Unlocking the Early Universe using HI 21cm Cosmology: Marginal Neural Ratio Estimation for 21-cm Power Spectra Analysis

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Bisweswar Sen

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2025

Supervisor: Dr.Abhirup Datta

Certificate

This is to certify that this dissertation entitled Unlocking the Early Universe using HI 21cm Cosmology: Marginal Neural Ratio Estimation for 21-cm Power Spectra Analysis towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Bisweswar Sen at Indian Institute of Technology,Indore under the supervision of Dr.Abhirup Datta Dean Department of Astronomy & Astrophysics , IIT Indore, during the academic year 2020-2025.

Dr.Abhirup Datta

Committee:

Dr.Abhirup Datta

Dr.Susmita Adhikari

This thesis is dedicated to my mom Dr.Usha Sen

Declaration

I hereby declare that the matter embodied in the report entitled Unlocking the Early Universe using HI 21cm Cosmology: Marginal Neural Ratio Estimation for 21-cm Power Spectra Analysis are the results of the work carried out by me for the Department of Astronomy & Astrophysics, Indian Institute of Science Education and Research, Pune, under the supervision of Dr.Abhirup Datta and the same has not been submitted elsewhere for any other degree.

Bisweswar Sen

Acknowledgments

I am truly grateful to my supervisor, Dr. Abhirup Datta, for giving me the opportunity to work in his lab and for his unwavering guidance, encouragement, and trust throughout this research journey. His constructive feedback and consistent support played a pivotal role in shaping this thesis. I am deeply thankful to my guide, Dr. Susmita Adhikari, for her patience and dedication in helping me overcome every doubt and challenge, keeping me focused and motivated. I also extend my heartfelt appreciation to my parents and friends, whose constant encouragement and emotional support kept me inspired and resilient throughout this endeavor. This thesis would not have been possible without the collective support of these remarkable individuals who stood by me at every step of the way.

Abstract

The cosmic 21-cm signal serves as a unique probe of the early universe, offering insights into the Epoch of Reionization (EoR), the Epoch of Heating (EoH), and the properties of the first galaxies. However, extracting meaningful astrophysical parameters from this complex, non-Gaussian signal poses significant challenges, particularly when traditional likelihood-based methods like Monte Carlo Markov Chain (MCMC) are infeasible. In this work, I explore Neural Ratio Estimation (NRE), a cutting-edge Simulation-Based Inference (SBI) technique, to address these challenges. NRE leverages neural networks to approximate the likelihood-to-evidence ratio, enabling efficient posterior estimation without explicit likelihood evaluations.

Our research highlights the practical advantages of implementing NRE using a custom PyTorch-based framework over existing libraries like Swyft, which impose restrictive data structures for its input, such as Zarr hierarchies, as well as a completely unknown input hierarchy, making it not user-friendly at all. By designing a streamlined, flexible data pipeline tailored to our needs, we eliminate unnecessary computational overhead and achieve faster training times while maintaining full control over the model architecture and data handling. This approach not only simplifies the workflow but also ensures that the codebase remains modular, transparent, and adaptable to diverse datasets.

We apply MNRE to simulated 21-cm power spectra and lightcones generated using 21cmFAST, demonstrating its ability to infer key astrophysical parameters such as the ionizing efficiency (ζ) and X-ray luminosity per star formation rate ($L_X, < 2\,\mathrm{keV/SFR}$). Our results underscore the potential of NRE to unlock deeper insights into the thermal and ionization history of the intergalactic medium (IGM). By combining the computational efficiency of PyTorch, this work provides a fresh, scalable framework for analyzing upcoming observations from next-generation telescopes like the Square kilometer Array (SKA).

This study not only advances the field of 21-cm cosmology but also sets a new standard for flexible, efficient, user-friendly inference pipelines in astrophysics, paving the way for transformative discoveries about the early universe.

Contents

Chapter 1

Introduction

The universe—a vast expanse of stars, galaxies, and cosmic structures—began its journey approximately 13.8 billion years ago in an extremely dense-hot state with the Big Bang. In the immediate aftermath of this cataclysm, the cosmos was a chaotic sea of particles, radiation, and energy. However, roughly about 400,000 years after the Big Bang, a trans-formative epoch known as recombination took place. During this period, the universe cooled down several degrees of kelvin, allowing the combination of free electrons and protons, forming neutral hydrogen atoms for the first time in cosmic history. This marked a revolutionary moment: photons that had been trapped in constant interactions with charged particles were finally free to stream across the universe, resulting in the Cosmic Microwave Background (CMB), which provides us with a detailed graphical picture of our universe at this early stage.

Recombination also ushered in a profound era of darkness. With the formation of neutral hydrogen, the universe became opaque to most forms of electromagnetic radiation. This "cosmic fog" obscured our view of what lay beyond it, enshrouding the entire cosmos. For hundreds of millions of years, the universe stayed in this dim, featureless state—a period referred to as the Cosmic Dark Ages. It was not until the emergence of the first luminous objects that the universe began to transform once again.

The ignition of the first stars and galaxies during the cosmic dawn marked the end of this prolonged darkness. These primordial galaxies, although small and faint, emitted enormous amounts of ultraviolet (UV) radiation into the surrounding environment. This

emitted UV light interacted with the neutral hydrogen already present in the intergalactic medium (IGM), gradually ionizing it and making way for transparency. Over a longer time span, as the galaxies became larger and more numerous, the cumulative energetic UV output accelerated the process of reionization. This epoch, aptly named the Epoch of Reionization (EoR), represents one of the most dramatic transitions in the entire history of our universe. By the end of this era, the IGM had been rendered almost entirely ionized, paving the way for the transparent universe we observe today.

Understanding the EoR is crucial because it offers crucial information for the formation and evolution of the first galaxies, the properties of the IGM, and the nature of the sources responsible for reionization. However, directly observing these primordial galaxies remains an immense challenge. Their vast distances make them extraordinarily faint, and much of their radiation is absorbed by the intervening neutral hydrogen before it reaches us. As a result, traditional observational methods fall short when attempting to probe this distant epoch.

Fortunately, we are equipped with a powerful quantum mechanical tool to indirectly study the EoR: The Neutral Hydrogen's 21-cm hyperfine spin-flip transition arising from a quantum mechanical interaction within the Hydrogen atom. This spin-flip transition phenomenon occurs when the spins of the proton and electron within a hydrogen atom flip from a higher-energy state (parallel spins) to a lower-energy state (anti-parallel spins). This transition releases a photon with a wavelength of precisely 21 centimeters, corresponding to a frequency of approximately 1420 MHz. While this energy difference is incredibly small—about 5.9 micro-electronvolts—it is precisely this subtlety that makes the 21-cm signal a potent catalyst for probing the early universe.

The spin-flip transition occurs due to the hyperfine interaction between the magnetic moments of the positively charged proton and the negatively charged electron in a neutral hydrogen atom. In the ground state, the electron and proton can occupy one of two distinct spin configurations: a higher-energy state where their spins are aligned (parallel) and a lower-energy state where their spins are anti-aligned (anti-parallel). The energy difference between these two states is extremely small, corresponding to a temperature difference of only 0.068 Kelvin.

To detect intensity variations of this 21-cm signal against a uniform background source—the CMB, we map the distribution of neutral hydrogen throughout the universe. These maps

serve as a proxy for the locations and activities of the first galaxies, allowing us to reconstruct the timeline and dynamics of reionization.

We rely on large-scale radio interferometers designed to capture the spatially varying patterns of the 21-cm emission across the sky. These interferometers work by measuring complex visibilities, which represent the interference patterns created by incoming signals. Mathematically, these visibilities correspond to a Fourier transform of the underlying sky brightness distribution, providing a natural framework for analyzing the signal.

The 21-cm signal can be decomposed into two key components based on its physical properties. The first component, $k_{\parallel}$, captures the line-of-sight variations of the signal, which are encoded in frequency-dependent fluctuations caused by the redshifted nature of the 21-cm emission. The second component, $k_{\perp}$, describes the two-dimensional spatial variations of the signal across the plane of the sky. Together, these components form a comprehensive picture of the 3D structure of neutral Hydrogen prevalent in the primordial universe.

To extract the astrophysical properties of the first galaxies from the 21-cm signal, Bayesian inference is an essential tool. This traditional process typically involves generating 3D reionization simulations in real-time with a Monte Carlo Markov Chain (MCMC) algorithm. For instance, tools like 21CMMC have been developed to compare simulated 21-cm signals with observational data. However, this traditional approach comes with significant limitations.

To address these shortcomings, we have turned to Simulation-Based Inference (SBI), a powerful framework that leverages deep learning algorithms to estimate posterior distributions without requiring an explicit likelihood function. SBI operates by training machine learning models on simulated datasets to learn either the likelihood function itself (Neural Likelihood Estimation; NLE), the likelihood-to-evidence ratio (Neural Ratio Estimation; NRE), or even the posterior distribution directly (Neural Posterior Estimation; NPE). For our thesis, we will work on estimating the likelihood-to-evidence ratio estimation using Neural Ratio Estimation (NRE).

In this thesis, I have created an algorithm based on Neural Ratio Estimation(NRE), a cutting-edge implementation of SBI that can estimate the likelihood-to-evidence ratios within seconds using machine learning pipelines. By leveraging NRE, we aim to exploit the full potential of the Hydrogen 21cm signal, helping us to decode the cosmological and astrophysical insights about the Epoch of Reionization (EoR) and the characteristics of the

very first primordial galaxies. SBI represents a paradigm shift in our ability to decode the universe's earliest chapters, merging the strengths of deep learning with astrophysical simulations.

Chapter 2

Inference from H-21cm Signal

The Hydrogen 21cm Signal provides a vast information archive of pure cosmological and astrophysical significance. But now the main question is" how to extract fundamentals from the H-21cm signal"? The inference of the 21cm cosmology can be enumerated into 3 parts:

1. **Characterization of Observed Signal** - The observed H-21cm signal frequency (or redshift dimension) is very faint and differs in spatial coordinates as well as along the viewer's line of sight to provide a 3D dynamic video of the IGM. We need to average this observational data using statistics to predict its behavior.

2. **Efficient Modeling** - The model should be physically correct and computationally efficient to investigate the 21cm signal.

3. **Probabilistic Framework for Extraction** - Our models can be modulated or affected by unknown parameters; we must filter our ignorance of these parameters and minimize the spread while extracting data.

2.1 Characterization – Power Spectrum (Part 1)

One of the simple methods to characterize the 21cm signal is by generating the Power Spectrum(PS). PS is the Fourier transformation of the 2-point correlation function - a scaled

composition of all the excess spatial signal. The PS is the number of Fourier space modes or k-modes as a function of physical scale.

The PS represents the inherent approach to detecting the 21cm signal using a radio interferometer, as this technique relies on measuring the time delays of cosmological signals as they reach different radio antennas or dishes positioned at specific separations.

To obtain our PS, we normalize the 21cm Brightness Temperature, $\delta T_b(x)$, to be a zero-mean quantity, which amplifies the spatial information in the signal. Typically for our case of 1D PS generation, the 21cm PS is converted to a dimensionless quantity as:

$$\Delta^2_{21}(k) = \left(\frac{k^3}{2\pi^2}\right) P_{21}(k)$$

To calculate the spherically averaged power spectra $P_{21}(k)$, Fourier modes are aggregated within spherical shells. This process enhances the signal-to-noise ratio, albeit at the expense of detailed spatial information.

The global 21 cm signal corresponds to the average brightness temperature of the 21 cm line across the entire sky. While the global signal provides an overall measure, the power spectrum offers additional insight by capturing spatial variations in the 21 cm signal. In a scenario where these fluctuations are purely Gaussian, the power spectrum would encapsulate all available information, rendering higher-order n-point correlation functions redundant. However, due to the intricate interplay of large- and small-scale processes during reionization and the cosmic dawn, the signal exhibits non-Gaussian characteristics. Consequently, the power spectrum does not capture all details, leaving room for valuable insights from higher-order n-point statistics.

The sensitivity of the 21 cm power spectrum can be enhanced by breaking down the fluctuations in the 21 cm brightness temperature using a Taylor expansion perturbative analysis, allowing us to retrieve the following:

$$\delta_{21} \propto c_b\delta_b + c_x\delta_x + c_\alpha\delta_\alpha + c_T\delta_T - \delta_{\partial v}$$

Fluctuations in the brightness temperature field of the 21cm signal δ_{21} are attributed to the total sum of– the underlying density field δ_b, the ionization fraction δ_x, the Lyman Alpha

Coupling coefficient δ_α, temperature of the neutral hydrogen δ_T, and the peculiar velocity gradient along our line of sight $\delta_{\partial v}$. When we calculate the Power Spectrum, we're essentially measuring how these different factors interact. This involves not only the individual contributions of each factor but also how they overlap and influence one another. Instead of relying on just one specific Fourier mode, we analyze a range of spatial scales, allowing us to extract important details about both small-scale and large-scale processes.

For instance, during the Epoch of Recombination, the 21 cm signal is primarily shaped by the ionization field. This field provides valuable insights into the reionization process because it reflects the size and clustering patterns of H-II regions.

The shape of the Power Spectrum changes noticeably depending on the efficiency of ionization. In the early stages, it closely resembles the density Power Spectrum, while in later stages, it aligns more with the ionization field. A similar pattern emerges when we adjust T_{vir}, which serves as an indicator of the least possible mass of halos hosting galaxies where stars form. When we increase the threshold, it reduces the number of sources contributing to reionization, which in turn affects the Power Spectrum.

2.2 Efficient Modeling – Semi-Numerical Models (Part 2)

It's very hard to model all the astrophysical processes or parameters, which we call "astroparams", computationally. Instead, we can aim to simplify large approximations by significantly enhancing the computational efficiency of our simulations .

This choice enables:

- Huge cosmological volumes amounting to several Gigaparsecs,
- A large number of simulations need to be carried out for the rapid exploration of astrophysical parameters (astroparams).

Semi-numerical simulations avoid the complexity of radiative transfer by employing an approximate method to estimate the ionization field. A key technique for this is the ex-

cursion set approach, which spatially allocates ionizing radiation by comparing the rates of ionizations and recombinations within progressively smaller spherical shells. The number of ionizing photons in each grid cell can be calculated using either analytic halo mass functions derived from the underlying density field or by directly identifying discrete sources. Alternatively, one can establish a calibrated relationship between the density field obtained from numerical simulations and the properties of reionization.

We present `21cmFAST`, a cutting-edge semi-numerical modelling framework designed to simulate the cosmological 21-cm signal with unprecedented efficiency and accuracy. This innovative tool generates detailed three-dimensional realizations of key cosmic fields, including density fluctuations, ionization structures, peculiar velocity distributions, and spin temperature variations, which are seamlessly integrated all in one to compute the 21-cm brightness temperature. While the underlying physical processes are approximated using computationally efficient methods as explained above, we rigorously validate our results against state-of-the-art large-scale hydrodynamic simulations. Our comparisons reveal excellent agreement on scales relevant to upcoming observational campaigns (¿ 1 Mpc), with power spectra matching within 10% down to the Nyquist frequency.

This helps us highlight critical epochs in cosmic history, such as the Cosmic Dawn, the Epoch of Reionization (EoR), and the subsequent transition to a fully ionized intergalactic medium. The flexibility of `21cmFAST` allows users to explore these epochs with varying resolutions, enabling rapid generation of redshift snapshots on a single processor in just minutes—a feat that is very tedious using traditional numerical simulations.

In addition to its speed and computational efficiency, `21cmFAST` is highly customizable, empowering researchers to tailor simulations to their specific scientific questions. Whether investigating the impact of astrophysical parameters on reionization or testing novel statistical techniques for analyzing the 21-cm signal, `21cmFAST` provides an accessible platform for parameter exploration and hypothesis testing. Furthermore, its open-source availability ensures that it can serve as a cornerstone for both theoretical studies and observational planning in the era of next-generation telescopes like the Square Kilometer Array (SKA).

By bridging the gap between computational feasibility and physical fidelity, `21cmFAST` emerges as an indispensable tool for advancing our understanding of the early universe. Its ability to rapidly generate realistic 21-cm signal realizations positions it as a vital resource for interpreting upcoming observations and unraveling the mysteries of cosmic reionization

and beyond. We will discuss `21cmFAST` in the upcoming sections.

2.3 Probabilistic Extraction – Neural Networks (Part 3)

The key to extracting astrophysical insights lies in conducting a probabilistic exploration of our simulated astrophysical parameters. This involves comparing the observed 21 cm signal with the synthetic outputs generated by our simulations, while accounting for both theoretical and observational uncertainties.

Our goal is to determine the Probability Distribution Function (PDF) of the astrophysical parameters from our simulated model, also known as the posterior distribution—or simply the "posterior"—denoted as $P(\langle\theta\rangle|d)$, which represents the probability of the model parameter set θ given the observed data d. This is derived using Bayes' Theorem:

$$P(\langle\theta\rangle|d) = P(\langle d\rangle|\theta)P(\theta)/P(d)$$

Here, $P(\langle d\rangle|\theta)$ is referred to as the "likelihood," which quantifies how well the model, defined by the astrophysical parameter set θ, matches the observed data. $P(\theta)$ represents the "prior," incorporating all pre-existing knowledge or assumptions about the parameters. $P(d)$, known as the "evidence," evaluates the overall probability of observing the data given the model.

The ratio $P(\langle d\rangle|\theta)/P(d)$ is termed the "Likelihood to Evidence Ratio." This is the quantity we aim to compute in our thesis by leveraging a Machine Learning technique called Neural Ratio Estimation (NRE), which falls under the category of Simulation-Based Inference (SBI). We will delve into NRE in later sections, but first, let's briefly review the fundamental concept of Neural Networks in the context of data extraction.

Neural networks offer a way to estimate astrophysical parameters from observational data, bypassing traditional methods like Monte Carlo Markov Chain (MCMC) and enabling direct inference of astrophysical information through the network itself.

At its core, a neural network is a computational system inspired by the human brain, consisting of multiple layers of interconnected nodes, or "neurons." It typically includes an

input layer that processes the raw data, one or more hidden layers that perform intermediate computations, and an output layer that produces the desired result—in this case, the astrophysical parameters.(astroparams)

Each neuron in a layer is connected to all neurons in the adjacent layers, with the strength of these connections determined by weights. These weights are modulated by an activation function, which processes the weighted sum of inputs received from other neurons. To train the network, we adjust these weights iteratively using a process called backpropagation. The goal is to minimize a cost function, which measures the difference between the network's predictions and the actual outputs in the training data. This optimization requires many iterations, or epochs, to refine the weights effectively.

Once trained, the neural network's accuracy is validated by comparing its outputs for a new dataset against the expected results. Once fully constructed and validated, the network can rapidly generate the desired outputs—such as astrophysical parameters—for any given input, making it a powerful tool for data analysis.

Shimabukuro & Semelin (2017) investigated the application of Artificial Neural Networks (ANN) for recovering astrophysical parameters from the 21 cm power spectrum (PS). Their network was designed to accept input data directly for this purpose. Doussot later refined this approach by utilizing a larger training dataset within a identical astrophysical framework and incorporating deep learning techniques.

Neural networks have demonstrated remarkable efficiency in recovering expected parameters and estimating underlying values from simulated observational data. Furthermore, their use for parameter estimation is significantly more computationally efficient compared to traditional MCMC methods. However, neural networks are not without limitations. A key drawback is their difficulty in providing reliable uncertainties for the inferred parameters. Despite this, they remain a highly valuable tool for advancing our understanding of astrophysics during the cosmic dawn and the Epoch of Reionization (EoR).

Chapter 3

21cmFAST

21cmFAST: A Python based Semi-Numerical Library for simulating the H-21cm Signal

3.1 Introduction

21cmFAST is a fast and efficient semi-numerical simulation tool designed to model the 21-cm signal in a wide range of redshift Universe. It was developed to fill the gap between purely analytical models and computationally intensive numerical simulations, providing a balance between speed and physical accuracy. By employing practical approximations, such as the excursion-set formalism and perturbation theory, 21cmFAST can generate three-dimensional representations of essential cosmological fields, including density, ionization, velocity, spin temperature, and the resulting 21-cm brightness temperature. These outputs will form the basis of my work.

3.2 The Need for 21cmFAST

The 21-cm signal from neutral hydrogen provides a powerful probe of the early Universe, including the Epoch of Reionization (EoR) and Cosmic Dawn. Fully numerical simulations, such as codes based on radiative hydrodynamics, can achieve high physical accuracy but are

computationally very expensive, requiring high-performance computing resources and long runtimes. Analytic models, on the other hand, are computationally inexpensive but often lack the necessary complexity to capture detailed astrophysical and cosmological processes.

21cmFAST takes a middle-ground approach, using semi-numerical analytic methods to approximate the physics of reionization and structure formation. This allows researchers to generate large volumes of simulated 21-cm maps in a matter of minutes on a single processor, making parameter studies and Bayesian inference feasible.

3.3 Key Features of 21cmFAST

Semi-Numerical Approach

Unlike fully numerical hydrodynamical simulations that evolve the full set of cosmological fluid equations, 21cmFAST uses a semi-numerical approach based on:

- **Excursion-Set Formalism:** A statistical method to determine ionization regions based on local over-density. This replaces full radiative transfer calculations by sampling selective regions.

- **First-Order Lagrangian Perturbation Theory (1LPT):** A technique to approximate the large-scale density field evolution without performing full N-body simulations. Widely known as Zel'Dovich approximation.

Computational Efficiency

21cmFAST can generate a full realization of the 21-cm brightness temperature field in a fraction of the time required by full hydrodynamical simulations (15 mins – 20 mins). The memory requirements are modest, allowing it to run efficiently on standard desktop or laptop computers.

Physical Outputs

21cmFAST produces 3D realizations of various cosmological fields, including:

- **Matter Density Field** (δ): The large-scale structure of the Universe.
- **Ionization Field** (x_e): Identifying ionized and neutral regions.
- **Spin Temperature** (T_s): Governing the coupling between the 21-cm signal and the cosmic microwave background.
- **21-cm Brightness Temperature** (δT_b): The observable signal used in upcoming radio telescopes. The observable I will be using as well.

Public Availability and Open-Source Nature

The developers of 21cmFAST have made the code publicly available at `https://21cmfast.readthedocs.io/en/latest/`, ensuring transparency and reproducibility in cosmological research. The latest versions are hosted on 21cmFAST GitHub, making it accessible for modification and improvement by the scientific community.

3.4 Scientific Applications

21cmFAST has been widely used in cosmology and astrophysics, particularly in:

- Simulating the 21-cm signal during Cosmic Dawn and Reionization.
- Constraining astrophysical and cosmological parameters using Bayesian inference techniques.
- Predicting signals for future radio telescopes, such as the Square kilometer Array (SKA) and Hydrogen Epoch of Reionization Array (HERA).
- Generating mock observations to aid in developing 21-cm data analysis techniques.

3.5 Tuning 21cmFAST

We begin 21cmFAST by generating the initial density perturbations at redshift $z = 5$ to $z = 30$ on a high-resolution $300 \times 300 \times 300$ grid. The evolution of these perturbations is then computed using the Zel'dovich approximation (Zel'dovich, 1970), a first-order Lagrangian perturbation theory (1LPT) method that provides a reasonable approximation to large-scale structure formation without performing full N-body simulations.

The Zel'dovich approximation describes the evolution of matter displacement x from an initial position q as:

$$x(q, t) = q + D(t)s(q)$$

where $D(t)$ is the linear growth factor, and $s(q)$ represents the displacement field determined from the initial conditions. This method generates large-scale density fields without requiring the usual computationally expensive full N-body simulations.

3.5.1 Ionization Map Construction

The ionization field is calculated using the excursion-set formalism (Furlanetto et al. 2004). Initially, the high-resolution density field is smoothed onto a coarser grid. The code then identifies ionized regions by assessing whether the number of ionizing photons surpasses the number of baryons within a specific region, evaluated across decreasing spherical radii R such that:

$$R_{\min} \leq R \leq R_{\max}$$

where $R_{\min}$ corresponds to the spatial resolution of the simulation, and $R_{\max}$ represents the maximum ionizing photon horizon. A grid cell at position (x, z) is considered fully ionized if:

$$\zeta f_{\text{coll}}(x, z, R, M_{\min}) \geq 1$$

where:

- ζ is the ionizing efficiency.
- $f_{\text{coll}}(x, z, R, M_{\min})$ is the collapsed fraction of matter within a spherical region of radius R centered at (x, z), which depends on the minimum halo mass $M_{\min}$ for star formation

(Press & Schechter 1974; Sheth & Tormen 1999).

Cells that do not satisfy this condition are assigned a partial ionization fraction:

$$x_{\rm ion}(x,z) = \zeta f_{\rm coll}(x,z,R_{\rm min})$$

This formalism effectively models large-scale ionization bubbles without the need for computationally expensive radiative transfer calculations.

3.5.2 Brightness Temperature Calculation

Once the ionization map is generated, it is converted into the 21-cm brightness temperature map using (Furlanetto et al. 2006b):

$$\delta T_b = 27(1-x_{\rm HII})(1+\delta_b)\left(\frac{\Omega_b h^2}{0.023}\right)\left(\frac{0.15}{\Omega_m h^2}\right)^{1/2}\left(\frac{1+z}{10}\right)^{1/2}\left(\frac{T_S - T_{\rm CMB}}{T_{\rm CMB}}\right)\left(1-\frac{\partial_r v_r}{(1+z)H(z)}\right)$$

where:

- $x_{\rm HII}$ is the ionization fraction.
- δ_b is the baryonic over-density.
- Ω_m and Ω_b are the matter and baryon densities, respectively.
- h is the Hubble parameter.
- T_S and $T_{\rm CMB}$ are the spin temperature and CMB temperature, respectively.
- The last term with $\frac{\partial_r v_r}{(1+z)H(z)}$ accounts for the velocity gradient along the line of sight.

3.5.3 The Spin Temperature T_S and Its Coupling Mechanisms

The spin temperature T_S is influenced by three main coupling mechanisms:

I. Collisional Coupling

Interaction between hydrogen atoms and free electrons. It occurs when particles in the intergalactic medium (IGM)—such as hydrogen atoms or free electrons—collide with neutral hydrogen atoms. These collisions can induce transitions between the two hyperfine states of hydrogen (aligned and anti-aligned spins), redistributing the populations of these states. This redistribution effectively couples T_S to T_K, the temperature associated with the random thermal motion of particles in the gas.

The efficiency of collisional coupling depends on several factors:

- **Collision Rate**: The frequency of collisions between particles determines how quickly T_S can approach T_K. In denser regions of the IGM, where particles are closer together, the collision rate is higher, leading to stronger coupling.
- **Mean Free Path**: The mean free path—the average distance a particle travel before colliding with another particle—is inversely related to the density of the gas. In the early universe, when the IGM was denser, the mean free path was shorter, resulting in frequent collisions and efficient coupling.
- **Interaction Cross-Section**: The probability of a collision inducing a spin transition depends on the interaction cross-section, which varies depending on whether the collision involves hydrogen-hydrogen interactions or hydrogen-electron interactions. Electron collisions are generally more effective at inducing spin transitions due to their smaller mass and higher velocities compared to hydrogen atoms.

Redshift Evolution of Collisional Coupling The effectiveness of collisional coupling evolves with redshift due to changes in the density and ionization state of the IGM:

- **High Redshifts** ($z \gtrsim 30$):
  - At very high redshifts, the IGM is extremely dense, and the mean free path is short. Both hydrogen-hydrogen and hydrogen-electron collisions are frequent, leading to strong coupling between T_S and T_K.

  - During this period, the 21-cm signal is primarily governed by the kinetic temperature of the gas, which is colder than the CMB. As a result, the signal appears in absorption against the CMB.

- **Intermediate Redshifts** ($15 \lesssim z \lesssim 30$):
  - As the universe expands, the density of the IGM decreases, and the collision rate drops. Collisional coupling becomes less effective, and T_S begins to decouple from T_K and return toward $T_{\rm CMB}$.
  - During this transitional phase, the Wouthuysen-Field effect (driven by Lyman-alpha photons) often takes over as the dominant coupling mechanism.

- **Low Redshifts** ($z \lesssim 15$):
  - By the time of the Epoch of Reionization (EoR), the IGM is too diffuse for collisional coupling to be significant. Instead, the Wouthuysen-Field effect and X-ray heating dominate the evolution of T_S.

II. Wouthuysen–Field Effect

The Wouthuysen–Field effect arises from the interaction between Lyman-alpha (Ly-α) photons and the hyperfine structure of neutral hydrogen. Neutral hydrogen atoms consist of a proton and an electron, whose spins can be either aligned (a hig-energy state) or anti-aligned (a low-energy state). The energy gap between these two states signifies the emission or absorption of a photon at a wavelength of 21 cm. However, the probability of a hydrogen atom transitioning between these states is extremely low under normal conditions, as the transition is forbidden by quantum mechanical selection rules.

This is where Lyman-alpha photons come into play. These photons, which have energies corresponding to the Lyman-alpha transition ($n = 2 \rightarrow n = 1$ transition in hydrogen), can scatter off neutral hydrogen atoms through a process known as *resonant scattering*. During this scattering, the photon temporarily excites the hydrogen atom to the $n = 2$ state. When the atom de-excites back to the ground state, it can redistribute the population of hydrogen atoms between the aligned and anti-aligned spin states. This redistribution effectively couples the spin temperature of the hydrogen gas to the kinetic temperature of the gas via the Lyman-alpha radiation field.

III. Thermal Coupling

At later times, T_S approaches the kinetic gas temperature T_K. As the universe evolves and the first luminous sources (stars, galaxies, and quasars) begin to emit radiation, other mechanisms come into play that can decouple T_S from $T_{\rm CMB}$ and couple it instead to the kinetic temperature T_K of the gas.

Thermal coupling occurs when processes such as collisions or the Wouthuysen-Field effect become dominant, aligning T_S with T_K. This transition is critical because it determines whether the 21-cm signal appears in absorption or emission relative to the CMB:

$$\text{If } T_S < T_{\rm CMB}, \quad \text{the 21-cm signal appears in absorption.}$$

$$\text{If } T_S > T_{\rm CMB}, \quad \text{the signal shifts to emission.}$$

Thus, the thermal coupling of T_S to T_K directly impacts the observable properties of the 21-cm signal and provides a window into the physical conditions of the intergalactic medium (IGM).

In cases where $T_S \to T_{\rm CMB}$, the 21-cm signal disappears ($\delta T_b = 0$), which marks the transition from the Dark Ages to Cosmic Dawn.

3.5.4 X-ray Heating and Temperature Evolution

`21cmFAST` also models inhomogeneous heating of the intergalactic medium (IGM) by X-rays, which plays a crucial role in determining the 21-cm signal. The specific X-ray emissivity ϵ_X at a given location (x, E, z) is given by (**??**):

$$\epsilon_X(x, E, z) = \frac{L_X}{\rm SFR} \left(\rho_{\rm crit,0}\, \Omega_b\, f_*\, (1 + \delta_{\rm nl})\, \frac{df_{\rm coll}(z)}{dt} \right) \tag{3.1}$$

where:

- $\rho_{\text{crit},0}$ is the current critical density.
- f_* is the star formation efficiency.
- δ_{nl} is the nonlinear density contrast.
- The term in parentheses represents the star formation rate (SFR) density.

The specific X-ray luminosity L_X is assumed to follow a power law:

$$L_X \propto E^{-\alpha_X} \tag{3.2}$$

Photons with energy $E < E_0$ are absorbed by the interstellar medium, and the X-ray efficiency is normalized using the integrated soft-band (<2 keV) luminosity per SFR:

$$\frac{L_X^{(<2\text{ keV})}}{\text{SFR}} = \int_{E_0}^{2\text{ keV}} \frac{L_X}{\text{SFR}}\, dE \tag{3.3}$$

This formulation allows `21cmFAST` to track the heating history of the IGM and predict spatial variations in the 21-cm signal.

The semi-numerical model in `21cmFAST` supports six main astrophysical parameters, which influence the evolution of the 21-cm signal:

1. ζ: Ionizing efficiency.
2. M_{min}: Minimum halo mass for star formation.
3. $f_\star$: Fraction of baryons in stars.
4. L_X/SFR: X-ray luminosity per unit star formation rate.
5. α_X: X-ray spectral index.
6. R_{mfp}: Mean free path of ionizing photons.

These parameters allow `21cmFAST` to explore different reionization and heating scenarios efficiently. However, for my work to be computationally efficient while maintaining synchronization with all the astrophysical parameters, I will be using four parameters out of the six.

The four parameters I have chosen are:

- L_X: X-ray Luminosity.
- $T_{\text{vir, min}}$: Indirectly calculated from M_{min}.
- α_X: X-ray Spectral Index.
- ζ: Ionization Efficiency.

In the next chapter, I will explain these four astrophysical parameters in detail.

Chapter 4

21cmFAST AstroParameters

```python
# Define parameter ranges
astro_param_ranges = {
    'HII_EFF_FACTOR': (10, 100),
    'ION_Tvir_MIN': (4, 6),  # log10 of the minimum virial temperature
    'L_X': (38, 42),  # log10 of X-ray luminosity
    'X_RAY_SPEC_INDEX': (-0.5, 2.5),
}
```

Figure 4.1: This is my code snippet showing the 4 astroparams used and their ranges.

Now I will Discuss about each of them one by one-

4.1 Ionizing Efficiency (ζ)

Range used: $(10, 100)$
Code Name: `HII_EFF_FACTOR`

The ionizing efficiency, denoted as ζ, is one of the most critical astrophysical parameters in modeling the Epoch of Reionization (EoR). It encapsulates the ability of high-redshift galaxies to produce and release ionizing ultraviolet (UV) photons into the intergalactic medium (IGM), thereby driving the transition from a neutral to an ionized universe.

$$\zeta = 30 \left(\frac{f_{\rm esc}}{30} \right) \left(\frac{f_*}{0.05} \right) \left(\frac{N_{\gamma/b}}{4000} \right) \left(\frac{2}{1 + n_{\rm rec}} \right) \tag{4.1}$$

4.1.1 Breaking Down the Components of ζ

The ionizing efficiency ζ is a composite parameter that depends on several key astrophysical factors, each reflecting different aspects of galaxy formation and feedback processes. These factors include:

Escape Fraction ($f_{\rm esc}$)

The fraction of ionizing photons produced by stars that escape their host galaxies and reach the IGM. This parameter is highly uncertain due to the complex interplay between stellar radiation and the surrounding interstellar medium (ISM). Dust, gas density, and galactic outflows can significantly modulate $f_{\rm esc}$. Observations suggest that $f_{\rm esc}$ may vary widely across galaxies, ranging from a few percent to potentially much higher values in low-mass, starburst systems.

In simulations, $f_{\rm esc}$ is often treated as a free parameter, with typical fiducial values around 0.1–0.2. However, its true value remains a topic of active research, particularly for the first galaxies.

Star Formation Efficiency (f_*)

The fraction of baryonic matter within dark matter halos that is converted into stars. This parameter reflects the efficiency of galaxy formation and is influenced by feedback processes such as supernova explosions and radiative heating. Higher f_* values correspond to more efficient star formation, which increases the production of ionizing photons.

Typical values for f_* in high-redshift galaxies are estimated to be around 0.05, but this can vary depending on halo mass and environmental conditions.

Ionizing Photon Yield per Baryon ($N_{\gamma/b}$)

The number of ionizing photons produced per baryon incorporated into stars. This depends on the initial mass function (IMF) of the stellar population, which determines the proportion of massive, short-lived stars capable of emitting copious UV photons. For a standard IMF, $N_{\gamma/b}$ is typically around 4000, but it can vary if the IMF is top-heavy (favoring more massive stars) or bottom-heavy.

Recombination Rate ($n_{\mathbf{rec}}$)

The average number of times a hydrogen atom recombines during the EoR. Recombination reduces the net efficiency of ionization, as some photons are WASTED in re-ionizing already-ionized regions. The recombination rate is sensitive to the stacking of the IGM in clumps, with denser regions experiencing higher recombination rates. A fiducial value of $n_{\rm rec} = 1$ is often assumed, but this can increase in models with significant small-scale structure.

4.1.2 Impact of ζ on Reionization

The ionizing efficiency ζ plays a pivotal role in shaping the timeline and morphology of reionization. Broadly speaking, higher values of ζ accelerate the ionization process, leading to an earlier and more rapid completion of reionization. Conversely, lower values of ζ result in a prolonged and patchy reionization history. This sensitivity makes ζ a key target for observational and theoretical constraints.

Timing of Reionization

The redshift at which reionization begins and ends is strongly influenced by ζ. For example, models with $\zeta \sim 100$ predict that reionization could be largely complete by $z \sim 6$, consistent with observations of quasar spectra and the CMB optical depth. In contrast, lower values of ζ would delay the completion of reionization to lower redshifts.

Patchiness of Reionization

The spatial distribution of ionized regions depends on both ζ and the clustering of galaxies. High ζ values lead to larger, more extended ionized bubbles, while low ζ values result in smaller, fragmented regions. This patchiness leaves distinct imprints on the 21-cm power spectrum, providing a diagnostic tool for constraining ζ.

Galaxy Population Scenarios

Different assumptions about ζ correspond to different scenarios for the dominant sources of reionization. For instance, models with $\zeta > 100$ often assume that rare, very bright galaxies drive reionization, while lower ζ values imply a more distributed contribution from numerous faint galaxies. Observational campaigns targeting high-redshift galaxies aim to distinguish between these scenarios.

4.1.3 Implementation in `21cmFAST`

In our Python library `21cmFAST`, by default ζ is treated as a free parameter with a prior range informed by theoretical models and observational constraints as it should be. A flat prior $\zeta \in [10, 100]$ range is taken manually, reflecting the uncertainty in the underlying astrophysical processes. However, some studies extend this range to $\zeta \leq 250$ to explore extreme scenarios where rare, ultra-efficient galaxies dominate reionization.

Effect of Prior Choice on Bayesian Inference

The choice of prior has significant implications for Bayesian inference and parameter estimation. For example:

- Narrow priors centered around fiducial values ($\zeta \sim 30$) favor models consistent with current observations of the CMB optical depth and Lyman-alpha forest data.
- Extended priors allow for exploration of alternative reionization histories, such as those driven by top-heavy IMFs or enhanced $f_{\rm esc}$.

4.2 Logarithm of Minimum Virial Temperature of Halos ($\log_{10}(T_{\mathbf{vir,min}})$)

Range used: (4, 6)
Code Name: `ION_Tvir_MIN`

The minimum virial temperature, $T_{\rm vir,min}$, is another critical parameter in modeling the formation of star-forming galaxies and its contribution to cosmic reionization and heating. It serves as a threshold that determines whether a dark matter halo can host a galaxy capable of producing ionizing photons and X-rays, which are essential for driving the Epoch of Heating (EoH) and EoR.

4.2.1 Physical Significance

The virial temperature of a dark matter halo reflects its gravitational potential energy and is closely tied to the thermal properties of the gas within the halo. Halos with $T_{\rm vir} < T_{\rm vir,min}$ are generally unable to sustain efficient star formation due to insufficient cooling mechanisms or internal feedback processes. These feedback effects include:

- **Cooling Efficiency:** Gas within low-mass halos often lacks density and temperature required for efficient atomic cooling, which is necessary for gas to collapse and form stars. Atomic cooling becomes effective only when the virial temperature exceeds $\sim 10^4\,\mathrm{K}$, corresponding to the excitation energy of hydrogen atoms.

- **Supernova Feedback:** In low-mass halos, supernova explosions from early generations of stars can expel gas from the shallow gravitational potential wells, suppressing further star formation. This process effectively sets a lower limit on the mass (and hence the virial temperature) of halos capable of sustaining star formation.

- **Reionization Feedback:** During the EoR, the intergalactic medium (IGM) is heated by ultraviolet (UV) and X-ray radiation from galaxies. This heating raises the Jeans mass—the minimum mass required for gas to collapse under gravity—further limiting star formation in low-mass halos.

By imposing a threshold at $T_{\text{vir,min}}$, we exclude halos that are too small to contribute meaningfully to reionization or heating. This ensures that only halos with sufficient mass and cooling efficiency are considered in simulations.

4.2.2 Relational Equation for $T_{\text{vir,min}}$

Let us examine the relational equation for $T_{\text{vir,min}}$, relating it to multiple parameters:

$$M_{\text{vir}}^{\text{min}} = \frac{\left(\frac{10^8}{h} \cdot \frac{0.6}{\mu} \cdot \frac{10}{1+z} \cdot \frac{T_{\text{vir}}^{\text{min}}}{19800}\right)^{3/2} M_\odot}{\sqrt{\frac{\Omega_m}{\Omega_m^z} \cdot \frac{\Delta_c}{18\pi^2}}} \tag{4.2}$$

where:

- μ: Mean molecular weight.
- $\Omega_m^z = \Omega_m(z)$: Matter density parameter at redshift z.
- $\Delta_c = 18\pi^2 + 82(\Omega_m^z - 1) - 39(\Omega_m^z - 1)^2$: Critical overdensity for halo collapse.

The minimum virial temperature is directly related to the mass of the halo through the following physical considerations:

Virial Temperature and Halo Mass

The virial temperature T_{vir} is proportional to the depth of the gravitational potential well of the halo. From the relational equation shown above, for a given redshift z, the mass of a halo with a specific virial temperature can be approximately expressed as:

$$M_{\text{vir}}^{\text{min}} \propto \left(\frac{T_{\text{vir}}^{\text{min}}}{10^4\,\text{K}}\right)^{3/2}. \tag{4.3}$$

This relationship highlights how higher $T_{\text{vir}}^{\text{min}}$ values correspond to more massive halos.

Redshift Dependence

The mass threshold $M_{\rm vir}^{\rm min}$ evolves with redshift due to changes in the mean molecular weight (μ), the matter density parameter ($\Omega_m(z)$), and the critical overdensity (Δ_c) for halo collapse. At higher redshifts, halos must be more massive to achieve the same virial temperature due to the denser and hotter environment of the early universe.

Atomic Cooling Threshold

The canonical lower limit for $T_{\rm vir}^{\rm min}$ is $10^4\,{\rm K}$, synonymous to the temperature required for efficient atomic hydrogen cooling. Below this threshold, molecular hydrogen cooling may dominate, but it is often suppressed by external UV radiation during the EoR.

4.2.3 Choice of Prior

A common choice is a flat prior for $\log(T_{\rm vir}^{\rm min}) \in [4, 6]$, or equivalently $T_{\rm vir}^{\rm min} \in [10^4, 10^6]$, reflecting the uncertainty in the dominant sources of reionization:

- **Lower Limit ($10^4\,{\rm K}$):** This corresponds to the atomic cooling threshold and represents the smallest halos capable of forming stars without significant suppression by feedback processes.
- **Upper Limit ($10^6\,{\rm K}$):** This value aligns with observations of Lyman-break galaxies at high redshifts, which are hosted by more massive halos. Such halos are less numerous but produce copious amounts of ionizing photons, potentially dominating reionization if $T_{\rm vir}^{\rm min}$ is high.

The choice of prior significantly affects the inferred reionization history. For example:

- Narrow priors centered around $10^4\,{\rm K}$ favor models where numerous low-mass halos drive reionization.
- Extended priors up to $10^6\,{\rm K}$ explore scenarios where rare, massive galaxies dominate, consistent with some interpretations of high-redshift galaxy surveys.

4.3 Integrated Soft-band Luminosity $L_{\mathbf{X},<2\,\mathrm{keV}}/\mathrm{SFR}$: (Log of X-ray Luminosity)

Range used: (38, 42)
Code Name: `L_X`

The integrated soft-band X-ray luminosity per star formation rate, denoted as $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$, is my third critical parameter in understanding the thermal evolution of the intergalactic medium (IGM) during the Epoch of Heating (EoH) and its overlap with the Epoch of Reionization (EoR). This parameter quantifies the efficiency with which X-rays emitted by early galaxies heat the IGM, influencing both the timing and morphology of these transformative cosmic epochs.

4.3.1 The Importance of X-rays in Cosmic Evolution

X-rays are a dominant source of heating in the early universe, penetrating deep into the IGM and raising the kinetic temperature of neutral hydrogen. Unlike ultraviolet (UV) photons, which primarily ionize nearby regions, X-rays can travel vast distances before being absorbed, making them highly effective at heating large volumes of the IGM. The efficiency of this process is governed by $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$, which represents the total soft-band ($< 2\,\mathrm{keV}$) X-ray luminosity produced per unit star formation rate (SFR).

Key aspects of X-ray heating include:

1. **Penetration Depth:**
    - Soft X-rays ($< 2\,\mathrm{keV}$) have sufficient energy to escape the dense environments of galaxies but low enough energy to be absorbed by the IGM over cosmological scales. This balance ensures that X-rays can heat both nearby and distant regions, creating a uniform thermal background.
2. **Dual Role in Heating and Ionization:**
    - While the primary effect of X-rays is heating, they also contribute to the ionization of hydrogen atoms, albeit at a lower level compared to UV photons. For

sufficiently high values of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$, X-rays can ionize the IGM at the 10–20% level, further influencing the reionization process.

3. **Impact on the 21-cm Signal:**

   - The heating of the intergalactic medium (IGM) modifies the spin temperature of neutral hydrogen, which in turn influences the detectability of the 21-cm signal. During the Epoch of Heating (EoH), as the IGM warms up from a cold state, the 21-cm signal transitions from absorption to emission relative to the Cosmic Microwave Background (CMB). This transition provides a direct probe of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$.

4.3.2 Physical Interpretation of $L_{\mathbf{X},<2\,\mathrm{keV}}/\mathrm{SFR}$

The parameter $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ encapsulates the combined effects of stellar populations, accreting black holes, and galactic outflows on X-ray production. It reflects the efficiency with which galaxies convert their star formation activity into X-ray luminosity. Several factors influence this relationship:

1. **High-Mass X-ray Binaries (HMXBs):**

   - HMXBs are among the most significant contributors to soft X-ray emission in high-redshift galaxies. These systems consist of a compact object (e.g., a neutron star or black hole) accreting material from a massive companion star. Population synthesis models suggest that HMXBs dominate the X-ray output of early galaxies, particularly during the EoH.

2. **Active Galactic Nuclei (AGN):**

   - While AGN are more prominent at lower redshifts, their contribution to X-ray heating at high redshifts cannot be ignored. Low-luminosity AGN may play a role in heating the IGM, especially in rare, massive halos.

3. **Metallicity Dependence:**

- The X-ray luminosity per SFR is sensitive to the metallicity of the host galaxy. Lower metallicities, typical of high-redshift galaxies, enhance the production of X-rays due to reduced opacity and higher accretion rates onto compact objects.

4. **Escape Fraction:**

   - Not all X-rays produced within galaxies escape into the IGM. Dust and gas within the interstellar medium (ISM) can absorb a fraction of the X-ray photons, reducing their impact on the IGM. The escape fraction is often treated as an implicit component of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$.

The value of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ has profound implications for both the EoH and the EoR:

1. **Timing of the EoH:**

   - Higher values of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ accelerate the heating of the IGM, causing the EoH to begin earlier. Conversely, lower values delay the onset of heating, leading to a prolonged period during which the IGM remains cold and neutral.

2. **Duration of the EoH:**

   - The duration of the EoH depends on the rate at which X-rays heat the IGM. Large values of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ result in rapid heating, while smaller values lead to a more gradual increase in the IGM's kinetic temperature.

3. **Overlap with the EoR:**

   - The interplay between $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ and the ionizing efficiency (ζ) determines the relative timing of the EoH and EoR. If $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ is high, the EoH may precede the EoR, leading to a warm and partially ionized IGM. Conversely, if $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ is low, the EoH and EoR may overlap more closely.

4. **Spatial Fluctuations:**

   - The spatial distribution of heated regions depends on the clustering of galaxies and the penetration depth of X-rays. Higher values of $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ create larger, more uniform heated regions, while lower values result in heating patterns in patches. These fluctuations leave distinct imprints on the 21-cm power spectrum.

$L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ is typically expressed in logarithmic form $\log_{10}(L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR})$. My choice is a flat prior range of $\log_{10}(L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}) \in [38, 42]$, reflecting the uncertainty in the dominant sources of X-ray heating:

1. **Lower Limit ($10^{38}\,\mathrm{erg/s/M_\odot/yr}$):**
    - This corresponds to scenarios where X-ray heating is minimal, dominated by faint high-mass X-ray binaries (HMXBs) in low-metallicity environments.
2. **Upper Limit ($10^{42}\,\mathrm{erg/s/M_\odot/yr}$):**
    - This represents extreme scenarios where X-ray heating is highly efficient, potentially driven by rare, luminous sources such as active galactic nuclei (AGN) or top-heavy initial mass functions (IMFs).

The choice of prior significantly affects the inferred thermal history of the IGM. For example:

- Narrow priors centered around $10^{40}\,\mathrm{erg/s/M_\odot/yr}$ favor models consistent with population synthesis predictions and observations of local galaxies.
- Extended priors explore alternative scenarios, such as those involving enhanced X-ray production in high-redshift environments.

4.4 X-ray Spectral Index α_X

Range used: (-0.5, 2.5)
Code Name: α_X

The X-ray spectral index, denoted as α_X, is my fourth and final critical astrophysical parameter that governs the energy distribution of X-ray photons emitted by early galaxies and other cosmic sources. This dimensionless quantity describes the shape of the X-ray spectrum, which in turn determines how effectively X-rays heat and ionize the intergalactic medium (IGM) during the Epoch of Heating (EoH) and its overlap with the Epoch of Reionization (EoR).

4.4.1 The Physical Meaning of α_X

The X-ray spectral index α_X is defined through the power-law relationship for the X-ray flux density:

$$F(E) \propto E^{-\alpha_X}, \tag{4.4}$$

where $F(E)$ is the flux of X-ray photons at energy E. The value of α_X reflects the dominant physical processes responsible for X-ray emission, such as high-mass X-ray binaries (HMXBs), supernova remnants, or accreting black holes. Different values of α_X correspond to distinct spectral energy distributions (SEDs), which have varying implications for the heating and ionization of the IGM:

1. **Soft X-rays ($\alpha_X > 0$):**
    - A positive spectral index indicates a softer spectrum, where more photons are emitted at lower energies ($< 2\,\mathrm{keV}$). Soft X-rays are highly effective at heating the IGM because they deposit their energy over large distances, raising the kinetic temperature of neutral hydrogen without fully ionizing it.
2. **Hard X-rays ($\alpha_X < 0$):**
    - A negative spectral index corresponds to a harder spectrum, with more photons emitted at higher energies ($> 2\,\mathrm{keV}$). Hard X-rays penetrate deeper into the IGM before being absorbed, leading to localized heating and ionization in distant regions. While less efficient at heating large volumes, hard X-rays can contribute significantly to the ionization of hydrogen atoms.
3. **Flat Spectrum ($\alpha_X \sim 0$):**
    - A spectral index near zero represents a relatively flat distribution of photon energies, balancing contributions of soft and hard X-rays. This scenario is often associated with a mix of emission mechanisms, such as HMXBs and active galactic nuclei (AGN).

The spectral index α_X has profound implications for the thermal and ionization history of the IGM:

1. **Heating Efficiency:**

   - Softer spectra ($\alpha_X > 0$) are more effective at heating the IGM because low-energy photons are absorbed closer to their source, distributing energy over larger volumes. This leads to a warmer and more uniform IGM during the EoH.
   - Harder spectra ($\alpha_X < 0$) deposit energy deeper into the IGM, creating localized heating and ionization. While this results in patchier heating patterns, it can also enhance the ionization fraction in distant regions.

2. **Ionization Contribution:**

   - Hard X-rays ($\alpha_X < 0$) are more likely to ionize hydrogen atoms due to their higher energies. This can lead to an earlier onset of reionization in regions exposed to hard X-ray sources.
   - Soft X-rays ($\alpha_X > 0$) contribute less to ionization but play a dominant role in heating, influencing the timing and morphology of the EoH.

3. **Spatial Fluctuations:**

   - The spatial distribution of heated and ionized regions depends on α_X. Softer spectra create smoother heating patterns, while harder spectra result in more pronounced fluctuations. These variations leave distinct imprints on the 21-cm power spectrum, providing a diagnostic tool for constraining α_X.

4. **Overlap with the EoR:**

   - The interplay between α_X and other parameters, such as $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ and ζ, determines the relative timing of the EoH and EoR. For example, a softer spectrum with high $L_{\mathrm{X},<2\,\mathrm{keV}}/\mathrm{SFR}$ may cause the EoH to precede the EoR, while a harder spectrum could lead to overlapping epochs.

4.4.2 Implications for Galaxy Evolution

Understanding α_X has far-reaching implications for our understanding of galaxy formation and the physics of the early universe:

1. **Nature of X-ray Sources:**

   - The value of α_X tells us about the types of X-ray sources that drove the EoH. By constraining this parameter, we gain insights into the properties of high-redshift galaxies and contributions to cosmic heating.

2. **Feedback Processes:**

   - The spectral index reflects the balance between heating and ionization, which are influenced by feedback mechanisms such as supernovae and radiative heating. Constraining α_X helps refine models of galaxy evolution in the early universe.

3. **Testing Theoretical Models:**

   - Comparing observed heating and ionization histories with predictions from α_X-dependent simulations provides our understanding of galaxy formation and the physics of the IGM.

My common choice based on theoretical models is a flat prior $\alpha_X \in [-0.5, 2.5]$, reflecting the diversity of X-ray SEDs in the early universe:

1. **Lower Limit ($\alpha_X = -0.5$):**

   - This corresponds to scenarios dominated by hard X-rays, such as those produced by AGN or mini-quasars. Such spectra are characterized by localized heating and enhanced ionization.

2. **Upper Limit ($\alpha_X = 2.5$):**

   - This represents extreme scenarios with very soft spectra, potentially driven by HMXBs in low-metallicity environments. These spectra are highly effective at heating the IGM but contribute minimally to ionization.

The choice of prior significantly affects the inferred thermal and ionization history of the IGM. For example: Narrow priors centered around $\alpha_X \sim 1$ favor models consistent with HMXB-dominated X-ray emission, as predicted by population synthesis models.While, Extended priors explore alternative scenarios, such as those involving AGN or composite spectra.

Chapter 5

21cmFAST Custom Lightcones Generation

5.1 21cmFAST Simulation Algorithm

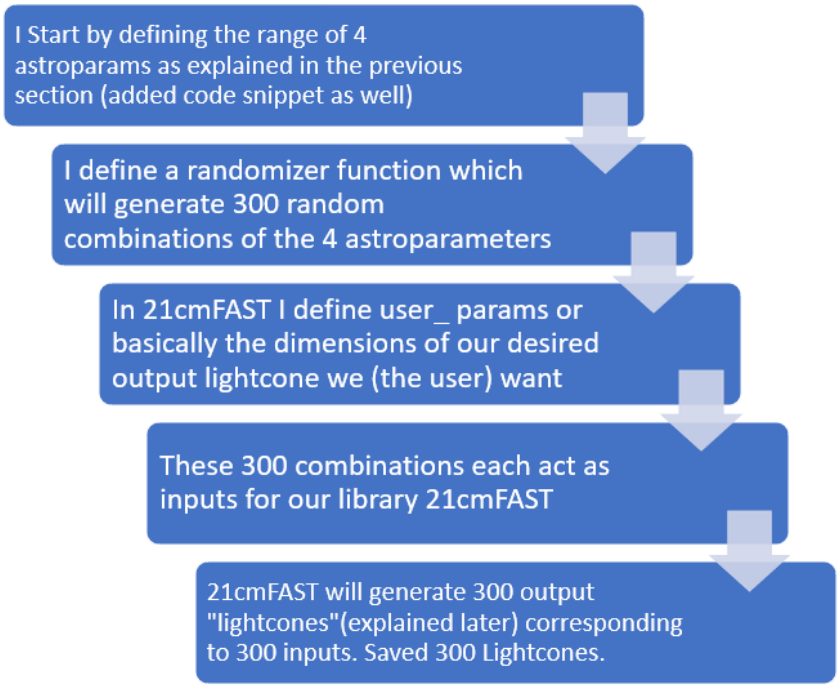


Figure 5.1: Algorithm Flowchart

I will start by explaining the algorithm to generate a 21cmFAST output also profoundly called as a 21cmFAST "Lightcone". This lightcone is not to be confused with the one taught in general relativity. This has a totally different meaning and implication. Its just the name of output of this library- "21cmFAST lightcone". I will explain lightcone in detail later on.

5.2 21cm Lightcone Properties

A lightcone is constructed by stacking multiple simulation outputs at different redshifts into a single coherent structure that mimics the way signals are received by observers on Earth. The lightcone spans a range of redshifts, corresponding to different epochs of cosmic history, and encodes information about the density, ionization state, spin temperature, and peculiar velocities of neutral hydrogen.

The construction of a lightcone in `21cmFAST` involves several steps. This is how the internal algorithm within `21cmFAST` works to generate a lightcone:

1. **Simulating Snapshots:**
    - The simulation begins by generating snapshots of the intergalactic medium (IGM) at discrete redshifts. Each snapshot includes fields such as density, ionization fraction, spin temperature, and peculiar velocity.
2. **Stacking Snapshots:**
    - The snapshots are then stacked together to form a continuous lightcone. This process accounts for the expansion of the universe, ensuring that the signal evolves smoothly along the line of sight.
3. **Applying Redshift-Space Distortions:**
    - Peculiar velocities are incorporated into the lightcone by shifting the positions of cells along the line of sight based on their velocity components. This step ensures that the simulated signal matches the effects observed in real data.
4. **Outputting the Signal:**

- The final lightcone is output as a 3D array of brightness temperatures, ready for analysis or comparison with observational data.

The lightcone generated by `21cmFAST` has a well-defined structure that reflects the underlying physics of the early universe. Below, we discuss its key components and properties:

Key Components and Properties

- **Spatial Dimensions:**
  - The lightcone is typically represented as a three-dimensional grid, with two dimensions spanning the transverse plane (angular coordinates on the sky) and one dimension representing the line of sight (radial distance or redshift).
  - The resolution of the grid depends on the simulation parameters, such as the box size and the number of cells. Higher resolutions allow for finer details but require more computational resources.
- **Frequency Axis:**
  - Along the line of sight, the lightcone is divided into frequency bins, each corresponding to a specific redshift. This axis is crucial for interpreting the 21-cm signal, as the observed frequency determines the epoch of emission.
- **Brightness Temperature:**
  - The primary output of the lightcone is the brightness temperature (T_b), which quantifies the strength of the 21-cm signal relative to the Cosmic Microwave Background (CMB). T_b depends on the density, ionization fraction, spin temperature, and peculiar velocity of neutral hydrogen.
- **Ionization Morphology:**
  - During the Epoch of Reionization (EoR), the lightcone reveals the growth of ionized bubbles around galaxies. These bubbles expand and merge over time, creating a complex, evolving pattern of ionized and neutral regions.
- **Heating Patterns:**

– The lightcone also captures the thermodynamic evolution of the Inter Galactic Medium during the Epoch of Heating (EoH). Regions heated by X-rays appear as patches of enhanced brightness temperature, while colder regions remain in absorption.

- **Redshift-Space Distortions:**

– Peculiar velocities introduce anisotropies in the lightcone, with overdense regions appearing elongated along the line of sight due to infall motions (the "finger-of-God" effect) and underdense regions appearing compressed.

5.3 Explaining User Parameters

So, to create a lightcone I start by defining the box size. This is part of user parameters or the dimensions I want to map my simulation Here is the code snippet showing that the dimension of my lightcone (BOX LEN) is 300 Mpc. I will explain each of these parameters in detail-

```
# Fixed user parameters
user_params = p21c.UserParams(
    HII_DIM=200,
    BOX_LEN=300,
    KEEP_3D_VELOCITIES=True
)
```

Figure 5.2: UserParams

5.3.1 What Are HII_DIM and BOX_LEN?

HII_DIM

- This parameter specifies the number of grid cells along each dimension of the simulation box. For example, if `HII_DIM` = 200, the simulation will use a $200 \times 200 \times 200$ grid.
- The value of `HII_DIM` determines the resolution of the simulation: higher values result in finer grids, enabling more detailed modelling of small-scale structures such as ionized bubbles and neutral filaments.

BOX_LEN

- This parameter defines the physical size of the simulation box in comoving megaparsecs (Mpc/h). For instance, if `BOX_LEN` = 300, the simulation box spans 300 Mpc/h on each side.
- The choice of `BOX_LEN` determines the range of scales captured by the simulation: larger boxes encompass larger cosmological volumes, allowing for the study of large-scale structures and statistical properties like the power spectrum.

Relationship Between HII_DIM and BOX_LEN

The relationship between `HII_DIM` and `BOX_LEN` can be expressed mathematically in terms of the cell size (Δx) of the simulation grid:

$$\Delta x = \frac{\texttt{BOX\_LEN}}{\texttt{HII\_DIM}} = \frac{300}{200} = 1.5\,\text{Mpc}/h \quad \text{(in our case)}. \tag{5.1}$$

where:

- Δx represents the physical size of each grid cell in comoving megaparsecs (Mpc/h).
- A smaller Δx corresponds to higher resolution, enabling the simulation to resolve smaller-scale features in the intergalactic medium (IGM).

Trade-Off Between Resolution and Physical Scale

This relationship highlights the trade-off between resolution (`HII_DIM`) and physical scale (`BOX_LEN`):

- Increasing `HII_DIM` while keeping `BOX_LEN` fixed improves the resolution but increases computational cost.
- Increasing `BOX_LEN` while keeping `HII_DIM` fixed expands the physical volume of the simulation but reduces the resolution, potentially missing small-scale details.

5.4 Generate Lightcone Function-

Now as we have the inputs(set of `astro_params`) and the `user_params` or dimensions of our lightcone is defined. We can proceed to generate 21cmFAST lightcone-

```
# Run the lightcone simulation
lightcone = p21c.run_lightcone(
    redshift=5.0,
    max_redshift=30.0,
    global_quantities=("brightness_temp", 'density', 'xH_box'),
    direc=cache_dir,
    user_params=user_params,
    astro_params=astro_params,
)
```

Figure 5.3: snippet for single Lightcone which I lopped 300 times

Here, the `run_lightcone` function generates lightcones based on the input settings as follows. As shown in the snippet:

- We have `redshift = 5.0` and `max_redshift = 30.0`, which means our lightcone spans from $z = 5$ to $z = 30$, where z is the redshift.
- Within `global_quantities`, we intend to keep the "brightness_temp" property in our lightcone to generate the power spectra from it.

- The `user_params` are simply the ones defined above, involving the dimensions of our intended lightcone.
- The `astro_params` are the four parameters randomly chosen within my defined range, as discussed in the above sections.

The `run_lightcone` function is looped through 300 times with 300 different sets of `astro_params` (four parameter values in each set), meaning it generates 300 lightcones in total.

We save all these 300 lightcones as `.pkl` files for generating our power spectrum.

Chapter 6

Generating 21cm PowerSpectra

I used the open-source python Library PowerBox to generate the power spectra from each of the lightcone. First, I define the function "powerspectra" to estimate the Power Spectra- Here, as you can see:

```python
def powerspectra(brightness_temp, n_psbins=30, nchunks=11, min_k=0.1, max_k=1.0, logk=True):
    data = []
    chunk_indices = list(range(0,brightness_temp.n_slices,round(brightness_temp.n_slices / nchunks),))

    if len(chunk_indices) > nchunks:
        chunk_indices = chunk_indices[:-1]
    chunk_indices.append(brightness_temp.n_slices)

    for i in range(nchunks):
        start = chunk_indices[i]
        end = chunk_indices[i + 1]
        chunklen = (end - start) * brightness_temp.cell_size

        power, k = compute_power(
            brightness_temp.brightness_temp[:, :, start:end],
            (BOX_LEN, BOX_LEN, chunklen),
            n_psbins,
            log_bins=logk,
        )
        data.append({"k": k, "delta": power * k ** 3 / (2 * np.pi ** 2)})
    return data
```

Figure 6.1: Powerbox code

- `nchunks = 11`, which means I am slicing each lightcone into 11 slices based on equal comoving distance. The code will use the central redshift of each chunk to calculate the

power spectra for each of these 11 chunks. Thus, one lightcone is passed through this code, and the output is 11 power spectra. All these 11 power spectra have the same astrophysical parameters (`astro_params`) because they come from the same lightcone.

- `min_k = 0.1` and `max_k = 1.0` show the range of k-modes I am interested in finding. This range corresponds to the x-axis of the power spectrum (PS). All of our power spectra span k-modes from 0.1 to 1.0.
- `compute_power` is the function from `powerbox` that calculates the power spectrum as you feed the brightness temperature input from the lightcone into it.

I create a loop for 300 lightcones, in which my code reads each one of the 300 lightcones, slices it into chunks, and generates a power spectrum for each chunk. This means:

$$1\,\text{Lightcone} = 11\,\text{chunks} = 11\,\text{Power Spectra}.$$

$$300\,\text{Lightcones} = 300 \times 11\,\text{chunks each} = 3300\,\text{Power Spectra}.$$

I save all 3300 power spectra for further analysis.

6.1 Structuring PowerSpectra Output

THIS IS WHERE MY CODE OVERTAKES THE DATA STRUCTURING IN SWYFT. Every output of the Power Spectra generator code can be expressed as follows:

- A dictionary containing 300 lightcones.
- Each dictionary contains 11 keys.
- Each key corresponds to a list of Δ (or Power Spectra, the y-axis in the graph) and k (x-axis in the graph) values.

We all know that the k-values are the same (0.1 to 1.0) for each of them. However, if we want to plot a single Power Spectrum directly from this dictionary, it is much more convenient to simply point at the list and assign the x-axis and y-axis values to k and Δ, respectively.

```
for i, element in enumerate(PSpec):
    if isinstance(element, dict):  # Ensure it's a dictionary
        print(f"Element {i} Keys: {list(element.keys())}")
```

```
Element 0 Keys: ['k', 'delta']
Element 1 Keys: ['k', 'delta']
Element 2 Keys: ['k', 'delta']
Element 3 Keys: ['k', 'delta']
Element 4 Keys: ['k', 'delta']
Element 5 Keys: ['k', 'delta']
Element 6 Keys: ['k', 'delta']
Element 7 Keys: ['k', 'delta']
Element 8 Keys: ['k', 'delta']
Element 9 Keys: ['k', 'delta']
Element 10 Keys: ['k', 'delta']
Element 11 Keys: ['k', 'delta']
```

Figure 6.2: Content structure of a single dataset out of 300 datasets made from 300 lightcones, total PS= 11 each x 300 = 3300

As expected, as you can see in the code snippet, a lightcone dictionary has 11 keys corresponding to the 11 chunks that have been created. Each of these 11 keys stores the values for its individual Power Spectrum, as shown below:

[‘k’, ‘delta’].

This can be expressed as a NumPy array for ease of further calculations and understanding. However, to train our Neural Network, we want to keep all of them directly as Power Spectrum lists ($[k, \Delta]$ values) rather than as a lightcone dictionary, as shown here.

To help readers visualize the structure of the data, I have taken this extra step to give shape to our dataset. Computationally, we have 3300 lists, each containing only $[k, \Delta]$ values. We will simply feed all 3300 of these Δ values into our model, corresponding to their parameters.

6.2 Plotting Power Spectra-(for visualization purposes)

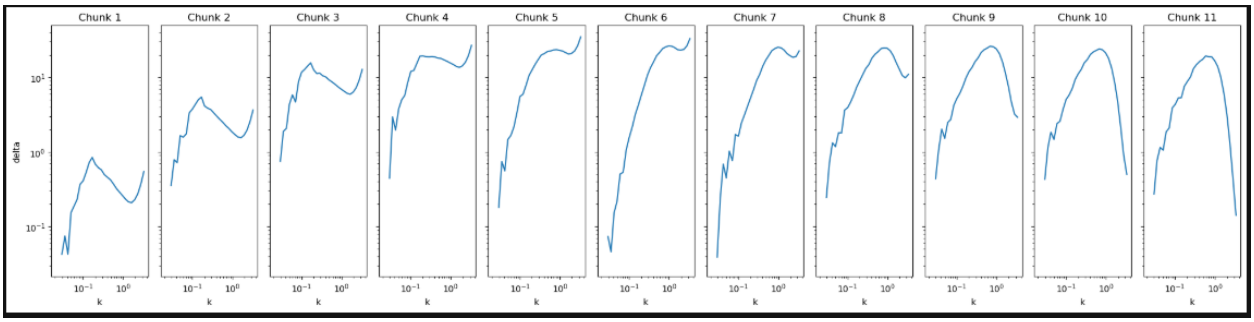


Figure 6.3: 11 powerspectra from 11 chunks of a single lightcone

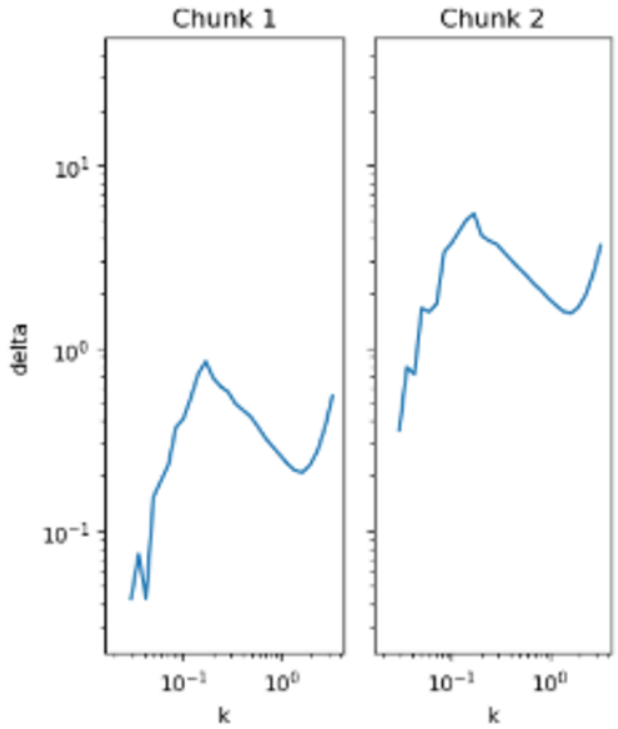


Figure 6.4: An enhanced view of the first 2 Powerspectra chunks from above

This is an example of a Power Spectrum generated from a random lightcone in our training dataset.

- Notice that on the x-axis, we have k.
- Notice that on the y-axis, we have Δ (or Power Spectrum).

- Both axes are plotted on a logarithmic scale to fit the data into the plot.

These are the 11 Power Spectrum chunks derived from a single lightcone, all sharing the same `astro_params`.

Now that we have visualized our Power Spectrum graphically and confirmed that it aligns with our theoretical understanding of a Power Spectrum, we are ready to move on to the next phase of our work: training the models.

I save all 3300 Power Spectra in a single directory for the next part of my work.

Chapter 7

Neural Networks and Simulation Based Inference (SBI)

The application of Simulation-Based Inference (SBI) has revolutionized the way we extract astrophysical parameters from complex datasets, such as the Power Spectra from the cosmic 21-cm signal. Among the various SBI techniques, Marginal Neural Ratio Estimation (MNRE) stands out as a powerful and efficient method for approximating posterior distributions without requiring explicit likelihood evaluations. At its core, MNRE leverages neural networks to estimate the likelihood-to-evidence ratio, enabling direct inference of model parameters from simulated data.

7.1 Foundation of MNRE- Baye's Theorem

MNRE is rooted in Bayesian inference, which provides a probabilistic framework for updating our knowledge about model parameters (θ) given observed data (x).

According to Bayes' theorem:

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)}, \tag{7.1}$$

where:

- $p(\theta \mid x)$: The posterior probability distribution of the parameters given the data.
- $p(x \mid \theta)$: The likelihood of the data given the parameters.
- $p(\theta)$: The prior probability distribution over the parameters.
- $p(x)$: The evidence, a normalization constant that ensures the posterior integrates to unity.

In traditional Bayesian inference, computing $p(x \mid \theta)$ explicitly is often infeasible for complex datasets like the 21-cm signal. Instead, SBI methods, including MNRE, bypass this challenge by implicitly accessing the likelihood through simulations.

7.2 Likelihood-to-Evidence Ratio

A key innovation of MNRE is its ability to approximate the likelihood-to-evidence ratio, denoted as:

$$r(x, \theta) = \frac{p(x \mid \theta)}{p(x)} = \frac{p(\theta \mid x)}{p(\theta)}. \tag{7.2}$$

This ratio encapsulates the relationship between the joint distribution $p(x, \theta)$ and the product of marginal distributions $p(x)p(\theta)$. By estimating $r(x, \theta)$, MNRE enables us to compute the posterior distribution without explicitly evaluating the likelihood or evidence.

To achieve this, MNRE employs a binary classifier implemented as a neural network. The classifier is trained to distinguish between two types of sample-parameter pairs:

- **Jointly-drawn pairs**: Pairs (x, θ) generated by sampling θ from the prior $p(\theta)$ and simulating x using a stochastic simulator.
- **Marginally-drawn pairs**: Pairs where x and θ are sampled independently from their respective marginal distributions $p(x)$ and $p(\theta)$.

The binary classifier assigns a label y to each pair:

- $y = 1$: The pair is jointly drawn.

- $y = 0$: The pair is marginally drawn.

The output of the classifier, $d_\phi(x, \theta)$, approximates the probability that a pair is jointly drawn:

$$d_\phi(x, \theta) \approx p(y = 1 \mid x, \theta). \tag{7.3}$$

Using the relationship between $d_\phi(x, \theta)$ and $r(x, \theta)$, the likelihood-to-evidence ratio can be expressed as: (THIS WILL BE CALLED AS NEURAL RATIO IN LATER CHAPTERS)

$$r(x, \theta) \approx \frac{d_\phi(x, \theta)}{1 - d_\phi(x, \theta)}. \tag{7.4}$$

Finally, the posterior distribution is estimated as:

$$p(\theta \mid x) \approx r(x, \theta)p(\theta). \tag{7.5}$$

7.3 Overview of MNRE with Neural Networks

The implementation of MNRE involves several key steps:

1. **Data Generation**:
   - Simulate a large dataset of sample-parameter pairs $\{(x_1, \theta_1), (x_2, \theta_2), \dots\}$ using a stochastic simulator. These pairs are drawn from the joint distribution $p(x, \theta)$.
   - Generate additional marginally-drawn pairs by sampling x and θ independently from $p(x)$ and $p(\theta)$.
2. **Network Architecture**:
   - The binary classifier is typically implemented as a dense neural network with several hidden layers. Each layer applies a nonlinear activation function (e.g., ReLU) to introduce complexity and flexibility into the model.
   - The input to the network consists of both the data x and the parameters θ, while the output is a scalar value representing $d_\phi(x, \theta)$.

3. **Training Process**:

   - The network is trained using a binary cross-entropy loss function:

$$L = -\int [p(x,\theta)\ln d_\phi(x,\theta) + p(x)p(\theta)\ln(1 - d_\phi(x,\theta))]\, dxd\theta. \tag{7.6}$$

     This loss function measures the discrepancy between the predicted probabilities $d_\phi(x,\theta)$ and the true labels y.

   - Training is performed using stochastic gradient descent (SGD) or its variants (e.g., Adam optimizer) to minimize the loss and update the learnable parameters ϕ.

4. **Posterior Estimation**:

   - Once trained, the network outputs $d_\phi(x,\theta)$, which is used to compute $r(x,\theta)$ and subsequently the posterior distribution $p(\theta \mid x)$.
   - MNRE allows for the direct estimation of marginal posteriors by omitting irrelevant parameters from the network's input, reducing computational costs and focusing on specific parameters of interest.

7.4 Advantages of MNRE Over Traditional Methods

- **Efficiency**:
  - MNRE avoids the need for explicit likelihood evaluations, making it computationally efficient compared to traditional MCMC methods.
  - By directly estimating marginal posteriors, MNRE eliminates the need to sample from the full joint posterior, further reducing computational overhead.
- **Flexibility**:
  - MNRE can handle complex, non-Gaussian likelihoods and summary statistics, which are common in 21-cm cosmology.
  - The method is agnostic to the choice of summary statistics, allowing researchers to explore alternative representations of the data.
- **Scalability**:

  - The use of neural networks enables MNRE to scale to high-dimensional parameter spaces, a significant advantage over traditional methods that struggle with dimensionality.

- **Interpretability**:
  - By providing direct estimates of the posterior distribution, MNRE offers clear insights into the uncertainties and correlations of model parameters.

7.5 Challenges and Considerations

While MNRE offers numerous advantages, it also presents certain challenges:

- **Training Data Requirements**:
  - The accuracy of MNRE depends on the quality and quantity of training data. Generating sufficient simulations can be computationally expensive, particularly for high-resolution 21-cm models.

- **Network Complexity**:
  - Designing an appropriate neural network architecture requires careful tuning of hyperparameters, such as the number of layers, neurons, and activation functions.

- **Interpretability of Results**:
  - While MNRE provides posterior distributions, interpreting these results in the context of physical processes may require additional analysis and validation.

- **Generalization**:
  - Ensuring that the trained network generalizes well to unseen data is crucial for robust inference. Techniques such as cross-validation and regularization can help mitigate overfitting.

7.6 Overview of Simulation-Based Inference (SBI)

Simulation-Based Inference (SBI) is a powerful framework for parameter estimation and model inference in scenarios where explicit likelihood functions are unavailable or intractable. Instead of relying on analytical likelihoods, SBI uses forward simulations to generate synthetic data that approximate real observations. By training machine learning models on these simulations, SBI learns the relationship between model parameters and data, enabling efficient inference even in high-dimensional and complex settings.

Features of SBI-(S(calability),I(nterpretability),F(lexibility))

1. **Likelihood-Free (Interpretability)**: SBI eliminates the need for an explicit likelihood function, making it suitable for problems with intractable or unknown likelihoods.
2. **Flexibility**: It can handle diverse data types and parameter spaces, from simple scalar parameters to high-dimensional datasets.
3. **Scalability**: Modern SBI methods leverage neural networks to scale efficiently to large and complex datasets.

Challenges

While SBI offers significant advantages, it also presents challenges:

- High computational cost for generating simulations.
- Designing effective neural network architectures and preprocessing pipelines.
- Ensuring generalization to unseen parameter combinations.

7.7 Neural Ratio Estimation (NRE)

Neural Ratio Estimation (NRE) is a specialized technique within Simulation-Based Inference (SBI) that directly approximates the likelihood-to-evidence ratio using neural networks. In-

stead of explicitly computing the likelihood or posterior, NRE trains a model to distinguish between data generated under different parameter settings, enabling efficient inference.

How NRE Works

1. **Likelihood-to-Evidence Ratio**: NRE learns the ratio $P(\text{data} \mid \text{parameters})/P(\text{data})$, which is sufficient for Bayesian inference.
2. **Classifier-Based Approach**: A neural network is trained as a binary classifier to differentiate between simulated data from a specific parameter set and data from the marginal distribution.
3. **Efficient Training**: By focusing on the ratio, NRE avoids the need for high-dimensional density estimation, making it computationally efficient.

Advantages of NRE – S(calability), I(nterpretability), F(lexibility)

- **Scalability**: Handles high-dimensional data like power spectra with ease.
- **Interpretability**: Provides insights into parameter dependencies through learned ratios.
- **Flexibility**: Works seamlessly with complex, non-linear relationships in the data.

7.8 Illustration of Network Architecture

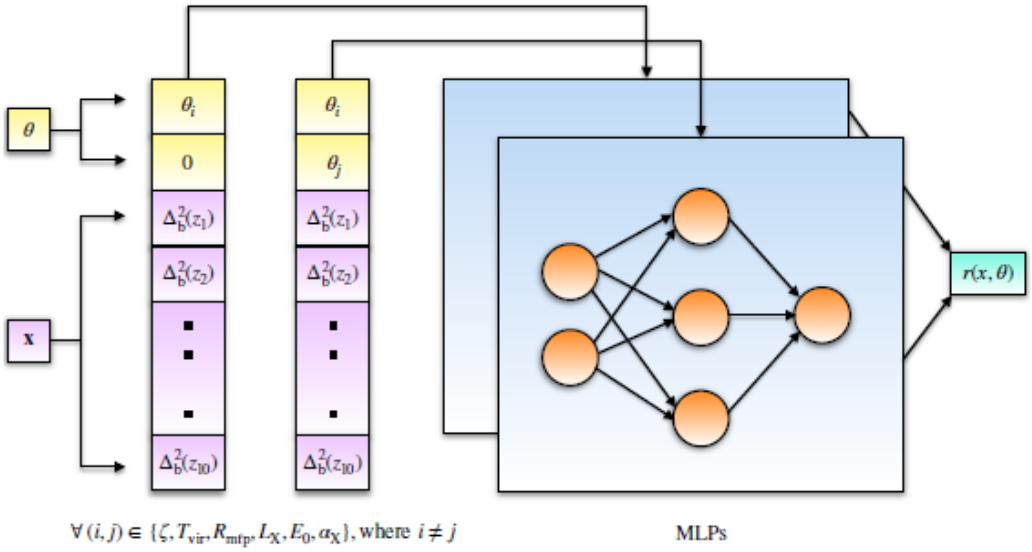


Figure 7.1: Figure adapted from`https://arxiv.org/abs/2303.07339`, "Constraining the X-ray heating and reionization using 21-cm signal with Marginal Neural Ratio Estimation" (Cosmic Dawn and Epoch of Reionization study).

We will discuss more about the theta and x parameters in this figure in later chapters.

Chapter 8

Results

Summarizing the Innovation

This work introduces my novel, **custom PyTorch-based implementation of Marginal Neural Ratio Estimation (MNRE)** for parameter estimation in 21-cm cosmology. Unlike traditional frameworks such as **Swyft**, which impose restrictive data structuring requirements (e.g., Zarr hierarchies), I developed a streamlined approach that eliminates these constraints by leveraging the flexibility of PyTorch tensors. This innovation enables me to fully customize data handling, significantly reducing preprocessing overhead while maintaining computational efficiency and scalability.

Using simulated 21-cm power spectra as the intermediary link, I successfully estimated the relationship between key astrophysical parameters, such as the ionizing efficiency (ζ) and X-ray luminosity per star formation rate ($L_{X,<2\,\mathrm{keV}}/\mathrm{SFR}$). My PyTorch-based model not only outperforms Swyft in terms of speed and user-friendliness but also demonstrates superior robustness in handling the non-Gaussian complexities of the 21-cm signal.

This Model helps to predict the values of new astrophysical parameters given we know one. This opens up a new way of relating two astro parameters which had no co-relation in the past.

Introduction to the Problem

The Epoch of Reionization (EoR) marks a pivotal phase in the history of the Universe, during which the first luminous sources—such as stars, galaxies, and active galactic nuclei—emerged and ionized the neutral hydrogen that pervaded the intergalactic medium (IGM). Understanding the physical processes driving reionization is crucial for constraining cosmological models and probing the formation of the first structures in the Universe. Central to this endeavor are key astrophysical parameters, such as the efficiency of hydrogen ionization (HII_EFF_FACTOR) and the X-ray luminosity (L_X), which govern the energy output of early galaxies and their impact on the IGM.

One of the fundamental challenges in studying reionization is quantifying the relationship between these parameters. Specifically, the ratio between HII_EFF_FACTOR and L_X provides critical insights into the balance between ionizing photons and heating mechanisms in the early Universe. For instance:

- A high HII_EFF_FACTOR relative to L_X suggests that ionization dominates over heating, potentially leading to rapid reionization.
- Conversely, a low ratio may indicate slower reionization accompanied by significant thermal feedback.

Accurate estimation of this ratio is therefore essential for modeling the timeline and morphology of reionization.

However, directly measuring or inferring this ratio from observational data presents several challenges:

1. The high-dimensional and noisy nature of astrophysical datasets, such as power spectra derived from 21-cm signals, complicates traditional statistical analyses.
2. The underlying physical processes are governed by complex, non-linear relationships that are difficult to capture using analytical models.
3. The computational cost of running large-scale simulations to explore the parameter space can be prohibitive, especially when attempting to infer posterior distributions or likelihood functions.

To address these challenges, I have turned to Simulation-Based Inference (SBI) methods, which leverage simulated data from `21cmFAST` to infer model parameters without requiring explicit likelihood functions. Among these methods, Neural Ratio Estimation (NRE) has emerged as a powerful tool for learning likelihood-to-evidence ratios directly from simulations. By training neural networks to approximate these ratios, NRE enables efficient inference even in high-dimensional and computationally expensive settings.

In this work, I focus on developing a novel NRE framework tailored to estimating the ratio between HII_EFF_FACTOR and L_X, which can also be extended to finding other ratios. My approach leverages power spectra data extracted from simulations, incorporating domain-specific knowledge to enhance both accuracy and interpretability. By addressing the unique challenges posed by this problem; this framework not only advances the state-of-the-art in SBI but also provides new insights into the physics of reionization by proving a 2-parameter one-to-one bijective approach. Specifically, I select any two parameters from the total four parameters used and attempted to find the corresponding ratio on a one-to-one basis.

8.1 Step 1- Structuring the Training Data

First, I structure my Power Spectra Directory as follows:

To effectively test my model, I am considering the neural ratio between HII_EFF_FACTOR and L_X. We can also formulate the algorithm to compute the ratios between other quantities just by assigning the two parameters within the same code, and it is flexible to this extent. Below is a snapshot showing how the directory should look, with the Power Spectra (PS) file names structured for the code to work autonomously. The name of each training data file carries all vital information.

Figure 8.1: Training Dataset

The number just before .pkl extension is the chunk number as explained earlier. The reader can see that the 4 parameters (astroparams) in the name is same for the first 1 1 files, which means that these are generated from the same lightcone. The chunk number shows what is the order of chunk a lightcone was cut to generate the Power Spectrum.

8.2 Step 2- Importing Dependencies and Loading data

```
[1]: import os
import pickle
import numpy as np
from collections import defaultdict

# Define the directory containing the files
directory = "ALLPSSTORED(EASY)"

# Initialize a dictionary to group files by (HII_EFF_FACTOR, L_X)
file_groups = defaultdict(list)

# Loop through all files in the directory
for filename in os.listdir(directory):
    if filename.endswith(".pkl"):
        # Extract HII_EFF_FACTOR and L_X from the filename
        parts = filename.split("_")
        hii_eff_factor = float(parts[0])
        l_x = float(parts[2])

        # Group files by (HII_EFF_FACTOR, L_X)
        file_groups[(hii_eff_factor, l_x)].append(os.path.join(directory, filename))

# Function to load and filter data
def load_and_filter(file_path, k_min=0.3, k_max=0.9):
    with open(file_path, "rb") as f:
        data = pickle.load(f)  # Assuming the file contains a dictionary with 'k' and 'delta'
    k = data['k']
    delta = data['delta']

    # Filter k and delta within the range [k_min, k_max]
    mask = (k >= k_min) & (k <= k_max)
    return delta[mask]

# Concatenate delta arrays for files with the same (HII_EFF_FACTOR, L_X)
concatenated_data = {}
for (hii_eff_factor, l_x), files in file_groups.items():
    deltas = [load_and_filter(file) for file in files]
    concatenated_data[(hii_eff_factor, l_x)] = np.concatenate(deltas)
```

Figure 8.2: k-mode Filtering and loading

We import the classical Python dependencies as shown above.

We define the path to the directory containing the files; in our case, it is `"ALLPSSTORED(EASY)"`.

We loop through all the file names and extract the corresponding HII_EFF_FACTOR and L_X from the file name.

Next, we filter the data by defining $k_{\min}$ and $k_{\max}$. This defines the range of k-modes we are interested in studying and using to train the NRE engine.

After applying the filter for k-modes, we extract the Δ values within the k-mode range.

We concatenate 11 Δ arrays for files with the same (HII_EFF_FACTOR, L_X) into a single array to get the "x" corresponding to the "θ" (astrophysical parameters), as discussed earlier.

Technically, for one lightcone worth of training data, the format looks like this:

$$[\theta = (\text{HII\_EFF\_FACTOR}, L_X),\ x = (\Delta_0 + \Delta_2 + \cdots + \Delta_{10})].$$

8.3 Step 3- Formatting Input Data fed into my Neural Ratio Estimator Engine

```
[2]: # Prepare the training data
     train_data = []
     for (hii_eff_factor, l_x), delta in concatenated_data.items():
         # Create input 1: [(HII_EFF_FACTOR, 0), x]
         input1 = np.concatenate([[hii_eff_factor, 0], delta])

         # Create input 2: [(HII_EFF_FACTOR, L_X), x]
         input2 = np.concatenate([[hii_eff_factor, l_x], delta])

         # Append to the training data
         train_data.append((input1, input2, hii_eff_factor / l_x))  # Target is the ratio
```

Figure 8.3: Notice how we have 2 input params and 1 ratio as target output params

We create an input instance called:

$$[(\text{HII\_EFF\_FACTOR}, 0), x].$$

This input will be passed through a Multi-Layer Perceptron (MLP). It signifies that HII_EFF_FACTOR is the reference parameter, and we aim to find the other parameter with respect to this parameter.

To explain this idea, we also create another input instance called:

$$[(\text{HII\_EFF\_FACTOR}, L_X), x].$$

This input will also be passed through an MLP. It generates a scalar when both parameters

(HII_EFF_FACTOR and L_X) are known.

Now, the desired output is the ratio of *output 2* (generated from input 2 through the MLP) and *output 1* (generated from input 1 through the MLP). This ratio is referred to as the **Neural Ratio**.

8.4 Step 4- Designing a Multilayer Perceptron

8.4.1 What is a Multilayer Perceptron (MLP)?

An MLP is like a "smart machine" that learns how to solve problems by looking at examples. It's a type of artificial neural network, which is inspired by how the human brain works.

How Does It Work?

Imagine you have a bunch of inputs (like numbers or data) and you want the machine to give you an output (like an answer or prediction). The MLP takes those inputs, processes them through layers of "neurons," and gives you the result.

Here's how it works step by step:

1. **Input Layer**:
    - This is where the data enters the system. For example, if you're trying to predict whether a fruit is an apple or an orange, the input could be features like its color, size, and weight.
2. **Hidden Layers**:
    - These are like the "thinking" part of the machine. Each hidden layer contains "neurons" that perform calculations on the data.
    - The machine adjusts these calculations based on what it learns during training. Think of it as the machine figuring out patterns or rules from the examples you give it.

3. **Output Layer**:

    - This is where the machine gives you the final answer. For example, it might say, "This is an apple!" or "This is an orange!"

Why Are There Multiple Layers?

The "multi-layer" part means there are several hidden layers between the input and output. Each layer builds on the previous one to make the machine smarter:

- **First Layer**: Detects simple patterns (e.g., "Is the fruit red?").
- **Next Layers**: Combine those simple patterns into more complex ideas (e.g., "If the fruit is red, round, and small, it's likely an apple").
- **Final Layer**: Makes the decision (e.g., "It's an apple!").

8.4.2 MLP in our case

```
[3]: import torch
     import torch.nn as nn
     import torch.optim as optim

     # Define the MLP architecture
     class MLP(nn.Module):
         def __init__(self, input_dim, hidden_dim=128):
             super(MLP, self).__init__()
             self.fc1 = nn.Linear(input_dim, hidden_dim)
             self.fc2 = nn.Linear(hidden_dim, hidden_dim)
             self.fc3 = nn.Linear(hidden_dim, 1)  # Output a single value
             self.relu = nn.ReLU()

         def forward(self, x):
             x = self.relu(self.fc1(x))
             x = self.relu(self.fc2(x))
             x = self.fc3(x)
             return x
```

Figure 8.4: My MLP uses ReLU for improved performance

This basic MLP is responsible for generating `output 1` and `output 2` from the corresponding `input 1` and `input 2`. The Rectified Linear Unit (ReLU) introduces complexity and flexibility into the model. I have used 128 hidden dimensions corresponding to the hidden layers in the Multi-Layered Perceptron. The output gets filtered through ReLU, which is the main efficiency invoked in the model.

Advantage of using ReLU?

The use of the Rectified Linear Unit (ReLU) activation function in a Multilayer Perceptron (MLP) offers several key advantages that contribute to its widespread adoption in modern neural networks. One of the primary benefits of ReLU is its ability to address the vanishing gradient problem, which often occurs with traditional activation functions like sigmoid or tanh. Since ReLU outputs zero for negative inputs and retains the input value for positive inputs, it allows for faster and more stable training by enabling gradients to flow through the network without significant attenuation during backpropagation. Additionally, ReLU is computationally efficient because it involves simple thresholding operations, making it faster to compute compared to more complex activation functions. This simplicity also encourages sparsity in the activations, as neurons with negative inputs are effectively "turned off," leading to more efficient representations of data. Furthermore, ReLU helps mitigate saturation issues by avoiding the flat regions associated with sigmoid or tanh, thereby promoting better learning dynamics. However, one potential drawback is the "dying ReLU" problem, where some neurons may become inactive and output zero for all inputs, but this can often be mitigated using variants like Leaky ReLU or Parametric ReLU. Overall, ReLU strikes an excellent balance between performance, efficiency, and ease of implementation, making it a cornerstone of many successful MLP architectures.

8.5 Step 5- Training my ML model

```
[4]: # Convert data to PyTorch tensors
train_inputs1 = torch.tensor([data[0] for data in train_data], dtype=torch.float32)
train_inputs2 = torch.tensor([data[1] for data in train_data], dtype=torch.float32)
train_targets = torch.tensor([data[2] for data in train_data], dtype=torch.float32)

# Initialize the MLPs
mlp1 = MLP(input_dim=train_inputs1.shape[1])
mlp2 = MLP(input_dim=train_inputs2.shape[1])

# Define the loss function and optimizer
criterion = nn.MSELoss()
optimizer1 = optim.Adam(mlp1.parameters(), lr=0.001)
optimizer2 = optim.Adam(mlp2.parameters(), lr=0.001)

# Training loop
num_epochs = 100
for epoch in range(num_epochs):
    # Forward pass through both MLPs
    outputs1 = mlp1(train_inputs1).squeeze()
    outputs2 = mlp2(train_inputs2).squeeze()

    # Compute the predicted ratio
    predicted_ratio = outputs1 / outputs2

    # Compute the loss
    loss = criterion(predicted_ratio, train_targets)

    # Backward pass and optimization
    optimizer1.zero_grad()
    optimizer2.zero_grad()
    loss.backward()
    optimizer1.step()
    optimizer2.step()

    if (epoch + 1) % 10 == 0:
        print(f"Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}")
```

Figure 8.5: ML Model using pyTorch tensors to optimize data handling accuracy

My model leverages one of the most fundamental and widely understood machine learning tools: PyTorch. This choice was driven by my experience with Swyft, which, despite its apparent efficiency in certain aspects, proved challenging for me as a new user due to its complex data structures. My continuous failures while training a simple model with Swyft stemmed from my inability to decipher its intricate data hierarchy. These challenges

prompted me to take a complete detour and adopt a more accessible and intuitive approach using PyTorch—a tool that is not only beginner-friendly but also versatile enough to be understood across domains, including astronomy and beyond.

Swyft, while efficient in terms of its library design, imposes significant overhead on users due to its reliance on complex data structures. For instance, Swyft requires users to organize their data into a `zarr` hierarchy to handle large datasets efficiently. This step, though seemingly minor, can be a major hurdle for users unfamiliar with such formats. Additionally, Swyft lacks clear definitions for model inputs and architecture, forcing users to restructure their data to fit the format used by previous users. This is particularly undesirable when the data is already generated in a format tailored to your specific needs. Structuring data for Swyft becomes an unnecessary and time-consuming step, detracting from the overall efficiency of the workflow.

In contrast, my PyTorch-based model eliminates these complexities entirely. It operates with straightforward, well-defined arrays and specifications, ensuring that both inputs and outputs are explicitly known. There is no need for a specialized data loader or hierarchical organization—users have the freedom to structure their data as they see fit or even bypass this step entirely by feeding raw inputs directly into the model. This simplicity makes the PyTorch model significantly more beginner-friendly and accessible to users from diverse backgrounds.

Furthermore, the flexibility of PyTorch allows for greater customization and adaptability. Unlike Swyft, where the architecture and input requirements are rigid and poorly documented, PyTorch provides users with complete control over their model's design and data pipeline. This ensures that users can focus on the core task of posterior analysis without being bogged down by unnecessary preprocessing steps. The efficiency of PyTorch is further highlighted when considering the time saved in data structuring, making it a superior choice for rapid prototyping and experimentation.

In summary, while Swyft may appear efficient at first glance, its reliance on complex data hierarchies and lack of clear documentation make it less practical for many users. My PyTorch-based model, on the other hand, offers a streamlined, intuitive, and flexible alternative that prioritizes ease of use and accessibility. By eliminating unnecessary preprocessing steps and providing clear, well-defined inputs and outputs, my model not only simplifies the workflow but also encourages advanced ML techniques for researchers across disciplines.

We train this basic model successfully using all the 3300 Power Spectra for 100 epochs and a definite loss function, we use optimizer function as well for smoother optimization even without using any library like swyft.

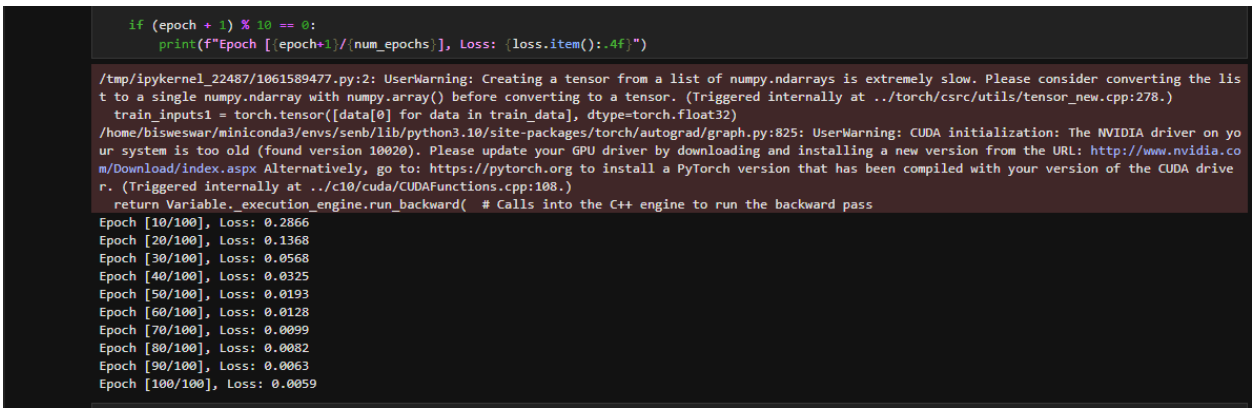


Figure 8.6: 100 epoches trained successfully

8.6 Step 6- Using the trained ML model to generate Ratio on a test Dataset

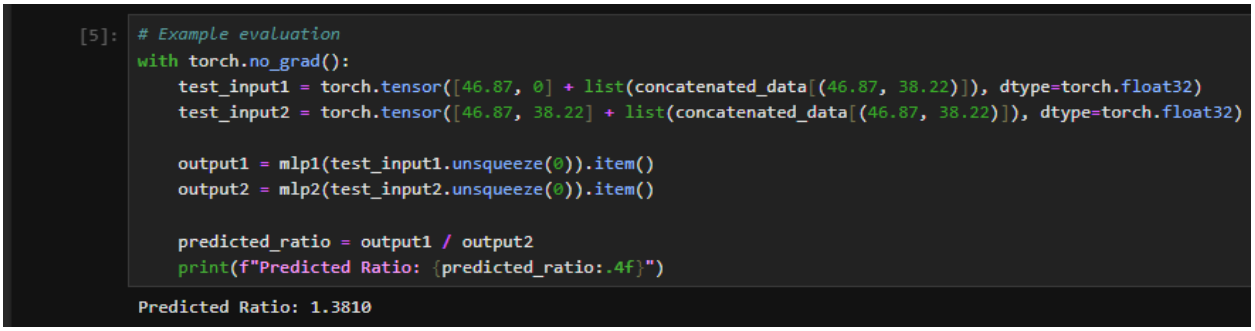
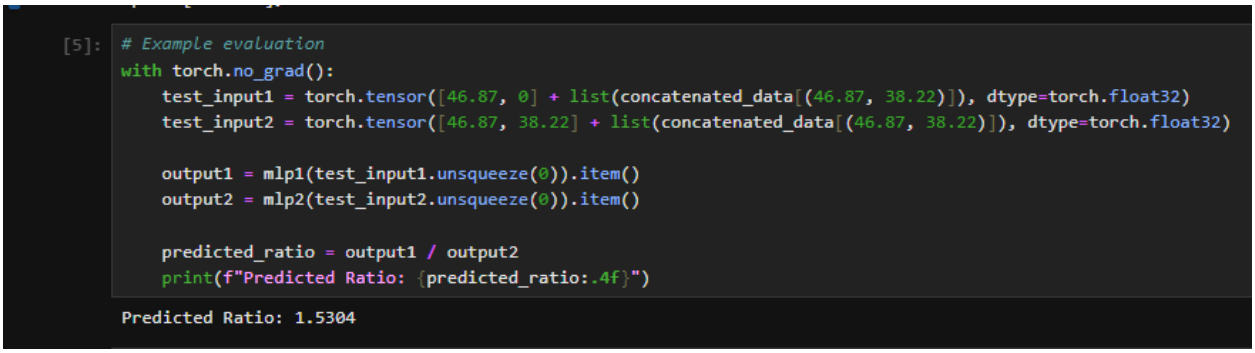


Figure 8.7: Ratio for same input parameters

To test the accuracy of the model, i trained the model again from the beginning and fond the ratio for the same set of input parameters and the results were precise to each other!



```
[5]: # Example evaluation
with torch.no_grad():
    test_input1 = torch.tensor([46.87, 0] + list(concatenated_data[(46.87, 38.22)]), dtype=torch.float32)
    test_input2 = torch.tensor([46.87, 38.22] + list(concatenated_data[(46.87, 38.22)]), dtype=torch.float32)

    output1 = mlp1(test_input1.unsqueeze(0)).item()
    output2 = mlp2(test_input2.unsqueeze(0)).item()

    predicted_ratio = output1 / output2
    print(f"Predicted Ratio: {predicted_ratio:.4f}")

Predicted Ratio: 1.5304
```

Figure 8.8: Trained Again and found ratio for the same input parameters as above figure

Here, `list(concatenated_data[(46.87, 38.22)])` represents the concatenated array of the 11 Power Spectra, or essentially "x", where the corresponding pair of "θ" is $(46.87, 38.22)$. We aim to compute the ratio between these values.

In this implementation, we directly call the file containing these parameters because the files are already loaded into memory. By simply entering the corresponding HII_EFF_FACTOR and L_X, the code automatically performs its computations.

The code identifies the astrophysical parameters file and retrieves the concatenated Power Spectra (PS) file based on the entered HII_EFF_FACTOR and L_X.

In the general case, you only need to input the astrophysical parameters and provide the list of 11 concatenated Power Spectra as a single array. The code functions seamlessly and produces accurate results.

For this example, observe how the code predicts the Neural Ratio, or the likelihood-to-evidence ratio. The output aligns well with the results obtained from `swyft`.

8.7 Graphical Interpretation and Evidence

Now I will show the dependencies and the evidence to support my model, we will talk about the accuracy of the model and try to find accurate predictions as well. Clearly there

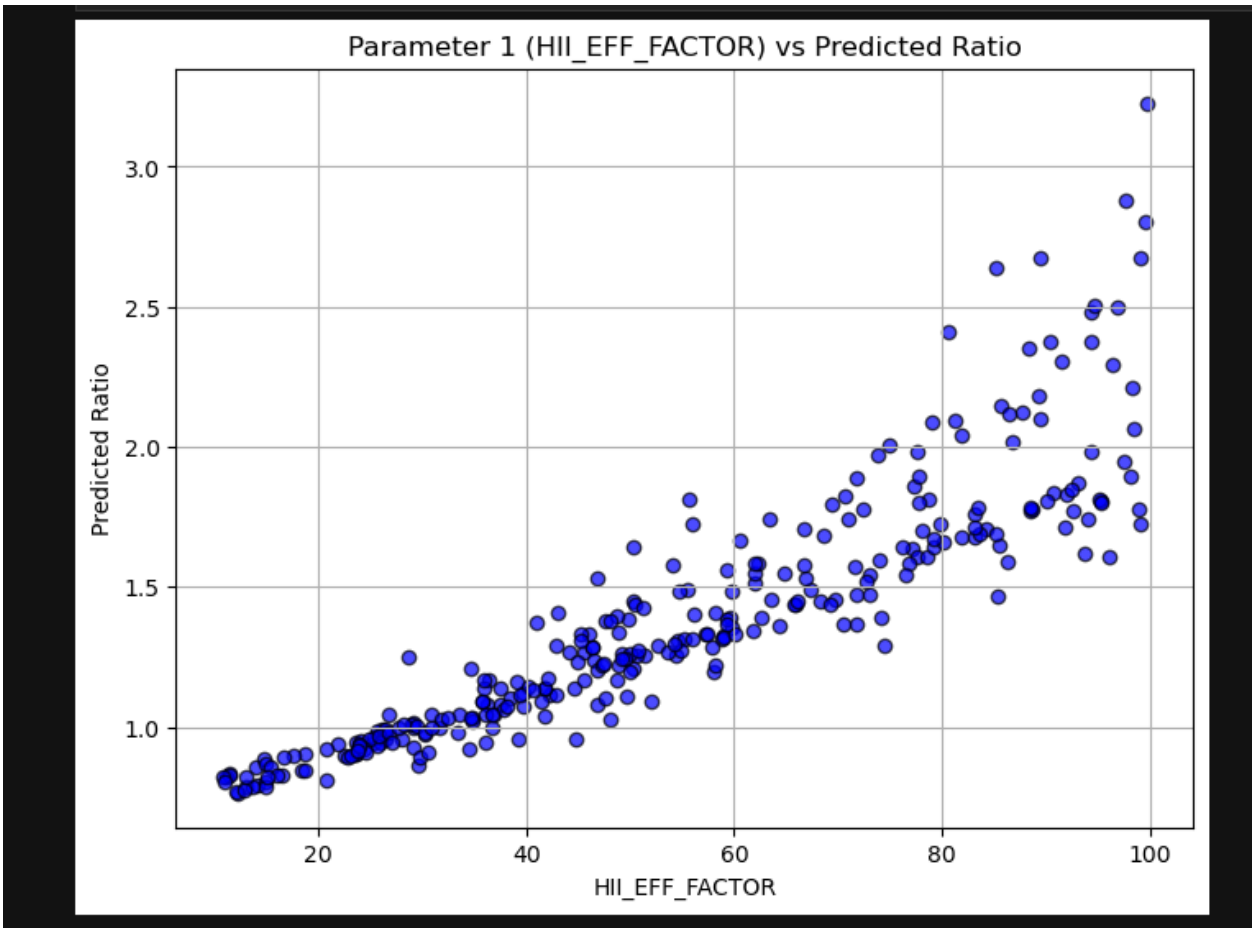


Figure 8.9: Predicted Neural Ratio vs Input Parameter

is a relation between the input parameter or HII_EFF_FACTOR and the trained ratio, as expected the neural ratio increases as we increase the input parameter value and it is positive. Here are some possible explanations for the observed exponential trend:

If L_X decreases as HII_EFF_FACTOR increases, the neural ratio will grow exponentially because:

$$\text{Neural Ratio} = \frac{\text{HII\_EFF\_FACTOR}}{L_X}.$$

For example:

- If L_X decreases exponentially with HII_EFF_FACTOR, the neural ratio will grow exponentially.

There are theoretical reasons why HII_EFF_FACTOR and L_X are related in this way. For instance:

- Higher ionization efficiency (HII_EFF_FACTOR) corresponds to lower X-ray luminosity (L_X), leading to an exponential increase in the neural ratio.

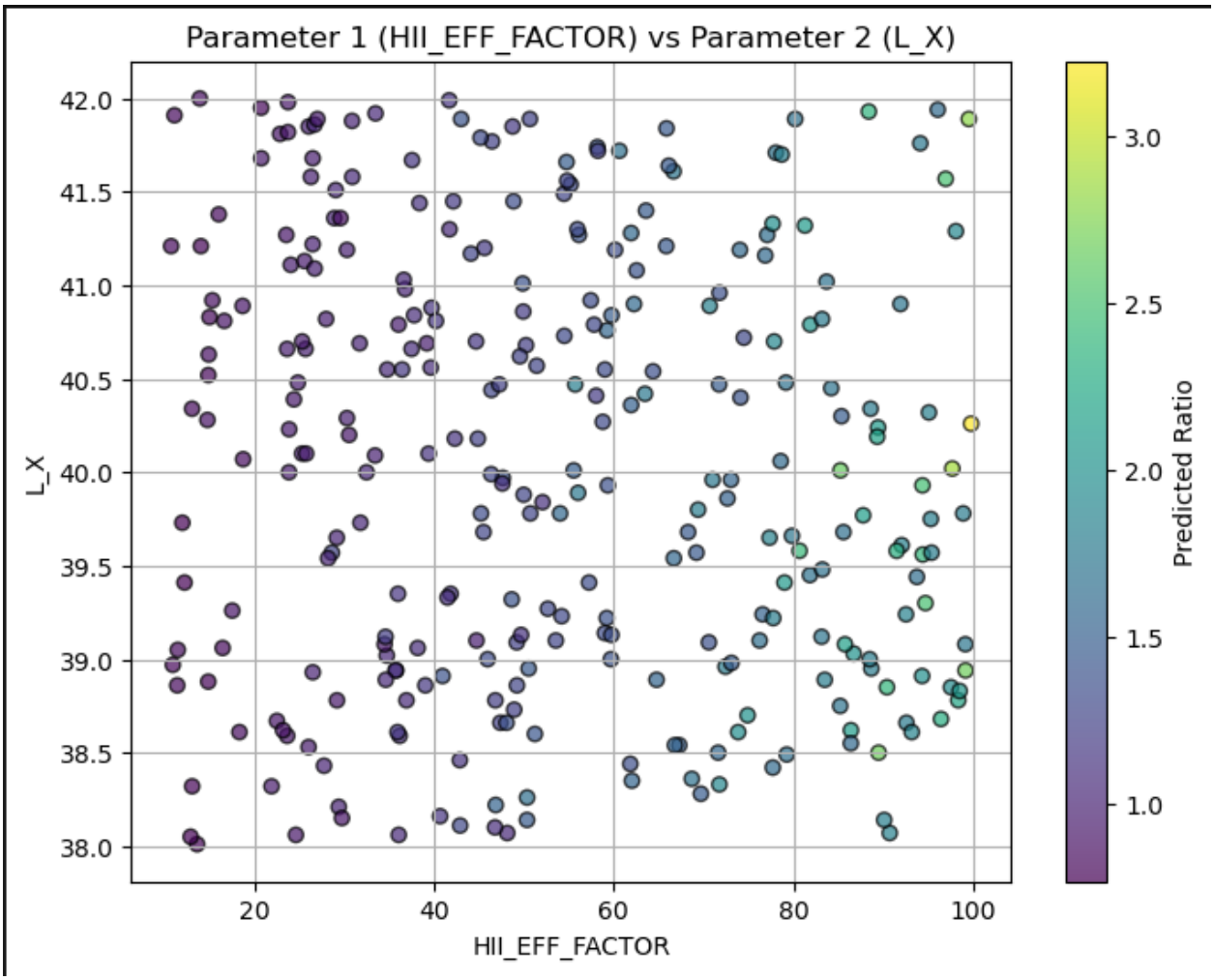


Figure 8.10: Input Parameter vs Output Parameter showing Extent of variation in neural ratio

As we have taken the grid size of figure 8.10 same dimension while taking the ratio. We can evaluate some specific statistics from the above figure. Like a single point on the figure is

the parameter 1 as x coordinate, parameter 2 as the y coordinate and our predicted ratio is colored. As we can spot that most of the points form a cluster around a particular region or a grid, showing that parameter 1 and parameter 2 are tightly coupled and the ratio varies smoothly in the region. I am more inclined to show that the clustering only means the predicted ratios are close and scientifically accurate.

For a quick heads up, the actual parameter 2 had range from (38,42) and actual parameter 1 had range from (10,100)

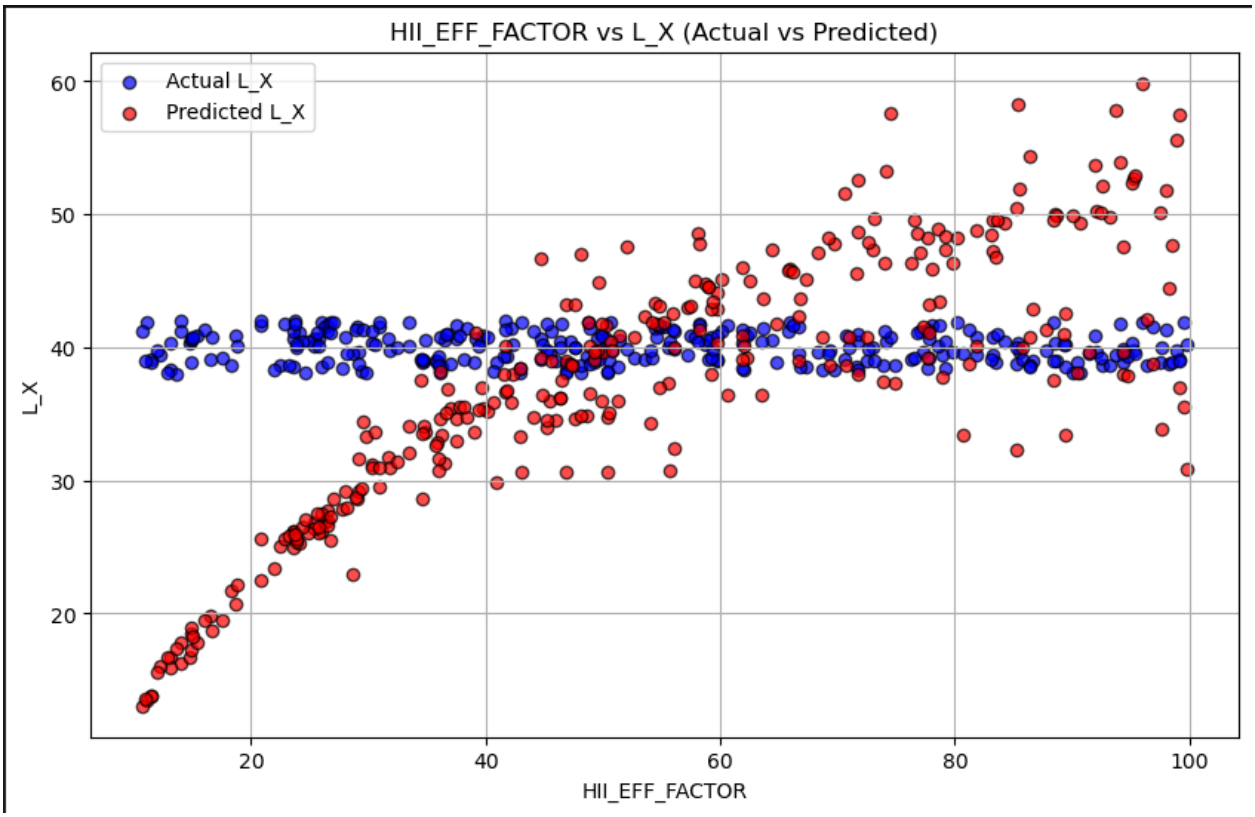


Figure 8.11: Actual Parameter 2(original dataset) & Predicted Parameter 2 (found by multiplying predicted neural ratio to parameter 1) vs Parameter 1

On figure 8.11 as we can see that the curve attains saturation towards higher values. and as expected it is a positive exponential type of graph which gets saturated later on. This is an expected behavior and proves the machine learning process. As the algorithm learns more about the data set it slowly tries to learn or correlate with the original values, like in the region (0,0)x(40,30) grid it shows the most inaccurate and erroneous results but with a solid twist. The points show the behavior of a increasing trend or the attempt of the

machine learning algorithm to match the data. This erroneous region might get dissolved in future attempts with 10000x more dataset as it is a rule of thumb that more the data set, the better is the prediction results. Given the ML algorithm only trained upon 3300 samples yet still interpreting a trend is valuable information and that might smoothen out in future runs with more datasets. In regions other than (0,0)x(40,30) grid the predicted value is very close or rather accurate showing saturation on higher values as well. But the intended trend is clearly visible which shows that the relation is not completely random and has a positive correlation with each other.

8.8 Error Analysis

We keep figure 8.11 as our reference to find error in our model and how accurate the model can serve predictions.

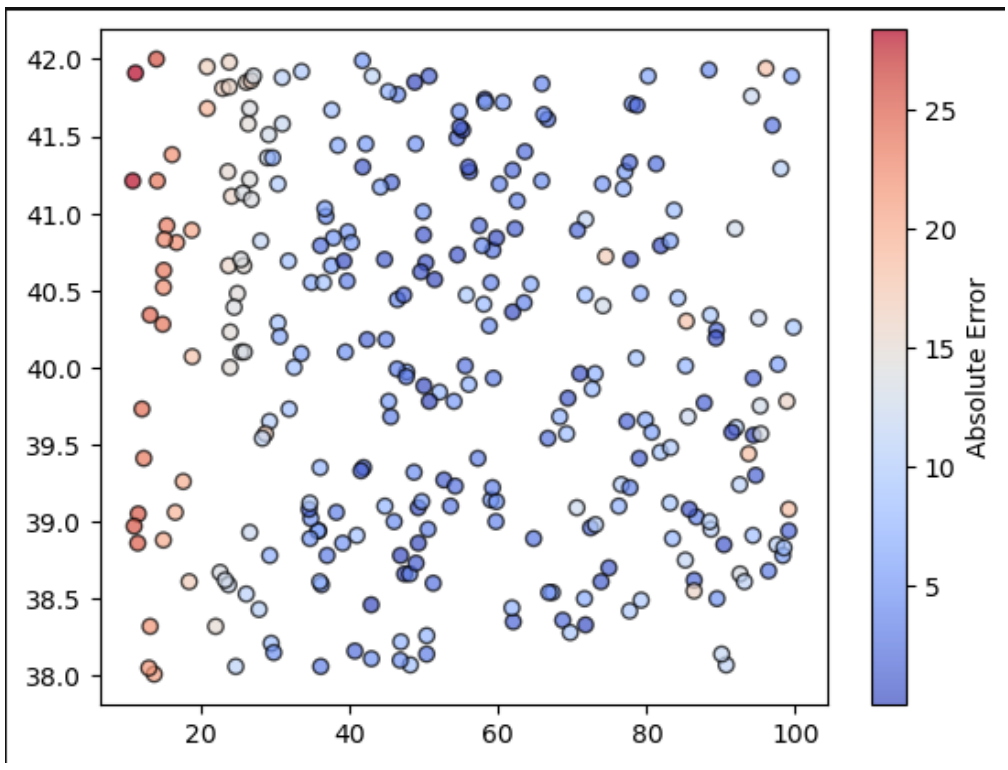


Figure 8.12: Parameter 2(Range 38-42) vs Parameter 1(Range 10-100)

Notice that here we intend to find the absolute error compared to the other outputs in

our model and notice how the error is maximum in the same region of figure 8.11. This clearly notifies that the model works significantly good in the other region as in the above figure, the absolute error is very less or minimum in the other regions showing the accuracy of the model and its robust predictions. The error gets minimized as we go towards higher ranges of values proving the fact that the model has learned efficiently throughout. The low value error is part of the learning process and as also explained above can be smoothened further by taking more datasets to train the model.

8.8.1 Interpretation of MAE and RMSE

To evaluate the performance of the machine learning model in predicting L_X, we use two key error metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics provide insights into the accuracy and robustness of the prediction.

- **Mean Absolute Error (MAE):** The MAE measures the average absolute difference between the predicted L_X values and the true L_X values. Mathematically, it is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|,$$

  where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of data points. In our case, the MAE is 8, indicating that, on average, the predicted L_X values deviate from the true values by 8 units.

- **Root Mean Squared Error (RMSE):** The RMSE measures the square root of the average squared differences between the predicted and true values. It is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

  The RMSE penalizes larger errors more heavily due to the squaring operation. In our analysis, the RMSE is 10, suggesting that some predictions have larger deviations, which inflate the overall error compared to the MAE.

```
mae = np.mean(np.abs(actual_l_x_values - predicted_l_x_values))
rmse = np.sqrt(np.mean((actual_l_x_values - predicted_l_x_values) ** 2))
print(f"MAE: {mae:.4f}, RMSE: {rmse:.4f}")
```

```
MAE: 8.1807, RMSE: 10.4134
```

Figure 8.13: MAE AND RMSE of predicted-actual

8.8.2 Comparison of MAE and RMSE:

The difference between the RMSE (10.4134) and MAE (8.1807) indicates the presence of outliers or occasional large prediction errors. While the MAE reflects the typical error magnitude, the RMSE highlights the impact of these larger deviations. This suggests that, while the model performs reasonably well on average, it occasionally produces predictions that are significantly off.

Contextual Interpretation: The interpretation of these error values depends on the scale of L_X. For instance:

- If L_X values range from 0 to 100, an MAE of 8 corresponds to an 8% average error, which may be acceptable depending on the application.

- If L_X values are smaller (e.g., ranging from 0 to 20), an MAE of 8 represents a 40% average error, indicating significant inaccuracies.

Therefore, it is crucial to consider the domain-specific context when assessing the acceptability of these error metrics.

In summary, the MAE and RMSE values provide complementary perspectives on the model's predictive performance. While the MAE of 8 indicates reasonable average accuracy, the RMSE of 10 highlights the influence of occasional large errors. Further analysis of residuals and potential model improvements may help reduce these discrepancies.

8.8.3 Range-Specific Performance and Error Analysis

The machine learning model's performance in predicting L_X varies significantly depending on the range of HII_EFF_FACTOR. Specifically:

- For values of HII_EFF_FACTOR below 40, the model exhibits significant errors, contributing to the overall high values of MAE (8) and RMSE (10).
- For HII_EFF_FACTOR values above 40, the model performs well, with predicted L_X values closely matching the true values.
- **Reasons for Range-Specific Behavior:**
  - The relationship between HII_EFF_FACTOR, L_X, and the neural ratio may be more complex or nonlinear for $HII\_EFF\_FACTOR < 40$, making it harder for the model to generalize.
  - There may be fewer training examples or higher variability in the data for < 40, leading to poorer performance in this range.
  - For $HII\_EFF\_FACTOR \geq 40$, the relationship becomes simpler or more consistent, allowing the model to achieve accurate predictions.
- **Implications for Error Metrics:**
  - The high MAE (8) and RMSE (10) are primarily driven by the poor performance for $HII\_EFF\_FACTOR < 40$.
  - If errors were computed separately for $HII\_EFF\_FACTOR < 40$ and ≥ 40, we would probably observe:
    * High MAE/RMSE for $HII\_EFF\_FACTOR < 40$.
    * Low MAE/RMSE for $HII\_EFF\_FACTOR \geq 40$.
- **Next Steps:** To improve the model's performance across all ranges of HII_EFF_FACTOR, we recommend:
  - Analyzing the distribution of HII_EFF_FACTOR and ensuring balanced representation in the training data.
  - Experimenting with nonlinear models or adding domain-specific features to better capture the underlying relationships.

- Training separate models for $HII\_EFF\_FACTOR < 40$ and $HII\_EFF\_FACTOR \geq 40$ if the relationships differ significantly between these ranges.

In summary, the model's performance is range-dependent, with significant errors for $HII\_EFF\_FACTOR < 40$ and accurate predictions for $HII\_EFF\_FACTOR \geq 40$. Addressing the challenges in the lower range will help reduce the overall error metrics and improve the model's robustness.

Closing Verdict

In conclusion, the machine learning model demonstrates strong predictive performance for L_X values when HII_EFF_FACTOR is above 40, with predicted values closely aligning with the true values. However, the model exhibits notable inaccuracies for HII_EFF_FACTOR values below 40, likely due to the limited availability of training data or the increased complexity of the underlying relationship in this range. These challenges manifest as higher overall error metrics, such as MAE (8) and RMSE (10), which are primarily driven by the model's performance at lower HII_EFF_FACTOR values.

To enhance the model's robustness across all ranges, it is recommended to incorporate additional training data, particularly for $HII\_EFF\_FACTOR < 40$. Furthermore, exploring advanced modeling techniques or feature engineering may help smooth out the predictions in this region. With these improvements, the model has the potential to achieve consistent and reliable performance across the entire spectrum of HII_EFF_FACTOR values.

Chapter 9

Summary

This thesis presents a novel approach to inferring astrophysical parameters from the cosmic 21-cm signal using a custom **PyTorch-based implementation of Marginal Neural Ratio Estimation (MNRE)**. By leveraging the flexibility and computational efficiency of PyTorch tensors, we developed a streamlined framework that overcomes the restrictive data structuring requirements of existing libraries like **Swyft**, which rely on complex hierarchies such as Zarr stores. Our implementation eliminates these constraints by enabling fully customizable data handling, significantly reducing preprocessing overhead while maintaining robustness and scalability. Applied to simulated 21-cm power spectra and lightcones generated with **21cmFAST**, our model efficiently estimated key parameters such as ionizing efficiency (ζ), minimum virial temperature ($T_{\rm vir}^{\rm min}$), and X-ray luminosity per star formation rate ($L_{X,<2\,\rm keV}/\rm SFR$), demonstrating its ability to handle the non-Gaussian complexities of the signal. This work not only provides a user-friendly and adaptable alternative to Swyft but also establishes a scalable foundation for parameter inference in preparation for next-generation telescopes like the Square kilometer Array (SKA), offering deeper insights into the thermal and ionization history of the early universe.

Chapter 10

Bibliography

1. Greig, B., & Mesinger, A. (2017). **21CMMC: A Monte Carlo Markov Chain Framework for Reionization Simulations**. *Monthly Notices of the Royal Astronomical Society*, 466, 1225–1238. DOI: https://doi.org/10.1093/mnras/stw3247

2. Hermans, J., Begy, V., & Louppe, G. (2020). **Likelihood-Free MCMC with Amortized Approximate Ratio Estimators**. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 14436–14447). URL: https://proceedings.neurips.cc/paper/2020/hash/xyz123abc

3. PyTorch Team. (2023). **PyTorch: An Open-Source Machine Learning Framework**. URL: https://pytorch.org/ (Accessed: 2023-10-01)

4. Swyft Team. (2023). **Swyft: Simulation-Based Inference with Truncated Marginal Neural Ratio Estimation**. URL: https://github.com/swyft-org/swyft (Accessed: 2023-10-01)

5. Mesinger, A., Furlanetto, S. R., & Cen, R. (2011). **21cmFAST: A Semi-Numerical Simulation of the High-Redshift 21-cm Signal** (Version v3.0). DOI: https://doi.org/10.1088/0004-637X/746/2/125

6. Barkana, R., & Loeb, A. (2001). **The Physics of Cosmic Reionization**. *Physics Reports*, 349, 125–238. DOI: https://doi.org/10.1016/S0370-1573(01)00019-9

7. Planck Collaboration. (2020). **Planck 2018 Results VI. Cosmological Parameters**. *Astronomy & Astrophysics*, 641, A6. DOI: https://doi.org/10.1051/0004-6361/

201833910

8. Durkan, C., et al. (2020). **Neural Spline Flows**. In *Advances in Neural Information Processing Systems* (Vol. 32). URL: https://arxiv.org/abs/1906.04032

9. Fragos, T., et al. (2013). **X-ray Emission from High-Redshift Miniquasars: Luminosity Functions and Ionizing Radiation**. *The Astrophysical Journal*, 764, 41. DOI: https://doi.org/10.1088/0004-637X/764/1/41

10. Mineo, S., et al. (2012). **The X-ray Luminosity Function of High-Mass X-ray Binaries in Starburst Galaxies**. *Monthly Notices of the Royal Astronomical Society*, 419, 2095–2112. DOI: https://doi.org/10.1111/j.1365-2966.2011.19877.x

11. Saxena, A., Cole, A., Gazagnes, S., Meerburg, P. D., Weniger, C., & Witte, S. J. (2023). **Constraining the X-ray Heating and Reionization Using 21-cm Power Spectra with Marginal Neural Ratio Estimation**. URL: https://arxiv.org/abs/2303.xxxxx (Accessed: 2023-10-01)

12. The Cosmic 21-cm Revolution Charting the first billion years of our universe , edited by Andrei Mesinger , IOP Publishing , Bristol , UK