AI Parameter Inference for the Epoch of Reionization: Addressing Issues of Model Generalizability and Interpretability

By

Jasper Solt

B.S. in Physics, Brown University, 2021

Sc.M. in Physics, Brown University, 2023

Thesis

Submitted in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy in the Department of Physics at Brown University

PROVIDENCE, RHODE ISLAND

May 2026

This dissertation by Jasper Solt is accepted in its present form by the Department of Physics as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date 4/13/2026 _____
Jonathan Pober, Co-Advisor

Date 4/13/2026 _____
Stephen Bach, Co-Advisor

Recommended to the Graduate Council

Date 4/13/2026 _____
Jonathan Pober, Reader

Date 4/13/2026 _____
Stephen Bach, Reader

Date 4/13/2026 _____
Ian Dell'Antonio, Reader

Approved by the Graduate Council

Date _____ _____
David P. Lindstrom, Dean of the Graduate School

Curriculum Vitae

Education

Brown University **Providence, RI**

PhD, Physics Expected May 2026

- Areas of expertise: Artificial Intelligence (AI), Machine Learning (ML), Data Analysis, Computational Physics, Cosmology, Radio Astronomy, Interferometry, Reionization, Cosmic Dawn.

BS, Astrophysics May 2021

- 3.94/4.00 GPA. Also fulfilled coursework required for a BA in Computer Science.

Publications and Presentations

- Solt, Jasper, Jonathan C. Pober, and Stephen H. Bach. "Mitigating Simulator Dependence in AI Parameter Inference for the Epoch of Reionization: The Importance of Simulation Diversity." arXiv preprint arXiv:2601.05229 (2026).
- AI Analysis of the Epoch of Reionization: Training Models Using Multiple Approximate Domains of Data. Poster. Physics in the AI Era, 2024, University of Pisa, Pisa, Italy.

Fellowships and Awards

- NASA Rhode Island Space Grant Fellow Summer 2023, Spring 2024
- Miss Abbott's School Alumnae Fellow Fall 2023, Spring 2024
- Smiley Prize for Excellent Contribution to the Astronomy Program Summer 2021

Research

Brown University **Providence, RI**

PhD Researcher May 2021 - May 2026

- Member of the Brown Radio Astronomy Group + BATS ML Research Group.
- Trained AI algorithms to extract parameters from simulated cosmological data.
- Analyzed / visualized results in Python using NumPy and Pyplot.
- Investigated how different sources of training data impact model overfitting / generalization.
- Used various explainable AI (XAI) methods to characterize model decision making.

Undergraduate Researcher, Brown Radio Astronomy Group Sep. 2018 - May 2021

- Trained AI algorithms to extract parameters from simulated cosmological data.

- Flagged radio frequency interference in Hydrogen Epoch of Reionization Array (HERA) data.

Teaching, Mentorship, and Outreach

Brown University **Providence, RI**

Teaching Assistant for PHYS0030 (Basic Physics A) Spring 2021, Spring 2024

- Ran discussion workshops where students collaborated on problemsets.
- Ran homework help sessions and office hours.
- Graded midterms and final exam.

Research Mentor Summer 2023

- Mentored undergraduate student on a research project involving AI analysis of cosmological data.

Teaching Assistant for PHYS0470 Labs (Electricity and Magnetism) Fall 2021

- Ran labs where students conducted experiments demonstrating various concepts in electricity and magnetism.

Teaching Assistant for Astronomy Courses Spring 2019, Fall 2019, Spring 2020

- Led labs where students identified constellations and planets, as well as took and processed CCD images of Messier objects using Brown's rooftop 16" SCT telescope.
- Held problem help sessions where students discussed homework and class material.
- Ran workshops where students collaborated on problemsets.

Ladd Observatory **Providence, RI**

Staff Member Sep. 2018 - May 2021

- Operated Ladd Observatory's telescope during public observation hours and private events.
- Identified night-sky objects and explained astronomical concepts for guests.

Skills and Interests

Coding Languages

- Expert: Python (especially PyTorch, Tensorflow / KERAS, NumPy, Astropy, SciPy, pandas, matplotlib), Linux / Bash scripting, LaTeX
- Familiar: Java, C# / .NET, C / C++, SQL

Software / Platforms

- Expert: 21cmFAST (cosmological simulation package) Google Workspace (docs, slides, sheets, etc.) Adobe Photoshop / Illustrator
- Familiar: Hugging Face Hub / API

Workflows

- Expert: Git / Github, Jupyter Notebooks / Google Colab, Slurm Workload Manager
- Familiar: Software development / unit testing / deployment, web scraping, database management

Data Formats

- Expert: Image data (all formats), HDF5, YAML, JSON, CSV

Acknowledgments

Over the last five years of my life I've been lucky to have the support of many wonderful people. First and foremost, I'd like to thank my advisors, Jonathan and Steve, who have acted as supportive and patient mentors from the start. None of this would have been possible without their guidance. I'd also like to thank the members of the Brown Radio Astronomy Group and the Bach's Awesome Team of Students (BATS) lab, who have provided community and assistance throughout the course of my research.

I'd also like to thank my family and friends for the love and commiseration given throughout the years: Eleanor, Luke, Nat, Miranda–thanks for always being there for me, and for helping me distract myself when the stress got to be too much. Mom, dad, Bekah, Elsa–thanks for being the best family I could ever ask for. I love you all and I can never express how much your support means to me. I'll also give a special thank you to my grandmother Stephanie, who without fail sent me a care package of baked goods and snacks every month for the entirety of my education. Those care packages saved me every time I was too busy or distracted to grocery shop.

Finally, I'd like to thank the two most important beings in my life: my husband, Evan, and my cat, Macchiato. Evan, I love you dearly. You've been here from the start, and have made the most material sacrifices to support my grad school ambitions out of anyone, following me to Providence and building your life around my program. I wouldn't have made it this far without your love, your support, and your care. Getting married to you was the best decision I've ever made. Macchi, thank you for distracting me, for

interrupting my zoom meetings, for walking on my keyboard, and for biting my legs when I sit for too long. You're a menace and have objectively made my work more difficult, but I wouldn't trade the experience for the world.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence. xii, 8–10, 15, 16, 19, 20, 22, 23, 30, 33, 36–38, 43, 55, 63

CMB Cosmic Microwave Background. 20

CNN Convolutional Neural Network. xii, xiii, 24, 25, 31

EoR Epoch of Reionization. xii, 3, 5, 7–11, 16–18, 20, 35–37, 39, 55, 57, 63

GPR Gaussian Process Regression. 9

IGM Intergalactic Medium. xi, 1, 3–7

ML Machine Learning. 8

MSE Mean Squared Error. xiii–xv, 25, 26, 30–34, 44–51, 56–58

OOD out-of-distribution. 8–11, 16, 57, 58, 60

XAI explainable AI. 37, 39

CHAPTER 1

The Epoch of Reionization

1.1 Observing the Intergalactic Medium: A Tale of Two Spectral Lines

The Intergalactic Medium (IGM) is the cosmic web of matter that exists between galaxies. With no stars to form metals, the baryonic matter of the IGM is almost entirely comprised of hydrogen and helium. As such, tracers of these elements can provide a wealth of information about the IGM. Hydrogen has two spectral lines of particular importance: the Lyman-α line, associated with the transition between hydrogen's ground and first excited states, and the 21cm line, caused by the hyperfine splitting of hydrogen's ground state (see Figure 1.1). The Lyman-α line provides significant information about the current state of the IGM, but it is the 21cm line that promises to reveal the most about the IGM's history.

First, we will consider what the Lyman-α spectra of quasars can tell us about the current state of the IGM. The Lyman-α optical depth τ_α is given by integrating the product of the effective scattering cross section σ_{eff} and the number density of neutral

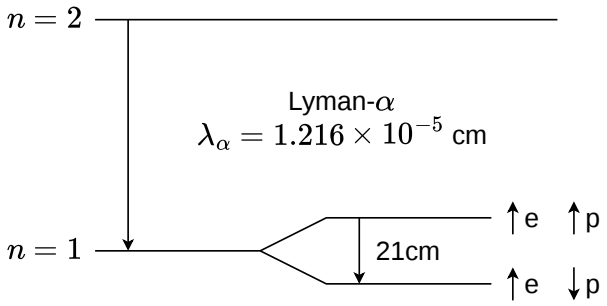

Figure 1.1: Two important energy transitions of hydrogen. The Lyman-α transition occurs when the hydrogen atom moves between the ground and first excited states, whereas the 21cm transition occurs when the atom's proton and electron go from aligned to anti-aligned.

hydrogen $n_{\rm HI}$ along the line of sight distance r:

$$\tau_\alpha = \int \mathrm{d}r \; \sigma_{\rm eff}(r) \; n_{\rm HI}(r) \tag{1.1}$$

It is more convenient to integrate across redshift z, so we use the relation $\mathrm{d}r = \frac{c}{H(a)} \frac{\mathrm{d}a}{a}$, where $a = \frac{1}{(1+z)}$ is the Hubble scale factor and $H(a)$ is the Hubble parameter. Using the Friedmann equation for a flat matter-dominated universe, we can also say that $H(a) = H_0 \sqrt{\Omega_m / a^3 + \Omega_\Lambda}$. Therefore:

$$\tau_\alpha = \frac{c}{H_0} \int \frac{\sigma_{\rm eff}(\nu_{\rm obs}/a) \; n_{\rm HI}(a)}{a \sqrt{\Omega_m / a^3 + \Omega_\Lambda}} \, \mathrm{d}a \tag{1.2}$$

We can ignore line-broadening effects and assume the Lyman-α line is narrow to define the cross section as a delta function centered on the Lyman-α frequency $\nu_\alpha = 2.47 \times 10^{15}$ Hz:

$$\sigma_{\rm eff}(\nu) = \sigma_\alpha \nu_\alpha \delta(\nu - \nu_\alpha) \tag{1.3}$$

where $\sigma_\alpha = 4.48 \times 10^{-18}$ cm^2 is the Lyman-α cross section. Substituting and integrating,

we find:

$$\tau_\alpha = \frac{\sigma_\alpha c \, n_{\rm HI}(z)}{H(z)} \tag{1.4}$$

Finally, we can express the number density of neutral hydrogen $n_{\rm HI}$ as the neutral fraction of hydrogen $x_{\rm HI}$ multiplied by a deviation δ from the mean cosmic density $\bar{n}_{\rm H}$:

$$\begin{aligned} \tau_\alpha &= \frac{\sigma_\alpha c \, x_{\rm HI} \bar{n}_{\rm H}(1+\delta)}{H(z)} \\ &= 1.6 \times 10^5 \, x_{\rm HI} \, (1+\delta) \left(\frac{1+z}{4}\right)^{3/2} \end{aligned} \tag{1.5}$$

Equation 1.5 tells us that the IGM is opaque to Lyman-α radiation at even very small neutral fractions ($x_{\rm HI} \sim 10^{-5}$). Therefore, for Lyman-α transmission to occur at all, the IGM must be nearly completely ionized. Our observations of quasar spectra tell us the universe is, indeed, highly transparent to Lyman-α radiation at $z \leq 5.5$ (Fan et al., 2006; Becker et al., 2015; see Figure 1.2). This means the IGM must be almost entirely ionized at low redshifts.

However, following recombination at $z \sim 1100$, the IGM should have been completely neutral. This means that sometime between then and $z = 5.5$, the IGM underwent a phase transition from neutral to fully ionized. Following Cosmic Dawn, the first stars and galaxies release ionizing radiation into the IGM, forming ionized bubbles around high density regions. These bubbles grow and merge until the entire IGM is ionized. This period during which this process occurs is referred to as the Epoch of Reionization (EoR).

The 21cm signal emitted by neutral hydrogen is expected to yield rich information about the EoR, as it serves as a tracer for the distribution of neutral hydrogen in the universe and can therefore directly map the fraction of neutral hydrogen $x_{\rm HI}$ over time. The 21cm line is "forbidden", meaning it has a very low likelihood of occurring for any individual ground state hydrogen atom. However, the abundance of hydrogen in the IGM is such that this rare transition still occurs in enough abundance as to be useful

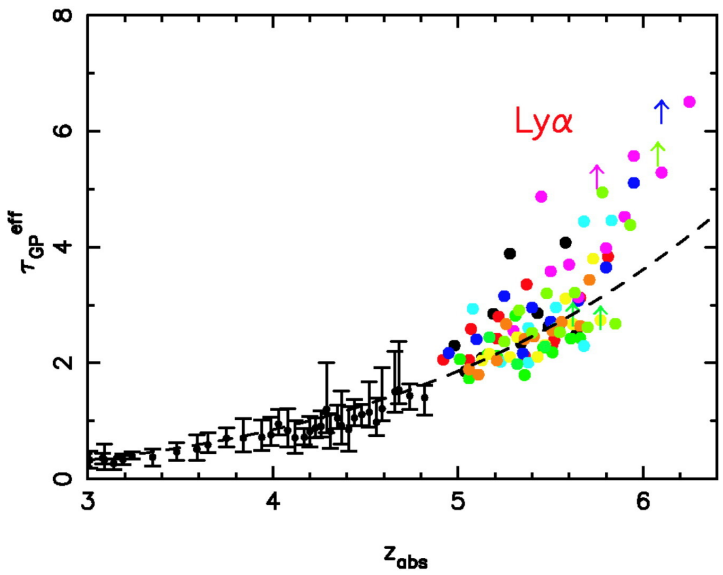

Figure 1.2: Measurements of the Lyman-α optical depth τ_α from the Lyman-α forests of quasars at $3 < z < 6$. The dashed line is a best-fit for the power law dependence of the $(z+1)^{3/2}$ term in Equation 1.5. At $z \sim 5.7$, τ_α notably diverges from this power law relationship. At $z \leq 5.7$, the neutral fraction becomes small enough (on the order of 10^-4) that τ_α is able to be dominated by the contribution of the $(z+1)^{3/2}$ term. This indicates that after this redshift the IGM is almost entirely ionized (Fan et al., 2006). More recent work pushes this lower limit closer to $z \leq 5.5$ (Becker et al., 2015).

for observation. The low likelihood of 21cm emission also lends itself to a low likelihood of reabsorption, meaning the IGM is transparent to the 21cm signal regardless of $x_{\rm HI}$, unlike the Lyman-α line. Because of this, the 21cm signal can tell us more than just when reionization ended: if detected, the redshifted 21cm line will map neutral hydrogen in both time and space, giving a detailed history of the IGM's evolution.

While the 21cm signal is our best bet for understanding the timeline of reionization, observing it doesn't come without difficulty. The 21cm signal is weak due to the forbidden nature of the spin flip transition, resulting in a low signal-to-noise ratio. The observing bandwidth is contaminated with bright astrophysical foregrounds, mainly galactic synchrotron radiation but also free–free emission, bright radio point sources, and unresolved point sources (Liu and Shaw, 2020). Additionally, observing the 21cm signal at any kind of meaningful angular resolution requires an interferometer, which comes with complex instrumental effects that compound the issues of noise and foregrounds. As a result, the

EoR remains poorly constrained: when and for how long the phase change occurred, the main sources of ionizing radiation, stellar properties and abundances, and the physics of the IGM are all relatively unknown factors of the era.

Ongoing and future low-frequency radio telescope experiments, such as the Low Frequency Array (LOFAR, Haarlem et al. 2013), the Murchison Widefield Array (MWA, Tingay et al. 2013), the Hydrogen Epoch of Reionization Array (HERA, DeBoer et al. 2017), and the Square Kilometer Array (SKA, Braun et al. 2015, Koopmans et al. 2015), promise to improve signal constraints and eventually provide a direct detection of the EoR.

However, actually extracting parameter constraints from observations remains a difficult task. Reionization involves a number of highly complex and interlinked processes, and as a result existing analytic solutions fail to capture the full scope of the physics involved. Additionally, deriving a set of parameters that describes reionization in a meaningful way is challenging. The rest of this chapter explores the challenges involved in both modeling reionization as well as fitting observations to the parameters of a given model.

1.2 EoR Parameter Estimation

To gain a basic understanding of the reionization process, as well as to highlight the complexities involved in modeling reionization, we begin building a simple analytic prescription for the ionization history of the IGM by counting photons. The rate at which the global ionization fraction $\bar{x}_i = 1 - \bar{x}_{\rm HI}$ changes is equal to the rate of ionizations $\dot{Q}_i$ minus the rate of recombinations $\dot{Q}_r$:

$$\frac{\mathrm{d}\bar{x}_i}{\mathrm{d}t} = \dot{Q}_i - \dot{Q}_r \tag{1.6}$$

Assume a fixed ionizing efficiency ζ for all galaxies such that

$$\dot{Q}_i = \zeta \frac{\mathrm{d}f_{\mathrm{coll}}}{\mathrm{d}t} \tag{1.7}$$

where f_{coll} is the fraction of baryons in collapsed halos with masses greater than some minimum mass m_{min}. Here, ζ represents the number of ionizing photons produced per baryon by a galaxy that go on to impact the IGM. For this simple model it is sufficient to define it as a constant:

$$\zeta = A_{\mathrm{He}} f_* f_{\mathrm{esc}} N_{\mathrm{ion}} \tag{1.8}$$

Here, f_* is the fraction of baryons in the galactic halo, f_{esc} is the escape fraction of ionizing photons, and N_{ion} is the number of ionizing photons produced per baryon ($A_{\mathrm{He}} = 1.22$ is a correction factor that accounts for ionizations lost to the first ionization of He I). If we take the fiducial values of $f_* \approx 8\%$, $f_{\mathrm{esc}} \approx 10\%$, and $N_{\mathrm{ion}} \approx 4,000$ as truth, ζ is ≈ 40 (Loeb and Furlanetto, 2013). However, these values are highly degenerate and uncertain for Population III stars, making ζ a fuzzy parameterization at best. While our model can be improved by modifying ζ to be a function of halo mass and redshift, a single parameter is ultimately unable to fully capture the intricacies involved in the radiative transfer and feedback processes between ionizing sources and the IGM.

Moving on to the second term of Equation 1.6, we can define the rate of recombinations as the product of the recombination rate $\alpha(T)$, the average electron density in ionized regions $n_{\mathrm{e}}(z)$, the global ionization fraction $\bar{x}_i(z)$, and a factor that accounts for variations in the density of the IGM's plasma $C(z, \bar{x}_i)$:

$$\begin{aligned} \dot{Q}_r &= \alpha(T) \langle n_{\mathrm{H}}(z) n_{\mathrm{e}}(z) \rangle \bar{x}_i(z) \\ &= \alpha(T) C(z, \bar{x}_i) n_{\mathrm{e}}(z) \bar{x}_i(z) \end{aligned} \tag{1.9}$$

This "clumping factor" $C(z, \bar{x}_i)$ is especially hard to estimate, and is highly dependent on

how the IGM ionizes. This leaves us with yet another fuzzy and uncertain parameter in our model.

We can now rewrite Equation 1.6 to get a very approximate "reionization equation":

$$\frac{\mathrm{d}\bar{x}_i}{\mathrm{d}t} = \zeta \frac{\mathrm{d}f_{\mathrm{coll}}}{\mathrm{d}t} - \alpha(T) C(z, \bar{x}_i) n_{\mathrm{e}}(z) \bar{x}_i(z) \tag{1.10}$$

As we have shown, it is relatively easy to define a basic model for the ionization history of the IGM. However, the processes involved in reionization are in reality extremely messy and interlinked, and thus cannot be accurately represented by such a simple model. Indeed, no existing analytic model exists to capture all the complexity of the physics involved in reionization, and therefore analytic models are not preferred for extracting parameter values from observation. Therefore, we use simulations to estimate EoR parameters from observations. Two ways of doing so are discussed below.

1.2.1 Bayesian Approaches

The first approach to using simulations to estimate EoR parameters from observations is to use probability distribution sampling algorithms to calculate posteriors for observations of parameters defined by a given simulator's input space. Outputs are compared to observations via some similarity metric, and posteriors are calculated accordingly. The current state-of-the-art method for this type of 21cm parameter estimation is `21CMMC`, a Markov Chain Monte Carlo tool designed to search the parameter space of `21cmFAST`, its corresponding model of reionization (Greig and Mesinger, 2015). A Markov Chain Monte Carlo method samples a probability distribution by performing a semi-random walk over the parameter space, and is used to calculate posteriors for distributions too high-dimensional or complex to be studied analytically.

This approach has the advantage of being based on well-understood Bayesian statistics and therefore inherently including uncertainty information in the analysis. However,

`21CMMC` has its drawbacks. `21CMMC` is limited to searching the input space of `21cmFAST`, the parameters of which are approximations to more complicated physics and therefore require significant interpretation when applied to data from other simulators. `21CMMC` also seems to suffer with simulator-dependence and struggles to accurately infer parameters from different data distributions. Berklas and Pober (2025) finds that `21CMMC` fails to accurately infer the parameters of different versions of `21cmFAST`, a fact that does not bode well for future applications in analyzing observational data, which is by necessity out-of-distribution (OOD), a term which we use here to mean any data drawn from a distribution to which the ground truth is inaccessible to our parameter estimator. Finally `21CMMC` uses the power spectrum as a summary statistic, which does not contain all information about the ionization field due to non-Gaussianities in the signal and therefore may be sacrificing valuable information (Majumdar et al. 2018, Shimabukuro, Yoshiura, et al. 2016). These problems have spurred interest in studying other methods of EoR parameter estimation.

1.2.2 Machine Learning Approaches

An alternative to `21CMMC` is to train Artificial Intelligence (AI) models (specifically neural networks) on simulated EoR data[1] for the eventual purpose of predicting on observational data.

These AI models are known for their ability to regress on complex, noisy data such as that of reionization. Neural networks are trained via gradient descent on a training data set to identify patterns in the distribution, updating the model's internal weights at each step. In contrast to `21CMMC`, AI models are flexible in that they can efficiently regress on many data formats, such as the power spectrum as well as raw image data. AI models also have the potential to synthesize information from many different sources of data, removing the inherent dependency on any one model of reionization. For these reasons,

[1] In this work we use the terms Artificial Intelligence (AI) and Machine Learning (ML) interchangeably.

interest in AI-based methods of EoR parameter estimation has increased in recent years.

There is extensive work investigating AI's capacity for image-based inference of EoR parameters, which can be roughly divided into one of two categories: experiments in which the initial matter density fields are varied, but the predicted parameters are simulator-specific and therefore do not map easily to data generated from other simulators (Gillet et al., 2019; Kwon et al., 2020; Hassan et al., 2020; Prelogović et al., 2021; Neutsch et al., 2022; Zhao et al., 2022), or experiments where the predicted parameters are model independent, but the data is generated from a single initial density field (La Plante and Ntampaka, 2019; Billings et al., 2021; Zhou and La Plante, 2022). The former makes it difficult to determine a model's ability to generalize across distributions while the latter can cause the model to overfit to specific regions or spatial features of the data (Sooknunan et al., 2024). There has also been significant research into AI power-spectrum based parameter estimation (Shimabukuro and Semelin, 2017; Doussot et al., 2019; Choudhury et al., 2022; Tiwari et al., 2022; Shimabukuro, Mao, et al., 2022). This would essentially serve as an alternative to `21CMMC`, being more computationally efficient and possibly more accurate while sacrificing the uncertainty information inherent to the Bayesian framework. Finally, AI is being integrated into EoR data analysis pipelines. In seeking to reduce the bias of LOFAR's Gaussian Process Regression (GPR) method of signal separation, Mertens et al. (2023) replaces their covariance prior model with a learned autoencoder, coining the term "ML-GPR" for their method. This pipeline has already been incorporated in analysis of LOFAR data (Mertens, F. G. et al., 2025; Munshi et al., 2025).

However, AI methods are not without their pitfalls, pitfalls that must be addressed in parallel with their increased presence in the literature. Predominantly among these pitfalls are the issues of generalizability and interpretability. Generalizability refers to a model's ability to accurately infer parameters from data drawn from an unseen distribution. Because all simulators of reionization used for parameter inference have some degree of approximation, observational 21cm data of the EoR is inherently OOD relative to

the training base (see Chapter 1.3 for a more detailed discussion). As such, the ability for an AI model to generalize to OOD data is critical for its eventual application to real-world data. However, Zhou and La Plante (2022) find that AI models trained on one simulator fail to generalize to another, a foreboding sign for their ability to generalize to observational data in the future. This problem of generalizability is shared with Bayesian methods of parameter analysis (Berklas and Pober, 2025) but is still one that must be addressed prior to application to ensure accurate parameter recovery. Chapter 2 proposes one method of improving OOD performance in models trained on simulations of the EoR.

Another difficulty in AI parameter estimation lies in interpreting the model's decisions. AI models are "black boxes" in the sense that they do not easily reveal the reasons for their decisions. Which factors are considered in the decision making process, and to what magnitude they are weighted, are obscured by the model's deep, nonlinear structure. Additionally, uncertainty information is not readily available as it is for Bayesian parameter estimation methods. This results in an AI model that returns a prediction for a given parameter, but without any explanation as to how that prediction was determined or how certain it is. This is, obviously, antithetical to the scientific process and must be rectified, especially given existing concerns over AI models' ability to generalize to unseen distributions. Therefore, methods must be developed to reveal which aspects of our data factor most into our parameter estimates and why. Chapter 3 explores the shortcomings of existing map-based feature attribution methods for images when applied to EoR data, and suggests an alternative that takes image-wide features into account in the form of counterfactual explanations.

1.3 Simulating Reionization

Because simulations are required for EoR parameter inference, it is worth discussing the various simulators that exist and the differences between them.

1.3.1 Numeric and Semi-Numeric Simulations

Simulators of the EoR can be divided into two loose categories: "numeric" and "semi-numeric". Broadly speaking, numeric simulators make fewer assumptions and take into account more complex physics (such as radiative transfer and hydrodynamic effects) than semi-numeric simulators. In contrast, semi-numeric simulators make smart assumptions to reduce computational cost, allowing more data at larger simulation volumes to be produced. Because parameter inference requires large amounts of high-volume data (simulation box lengths must be ≥ 300 Mpc for large-scale modes to converge; see Iliev et al., 2014; Kaur et al., 2020), semi-numeric simulators are preferred for this type of analysis.

All semi-numeric simulators of reionization will have some inherent non-negligible error, due to the assumptions and approximations characteristic of each individual simulator. Because of this, semi-numeric simulators are not perfectly reflective of the physics of the real universe. This means that observations of the 21cm signal will inherently be OOD to the space of all possible simulations for a given semi-numeric simulator, a fact that highlights the importance of EoR parameter estimation methods that are capable of extrapolating to similar but unseen distributions of data.

1.3.2 Simulator Examples

Here we discuss two relevant examples of semi-numerical simulators: `21cmFAST` and `zreion`. These are by no means the only two seminumeric simulators available; a more comprehensive list of existing reionization simulators, both numeric and semi-numeric, can be found in Gnedin and Madau (2022).

21cmFAST

`21cmFAST` (Mesinger et al., 2011; Greig and Mesinger, 2015; Murray et al., 2020) is a semi-numerical method for simulating the ionization of the IGM during the EoR. It uses Lagrangian perturbation theory to evolve the linear density field over time, then estimates

the ionization field directly using an approach based on the excursion set formalism which compares the number of ionizing photons to the number of baryons over a spherical region of decreasing radius R. At a given redshift z, a region with radius R centered at position $\mathbf{x}$ is considered ionized when the product of the ionization efficiency ζ (defined in Equation 1.8) and the collapse fraction $f_{\rm col}$ is greater than 1:

$$\zeta f_{\rm col}(\mathbf{x}, z, R, \bar{M}_{\rm min}) \geq 1 \tag{1.11}$$

Here, we consider ζ a constant, and $f_{\rm col}$ is the fraction of collapsed matter residing in halos larger than $\bar{M}_{\rm min}$. Partial ionization is simulated by setting the ionization fraction to $\zeta f_{\rm col}(\mathbf{x}, z, R, \bar{M}_{\rm min})$ when R equals the cell size.

The condition presented in Equation 1.11 can be applied to identify ionized bubbles using two different algorithms: central-voxel flagging and full-sphere flagging. Central-voxel flagging is the default behavior, in which only the voxel at point $(\mathbf{x}, z)$ is considered ionized when the condition in Equation 1.11 is met. Full-sphere flagging, in contrast, marks every point within R as ionized. There is conflicting evidence as to which flagging algorithm is more reflective of real-world physics, though both are physically plausible (Mesinger et al., 2011; Hutter, 2018). These algorithms produce results dissimilar enough to be considered two distinct simulators, a fact we take advantage of in Section 2.1.1.

`21cmFAST` can be further extended by assuming ζ to be a function of the halo mass $M_{\rm h}$. Equation 1.11 assumes ζ to be constant, independent of halo mass $M_{\rm h}$. In a new parameterization introduced by Park et al. (2019), ζ is allowed to vary with the halo mass by redefining f_* and $f_{\rm esc}$ to have a power-law dependence on $M_{\rm h}$:

$$f_* = f_{*,\,10}\left(\frac{M_{\rm h}}{10^{10} M_\odot}\right)^{\alpha_*} \qquad\qquad f_{\rm esc} = f_{{\rm esc},\,10}\left(\frac{M_{\rm h}}{10^{10} M_\odot}\right)^{\alpha_{\rm esc}} \tag{1.12}$$

Here $f_{*,\,10}$ and $f_{{\rm esc},\,10}$ are normalization constants while α_* and $\alpha_{\rm esc}$ are power-law scaling factors. This mass-dependent ζ parameterization approximates how the star formation

rate and the emission of ionizing photons scales with halo mass rather than assumes those values to be constant, adding an additional factor of realism to the model.

The mass-dependent and mass-independent ζ formulations of `21cmFAST` produce significantly different reionization histories, and can thus be treated as two unique "simulators" of reionization. We use both parameterizations to produce datasets representing unique underlying distributions in Section 2.1.1.

zreion

`zreion`[2] is an improved implementation of Reionization at Large Scales (RLS), a method presented in Battaglia et al. (2013) developed as a lightweight way to simulate reionization for very large box sizes. `zreion` assumes the redshift of reionization field δ_z is a linear tracer of the matter overdensity field δ_m at large scales. The model is defined by the bias parameter $b_{zm}^2(k)$ which relates the two fields in Fourier space:

$$b_{zm}^2(k) = \frac{P_{zz}(k)}{P_{mm}(k)} = \frac{b_0^2}{\left(1 + \frac{k}{k_0}\right)^{2\alpha}} \tag{1.13}$$

where P_{zz} and P_{mm} are the power spectra of δ_z and δ_m respectively. Setting $b_0 = \frac{1}{\delta_c} = 0.593$ (δ_c being the critical overdensity in the spherical collapse model) leaves `zreion` three free parameters with which to model reionization: the wavenumber scaling factor k_0, the power law index α, and the mean redshift of reionization $\bar{z}$. Unlike `21cmFAST`, `zreion` does not simulate partial cell ionization, instead expressing the neutral fraction $x_{\rm HI}$ as a binary 0 or 1.

Because `zreion` treats the ionization field as a linear function of the density field, it fails to capture aspects of reionization other simulators can, such as the characteristic growth of ionized bubbles around individual radiation sources. However, these differences become less important on large scales (Battaglia et al. 2013). Though `zreion` is less

[2] `https://github.com/plaplant/zreion`

physically motivated in its parameterization than `21cmFAST`, its usefulness to this work lies in the fact that it represents a significantly different set of possible reionization histories than those which can be generated with `21cmFAST`, a fact that will be discussed in greater detail in Chapter 2.

1.3.3 Which Simulator is Best for Parameter Estimation?

Methods of parameter estimation are only as good as the data they're built on. This raises a natural question: which simulator is "best" for parameter estimation? The answer is this: there is no definitive "best" simulator, only more and less approximate. Semi-numeric simulations are used for parameter analysis because numeric simulations are not tractable at high simulation volumes and dataset sizes, but each semi-numeric simulator makes their own approximations that leave them inaccurate in unique ways. Oftentimes it is unclear which approximations are more physically appropriate; one example of this is the difference between the central-voxel and full-sphere flagging algorithms described in Section 1.3.2, where no consensus exists on which method returns more realistic ionization histories. There is, of course, a spectrum of authenticity, with some simulators being more physically motivated than others (for example `21cmFAST` and all its variants are generally considered "more realistic" than `zreion`). However, as we will discuss in Chapter 2, the fact that these simulators make different approximations (and are therefore inaccurate in different ways) may render them useful still in "averaging-out" the errors of other more physically motivated simulators, leaving only the useful commonalities behind for our parameter estimation methods to incorporate into their analysis.

CHAPTER 2

Simulation Diversity in AI Parameter Estimation of the EoR

Chapter 1 raises the question: how do we ensure AI models trained on imperfect simulations can make accurate predictions about real-life 21cm observational data? This chapter proposes a solution that relies on increasing the diversity of the simulation space by including more simulators in the data generation process.

A classic strategy in AI research involves increasing the diversity of the training data to avoid overfitting to spurious features. By increasing variation within the training data, incorrect or irrelevant information is "averaged-out", leaving only useful commonalities behind. Because all semi-numeric simulators make different approximations, each is flawed in unique ways. Incorporating multiple simulators into our analysis, therefore, may reduce simulator-specific bias and improve accuracy when predicting on data from unseen distributions, such as real-world observation data. This chapter investigates that claim.

2.1 Method

In recent years, AI research has trended towards a data-centric (as opposed to model-centric) approach, emphasizing the curation of high-quality training data over the development of better model designs. This shift comes as a result of a growing body of research showing that the architecture of the model matters much less than the quantity and quality of the data (Zha et al., 2025). Following this approach, this work therefore investigates how the data used for parameter inference affects the accuracy of the inferred parameters.

Increasing training data diversity allows AI models to focus on commonalities while discarding spurious and conflicting features of individual simulators, thereby creating a more informed and cohesive picture of the distribution. Therefore, we propose that the greater the number of EoR simulation algorithms included during analysis, the more universal properties of the set can be inferred by our parameter inference method of choice. This concept of increasing dataset diversity to improve generalizability motivates our approach.

To study how diversity affects OOD performance, we train models on data from one, two, and three simulators of reionization before predicting on data from a held-out simulator. In this work, we use the term "out-of-distribution" to refer to data generated from any simulator not represented in the training data. Following this, we compare the performance of the single, double, and triple simulator models to see if the OOD performance has improved. If generalizability to an unknown distribution does improve with increased data diversity, that implies AI parameter inference of real-world EoR data would benefit from a large and varied suite of training data, drawn from many different simulators of reionization.

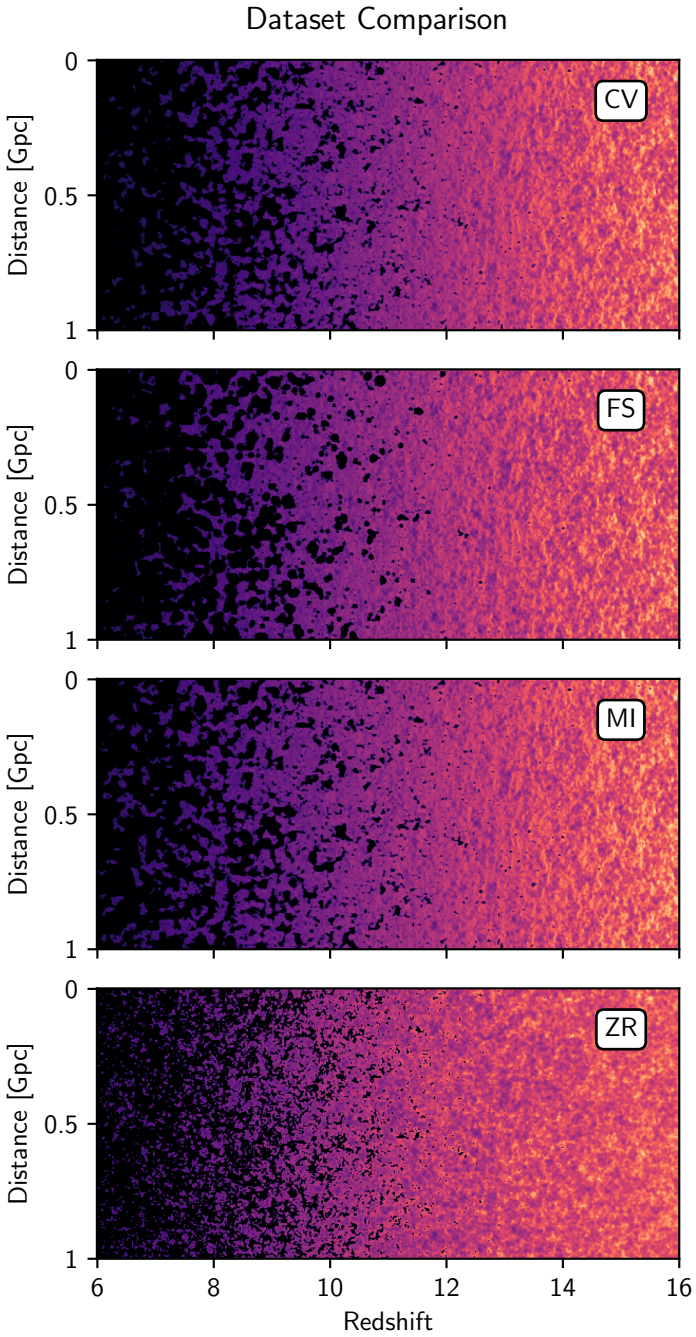

Figure 2.1: A comparison of the different algorithms used in this work to simulate the evolution of the EoR brightness temperature field. For this comparison, each EoR history uses the same underlying density field, though in the actual datasets no two instances share the same initial density field. For this figure, input parameters were adjusted in order to produce instances with similar reionization histories while still showcasing the qualitative differences between simulators. For example, note how Dataset `CV` results in smaller more jagged bubbles while Dataset `FS` produces comparatively larger and rounder bubbles, reflecting their respective bubble-flagging methods.

Dataset Name	Abbrev.	Source Simulator	Mass-dependent ζ	Bubble Finding Algorithm	Box Length (Mpc)
"Central-Voxel"	`CV`	`21cmFAST`	Yes	Central-voxel	1000
"Full-Sphere"	`FS`	`21cmFAST`	Yes	Full-sphere	1000
"Mass-Independent Zeta"	`MI`	`21cmFAST`	No	Central-voxel	2000
"Zreion"	`ZR`	`zreion`	-	-	1000

Table 2.1: A summary of the differences between Datasets `CV`, `FS`, `MI`, and `ZR`. A more detailed list of generation parameters can be found in Appendix A.1.

2.1.1 Data

In order to test how a model performs on an unseen distribution, four sets of data were generated, each of which used different approximations to calculate the evolving $x_{\rm HI}$ field. Two different EoR simulation packages were used to generate data: `21cmFAST` and `zreion`. `21cmFAST` was used to generate Datasets `CV` ("Central-Voxel"), `FS` ("Full-Sphere"), and `MI` ("Mass-Independent Zeta"). This naming scheme is based on the different versions of `21cmFAST` used to generate each: Datasets `CV` and `FS` use the mass-dependent ζ parameterization, but use the central-voxel and full-sphere bubble flagging methods, respectively, while Dataset `MI` uses the mass-independent ζ parameterization[1]; more details about the differences between these versions of `21cmFAST` can be found in Section 1.3.2. Dataset `ZR` ("Zreion"), meanwhile, was generated using the more approximate semi-numerical simulator `zreion` (details can be found in Section 1.3.2). Table 2.1 summarizes the differences between the simulation algorithms used to generate each of our four datasets.

Labels

In this work, we choose to predict on the duration of reionization $\Delta z = z_{25} - z_{75}$, where z_{25} and z_{75} are the redshifts at which 25% and 75% of the neutral hydrogen is ionized

[1]Like Dataset `CV`, Dataset `MI` also uses central-voxel bubble flagging, but this was omitted from its name to avoid a cumbersomely long abbreviation.

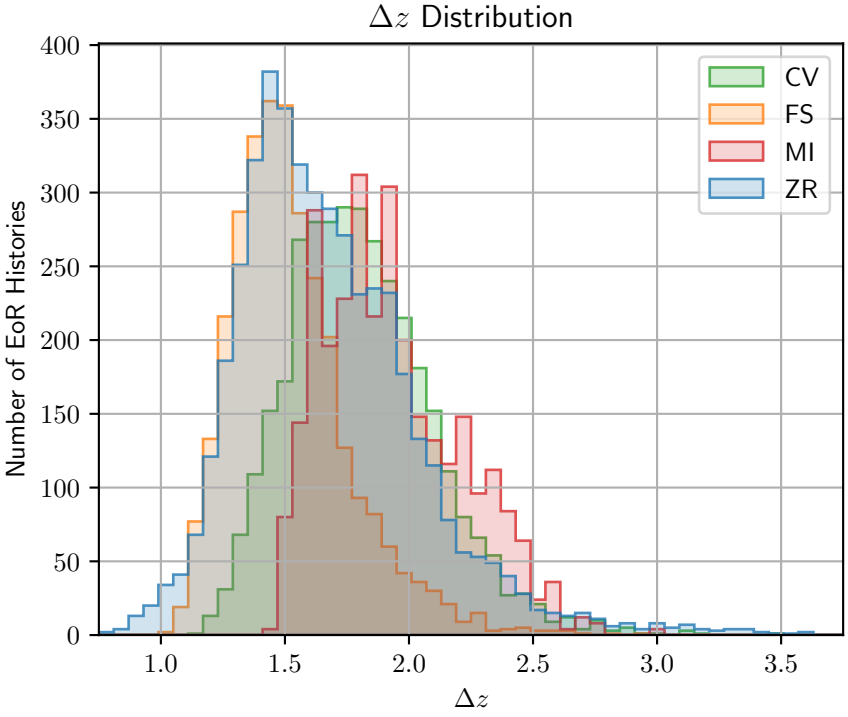

Figure 2.2: Distribution of Δz across all four datasets. It is important that these distributions overlap, as AI models traditionally struggle to predict on label ranges outside that of their training data (Sooknunan et al., 2024).

respectively. This choice was motivated by the need for a prediction parameter with a concrete physical interpretation. Since no analytic model fully describes reionization, any set of abstract input parameters will inherently be nonphysical. In contrast, Δz is a value with real physical meaning, which can be determined from the neutral fraction history of any reionization simulation. In addition, Δz also has the advantage of being simulator independent (see discussion of simulator dependent vs. simulator independent parameters in Section 1.2.2). How Δz relates to other reionization parameters, such as the Cosmic Microwave Background (CMB) optical depth τ, is a question of ongoing research. Nevertheless, Δz remains an important quantity in and of itself, especially given the limited existing observational constraints on the EoR.

For Datasets `CV`, `FS`, and `MI`, Δz was calculated from the global neutral fraction, a value automatically tracked by `21cmFAST`. For Dataset `ZR`, the neutral fraction history was calculated directly from the ionization field.

Figure 2.2 shows a histogram of the Δz values for all four datasets. During data generation, the range of allowed input parameters was adjusted to ensure the Δz distributions overlapped (see Appendix A.1 for a detailed breakdown of the range of dataset generation parameters). Overlapping output ranges between the training and testing datasets has been found to play an important role in AI model performance (Sooknunan et al., 2024). While the distributions in Figure 2.2 are not identical, there exists a region where all four datasets overlap such that our models are rarely predicting a Δz value outside the range seen during training.

Data Generation and Preprocessing

Histories were generated for Datasets `CV`, `FS`, and `MI` using the semi-numerical simulation package `21cmFAST`, with parameters randomly sampled over the ranges listed in Appendix A.1.

Initially, 4399 histories were generated for Dataset `CV`, 3981 histories were generated

for Dataset `FS`, and 1031 histories were generated for Dataset `MI`. Datasets `CV` and `FS` were generated using a simulation box length of 1000 Mpc and a plane-of-sky resolution of 256×256, resulting in a cell size of 3.9 Mpc. Dataset `MI` was generated using a simulation box length of 2000 Mpc and a plane-of-sky resolution of 512×512, resulting in the same cell size of 3.9 Mpc. Reionization histories for all three datasets were evolved over the redshift range $[6.0, 16.0]$.

We used `21cmFAST`'s inbuilt interpolator to knit our evolving brightness temperature fields into brightness temperature lightcones. Here, the term "lightcone" refers to a simulated reionization scenario in which the xy-plane represents the plane of the sky, and the z-axis represents evolving redshift. These lightcones were then trimmed to span the redshift range [6.0, 14.7], resulting in a z-axis length of 512.[2]

To ensure each lightcone had a defined value for Δz, any generated histories where $z_{25} \geq 16.0$ or $z_{75} \leq 6.0$ were discarded at the time of generation. This resulted in the final dataset sizes of 3445 lightcones for Dataset `CV`, 3062 lightcones for Dataset `FS`, and 740 lightcones for Dataset `MI`. This corresponds to a roughly 25% loss in data.

The discrepancy in simulation volume between Datasets `MI` and Datasets `CV` and `FS` is due to the fact that Dataset `MI` was generated first; the box length was halved for subsequent datasets because the simulation volume was determined to require too much computational time to be feasible for all four datasets. Because the cell size was consistently 3.9 Mpc across all datasets, Dataset `MI` lightcones are essentially twice as large as those from other datasets along the x and y axes. Therefore, to make all data cubes have the same shape, Dataset `MI` lightcones were quartered on the xy-plane, resulting in a final dataset size of 2960 reionization histories for Dataset `MI`. While it is possible the increased simulation volume impacted the resulting brightness temperature field due to the availability of more spatial modes, we do not consider this to be a problem, as this

[2] 14.7 was chosen as the maximum redshift simply because it corresponds to the redshift slice at index 512 along the z-axis, and is otherwise completely arbitrary.

would only serve to diversify our four datasets further.

Dataset `ZR` is based on a different semi-numerical simulation package than the other datasets, and was therefore generated using a different process. Because `zreion` does not include a way to generate evolving cosmological density fields, the density fields for Dataset `ZR` were created and perturbed over time using `21cmFAST`'s inbuilt method. 4538 perturbed density fields were generated in this way, using a simulation box length of 1000 Mpc and a plane-of-sky resolution of 256×256, resulting in a cell size of 3.9 Mpc, identical to that of our three other datasets. These perturbed density fields were then interpolated to form density lightcones spanning the redshift range [6.0, 14.7], identically to how lightcones were generated for the previous three datasets. These density lightcones were then used as inputs to the `zreion` simulator, with parameters randomly sampled over the ranges and priors listed in Appendix A.1. Unique density fields were generated for each reionization history across all datasets (see Section 1.2.2 for a discussion on the importance of unique density fields).

Building on previously published approaches (La Plante and Ntampaka, 2019; Zhou and La Plante, 2022), each lightcone was divided into 30 256×256 mean-subtracted slices of the brightness temperature field, sampled evenly along the full redshift range of [6.0, 14.7]. We refer to these redshift-sliced lightcones as "data cubes."

Besides mean subtraction, no foregrounds, sources of noise, or instrumental effects were modeled in any of the data. Because of this, one might naively believe that Δz can be trivially calculated by assuming all cells where the differential brightness temperature field $\delta T_{\rm b}(\mathbf{x}, z) = 0$ are ionized, giving us an estimate of $x_{\rm HI}(z)$. This is a decent assumption for `zreion`, but it does not hold for `21cmFAST` data due to the fact that `21cmFAST` simulates partial cell ionization as described in Section 1.3.2. In Section 2.2, we will show that our model struggles to generalize to data from unseen simulators even without foregrounds or noise modeled in the data. Therefore, we treat this work as an exploratory proof-of-concept for a method to reduce simulator specific bias in AI models by increasing training data

diversity, leaving explorations of how sources of information loss affect model performance on out-of-distribution data for future works.

Prior to training, data cubes were normalized to the interval $[0, 1]$ for ease of regression. Each dataset was divided into training, validation, and testing subsets, with 2400, 280, and 280 data cubes respectively. This size was chosen because our smallest dataset, Dataset `MI`, includes $2400 + 280 + 280 = 2960$ data cubes; in this way, we control for training dataset size.

Because every reionization history was initialized with a unique random seed, there is no cross-contamination of density field information across different splits of the data. This is essential to prevent the AI model from learning to look for specific regions of structure in the data (Sooknunan et al., 2024). This is even the case with the subdivided Dataset `MI` lightcones, as the four quadrants of the xy plane across which each lightcone was split were not correlated in any way.

For models trained on multiple simulators, the size of the training dataset was limited to 2400 cubes by splitting the pool of training data proportionally between the represented simulators, i.e., models trained on data from two simulators had training datasets composed of 1200 data cubes from each, and models trained on data from three simulators had training datasets composed of 800 data cubes from each. This choice was made to control for the size of the training dataset, which is known to impact model performance. This dataset splitting was not done for the test and validation sets, which were kept at 280 data cubes per simulator. This was done so that the performance of each model on the test data from each simulator could be equivalently compared.

k-Fold Validation

k-fold cross-validation was performed over 8 folds to ensure the results are not dependent on different train / validation / test partitions of the dataset, via the following procedure:

1. Each simulator dataset of size 2960 were split into 10 groups of size 280, leaving 160 data cubes ungrouped. These ungrouped data cubes were considered "remainders" and were never folded into the test / validation groups. This was done to preserve the train / validation / test split of 2400 / 280 / 280; having a training dataset of a length divisible by 2 and 3 is necessary to include equal proportions of data from each simulator in the case of models trained on data from two and three simulators.

2. For fold i of 10, group i was selected as the test group and group $(i + 1) \mod 10$ was selected as the validation group. This leaves all other groups (plus the 160 "remainders") available to be used in the training data. For single-simulator models, this data was used as the training group.

3. In the case of models trained on data from two or three simulators, where the training dataset is already split into groups of 1200 and 800 data cubes from each simulator respectively, we also fold along this partition, taking data cubes from each simulator dataset starting at the index for group $(i + 2) \mod 10$. In this way, the training data changes with every fold, even when the test and validation groups do not require it to change (aka, when the test and validation groups are in the proportion of the simulator dataset that would not have been used for training anyway).

We report our results as mean ± standard error across all 8 folds.

2.1.2 Model

Our model is a Convolutional Neural Network (CNN) based on the architecture used in Zhou and La Plante (2022), and is summarized in Figure 2.3 (a more detailed breakdown of our model's architecture can be found in Appendix A.2). It consists of three downsampling 2D convolutional layers, a global pooling layer, and three linear layers. Our model takes data cubes with dimensions $30 \times 256 \times 256$ as inputs. The 30 redshift slices are stacked along the channel dimension such that the convolutional kernel convolves over the xy plane.

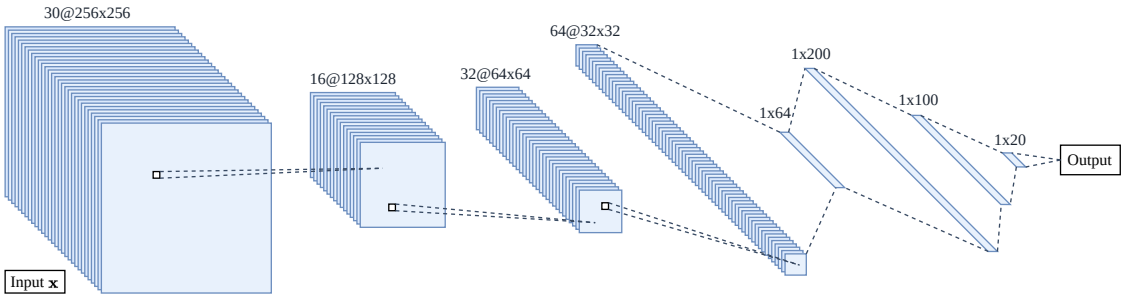

Figure 2.3: A simple diagram of our CNN. Our model consists of three 2D convolutional downsampling layers, a global pooling layer, and three linear layers, with a resulting scalar output. Our model takes data cubes of size $30 \times 256 \times 256$ as inputs to the network. Each cube is made up of a set of 30 brightness temperature slices sampled evenly along the redshift axis, treated as color channels such that the convolutional kernel convolves over the xy plane.

Parameter	Value
Batch Size	64
Stage I Learning Rate	5×10^{-3}
Stage I Epochs	1000
Stage II Learning Rate	1×10^{-3}
Stage II Epochs	2000

Table 2.2: Model learning parameters used during training.

In this way redshift information is treated as analogous to color channel information for a more traditional image-analysis CNN. In some previous research, 3D convolutional layers have shown promise in more effectively integrating information along the redshift axis (Zhao et al., 2022), but we do not explore that here.

Models were trained to convergence using the learning hyperparameters summarized in Table 2.2. The batch size and learning rate were optimized for via a parameter grid search prior to training. Training took place in two stages, with a stepped-down learning rate after Epoch 1000. The loss function was Mean Squared Error (MSE). The same architecture and learning parameters were used across all models trained.

2.2 Results

We find that, when trained on data drawn from only one distribution, our models do not generalize to data from a second unseen distribution. Figure 2.4 shows the predicted vs. true Δz of both in-distribution vs. out-of-distribution test cubes for models trained on Datasets `CV`, `FS`, `MI`, and `ZR`. Here, we choose to primarily report average MSE rather than average percent error, due to the the fact that the Δz distributions of our datasets are all slightly different, which means percent error cannot be used as a relative measure to compare between them. The single-dataset model is able to recover the value of Δz from in-distribution reionization histories (shown in gray) with an average MSE of 0.01. However, when asked to predict the value of Δz for out-of-distribution data cubes (shown in color), the model does so with substantial bias, with an average MSE of 0.32 across all out-of-distribution dataset / model pairs. The one exception to the poor out-of-distribution performance is that models trained on Dataset `CV` seem to perform well on Dataset `FS` and vice versa; a more in-depth discussion of how different simulators generalize to one another can be found in Section 2.3.1.

We emphasize that *none of these test points were seen during training*, as the data was split into training / validation / testing subsets prior to any training, and data from the test subsets were never used to train models; the distinction between an in-distribution and out-of-distribution data cube comes from which simulator was used to generate it, not whether it specifically was included in the training data.

This failure to generalize mirrors the findings of Zhou and La Plante (2022), with more datasets and unique density fields. We demonstrate here that the added complexity and diversity introduced by unique underlying density fields is not sufficient to allow the models to interpolate to unseen distributions, thus motivating our investigation into diversifying our training data with multiple simulators.

The predicted vs. true value of Δz for models trained on two and three datasets is

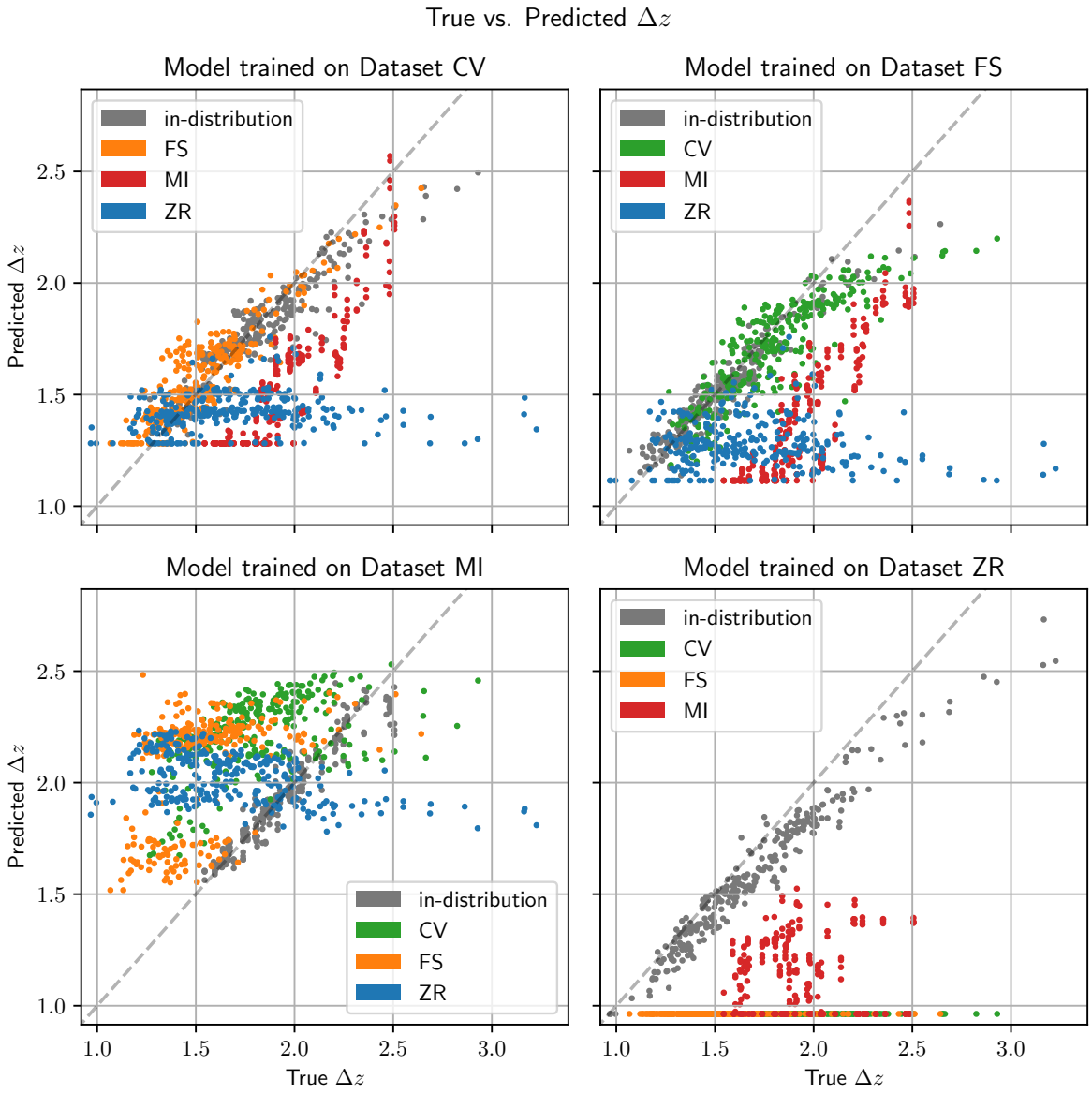

Figure 2.4: Predicted vs. true Δz of single-dataset models across all four datasets. Note the clear difference between in-distribution and out-of-distribution performance. For the models trained on Datasets `CV`, `FS`, and `ZR`, the model will not predict values of Δz below the minimum value seen during training, resulting in a horizontal line of predictions at the minimum. This is likely due to the sparsity of the trained model's latent space, an issue that could be rectified by tweaking the model architecture or further hyperparameter optimization, though doing so is unlikely to result in a significant boost in accuracy.

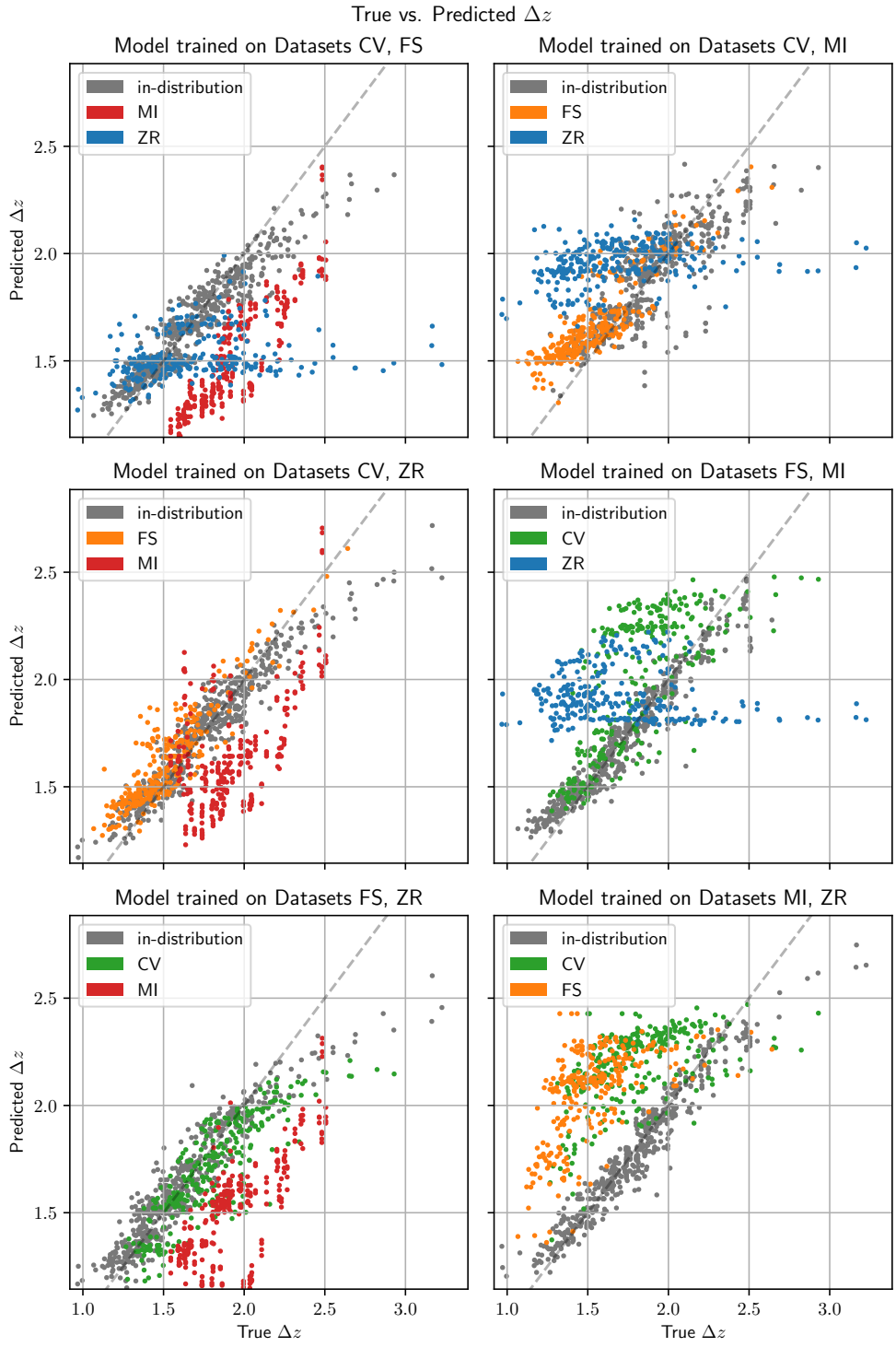

Figure 2.5: Predicted vs. true Δz for models trained on two datasets. Note that unlike the training data, which was split equally between the two simulators used during training, the test datasets plotted here remain at a fixed size, in order to better compare the test performance across models.

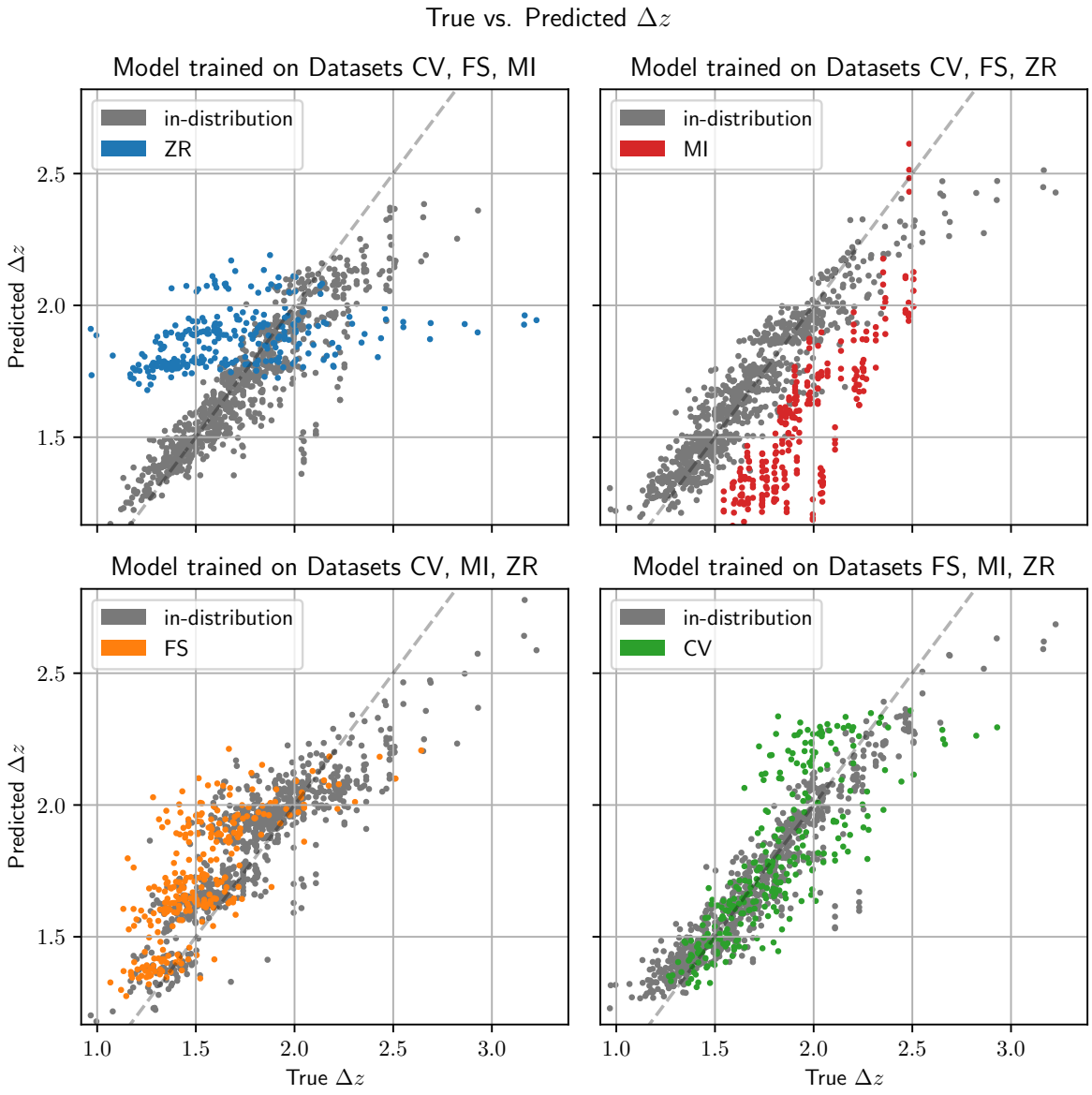

Figure 2.6: Predicted vs. true Δz for models trained on three datasets. Note that unlike the training data, which was split equally between the three simulators used during training, the test datasets plotted here remain at a fixed size, in order to better compare the test performance across models.

plotted in Figures 2.5 and 2.6, respectively. Accuracy for in-distribution datasets (shown in gray) remains high, with an average MSE of 0.02 and 0.03 for the double and triple dataset models, respectively. Similarly, parameter recovery for out-of-distribution data (shown in color) remains poor, with an average MSE of 0.17 and 0.13 across all out-of-distribution dataset / model pairs for the double and triple dataset models, respectively. Compared to the average out-of-distribution MSE of the single dataset models 0.32, adding data from additional simulators to the suite of training data results in a clear improvement in an AI model's ability to generalize. In addition to being quantitatively better as a function of the number of included simulators, the out-of-distribution performance is visually improved between Figures 2.4, 2.5, and 2.6. While parameter recovery never quite reaches the level of accuracy achieved for in-distribution data, a reduction in MSE of $\approx 60\%$ from the single-dataset case to the triple-dataset case is a striking improvement.

Figure 2.7 plots shows the average MSE prediction error for out-of-distribution datasets across all models, $\pm$ the standard error across folds. It is worth noting the impact of including Dataset `ZR`, the most dissimilar from the other datasets and the dataset generated from the most approximate semi-numerical simulator, affects the out-of-distribution performance of the other three datasets. The results here are inconclusive: in some cases including Dataset `ZR` improves performance (the MSE when predicting on Dataset `MI` is 0.28 ± 0.01 for the model trained on `FS`, vs. 0.20 ± 0.01 for the model trained on both `FS` and `ZR`), but in other cases it does not significantly impact performance at all (the MSE when predicting on Dataset `FS` is 0.02 ± 0.00 for both the model trained on `CV` and the model trained on both `CV` and `ZR`). In only one case does the addition of `ZR` actively hurt performance: the MSE when predicting on Dataset `FS` is 0.03 ± 0.00 for the model trained on `CV` and `MI`, vs. 0.06 ± 0.01 for the model trained on `CV`, `MI`, and `ZR`. But to truly compare how adding data from a new distribution impacts out-of-distribution performance, we must consider how adding an additional dataset on average impacts the out-of-distribution performance.

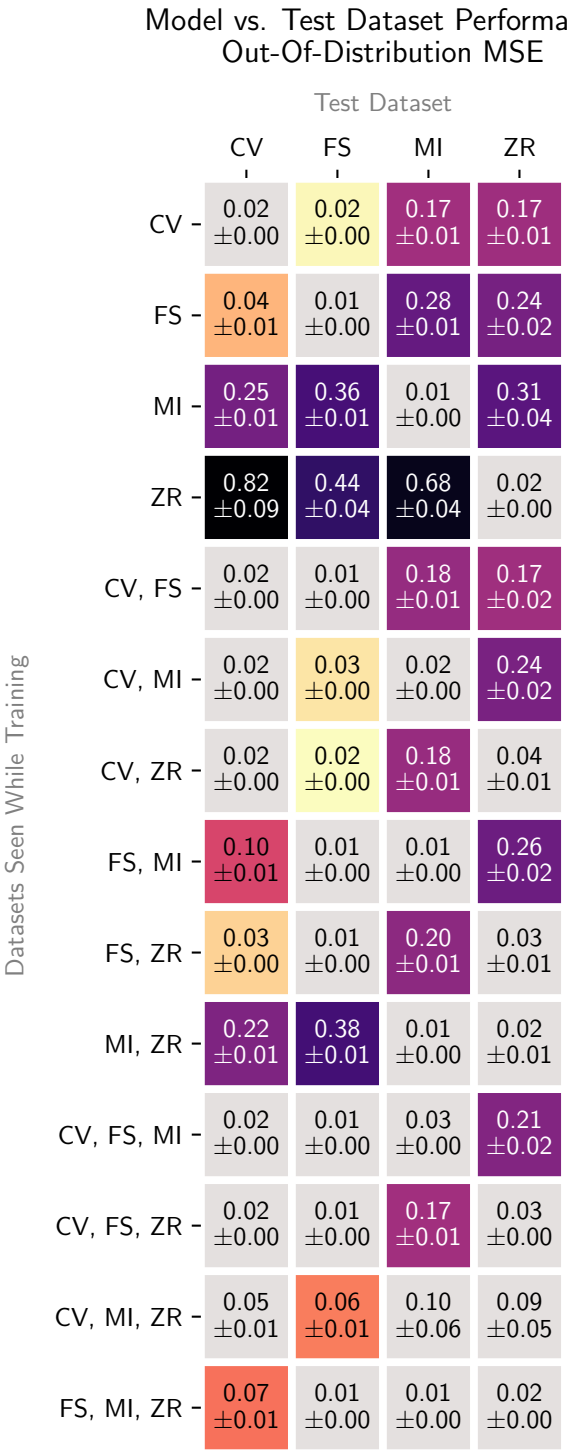

Figure 2.7: Average prediction MSE per test dataset of CNN trained on different combinations of training sets, ± standard error over 8 folds. In this figure, the row represents the simulators represented in the model's training data and the column represents the test dataset. In-distribution model-dataset pairs are shown in gray and out-of-distribution pairs are shown in color. Error values < 0.005 have been rounded to 0.00.

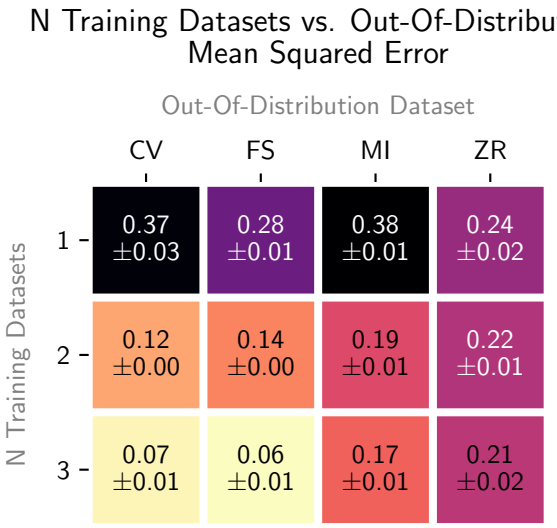

Figure 2.8: Out-of-distribution performance heatmap averaged across number of simulators represented in the training data, ± standard error over 8 folds. This figure groups the data presented in Figure 2.7 by number of simulators represented in the training pool then averages along the columns. Note that the averaging takes place over a very small sample size, as the number of out-of-distribution prediction instances are limited by the fact that we only have four available datasets. Error values < 0.005 have been rounded to 0.00.

To compare the average out-of-distribution MSE for models trained on one, two, and three datasets, we group the columns of Figure 2.7 by the number of datasets included in the training data then average across the out-of-distribution values to obtain Figure 2.8. For example, the value of cell (1, `CV`) in Figure 2.8 is the average of cells ([`FS`], `CV`), ([`MI`], `CV`), and ([`ZR`], `CV`) in Figure 2.7; similarly, cell (3, `CV`) in Figure 2.8 is simply cell ([`FS`, `MI`, `ZR`], `CV`) in Figure 2.7. Therefore, Figure 2.8 shows us the average out-of-distribution prediction error as a function of the number of distributions included in the data. Figure 2.8 shows that for test datasets `CV`, `FS`, and `MI`, the average prediction MSE of unseen datasets is significantly improved when an additional distribution is added to the pool of training data. For Dataset `ZR`, the results are more inconclusive, with the improvements in performance falling outside the range of significance. It also does not, however, appear to significantly worsen performance, which is also important when considering the impact of training models meant for real data on simulations. It is notable that `ZR` is the most

dissimilar of the four simulators, as well as the least physically motivated; this may explain the lack of improvement in its out-of-distribution performance in relation to the other simulator datasets. Overall, these findings imply that increasing the diversity of the training data by including data drawn from multiple different sources will be important when using AI models to analyze real-world data.

2.3 Discussion

2.3.1 Cross-Simulator Performance Transfer

Figure 2.9 estimates how well a model trained on data from one simulator will generalize to data from another unseen simulator by averaging the out-of-distribution prediction MSE by included dataset. In other words, Figure 2.9 was derived in the same way as Figure 2.8, except in 2.9 the columns were averaged by whether a given dataset was seen during training or not. For example, the value of cell (`FS`, `CV`) in Figure 2.9 is the average of cells ([`FS`], `CV`), ([`FS`, `MI`], `CV`), ([`FS`, `ZR`], `CV`), and ([`FS`, `MI`, `ZR`], `CV`), in Figure 2.7. Figure 2.9 does not uncouple the contributions of individual simulators to the model's ability to generalize to data from the unseen simulator; instead, it provides a rough metric of how well the model is able to apply knowledge of one dataset to another in order to perform accurate parameter recovery.

Figure 2.9 shows that including certain datasets in the training pool allows the model to infer the parameters of some unseen datasets better than others. For example, models trained on Dataset `CV` tend to perform better on an out-of-distribution Dataset `FS` (with an average MSE of 0.03 ± 0.00) than on out-of-distribution Datasets `MI` or `ZR` (with an average MSE of 0.17 ± 0.01 and 0.20 ± 0.01, respectively). It is unclear why exactly the model is able to generalize better between Datasets `CV` and `FS` and others, beyond suggesting that the model places less importance on the distribution of bubble sizes in comparison to other features of the datasets, as the only difference between Datasets `CV`

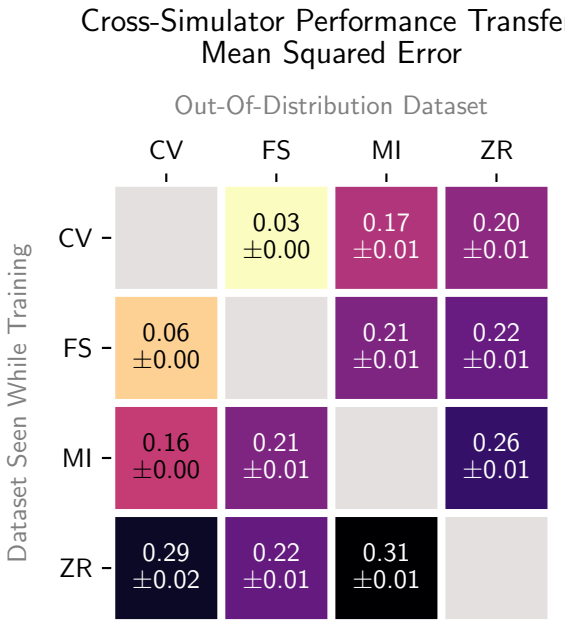

Figure 2.9: Cross-simulator performance transfer. This plots the average MSE performance for models where the row dataset is included in the training data while the column dataset is not, ± standard error over 8 folds. For example, cell (`CV`, `FS`) is the performance on Dataset `FS` averaged across all models trained on `CV` but not `FS` (aka, models trained on `CV`; `CV` and `MI`; `CV` and `ZR`; and `CV`, `MI`, and `ZR`). This metric does not disentangle the effects of multiple distributions being present in the training data, but rather gives a general sense of how transferable the model's knowledge is between two simulators. Error values < 0.005 have been rounded to 0.

and `FS` is in how ionized bubbles are flagged.

Exactly quantifying this cross-simulator performance transfer in a way that would allow one to predict how two simulators might generalize to one another in advance is difficult, and outside the scope of this work.

2.3.2 Quantity vs. Quality

A natural question these results inspire is one of data quantity vs. quality. Figure 2.9 tells us that certain datasets are more valuable than others when predicting on data from an unseen distribution. For example, if we take Dataset `CV` to be our proxy for the real universe, a model trained on Dataset `FS` would give significantly better parameter estimates than one trained on Dataset `ZR`.

However, Figure 2.8 tells us that the diversity of the training dataset is also important. When recovering EoR parameters, we cannot be sure which of our simulators will allow our model to best generalize to real data. We may have a fuzzy spectrum of most to least physically motivated (for example, `21cmFAST` is more physically motivated than `zreion`, and therefore we assume it to be more reflective of the real universe), but for neighbors on that spectrum, which simulator is best suited to allow a model to accurately generalize to 21cm observations is an open question (for example, it is unclear whether central-voxel or full-sphere flagging is better reflective of reionization). Therefore, we can only rely on the out-of-distribution performance averaged as a function of the number of distributions (aka, simulators) present in the training data. With this in mind, Figure 2.8 tells us that, on average, out-of-distribution performance improves with increased training data diversity, thus suggesting it is best to include data from as many sources as possible.

More approximate simulators can be run more quickly to produce greater quantities of data. The diversity argument would favor including this data in our training set, while the quality argument states there's a limit to how useful this inclusion can be corresponding to how closely this simulator overlaps with the true distribution of possible universes. Figure 2.8 seems to imply there is no negative impact, and some potential benefit, to including high volumes of lower-accuracy data, such as that generated by `zreion`, in the training pool.

There is, as always, a balance between quantity and quality to be found. As an extreme example, including "lightcones" composed of random Gaussian noise in the training pool will not contribute any meaningful physical information to the model, and therefore will not improve parameter recovery, despite this technically increasing the diversity of the training pool. This raises an important question: which datasets are "good enough" to be worth including in the training pool?

Because we do not have access to complete information about the ground truth (aka, the real universe), this is a difficult question to answer. Future work may address this

question by studying how accurately models trained on semi-numeric simulators can recover the parameters of data generated from high-fidelity numeric simulators.

Still, with the assumption that our simulators at least capture some useful physical information, our results support the creation of large, multifaceted training datasets made up of data drawn from simulators with a range of numerical complexities over the curation of small, highly accurate datasets sourced from one or two models of reionization. This has implications for future works involving AI-based EoR parameter inference, as well as ongoing data analysis efforts that involve the use of AI, such as LOFAR's introduction of ML-GPR into their analysis pipeline (Mertens et al., 2023; Mertens, F. G. et al., 2025; Munshi et al., 2025).

2.3.3 Future Work

An obvious continuation of this work would be to produce additional datasets from new EoR simulation algorithms to see whether the trend of training set diversity improving out-of-distribution performance in accordance with the trend shown in Figure 2.8 continues. This would be challenging due to the time and resources involved in data generation, as well as due to the lack of software maintenance many older reionization simulation packages suffer from. Generating a small set of reionization histories from highly accurate radiative transfer simulators would be especially useful in characterizing how AI models trained on more approximate semi-numerical simulators respond to more realistic data.

Another route would be to perform the same experiment with modeled foregrounds and noise. It is possible, though somewhat unlikely, that the removal of foreground-contaminated k-modes from the data would cause our AI models to focus less on simulator-specific features, leading to greater out-of-distribution performance. This would, of course, come with some corresponding decrease in overall performance as information is lost from the data. How noise and foregrounds impact model generalization likely hinges on how much simulator-specific information is contained in k-modes we expect to be contaminated

by foregrounds.

Another followup involves precisely defining and quantifying some metric of "simulator similarity" that would allow one to accurately predict how training a model on data from one simulator will affect performance on data from an unseen second simulator. Figure 2.9 gives a rough metric of which datasets the model sees as comparable, but does nothing to predict how training an AI on any given distribution would impact its performance on a second unseen distribution. Studying the topological space of these simulators and how they overlap (or fail to overlap) may yield useful information in this vein.

In addition to further characterizing our simulators, we can also characterize the models themselves, identifying what features of the data factor into their predictions and why. This falls under the umbrella of explainable AI (XAI) research: work to develop new methods and algorithms capable of shedding light on the "black box" of a machine learning model. Gradient-based techniques such as Saliency Mapping and Integrated Gradients have been used in prior research involving AI-based EoR parameter recovery in an attempt to identify which parts of the input images feature most heavily in the model's decision making (Lahiry et al., 2025; La Plante and Ntampaka, 2019; Prelogović et al., 2021). We have found these localized feature attribution methods to be insufficient for this task: there are visible differences in the interpretability metrics between models trained with more simulators when compared with model trained on a single simulator, but they vary significantly from sample to sample and no clear trend presents itself. Future research developing non-localized interpretability methods is already in the works.

The ultimate goal of this simulation-based AI parameter inference of the EoR would be to develop a suite of training data that maximizes an AI's ability to perform well on unseen reionization scenarios. This would allow for the eventual use of AI on analysis of real-world EoR data.

CHAPTER 3

Counterfactuals for Data Augmentation and Interpreting AI Models

As discussed in Section 1.2.2, AI models can be considered "black boxes" in that the reasons for their decisions are not readily accessible to those looking to interpret them. To use AI models for science, we must have insight into the physical reasoning they operate under; otherwise, we have no way of checking their work or deriving meaningful insights from their determinations. Therefore, it is important to find interpretability methods that tell us useful information about the decision-making of AI models if we are to gain any insight about the patterns in our data. Counterfactual explanations are a method of interpreting black box models that uses counterexamples to highlight attributes important to the model's decision making process. Counterfactuals may also act as a form of data augmentation, generating new inputs from existing data in order to patch holes in the model's reasoning.

In this chapter we first discuss the shortcomings of localized feature attribution methods (interpretability methods meant to uncover features relevant to decision making

processes) then delve into counterfactual generation as well as two potential use cases for counterfactuals: interpretation and data augmentation.

3.1 Shortcomings of Localized Feature Attribution

Many existing feature attribution methods for image data operate in the local regime; i.e., they tell us what regions of a given input image were most important to the decisions made. These methods include saliency maps (Simonyan et al., 2014), integrated gradients (Sundararajan et al., 2017), and other more sophisticated map-based algorithms (for a recent review of various XAI algorithms see Dwivedi et al., 2023 or Bodria et al., 2023). These methods are easily implemented and provide post-hoc explanations for pretrained models, which makes them a popular first choice for interpreting models trained on EoR data (La Plante and Ntampaka, 2019; Prelogović et al., 2021; Lahiry et al., 2025).

However, there are limitations to these approaches. Because they are designed to highlight local features, they cannot identify image-wide statistical features in the data that typically drive cosmological analysis (such as the power spectrum or distribution of ionized bubble sizes). The resulting maps are therefore hard to interpret meaningfully. As an example, Figures 3.1 to 3.3 show the integrated gradients attribution maps for a given out-of-distribution sample passed through two of the models trained in Chapter 2, a single-simulator model and a triple-simulator model. Samples where the performance improved from the single to the triple simulator model were specifically selected for comparison, to see if differences in the attribute mapping could reveal why the out-of-distribution performance improved. Figures 3.1 to 3.3 highlight the difficulty in gleaning meaningful insights from these maps; the integrated gradients differ between the two models, but there is no obvious trend that can be invoked to explain the improvements achieved by the triple-simulator models. Similar results are seen with saliency maps.

For analysis of EoR data, we require an interpretability method that highlights global

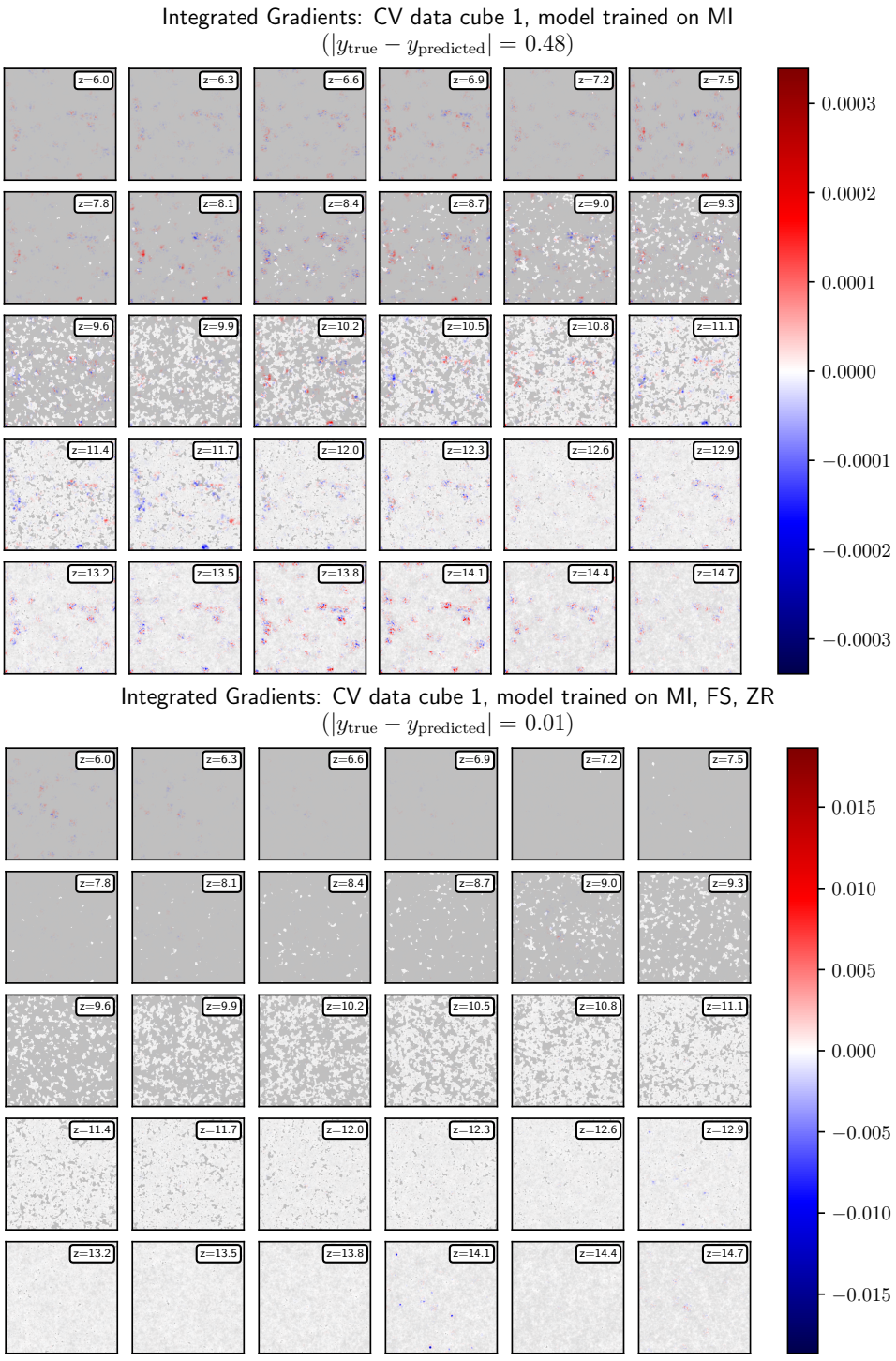

Figure 3.1: Integrated Gradient map for a single and triple simulator model predicting on an out-of-distribution sample from Dataset `CV`. Each 256×256 image represents one of the 30 redshift slices that make up a single data cube. The gray and white represents the brightness temperature while the red and blue represents the integrated gradients. For ease of viewing, a gaussian smoothing filter was applied to the attribution map with $\sigma = 1$. In this case, integrated gradients does not provide a clear explanation for the improvement in performance between the two models.

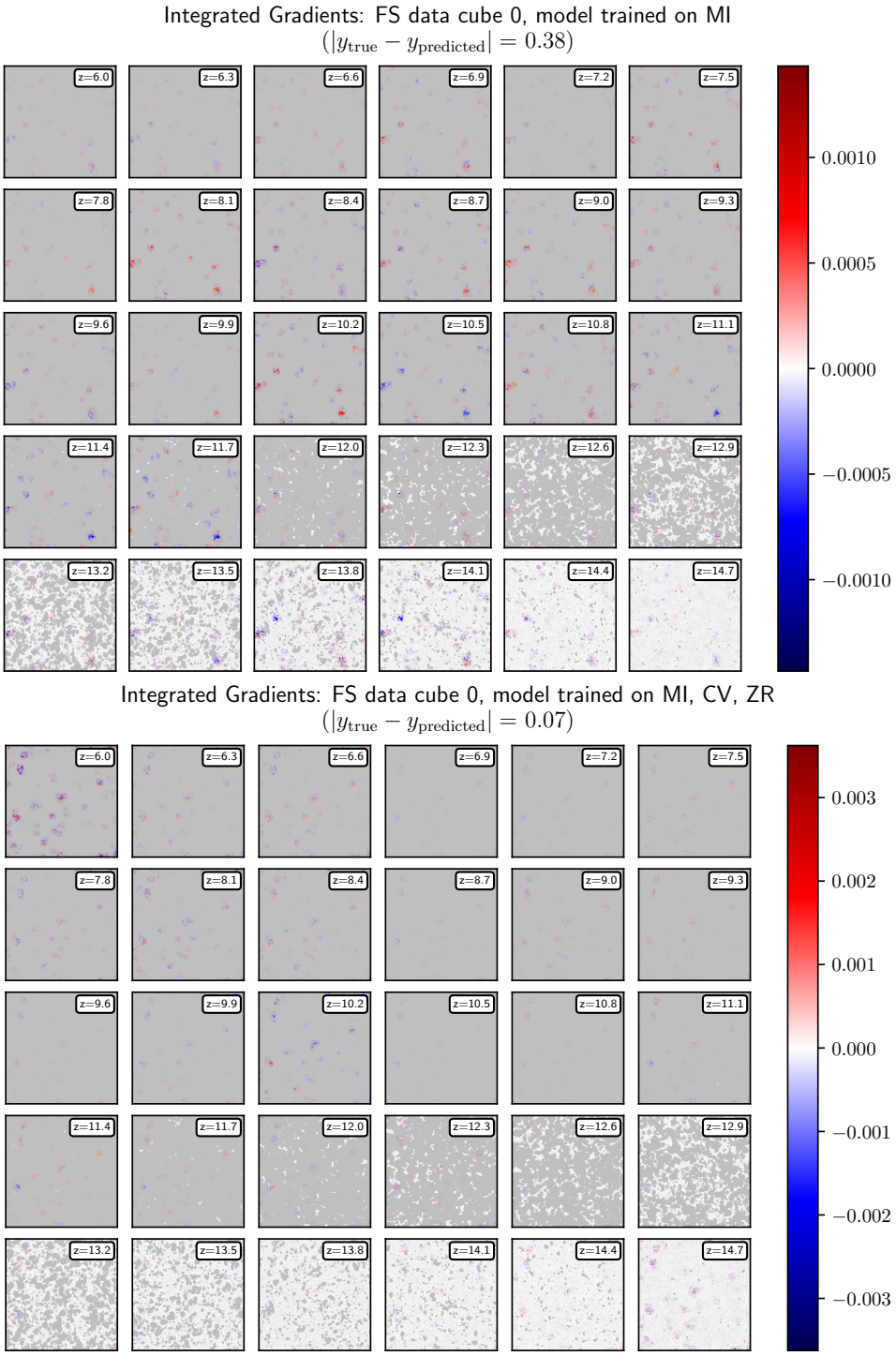

Figure 3.2: Integrated Gradient map for a single and triple simulator model predicting on an out-of-distribution sample from Dataset `FS`, analogous to Figure 3.1.

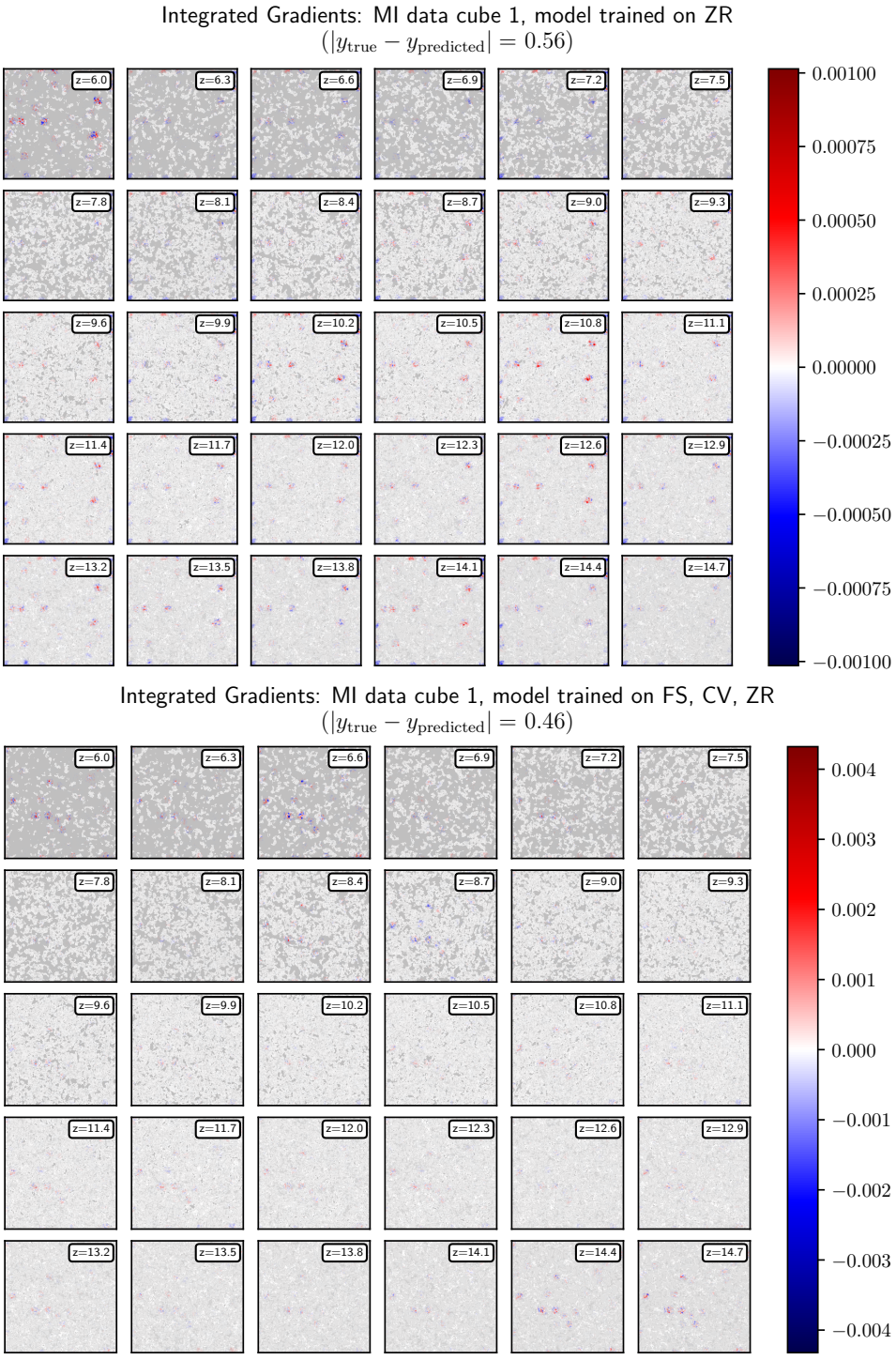

Figure 3.3: Integrated Gradient map for a single and triple simulator model predicting on an out-of-distribution sample from Dataset `MI`, analogous to Figure 3.1.

features of the image space, can be applied to a model post-hoc, and are applicable to regression models where classes / features are not predefined. For the rest of this section, we explore counterfactual explanations as an interpretability method that potentially fulfills all three of these requirements.

3.2 Generating Counterfactuals

Counterfactuals are an AI interpretability method that determines the minimum change we can make to an input of our model to get a different result. This is most commonly applied to classification tasks (e.g., what do I need to change about my loan application for it to be approved?) but can also be used in regression tasks (i.e., what are the smallest changes I can make to my financial profile to achieve the biggest improvement in my credit score?). Counterfactuals provide explanation by way of cause-and-effect: if changing a given feature alters the outcome, then that feature must be key to the model's decision making process.

Counterfactuals are generated by performing iteratively permuting a latent representation of the input via gradient descent[1]. To do so, we must define an optimization function with which to compute our gradients. Let f be an encoder that maps input $\mathbf{x}$ to some latent vector $\mathbf{v}$ and h be a regressor that translates $\mathbf{v}$ into a predicted label y. Our "model" in this case is the nested function $h \circ f$, which is pretrained and remains static for the duration of the counterfactual generation. The latent vector $\mathbf{v}$ is key to our optimization process in that it represents a lower-dimensional feature representation of $\mathbf{x}$. By permuting elements of $\mathbf{v}$, we are turning the knobs and dials of $\mathbf{x}$'s features as defined by our pretrained encoder. A decoder g is then required to map $\mathbf{v}$ back to $\mathbf{x}$, with some level of information loss relative to the quality of the decoder. Our goal is to generate a counterfactual $g(\mathbf{v}') = \mathbf{x}'$ that is similar to $\mathbf{x}$ by some metric, but where the new predicted

[1]This is similar to the process of training the initial model (and indeed uses the same algorithms), but instead of updating the model's weights at every iteration we slowly permute the underlying latent feature representation of the input.

value $h(f(\mathbf{x}')) = h(\mathbf{v}') = y'$ maximally differs in some way from the original prediction $h(f(\mathbf{x})) = h(\mathbf{v}) = y$. Therefore, our optimization function must take the form

$$\mathcal{L} = \arg\min_{\mathbf{v}'} \quad d_x\left(g(\mathbf{v}'), \mathbf{x}\right) - d_y\left(h(\mathbf{v}'), h\left(f(\mathbf{x})\right)\right) \tag{3.1}$$

where d_x and d_y are distance functions for the input and output spaces, respectively.

If there exists no decoder that can reliably decompress the latent vector, we can choose to optimize over image space instead. This reduces our optimization function to

$$\mathcal{L} = \arg\min_{\mathbf{x}'} \quad d_x\left(\mathbf{x}', \mathbf{x}\right) - d_y\left(h\left(f(\mathbf{x}')\right), h\left(f(\mathbf{x})\right)\right) \tag{3.2}$$

This is non-ideal because we lose the benefit of inherently global feature attribution: it is the manipulation of the feature vector $\mathbf{v}'$ that gives counterfactual generation its global quality, and optimizing over image space puts us back in the local regime. We make this concession when generating counterfactuals, as when training the models in Chapter 2 we did not concurrently train a decoder to decompress the latent vector. Future research in this lane should focus first on developing an autoencoder with a robust feature space capable of mapping $\mathbf{x} \to \mathbf{v} \to \mathbf{x}$ with minimal loss.

3.2.1 MSE based optimizers

In their simplest forms, the distance functions d_x and d_y can be expressed as the MSE between the original and counterfactual cases, weighed in impact by w_x and w_y, respectively. In this case Equation 3.1 becomes

$$\mathcal{L}_{\text{MSE}} = \arg\min_{\mathbf{v}'} \quad w_x||\mathbf{x} - g(\mathbf{v}')||^2 - w_y||h\left(f(\mathbf{x})\right) - h(\mathbf{v}')||^2 \tag{3.3}$$

or, when optimizing over image space,

$$\mathcal{L}_{\text{MSE}} = \arg\min_{\mathbf{x}'} \quad w_x||\mathbf{x} - \mathbf{x}'||^2 - w_y||h\left(f(\mathbf{x})\right) - h\left(f(\mathbf{x}')\right)||^2 \tag{3.4}$$

Values of w_x and w_y are tunable parameters which require some trial and error to optimize. Other task-agnostic distance functions have been explored in the literature, and a review can be found in Section 4.3 of Guidotti (2024) (some of which may be more appropriate for the task at hand), but in this exploratory work we focus on this basic case.

When generating counterfactuals for a regression task, additional stipulations are required in the optimization function in order to prevent $d_x \to -\infty$ and $d_x \to \infty$. There are two ways of doing this: define distance functions that converge to a given value or add a balancing term to the optimization function. In this work we mostly explore the latter option, using MSE-based distance functions as in Equation 3.4 while adding additional penalty terms that limit the range of potential $\mathbf{x}'$ values (future work may explore using both in conjunction for added stability if continuing to optimize over image space). We define an additional term d_{ramp} to penalize the counterfactual when any pixel $x'_{i,j,k}$ in $\mathbf{x}'$ exits the range of the minimum and maximum pixel values of the original redshift slice $\mathbf{x}_{\text{z}}$, where d_{ramp} can be expressed in terms of ramp functions $R(x) = \max(x, 0)$:

$$\begin{aligned} d_{\text{ramp}} &= (d_{\text{ramp,+}} + d_{\text{ramp,-}}) \\ &= w_{ramp} \sum_{i,j,k=0}^{x,y,z} R\left(\left[x'_{i,j,k} - \max(\mathbf{x}_k)\right]^2\right) + R\left(\left[-x'_{i,j,k} + \min(\mathbf{x}_k)\right]^2\right) \end{aligned} \tag{3.5}$$

The addition of this term was motivated by values of $\mathbf{x}'$ blowing up to ∞ when using Equation 3.4 as an optimization function. Terms inside R are squared to ensure smoothness in gradient space.

3.2.2 Cross-correlation based optimizers

It is also possible to define a d_x specific to the task at hand. When optimizing over image space as in Equation 3.4, we lose the benefit of global feature attribution; that is, by not optimizing over a latent feature space, the counterfactual optimization process works in pixel space, ignoring image-wide features of the data. A carefully designed d_x may mitigate some of the loss of nonlocal feature attribution by directing the optimization process to focus on a user-defined feature space.

The cross-correlation is a common metric of similarity used in cosmological image analysis. It has a range of [-1, 1], with $a \star b = 1$ indicating the signals are perfectly correlated (when a increases in magnitude so does b, and vice versa), $a \star b = -1$ indicating the signals are perfectly anti-correlated, and $a \star b = 0$ meaning the signals are uncorrelated. The cross-correlation provides a global measure of similarity between two inputs, and so it makes sense to define a d_x in terms of it if we want to recover some degree of nonlocality in our feature attribution:

$$d_x = -w_{xcorr}(\mathbf{x} \star \mathbf{x}') + w_{mse}||\mathbf{x} - \mathbf{x}'||^2 \tag{3.6}$$

The MSE term is included because $-w_{xcorr}(\mathbf{x} \star \mathbf{x}')$ can be minimized with $\mathbf{x}$ and $\mathbf{x}'$ still having very different magnitudes.

3.2.3 Method

We generate counterfactuals for the model trained in Section 2.1.2, minimizing difference in image space while maximizing difference between labels. In this way, we attempt to generate a counterfactual that makes superficial changes to the reionization history that still results in a radically different predicted durations of reionization. In doing so, we hope to reveal flaws in the model's reasoning.

We initialize each counterfactual $\mathbf{x}'$ as the original data cube $\mathbf{x}$ (normalized to the

$\mathcal{L}$	w_{mse}	w_{xcorr}	w_y	w_{ramp}
$\mathcal{L}_{\text{MSE}}$	10^5	-	1	100
$\mathcal{L}_{\text{XCORR}}$	10^5	10^5	1	100

Table 3.1: Relative weights of terms in the optimization functions used to generate counterfactuals. These are tunable parameters which control how much influence a particular term has on the counterfactual generation—for example, increasing w_{mse} relative to w_y makes the optimizer favor preserving similarities between $\mathbf{x}$ and $\mathbf{x}'$ over maximizing the difference between y and y'. w_{ramp} is low in comparison to w_{mse} and w_{xcorr} because, as coded, d_{ramp} performs a sum over all the values of $\mathbf{x}'$ whereas d_{mse} takes the mean. If d_{ramp} averaged over the values of $\mathbf{x}'$, w_{ramp} would need to be increased, as its effectiveness as a penalty term requires it to be weighted heavily.

interval [0, 1] and preprocessed as described in Section 2.1.1) plus a field of random gaussian noise drawn from $\mathcal{N}(0, w_{\text{init}})$, where w_{init} is a tunable parameter (we use a $w_{\text{init}} = 0.001$). The addition of a small initial displacement from $\mathbf{x}$ is necessary for the gradients to be nonzero and propagate through the model.

We use PyTorch as a framework to perform the iterative gradient descent, defining an Adam optimizer with $\mathbf{x}'$ as the optimization target. We iterate for 5000 iterations, with a multistep learning rate that starts at 0.001 and is multiplied by a factor of 0.1 every 2000 iterations. Optimization function weights were tuned by trial and error (values used can be found in Table 3.1).

Two optimization functions were tested: an MSE-based optimizer given by Equation 3.4 and a cross-correlation based optimizer given by Equation 3.6. Both included the additional d_{ramp} term given by Equation 3.5 to penalize each slice of $\mathbf{x}'$ from leaving the range of the corresponding redshift slice of the original data cube $\mathbf{x}$.

3.2.4 Results

Figures 3.4 to 3.6 show three examples of counterfactuals generated with the MSE based optimizer, along with the evolution of terms of the optimization function. Each 256×256 grayscale image represents one of the 30 redshift slices that make up the

original data cube, while the overlaid red and white represents the difference between the counterfactual and original $\mathbf{x}' - \mathbf{x}$.

These examples are comparable to the Integrated Gradient feature attribute maps shown in Figures 3.1 to 3.3, highlighting the fact that optimizing over image space without a specially designed distance function results in localized feature attribution.

Despite the shortcomings of this optimization function, however, it still highlights undesirable behaviors in the model. Predicted values for the duration of reionization ranging from 15-30 are highly improbable, especially given the relatively minimal changes made to the brightness temperature slices of the input. This tells us that our model has a flaw in its internal representation of the task. Addressing this sort of flaw with our generated counterfactuals will be explored in Section 3.3.

Figures 3.7 to 3.9 show three examples of counterfactuals generated with the cross-correlation based optimizer, along with the evolution of terms of the optimization function. These counterfactuals, in contrast with those generated with the MSE based optimizer, display attribute changes much more global in scope. Still, which attributes impact the change in predicted duration are not clear. One interesting feature that is readily observable is the change in overall magnitude between redshift slices relative to $\mathbf{x}$. This is especially identifiable in Figure 3.9, though is present in all examples shown here. At low redshift values, the global magnitude of $\mathbf{x}'$ is increased relative to $\mathbf{x}$ (appears more red), and at high redshift values, the relative global magnitude is decreased (appears more blue). This means that in maximizing distance between y and y', the optimizer found the path of least resistance to be reducing the change in global brightness temperature over time. This actually highlights intelligent behavior on the model's part: a longer reionization (and all examples presented here have $y' > y$) would indeed result in a more gradual change in global brightness temperature as a function of redshift. This, then, shows that the overall change in magnitude of the signal may be a feature the model considers when making a prediction.

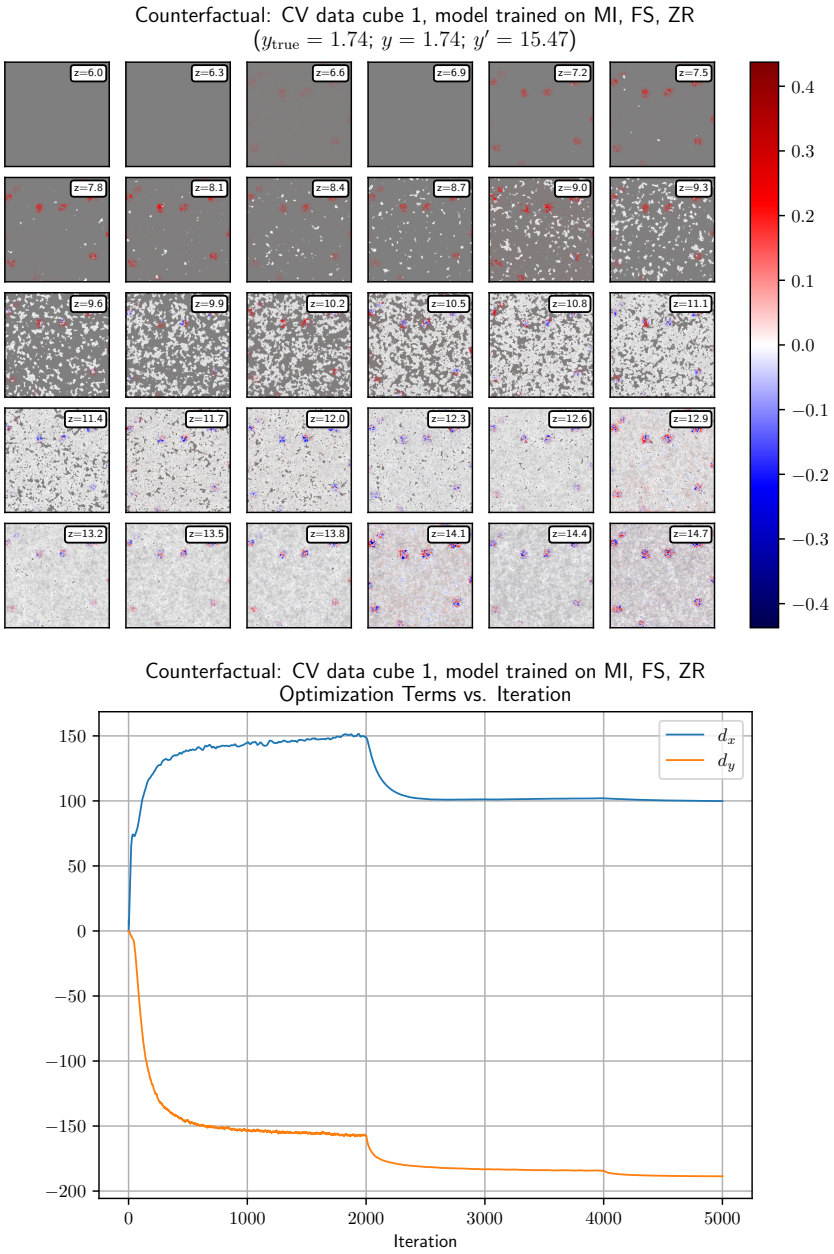

Figure 3.4: (Top) Example counterfactual generated from a triple simulator model and out-of-distribution dataset `CV` sample, using an MSE based optimization function. Each 256×256 grayscale image represents one of the 30 redshift slices that make up the original data cube, and is normalized to the interval [0,1]. The overlaid red and blue represents the difference between the counterfactual and original $\mathbf{x}' - \mathbf{x}$. A gaussian smoothing filter with $\sigma = 1$ was applied to the overlay. Similar to the integrated gradients maps in Figures 3.1 to 3.3, the feature attribution resulting from this naive optimization function is local in scope and difficult to interpret. (Bottom) Optimization function terms vs. iteration. d_{ramp} is excluded as it is significantly higher in magnitude than the other terms. Sharp changes in the lines at iterations 2000 and 4000 are a result of the drop in learning rate.

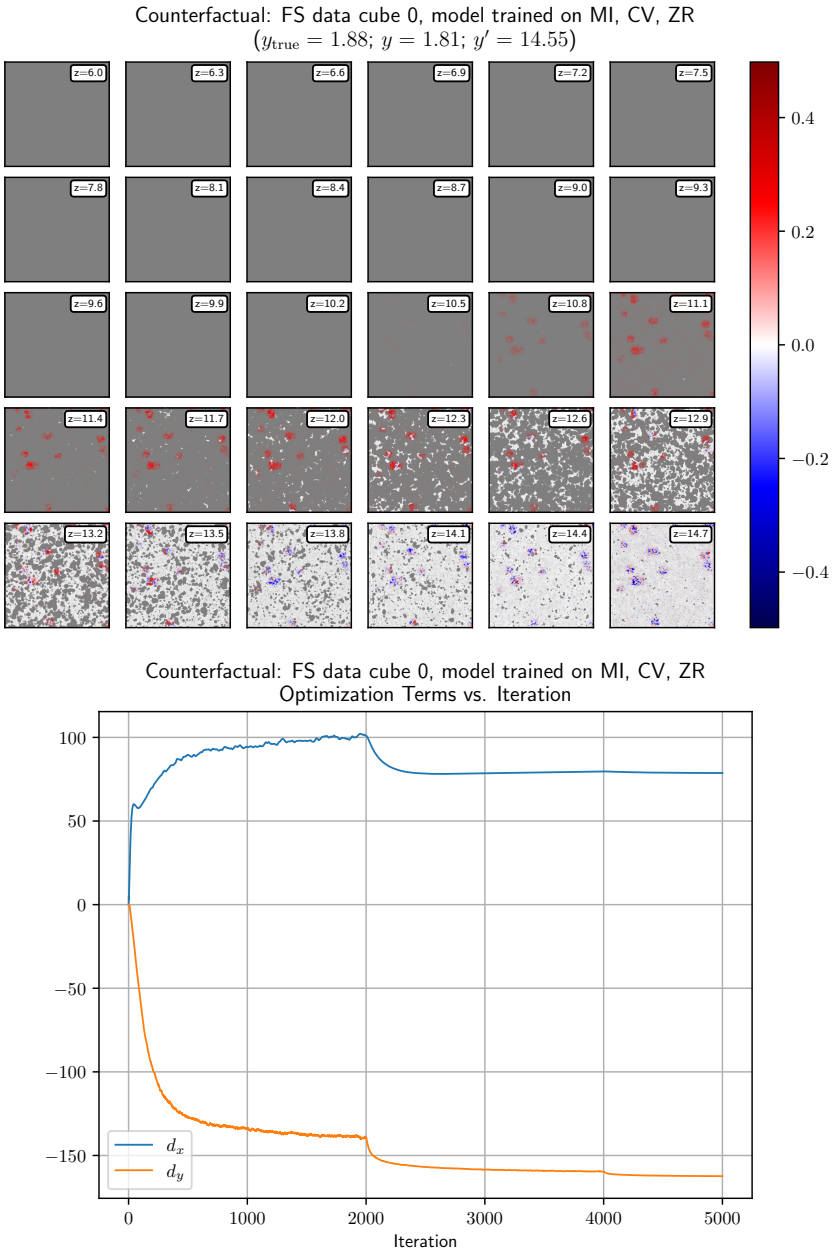

Figure 3.5: (Top) Example counterfactual generated from a triple simulator model and out-of-distribution dataset `FS` sample, using an MSE based optimization function. Otherwise identical to Figure 3.4. (Bottom) Optimization function terms vs. iteration.

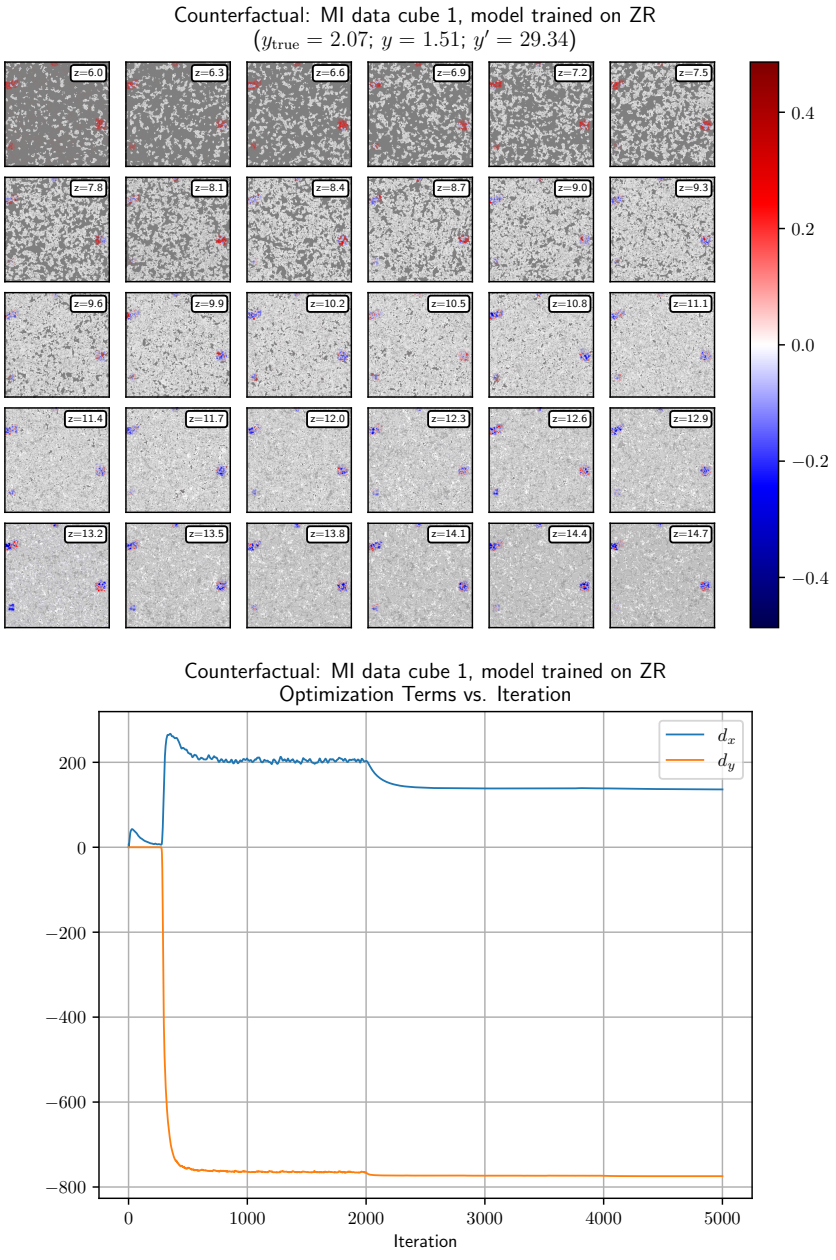

Figure 3.6: (Top) Example counterfactual generated from a model trained on dataset `ZR` and out-of-distribution dataset `FS` sample, using an MSE based optimization function. Otherwise identical to Figure 3.4. (Bottom) Optimization function terms vs. iteration.

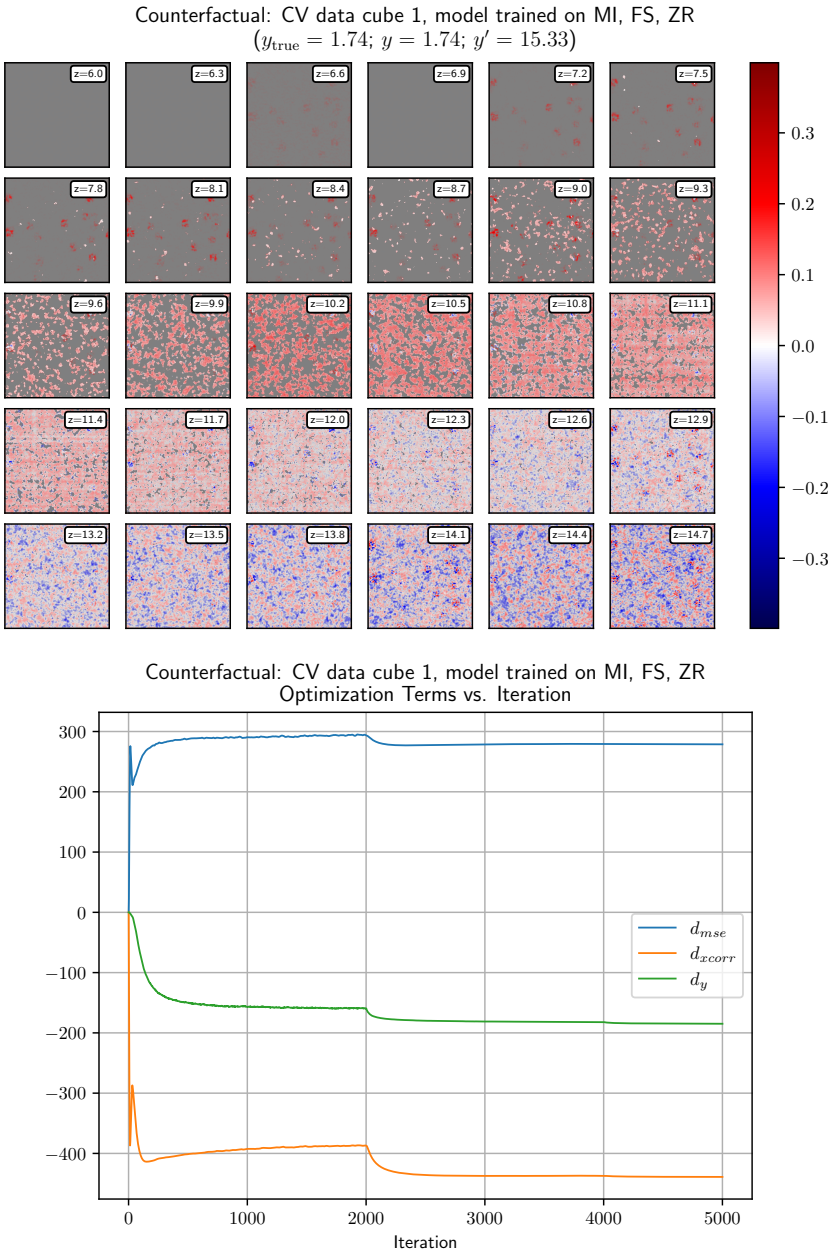

Figure 3.7: (Top) Same as Figure 3.4, except counterfactual was generated using a cross-correlation based optimization function. (Bottom) Optimization function terms vs. iteration. d_{xcorr} is shifted up in the y-axis for ease of comparison, with $d_{xcorr,0}$ being set to 0.

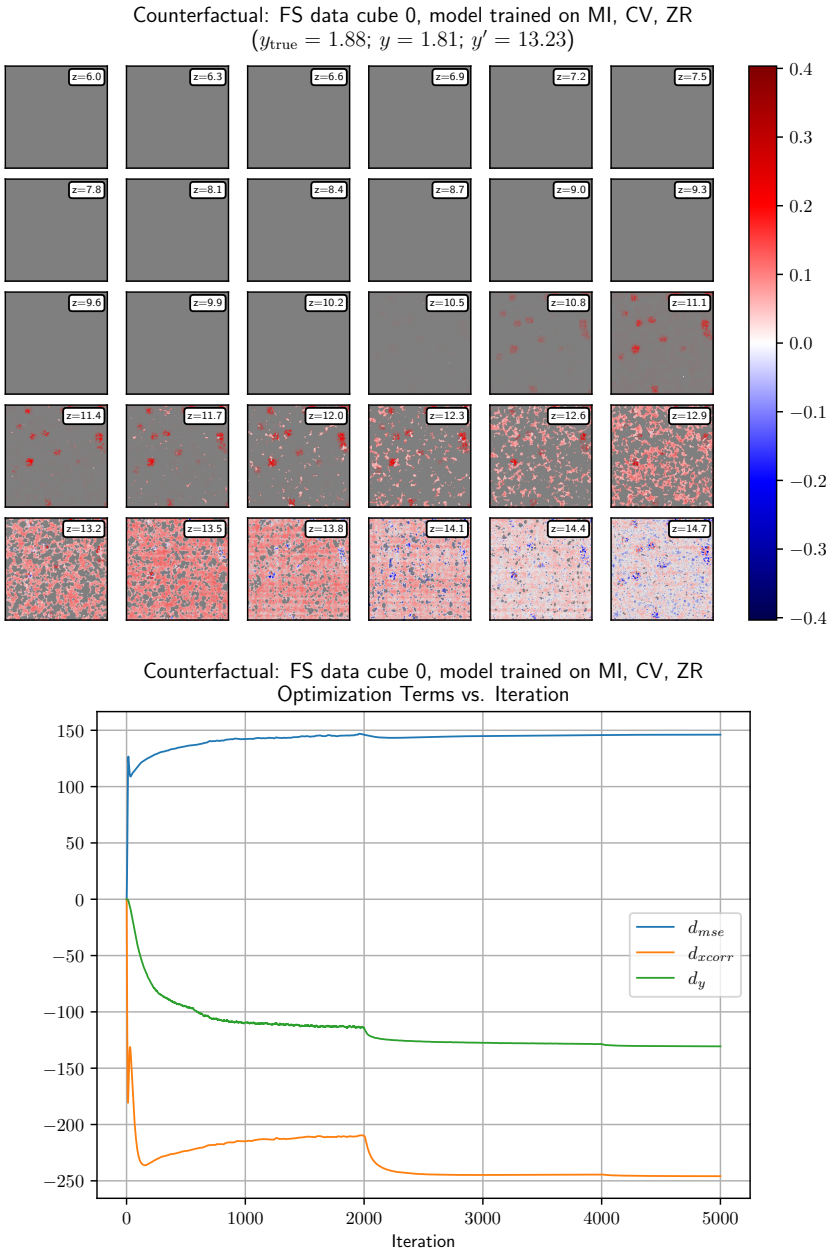

Figure 3.8: (Top) Example counterfactual generated from a triple simulator model and out-of-distribution dataset `FS` sample, using a cross-correlation based optimization function. Otherwise identical to Figure 3.7. (Bottom) Optimization function terms vs. iteration.

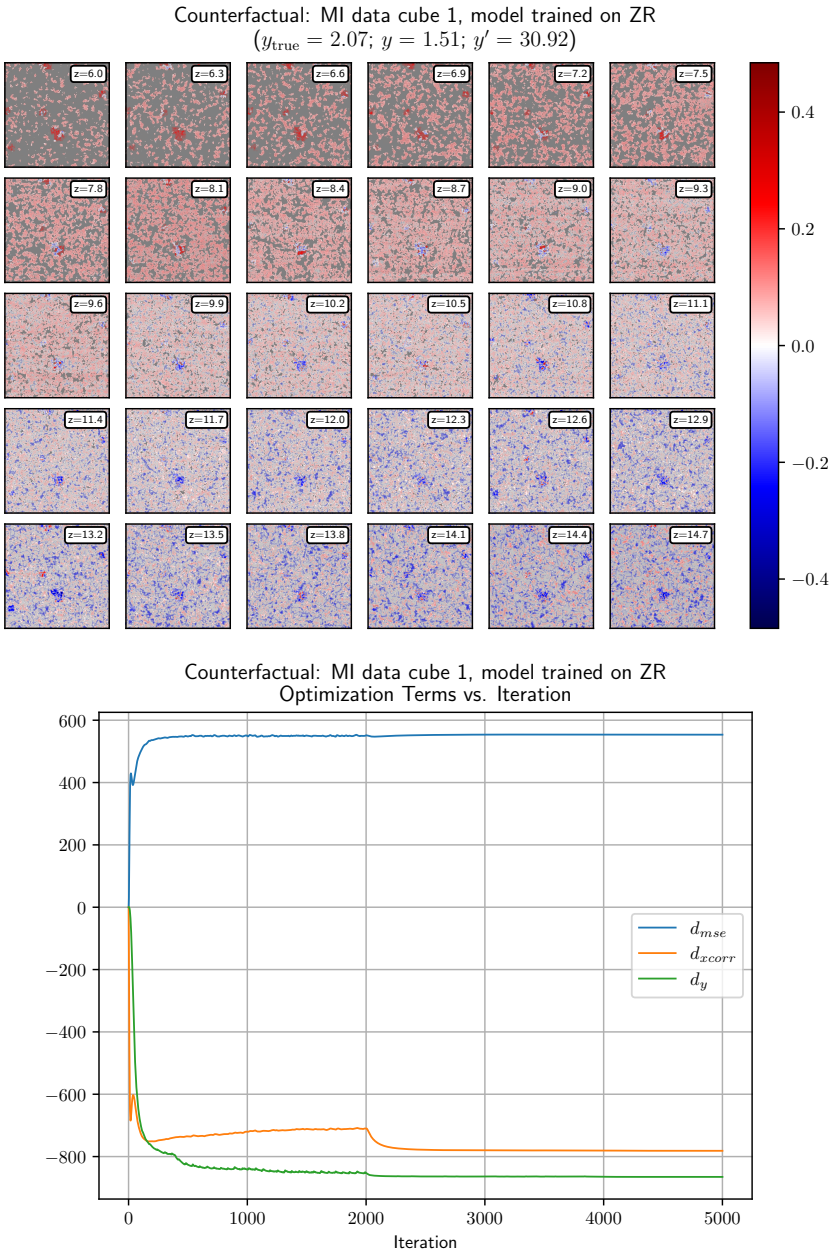

Figure 3.9: (Top) Example counterfactual generated from a model trained on dataset `ZR` and out-of-distribution dataset `FS` sample, using a cross-correlation based optimization function. Otherwise identical to Figure 3.7. (Bottom) Optimization function terms vs. iteration.

3.2.5 Discussion

Further investigation into different counterfactual optimization functions is needed before any conclusions can be made about the viability of using counterfactuals as interpretability tools for AI parameter estimation of the EoR. However, the examples provided here still generate some content worthy of discussion.

Counterfactual Realism

An obvious question worth asking is "are these counterfactuals physically reasonable?" A counterfactual that presents an impossible reionization history provides little useful information about our model, only telling us that an unrealistic input results in an extreme output. Therefore, it is worth expending the effort to make counterfactuals with plausible reionization histories. As physical realism is not a definite quantity one can test for, we adopt a "by-eye" approach to assessing a counterfactual's realism.

Some safeguards have been included in the optimization functions used in this work to ensure a degree of physical realism. For example, d_{ramp} individually penalizes each redshift slice of $\mathbf{x}'$ for values that exceed the range set by the maximum and minimum values of the corresponding slice of $\mathbf{x}$. This encourages the counterfactual to have a reionization history with roughly the same duration as the original sample and also ensures that reionization proceeds in a steady manner, with each subsequent redshift slice having less neutral hydrogen than the once before it, as expected of the progression of reionization. d_{ramp} also discourages extreme bright spots and negative temperature values, which would be nonphysical. Still, other restrictions could be built into the optimization function to ensure even more realistic reionization histories, a potential topic for future exploration.

Undesirable Features

The example counterfactuals presented here display several features that we would prefer they not have, the most notable of which is the repetition of structure across the

redshift axis. Using Figure 3.6 as an example, almost every slice has the same pattern of local changes, with two major spots of interest in the top left quarter of every image. This z-axis structure is likely a relic of backpropagating through 2D convolutional layers and may bolster some research that suggests using 3D convolutional layers is a better choice for this sort of model (Zhao et al., 2022).

Another odd feature is a gridlike striation pattern present in some of the counterfactuals generated with the cross-correlation optimizer. The reason for this pattern is unknown, and is likely either a bug in the code or a result of the difficulties inherent to backpropagating gradients through the operators involved in calculating the cross-correlation.

Interpretation

Ultimately, the counterfactuals generated by either of the considered optimization functions are difficult to interpret, somewhat belying their potential as interpretability tools. The MSE optimized counterfactuals are about as opaque to explanation as the integrated gradients maps. The cross-correlation optimized counterfactuals, on the other hand, seemingly identify at least one feature the model considers important when estimating the duration of reionization—the change in the magnitude of the global brightness temperature. However, these counterfactuals still prove difficult to interpret by eye.

It is possible a more thorough analysis would uncover statistical differences between counterfactuals and their parent inputs that indicate other features deemed important by the model. However, a more promising next step involves concurrently training a robust autoencoder $g(f(\mathbf{x})) = \mathbf{x}$ alongside our parameter estimator $h(f(\mathbf{x})) = y$. This would naturally create a latent feature space over which to optimize, a fact that may result in more interpretable counterfactuals.

Training an autoencoder that compresses a complex 3D input $\mathbf{x}$ to a lower-dimensional vector $\mathbf{v}$ and then maps $\mathbf{v}$ back to $\mathbf{x}$ with minimal error is not a trivial task. Training an autoencoder from scratch was initially attempted for this work but was abandoned

when the compression to latent space continued to result in a huge loss of information. Using large pretrained autoencoders such as Stable Diffusion's off-the-shelf Variational Autoencoder[2] was able to encode / decode EoR data with minimal information loss, but the latent feature space was large enough that the features important specifically to EoR data cubes were difficult to distinguish. Future attempts utilizing smarter autoencoder architectures may find a comfortable middle ground between these two extremes, with a latent feature space complex enough to recreate EoR data cubes with minimal loss while simple enough that each EoR cube spans the breadth of feature space.

3.3 Data Augmentation

Counterfactuals are most frequently discussed in the context of model interpretability. However, they also have potential in the realm of data augmentation. The counterfactuals presented in Section 3.2.4 identified gaps in the model's understanding of the data—namely, that relatively minor changes in the model's input could result in a drastic, unphysical output. Following up on the results presented in Chapter 2, a dataset of counterfactuals generated with a given optimizer may act as an additional distribution with which to add diversity to the pool of training data, potentially improving our models' OOD performance without the costly overhead of generating new EoR simulations. The rest of this chapter conducts a preliminary investigation of this possibility, using counterfactuals generated from the existing data with the basic MSE optimizer to finetune our pretrained models.

3.3.1 Method

We generated a supplementary dataset of 2400 counterfactuals using the same procedure as in 3.2.3 with the MSE based optimization function for each of the 14 models trained in Chapter 2. The counterfactuals were generated using only in-distribution samples for the given model; in the case of models trained on multiple datasets, the representation

[2]https://huggingface.co/CompVis/stable-diffusion-v1-4

of each dataset in the base data was kept equal, with 1200 samples per dataset for the double dataset models and 800 for the triple dataset models. This is identical to how the data was partitioned in Section 2.1.1. No counterfactuals were added to the validation or test data.

We then finetuned the base models with the original training dataset augmented by the counterfactual dataset, thereby doubling the training dataset in size without generating any new reionization histories. Finetuning involves loading a pretrained model and subjecting it to further training iterations with different data. In this case the original training dataset was included in the finetuning alongside the counterfactuals in order to prevent the model from "forgetting" its original training base. We finetune each model for 1000 epochs with a fixed learning rate of 0.001. Otherwise, the finetuning procedure was identical to that of the original training described in Section 2.1.1. No k-fold validation was performed for this analysis.

Counterfactuals were labeled according to their original data cubes such that each $\mathbf{x}'$ shares a y_{true} value with its corresponding $\mathbf{x}$, aka $y_{true} = y'_{true}$. The realism of this choice is discussed in Section 3.3.3.

3.3.2 Results

The top right panel of Figure 3.10 gives the average MSE prediction error grouped by the number of distributions included in the training data, analogous to Figure 2.8. The bottom panel of Figure 3.10 presents that same data minus the 8-fold validated results of the base model given by Figure 2.8. The bottom panel of Figure 3.10 seems to imply some small improvement in OOD prediction error as a function of the number of simulators represented in the data when models are finetuned with data supplemented by counterfactuals, though the significance of these results require further validation to confirm. If these results are taken at face value, however, it is notable that the impact of adding counterfactuals to the finetuning data seems dependent on how many distributions

Figure 3.10: (top left) Out-of-distribution performance heatmap for the base models averaged across number of simulators represented in the base training data, identical to Figure 2.8. (top right) The same results for the models finetuned with counterfactuals. This heatmap does not include standard error because k-fold validation was not performed during finetuning. (bottom) The difference between the two. This heatmap seemingly implies that finetuning with counterfactuals does improve performance slightly for the triple-simulator models, though further validation of the finetuned models is required before this result can be confirmed.

are represented in the original training data. The OOD performance of models trained on a single distribution are either unchanged or actively worsened by the addition of counterfactuals to the finetuning data. Meanwhile, the OOD performance models trained on data from three simulators universally improves. Reasons for this are unclear and will be speculated upon in Section 3.3.3.

3.3.3 Discussion

Variable Impact of Counterfactuals as Supplementary Data

Figure 3.10 seems to imply that the benefit counterfactuals provide as supplementary data is related to the diversity of the base training data; i.e., that models with exposure to more data distributions either generate more useful counterfactuals or benefit more from the additional diversity the data augmentation provides. The second possibility is difficult to investigate without additional datasets generated from new simulators, as with only one triple-simulator OOD model per dataset there are currently too few data points to identify a trend either which way. Performing a systematic analysis on the counterfactuals generated by models trained on one, two, and three simulators may shed light on the first, but this analysis will be left to future work.

Label Choice

In finetuning our models with our counterfactual-augmented dataset, we chose to carry over the "true" duration of reionization for our counterfactuals from their parent reionization history. The realism of this assumption is suspect and highly dependent on the degree of similarity between the original sample and its corresponding counterfactual. For example, the counterfactual presented in Figure 3.5 introduces structure at low redshifts that isn't present in the original data cube. Theoretically, this would lengthen the duration of reionization, as neutral hydrogen clouds remain later in cosmological history, and thus increase y'_{true}. We make the assumption $y'_{true} = y_{true}$ out of necessity, as estimating the

duration of reionization from a sliced data cube has a high degree of error for simulators with partial voxel ionization such as `21cmFAST`(see Section 1.3.2). We attempt to account for this mislabeling by prioritizing similarity in image space over difference in label space, aka, setting $w_x >> w_y$. Additionally, the difference between y_{true} and y'_{true} is typically inconsequential when compared to the difference between y_{true} and y', thus justifying the small labeling error as a "good enough" correction.

Equivalency of Base and Finetuned Models

When evaluating the change in performance following finetuning, we compare the performance of the 8-fold validated base model to our single finetuned model. The equivalence of these two models as a basis for comparison is worth looking into.

Firstly, as no k-fold validation was performed for the finetuned model, the standard error of our finetuned results is unknown. An equivalent validation process would need to be performed for our finetuned models before any conclusions can be drawn from the data.

There is also the matter of dataset size. Typically, a larger set of training data contains more information and therefore results in better model performance. However, as the counterfactuals were generated only from samples present in the training data, there was theoretically no additional information included from outside sources: all data cubes used for finetuning were either seen during pretraining or derived from those seen during pretraining, and any benefit in performance came from the changes introduced by the counterfactual optimization process. We argue then that so long as the base training datasets are equivalent, model performance can be compared without the size of the total training set being a variable factor.

Finally, there is the matter of training duration. While the original model was trained to convergence, there is a slight possibility that an additional 1000 epochs of training might impact model performance in some way. Future work may test this by finetuning

the base model twice: once with just the base training data, and once with the addition of counterfactuals as supplement. However, the likelihood that this would change the results is very low.

CHAPTER 4

Conclusion

AI is a promising tool for constraining parameters of reionization, but it is not without its pitfalls. AI models are known to struggle with out-of-distribution prediction tasks, which is precisely the type of task analyzing real-world 21cm data will require.

Regressing on parameters of the EoR, both via AI or more traditional Bayesian methods, has largely been studied only in the context of one parameter space. This is a problem because different simulators of reionization have different strengths and weaknesses. A method that combines the strengths of multiple simulators while compensating for their weaknesses would allow for a more unbiased, simulator-independent treatment of the data.

This work investigates the cross-simulator performance of AI models, and finds that in general AI models trained on data from a single simulator of reionization fail to generalize to data from a second unseen simulator. When the training data is diversified by the inclusion of data from multiple simulators in the training pool, the AI model's performance on a held-out dataset improves. This implies that a diverse training set with data drawn from many different EoR simulators is essential to accurately predicting on out-of-distribution data, such as actual 21cm observation data.

Finally, this work conducts a preliminary investigation into counterfactuals as a tool

for both model interpretability and data augmentation. The feature space over which the optimization is performed and the form of the optimization function are the primary factors that determine how counterfactuals are generated and how they contrast with their parent input. Counterfactuals can highlight holes in the model's reasoning as well as features important to the model's decision making process. Further investigation is needed to determine the efficacy of counterfactuals as a form of data augmentation.

Appendix A

Detailed Data and Model Parameters

A.1 Data Generation Parameters

Table A.1 details the parameter ranges used when generating our datasets.

A.2 Detailed Model Architecture

Table A.2 describes the detailed architecture of our model. Layer names and parameters are those used in the PyTorch implementation of our model.

Dataset Generation Parameters						
Parameter		Unit	Value (Dataset `CV`)	Value (Dataset `FS`)	Value (Dataset `MI`)	Value (Dataset `ZR`)
HII_DIM		-	256	256	512	256
BOX_LEN		Mpc	1000	1000	2000	1000
USE_MASS_DEPENDENT_ZETA		-	True	True	False	-
IONISE_ENTIRE_SPHERE		-	False	True	False	-
Parameter	Flat Prior	Unit	Allowed Range (Dataset `CV`)	Allowed Range (Dataset `FS`)	Allowed Range (Dataset `MI`)	Allowed Range (Dataset `ZR`)
$f_{*,10}$	log	-	$[0.001, 1]$	$[0.01, 1]$	-	-
α_*	linear	-	$[-0.5, 1]$	$[-0.5, 1]$	-	-
$f_{\rm esc,10}$	log	-	$[0.001, 1]$	$[0.01, 1]$	-	-
$\alpha_{\rm esc}$	linear	-	$[-1, 0.5]$	$[-1, 0.5]$	-	-
$M_{\rm turn}$	log	$\rm M_\odot$	$[10^8, 10^{10}]$	$[10^8, 10^{10}]$	-	-
t_*	linear	-	$[0, 1]$	$[0, 1]$	-	-
ζ	linear	-	-	-	$[10, 250]$	-
$T_{\rm vir}^{\rm min}$	log	K	-	-	$[10^4, 10^6]$	-
$\bar{z}$	linear	-	-	-	-	$[7, 13]$
α	linear	-	-	-	-	$[-0.5, 0.5]$
k_0	linear	$h\rm Mpc^{-1}$	-	-	-	$[0.1, 1]$

Table A.1: Settings and parameter ranges used to generate our datasets. If not otherwise specified, the default values were used. The allowed ranges were informed by Park et al., 2019 and Battaglia et al., 2013, but were adjusted to produce datasets with overlapping Δz distributions (see Figure 2.2).

Model Architecture			
Name	**Layer Class**	**Input Params**	**Trainable Params**
conv_stack_1	Conv2d	out_channels=16	4336
	ReLU	-	-
	BatchNorm2d	-	32
	MaxPool2d	kernel_size=2	-
conv_stack_2	Conv2d	out_channels=32	4640
	ReLU	-	-
	BatchNorm2d	-	64
	MaxPool2d	kernel_size=2	-
conv_stack_3	Conv2d	out_channels=64	18496
	ReLU	-	-
	BatchNorm2d	-	128
	MaxPool2d	kernel_size=2	-
global_maxpool	MaxPool2d	kernel_size=32	-
	Flatten	-	-
linear_stack_1	Dropout	-	-
	Linear	out_features=200	13000
	ReLU	-	-
linear_stack_2	Dropout	-	-
	Linear	out_features=100	20100
	ReLU	-	-
linear_stack_3	Dropout	-	-
	Linear	out_features=20	2020
	ReLU	-	-
output	Linear	out_features=1	21
	Total no. trainable parameters:		62837

Table A.2: A detailed description of our CNN architecture, as implemented in PyTorch. All Conv2d layers use the parameters `kernel_size=3` and `padding=same`. All Dropout layers use `p=0.2`.

Bibliography

Battaglia, Nicholas et al. (Oct. 2013). "REIONIZATION ON LARGE SCALES. III. PREDICTIONS FOR LOW-↕ COSMIC MICROWAVE BACKGROUND POLARIZATION AND HIGH-↕ KINETIC SUNYAEV–ZEL'DOVICH OBSERVABLES". In: *The Astrophysical Journal* 776.2, p. 83. DOI: 10.1088/0004-637X/776/2/83.

Becker, George D. et al. (Jan. 2015). "Evidence of patchy hydrogen reionization from an extreme Lyα trough below redshift six". In: *Monthly Notices of the Royal Astronomical Society* 447.4, pp. 3402–3419. ISSN: 0035-8711. DOI: 10.1093/mnras/stu2646. eprint: https://academic.oup.com/mnras/article-pdf/447/4/3402/5695265/stu2646.pdf. URL: https://doi.org/10.1093/mnras/stu2646.

Berklas, August and Jonathan Pober (Nov. 2025). "Exploring the Model Dependence of MCMC-Based 21 cm Power Spectrum Parameter Constraints". In: *arXiv preprint arXiv:2511.13854*. DOI: 10.48550/ARXIV.2511.13854. arXiv: 2511.13854. URL: https://arxiv.org/abs/2511.13854.

Billings, Tashalee S., Paul La Plante, and James E. Aguirre (Mar. 2021). "Extracting the Optical Depth to Reionization τ from 21 cm Data Using Machine Learning Techniques". In: *Publications of the Astronomical Society of the Pacific* 133.1022, p. 044001. DOI: 10.1088/1538-3873/abe9a0.

Bodria, Francesco et al. (2023). "Benchmarking and survey of explanation methods for black box models: F. Bodria et al." In: *Data Mining and Knowledge Discovery* 37.5, pp. 1719–1778.

Braun, Robert et al. (May 2015). "Advancing Astrophysics with the Square Kilometre Array". In: *PoS*. AASKA14 AASKA14, p. 174. DOI: 10.22323/1.215.0174.

Choudhury, Madhurima, Abhirup Datta, and Suman Majumdar (Mar. 2022). "Extracting the 21-cm power spectrum and the reionization parameters from mock data sets using artificial neural networks". In: *Monthly Notices of the Royal Astronomical Society* 512.4, pp. 5010–5022. ISSN: 0035-8711. DOI: 10.1093/mnras/stac736.

DeBoer, David R. et al. (Mar. 2017). "Hydrogen Epoch of Reionization Array (HERA)". In: *Publications of the Astronomical Society of the Pacific* 129.974, p. 045001. DOI: 10.1088/1538-3873/129/974/045001.

Doussot, Aristide, Evan Eames, and Benoit Semelin (Sept. 2019). "Improved supervised learning methods for EoR parameters reconstruction". In: *Monthly Notices of the Royal Astronomical Society* 490.1, pp. 371–384. ISSN: 1365-2966. DOI: 10.1093/mnras/stz2429.

Dwivedi, Rudresh et al. (Jan. 2023). "Explainable AI (XAI): Core Ideas, Techniques, and Solutions". In: *ACM Comput. Surv.* 55.9. ISSN: 0360-0300. DOI: 10.1145/3561048. URL: https://doi.org/10.1145/3561048.

Fan, Xiaohui et al. (June 2006). "Constraining the Evolution of the Ionizing Background and the Epoch of Reionization with z 6 Quasars. II. A Sample of 19 Quasars". In: *The Astronomical Journal* 132.1, p. 117. DOI: 10.1086/504836. URL: https://doi.org/10.1086/504836.

Gillet, Nicolas et al. (Jan. 2019). "Deep learning from 21-cm tomography of the Cosmic Dawn and Reionization". In: *Monthly Notices of the Royal Astronomical Society* 484.1, pp. 282–293. ISSN: 1365-2966. DOI: 10.1093/mnras/stz010.

Gnedin, Nickolay Y. and Piero Madau (Nov. 2022). "Modeling cosmic reionization". In: *Living Reviews in Computational Astrophysics* 8.1, p. 3. ISSN: 2365-0524. DOI: 10.1007/s41115-022-00015-5.

Greig, Bradley and Andrei Mesinger (2015). "21CMMC: an MCMC analysis tool enabling astrophysical parameter studies of the cosmic 21 cm signal". In: *Monthly Notices of the Royal Astronomical Society* 449.4, pp. 4246–4263. DOI: 10.1093/mnras/stv571.

Guidotti, Riccardo (Apr. 2024). "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Mining and Knowledge Discovery* 38.5, pp. 2770–2824. ISSN: 1573-756X. DOI: 10.1007/s10618-022-00831-6.

Haarlem, M. P. van et al. (July 2013). "LOFAR: The LOw-Frequency ARray". In: *Astronomy & Astrophysics* 556, A2. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201220873.

Hassan, Sultan, Sambatra Andrianomena, and Caitlin Doughty (2020). "Constraining the astrophysics and cosmology from 21 cm tomography using deep learning with the SKA". In: *Monthly Notices of the Royal Astronomical Society* 494.4, pp. 5761–5774. DOI: 10.1093/mnras/staa1151.

Hutter, Anne (Mar. 2018). "The accuracy of seminumerical reionization models in comparison with radiative transfer simulations". In: *Monthly Notices of the Royal Astronomical Society* 477.2, pp. 1549–1566. ISSN: 0035-8711. DOI: 10.1093/mnras/sty683. eprint: https://academic.oup.com/mnras/article-pdf/477/2/1549/25009549/sty683.pdf. URL: https://doi.org/10.1093/mnras/sty683.

Iliev, Ilian T. et al. (2014). "Simulating cosmic reionization: how large a volume is large enough?" In: *Monthly Notices of the Royal Astronomical Society* 439.1, pp. 725–743. DOI: 10.1093/mnras/stt2497.

Kaur, Harman Deep, Nicolas Gillet, and Andrei Mesinger (May 2020). "Minimum size of 21-cm simulations". In: *Monthly Notices of the Royal Astronomical Society* 495.2, pp. 2354–2362. ISSN: 0035-8711. DOI: 10.1093/mnras/staa1323.

Koopmans, Leon et al. (May 2015). "The Cosmic Dawn and Epoch of Reionisation with SKA". In: *Proceedings of Advancing Astrophysics with the Square Kilometre Array — PoS(AASKA14)*. AASKA14. Sissa Medialab. DOI: 10.22323/1.215.0001.

Kwon, Yungi, Sungwook E. Hong, and Inkyu Park (2020). "Deep-learning study of the 21-cm differential brightness temperature during the epoch of reionization". In: *Journal of the Korean Physical Society* 77.1, pp. 49–59. DOI: 10.3938/jkps.77.49.

La Plante, Paul and Michelle Ntampaka (July 2019). "Machine Learning Applied to the Reionization History of the Universe in the 21 cm Signal". In: *The Astrophysical Journal Letters* 880.2, p. 110. ISSN: 1538-4357. DOI: 10.3847/1538-4357/ab2983.

Lahiry, Arnab, Adrian E. Bayer, and Francisco Villaescusa-Navarro (Nov. 2025). "Interpreting Cosmological Information from Neural Networks in the Hydrodynamic Universe". In: *The Astrophysical Journal* 994.1, p. 129. ISSN: 1538-4357. DOI: 10.3847/1538-4357/ae0b5a. URL: http://dx.doi.org/10.3847/1538-4357/ae0b5a.

Liu, Adrian and J. Richard Shaw (Apr. 2020). "Data Analysis for Precision 21 cm Cosmology". In: *Publications of the Astronomical Society of the Pacific* 132.1012, p. 062001. DOI: 10.1088/1538-3873/ab5bfd. URL: https://doi.org/10.1088/1538-3873/ab5bfd.

Loeb, A. and S.R. Furlanetto (2013). *The First Galaxies in the Universe*. Ed. by Steven R. Furlanetto. Princeton Series in Astrophysics. Princeton: Princeton University Press. 154016 pp. ISBN: 9781400845606.

Majumdar, Suman et al. (2018). "Quantifying the non-Gaussianity in the EoR 21-cm signal through bispectrum". In: *Monthly Notices of the Royal Astronomical Society* 476.3, pp. 4007–4024. DOI: 10.1093/mnras/sty535.

Mertens, Florent G., Jérôme Bobin, and Isabella P. Carucci (Nov. 2023). "Retrieving the 21-cm signal from the Epoch of Reionization with learnt Gaussian process kernels". In: *Monthly Notices of the Royal Astronomical Society* 527.2, pp. 3517–3531. ISSN: 0035-8711. DOI: 10.1093/mnras/stad3430.

Mertens, F. G. et al. (June 2025). "Deeper multi-redshift upper limits on the epoch of reionisation 21 cm signal power spectrum from LOFAR between z = 8.3 and z = 10.1". In: *Astronomy & Astrophysics* 698, A186. ISSN: 1432-0746. DOI: 10.1051/0004-6361/202554158.

Mesinger, Andrei, Steven Furlanetto, and Renyue Cen (Feb. 2011). "21cmfast: a fast, seminumerical simulation of the high-redshift 21-cm signal". In: *Monthly Notices of the Royal Astronomical Society* 411.2, pp. 955–972. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2010.17731.x.

Munshi, S. et al. (Aug. 2025). "Improved upper limits on the 21-cm signal power spectrum at z = 17.0 and z = 20.3 from an optimal field observed with NenuFAR". In: *Monthly Notices of the Royal Astronomical Society* 542.4, pp. 2785–2807. ISSN: 0035-8711. DOI: 10.1093/mnras/staf1386.

Murray, Steven G. et al. (Oct. 2020). "21cmFAST v3: A Python-integrated C code for generating 3D realizations of the cosmic 21cm signal." In: *The Journal of Open Source Software* 5.54, p. 2582. DOI: 10.21105/joss.02582.

Neutsch, Steffen, Caroline Heneka, and Marcus Brüggen (2022). "Inferring astrophysics and dark matter properties from 21 cm tomography using deep learning". In: *Monthly Notices of the Royal Astronomical Society* 511.3, pp. 3446–3462. DOI: 10.1093/mnras/stac218.

Park, Jaehong et al. (2019). "Inferring the astrophysics of reionization and cosmic dawn from galaxy luminosity functions and the 21-cm signal". In: *Monthly Notices of the Royal Astronomical Society* 484.1, pp. 933–949. DOI: 10.1093/mnras/stz032.

Prelogović, David et al. (Nov. 2021). "Machine learning astrophysics from 21cm lightcones: impact of network architectures and signal contamination". In: *Monthly Notices of the Royal Astronomical Society* 509.3, pp. 3852–3867. ISSN: 0035-8711. DOI: 10.1093/mnras/stab3215.

Shimabukuro, Hayato, Yi Mao, and Jianrong Tan (2022). "Estimation of H ii Bubble Size Distribution from 21 cm Power Spectrum with Artificial Neural Networks". In: *Research in Astronomy and Astrophysics* 22.3, p. 035027. DOI: 10.1088/1674-4527/ac4ca3.

Shimabukuro, Hayato and Benoit Semelin (2017). "Analysing the 21 cm signal from the epoch of reionization with artificial neural networks". In: *Monthly Notices of the Royal Astronomical Society* 468.4, pp. 3869–3877. DOI: 10.1093/mnras/stx734.

Shimabukuro, Hayato, Shintaro Yoshiura, et al. (2016). "21 cm line bispectrum as a method to probe cosmic dawn and epoch of reionization". In: *Monthly Notices of the Royal Astronomical Society* 458.3, pp. 3003–3011. DOI: 10.1093/mnras/stw482.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2014). *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. arXiv: 1312.6034 [cs.CV]. URL: https://arxiv.org/abs/1312.6034.

Sooknunan, Kimeel et al. (Dec. 2024). "Reproducibility of machine learning analyses of 21 cm reionization maps". In: *arXiv preprint arXiv:2412.15893*. DOI: 10.48550/ARXIV.2412.15893.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). *Axiomatic Attribution for Deep Networks*. arXiv: 1703.01365 [cs.LG]. URL: https://arxiv.org/abs/1703.01365.

Tingay, S. J. et al. (2013). "The Murchison Widefield Array: The Square Kilometre Array Precursor at Low Radio Frequencies". In: *Publications of the Astronomical Society of Australia* 30, e007. ISSN: 1448-6083. DOI: 10.1017/pasa.2012.007.

Tiwari, Himanshu et al. (Apr. 2022). "Improving constraints on the reionization parameters using 21-cm bispectrum". In: *Journal of Cosmology and Astroparticle Physics* 2022.04, p. 045. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2022/04/045.

Zha, Daochen et al. (Jan. 2025). "Data-centric Artificial Intelligence: A Survey". In: *ACM Comput. Surv.* 57.5. ISSN: 0360-0300. DOI: 10.1145/3711118.

Zhao, Xiaosheng et al. (Feb. 2022). "Simulation-based Inference of Reionization Parameters from 3D Tomographic 21 cm Light-cone Images". In: *The Astrophysical Journal* 926.2, p. 151. DOI: 10.3847/1538-4357/ac457d.

Zhou, Yihao and Paul La Plante (Apr. 2022). "Understanding the Impact of Semi-numeric Reionization Models when Using CNNs". In: *Publications of the Astronomical Society of the Pacific* 134.1034, p. 044001. ISSN: 1538-3873. DOI: 10.1088/1538-3873/ac5f5d.