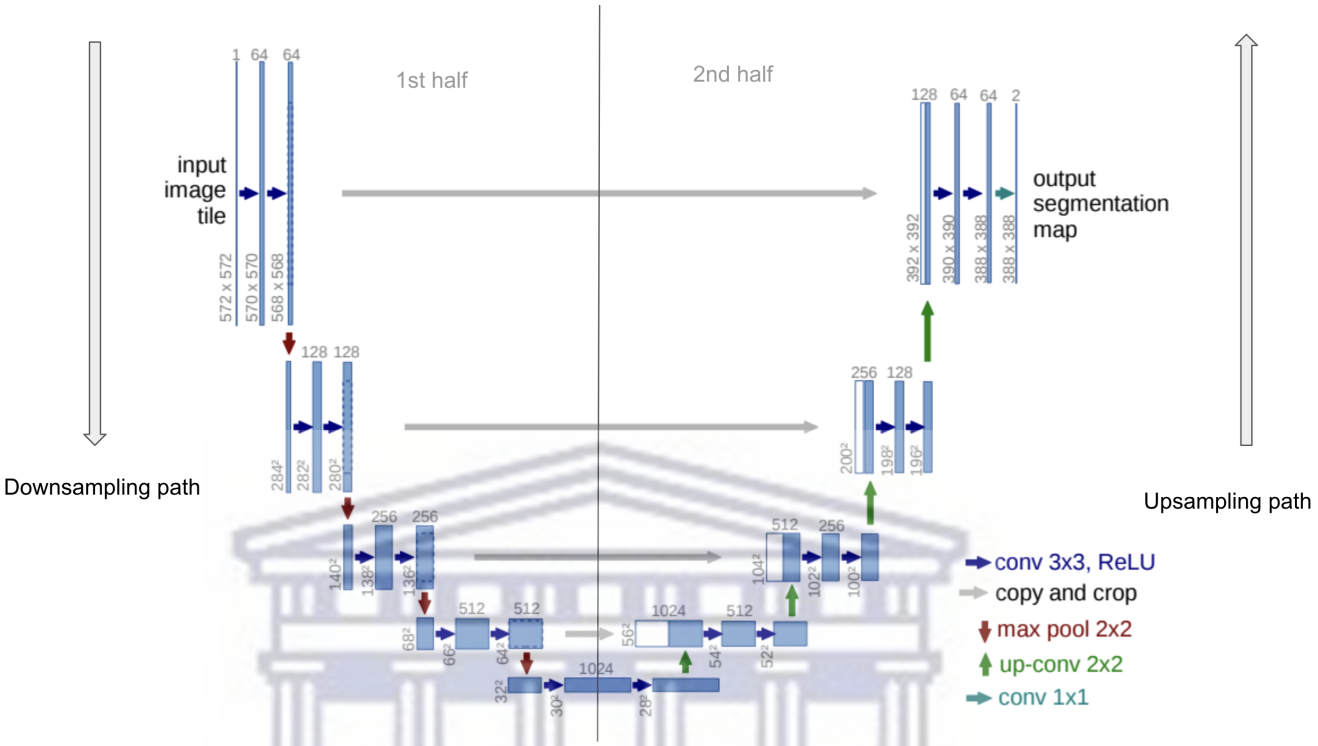

FIGURE 2.15: A U-Net structure adapted from Ronneberger, Fischer, and Brox, 2015 showing the different layers and connections from the contracting and expansive paths.

To this end, we have described the basic ingredients and flow of DL modeling, with a particular focus on generative models. Next, we describe how these techniques are used to accelerate reionization simulations.

3 Accelerating Reionization Simulations with Machine Learning

The work presented in this thesis focuses on accelerating reionization simulations using machine learning. In particular, we focus on accelerating SimFast21, a semi-numerical simulation of reionization, which we describe in more detail in this Chapter. We describe how the training data is generated and preprocessed. We also discuss the training and testing strategies of three different generative models used to generate density fields from ionization fields. This chapter is organized as follows; Section 3.1 discusses our reionization simulations using SimFast21, and Section 3.2 explains our strategy to accelerate SimFast21. The training dataset is described in Section 3.3, and Section 3.4 details the training and testing strategies of our generative models.

3.1 Simulations

Reionization simulations provide deep insights into the complex interplay between the ionizing sources (e.g. stars, galaxies) and sinks (e.g. absorption by neutral hydrogen) over cosmological scales. Ideally, large-scale reionization simulations should resolve the evolution of sources from sub-kpc to hundreds of Mpc volumes to track the evolution of ionizing radiation from the Interstellar medium (ISM), through the Circumgalactic Medium (CGM), and up to the Intergalactic Medium (IGM). There are different ways to simulate reionization, including radiative transfer hydrodynamic simulations (e.g. Rosdahl et al., 2018, Kannan et al., 2021, Bird et al., 2022) and semi-numerical simulations (e.g. Mesinger, Furlanetto, and Cen, 2010, Santos et al., 2010). Radiative transfer hydrodynamic simulations are the most accurate and detailed reionization simulations. These types of simulations include radiative transfer algorithms to model the evolution and radiation field. The algorithms take into account several processes involved such as scattering, emission, and absorption but they are computationally expensive. Semi-numerical simulations combine numerical and some approximation techniques for efficient simulation

of reionization. They involve an approximated radiative transfer scheme and use analytical techniques to evolve the density field. These simulations are capable of producing similar ionization morphology to full radiative transfer simulations (Majumdar et al., 2014; Molaro et al., 2019). 21cmFast (Mesinger, Furlanetto, and Cen, 2010) and SimFast21 (Santos et al., 2010) are semi-numerical codes that are capable of modeling cosmological 21cm signal during the EoR. These codes are comparable and in this work, we focus on SimFast21 which is described next.

3.1.1 SimFast21

SimFast21 is a semi-numerical code used to evolve reionization and generate the 21cm signal on large scales. This model has been first introduced in Santos et al. (2010). We here briefly describe the main framework of SimFast21. An initial density field will be generated in the linear regime using a Gaussian distribution, which then will be evolved to the nonlinear regime. This is done based on the second-order linear perturbation theory that is specified by the Zel'dovich approximation (Zel'dovich, 1970) following an analytic model of structure formation (White, 2014). The second step is the identification of dark matter halos, using the well-known excursion set formalism (Bond et al., 1991). An excursion set formalism (Furlanetto, Zaldarriaga, and Hernquist, 2004) will be applied to generate the ionization field out of the density and source fields. The source model used to generate the dataset accounts for ionizing emissivity from star-forming galaxies, where ionizing photons per second (R_{ion}) is modeled from a combination of large-volume hydrodynamic galaxy formation simulations (Davé et al., 2013) and high-resolution full radiative transfer hydrodynamic simulations (6/256-RT; Finlator et al., 2015) as follows:

$$\frac{R_{ion}}{M_h} = A_{ion}(1+z)^{D_{ion}}\,(M_h/B_{ion})^{C_{ion}}\exp\left(-(\mathrm{B_{ion}/M_h})^3\right), \tag{3.1}$$

where A_{ion}, C_{ion}, and D_{ion}, M_h, and z represent the ionizing emissivity amplitude, halo mass power-law index, redshift evolution index, halo mass, and redshift respectively. Using the 6/256-RT simulation, the amount of recombining neutral atoms per second (R_{rec}) is parameterized as follows:

$$\frac{R_{rec}}{V} = A_{rec}(1+z)^{D_{rec}}\left[\frac{(\Delta/B_{rec})^{C_{rec}}}{1+(\Delta/B_{rec})^{C_{rec}}}\right]^4, \tag{3.2}$$

where Δ is the overdensity, and V is the spherical volume. The values for $A_{rec} = 9.85 \times 10^{-24} cm^{-3} c^{-1}$, $B_{rec} = 1.76$, $C_{rec} = 0.82$ and $D_{rec} = 5.07$. For more information on the two simulations mentioned and the parameterization of R_{ion} and R_{rec}, we refer the interested reader to Hassan et al., 2015, and Hassan et al., 2016.

For a region to be ionized, it must meet the criteria that the product of R_{ion} and the photon escape fraction (f_{esc}) be greater than or equal to R_{rec}, where f_{esc} is the number of ionizing photons that escape a galaxy into the IGM. This ionization condition reads:

$$f_{esc} R_{ion} \geq R_{rec}. \tag{3.3}$$

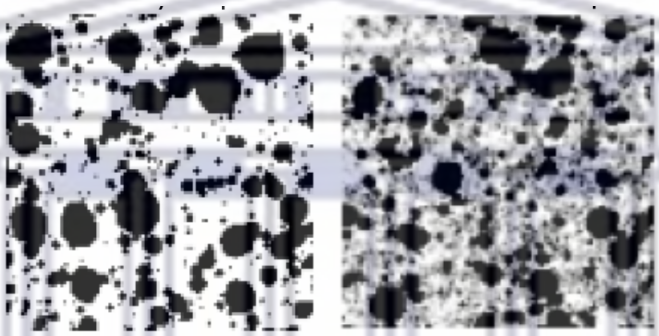

FIGURE 3.1: A comparison between a binary (left) and continuous ionization map (right) from SimFast21.

The vanilla SimFast21 generates ionization maps with binary values. Once the ionization condition is satisfied, fully ionized cells are assigned the value $x_{\rm HII} = 1$, and otherwise $x_{\rm HII} = 0$ for fully neutral cells. However, recombination is expected in fully ionized regions which would result in the presence of residual neutral fraction HI. Hassan et al. (2015) introduced a recipe to account for this residual neutral fraction to allow for low ionized fractions in post-reionization following a homogeneous ionizing background, which then results in a continuous map. Figure 3.1 shows a comparison between a binary (left) and continuous (right) ionization map.

3.2 Accelerating SimFast21 with Machine learning

In Chapter 1, we have discussed several upcoming 21cm experiments such as SKA that are expected to go beyond the PS and produce a full 3D mapping of the 21cm signal. SKA will also produce huge amounts of data, hence we must develop new codes and simulations to efficiently analyze these upcoming data.

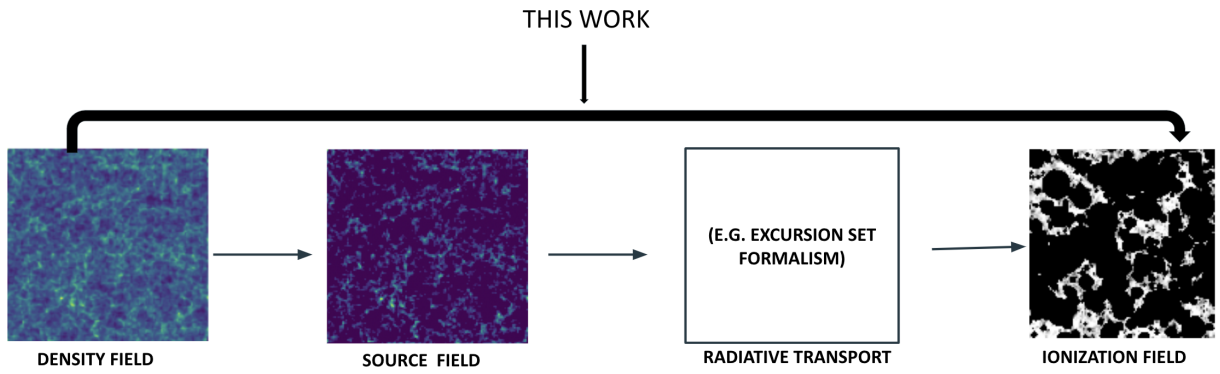

FIGURE 3.2: The basic workflow of all reionization simulations (e.g. SimFast21) is as follows: generating the initial density field (left), identifying sources (middle left), and applying radiative transfer (middle right) to obtain the ionized bubbles (right).

Semi-numerical simulations are the leading candidates to evolve reionization on large scale but generating high dimensional datasets from these simulations is computationally expensive, particularly for training CNNs to perform parameter recovery. There are some ML techniques that are capable of accelerating the generation of these data such as generative models including autoencoders and U-Nets. The most expensive step in these simulations is the radiative transfer module which generates the large-scale ionized bubble distributions. For instance, as discussed in the introduction, Chardin et al. (2019) have used an autoencoder as a generative model to simulate reionization using both gas density and star particle number density. In this thesis, we aim to accelerate SimFast21, the semi-numerical model described above, by skipping the source identification (halos/galaxies...etc) and radiative transfer step. We aim to directly generate the ionization maps from the initial density field without using the sources location nor following the detailed radiative transfer process. Figure 3.2 above shows a basic workflow of all reionization simulations (e.g. SimFast21) starting with the density field generation, through source identification, and then generating ionized field via radiative transfer. The arrow on top indicates the approach to be taken in this thesis to accelerate SimFast21.

3.3 Training Datasets

The training datasets are generated using SimFast21, a semi-numerical code described in Section 3.1 above. All data are generated using a box size of 250 Mpc with a number of voxels of 128^3. We adopt the ΛCDM model with the following cosmological parameters, the Hubble parameter h=0.7, matter density Ω_m=0.3, baryon density Ω_b=0.045, density parameter for dark

energy Ω_Λ=0.7 and matter fluctuation amplitude σ_8=0.8, consistent with Planck Collaboration et al. (2018).

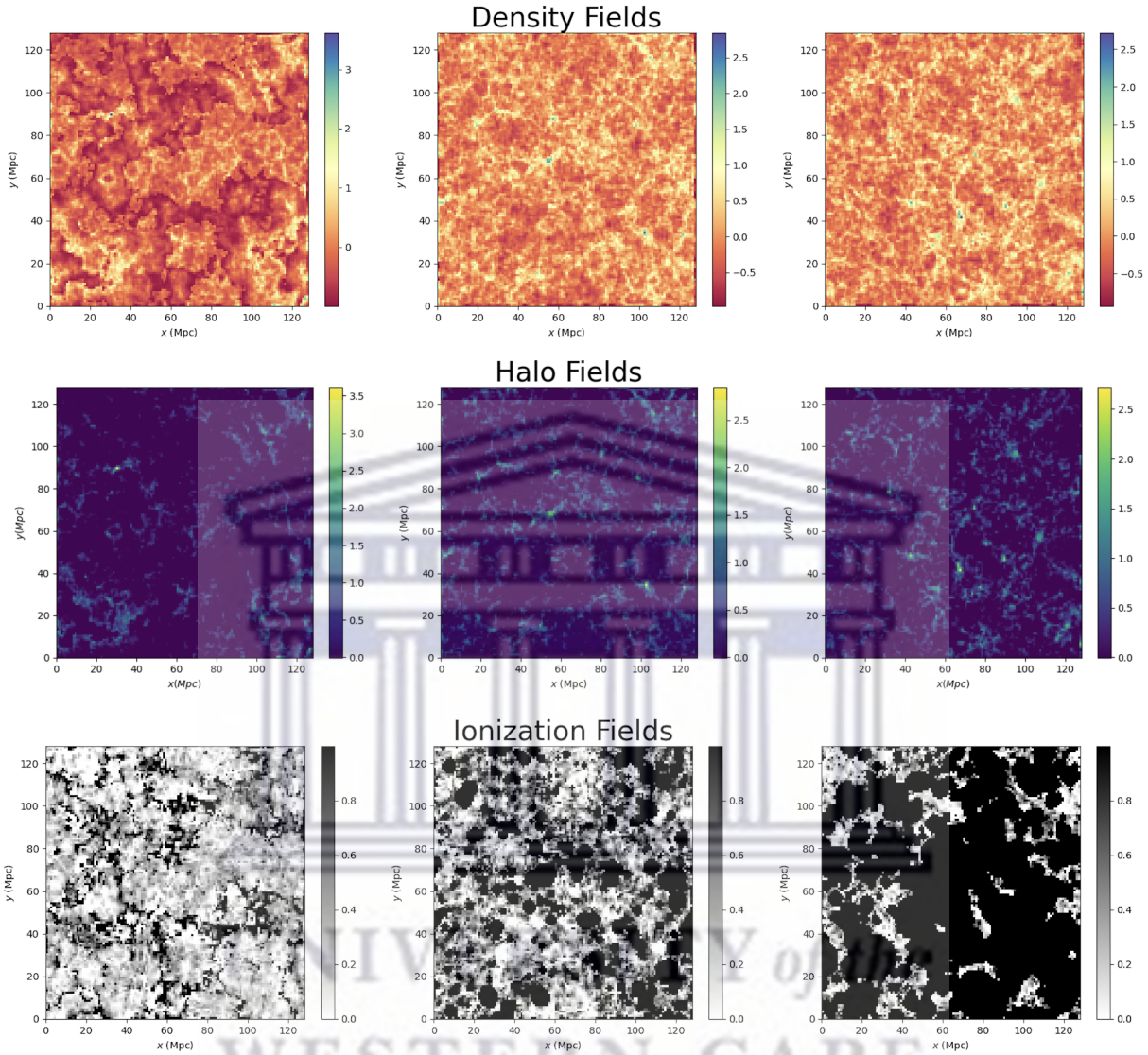

FIGURE 3.3: Random examples of the density fields in the top row, halo fields in the middle, and ionization fields in the bottom row. The halo and ionization fields shown are generated from the density fields shown. The first column shows the generated ionization field with a fraction that is ~ 0.3, the second column ~ 0.5, and the third column ~ 0.8.

To create a diverse and balanced dataset that contains a variety of density fields and their corresponding ionization fields we varied several free parameters of the ionization rate $R_{\rm ion}$ such as the photon escape fraction ($f_{\rm esc}$), ionizing emissivity amplitude ($A_{\rm ion}$) and the halo mass power index ($C_{\rm ion}$) which controls the size of the ionized bubbles over the following uniform prior range $f_{\rm esc} = 0 - 1$, $\log_{10} A_{\rm ion} = 38 - 42$, and $C_{\rm ion} = -2 - 2$.

Figure 3.3 shows random samples of images from the training dataset. The different density

fields are shown at the top row, with the source (halos/galaxies) fields shown in the middle row, and the ionization field at the bottom row. The ionization fields range from low ionization fraction (left map, $x_{\rm HII} \sim 0.3$) to high ionization fraction (right map, $x_{\rm HII} \sim 0.8$).

The generated dataset consists of 15,600 images of density fields and their corresponding ionization fields. The dataset was split into two sets with a ratio of 9:1 for training and testing respectively. We then perform all the analysis using ionization maps corresponding to the mid-point of reionization, with $0.4 < x_{\rm HI} < 0.6$. We choose to test the different methods using realizations of the mid-point of reionization to maximize the non-linearity due to the diverse bubble sizes at this limit. In highly ionized or highly neutral Universe, the maps contain less information and less diversity of the ionized bubble distributions. While choosing this limit is somewhat arbitrary, the analysis presented here is a proof-of-concept to test the viability of using generative models to emulate the radiation transport on cosmological scales. This subsample was augmented (both the training and testing set were increased by creating copies of the existing data) by flipping and rotating the data at different angles in both the y and x directions (y and x-axis). The augmented data were concatenated with the existing data to form new training and testing datasets. The datasets were normalized to values in the range between 0-1.

3.4 Generative Models

In the previous chapter, we have discussed in detail the components that make up a neural network, their importance, and functionalities. These include activation functions to add non-linearity to the network, loss functions to quantify the error between the predicted and true values, optimizers to minimize the loss error, and regularisation techniques to prevent overfitting.

A successful ML model requires careful and extensive training and hyperparameter tuning. We seek the optimal architecture and the best hyperparameters to design a powerful and accurate model that will be successful in solving our problem. We have explored different techniques to generate ionization fields directly from density fields. The two main generative architectures used are autoencoders and U-Nets. These models are suitable for our problem and they are able to support the nature of our data. Among the architectures that we explored were a dense

autoencoder, a convolutional autoencoder, a VAE, a basic U-Net, and a denoising U-Net. Here, we show the architectures and tables to summarise the models used to generate ionization fields directly from density fields. The architectures for the dense and VAE are omitted due to their low performance as compared to other models presented here.

All our codes are written using TensorFlow (Abadi et al., 2015). TensorFlow is an open-source Python library used in academia and commercially. TensorFlow was developed by Google in 2015 and modified in 2019 and now works in conjunction with Keras (Ketkar, 2017), an Application Programming Interface (API). In this project, we used TensorFlow 2.0 (2019 version). We train our models using a single NVIDIA Tesla V100-32GB SXM2 GPU on Bridges-2. Bridges is a high-performance supercomputing facility from the Pittsburgh Supercomputing Cluster (PSC).

3.4.1 Convolutional Autoencoder

CNNs have been known to be very efficient in designing ML models that work on images with complex features. In our attempt to generate the ionization fields directly from the density fields, we have used the convolutional autoencoder whose architecture is summarised in Figure 3.4 and Table 3.1. Figure 3.4 shows a convolutional autoencoder architecture with the input as the density field, encoder, latent space, decoder, and output as the ionization field.

The input to the autoencoder is a 2D array of the density field with a 128 x 128 dimension. The encoder has two 3x3 convolutional layers, with filters 64 and 32 applied to the first and second layers, respectively. A stride of two and the "same" padding is applied to these convolutional layers. The padding used allows more space for the kernel to cover the image. Using "same" padding ensures that the filter is applied to all elements of the input by extending the area of an image. The stride parameter depends on the number of parameters such as the network architecture, the trade-off between spatial resolution and computation efficiency, and the problem the network is solving. Setting the parameter to two works best for our networks. The regularization technique we used here is Batch Normalization, and this layer is followed by a LeakyReLU activation function layer, with an alpha value of 0.2. The latent space contains a dense layer with 16 units. Batch Normalization and LeakyReLU are also applied to this layer.

The decoder and encoder are symmetric. The decoder layer consists of two 3x3 convolutional operations (Conv2DTranspose layers) for upsampling. As with the convolutional operations in the encoder, the filters are 32 and 64, and the stride is set to 2. The Conv2DTranspose layers are followed by Batch Normalization and LeakyReLU layers. The output layer has no regularization technique applied to it and sigmoid is used as the activation function. The output is a 2D array of the ionization field with dimensions 128 x 128.

Layer	Size
Input	
Input Layer	128 x 128 x 1
Encoder	
Conv2D layer1 BatchNormalization LeakyReLU	64 filters
Conv2D layer2 BatchNormalization Leaky ReLU	32 filters
Latent Space	
Dense layer BatchNormalization Leaky ReLU	16 units
Decoder	
Conv2DTranspose layer 2 BatchNormalization Leaky ReLU	32 filters
Conv2DTranspose layer 3 BatchNormalization Leaky ReLU	64 filters
Output Layer	
Conv2DTranspose layer Sigmoid	128 x 128 x 1

TABLE 3.1: The summary for the architecture of our convolutional autoencoder. It consists of two layers in both the encoder and decoder and has a latent space with a size of 16 units.

Our model uses Adam optimizer with a learning rate of 10^{-4} and the MSE loss function. The autoencoder has a total of about 1,129,553 parameters to learn, where 1,129,169 of these parameters are trainable, and about 384 are non-trainable. Trainable/learnable parameters refer to the values that are optimized during training to make the model perform better. These include weight and biases. Non-trainable/non-learnable parameters are what we call hyperparameters and they are set before training could take place. These are adjusted by the user as needed (trial and error). Non-learnable parameters include the number of layers, learning rate, and activation function, just to name a few. The model is trained for 200 epochs, and training takes approximately seconds per epoch. Using a batch size of 16, the model trains for approximately 20 minutes.

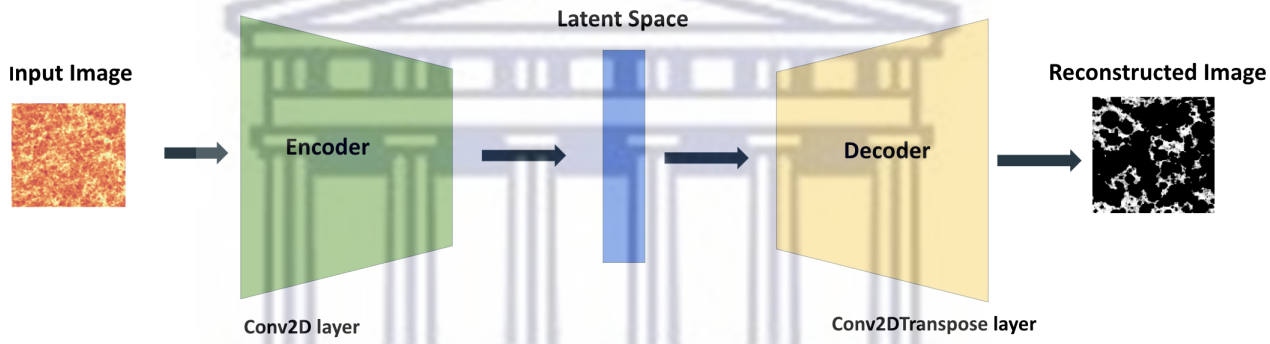

FIGURE 3.4: Network for the convolutional autoencoder.

3.4.2 U-Nets

Two other generative models we are using to generate ionization fields directly from density fields are a basic and denoising U-Net. These models have the same architecture with the same number of layers in both the downsampling and the upsampling path. The difference between these two models is in the input. The basic U-Net uses only the density field as an input while the denoising U-Net considers a two-channel input, namely a density field and either a white noise or corrupted version of the ionization field during training. White noise is a type of Gaussian noise that is determined by the mean and standard deviation, it follows a Gaussian distribution. The corrupted versions of the data were simulated by adding white noise to the ionization fields. The type of noise used during training is sampled randomly with a 50/50 probability.

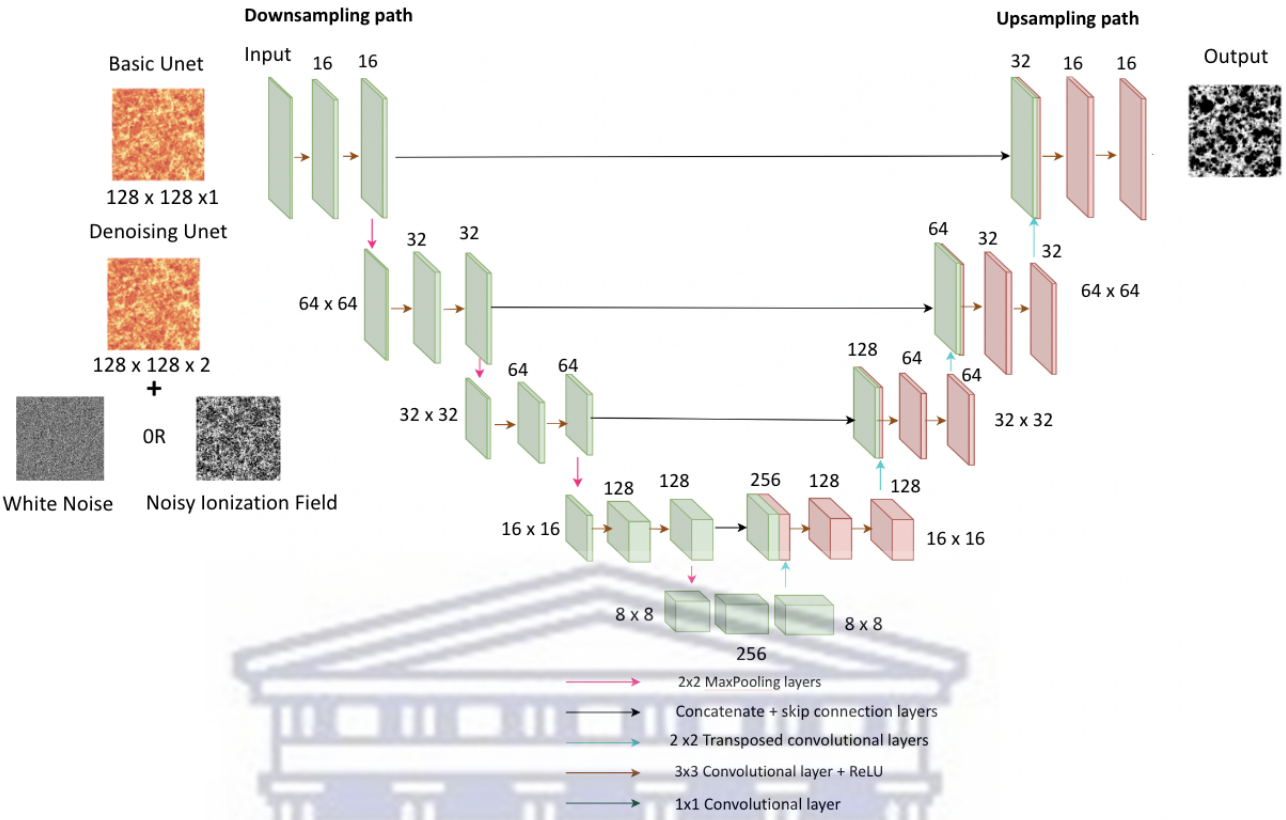

FIGURE 3.5: A visual summary of the U-Net architecture showing the two different strategies for training using either a single input for the basic version or two inputs for the denoising model.

Figure 3.5 shows the two versions of the U-Net and the main architecture is summarised in Table 3.2. The downsampling (Encoder) path consists of four main layers with 3x3 double convolutions, where each convolution is followed by Batch Normalization and ReLU activation function. Max pooling is applied at the output of each double convolution. The first double convolutional layer has a filter of 16. Two filters are applied in the next layer to double the number of features. If the filters in the two first layers are 16, they are then doubled to be 32 in the next layer, and so on. The filters used for the four layers are of sizes 16, 32, 64, and 128.

The bridge connects the downsampling and upsampling paths to control the flow of information between these two. It consists of a 3x3 two-convolutional layer with a filter of 256, followed by Batch Normalization and ReLU activation function. The upsampling path of the U-Net consists of four main layers with each layer having a convolutional operation Conv2DTranspose with a 3x3 double convolutional layer to reduce the number of features in half. The filters used are 128, 64, 32, and 16. The downsampling and upsampling paths are symmetric, with a concatenation layer that passes the information from the downsampling to the upsampling

in the form of cropped feature maps. Batch Normalization and ReLU are applied after each convolutional layer. The output of both U-Net architectures is the ionization field.

Layer	Size
Input Layer	
Input	128 x 128 x 1
Downsampling	
Conv2D layer1	16 filters
BatchNormalization	
ReLU	
Conv2D layer2	16 filters
BatchNormalization	
ReLU	
MaxPooling2D	
Conv2D layer3	32 filters
BatchNormalization	
ReLU	
Conv2D layer4	32 filters
BatchNormalization	
ReLU	
MaxPooling2D	
Conv2D layer5	64 filters
BatchNormalization	
ReLU	
Conv2D layer6	64 filters
BatchNormalization	
ReLU	
MaxPooling2D	
Conv2D layer7	128 filters
BatchNormalization	
ReLU	
Conv2D layer8	128 filters
ReLU	
BatchNormalization	
MaxPooling2D	
Bridge	
Conv2D layer9	256 units
BatchNormalization	
ReLU	
Conv2D layer10	256 filters
ReLU	
BatchNormalization	

Upsamling	
Conv2DTranspose layer 1	128 filters
Concatenate	
Conv2D layer11	128 filters
BatchNormalization	
ReLU	
Conv2D layer12	128 filters
ReLU	
BatchNormalization	
Conv2DTranspose layer 2	64 filters
Concatenate	
Conv2D layer13	64 filters
ReLU	
BatchNormalization	
Conv2D layer14	64 filters
BatchNormalization	
ReLU	
Conv2DTranspose layer 3	32 filters
Concatenate	
Conv2D layer15	32 filters
BatchNormalization	
ReLU	
Conv2D layer16	32 filters
BatchNormalization	
ReLU	
Conv2DTranspose layer 4	16 filters
Concatenate	
Conv2D layer17	16 filters
BatchNormalization	
ReLU	
Conv2D layer18	16 filters
BatchNormalization	
ReLU	
Output Layer	
Conv2DTranspose layer	128 x 128 x 1
Sigmoid	

TABLE 3.2: The summary for the architecture of a U-Net downsampling path on the left and the upsampling path of a U-Net.

The total number of learnable parameters of the basic U-Net is 1,946,705, while for the denoising U-Net model is 1,946,849, and only 2,944 are not learnable in both cases. Both models are

trained for 200 epochs to observe a plateau in the loss evolution. We use Adam to optimize the models and the MSE as a loss function. The learning rate for both U-Nets is set to 10^{-3} and the basic U-Net is trained using the batch size of 16 while the denoising U-Net's batch size during training is 64. The denoising model was trained iteratively (using a for loop). Since the noise sampling process is random, during the iterative process the network has a 50/50 chance of choosing either the white noise or the noisy version of the ionization.

3.4.3 Testing protocol for the denoising U-Net model

This subsection presents the testing strategies for our three best-performing models, with more focus on the denoising U-Net model since it performs better than the autoencoder and the basic U-Net. The testing protocol for the denoising U-Net is an iterative process.

- **Autoencoder and basic U-Net** take a single input, namely the density field, to generate the ionization field. In this case, the method is deterministic and there exists a unique solution for each input. Hence, a single forward pass of the input through the networks is sufficient for testing.

- **Denoising U-Net** takes two inputs, namely the density field and either a white noise or a noisy version of the ionization field during training, to generate the ionization field. In this case, the method is stochastic and there exist infinite solutions for each input, depending on the initial random seed. During testing, we first feedforward a white noise next to the density field. We then feedforward the output into the network to generate the second output. We continue to feedforward the output recursively till convergence in the loss evolution is achieved and the predicted power spectrum achieved is similar to the target.

In Figure 3.6, we show the recurrent testing approach used for the denoising U-Net model. The recursive testing is implemented using the following equation: $y_n = \alpha y_{n-1} - (1-\alpha)\mathrm{F}\,(x, y_{n-1})$, where F is the trained denoising U-Net model, x is the density field and y_{n-1} is the previous prediction, and the regularisation parameter α is set to 0.4 to prevent ionized bubble overgrowth. The number of iterations is controlled by α and we stop the iterations when the PS for a specific iteration is similar to that of the target. We will discuss this in more detail in Chapter 4.

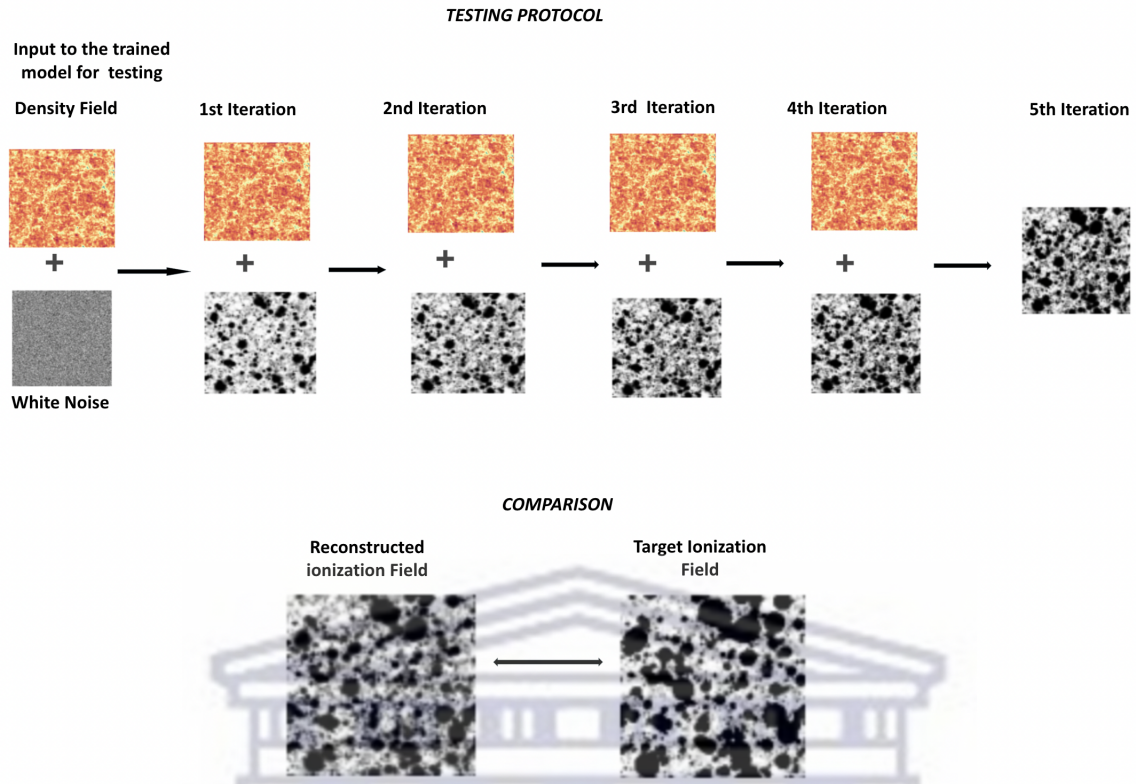

FIGURE 3.6: Iterative testing for the denoising model where it takes five iterations to generate ionization maps that are virtually similar to the testing set

To this end, we have presented in detail the training and testing strategies for our generative models to emulate the radiative transfer and accelerate SimFast21. Next, we present our key results.

4 Results and Discussion

This chapter presents the results from our three best-performing models, which are the convolutional autoencoder, basic U-Net, and denoising U-Net. We present the results in two sections, where Section 4.1 focuses on the results using continuous ionization maps and Section 4.2 binary ionization maps. The performance of the three models is examined using the loss evolution, visualization of the target and predicted maps, the power spectra, and the bubble size statistics.

4.1 Performance on Continuous Maps

Since the denoising U-Net is a probabilistic model and requires different testing protocol we will present this model first.

4.1.1 Denoising U-Net

- **Loss Evolution**

  As mentioned earlier, the loss function is an error metric used to evaluate the performance of training and testing. As with all the other models and as already mentioned in the previous chapter, we have used MSE as our loss function. During training the model sees the noisy/corrupted version of the ionization field and the loss for training converges around 100 epochs. Figure 4.1, shows that the loss function is almost zero, indicating that there is a minimal loss of information during the training of this model. The fluctuations in the validation loss are due to validating the model in a single iteration and using the white noise beside the density fields. There is no overfitting in the loss evolution as the validation loss is not increasing. The training loss is shown in a green solid line while the validation loss is shown in a dashed green line.

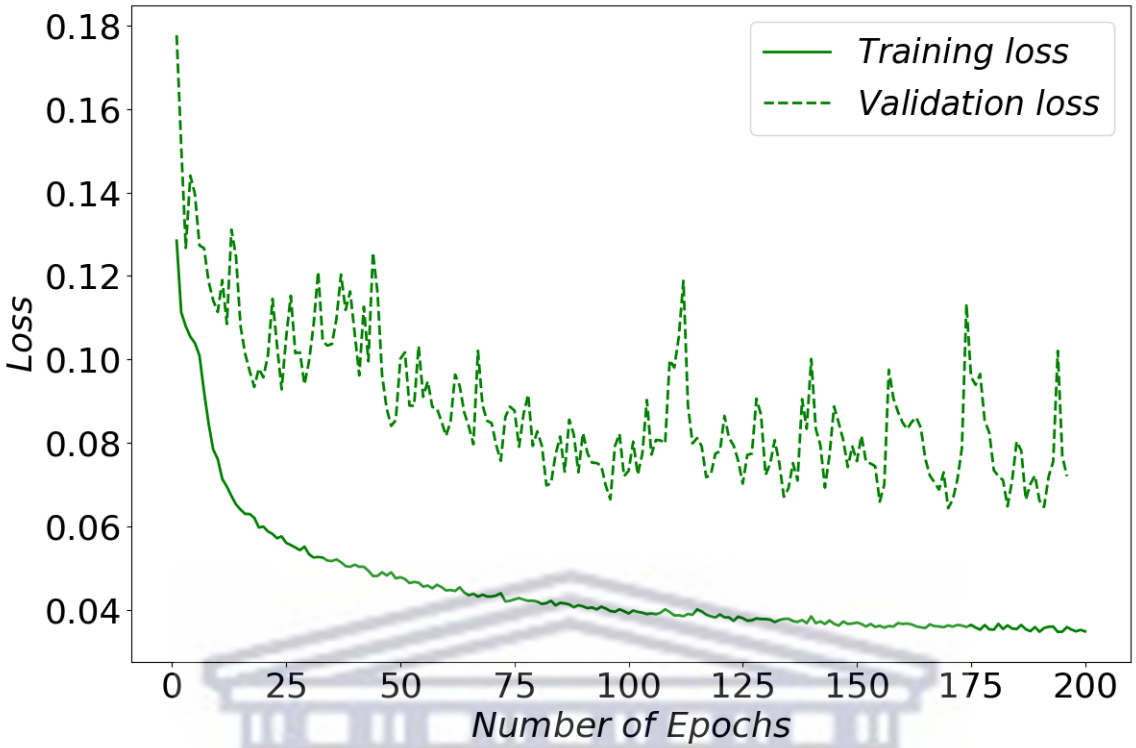

FIGURE 4.1: Loss evolution showing the training and validation loss of the denoising U-Net. Both losses decrease over training and testing epochs and convergence is achieved roughly after 100 epochs and the model shows no signs of overfitting.

- **Comparing models at the map level**

  Ionization maps contain small and large scale bubbles. Small-scale bubbles are ionization bubbles that are a few hundred comoving kiloparsecs (Kpc) across in size while the large-scale bubbles are typically several comoving megaparsecs (Mpc) across. Small-scale bubbles are the ionized regions around individual galaxies and large-scale bubbles refer to the overall structure and evolution of the ionized regions. Individual small-scale bubbles overlap and merge to form large-scale bubbles.

  The loss evolution gives us insight as to how well the model is performing but here we plot the images to see how well the bubbles are recovered in each case. Figure 4.2 below shows the evolution of the ionized bubbles as a function of iteration. During the testing process of the denoising U-Net, we first feed-forward the density field and white noise to obtain the ionization field. For the first iteration, the noise is replaced with the ionization field predicted in the first step. It takes about five iterations for the generated maps from the denoising U-Net to be similar to the target maps, by visual inspection.

Figure 4.2 shows the comparison between the original ionization maps and the iterations from the denoising U-Net model during testing. These are randomly selected samples from the testing set. As we have already mentioned that we start with the white noise, the first column in Figure 4.2 shows the white noise, while the 2nd, 3rd, 4th, and 5th column represents the denoising U-Net outputs for different iterations, and the last column represents the true ionization maps. Here we only plot the maps generated from the 1st, 3rd, 5th, and 6th iterations to show the denoising process over iterations. In summary, the iterative testing implemented for the denoising U-Net is able to produce ionization bubbles that are visually similar to the ones produced by the simulation (original/true ionization maps).

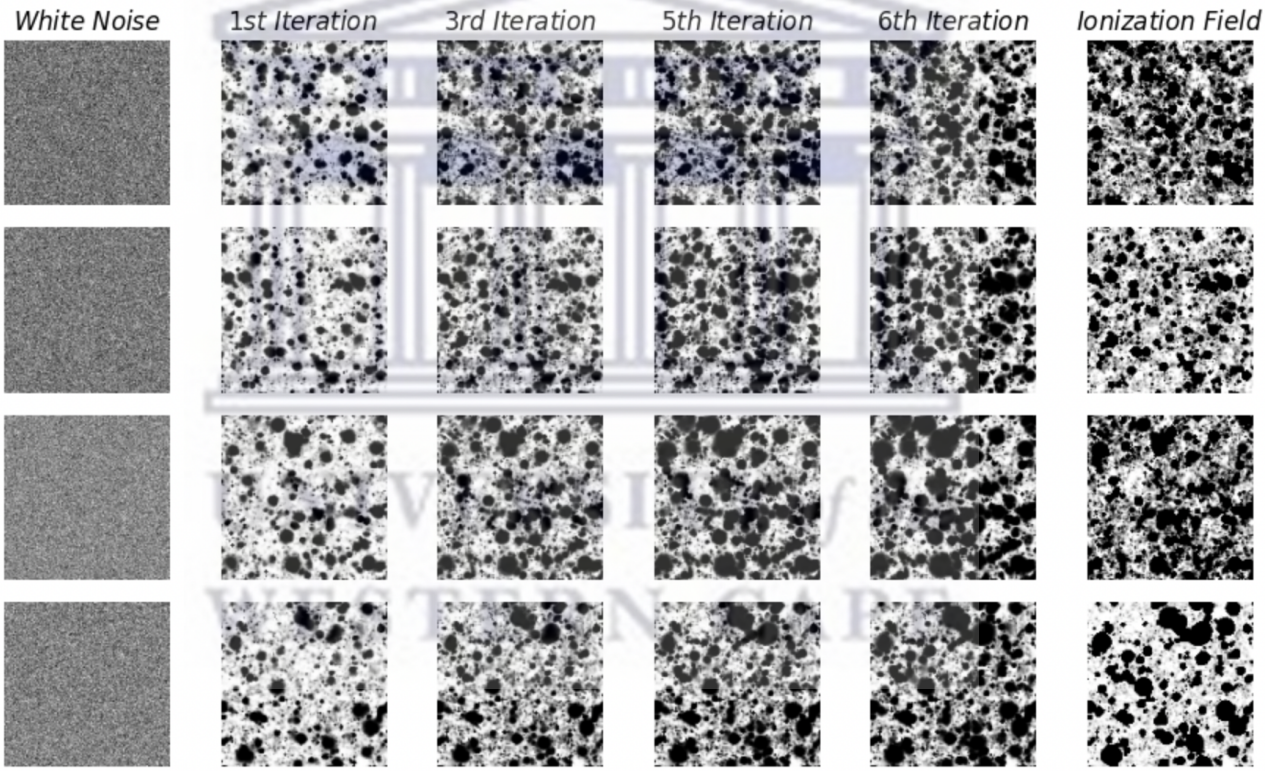

FIGURE 4.2: Visualization of randomly selected samples of the reconstructed ionization fields over iterations. By visual inspection, five iterations are sufficient to generate a similar ionization map to the target.

- **Comparing models on the level of Power Spectrum**

  We now turn our attention to evaluating the denoising U-Net over all the testing dataset using the PS as a summary statistic since it is the observable target for many of the current and upcoming 21cm surveys. Figure 4.3 shows the average power spectra as a function of wavenumber (k) for each iteration of the denoising U-Net model. The target PS is shown

in red, with the white noise in black, the first iteration in cyan, the third in orange, the fifth in purple, and the sixth in lime color. The PS approaches the target as the number of iterations increases. Here, we show that after 5 iterations the PS matches the target PS on large scale. On the sixth iteration, the PS surpasses the target. This model however is unable to recover the small-scale power where non-linearity is maximum. This non-linearity relationship at small scale is a contributing factor to the poor performance of the model since the model is not able to capture it. There are other factors that can result in a model not being able to capture the small-scale structures. These include the amount of data available, the resolution of the simulation or observation, and the accuracy of the simulation. SimFast21 uses some approximation of the radiative transfer scheme and this might also be one of the reasons why the model is not able to generalize well on small scales. The resolution of the simulation might also be a contributing factor since to capture the small-scale structure the resolution of the simulation must be high. From here on, in this subsection, the plots for the denoising U-Net model will be based on the 5th iteration.

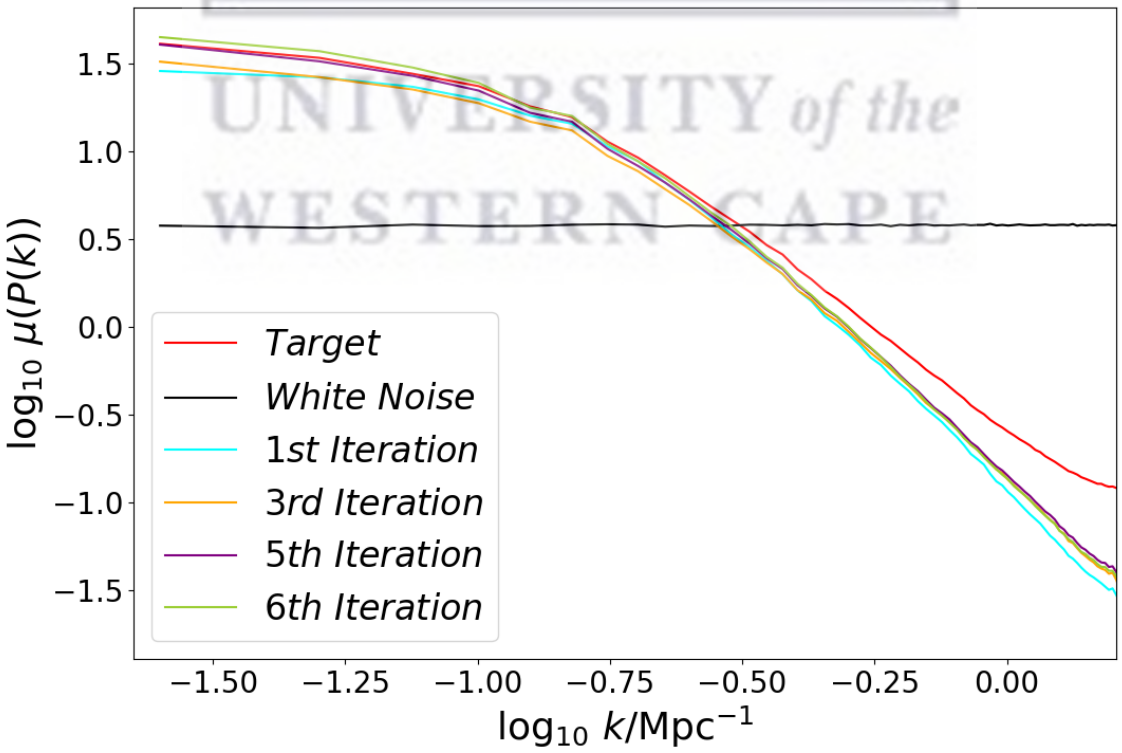

FIGURE 4.3: Evolution of the power spectrum from white noise to the reconstructed ionization field over iteration. Five iterations are sufficient to produce a similar large-scale power to the target.

4.1.2 Performance on all the models

- **Loss evolution**

  We now compare the MSE loss evolution between all models in Figure 4.4. We show the basic U-Net, Autoencoder, and Denoising U-Net in blue, green, and magenta colors, respectively. Solid and dashed lines represent the loss during the training and validation phases, respectively. The loss in the basic U-Net is lower compared to the loss for the autoencoder. Since the denoising U-Net is the best-performing model its loss is expected to be lower than that of the basic U-Net. As can be seen from the plot, denoising U-Net has the lowest loss, indicating better performance. In all the plots, the training loss decreased after each and every epoch and converged at about the 100th epoch, indicating that the models are able to generalize the training data. All losses converge and there is no sign of overfitting or underfitting in all the models. Because of the noisy ionization fields that were added during the training process of the denoising U-Net model, its loss is lower compared to the other two models.

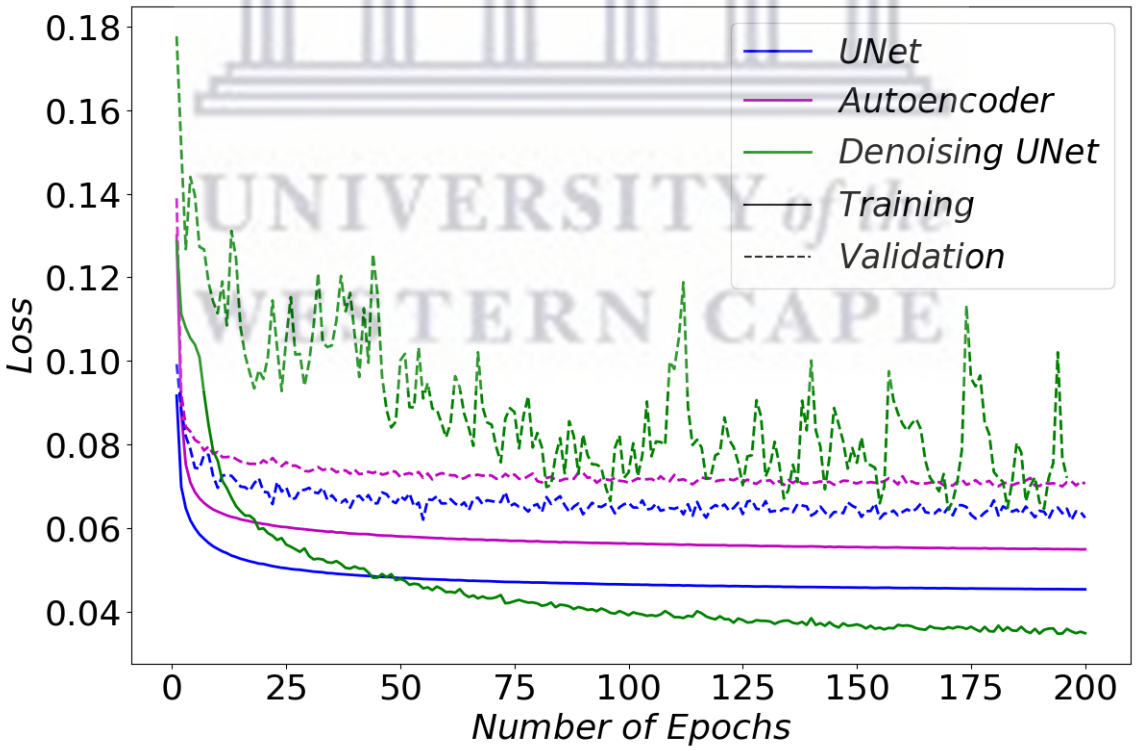

FIGURE 4.4: Comparison between the autoencoder (magenta), basic U-Net (blue), and denoising U-Net (green) in terms of the loss evolution over training epochs. This figure shows that the denoising U-Net has the lowest MSE loss, indicating the best performance

.

- **Comparing models at the map level**

  Figure 4.5 shows the random visualization from the testing set. The first column is the density field, the second column is the original ionization field, the third column is the reconstruction from the autoencoder, the fourth column is the reconstruction from the basic U-Net and the last column shows the reconstruction from the denoising U-Net.

  Both the autoencoder and the U-Net reconstruct similar ionization bubbles. These two models are unable to detect most of the large-scale bubble edges. Unlike the autoencoder and basic U-Net, the denoising U-Net model is able to detect them. Adding noisy ionization fields to the network during training and performing recurrent testing on the trained model allowed for a more refined way to recover the ionization bubbles. The edges of the bubbles can be detected and clearly identified in the generated ionization maps from the denoising U-Net.

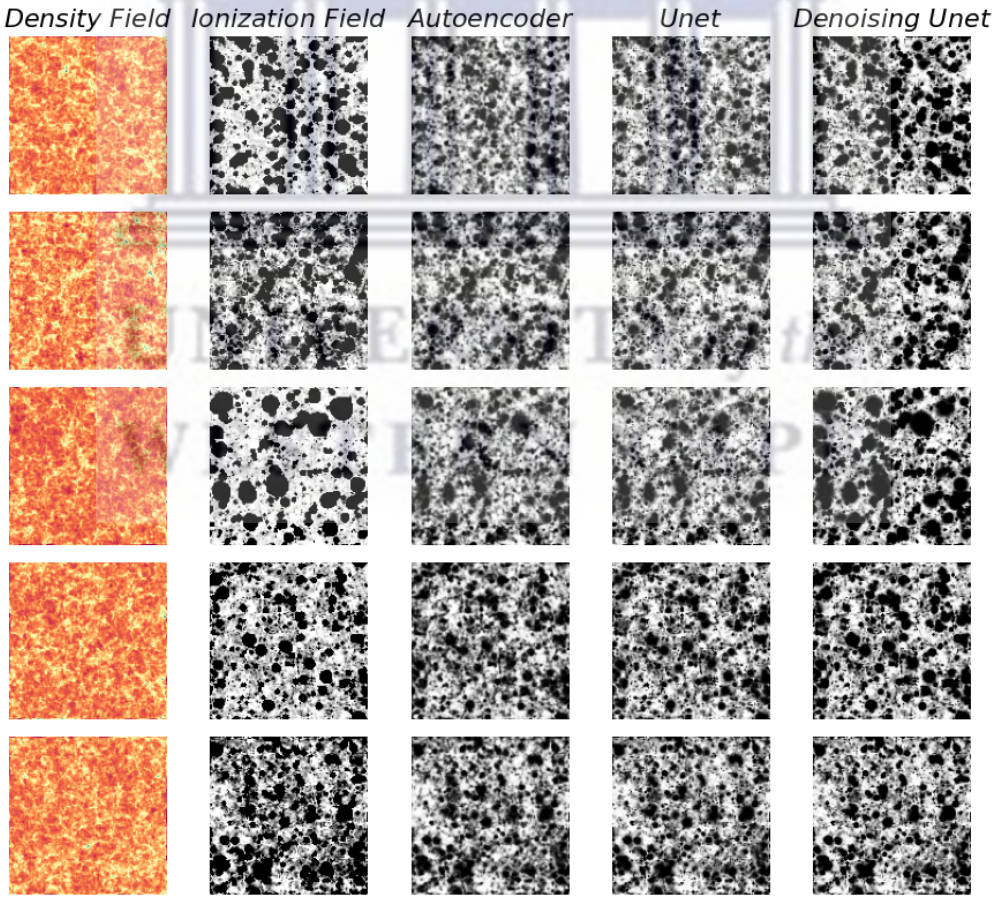

FIGURE 4.5: Visualisation of random maps from the testing set for all the models where the autoencoder and U-Net models construct the bubbles similar to each other, while the denoising U-Net constructs bubbles that are similar to the testing set.

- **Comparing models on the level of Power Spectrum**

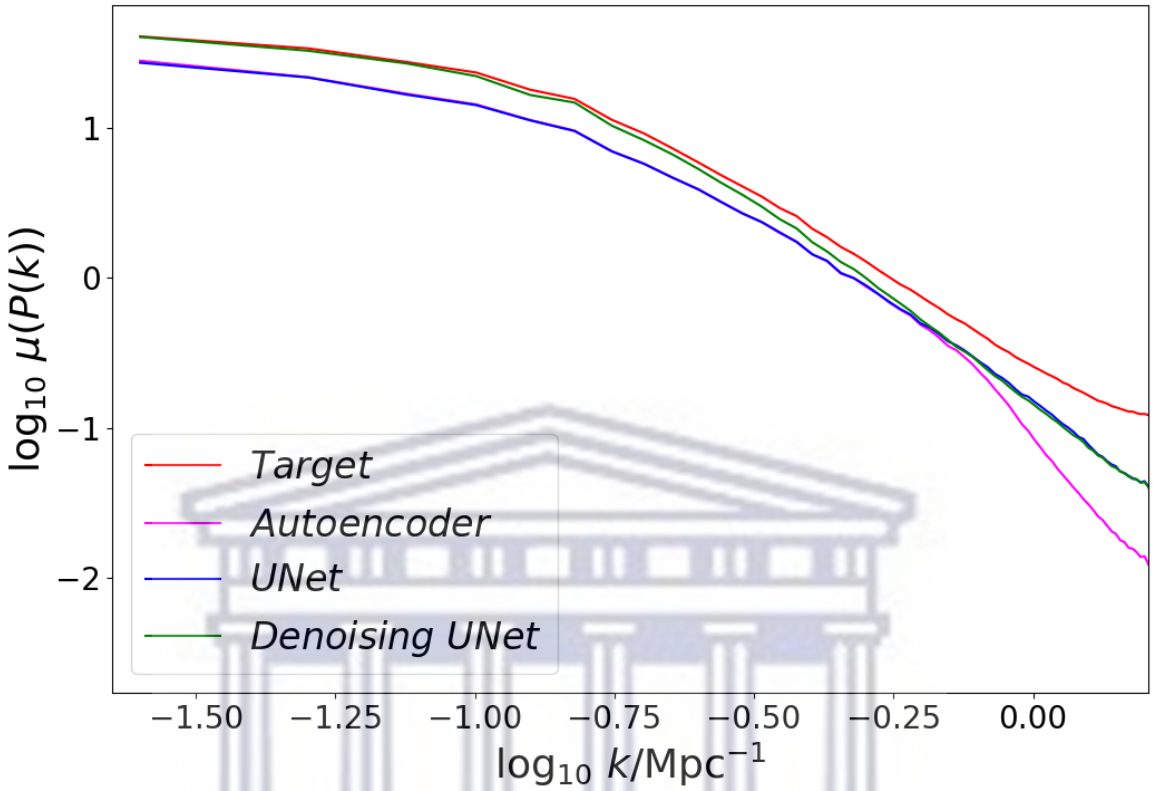

FIGURE 4.6: The PS from all the best performing models, the figure shows the improvements made by the denoising U-Net on large scale.

The figure above plots the power of the ionization field at different scales. The PS shown in Figure 4.6 is the mean power from all the testing data. The target, autoencoder, basic U-Net, and denoising U-Net spectra are shown in red, magenta, blue, and green, respectively. In the case of the denoising U-Net, the PS plotted was the average of the power after recurrent testing. Both the autoencoder and the U-Net models construct the large and intermediate scales similarly, the difference in the power spectra can be seen on small scale where the U-Net power is closer to the target power and the autoencoder power deviated from both the U-Net and the target spectra. The difference in the PS of the autoencoder and the basic U-Net is due to the fact that a U-Net model is able to assign labels pixel by pixel and the skip connection in the U-Net allows a model to retain more refined information. The PS of the denoising U-Net model showed a good agreement with that of the original maps on large scales, thus it performed better than the U-Net and the autoencoder, which both underestimated the power. Even though the denoising

model showed good agreement on a large scale, the power was underestimated on a small scales. This poor performance on small scale is expected since non-linearity is highest. However, this figure demonstrates that our denoising U-Net can still be suitable for studies that focus on scales larger than ~ 30 Mpc (k~0.2 Mpc^{-1}).

- **Comparing models in terms of the Bubble Size Distribution**

  The bubble size distribution (BSD) describes how many ionized regions of a given size exist in the data. There are several ways to describe the BSD that provide a largely similar representation of the bubbles. There are three well-known methods that can be used to define the BSD. The first method is Friends of Friends (FoF, Iliev et al., 2006) and it focuses on the connectivity of the ionized bubbles. Second, the Spherical-average method (Zahn et al., 2007) which focuses on the largest spherical volume that fits inside the distribution of ionized regions. Third, the mean free path (mfp) method (Mesinger and Furlanetto, 2007), which finds the distribution of the distances to the edges of an ionized region from a large collection of random points and directions. The bubble size distribution in this work was computed using Tools21cm (Giri, Mellema, and Jensen, 2020) using only the mfp method as a reference.

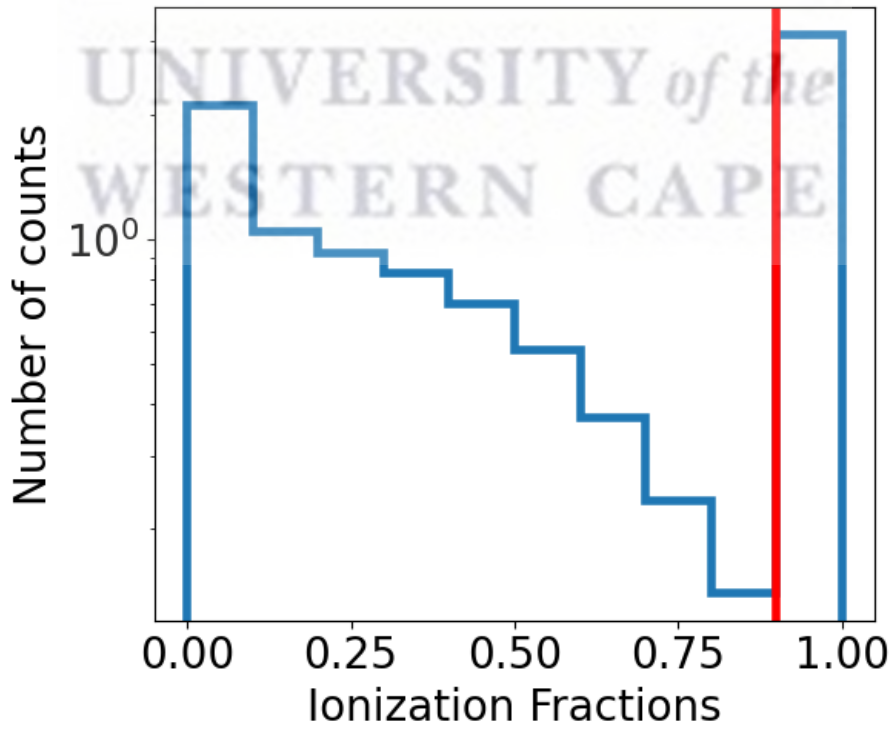

FIGURE 4.7: Histogram showing a number of bubbles within the training dataset as a function of their corresponding ionization fraction.

The transition from highly ionized to highly neutral is very sharp and rapid (Iliev et al., 2009). We have experimented with different ionization thresholds and have chosen 0.9 because most of the ionization fractions are above this limit. This can be observed in Figure 4.7, which shows the histogram of the ionization fraction from all pixels over all the training dataset. This figure shows that most of the ionization fractions are above this limit. For this reason, we set pixels that have an ionization fraction greater than 0.9 to unity and otherwise to zero. Figure 4.8 shows the BSD using the mfp method. To compute the BSD, the ionization maps are converted to binary. The BSDs for both the autoencoder and the basic U-Net are lower than the target, indicating that the bubbles in these models are smaller than in the original field. This further strengthened the results shown by the loss evolution, visualization, and PS.

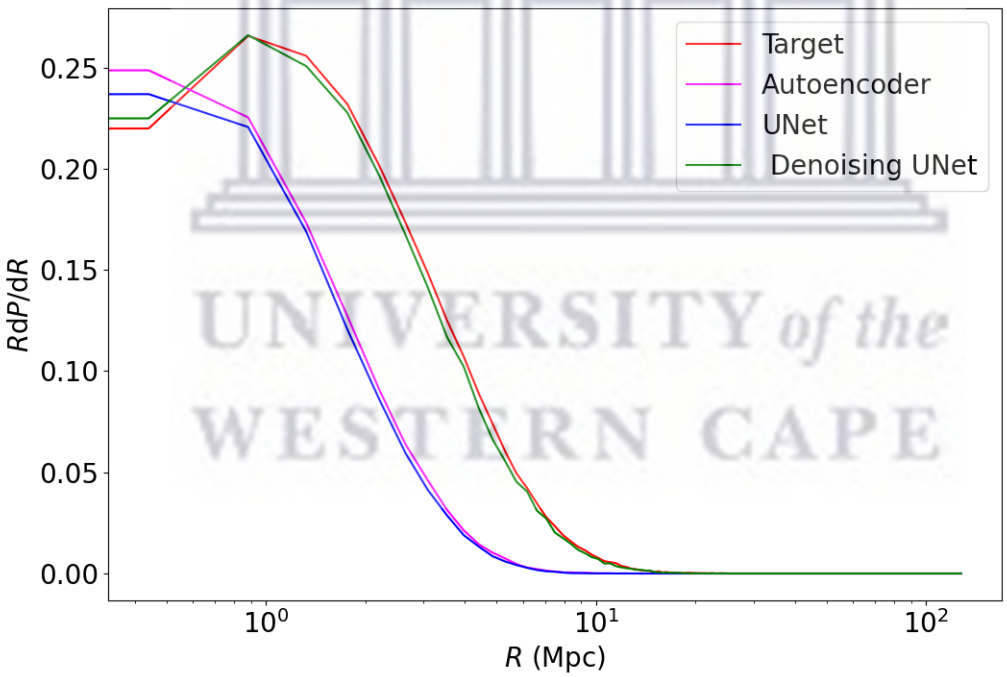

FIGURE 4.8: The BSD using the Mean free path (mfp) for all models, where the ionization threshold (xHII) is set to 0.9. The denoising U-Net model produces a similar BSD as the original ionization field

4.2 Performance on binary maps

As already mentioned in Chapter 3, Section 3.1.1 that SimFast21 can generate both binary and continuous maps. Here we present the results of the models when evaluated on binary maps.

The maps are converted to binary using the criteria described in the previous section. It is important to note that we do not re-train the models on binary maps but we use the pre-trained models from continuous maps for this evaluation. As with the previous chapter, we present the results from the denoising U-Net first before comparing all the models.

4.2.1 Denoising U-Net

- **Comparing models at the map level**

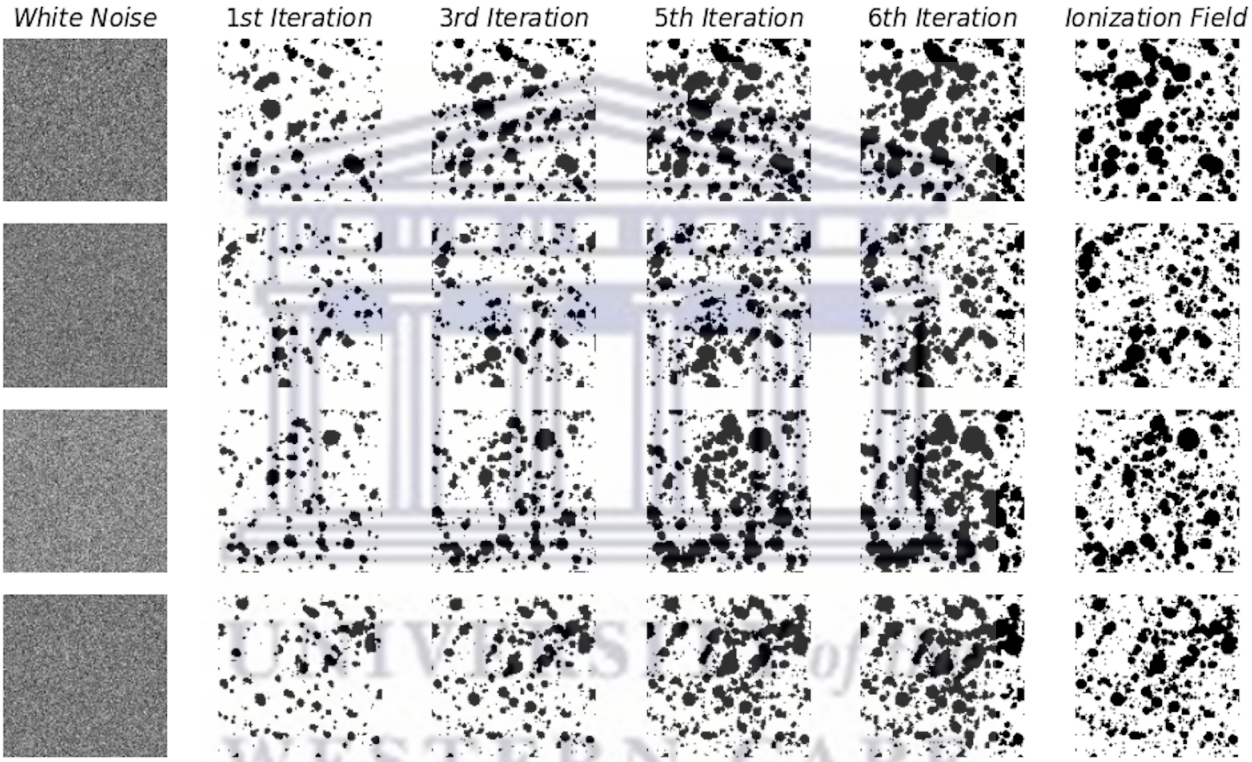

FIGURE 4.9: The visualization comparison of the ionization maps as a number of iterations when the maps are binary.

The testing protocol for the binary maps is similar to that of the continuous maps. It takes about five iterations for the generated ionization maps to match the true ionization maps, by visual inspection. On the 6th iteration, the ionization bubbles are bigger than those in the true maps. Figure 4.9 shows the comparison between ionization fields generated by the model and the simulation. White noise, 1st iteration, 3rd iteration, 5th iteration, 6th iteration, and the true ionization maps are represented by the 1st, 2nd, 3rd, 4th, 5th, and 6th column, respectively. Like the case in the continuous maps, the ionization bubbles

grow over iterations. By the fifth iteration, the predictions visually match the original maps.

- **Comparing models on the level of Power Spectrum**

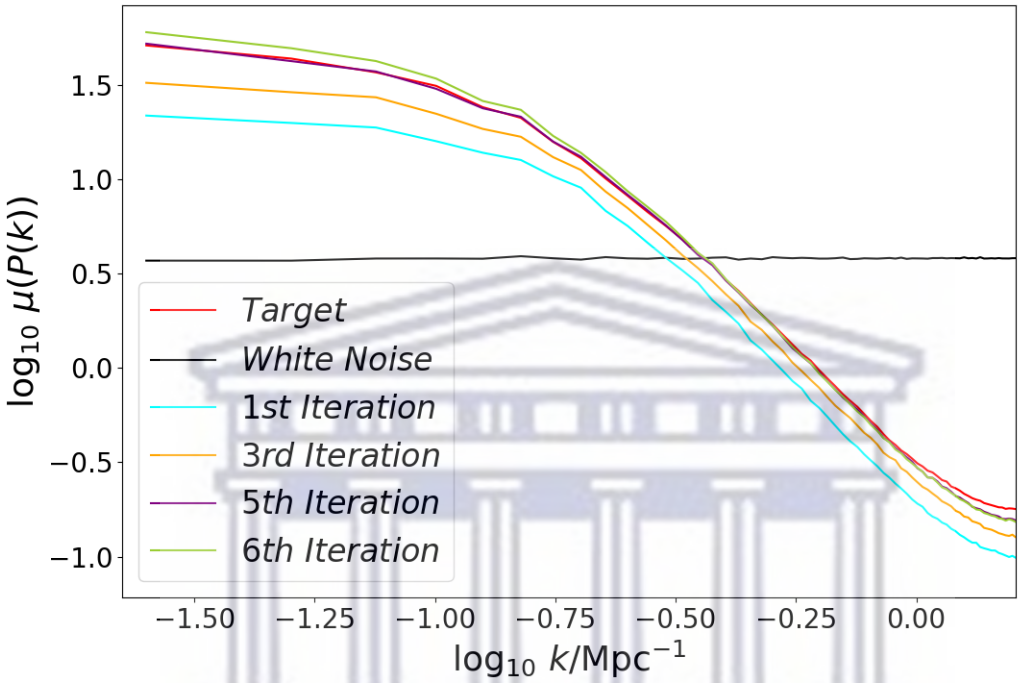

FIGURE 4.10: Original PS compared to the reconstructed PS as a function of iterations, where at the 5th iteration a perfect match to the target is observed.

The power spectra are shown by Figure 4.10. This figure shows the PS as a function of iterations for the denoising U-Net model. The power spectra for the target, white noise, the first iteration, 3rd iteration, 5th iteration, and 6th iteration are shown in red, black, cyan, orange, purple, and, lime, respectively. The figure shows that the PS approaches the target PS as the number of iterations increases. On the 5th iteration, the denoising U-Net perfectly matches the target spectrum, and the 6th iteration surpasses the target. Testing with binary maps provides a better fit to the PS specifically on small scales. The better agreement on the small-scale power as compared to testing with continuous maps is due to the fact that those intermediate features ($0.2 \leq x_{HII} \leq 0.8$) are now removed which mainly affect the small-scale bubble distributions by reducing their abundance. The large-scale power is still perfectly recovered since the morphology of large-scale bubbles does not change in either binary or continuous maps. However, most of upcoming 21cm

surveys will be mainly sensitive to detect those large-scale bubbles, and hence recovering the small-scale bubble distribution performed here is for completeness to minimize loss of information.

4.2.2 Performance on all models

- **Comparing models at the map level**

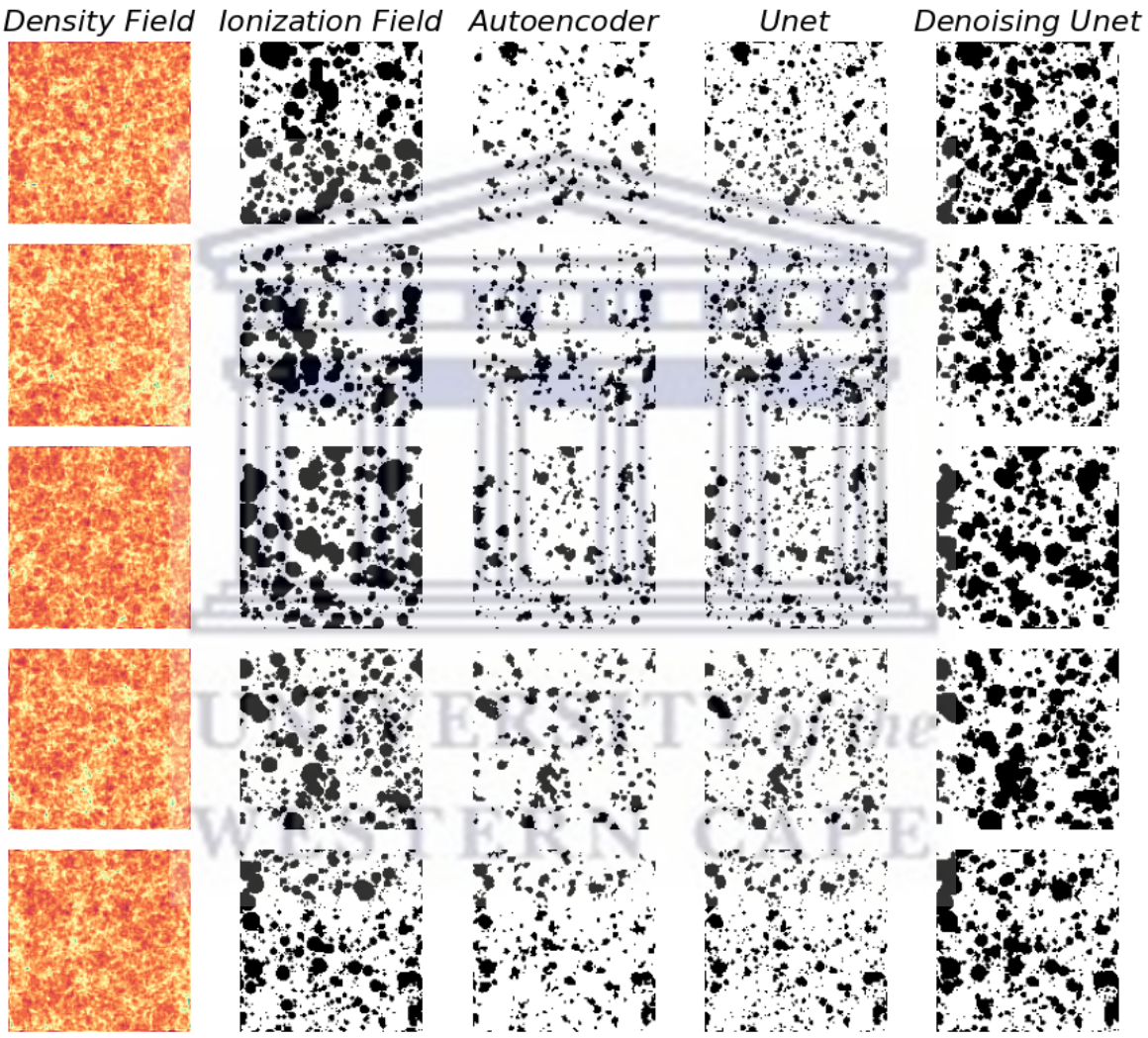

FIGURE 4.11: Comparison among the ionization fields generated from density fields using autoencoder (3rd column), basic U-Net (4th column), and the Denoising U-Net (5th column). The denoising U-Net is able to generate ionization maps that are almost identical to the target.

Figure 4.11 shows that the same trend that was observed with continuous maps still holds. The autoencoder and the basic U-Net models construct similar ionization bubbles, while with the denoising U-Net model, the ionization bubbles are much closer to the target.

The density field, ionization field (target), autoencoder output, basic U-Net output, and denoising U-Net output are shown in the 1st, 2nd, 3rd, 4th, and 5th columns, respectively. By visual inspection, we can see that the morphology of the ionized bubble indeed remains the same after the maps are converted to binary.

- **Comparing models on the level of Power Spectrum**

  The PS comparison between our different models for binary maps is shown in figure 4.12 below. The original PS (target) is shown in red, the autoencoder in magenta, the basic U-Net in blue, and the denoising U-Net in green. As we have already established, the denoising U-Net is the superior model, converting the maps from continuous to binary maps improves the recovery of the PS on both small and large scales.

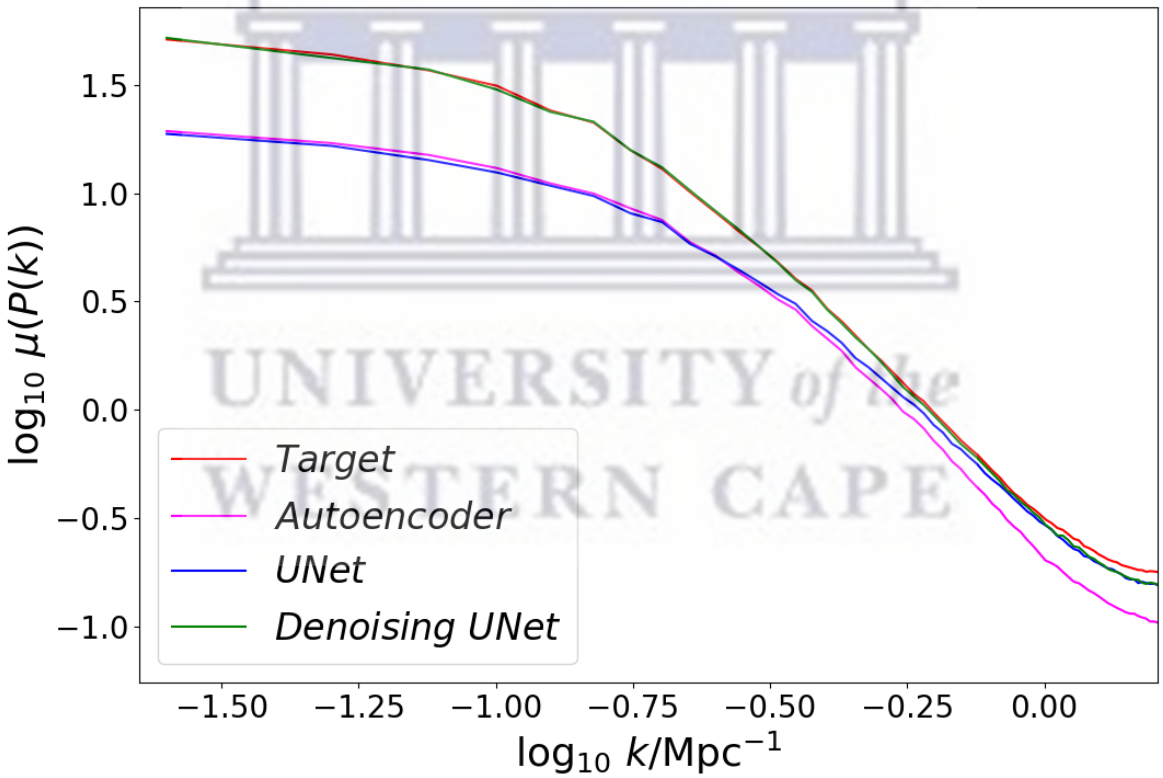

FIGURE 4.12: The PS comparison for the different models for binary maps. The denoising U-Net produces ionization maps with a power spectrum that is similar to the target.

- **Comparing models in terms of the Bubble Size Distribution**

  The BSD from binary maps is calculated following the same mfp method as with continuous maps. Figure 4.13 shows the number of ionized bubbles as a function of scales

in each model. The bubble sizes in both the autoencoder and the basic U-Net are much smaller than the predictions from the denoising U-Net and the simulation (target). The denoising U-Net model produces bubbles that are similar in size to the target over all scales. This finding corroborates the results that are shown by the visualization of the random ionization fields and the PS that the denoising U-Net is the best-performing model. Please note that the BSD from binary maps and the continuous maps are similar mainly due to the fact that computing the BSD requires converting maps to binary.

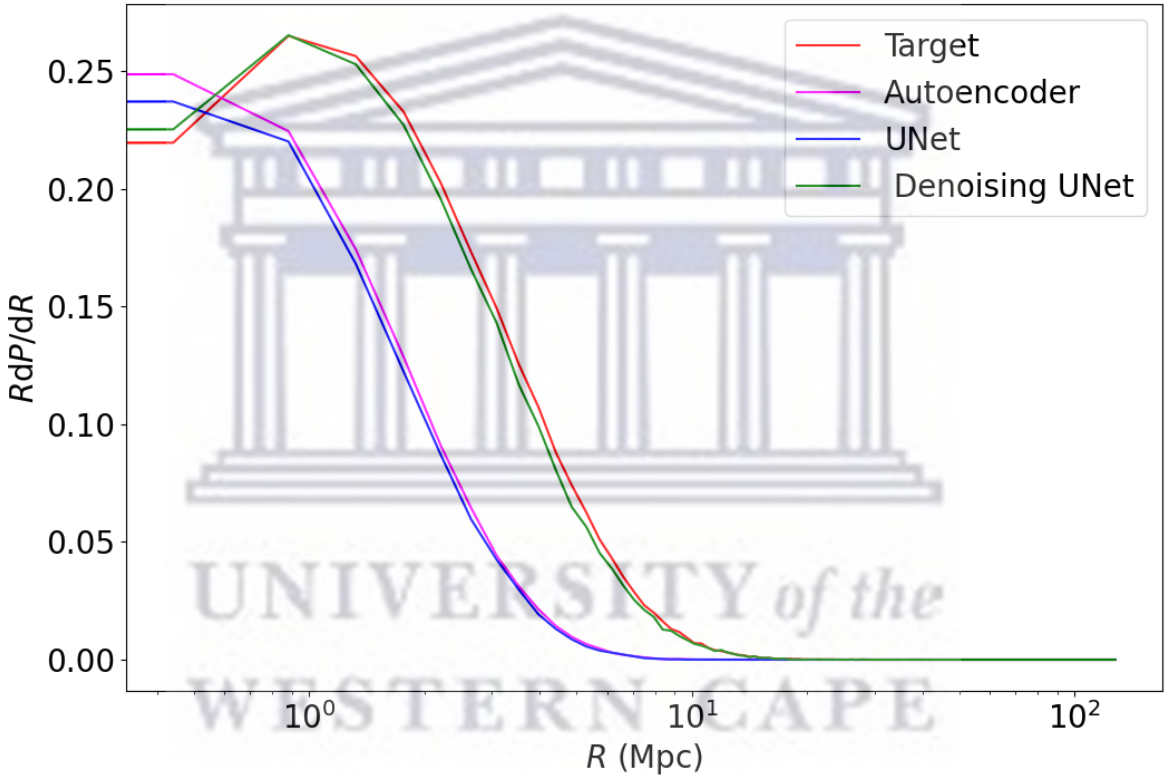

FIGURE 4.13: The BSD comparison between our different models for binary maps. The denoising U-Net is able to recover the target bubble distribution over all scales.

The aim of this thesis is to accelerate SimFast21 using machine learning. We have managed to build different generative models to accelerate SimFast21 and we found that the denoising U-Net model is the best-performing model to generate ionization fields directly from the density fields with PS and BSD in good agreement with that of the target.

5 Summary and Conclusions

SKA is one of the most anticipated future large-scale surveys of reionization that will provide huge amounts of high-dimensional datasets. These vast amounts of datasets will require efficient generation of new codes and simulations to analyze. Current reionization simulations are too computationally expensive to efficiently generate huge amounts of data to train neural networks and extract information. The most expensive part of all reionization simulations is the radiative transfer process, which we have attempted to accelerate using state-of-the-art techniques in generative models. We here provide a summary of the work presented in this thesis and plans for future work.

In Chapter 1 of this thesis we have reviewed several physical processes during EoR, key fundamental questions, and discussed how the IGM was reionized. We have reviewed the different indirect observational constraints on reionization such as GP optical depth observed in the Lyα absorption spectra of quasars, the HI evolution using high redshift quasars, polarization, and temperature anisotropies from CMB surveys, and highlighted the importance of 21cm as a direct probe to EoR. Given the importance of 21cm in this thesis, this chapter also discussed the different ongoing and planned 21cm experiments and some exploration of 21cm cosmology with machine learning techniques.

In Chapter 2, we have reviewed the basics of Machine Learning, and the differences between machine learning algorithms, and applications. We have compared supervised, unsupervised, and semi-supervised ML methods and highlighted the applications of the different methods. We have highlighted the difference between shallow and deep neural networks and why the latter was more relevant to our work. We have introduced all the ingredients required to build a successful DL model, such as loss functions, activation functions, optimizers, and regularisation techniques. We have discussed the two main generative models that we plan to use to accelerate SimFast21.

“Accelerating reionization simulations with machine learning", is the third chapter of this thesis. Here, we have discussed the semi-numerical simulation SimFast21 and provided a brief discussion on how this simulation generates the 21cm ionization map from density fields. We have shown and explained how we plan to accelerate this simulation using ML. In Chapter 3, we have also detailed the data generation process using SimFast21 and how the data was pre-processed for testing and training. Figure 3.3 showed random examples of density, halo, and ionization fields generated. The generative models used to generate ionization fields from density fields have been explained in detail with figures and tables. Finally, we have explained our testing protocols to evaluate the performance of the denoising U-Net model.

Results and discussion formed Chapter 4 of this thesis. The results are presented in two sections, where the first section presents the performance of the generative models for continuous ionization maps and the second section showed the performance on binary maps. The results are shown in terms of main four metrics; the loss evolution, visualization of the maps, PS, and BSD. Here, we have shown that the addition of noise in the training process of the denoising U-Net and the implementation of the iterative testing allowed for a more refined method to detect ionization bubbles. This was the best-performing model.

In this thesis, we have presented different machine learning models to accelerate SimFast21, a semi-numerical model that simulates the 21cm cosmological signal. We have compared three different machine learning models to see how well they generate the ionization fields from density fields. The autoencoder and the basic U-Net produce maps that are similar by visual inspection. This is shown in Figure 4.5. Both these models are able to recover the small bubbles but miss the edges of the bigger bubbles. By comparing the power spectra of these two with the original spectrum we were able to deduce that the U-Net makes an improvement on the small-scale power. Motivated by this, we have introduced a third model, which was the adaptation of the basic U-Net by adding noise in the second channel of the input images during training and implementing iterative testing. The denoising U-Net model makes a major improvement on the maps and the power spectrum, this is our best-performing model.

The Autoencoder, basic U-Net, and the denoising U-Net required 20 minutes, 40 minutes, and 2 hours respectively to train. All the models were trained for 200 epochs to observe the change in the loss evolution and to see if they converge. The trends in the loss evolution show that

all the models are not overfitting or underfitting the training data. Testing these models took a couple of a second to generate about 1000 ionization maps. This is a great improvement on the semi-numerical simulation comparing 5 minutes to a fraction of a second. To evaluate the performance of our three models on binary maps we have used the pre-trained models on continuous maps, and found that the small-scale power is perfectly recovered in this case.

Future works will include training the models on binary maps to better evaluate their performance against continuous maps. Training a deeper autoencoder architecture using the density field plus noise will also be explore since the performance of the autoencoder is comparable to the basic U-Net. We will also condition these machine learning models on other source model parameters such as the photon escape fraction (f_{esc}) and other cosmological parameters such as the matter density parameter (Ω_m). Another probable future work will involve training the same models on different source populations such as AGNs, which were explored in Hassan et al. (2018). These models can also be re-trained after adding various instrumental effects such as thermal noise, angular resolution, and foreground cleaning for different 21cm surveys. These models can be used to generate intensity maps for Carbon Monoxide (CO) and the fine-structure line of the ionized carbon atom (CII), which will enable joint analysis studies such as cross-correlation of other intensity mapping surveys with 21cm.

Bibliography

Abadi, Martín et al. (May 2015). "TensorFlow: A system for large-scale machine learning". In.

Abdurashidova, Zara et al. (Jan. 2022). "HERA Phase I Limits on the Cosmic 21 cm Signal: Constraints on Astrophysics and Cosmology during the Epoch of Reionization". In: 924.2, 51, p. 51. DOI: 10.3847/1538-4357/ac2ffc. arXiv: 2108.07282 [astro-ph.CO].

Ade, Peter et al. (Feb. 2015). "Planck 2015 results. XIV. Dark energy and modified gravity". In.

Allamy, Haider (Dec. 2014). "METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)". In.

Almeida, L.B. (1997). *Multilayer perceptrons*. IOP Publishing Ltd and Oxford University Press.

Alvarez, Marcelo et al. (June 2020). "Mitigating the optical depth degeneracy using the kinematic Sunyaev-Zel'dovich effect with CMB-S4". In.

Alzubaidi, L., J. Zhang, and A.J. Humaidi (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *J Big Data* 53 (8). DOI: https://doi.org/10.1186/s40537-021-00444-8.

Arthur, David and Sergei Vassilvitskii (Jan. 2007). "K-Means++: The Advantages of Careful Seeding". In: vol. 8, pp. 1027–1035. DOI: 10.1145/1283383.1283494.

Atul (2022). "AI vs Machine Learning vs Deep Learning". In: *Edureka*.

Baldi, Pierre (2012). "Autoencoders, Unsupervised Learning, and Deep Architectures". In: *JMLR: Workshop and Conference Proceedings* 27, pp. 37–50.

Ball, Nicholas M. et al. (Oct. 2006). "Robust Machine Learning Applied to Astronomical Data Sets. I. Star-Galaxy Classification of the Sloan Digital Sky Survey DR3 Using Decision Trees". In: 650.1, pp. 497–509. DOI: 10.1086/507440. arXiv: astro-ph/0606541 [astro-ph].

Bank, D. et al. (2020). "Autoencoders". In: *ArXiv*. DOI: https://doi.org/10.48550/arXiv.2003.05991.

Barkana, Rennan and Abraham Loeb (2001). "In the Beginning: The First Sources of Light and the Reionization of the Universe". In: *arXiV Astro-PH*.

Barry, N. et al. (Oct. 2019). "Improving the Epoch of Reionization Power Spectrum Results from Murchison Widefield Array Season 1 Observations". In: 884.1, 1, p. 1. DOI: 10.3847/1538-4357/ab40a8. arXiv: 1909.00561 [astro-ph.IM].

Battaglia, N. et al. (Oct. 2013). "Reionization on Large Scales. III. Predictions for Low-l Cosmic Microwave Background Polarization and High-l Kinetic Sunyaev-Zel'dovich Observables". In: 776.2, 83, p. 83. DOI: 10.1088/0004-637X/776/2/83. arXiv: 1211.2832 [astro-ph.CO].

Beardsley, A. P. et al. (Dec. 2016). "First Season MWA EoR Power spectrum Results at Redshift 7". In: 833.1, 102, p. 102. DOI: 10.3847/1538-4357/833/1/102. arXiv: 1608.06281 [astro-ph.IM].

Becker, Robert et al. (Aug. 2001). "Evidence for Reionization at z 6: Detection of a Gunn-Peterson Trough in a z=6.28 Quasar". In: *The Astronomical Journal* 122. DOI: 10.1086/324231.

Becker, S. and M. Plumbley (1996). "Unsupervised neural network learning procedures for feature extraction and classification". In: *Applied Intelligence* 6, 185–203. DOI: https://doi.org/10.1007/BF00126625.

Bennett, C. L. et al. (2003). "First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Preliminary Maps and Basic Results". In: *Arxiv*. DOI: 10.1086/377253.

Bennett, C. L. et al. (2012). "Nine-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Final Maps and Results". In: *ArXiv*. DOI: https://doi.org/10.1088/0067-0049/208/2/20.

Bennett, C.L. et al. (June 1996). "Four-Year COBE DMR Cosmic Microwave Background Observations: Maps and Basic Results". In: *The Astrophysical Journal* 464, p. L1. DOI: 10.1086/310075. arXiv: astro-ph/9601067 [astro-ph].

Bernardis, P. de et al. (2000). "A flat Universe from high-resolution maps of the cosmic microwave background radiation ". In: *Nature* 404, pp. 955–959.

Bewtra, A (2022). "The Ultimate Guide to Semi-supervised learning". In: *v7*.

Bird, Simeon et al. (Mar. 2022). "The ASTRID simulation: galaxy formation and reionization". In: *Monthly Notices of the Royal Astronomical Society* 512. DOI: 10.1093/mnras/stac648.

Blumenthal, G. R. et al. (Oct. 1984). "Formation of galaxies and large-scale structure with cold dark matter." In: 311, pp. 517–525. DOI: 10.1038/311517a0.

Bond, JR et al. (1991). "Excursion set mass functions for hierarchical Gaussian fluctuations". In: *The Astrophysical Journal* 379, pp. 440–460.

Bowman, Judd D et al. (2013). "Science with the murchison widefield array". In: *Publications of the Astronomical Society of Australia* 30.

Breiman, L. (2001). "Random Forests". In: *Machine Learning* (45), 5–32. DOI: https://doi.org/10.1023/A:1010933404324.

Busuttil, Steven, John Abela, and Gordon J. Pace (2004). "Support vector machines with profile-based kernels for remote protein homology detection". In: *Genome Inform* 15.2, pp. 191–200.

Bégin, Joëlle-Marie, Adrian Liu, and Adélie Gorce (Dec. 2021). "Joint constraints on reionization: a framework for combining the global 21cm signal and the kinetic Sunyaev-Zel'dovich effect". In.

Chan, Stanly (2020). *Machine Learning, Lecture 29: Bias and Variance*.

Chapelle, O., M. Chi, and A. Zien (2006). "A continuation method for semi-supervised SVMs". In: *In Proceedings of the 23rd international conference on machine learning*, 185–192.

Chardin, Jonathan et al. (2019). "A deep learning model to emulate simulations of cosmic reionization". In: *Monthly Notices of the Royal Astronomical Society* 490 (1), 1055–1065.

Choudhury, Tirthankar, Suvodip Mukherjee, and Sourabh Paul (Nov. 2020). "CMB constraints on a physical model of reionization". In: *Monthly Notices of the Royal Astronomical Society Letters*. DOI: 10.1093/mnrasl/slaa185.

Ciampiconi, Lorenzo et al. (2023). "A survey and taxonomy of loss functions in machine learning". In: *arxiv*.

Cios, K.J. et al. (2007). *Data Mining*. Springer, Boston, MA. Chap. Unsupervised Learning: Association Rules, pp. 289–306. DOI: https://doi.org/10.1007/978-0-387-36795-8_10.

Coc, Alain, Jean-Philippe Uzan, and Elisabeth Vangioni (Mar. 2014). "Standard big bang nucleosynthesis and primordial CNO Abundances after Planck". In: *Journal of Cosmology and Astroparticle Physics* 2014. DOI: 10.1088/1475-7516/2014/10/050.

Coil, Alison L. (2013). *The Large-Scale Structure of the Universe*. Dordrecht: Springer, 387–421.

Cortes, C. and V. Vapnik (1995). "Support-vector networks". In: *Machine Learning* (20), 273–297. DOI: https://doi.org/10.1007/BF00994018.

Cunningham, P., M. Cord, and S.J. Delany (2008). *Machine Learning Techniques for Multimedia*. Springer, Berlin, Heidelberg. Chap. Chapter 2: Supervised Learning, pp. 21–49.

Dauphim, YN et al. (2017). "Language Modelling with Gated Convolutional Networks". In: *arXiv*.

Dauphin, Yann et al. (Feb. 2015). "RMSProp and equilibrated adaptive learning rates for non-convex optimization". In: *arXiv* 35.

Davé, Romeel et al. (2013). "The neutral hydrogen content of galaxies in cosmological hydrodynamic simulations". In: *Monthly Notices of the Royal Astronomical Society* 434.3, pp. 2645–2663.

De Zotti, G et al. (2019). "Early evolution of galaxies and of large-scale structure from CMB experiments". In: *Arxiv*. DOI: https://doi.org/10.48550/arXiv.1904.04531.

DeBoer, David R et al. (2017). "Hydrogen epoch of reionization array (HERA)". In: *Publications of the Astronomical Society of the Pacific* 129.974, p. 045001.

DeLua, J (2021). "Supervised vs unsupervised Learning: What's the difference". In: *IBM Analytics*.

Deng, Li (2014). "A tutorial survey of architectures, algorithms, and applications for deep learning". In: *APSIPA Transactions on Signal and Information Processing* 3.1.

Dubey, S. R., S. K. Singh, and B. B. Chaudhuri (2021). "Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark". In: *ArXiv*. DOI: https://doi.org/10.48550/arXiv.2109.14545.

European Southern Observatory (2016). *Schematic diagram of the history of the Universe*. https://cdn.eso.org/images/publicationjpg/eso1620a.jpg. online accessed : 24 March 2023.

Fan, Xiaohui et al. (July 2006). "Constraining the Evolution of the Ionizing Background and the Epoch of Reionization with z∼6 Quasars. II. A Sample of 19 Quasars". In: 132.1, pp. 117–136. DOI: 10.1086/504836. arXiv: astro-ph/0512082 [astro-ph].

Ferrara, Andrea and Stefania Pandolfi (2014). ""Reionization of the Intergalactic Medium."" In: *ArXiv*. Accessed March 15, 2023. DOI: https://doi.org/10.48550/arXiv.1409.4946..

Fields, Brian, P. Molaro, and Subir Sarkar (Dec. 2014). "Big-Bang Nucleosynthesis". In.

Finlator, Kristian et al. (2015). "The reionization of carbon". In: *Monthly Notices of the Royal Astronomical Society* 447.3, pp. 2526–2539.

Franco-Árcega, Anilú, L.G. Flores-Flores, and Ruslan Gabbasov (Nov. 2013). "Application of Decision Trees for Classifying Astronomical Objects". In: pp. 181–186. ISBN: 978-1-4799-2604-6. DOI: 10.1109/MICAI.2013.29.

Furlanetto, S., S. P Oh, and F. Briggs (2006). "Cosmology at Low Frequencies: The 21 cm Transition and the High-Redshift Universe". In: *ArXiv*. DOI: https://doi.org/10.1016/j.physrep.2006.08.002.

Furlanetto, S., M. Zaldarriaga, and L. Hernquist (2004). "The Growth of HII Regions During Reionization". In: *ArXiv*. DOI: https://doi.org/10.1086/423025.

Furlanetto, S.R (2006). "The global 21-centimeter background from high redshifts". In: *Monthly Notices of the Royal Astronomical Society* 371 (2), pp. 867–878. DOI: https://doi.org/10.1111/j.1365-2966.2006.10725.x.

Gao, Dan, Yan-Xia Zhang, and Yong-Heng Zhao (Feb. 2009). "Random forest algorithm for classification of multiwavelength data". In: *Research in Astronomy and Astrophysics* 9.2, pp. 220–226. DOI: 10.1088/1674-4527/9/2/011.

Gaurang, Panchal et al. (Apr. 2011). "Determination of Over-Learning and Over-Fitting Problem in Back Propagation Neurl Network". In: *International Journal on Soft Computing (IJSC)* 2. DOI: 10.5121/ijsc.2011.2204.

Geurts, Pierre (2002). *Bias and variance in Machine Learning*.

Giallongo, E. et al. (1994). "The Gunn-Peterson Effect in the Spectrum of the Z=4.7 QSO 1202-0725: The Intergalactic Medium at Very High Redshift". In: *ArXiv*. DOI: https://doi.org/10.1086/187296.

Gillet, Nicolas et al. (Jan. 2019). "Deep learning from 21-cm tomography of the Cosmic Dawn and Reionization". In: *Monthly Notices of the Royal Astronomical Society* 484. DOI: 10.1093/mnras/stz010.

Giri, Sambit, Garrelt Mellema, and Hannes Jensen (Aug. 2020). "Tools21cm: A python package to analyse the large-scale 21-cm signal from the Epoch of Reionization and Cosmic Dawn". In: *The Journal of Open Source Software* 5.52, 2363, p. 2363. DOI: 10.21105/joss.02363.

Gnedin, Nickolay Y. (June 2000). "Cosmological Reionization by Stellar Sources". In: 535.2, pp. 530–554. DOI: 10.1086/308876. arXiv: astro-ph/9909383 [astro-ph].

Goodfellow, I. J. et al. (2014). "Generative Adversarial Networks". In: *ArXiv*. DOI: https://doi.org/10.48550/arXiv.1406.2661.

Greig, Bradley et al. (Jan. 2021). "Exploring reionization and high-z galaxy observables with recent multiredshift MWA upper limits on the 21-cm signal". In: 500.4, pp. 5322–5335. DOI: 10.1093/mnras/staa3494. arXiv: 2008.02639 [astro-ph.CO].

Gron, Oyvind (Dec. 2018). "The Discovery of the Expansion of the Universe". In: *Galaxies* 6, p. 132. DOI: 10.3390/galaxies6040132.

Gunn, James E and Bruce A Peterson (1965). "On the Density of Neutral Hydrogen in Intergalactic Space." In: *The Astrophysical Journal* 142, pp. 1633–1641.

Haarlem, M áP van et al. (2013). "LOFAR: The low-frequency array". In: *Astronomy & astrophysics* 556, A2.

Haider, Allamy (Dec. 2014). "METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)". In.

Handel, Ramon van (2008). "The stability of conditional Markov processes and Markov chains in random environments". In: *arXiV*.

Hassan, S. et al. (2015). "Simulating the 21-cm signal from reionisation including non-linear ionisations and inhomogeneous recombinations". In: *ArXiv*. DOI: https://doi.org/10.1093/mnras/stv3001.

Hassan, S. et al. (2016). "Epoch of Reionisation 21cm Forecasting From MCMC-Constrained Semi-Numerical Models". In: *ArXiv*. DOI: https://doi.org/10.1093/mnras/stx420.

Hassan, S. et al. (2018). "Constraining the contribution of active galactic nuclei to reionization". In: *Monthly Notices of the Royal Astronomical Society*.

Hassan, Sultan, Sambatra Andrianomena, and Caitlin Doughty (June 2020). "Constraining the astrophysics and cosmology from 21 cm tomography using deep learning with the SKA". In: *Monthly Notices of the Royal Astronomical Society* 494, pp. 5761–5774. DOI: 10.1093/mnras/staa1151.

Hassan, Sultan et al. (Dec. 2018). "Identifying Reionization Sources from 21cm Maps using Convolutional Neural Networks". In: *Monthly Notices of the Royal Astronomical Society* 483. DOI: 10.1093/mnras/sty3282.

He, Wei and Gang Zhao (2019). "A PCA approach to stellar abundances I. testing of the method validity". In: *Research in Astronomy and Astrophysics* 19.10. DOI: 10.1088/1674-4527/19/10/140.

Hinshaw, G. et al. (2006). "Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Temperature Analysis". In: DOI: https://doi.org/10.1086/513698.

Hinton, Geoffrey et al. (July 2012). "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint* arXiv.

Hubble, Edwin (Mar. 1929). "A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae". In: *Proceedings of the National Academy of Science* 15.3, pp. 168–173. DOI: 10.1073/pnas.15.3.168.

Iliev, I. T. et al. (July 2006). "Simulating cosmic reionization at large scales - I. The geometry of reionization". In: 369.4, pp. 1625–1638. DOI: 10.1111/j.1365-2966.2006.10502.x. arXiv: astro-ph/0512187 [astro-ph].

Iliev, Ilian T. et al. (Dec. 2009). "Cosmological radiative transfer comparison project - II. The radiation-hydrodynamic tests". In: 400.3, pp. 1283–1316. DOI: 10.1111/j.1365-2966.2009.15558.x. arXiv: 0905.2920 [astro-ph.CO].

Ioffe, S and C Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covarience Shift". In: *Google Inc*.

Jain, A.K., M. N. Murty, and P. Flynn (1999). "Data clustering: A review". In: *ACM Computing Surveys* 31.3, 264–323.

Jain, V (2019). "Everything you need to know about Activation Functions in deep learning models". In: *Towards Data Science*.

Jin, X. and J. Han (2011). *In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning*. Springer, Boston, MA. Chap. K-Means Clustering. DOI: https://doi.org/10.1007/978-0-387-30164-8_425.

Jones, E and J Singal (2017). "Analysis of a custom support vector machine for photometric redshift estimation and the inclusion of galaxy shape information". In: *Astronomy and Astrophysics Journal*.

Kannan, R et al. (Dec. 2021). "Introducing the thesan project: radiation-magneto-hydrodynamic simulations of the epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 511. DOI: 10.1093/mnras/stab3710.

Karlik, B and A Vehbi (2011). "Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks". In: *International Journal of Artificial Intelligence and Expert Systems (IJAE)* 1.4, 111–122. URL: http://www.cscjournals.org/library/manuscriptinfo.php.

Ketkar, Nikhil (Oct. 2017). "Introduction to Keras". In: pp. 95–109. ISBN: 978-1-4842-2765-7. DOI: 10.1007/978-1-4842-2766-4_7.

Kingma, Diederik and Jimmy Ba (Dec. 2014). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*.

Komatsu, E. et al. (2008). "Five-Year Wilkinson Microwave Anisotropy Probe Observations: Cosmological Interpretation". In: *arxiv*. DOI: 10.1088/0067-0049/180/2/330. arXiv: 0803.0547 [astro-ph].

Komatsu, E. et al. (2010). "Seven-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation". In: DOI: https://doi.org/10.1088/0067-0049/192/2/18.

Komatsu, E. et al. (Feb. 2011a). "Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation". In: 192.2, 18, p. 18. DOI: 10.1088/0067-0049/192/2/18. arXiv: 1001.4538 [astro-ph.CO].

Komatsu, E. et al. (2011b). "Seven-year Wilkinson Microwave Anisotropy Probe(WMAP) Observations: Cosmological Interpretation". In: *The Astrophysical Journal* 192, p. 18.

Koopmans, LVE et al. (2015). "The cosmic dawn and epoch of reionization with the square kilometre array". In: *arXiv preprint arXiv:1505.07568*.

Kotsiantis, S. B. (2007). "Supervised Machine Learning: A Review of Classification Techniques". In: *Informatica* 31.3, pp. 249–268.

LeCun, Yann et al. (1989). "Handwritten Digit Recognition with a Back-Propagation Network". In.

Li, W. et al. (Dec. 2019). "First Season MWA Phase II Epoch of Reionization Power Spectrum Results at Redshift 7". In: 887.2, 141, p. 141. DOI: 10.3847/1538-4357/ab55e4. arXiv: 1911.10216 [astro-ph.CO].

Maas, A, A Hannun, and A Ng (2013). "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *International Conference on Machine Learning(ICML)*.

Majumdar, Suman et al. (Mar. 2014). "On the use of semi-numerical simulations in predicting the 21-cm signal from the epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 443. DOI: 10.1093/mnras/stu1342.

Malik, F (2019). "Neural Networks Bias and Weights- Understanding The Two Most Important Components". In: *FinTechExplained*.

Mangena, Tumelo, Sultan Hassan, and Mario Santos (May 2020). "Constraining the reionization history using deep learning from 21-cm tomography with the Square Kilometre Array". In: *Monthly Notices of the Royal Astronomical Society* 494, pp. 600–606. DOI: 10.1093/mnras/staa750.

Mason, C. A. et al. (2019). "Model-independent constraints on the hydrogen-ionizing emissivity at z > 6". In: *Monthly Notices of the Royal Astronomical Society* 489 (2), pp. 2669–2676. DOI: https://doi.org/10.1093/mnras/stz2291.

McGreer, Ian D., Andrei Mesinger, and Valentina D'Odorico (Feb. 2015). "Model-independent evidence in favour of an end to reionization by z ≈ 6". In: 447.1, pp. 499–505. DOI: 10.1093/mnras/stu2449. arXiv: 1411.5375 [astro-ph.CO].

McGreer, Ian D., Andrei Mesinger, and Xiaohui Fan (Aug. 2011). "The first (nearly) model-independent constraint on the neutral hydrogen fraction at z ~ 6". In: 415.4, pp. 3237–3246. DOI: 10.1111/j.1365-2966.2011.18935.x. arXiv: 1101.3314 [astro-ph.CO].

Mellema, Garrelt et al. (Aug. 2013). "Reionization and the Cosmic Dawn with the Square Kilometre Array". In: *Experimental Astronomy* 36.1-2, pp. 235–318. DOI: 10.1007/s10686-013-9334-5. arXiv: 1210.0197 [astro-ph.CO].

Mertens F G andMevius, M et al. (2020). "Improved upper limits on the 21cm signal power spectrum of neutral hydrogen at z 9.1 from LOFAR". In: *Monthly Notices of the Royal Astronomical Society* 493 (2), 1662–1685. DOI: https://doi.org/10.1093/mnras/staa327.

Mesinger, A., S. Furlanetto, and R. Cen (2010). "21cmFAST: A Fast, Semi-Numerical Simulation of the High-Redshift 21-cm Signal". In: *arXiv*. DOI: https://doi.org/10.1111/j.1365-2966.2010.17731.x.

Mesinger, Andrei (2010). "Was reionization complete by z 5–6?" In: *Monthly Notices of the Royal Astronomical Society* 407 (2), 1328–1337. DOI: https://doi.org/10.1111/j.1365-2966.2010.16995.x.

Mesinger, Andrei and Steven Furlanetto (2007). "Efficient Simulations of Early Structure Formation and Reionization". In: *The Astrophysics Journal* 669.2, p. 663. DOI: 10.1086/521806.

Mesinger, Andrei, Matthew McQuinn, and David Spergel (Dec. 2011). "The kinetic Sunyaev-Zel'dovich signal from inhomogeneous reionization: A parameter space study". In: *Monthly Notices of the Royal Astronomical Society* 422. DOI: 10.1111/j.1365-2966.2012.20713.x.

Molaro, Margherita et al. (Jan. 2019). "ARTIST: Fast radiative transfer for large-scale simulations of the epoch of reionisation". In.

Murchison Widefield Array (MWA) (2009). *The first stage of the MWA was a 32-tile prototype*. https://www.mwatelescope.org/about/. Online accessed: 24 March 2023.

Na, Shi, Liu Xumin, and Guan Yong (2010). "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm". In: *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pp. 63–67. DOI: 10.1109/IITSI.2010.74.

National Academies of Sciences, Engineering, and Medicine (2021). *Pathways to Discovery in Astronomy and Astrophysics for the 2020s*. The National Academies Press. DOI: https://doi.org/10.17226/26141.

National Center for Radio Astrophysics-Tata Institute of Fundamental Research (NCRA-TIFR) (1999). *Giant Metrewave Radio Telescope*. http://www.gmrt.ncra.tifr.res.in/gmrt/gmrt.html. Online accessed : 24 March 2023.

National Research Council (2010). *New Worlds, New Horizons in Astronomy and Astrophysics*. The National Academies Press. DOI: https://doi.org/10.17226/12951.

Neal, RM (1992). "Connectionist learning of belief networks". In: *Artificial Intelligence* 56.1, pp. 71–113. URL: https://doi.org/10.1016/0004-3702(92)90065-6.

Nemec, Lydia (2022). "Principal Component Analysis (PCA): A Physically Intuitive Mathematical Introduction". In: *Towards Data Science*.

Ng, Andrew and Michael Jordan (2001). "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in Neural Information Processing Systems*.

Nwankpa, Chigozie et al. (2018). "Activation Functions : Comparison of Trends in Practice and Research for Deep Learning". In: *arXiv*.

Năstase, Horaţiu (Apr. 2019). "Evidence for Dark Matter and the ΛCDM Model". In: pp. 41–51. DOI: 10.1007/978-3-030-15077-8_4.

Orr, Mark J L (1996). "Introduction to Radial Basis Function Networks". In.

Paciga, Gregory et al. (Jan. 2013). "A simulation calibrated limit on the HI power spectrum from the GMRT Epoch of Reionization experiment". In: *Monthly Notices of the Royal Astronomical Society* 433. DOI: 10.1093/mnras/stt753.

Pant, Ayush (2019). "Workflow of a Machine Learning Project". In: *Towards Data Science*.

Papachristoudis, Giorgos (2019). "The Bias-Variance Tradeoff". In: *Towards Data Science*.

Park, Hyunbae et al. (Jan. 2013). "The Kinetic Sunyaev-Zel'dovich effect as a probe of the physics of cosmic reionization: the effect of self-regulated reionization". In: *The Astrophysical Journal* 769. DOI: 10.1088/0004-637X/769/2/93.

Patil, AH et al. (Feb. 2017). "Upper Limits on the 21 cm Epoch of Reionization Power Spectrum from One Night with LOFAR". In: *Astrophysical Journal* 838. DOI: 10.3847/1538-4357/aa63e7.

Patra, Narendra et al. (Mar. 2019). "The expanded Giant Metrewave Radio Telescope". In: *Monthly Notices of the Royal Astronomical Society* 483, pp. 3007–3021. DOI: 10.1093/mnras/sty3266.

Penzias, A. A. and R. W. Wilson (July 1965). "A Measurement of Excess Antenna Temperature at 4080 Mc/s." In: 142, pp. 419–421. DOI: 10.1086/148307.

Pierre, Astier and Reynald Pain (2012). "Observational evidence of the accelerated expansion of the universe". In: *Comptes Rendus Physique* 13 (6-7), pp. 521–538. URL: https://doi.org/10.1016/j.crhy.2012.04.009..

Planck Collaboration et al. (2013). "Planck 2013 results. XVI. Cosmological parameters". In: *ArXiv*. DOI: 10.1051/0004-6361/201321591. arXiv: 1303.5076.

Planck Collaboration et al. (2015). "Planck 2015 results. XIII. Cosmological parameters". In: *ArXiv*. DOI: 10.1051/0004-6361/201525830. arXiv: 1502.01589 [astro-ph.CO].

Planck Collaboration et al. (2018). "Planck 2018 results. VI. Cosmological parameters". In: *ArXiv*. DOI: 10.1051/0004-6361/201833910. arXiv: 1807.06209 [astro-ph.CO].

Popescu, Marius-Constantin et al. (July 2009). "Multilayer perceptron and neural networks". In: *WSEAS Transactions on Circuits and Systems* 8.

Popolo, A. and Morgan Le Delliou (June 2016). "Small scale problems of the ΛCDM model: a short review". In: *Galaxies* 5. DOI: 10.3390/galaxies5010017.

Pothuganti, Swathi (Sept. 2018). "Review on over-fitting and under-fitting problems in Machine Learning and solutions". In: *International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering* 7, pp. 3692–3695. DOI: 10.15662/IJAREEIE.2018.0709015.

Quinlan, J.R. (1986). "Induction of decision trees". In: *Machine Learning* (1), 81–106. DOI: https://doi.org/10.1007/BF00116251.

Robbins, H. and S. Monro (1951). "A stochastic approximation method". In: *The annals of mathematical statistics* 22 (3), pp. 400–407.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (Oct. 2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *LNCS* 9351, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

Rosdahl, Joakim et al. (Sept. 2018). "The SPHINX cosmological simulations of the first billion years: the impact of binary stars on reionization". In: 479.1, pp. 994–1016. DOI: 10.1093/mnras/sty1655. arXiv: 1801.07259 [astro-ph.GA].

Ruder, S. (2016). "An overview of gradient descent optimization algorithms". In: *ArXiv*. DOI: https://doi.org/10.48550/arXiv.1609.04747.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1985). "Learning internal representations by error propagation". In.

Rumelhart, D.E, G.E Hinton, and R.J Williams (1986). *Learning internal representations by error propagation*. Foundations. MIT Press, Cambridge, MA.

Sah, Shagan (July 2020). "Machine Learning: A Review of Learning Types". In: DOI: 10.20944/preprints202007.0230.v1.

Santos, Mario et al. (Oct. 2010). *SimFast21: Simulation of the Cosmological 21cm Signal*. Astrophysics Source Code Library, record ascl:1010.025. ascl: 1010.025.

Sarle, Warren (1995). "Stopped Training and Other Remedies for Overfitting". In.

Savin, Daniel et al. (Sept. 2019). "Astro2020: Decadal Survey on Astronomy and Astrophysics : State of the Profession Considerations for Laboratory Astrophysics". In.

Shimabukuro, Hayato and Benoit Semelin (May 2018). "Analysing 21cm signal with artificial neural network". In: *Peering towards Cosmic Dawn*. Ed. by Vibor Jelić and Thijs van der Hulst. Vol. 333, pp. 39–42. DOI: 10.1017/S174392131701081X.

Siegel, Ethan (2022). *Why 21 cm is the magic length for the Universe*. https://bigthink.com/starts-with-a-bang/21cm-magic-length/. Online Accessed : 07 April 2023.

Singh, P. Harinder, K. Ravi Gulati, and Ranjan Gupta (1998). "Stellar spectral classification using principal component analysis and artificial neural networks". In: *Monthly Notices of the Royal Astronomical Society* 295 (2), pp. 312–318. DOI: https://doi.org/10.1046/j.1365-8711.1998.01255.x.

SkAO (2022). https://www.skao.int/en/news/441/ska-observatory-celebrates-start-telescope-construction-australia-and-south-africa. Accessed March 24,2023.

South African Radio Astronomy observatory (SARAO) (2016). *Hydrogen Epoch of Reionisation Array radio telescope*. https://www.sarao.ac.za/science/hera/. Online accessed : 24 March 2023.

Spergel, D. N. et al. (2007). "Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology". In: *ArXiv*. DOI: 10.1086/513700. arXiv: astro-ph/0603449 [astro-ph].

Spergel, D.N. et al. (2003). "First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Determinationof Cosmological Parameters". In: *The Astrophysical Journal* 148, 175–194.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (1), pp. 1929–1958.

Sunyaev, R. A. and Ia. B. Zeldovich (Jan. 1980). "Microwave background radiation as a probe of the contemporary structure and history of the universe". In: 18, pp. 537–560. DOI: 10.1146/annurev.aa.18.090180.002541.

Sutton, Charles and Andrew McCallum (2011). "An Introduction to Conditional Random Fields". In: *Foundations and Trends in Machine Learning* 4 (4), 267–373.

Swarup, Govind et al. (Jan. 1996). "The Giant Metrewave Radio Telescope". In: *High-Sensitivity Radio Astronomy*, pp. 51–.

Szandała, T. (2020). "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks". In: *ArXiv*. DOI: https://doi.org/10.1007/978-981-15-5495-7.

Tai, Y. (2021). "A Survey Of Regression Algorithms And Connections With Deep Learning". In: *ArXiv*. DOI: https://doi.org/10.48550/arXiv.2104.12647.

The HERA Collaboration et al. (Oct. 2022). "Improved Constraints on the 21 cm EoR Power Spectrum and the X-Ray Heating of the IGM with HERA Phase I Observations". In: *arXiv e-prints*, arXiv:2210.04912, arXiv:2210.04912. DOI: 10.48550/arXiv.2210.04912. arXiv: 2210.04912 [astro-ph.CO].

Tieleman, T and G. Hinton (2012). "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". In: 4 (2), pp. 26–31.

Tingay, S. J. et al. (Jan. 2013). "The Murchison Widefield Array: The Square Kilometre Array Precursor at Low Radio Frequencies". In: 30, e007, e007. DOI: 10.1017/pasa.2012.007. arXiv: 1206.6945 [astro-ph.IM].

Tiwari, Himanshu et al. (Aug. 2021). "Improving constraints on the reionization parameters using 21-cm bispectrum". In.

Turner, Michael (Sept. 2021). "ΛCDM: Much more than we expected, but now less than what we want". In.

Villaescusa-Navarro, Francisco et al. (Oct. 2018). "Ingredients for 21 cm Intensity Mapping". In: 866.2, 135, p. 135. DOI: 10.3847/1538-4357/aadba0. arXiv: 1804.09180 [astro-ph.CO].

Wang, Qi et al. (Apr. 2022). "A Comprehensive Survey of Loss Functions in Machine Learning". In: *Annals of Data Science* 9. DOI: 10.1007/s40745-020-00253-5.

White, Martin (2014). "The Zeldovich approximation". In: *Monthly Notices of the Royal Astronomical Society*.

Zahn, Oliver et al. (Jan. 2007). "Simulations and Analytic Calculations of Bubble Growth during Hydrogen Reionization". In: 654.1, pp. 12–26. DOI: 10.1086/509597. arXiv: astro-ph/0604177 [astro-ph].

Zel'dovich, Ya. B. (Mar. 1970). "Gravitational instability: An approximate theory for large density perturbations." In: 5, pp. 84–89.

Zeldovich, Ya. B. and R. A. Sunyaev (July 1969). "The Interaction of Matter and Radiation in a Hot-Model Universe". In: 4.3, pp. 301–316. DOI: 10.1007/BF00661821.

Zhang, Y. and Y. Zhao (May 2014). "Applications of Support Vector Machines in Astronomy". In: *Astronomical Data Analysis Software and Systems XXIII*. Ed. by N. Manset and P. Forshay. Vol. 485. Astronomical Society of the Pacific Conference Series, p. 239.

Zhu, X. and J. Lafferty (2005). "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning". In: *In Proceedings of the 22nd international conference on machine learning*, 1052–1059.