Computer Vision Analysis of Reionization

Jasper Solt

April 19, 2021

Contents

I Reionization **3**

1 Introduction to Reionization **3**

2 The Ionization of the IGM **3**

3 The Dynamics of Reionization **8**

4 Data Collection **9**

4.1 21cm Signal 9

4.2 Kinetic Sunyaev-Zeldovich Effect 10

5 Numerical & Semi-Numerical Simulations **12**

6 Conclusion **13**

II Machine Learning **15**

7 Introduction to Machine Learning **15**

8 Training, Testing, and Validation Data **15**

9 Neural Nets **16**

10 Convolutional Neural Nets **18**

11 Hyperparameter Optimization **21**

12 Other Considerations when Using ML **22**

III Training a Model with Reionization Data **24**

13 Applying Machine Learning Techniques to Cosmological Data **24**

14 Data **26**

15 Model Architecture **27**

IV Model Performance **29**

16 Model Performance on Full Data Set **29**

16.1 21cm slices 29

16.2 kSZ snapshot 29

16.3 21cm slices + normalized kSZ channel 32

17 Model Performance on Data Subsets **34**

17.1 Training on fewer 21cm channels 34

17.2 Reduction of Redshift Range 36

17.3 Addition of kSZ as channel 37

V Discussion **38**

18 Data Processing Choices **38**

19 Model Architecture Choices **39**

20 Alternate model structures **39**

VI Conclusion **40**

Part I

Reionization

1 Introduction to Reionization

The dark voids that lie between galaxies seem unassuming at first glance, but these spaces are filled with interesting phenomena: the Intergalactic Medium, a low-density plasma of ionized hydrogen, permeates intergalactic space, and contains clues about the evolution of our universe. Our observations have made it clear that the IGM underwent a major phase change sometime between recombination and now, going from entirely neutral to entirely ionized. This reionization process hinges on the physics of the earliest baryonic structures, and is therefore just as mysterious. Observing the 21cm line of neutral hydrogen promises rich physical insight into reionization, but this is made difficult by the low signal-to-noise ratio of the data. In this chapter we discuss physical evidence for reionization (Section 2); the dynamics of reionization (Section 3); methods for collecting data on reionization (Section 4); and the role of semi-numerical simulations in reionization research (Section 5).

2 The Ionization of the IGM

Most baryons exist as gaseous hydrogen outside galaxies, in a material called the Intergalactic Medium (IGM). Structurally, this gas exists as a "cosmic web" of high-density filaments surrounded by voids. Therefore, the density of the IGM is highly variable. Crucially, because the IGM exists across intergalactic space, photons emitted from distant sources can scatter off of or be absorbed by particles within the IGM. Therefore, the optical depth of light from faraway quasars or the CMB itself can provide clues about the structure, ionization, and density of the IGM.

Current evidence suggests that the IGM is almost entirely ionized. However, this was not always the case. In the hot, dense, early universe, high energy photons suppressed

recombination of protons and electrons. As the universe expanded, the Hubble parameter eventually surpassed the photon scattering rate, decoupling photons from electrons. The remnant of this process can be seen in the Cosmic Microwave Background (CMB), an imprint of the photons' surface of last scattering. This decoupling allowed electrons and protons to combine into neutral Hydrogen in a process known as recombination, effectively resulting in a neutral IGM. Therefore, between recombination and now, some process must have occurred that resulted in the full *re*-ionization of the IGM. What can we determine about this process?

The premier way of observing the IGM at low redshifts is the Lyman-α forest. The Lyman-α transition occurs when a hydrogen atom moves from its base electronic state to its first excited state. When photons with Lyman-α resonant wavelengths hit a neutral hydrogen gas, they will be absorbed, exciting the atoms of the gas and removing those wavelengths from the spectrum. As photons continue to travel through this gas, space expands and light is redshifted by a factor of (z + 1), causing the spectrum to shift redward and new absorption lines to appear at the Lyman-α resonances. In this way, the spectra of distant objects provide a map of neutral hydrogen in the universe.

If the neutral fraction of hydrogen x_{HI} is over some threshold, unabsorbed photons (i.e., those not at Lyman-α resonant wavelength) will be redshifted by the expansion of the universe into Lyman-α resonant wavelengths, broadening the range of wavelengths absorbed by the gas they travel through. This broadens the spectral lines and creates a "trough" of absorption in the spectrum, dubbed the Gunn-Peterson (GP) trough. Because the absorption cross section is very high, the threshold x_{HI} necessary for the GP trough to appear is very small. Analytically, this can be computed via the Lyman-α optical depth, which is given by [9]:

$$\tau_\alpha(\nu_0) = \int_O^S \sigma_\alpha(\nu) n_{HI} dl/(1+z) \tag{1}$$

Where dl is the comoving distance from the observer to the distant photon source, ν_0 is the observed frequency, $\nu = \nu_0(1+z)$ is the frequency at dl, σ_α is the Lyman-α cross

section, n_{HI} is the neutral fraction, and O and S are the locations of the observer and source, respectively. This can be re-written using the FLRW metric as:

$$\tau_\alpha(\nu_0) = \int \sigma_\alpha(\nu) n_{HI} \frac{cH_0^{-1}dz}{(1+z)\sqrt{\Omega_m(1+z)^3+\Omega_\Lambda}} \tag{2}$$

Since the neutral fraction x_{HI} is defined as $x_{HI} = \frac{n_{HI}}{n_H}$, solving the integral we find that

$$x_{HI} \approx 10^{-4}\Omega_m^{1/2} h (1+z)^{3/2} \tau_\alpha \tag{3}$$

The GP trough appears when the average $\tau_\alpha \geq 1$, which Equation 3 tells us corresponds to $x_{HI} \approx 10^{-4}$. Therefore, when the GP trough appears gives a good estimate as to when the neutral fraction of the IGM is very low, and therefore when reionization ended. Looking at Lyman-α data, we see that the trough begins to appear at around $z = 6$ (Figure 1). Therefore, at $z < 6$, the IGM must be almost 100% ionized.

Data from the CMB also gives evidence for a highly ionized IGM at low redshifts. Contrary to the Lyman-α photons, whose optical depth is related to the number of neutral hydrogen atoms, CMB photons are scattered by free electrons, distorting the CMB on small angular scales. Therefore, the optical depth of the CMB is related to the number of ionized hydrogen atoms. For a flat universe where $\Omega_\Lambda + \Omega_m = 1$, they are related by [9]:

$$\tau_{es} = \int n_e(z)\sigma_T(cdt/dz)dz \tag{4}$$

Where τ_{es} is the electron scattering optical length, $n_e(z)$ is the number of free electrons and $\sigma_T = 6.65 \times 10^{-25} cm^2$ is the Thomson scattering cross section. Assuming reionization happened instantaneously (an erroneous assumption, as we shall explore later), the redshift of reionization z_{reion} can be calculated via the solution to the above integral:

$$\tau_{es} = 4.44 \times 10^{-3} \times (\sqrt{\Omega_\Lambda + \Omega_m(1+z_{reion})^3}) - 1) \tag{5}$$

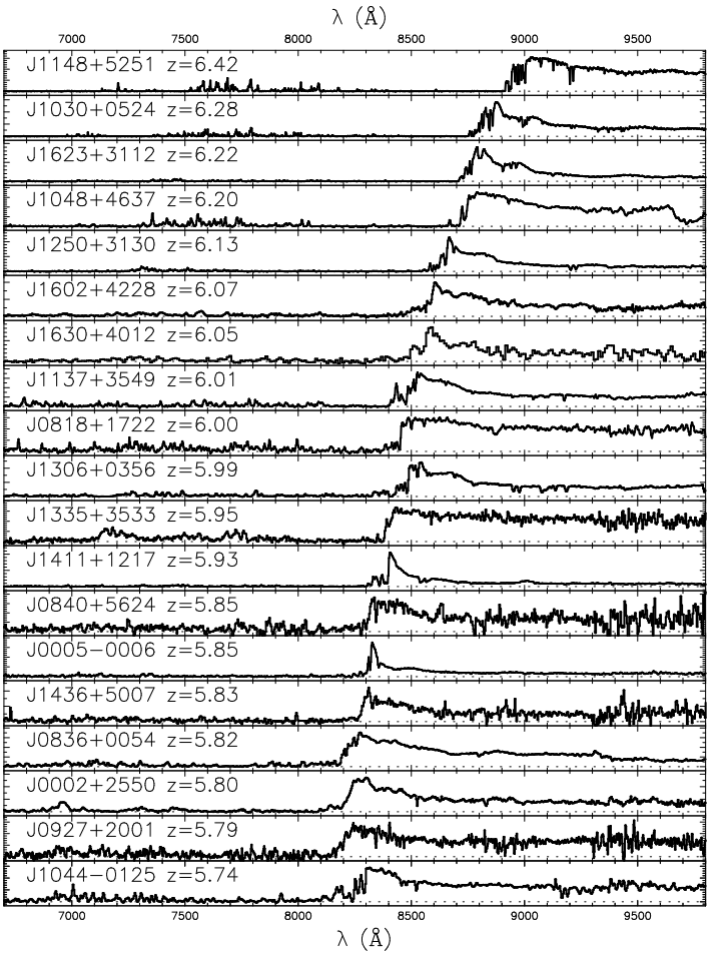


Figure 1: Lyman-α forest for 19 quasars, at redshifts from 5.74 to 6.42. Signal suppression can be seen blueward of the Lyman-α line in the forest of the quasars around $z = 6$, indicating the presence of a Gunn-peterson trough. The lack of a GP trough in lower redshifts is evidence that after $z \approx 6$, the universe must be fully ionized. [5]

Intuitively, then, earlier reionization scenarios correspond to bigger angular-scale distortions of the CMB. With any $\tau_{ES} \geq 1$, the CMB would be completely scattered by the IGM and therefore be undetectable; given that we *have* detected the CMB signal, with distortions occurring only on small angular scales, hints that reionization happened relatively recently [11].

The polarization of the CMB contains even more detailed information about the electron scattering optical depth of the CMB, and therefore about the redshift of reionization. When an observer detects a photon that has scattered off an electron, only polarization that is

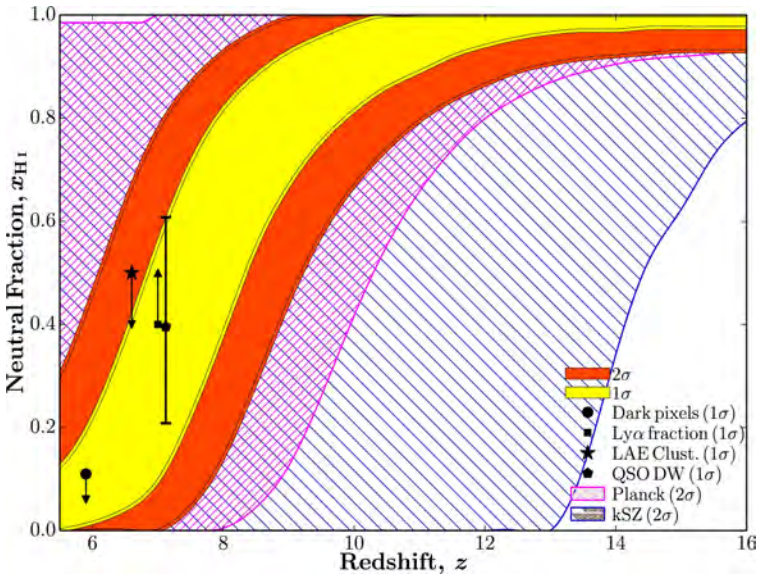


Figure 2: The current constraints on the evolution of reionization, as modelled by the neutral fraction as a function of redshift. The black points are a combination of Lyman-α and high-redshift galaxy observations, the hatched regions are CMB constraints, and the solid color contours are best-fit reionization models given all current available constraints. [6]

perpendicular to the plane created by the source, observer, and scattering electron reaches that observer. This is because the polarization parallel to that same plane cannot form a transverse wave travelling on the path towards the observer. If the CMB were perfectly isotropic, the same number of photons would come from every direction, and thus result in perfectly unpolarized light. However, the CMB is anisotropic at small angular scales; therefore, the polarization of the incoming photons will be biased based on the anisotropy of the CMB and the density of free electrons between the observer and source. Therefore, given the CMB polarization and anisotropy, one can estimate the redshift of reionization.

Using anisotropy data from the PLANCK satellite leads to $\tau_{EM} = 0.058 \pm 0.012 (1\sigma)$ [1], which results in $z_{reion} \approx 7.7$. This calculation, however, hinges on the false assumption that reionization occurred instantaneously; in fact, reionization scenarios consistent with our physical understanding of energetic transfer between the IGM and ionization background predict that reionization took place over a range of redshifts, a duration that is at present relatively poorly constrained (See Figure 2).

3 The Dynamics of Reionization

After the CMB was too cold to keep the IGM ionized, what energy source eventually took its place suppressing IGM recombination? The answer is intuitive: reionization coincides with the "Cosmic Dawn" when the first baryonic structures such as stars began emitting light. Photons with energy $hf \geq 13.6$ eV can ionize hydrogen atoms. Two sources of such photons were present during the Cosmic Dawn: early galaxies containing bright O stars, and Active Galactic Nuclei (AGNs). O stars with masses $M \geq 30M_{\odot}$ are abundant during this era, with 400 per comoving cubic parsec at any time, and can produce up to 10^{48} ionizing photons per second. The oldest galaxies ever detected were found around $z \approx 10$, which fits the timeline of reionization nicely. Conversely, AGNs produce orders of magnitude more ionizing photons (at approximately $10^{56}s^{-1}$) but are exceedingly rare, especially at $z > 6$ [11]. These sources make up an "ionizing background" of baryonic structures.

Therefore, reionization can be modelled as a patchy "inside-out" phase change, where ionization fronts balloon out from high density regions in the ionizing background, creating growing bubbles of ionized material. However, the exact hydrodynamics of this phase change are not fully understood. The IGM is not uniformly dense: its web-like structure creates areas of high and low density, which affect the rate and efficiency of ionization in complex ways. Spontaneous recombination also complicates reionization models because ionization fronts propagate outward in an equilibrium, settling in a radius where recombination and ionization are balanced, the exact mechanics behind which are complex and not fully understood. What fraction of ionizing photons escape from sources within galaxies, and therefore how much ionizing power any source can have, is also not well known and highly dependent upon the structure of the individual galaxies. Feedback processes between the ionization background and IGM also require more investigation [9]. Ultimately, with the current data and theory, we don't know enough about early structure formation at this point to narrow down all possible reionization scenarios. As a result research has focused on constraining the parameters

of reionization rather than fitting to any particular model: see Figure 2 for our current best bounds. As constraints improve, however, extracting the properties of sources driving reionization will become increasingly important.

4 Data Collection

What is the best way to gather more data on reionization? The Lyman-α optical depth is opaque above $z = 6$, and therefore tells us only when reionization stopped. To probe further back in time, we will need a different method.

4.1 21cm Signal

The 21cm signal promises to contain a wealth of information about reionization. The base state of neutral hydrogen is split into two "hyper-fine" states that differ slightly in energy: a parallel state, where the spins of the electron and proton are aligned, and an anti-parallel state, where they are opposite. When a ground-state neutral hydrogen atom transitions from its parallel to anti-parallel state, it releases a photon with wavelength $\lambda = 21$cm. While this transition is rare, the sheer abundance of hydrogen in the universe means this transition occurs frequently per unit volume of space. Therefore, the 21cm signal acts as a tracer for neutral hydrogen. Additionally, because the signal has such low energy, its optical depth is very low for almost all mediums through which it could travel, making it ideal for studying the murky eras of reionization where the Lyman-α signal cannot penetrate. The signal redshifts with the expansion of space, and so the whole timeline of reionization can be seen in the signal's spectrum. Therefore, in theory, one could obtain the full history of $x_{HI}(z)$ from the 21cm signal.

However, 21cm observation comes with its own difficulties, and the signal has not yet been recovered. The main challenge stems from the low signal-to-noise ratio of the data. Radio sources between us and the IGM in the bandwidth of the signal, referred to broadly as foreground emissions, are of order 5 orders of magnitude more intense than the 21cm

signal. Additionally, the radio bandwidth the redshifted 21cm signal inhabits is well-tread by human narrow and broad-band radio, cell phone, and TV transmissions, scarring ground-based observations with unusable lines of noise. The unique hardware challenges involved in radio interferometry also produces artifacts in the data that are non-trivial to remove.

Radio interferometry uses correlated pairs of ground-based antenna to produce an interference pattern from the sky. The interference pattern from each baseline pair corresponds to a Fourier mode at angular scales proportional to the distance between the baseline pairs. Therefore, the resolution of the telescope is determined by the longest baseline pair, and the field of view (FOV) is determined by the size of each individual antenna. With enough baseline pairs with varying distances between them, enough Fourier modes can be collected and assembled to produce an image in Fourier space. This image can then be inversely Fourier transformed to produce an image of the sky. While this is the premier way of observing the low-frequency sky at high angular scales, collecting discreet Fourier modes to reassemble an image proves challenging in practice. Noise, foregrounds, and instrumental effects, which are already magnitudes higher intensity than the desired signal, become even more complex when transformed from Fourier space. Overall, these challenges have made observing the 21cm signal an uphill battle, one that has yet to be won.

4.2 Kinetic Sunyaev-Zeldovich Effect

While the 21cm signal is the most promising in terms of the quality and quantity of reionization data to be gained, other observational sources could hold clues about the parameters of reionization as well. One notable example is the Sunyaev-Zeldovich (SZ) effect. The SZ effect occurs when CMB photons inverse Compton scatter off of high energy electrons. The SZ effect can be split into two components: thermal (tSZ) and kinetic (kSZ) interactions. The tSZ effect occurs when high-temperature electrons transfer thermal energy into CMB radiation, giving the CMB a bias towards higher energies. The kSZ effect is a 2nd order effect where CMB photons are Doppler shifted when scattering off of bulk-moving free elec-

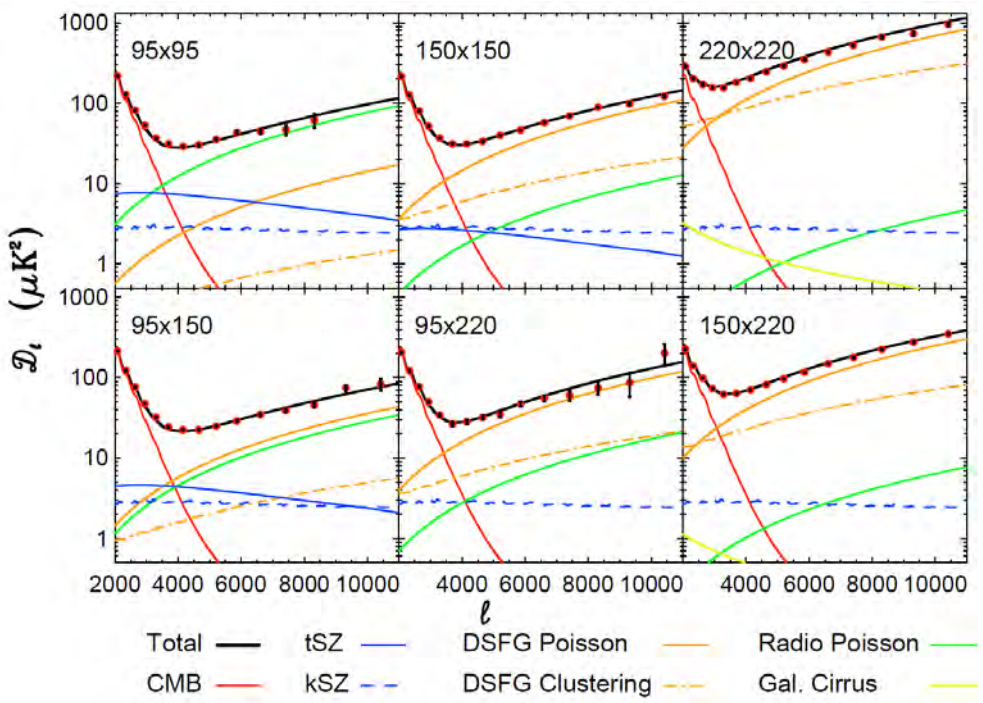


Figure 3: Auto- and cross-spectra from the South Pole Telescope (SPT) at 95, 150, and 220 GHz showing best fit lines that include the kSZ signal. This implies a $\Delta z_{reion} < 4.1$ with 95% confidence. While this does not improve upon previously determined constraints, it is notable because it comes from the kSZ signal, independent of previous observations. For a more in-depth explanation see Reichardt, 2020. [10]

trons. The kSZ effect is of notable interest for studying reionization due to the bulk motion of electrons expected during the propagation of the ionization fronts. Therefore, the kSZ effect is highly sensitive to the duration of reionization.

The benefit of the kSZ effect is that, unlike the 21cm signal, it has recently been detected (see Figure 3) and research is underway to reconstruct images of this effect from the CMB [2]. The drawback is that it contains much less information than the 21cm signal. Estimates of the duration of reionization from the recent measurement of the signal place $\Delta z_{reion} < 4.1$ with 95% confidence [10]. While this does not improve bounds from previous measurements, it is interesting because it comes from the kSZ signal, independent of previous measurements. It's possible that other methods of analysis, new constraints on the best-fit components of the spectra, and clearer kSZ images can yield even tighter bounds.

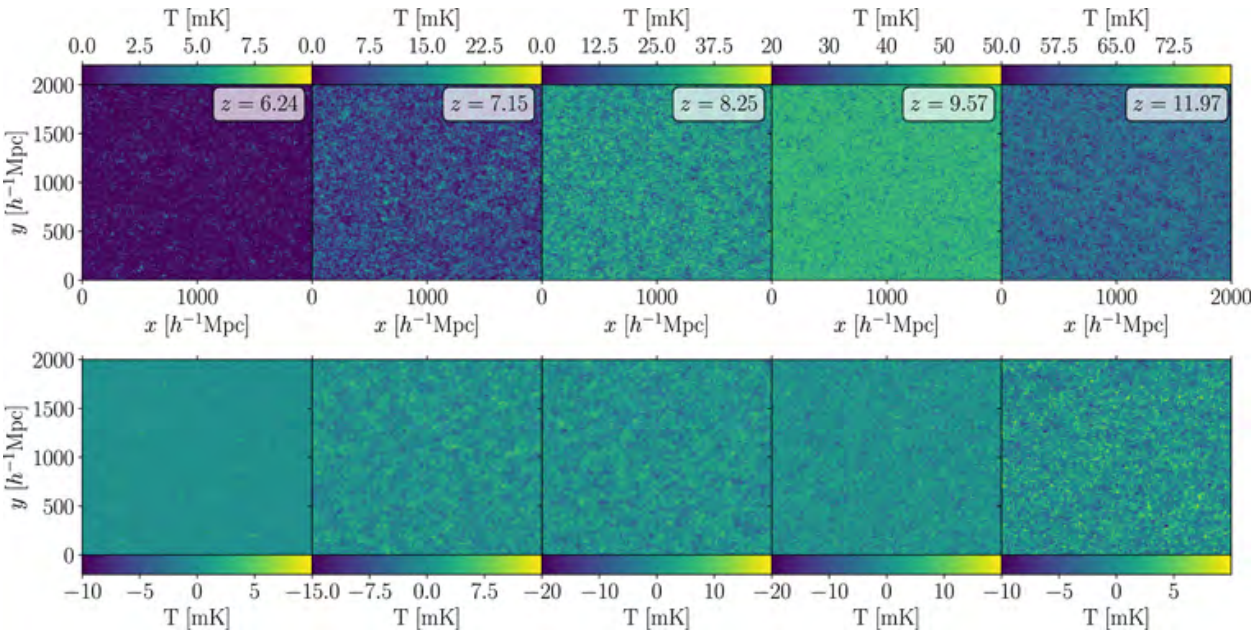


Figure 4: Semi-Numerical simulations of 21cm reionization data without modelled foregrounds (top) and with foreground effects applied (bottom). The reionization model used for this simulation is described in LaPlante, 2019. [8]

Because no measurements of the 21cm signal have been successful, data from simulations of reionization can be informative in developing data analysis pipelines for 21cm data and exploring the physics of IGM ionization. Full numerical simulations would include the hydrodynamics driving complex gas interactions in the IGM, gravitational dynamics that determine the density perturbations of baryonic matter, and the radiative transfer of energy between the ionizing background and IGM [9]. However, because the physics of early galaxies is still deeply uncertain, the predictions made by even the most robust simulation of reionization physics must be interpreted cautiously.

Furthermore, hydrodynamical simulations are computationally expensive, and therefore unfeasible for Gigaparsec-scale simulations on current hardware. Hydrodynamical insights can be gained from smaller-scale simulations of single ionization sources, but this tells us little about the large-scale feedback processes between the IGM and ionization background

that complicate reionization considerably.

Instead, semi-numerical simulations can be used as a low-overhead alternative to more dynamically rigorous "numerical" simulations. A semi-numerical simulation relies on mathematical approximations to mimic a more robust physical simulation, at the expense of some (hopefully small) degree of precision. The approximations and assumptions involved differentiate various semi-numerical models from one another. A common approach to semi-numerical reionization simulation is to create a Cold Dark Matter (CDM) density distribution at some redshift by linearly evolving from initial conditions, identify ionizing sources, then create dynamic ionized bubbles around the sources based on the number of ionizing photons they emit [9]. While this approach is naive in smoothing out some important physical interactions, it agrees well enough with more robust numerical simulations to be useful for research purposes. An example of a semi-numerical simulation of 21cm reionization data can be seen in Figure 4. This simulation contains two subsets of data: a raw 21cm signal, and a signal where Fourier modes commonly contaminated by foregrounds have been removed from the data. Analyzing how modelled foregrounds, noise, and instrumental effects change simulated data gives insight into how the true 21cm signal might be uncovered from the radio clamor that surrounds it.

6 Conclusion

Analysis of the Lyman-α forest and CMB polarization provides strong evidence that the IGM is almost completely ionized at $z < 6$. This is contrary to what one might expect, given that recombination left the IGM completely neutral; therefore, some process must have caused the IGM to ionize between the time of last scattering and now.

This reionization process is driven by an ionizing background of baryonic structures releasing photons energetic enough to strip electrons from neutral hydrogen. Fronts of ionization bubble outwards from these sources, creating a patchy "inside out" phase change. Because the physics of early baryonic structures are still poorly understood, the exact pa-

rameters of reionization are sketchy. The Lyman-α forest puts a strong bound on when reionization concluded, and the optical depth and polarization of the CMB also provides constraints on its timeframe. Direct observation of reionization via the 21cm signal has proved challenging due to the low signal-to-noise nature of the data, and to date the signal has not been recovered. Data from the kinetic Sunyaev-Zeldovich effect has recently been recovered, and provided some loose bounds on the duration of reionization. Due to the deficit of "real" 21cm data, researching reionization often hinges on analyzing simulated data, with modelled noise, foregrounds, and instrumental effects.

New telescopes are poised to probe the 21cm signal with unprecedented clarity in the near future. The Hydrogen Epoch of Reionization Array (HERA) is currently under construction, set to be complete late next year. HERA is order of magnitude more sensitive than previous telescopes, and therefore promises to yield great insight into reionization. The next few years could yield significant advances in our understanding of reionization, given HERA datasets and the ever-increasing computational power of new GPUs for simulations.

Data analysis tools are needed in order to regress on cosmological parameters from HERA data. Reionization depends on a tangled web of codependent physical processes such as star formation, supernova feedback, and black hole accretion; as a result there is a deficit in robust data analysis methods to extract information from the 21cm signal. However, one data analysis technique has proven itself highly capable in untangling the complex mechanics hiding behind messy data: Machine Learning (ML). In the following section this thesis will discuss how ML algorithms work, the advantages and disadvantages of using ML in cosmology, and how ML techniques can be applied to extract reionization information from the 21cm signal.

Part II

Machine Learning

7 Introduction to Machine Learning

In broad terms, a Machine Learning (ML) algorithm is an algorithm that learns from data. On a more granular level, ML algorithms perform empirical risk minimization via a gradient descent algorithm over many iterations to regress or classify on a set of data. ML algorithms can be thought of as performing statistics through iteration, adjusting their parameters at each stage to yield an answer with a smaller error function each time.

There are many different types of ML algorithms, each of which implements this concept in different ways, and each of which has unique strengths and weaknesses. In this project, a Convolutional Neural Net (CNN) was trained on semi-numerically simulated reionization data; therefore, the ML section of this paper will largely discuss Deep Neural Nets with convolutional layers.

8 Training, Testing, and Validation Data

A weakness of supervised ML algorithms (such as those used in this work) is that they require a labelled input dataset: a dataset that spans the domain of the data space, where the classification has already been performed. This presents a challenge when such data sets are not readily available; further discussion of this problem in relation to this project is continued in Section 14. This section will discuss what an ML algorithm gains from the training, validation, and testing subsets of its input.

As a general rule, the more input data you have, the more robust your model will be. Typically, the data will be split around 80:20 into training and testing sets; the training set is then further divided 80:20 into a training and validation set. Training data is the data that is iteratively regressed upon. Validation data is predicted on at each iteration: the resulting

percent error, or validation loss, is compared to the training loss (percent error of model when predicting on the data it was trained on). This comparison tells us, as a function of iteration, how accurately our model regresses on new data, versus data it has seen before. Finally, the testing dataset is what is ultimately predicted on to determine the model's final accuracy. Commonly, once a model's hyperparameters have been optimized (see Section 11), the validation dataset is done away with, in favor of a larger training dataset.

9 Neural Nets

Neural nets are made up of layers of neurons: linear functions that take in a collection of inputs, multiply over a weight vector, and apply an activation function. This is often described as mimicking how neurons fire in the human brain, but it can also be thought of as a very complicated functional decomposition of the underlying patterns of a dataset.

The input $a_{t,j}$ to a neuron $v_{t,j}$ at layer V_t for a given input vector $\mathbf{x}$ is calculated by the following recursive function:

$$a_{t+1,j}(\mathbf{x}) = \sum_{r:(v_{t-1,r},v_{t,j})\in E} \omega(v_{t-1,r}, v_{t,j}) o_{t-1,r} \tag{6}$$

where

$$o_{t-1,r} = \sigma(a_{t-1,r}(\mathbf{x})) \tag{7}$$

is the output of neuron $v_{t-1,r}$, $\omega((v_{t-1,r}, v_{t,j}))$ is the weight between the neuron $v_{t,j}$ and a neuron from the previous layer $v_{t-1,r}$, E is the set of all indices of neurons in layer V_{t-1} connected to $v_{t,j}$, and σ is the activation function. Restated, each neuron receives the output of all neurons connected to it, sums them, then applies an activation function before passing its signal further down the network.

For these weights to regress on the input data, the model needs to adjust them at each iteration. This is done through a back-propagation algorithm that calculates the change δ_t, i

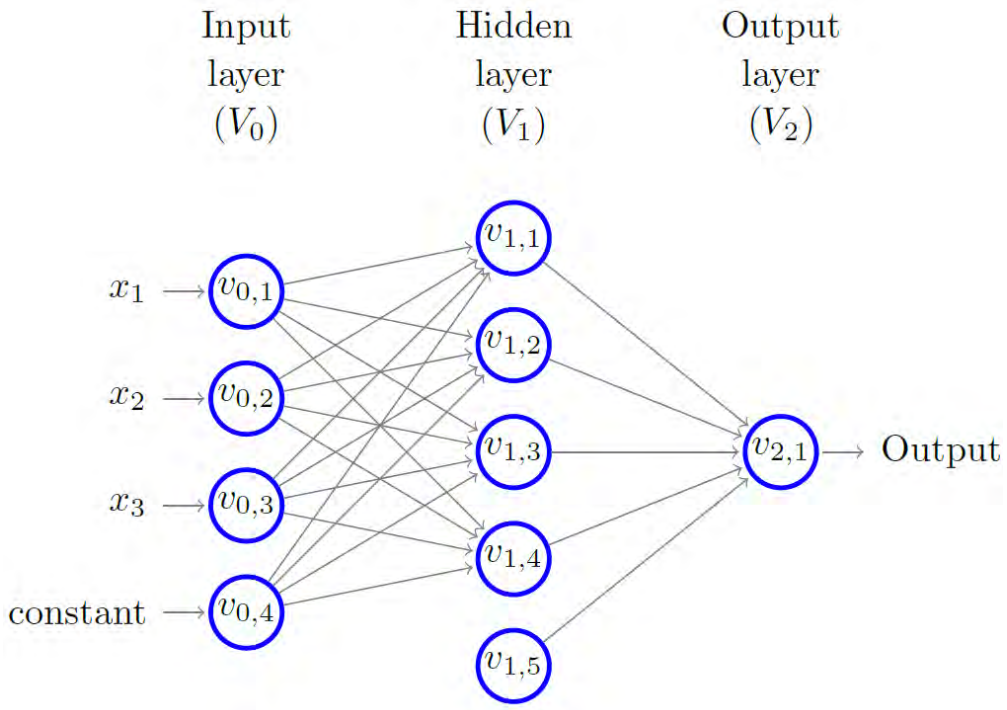


Figure 5: An example Dense Neural Net with depth 2, size 10, and width 5. This NN is "Dense" because each neuron in a layer is connected to each neuron in the adjacent layers. $v_{0,4}$ and $v_{1,5}$ are constant bias terms which do not receive input. [12]

of each weight. These δs can be found by working recursively backwards through the layers via the chain rule:

$$\delta_{t,i} = \sum_{i:(v_{t+1,j},v_{t,i})\in E} \omega(v_{t+1,j}, v_{t,i})\delta_{t+1,j}\sigma'(a_{t+1,j}) \tag{8}$$

where σ' is the derivative of the activation function. The gradient vector $\vec{g}$ can then be calculated element-wise:

$$g(v_{t-1,j}, v_{t,i}) = \delta_{t,i}\sigma'(a_{t,i})o_{t-1,j} \tag{9}$$

The weight vector is then updated via gradient descent:

$$\vec{\omega}_{new} = \vec{\omega} - \eta(\vec{g} + \lambda\vec{\omega})$$

(10)

where η is the learning rate (typically small, to the order of 10^{-2} to 10^{-4}) and λ is a regularization parameter. This algorithm iterates until a certain threshold is reached, either a maximum number of iterations or the results fall below a given percent error [12]. In this way, an ML algorithm can iteratively regress on a dataset.

Oftentimes ML is characterized as a "magic bullet" to solve any data problem; this characterization is unfortunate, and only serves to obfuscate the statistical reasoning ML algorithms use to regress on data. Equations 6, 8, and 10 remind us that ML algorithms are decidedly unmagical: they simply perform a functional decomposition of some underlying property shared among the input dataset, albeit a very complicated and opaque one. Keeping this statistical perspective is crucial for understanding how an ML algorithm learns from a given input, and for understanding what ML algorithms can and cannot detect.

10 Convolutional Neural Nets

Section 9 presents an introduction to NNs as a category. NNs can, however, be split into further groups based on their architecture and typical use cases.

The most common type of Neural Net is a Dense Neural Net (DNN). These networks are "dense" because each neuron in layer V_t receives input from every neuron in layer V_{t-1}. However, these models are not optimal for regressing on image data. Firstly, DNNs do not record spatial information. While each neuron is connected to every neuron in the previous layer, the neurons within a layer do not communicate. This can result in a loss of spatial information. Secondly, DNNs require huge memory resources for large data objects. Given an input image cube of size $h \times w \times c$ (height, width, and number of color channels), each neuron in the subsequent layer is connected to $h \times w \times c$ input neurons. If your layer has n

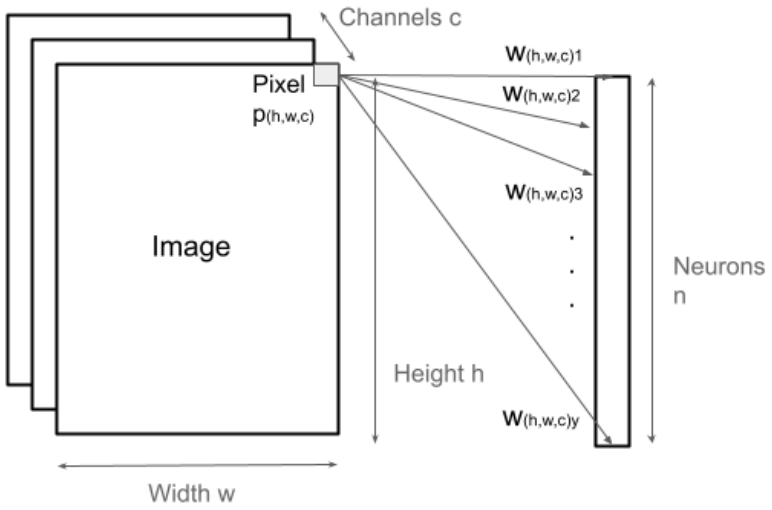
neurons, that results in $h \times w \times c \times n$ weights in the input layer. As an example, this project uses input cubes of size $256 \times 256 \times 30$ (channels are redshift), the input layer will have almost 2×10^6 weights per neuron. Given that hidden neuron layers typically have anywhere from 10 to hundreds of neurons, depending on the complexity of the data and choices made in hyperparameter optimization, a DNN with this size input could potentially hold hundreds of millions of trainable parameters. This is important, both for the speed of model training and for the ultimate memory usage of the finished model.

Therefore, an alternative model structure has been developed for NN image processing: the Convolutional Neural Net (CNN). Instead of having a weight for each pair of neurons in adjacent layers, a convolutional neuron layer uses kernels of weights the convolve over the layer. The output of these convolutions is then passed through the activation function and passed along to the next layer.

Kernels can be characterized by a characteristic size x, which is typically odd-valued to convolve symmetrically (commonly, $x = 3$). There are two types of convolutional layers commonly used in CNNs: 2D layers and 3D layers. A 2D layer performs 2D convolutions, where each color channel of an input image is convolved over its height and width by an $x \times x$ slice of the kernel (therefore, each kernel contains $x \times x \times c$ parameters). A 3D layer performs 3D convolutions, where an $x \times x \times x$ kernel convolves over the height, width, and channels of the image. 2D kernels consider spatial information from the height and width dimensions, but not the channel dimension; 3D kernels consider spatial information from every dimension. Therefore, 2D convolutions don't care what order the input channels are in, so long as each data cube in the set is ordered in the same way; 3D convolutions treat channels as being ordered, either in time or space. In this project exclusively 2D convolutions are used.

CNNs solve both problems DNNs have with image analysis. Both 2D and 3D kernels consider spatial information: the greater the characteristic dimension x of the kernel, the larger the scale considered. Convolutional layers also use less memory: A typical convolutional

Dense Layer:



2D Convolutional Layer:

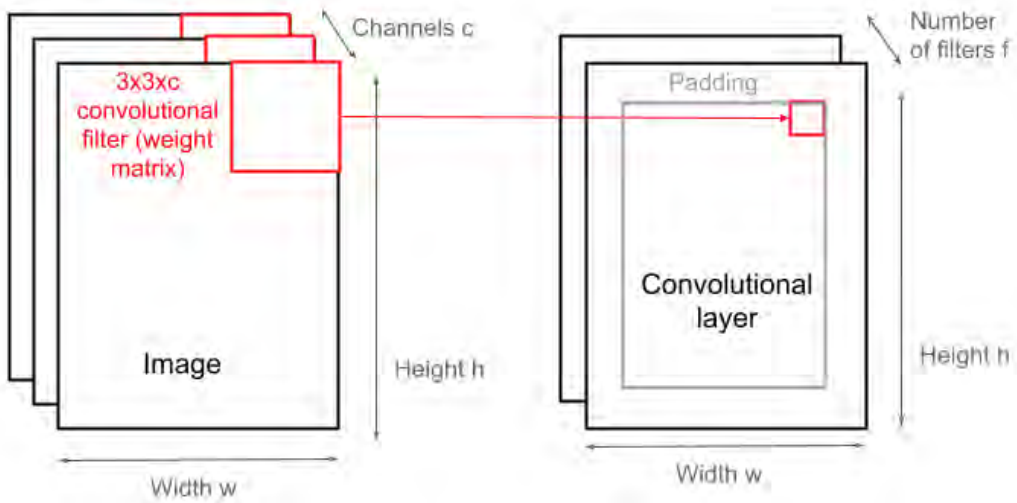


Figure 6: A visualization of the difference between Dense and 2D Convolutional neural layers.

layer has anywhere from 10 - 1000 filters: Therefore, there are an estimated $f \times 3 \times 3 \times c$ weights per layer. For this project the maximum $f \times c$ for any layer is $256 \times 128 \approx 3 \times 10^4$, so a rough estimate for the maximum possible weights per layer is 3×10^5. This is an order of magnitude smaller than the approximately 2×10^6 weights per layer for a DNN analyzing the same data.

Given that there are fewer trainable parameters (weights), CNNs can take longer to train or have higher prediction error than DNNs. However, these problems can be eliminated with proper hyperparameter optimization, discussed in Section 11.

11 Hyperparameter Optimization

Many coding libraries have been developed to handle the mathematical backend of ML algorithms. For example, this project uses Keras [3], an API of Tensorflow (a Python package of ML tools) to handle Equations 6-10 behind the scenes. Because of these libraries, building and training models becomes a trivial task. Therefore, the bulk of work in building ML models is in optimizing the model's hyperparameters.

The hyperparameters of an ML model refer to the statistical scaffolding upon which the neurons are hung. This includes the size and number of layers, learning rate η, batch size (how many samples are trained upon at once), activation function, number of iterations, etc. Hyperparameter optimization is considered more art than science: though understanding how each hyperparameter affects the model can give the designer an intuition for what choices to make, optimization is ultimately a trial and error process, heavily influenced by the properties of the input data. As a result there's no methodological way of determining whether one has designed the "best" model for a given dataset. This is challenging because it is unclear what is the maximum amount of information (i.e., minimum possible standard deviation of prediction) that a NN can extract from a given dataset. For a perfect dataset, the optimal model would perform with 0% prediction error. However, even with optimal hyperparameters and infinite training data, real-world datasets will still have some non-zero

prediction error due to noise and domain variation within the input data.

This matters because, when doing research with ML, there's always room for improvement: there's no way to prove your model is the "best," only that it performs well relative to some benchmark.

12 Other Considerations when Using ML

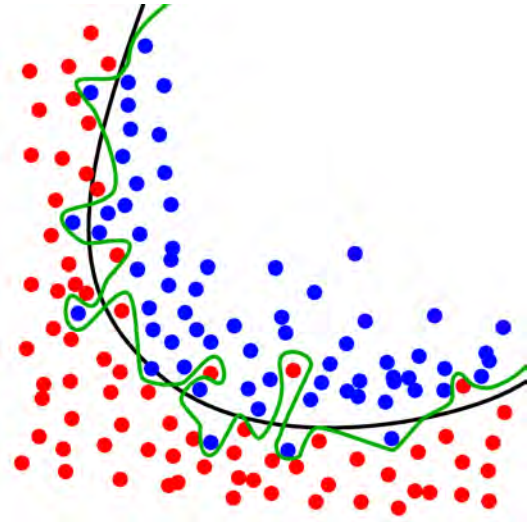


Figure 7: An example of overfitting. The black line represents the correct underlying distribution, and the green line represents the overfitting model. [4]

A classic problem ML algorithms encounter is called overfitting. Overfitting occurs when a model loses its ability to generalize, falsely classifying natural variation in the input data as part of the dataset's underlying distribution. A visual example can be found in Figure 7: in this case, the overfitting model (shown in green) creates an artificially complex boundary in order to perfectly categorize the training data. However, if we were to increase the number of data points n to the limit $n \to \infty$, the prediction error of the green classifier will be higher than the prediction error of the black classifier. Overfitting is indicated by low training error but high testing error: the model performs well on data it has seen before, but cannot generalize. This can be observed from the loss curve, a plot of prediction error as a function of training epoch, of the model. A loss curve that shows the training loss decreasing or

stagnating while validation loss begins to increase may indicate overfitting has occurred.

Overfitting indicates that the model is too complex; i.e., the model has too many trainable parameters. Overfitting can be mitigated in various ways: decreasing the number and size of layers, or reducing the number of training iterations, or introducing layers that strategically "drop out" neurons, smoothing out the regression. In this project, because Convolutional layers have so fewer trainable parameters than Dense layers, overfitting never became an issue.

A more challenging, and pressing, issue is the opacity of the trained models. When we train a network on a dataset, we produce a model that predicts with an accuracy described by a bias from the true distribution and a standard deviation about the mean. This, along with the relative values of the testing and training error, provide some degree of insight into how accurate or reliable our model is. However, it is difficult to extract *why* the model made its decisions, and how much the errors present in the data affect the prediction error of the model. This makes it challenging to extract physics from a model trained on cosmological data: our model can tell us *what* it thinks the values of certain cosmological parameters are, but not *how* it got that answer, or how the errors propagated through the network. In short, NNs do not like to show their work.

While there exists literature proposing solutions to these problems, there are no blanket solutions or ready-made code packages to extract the physics a model sees in its data, because analyzing the decision-making and error propagation of networks is still cutting edge research. As a result, including methods to divine some clarity from your models will involve some overhead in the form of time and knowledge. Sadly, addressing problems of model clarity and error propagation are beyond the scope of this project.

Part III

Training a Model with Reionization Data

13 Applying Machine Learning Techniques to Cosmological Data

The most direct way to constrain phenomenological parameters of the EoR is through interferometric observation of the 21cm signal, using telescopes such as the Low Frequency Array (LOFAR), the Murchison Widefield Array (MWA), and the currently under-construction Hydrogen Epoch of Reionization Array (HERA).

Extracting information from interferometric data is challenging because of its underlying physical complexity, and because the signal-to-noise ratio of the 21cm line is very low. The state change of the EoR is determined by a tangled web of codependent physical processes, and the signal is faint compared to brighter foreground and noise signals, and as a result there is a deficit in robust data analysis methods to extract information from EoR data.

However, Neural Networks have proven highly capable of regressing on complex distributions hiding behind extensive noise. Neural Nets have the potential to outperform other analysis techniques when applied to reionization because NNs have a unique ability to regress on physical information in complex and noisy images, a characteristic feature of the EoR signal.

Additionally, most existing data analysis techniques for the 21cm signal analyze the 21cm power spectrum, rather than the raw interferometric images. Because the power spectrum only contains information about the intensity of spatial modes, we lose information present in the specific 21cm distribution across the sky; this means techniques that only analyze the power spectrum have less information to regress on. By contrast, Convolutional NNs can be trained to regress on raw image data, bypassing the power spectrum entirely and potentially

resulting in a more robust analysis.

In this work we apply a CNN to reionization data and attempt to regress on three physical observables: the midpoint ($z_{50\%}$), duration ($\Delta z = z_{25\%} - z_{75\%}$), and mean z ($\bar{z}$, or the redshift halfway between $z_{25\%}$ and $z_{75\%}$). Note that the midpoint and mean are not always equal, because reionization may not be a symmetric phase change: earlier redshifts could be ionized faster or slower than later redshifts, and the time that marks the halfway point of the state change may not equal the time at which 50% of hydrogen is ionized. While similar research has been done previously [8], this work is unique because we regress on 3 observable parameters at once. We also investigate how limiting the redshift range of the input affects analysis, and whether adding kSZ data affects the regression.

At first glance, deriving these physical parameters from the given data seems like a trivial interpolation problem; simply calculate the percentage of ionized hydrogen at each z slice and find the redshift values associated with each parameter. However, our data includes a rough approximation of foreground contamination that makes the percent of ionized hydrogen per z slice much less obvious (details about foreground contamination modelling can be found in section 14).

One difficulty in applying ML to cosmological data is that ML models require a labelled training dataset: in this instance, that would mean a robust set of 21cm images from many different EoR scenarios, each with the cosmological parameters already well-constrained. Obviously no such set of real observational data is readily available (and further research would be unnecessary if it did). Instead, we train our CNN on simulated EoR scenarios with pre-defined parameters. This introduces an efficacy question: any ML model is only as accurate as the training dataset is reflective of the true data. Therefore, for a model to be incorporated as part of a HERA data analysis pipeline, the training dataset must accurately model not only the physics of reionization but also the noise, foregrounds, and instrumental effects observed in HERA data. We sidestep this question for now, instead focusing on building a model that performs well on data generated by the same semi-numerical simulation

as it was trained on. Future work will focus on better modelling HERA data and EoR physics, to improve the NN's fidelity.

14 Data

Here we provide a brief overview of the data set used in training. For a more comprehensive explanation of the data, see LaPlante 2019 and 2020 [8] [7].

The set of input data consists of 1000 reionization scenarios, each evolved from the same dark matter density field. Each reionization scenario is partitioned into 30 redshift slices, each of which is $2h^{-1}$ Gpc $\times 2h^{-1}$ Gpc, or 512×512 pixels, in area. These slices exist in the redshift range $z \in [6, 15]$ and are partitioned evenly in comoving space. Therefore, each scenario can be represented as a $512 \times 512 \times 30$ time-evolving cube, with the redshift axis representing both line of sight and time. Raw snapshots of the kSZ were also generated for each reionization scenario [7].

The reionization scenarios were then Fourier transformed and the effects of foreground contamination and the limits of HERA's resolution were applied. Interferometers have a complicated point-spread response function (psf) that is both frequency and spatially dependant. This response function interacts with spectrally smooth foregrounds to create a "wedge" of contamination in the Fourier plane. The spatial modes inside this region of contamination are therefore so deeply contaminated that current analysis techniques cannot extract any information from them. To simulate this contamination effect in our simulated data, the spatial modes within this region are removed in our data. The difference between raw and wedge-filtered data can be seen in Figure 4.

After removing the wedge, the image cubes were inverse Fourier transformed to roughly approximate a simulated HERA dataset with foregrounds [8]. While this approach models one of the most egregious sources of interferometric contamination in our data, it ignores other sources of interference (such as noise). For our data to accurately mimic HERA's output these contamination sources would need to be modelled in our simulated cubes; for

this work, only the wedge contamination is modelled. No wedge filter was applied to the kSZ, as the kSZ is derived from non-interferometric CMB observations and the wedge is a uniquely interferometric contamination source.

Before training, each image cube was randomly subsampled down in area to $256 \times 256 \times 30$ cubes. This was done to increase the hypothesis space of our training data (essentially, how many examples our model gets to learn from). The data was split 800/200 into training and validation sets, for training the model and checking for overfitting, respectively. The model was then tested on the full dataset, and the results were plotted. In an ideal scenario a second dataset would be used to test the model's fidelity; however, due to time and computational constraints we only confirm our results for our given dataset. Because our model shows no signs of overfitting from the validation data (see Section 16.1), and because this data already fails to perfectly mimic HERA data, this breach in best practice can be tolerated for now as part of an exploratory work; however, if this research were to continue a test dataset would need to be created.

15 Model Architecture

The model was built using the Keras API of Tensorflow, with a Python backend. It is a fully convolutional network (FCN), meaning it contains no dense layers (layers where each node in a given layer is connected to every node of the adjacent layers). In an FCN, each layer acts as a matrix transformation weighted by the elements of the convolutional filter.

The model architecture can be seen in Figure 8. It is structured as an input layer consisting of an $256 \times 256 \times Z$ matrix (where Z = number of channels) followed by 4 "stacked" layers consisting of a 2d convolutional layer followed by pooling and activation layers which compress and transform the cubes to transform the cubes into the desired shape and improve regression, respectively. These stacked layers are followed by a final convolutional layer and pooling layer to format the output into the correct shape.

The convolutional layers convolved over the input cube with a $3 \times 3 \times Z$ filter. There-

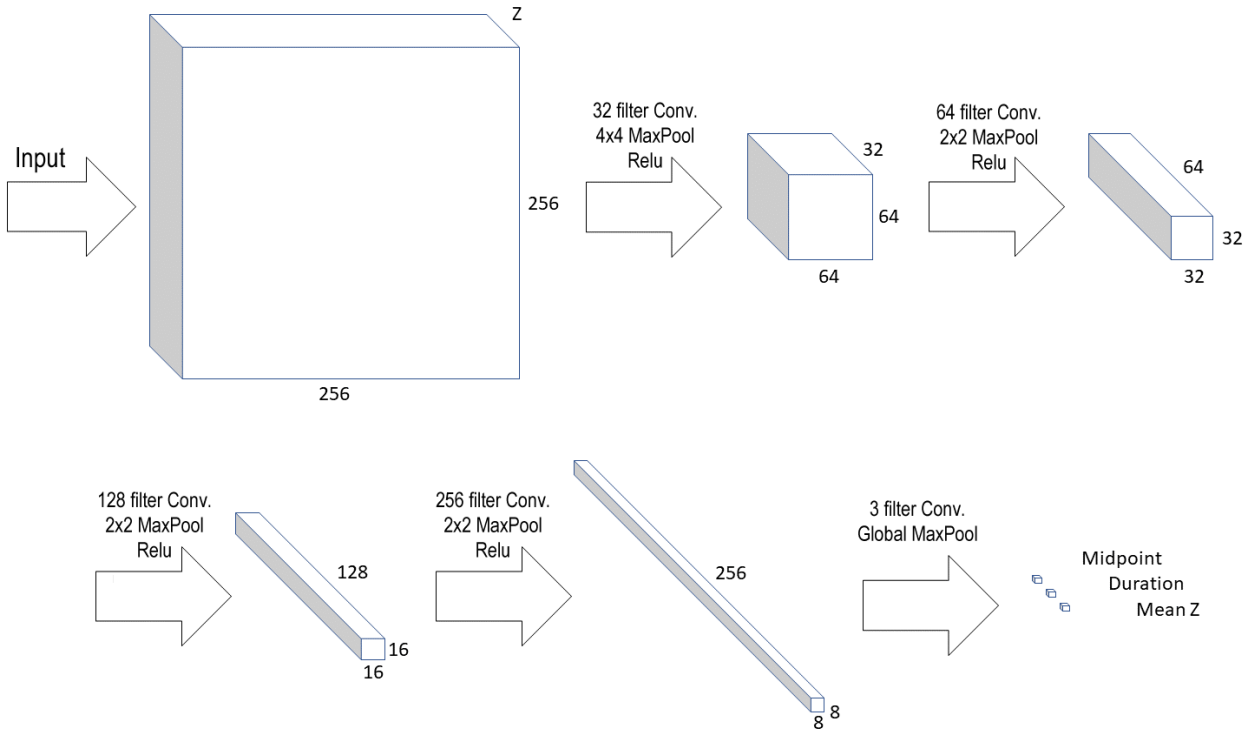


Figure 8: The Fully Convolutional Network architecture used in this work. This model contains no dense layers, instead relying on dimensionality reductions from MaxPooling layers to produce a prediction array of the desired dimensions. Each filter used a $3 \times 3 \times Z$ kernel to perform a 2D convolution on the input. This model was implemented with the Keras Functional API.

fore each filter has $3 \times 3 \times Z$ trainable parameters, and each layer has $3 \times 3 \times Z \times (number\_of\_filters)$ trainable parameters.

Because of the fully convolutional architecture, the input layer can have any arbitrary number of channels, so long as the images are all the same size. We utilize this property in Sections 4 and 5, when adding kSZ data as an additional channel and removing 21cm channels to simulate data loss, respectively.

Over each epoch, the model was trained on the full training dataset, for a number of iterations equal to the size of the training dataset over the batch size (a hyperparameter set by hand before training). For example, our model was trained with a batch size of 160 and a training dataset of size 800, so each epoch our model was trained on a unique permutation of 160 data cubes for 5 iterations.

Part IV

Model Performance

16 Model Performance on Full Data Set

16.1 21cm slices

When trained on a set of 1000 30 slice reionization scenarios for 100,000 epochs, the model achieves an average prediction error rate of approximately 1% when predicting on midpoint and meanz, and an average prediction error rate of approximately 5% when predicting on duration (see Figure 9 and Table 1).

These results show that even with the contaminated Fourier modes in the "wedge" removed, the 21cm signal still contains a wealth of information about reionization.

16.2 kSZ snapshot

To establish a baseline of how the model performed when trained only on kSZ data, the model was trained on the kSZ snapshots of the same set of 1000 reionization scenarios.

Overall, the model is unable to match the performance of the model trained on 21cm data: where the model trained on 21cm data saw prediction errors of about 1% for midpoint and mean z and 5% for duration, the model trained on exclusively kSZ saw about 10% for midpoint and mean z and 15% for duration (See Table 1 and Figure 10). This poor performance is despite the fact that the kSZ snapshots had no contamination effects applied, unlike the 21cm data; this illustrates just how data-rich the 21cm signal is.

The model trained on only the kSZ snapshots performs comparatively best when predicting Δz, which is to be expected since the kSZ signal strongly correlates to the duration of the EoR. However, the model trained on the 21cm signal still outperforms the model trained on kSZ in this area.

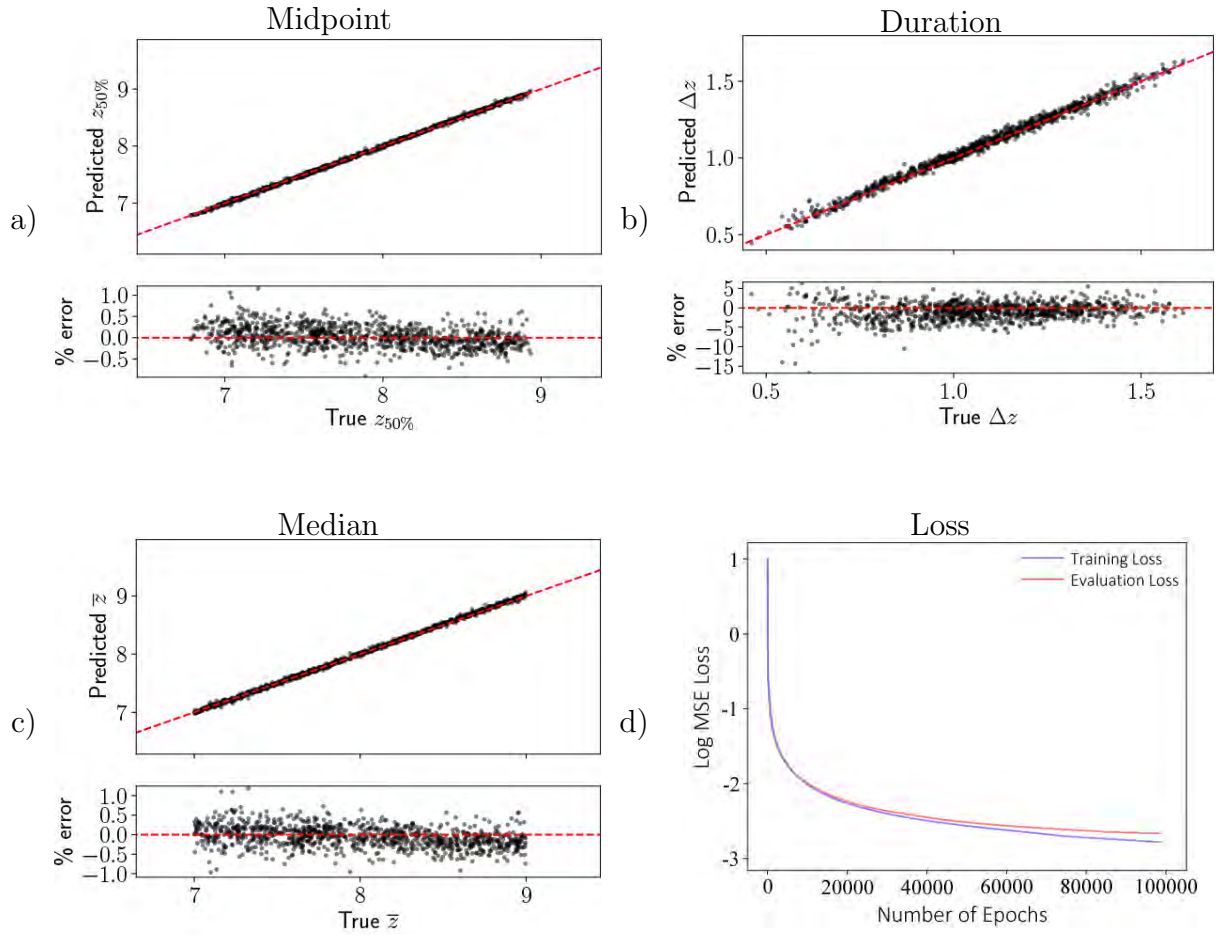


Figure 9: The model's predictions (black datapoints) versus the true data labels (the dotted red line) for a model trained on 30 21cm slices for 100,000 epochs with a batch size of 160. The model regresses on the physical parameters of reionization: a) midpoint, b) duration, and c) mean z. The average displacement of the predicted value from the true value measures the bias of the model and the scatter around the mean measures the precision of the model. These two values give a metric for model performance. The loss curve d) shows no signs of the model overfitting, as the evaluation (validation) loss does not increase at any point (see Section 12 for more details on overfitting)

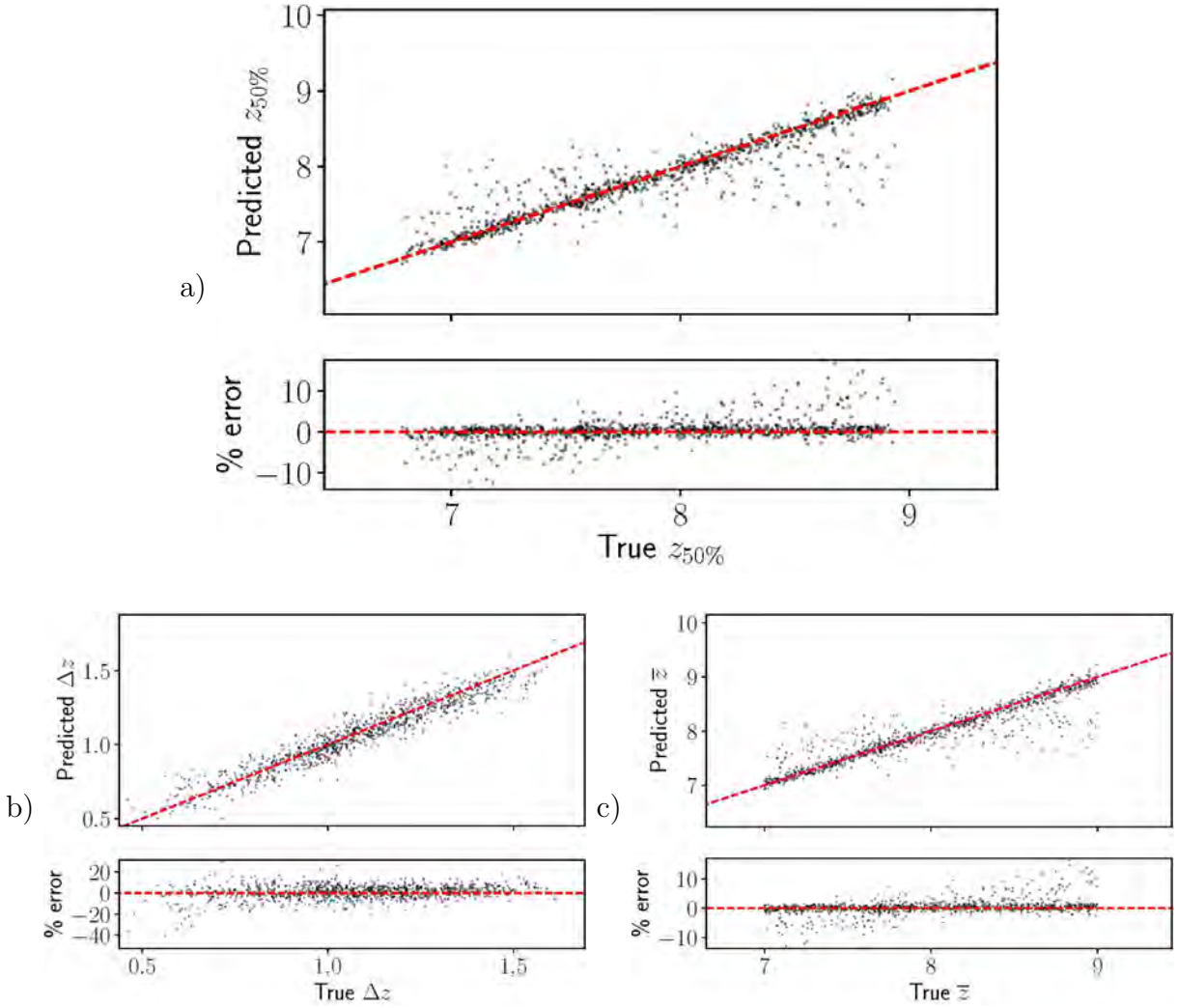


Figure 10: The model's predictions (black datapoints) versus the true data labels (the dotted red line) for a model trained on 1 kSZ snapshot for 100,000 epochs with a batch size of 160. The model regresses on a) midpoint, b) duration, and c) mean z.

16.3 21cm slices + normalized kSZ channel

Having established a baseline of performance when trained on kSZ data alone, the model was trained on both kSZ and 21cm data in an effort to combine the information present in both.

A challenging problem in training on two data sources is how to structure the data and model architecture so that information from both inputs is combined maximally. A more thorough discussion of this problem can be found in Section 20. For this model, we normalized the kSZ relative to the 21cm signal and appended the normalized kSZ to the image cube as an additional channel, resulting in a combined image cube of size $512 \times 512 \times 31$.

This solution may seem counter-intuitive: since the kSZ and 21cm signal are completely different sources of information, why would you treat the kSZ as if it were an additional 21cm channel? Indeed, this solution only works because this model performs only 2D convolutions: as a result, the model does not interpret the data as correlated along the time axis. On a more granular level, the first stacked layer has a set of $(kernel\_size)^2 \times (filter\_size) \times (num\_kSZ\_channels) = 3^2 \times 32 \times 1 = 288$ weights that exclusively weigh the kSZ signal, before that data is folded in with the rest of the convolution, allowing the kSZ to add linear terms to the regression. However, folding kSZ data into the 21cm convolutions on the first layer results in a shallow analysis of the kSZ snapshots at best. 288 free parameters in just 1 convolutional layer are likely not enough to allow the kSZ to contribute meaningfully to the regression, a prediction supported by the results of this model. Future attempts to build a dual-input model that maximally combines 21cm and kSZ information will likely need to be "forked" input models, where several layers convolve over the 21cm and kSZ inputs separately before combining them and performing additional convolutions on the combined data object. This, however, is beyond the scope of this work.

Overall, the model trained with 30 21cm slices + a kSZ snapshot appended as an additional input channel does not perform significantly better than a model trained on 30 21cm channels alone (See Table 1). The fact that the prediction of Δz did not significantly im-

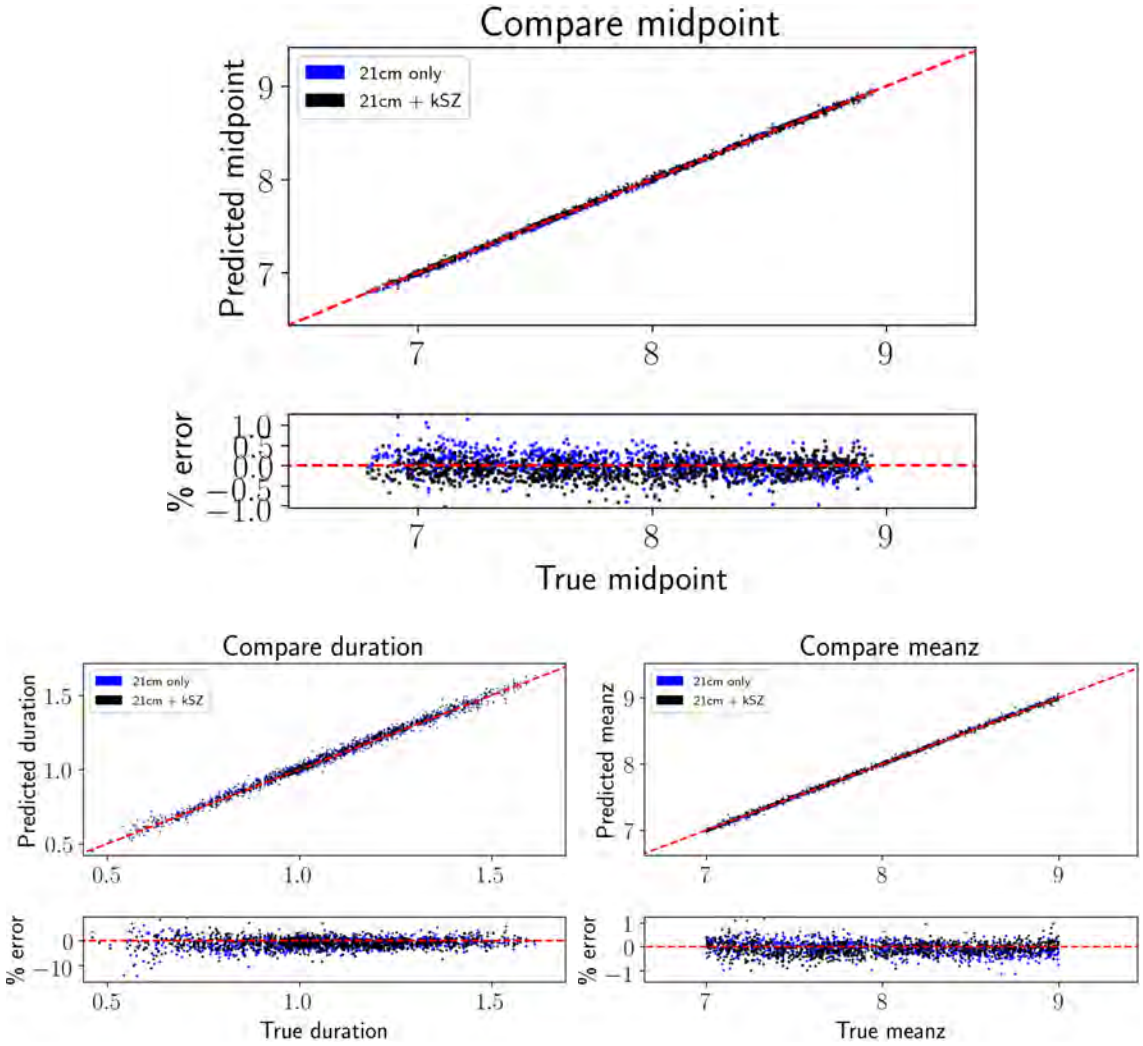


Figure 11: The predictions vs. true values of a model trained and tested on (blue) 30 21cm versus (black) 30 21cm and kSZ. There is little discernible difference between the two distributions; this indicates that either this model does not optimally combine information from the kSZ and 21cm signal, or the 21cm signal contains so much data any benefit from the kSZ is imperceptible, or both.

	21cm only		21cm and kSZ		kSZ only	
	bias	std	bias	std	bias	std
midpoint	0.0012	0.0197	-0.0071	0.0190	0.0307	0.2210
duration	-0.0096	0.0225	-0.0094	0.0209	0.0126	0.0672
meanz	-0.0052	0.0221	-0.0056	0.0206	0.0292	0.2184

Table 1: Bias (average distance from the true value) and standard deviation (scatter) for models trained on 30 21cm slices, 1 kSZ image, and 30 21cm slices with kSZ as an additional channel. All models trained on 100,000 epochs, 160 batch size.

prove, despite the fact that the kSZ only and 21cm only models performed comparatively well in this area, confirms this method of combining the two input sources does not optimally combine their information.

17 Model Performance on Data Subsets

When trained on a set of 1000 reionization cubes represented by 30 21cm slices, our FCN was able to regress on $z_{50\%}$, Δz, and $\bar{z}$ with high degrees of accuracy. However, the data used in training was unrealistically pristine, without the inclusion of noise, instrumental effects, and non-wedge foregrounds. Therefore, we wanted to see how our model performed as a function of data quality. Unfortunately, accurately modelling interferometric noise and realistic foregrounds is a non-trivial task. However, an easy way to remove information from our model's input is to simply reduce the number of redshift slices in each image cube. In the next few sections, we explore how the model's performance degrades with data quality.

17.1 Training on fewer 21cm channels

First, we trained the model on image cubes with redshift slices removed at set intervals. Therefore, the degraded input cubes spanned the same redshift range as the original 21cm cubes, but contained less information on the evolution of reionization (for example, the model trained on 3 redshifts was trained on the redshift slice at indices 0, 10, and 20). The model was trained on reionization cubes with 15, 10, 5, 3, 2, and 1 input channels, and the degradation of performance as a function of data quality can be seen in Figure 12.

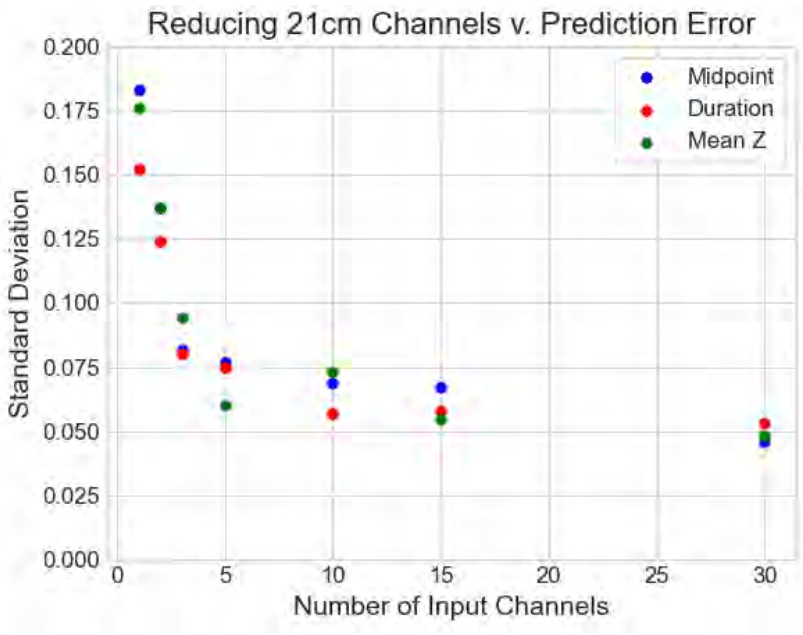


Figure 12: The standard deviation of the predicted values around the true values of the physical parameters of reionization for models with increasingly fewer, increasingly sparse redshift channels. Redshift slices were removed from the cubes at equally spaced intervals, so each model still receives information from the full redshift range of reionization. These models were trained at batch size 160 for 2,500 epochs.

When trained on data degraded in this way, the model performs comparably well until trained on image cubes with fewer than 5 slices, where errors increase significantly. This, again, highlights the density of information present in the 21cm signal; data loss on the order of $\times 6$ still results in comparable accuracy, so long as the image cubes still span the full redshift range of reionization.

When trained on 1 21cm slice, the model could not regress at all; i.e., it assigned every scenario the same labels. Therefore, the model must need at least two points during the ionization state change in order to predict its initial conditions. This indicates that even though the model only utilizes 2D convolutions (and therefore does not consider the changes between time-adjacent slices to be significant), the model requires at least two snapshots of reionization to interpolate between in order to begin regressing on the data. This data was also trained on significantly fewer epochs than the three models considered in Section 16, and so exact standard deviation values should be taken with a grain of salt.

Low z

	21cm		21cm + kSZ	
	bias	std.	bias	std.
midpoint	0.007	0.032	0.014	0.040
duration	-0.009	0.037	-0.002	0.054
meanz	0.009	0.035	0.012	0.043

Mid z

	21cm		21cm + kSZ	
	bias	std.	bias	std.
midpoint	-0.002	0.056	0.017	0.064
duration	-0.002	0.062	0.017	0.064
meanz	0.003	0.055	0.022	0.063

High z

	21cm		21cm + kSZ	
	bias	std.	bias	std.
midpoint	-0.055	0.292	0.018	0.188
duration	0.037	0.130	0.002	0.084
meanz	-0.045	0.281	0.023	0.185

Table 2: Performance of the FCN trained on the low z, mid z, and high z subsets, with and without the addition of a kSZ channel. An additional kSZ channel results in higher bias and std. values for the low and mid z subsets, and lower values for the high z subset, implying adding a kSZ channel to the model only improves performance when the 21cm data contains little information (as in the high z case)

After removing redshift slices at regular intervals to degrade the data while retaining the full redshift range over which reionization takes place, we then trained the model on three smaller redshift range subsets, to gauge how the model performed as a function of the input's redshift range.

These three subsets each contained 1000 image cubes with only three z slices: the low z subset (where the model was trained on cubes with slices $z = \{6.24, 6.77, 7.36\}$), the middle z subset ($z = \{8.24, 9.01, 9.87\}$) and the high z subset ($z = \{11.21, 12.37, 13.72\}$).

The model was able to extract the most information from the low z subset, followed by the mid z subset. The model performed poorly on the high z subset (see Table 2 and Figure 13). This implies that the most useful information about reionization for our model occurs

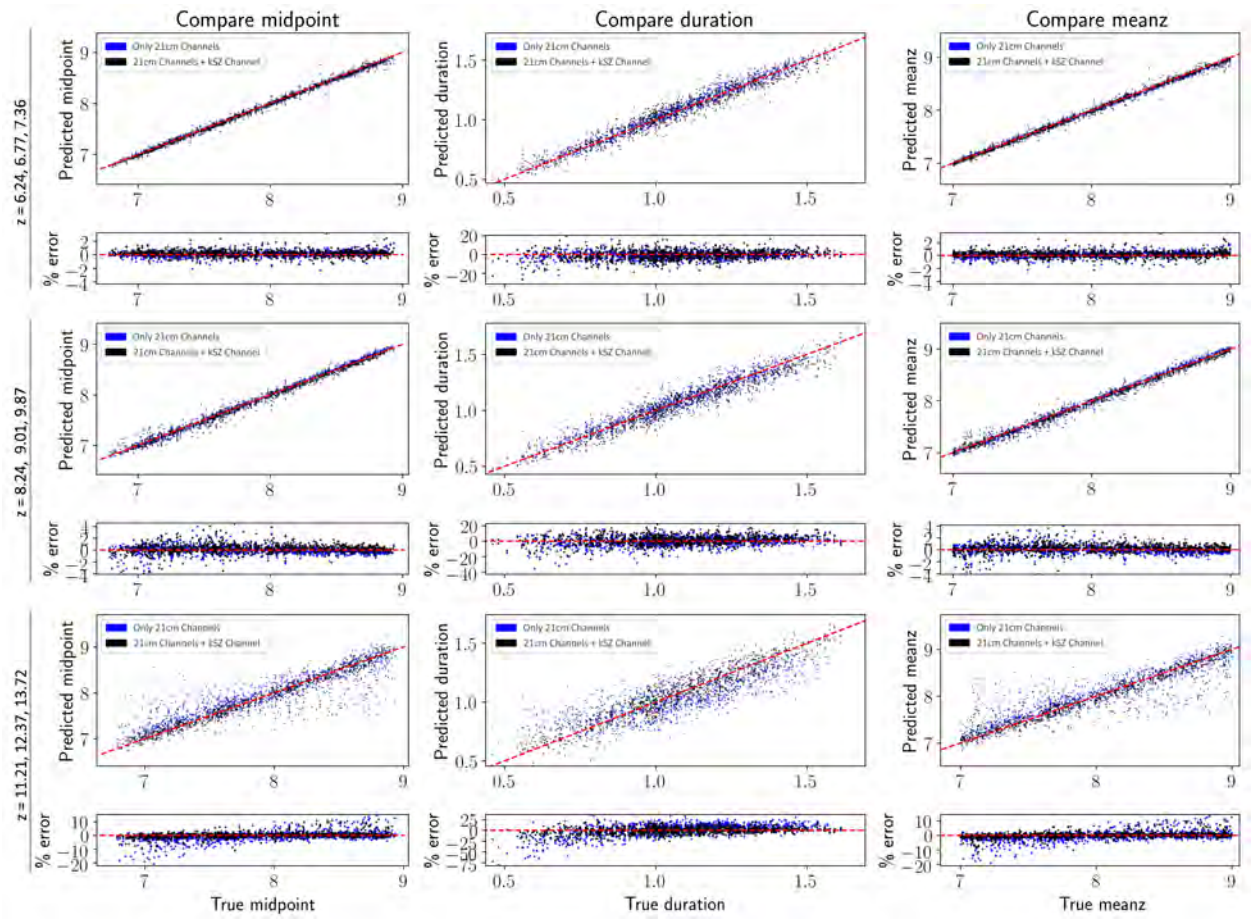


Figure 13: Performance of the FCN trained on the low z, mid z, and high z subsets, with (black) and without (blue) the addition of a kSZ channel. The scatter of the data clearly increases as the data goes back in time, from the low to high z subsets.

later in the state change.

Given the model's redshift range dependent performance, could adding a kSZ channel change the results of any of the smaller range subsets?

17.3 Addition of kSZ as channel

To see how adding a kSZ snapshot as an additional channel affects model performance on smaller redshift range data subsets, we trained models for 10,000 epochs on the same low z, mid z, and high z subsets, this time with an additional kSZ channel appended to the image cubes. The results can be seen in the comparison between the blue and black scatter plots in Figure 13 and Table 2. For the low z and mid z subsets the addition of the kSZ channel

caused the model to perform slightly worse. However, for the high z subset, the model trained with the additional kSZ channel performs better than its 21cm only counterpart. It's possible that this method of combining the 21cm and kSZ information only improves performance when the information in the 21cm input is sparse, as is the case for the high z subset.

However, these results remain inconclusive: because these models were only run for 10,000 epochs, they may not have fully regressed on the input. When training on the full dataset of 30 redshift slices, it was found that the model does not converge until significantly past this point, with the loss continuing to decrease well into the tens of thousands of epochs. Furthermore, on a theoretical level, models trained on a more complex input space should take more iterations to fully regress; therefore, it's possible the models trained with additional kSZ data needs more training time to match or beat the 21cm-only model's performance. Unfortunately, the computational requirements of running multiple additional 100,000-epoch runs placed this analysis beyond the scope of the present work.

Overall, the most significant thing that can be observed from these tests is that the performance of the model depends on the redshift range of the input, and that this FCN architecture could regress on the physical parameters of reionization much more accurately when trained on later redshifts.

Part V

Discussion

18 Data Processing Choices

When processing our data, we subsample the image cubes from $512\times512\times30$ to $256\times256\times30$. When this was originally implemented, we subsampled down to randomly sized cubes instead. It was hoped that subsampling data into randomly sized cubes could increase the size of our

training database, and that training on many angular scales could make the model more robust and flexible. Unfortunately, this yielded much poorer results than when the model was trained on fixed size data cubes. This implies that important data lies in the size scales of the cubes: an intuitive fact, given that the power spectrum of 21cm images contain important data on the reionization process.

19 Model Architecture Choices

The model used in this work is a Fully Convolutional Network (FCN). This means that no dense layers were used in this model. This is an artifact of a previous investigation into using Extended Kalman Filtering (EKF) as an error propagation technique; having a model without dense layers made the mathematics of this technique more tenable. However, even after this investigation concluded and EKF was removed from the model, this architecture performed well enough to be used again for this work. When dense layers were added to this model to investigate whether they could improve the regression, performance did not meaningfully change.

Perhaps because no dense layers were used, this model showed more evidence of underfitting than overfitting (evidence of overfitting being when the validation loss begins to increase as the training loss flatlines or decreases). This might be because convolutional layers "lose" information at a higher rate than dense layers (because convolutional layers convolve over a kernel rather than directly weighing subsequent nodes). As a result, several techniques standard in avoiding overfitting, such as the inclusion of dropout layers, were not used.

20 Alternate model structures

In an attempt to combine information from the kSZ and 21cm signal in our model, we chose to add the kSZ as an additional channel in our input data, a choice discussed more in detail in Section 14. Based on our results, it is inconclusive whether this approach to multi-input model architecture caused any improvement in model performance. Because our

model architecture does not have any hidden layers devoted only to regressing on the kSZ, other model architectures that do would likely be more effective in extracting and combining 21cm and kSZ information.

One model architecture with the potential to more optimally combine kSZ and 21cm information is a dual-input model, in which the 21cm and kSZ data are convolved over separately, merged, and further convolved over before generating a prediction. Initial attempts at designing such an architecture resulted in sub-optimal performance, and therefore our attention shifted to the channel-based approach used in this work; however, there is evidence to suggest that such an architecture could more optimally combine the two data sources, if properly implemented. This is likely the direction of future research, should this work continue.

Part VI

Conclusion

In this work we introduce the observational evidence that the IGM underwent a state change in recent cosmological history, where the previously neutral hydrogen gas making up the IGM is completely ionized. This period, known as the Epoch of Reionization, was spurred by radiation produced by stars following the Cosmic Dawn, and relies on complicated gravitational, thermal, and hydrodynamical interactions. This state change is constrained to recent cosmological history, occurring sometime within $z \in [6, 12]$, but the exact parameters describing its timing, duration, and rate are not known. This is partially due to the challenges of 21cm interferometry, currently our best observational method of analyzing reionization. The 21cm signal, emitted by collections of neutral hydrogen gas, can in theory map the evolution of reionization through different redshifts: however, because of the low signal to noise ratio of the 21cm line, direct observations of reionization remain elusive. However, one recent

observational breakthrough can be found in the recent detection of the kSZ signal, a secondary anisotropy of the CMB caused by the bulk motion of electrons during reionization. While this signal is not as information rich as the 21cm line, it could still yield important constraints on reionization, especially on the duration of the state change.

This work also provides an overview of ML techniques, especially Neural Nets (NNs). NNs rely on a gradient decent algorithm performed over many iterations to regress on some underlying distribution of the input data. NNs excel at finding pattern in chaos, an attractive quality for analyzing data with a low signal to noise ratio such as 21cm images. The key challenges in applying NNs to cosmological data are the necessity of training on simulated data and the statistical opacity of NNs; in this work we address some of these challenges and sidestep others. Overcoming these two hurdles will be essential to future research in this area.

We develop a Fully Convolutional Network (FCN) to regress on a set of 1000 simulated reionization scenarios, each represented by a cube with dimensions $2h^{-1}$ Gpc $\times 2h^{-1}$ Gpc $\times 30$ redshift slices in the range $z \in [6, 12]$, evenly spaced in comoving coordinates. The data cubes had a "wedge" of Fourier modes removed to model foreground contamination in real interferometric data, but no other noise modelling was applied.

The FCN regressed on three physical observables of reionization: $z_{50\%}$ (when the IGM was 50% ionized), Δz (how long reionization took), and $\bar{z}$ (the halfway point between the start and end of reionization). When trained on the full dataset, the FCN could recover $z_{50\%}$ and $\bar{z}$ with $\approx 1\%$ error and Δz with $\approx 5\%$ error. When an additional channel containing the kSZ snapshot for the reionization scenario was appended to the cube, the model performed without measurable change.

Training the same FCN on degraded 21cm data revealed that, so long as the redshift range of the input data covers all of reionization, removing z slices from the FCN's input data did not meaningfully impact performance until data cubes contained fewer than 5 redshift slices, after which performance deteriorated linearly with the reduction of slices in the input,

culminating in the model unable to begin regressing on one z snapshot of reionization.

Training the FCN on input cubes with limited redshift range revealed that the model's performance degraded significantly when trained on higher redshift ranges, implying that the most information available for the model lies in later redshifts. When appending a kSZ snapshot to these limited redshift range subsets, the kSZ actively degraded performance on the lower and middle redshift ranges while seemingly improving performance on the high redshift ranges. This result does not conclusively indicate that this approach to combining the 21cm and kSZ data sources can improve performance, because there is evidence to suggest that these models did not have time to fully regress on the data. Overall, it is clear that other model architectures that process the kSZ and 21cm data individually for longer before analyzing them together would have combined their information more efficiently; this is a direction for future research, if this work were to continue.

ML analysis techniques are somewhat controversial in research circles, largely because of their perceived "trendiness" and their resistance to statistical scrutiny. For example, we trained a NN to predict the physical parameters of reionization (and it does so quite accurately), but we don't know why the model makes the decisions it does, or what physical laws or processes it observes to make its classifications. It's easy to get an answer from an ML model; it's hard to get a solution. In short, ML algorithms don't like to show their work–a crucial problem for their use in science.

Significant work needs to be done to elucidate the decision making processes of NNs, because otherwise their application to physics research is limited. ML models promise to be fruitful statistical tools, and future work done to improving their interpretability could allow them to be game-changing in helping us uncover new physical patterns and laws in our universe.

References

[1] R Adam, N Aghanim, Mark Ashdown, J Aumont, C Baccigalupi, M Ballardini, AJ Banday, RB Barreiro, N Bartolo, Suman Basak, et al. Planck intermediate results-xlvii. planck constraints on reionization history. *Astronomy & Astrophysics*, 596:A108, 2016.

[2] LE Bleem, TM Crawford, B Ansarinejad, BA Benson, S Bocquet, JE Carlstrom, CL Chang, R Chown, AT Crites, T de Haan, et al. Cmb/ksz and compton-y maps from 2500 square degrees of spt-sz and planck survey data. *arXiv preprint arXiv:2102.05033*, 2021.

[3] François Chollet et al. Keras. https://keras.io, 2015.

[4] Wikimedia Commons. Diagram showing overfitting of a classifier, 2008.

[5] Xiaohui Fan, Michael A Strauss, Gordon T Richards, Joseph F Hennawi, Robert H Becker, Richard L White, Aleksandar M Diamond-Stanic, Jennifer L Donley, Linhua Jiang, J Serena Kim, et al. A survey of z¿ 5.7 quasars in the sloan digital sky survey. iv. discovery of seven additional quasars. *The Astronomical Journal*, 131(3):1203, 2006.

[6] Bradley Greig and Andrei Mesinger. The global history of reionization. *Monthly Notices of the Royal Astronomical Society*, 465(4):4838–4852, 2017.

[7] Paul La Plante, Adam Lidz, James Aguirre, and Saul Kohn. The 21 cm ksz–ksz bispectrum during the epoch of reionization. *The Astrophysical Journal*, 899(1):40, 2020.

[8] Paul La Plante and Michelle Ntampaka. Machine learning applied to the reionization history of the universe in the 21 cm signal. *The Astrophysical Journal*, 880(2):110, 2019.

[9] Abraham Loeb and Steven R Furlanetto. *The first galaxies in the universe*, volume 21. Princeton University Press, 2013.

[10] CL Reichardt, S Patil, PAR Ade, AJ Anderson, JE Austermann, JS Avva, E Baxter, JA Beall, AN Bender, BA Benson, et al. An improved measurement of the secondary cosmic microwave background anisotropies from the spt-sz+ sptpol surveys. *arXiv preprint arXiv:2002.06197*, 2020.

[11] Barbara Ryden. *Introduction to cosmology*. Cambridge University Press, 2017.

[12] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.