Understanding the Impact of Semi-numeric Reionization Models when Using CNNs

Yihao Zhou1 and Paul La Plante2,3

1 School of Science, Xi’an Jiaotong University, No. 28 West Xianning Road, Xi’an 710049, People’s Republic of China; yihaozhou20@gmail.com
2 Astronomy Department, University of California, Berkeley, CA 94720, USA
3 Berkeley Center for Cosmological Physics, University of California, Berkeley, CA 94720, USA

Received 2021 December 7; accepted 2022 March 21; published 2022 April 26

Abstract

Interpreting 21 cm measurements from current and upcoming experiments like HERA and the SKA will provide new scientific insights and exciting implications for astrophysics and cosmology regarding the Epoch of Reionization (EoR). Several recent works have proposed using machine learning methods, such as convolutions neural networks (CNNs), to analyze images of reionization generated by these experiments since they could take full advantage of the information contained in the image. Generally, these studies have used only a single semi-numeric method to generate the input 21 cm data. In this work, we investigate the extent to which training CNNs for reionization applications depends on the underlying semi-numeric models. Working in the context of predicting CMB τ from 21 cm images, we compare networks trained on similar data sets from 21cmFAST and zreion, two widely used semi-numeric reionization methods. We show that neural networks trained on input data from only one model produce poor predictions on data from the other model. Satisfactory results are only achieved when both models are included in the training data. This finding has important implications for future analyzes on observation data, and encourages the use of multiple models to produce images that capture the full complexity of the EoR.

Key words: Reionization – Computational methods – Intergalactic medium

1. Introduction

The Epoch of Reionization (EoR, 6 ≤ z ≤ 15) refers to a period of the universe preceded by the Cosmic Dawn (15 ≤ z ≤ 30), when the first stars formed, and the Dark Ages (30 ≤ z ≤ 200), the period following recombination (Loeb & Furlanetto 2012). During the EoR, some of the earliest luminous sources emitted a sufficient number of ultraviolet and soft X-ray photons that escaped into the intergalactic medium (IGM) such that it reionized neutral hydrogen H I. As a result, the majority of baryonic matter in the IGM transitioned to a highly ionized state (Furlanetto 2016), which is how it remains today.

There are several ways to study this major phase transition: distant quasars, the cosmic microwave background (CMB), and the 21 cm line of H I. Analyzing the absorption properties of the IGM using quasars has provided important constraints on reionization, showing that reionization was likely only partially completed at 7 ≤ z ≤ 8 (Schenker et al. 2014; Davies et al. 2018; Mason et al. 2019). Absorption spectra from lower redshift measurements argue that reionization was mostly completed by z ~ 6 (Fan et al. 2006), though other measurements may indicate that reionization was still ongoing as late as z ~ 5.5 (Bosman et al. 2021). Complementing this, the relatively low CMB optical depth τ (Planck Collaboration et al. 2020) suggests that reionization happened relatively late and was extended in redshift. Future detection of the redshifted 21 cm line emission will be an essential probe of the thermal and ionization state of the IGM (Fan et al. 2006; Furlanetto 2016). The neutral hydrogen gas in the IGM emits radiation due to the hyperfine transition of the H I ground state, with a wavelength of λ = 21 cm in the hydrogen rest-frame (Zaldarriaga et al. 2004). Currently, several experiments have been designed to measure the global temperature change of 21 cm, such as the Experiment to Detect the Global EoR Signature (EDGES; Bowman et al. 2018), or target a statistical detection through the power spectrum, including the Low Frequency Array (LOFAR; van Haarlemen et al. 2013) and the Murchison Widefield Array (MWA; Tingay et al. 2013). The Hydrogen Epoch of Reionization Array (HERA; DeBoer et al. 2017) has recently provided the tightest constraints from an interferometer on the 21 cm power spectrum (The HERA Collaboration et al. 2021a), which has ruled out part of the physically allowed phase space of galaxy models (The HERA Collaboration et al. 2021b). The Square Kilometre Array (SKA; Koopmans et al. 2015) will provide high signal-to-noise imaging of the EoR and the Cosmic Dawn, mapping the first billion years of our universe.

The 21 cm signal encodes a panoply of information about the first generation of galaxies, through which we can get constrains on key properties of reionization and gain an insight into the relating astrophysical process (McQuinn et al. 2007;
Fialkov et al. (2016). However, how to extract information from the images generated by the arrays mentioned above still remains a great challenge due to the strong foreground contamination that is several orders of magnitude larger than the EoR signal (Morales & Wyithe 2010; Pober et al. 2013). Computing the power spectrum is a reliable way given these measurement difficulties because the foreground contamination is intrinsically smooth while the target signal fluctuates rapidly. Several recent studies have developed robust methods for using the 21 cm power spectrum to infer properties of early galaxies from the EoR (Liu et al. 2016; Greig & Mesinger 2018; Qin et al. 2020). While because the 21 cm field is highly non-Gaussian (Shimabukuro et al. 2016; Majumdar et al. 2018), in principle there is more information that can be extracted by using a higher-point estimator such as the bispectrum, or by working in configuration space directly (Hutter et al. 2020; Watkinson et al. 2022). By configuration space, we mean the real or redshift space data, which is in contrast to Fourier space.

One such solution to this problem is to use machine learning (ML) techniques. Machine learning is a powerful tool to find complex relationships among data without requiring much a priori information about the functional forms of statistics used to extract the desired information. Crucially, because these methods work natively in configuration space, they are not limited to Gaussian information, allowing us in principle to take full advantage of all information in the 21 cm maps. Machine learning has been used in a variety of works to analyze maps relevant to 21 cm cosmology. For example, some studies have used ML to infer galaxy model properties from 21 cmFAST (Gillet et al. 2018), model-independent parameters such as the duration of reionization $\Delta z$ (La Plante & Ntampaka 2019), or cosmological parameters such as $\tau$ and $\sigma_8$ (Horta et al. 2020; Billings et al. 2021). Recently, Zhao et al. (2022) use 3D CNN to compress 3D image data and perform a Bayesian inference of reionization parameters, which outperforms earlier analysis based on two-dimensional 21 cm images. Other studies have attempted to reconstruct the underlying ionization field (Gagnon-Hartman et al. 2021) or dark matter density field (Villanueva-Domingo & Villaescusa-Navarro 2021) from single-model input data. Besides that, generative adversarial networks are used in some works to generate maps of neutral hydrogen ( Zamudio-Fernandez et al. 2019) or predicted different summary statistics using CNN architectures (Kwon et al. 2020).

In most studies mentioned above, a semi-numeric method is used to convert the initial conditions and dark matter density fields into ionization fields and ultimately 21 cm maps. Two particular frameworks to model and simulate reionization are 21 cmFAST (Mesinger et al. 2011; Murray et al. 2020) and zreion (Battaglia et al. 2013), which produce qualitatively similar yet distinct predictions. Generally, only one semi-numeric method is used to generate testing and training data without a detailed treatment of the uncertainty induced by using solely one method. In order to understand the applicability of ML techniques to real observation data in the future, we must endeavor to build and train networks that are as flexible and robust as possible. In particular, these networks should be insensitive to the particular details of a specific semi-numeric method when predicting a model-independent parameter such as $\tau$ or $\sigma_8$. In other words, the network is supposed to be independent of specific semi-numeric EoR models. We aim to investigate this problem in this work in the context of regressing optical depth to the CMB optical depth. We compare the behavior of convolutional neural networks (CNNs) on zreion and 21 cmFAST. We also discuss the possibility of having one network that works on both of the two models.

We organize this paper as follows. In Section 2, we introduce and compare two semi-numeric models used in this work. In Section 3, we describe the ML techniques we employed. In Section 4, we demonstrate the performances of the CNN on the two models. We discuss the results in Section 5 and conclude in Section 6 with a summary and avenues for future study. Throughout the work, unless stated otherwise, we employ a $\Lambda$CDM cosmology with cosmological constants consistent with the Planck 2018 results (Planck Collaboration et al. 2020).

2. Semi-numeric Reionization Methods

Accurately simulating the EoR with sufficient fidelity requires a delicate balancing act between enormously different length scales and the sophistication of the relevant physics. On small-scales, the formation of Population III and Population II stars as the predominant engines driving reionization happens on the parsec (pc) scale (Regan et al. 2020). Alternatively, the typical size of ionized bubbles during the EoR is tens to hundreds of Mpc. This severe difference in length scale means that accurately resolving all relevant physics is generally not feasible given current computational resources. Furthermore, the complicated nature of reionization generally requires to include N-body methods and hydrodynamics for galaxy formation, and radiative transfer to capture the large-scale structure, the feedback between photons and baryons in the IGM.

Considering that upcoming experiments are expected to have relatively poor angular resolution, large-scale predictions that are accurate on Mpc scales are generally sufficient for understanding and interpreting 21 cm measurements. Because it is most important to capture the largest scales, various approximations can be made. In particular, in the literature there are several semi-numeric simulation methods that do not require full radiative transfer simulations to generate predictions for reionization that are reasonably accurate. Two specific ones are zreion described in Battaglia et al. (2013), and 21 cmFAST described in Mesinger et al. (2011). Below, we...
briefly summarize their key features and refer the reader to the original publications for a more detailed discussion of each model.

2.1. zreion

The zreion method\(^4\) is presented in Battaglia et al. (2013), and has been applied to the kinetic Sunyaev-Zel’dovich effect (Battaglia et al. 2013; Natarajan et al. 2013) and the 21 cm field (La Plante et al. 2014; La Plante & Ntampaka 2019; La Plante et al. 2020). The method treats the fluctuation of “redshift of reionization field” \(\delta_z(x)\) as a biased version of the matter overdensity field \(\delta_m(x)\) on large scales (\(\gtrsim 1\ h^{-1}\)Mpc). First, we define \(\delta_m(x)\) as

\[
\delta_m(x) \equiv \frac{\rho_m(x) - \bar{\rho}_m}{\bar{\rho}_m},
\]

where \(\bar{\rho}_m\) is the mean matter density, and \(\delta_z(x)\) as

\[
\delta_z(x) \equiv \frac{[z_{re}(x) + 1] - [\bar{z} + 1]}{\bar{z} + 1},
\]

where \(z\) is the mean value for the \(z_{re}(x)\) field. zreion takes the advantage of the fact that \(\delta_m(x)\) is highly correlated with \(\delta_z(x)\). We introduce a bias parameter \(b_{zm}(k)\) to quantify the relationship between the two fields in Fourier space:

\[
b_{zm}^2(k) \equiv \frac{\langle \delta_z^* \delta_z \rangle_k}{\langle \delta_m^* \delta_m \rangle_k} = \frac{P_z(k)}{P_{mm}(k)},
\]

where \(P_{mm}(k)\) is the power spectrum of \(\delta_m\) and \(P_z(k)\) is that of \(\delta_z\). To parameterize the bias, the method uses two parameters \(k_0\) and \(\alpha\) to write \(b_{zm}\) as a function of Fourier wavenumber \(k\):

\[
b_{zm}(k) = \frac{b_0}{\left(1 + \frac{k}{k_0}\right)^\alpha}.
\]

We use the value of \(b_0 = 1/\delta_z = 0.593\), where \(\delta_z\) is the critical overdensity in the spherical collapse model. Given a cosmological density field, the method relies only on the value of the three parameters \([\bar{z}, k_0, \alpha]\) to determine \(z_{re}(x)\). The midpoint of reionization depends largely on the mean value \(\bar{z}\) (though they are not equivalent in general unless the mean and median redshift are the same), and the duration is controlled by \(k_0\) and \(\alpha\).

In the original work of Battaglia et al. (2013), the values of \(\bar{z}\), \(k_0\), and \(\alpha\) were calibrated by hydrodynamic simulations of reionization with radiative transfer. Using this approach, the best-fit values for \(k_0\) and \(\alpha\) were \(k_0 = 0.185\ h\)Mpc\(^{-1}\) and \(\alpha = 0.564\). However, the method is sufficiently flexible that varied values could be used when generating the reionization field \(\delta_z(x)\). This flexibility allows for a broad range of different reionization scenarios to be generated relatively quickly. Nevertheless, there are some limitations associated with this method. Due to the fact that this method does not explicitly use the sources of ionizing radiation when generating the reionization history, it is not straightforward to include specific models of galaxy formation. Instead, reionization is treated in a statistical manner, without considering the properties of individual galaxies. Also, due to the treatment of the “redshift of reionization” field as a biased tracer of the dark matter field, it is difficult to generate reionization histories that are very extended in duration, or very asymmetric about the midpoint of reionization. For instance, zreion could hardly produce a reionization history that has an early low-level of ionization (\(\sim 10\%\)) for a long time before the midpoint of reionization, as might be caused by a significant amount of radiation from Pop III stars (Miranda et al. 2017). It also does not easily account for quasar-like sources that generate large (\(\sim 100\ h^{-1}\)Mpc\(^{-1}\)) ionized regions. This is largely due to the fact that the dark matter field is only weakly non-Gaussian on large scales at high redshift, and the resulting \(\delta_z(x)\) will also be only weakly non-Gaussian.

Despite these limitations, zreion can produce similar reionization histories to those from 21cmFAST or hydrodynamic simulations with much less time. Another benefit of zreion as a method is that it treats the dark matter density field as an external input, which facilitates an apples-to-apples comparison with other methods like 21cmFAST. It also supports running on matter fields generated from N-body simulations, as was originally done in Battaglia et al. (2013), or from second-order Lagrangian perturbation theory (2LPT).

2.2. 21cmFAST

The 21cmFAST method\(^5\) (Mesinger & Furlanetto 2007; Mesinger et al. 2011; Murray et al. 2020) assumes galaxy-driven reionization and includes a more involved treatment of reionization sources. Using the excursion set formalism (Bond et al. 1991) and Lagrangian perturbation theory, the dark matter distribution of a volume can be converted into a “halo field” that form the sources of ionizing radiation. Starting from initial conditions, the halo field is computed in a comoving volume at decreasing redshift values. Given the halo field at a particular redshift, 21cmFAST uses a series of astrophysical parameters pertaining to galaxy and star formation to compute the corresponding radiation field. Once the cumulative number of hydrogen-ionizing photons has been calculated from UV and X-ray sources, the number of neutral atoms surrounding these sources are computed within regions of decreasing \(R\), which begins at a parameter defining the maximum mean-free path of ionizing radiation, \(R_{mfp}\), and decreases down to the resolution of an individual cell, \(R_{cell}\). A voxel is considered fully ionized.

\(^4\) https://github.com/plaplant/zreion

\(^5\) https://github.com/21cmfast/21cmFAST
when (Greig & Mesinger 2018)

\[ \xi_{\text{coll}}(x, z, R, \tilde{M}_{\text{min}}) \geq 1, \]

where \( f_{\text{coll}}(x, z, R, \tilde{M}_{\text{min}}) \) is the fraction of collapsed matter residing within halos more massive than \( M_{\text{min}} \) (Press & Schechter 1974; Bond et al. 1991; Lacey & Cole 1993; Sheth & Tormen 1999), and \( \xi \) is an ionizing efficiency factor which accounts for how dark matter halos convert gas into ionizing photons.

This approach assumes a constant relationship between the ionizing luminosity and the halo mass down to some minimum mass \( \tilde{M}_{\text{min}} \), below which star formation is inefficient due to feedback and/or cooling effects. This mass can be expressed in terms of a redshift-dependent halo virial temperature \( T_{\text{vir}} \) (Barkana & Loeb 2001):

\[
M_{\text{vir}}^{\text{min}}(z) = 10^9 h^{-1} \left( \frac{\mu}{0.6} \right)^{-3/2} \left( \frac{\Omega_{M,0} \Delta_c}{\sqrt{\Omega(z) 18 \pi^2}} \right)^{-1/2} \times \left( \frac{T_{\text{vir}}}{1.98 \times 10^8 K} \right)^{3/2} \left( \frac{1 + z}{10} \right)^{-3/2} M_{\odot},
\]

where \( \mu \) is the mean molecular weight, \( \Omega(z) = \Omega_{M,0} (1 + z)^3 / \{ \Omega_{M,0} (1 + z)^3 + \Omega_{\Lambda} \} \), and \( \Delta_c = 18 \pi^2 + 82 d - 39 d^2 \), and \( d = \Omega(z) - 1 \).

Once the the relations between halo mass and ionizing efficiency are defined, \( k \)-space filtering is done to determine the scales at which the amount of ionization radiation produced is equal to the amount of neutral hydrogen gas, which informs which regions have been ionized. Following this, the ionization field is generated and is then converted into a 21 cm brightness field. For our purposes, we extract the ionization history given this ionization field as well as snapshots of the 21 cm brightness temperature. The midpoint and duration of reionization are thus not controlled directly by the user, and instead are the result of the complicated interplay of the different astrophysical parameters.

### 2.3. Comparison

In the following analysis, we generate snapshots of the 21 cm field using both \textit{zreion} and \textit{21cmFAST} that serve as training data input for machine learning methods (described more below in Section 3). We use the same cosmology, comoving box size of 2 \( h^{-1} \) Gpc, and 512 output resolution elements per linear dimension for both simulations. The input data are two-dimensional images of the 21 cm brightness temperature snapshots generated at 30 different redshift slices, which are the same for the two methods.

There are two primary differences between \textit{zreion} and \textit{21cmFAST} methods: the means of generating the underlying dark matter density field (\( N \)-body simulations for \textit{zreion}, second-order Lagrangian perturbation theory (2LPT) for \textit{21cmFAST}) and the treatment of ionization sources (implicit sources for \textit{zreion}, explicit galaxies for \textit{21cmFAST}). As mentioned in Section 2.2, \textit{21cmFAST} uses 2LPT to generate a comparatively low-resolution grid from the high-resolution Gaussian initial conditions at a very high redshift. The ionization field and associated calculations are implemented on this lower-resolution field. For this study, \textit{zreion} uses \( N \)-body simulation snapshots and work all the way on the high-resolution field until the last step before outputting, which makes \textit{zreion} to have more accurate density fields at the cost of computation time. For the method to generate brightness temperature, \textit{21cmFAST} works mostly in configuration space while \textit{zreion} works in Fourier space.

Figure 1 shows a qualitative comparison of the 21 cm brightness temperature as generated by \textit{zreion} and \textit{21cmFAST} using the same underlying density field and a comparable reionization history. As can be seen, the two simulations are capable of producing comparable results since the general topology of the ionization field is similar between the two methods except the higher peaks in the \textit{zreion} simulation at lower redshift. The 21 cm power spectra at high redshift (\( z = 15 \)) and low redshift (\( z = 6 \)) shown in Figure 2 provide another comparison of the \textit{zreion} and \textit{21cmFAST} in Fourier space. At high redshift, the shapes of the power spectra are very similar, though they have an overall normalization difference. At low redshift, however, there is a difference both in amplitude and the shape as a function of \( k \). Without considering the general amplitude level, \textit{21cmFAST} has relatively less power for large \( k \) values (smalls scales) compared with \textit{zreion}. Since different radiation feedback scenarios, i.e., varied parameter sets, can hard affect the power spectra at high \( k \) value, especially near the end of reionization, \textit{(Hutter et al. 2021)}, the distinctions in configuration space and Fourier space could both be attributed to the different methods of generating the ionization fields and the brightness temperature, particularly with respect to the low-density regions that are partially or totally neutral near the end of reionization.

Although the large-scale differences can be made to agree through an overall renormalization, the divergent behavior on small scales has important impacts on generalizations when making predictions on parameters that are ostensibly model-independent, which we show more below in Section 4.

### 2.4. Generating Snapshots

As mentioned previously, our input data are a series of two-dimensional snapshots of the 21 cm brightness temperature generated at different redshift values. Crucially, without directly using the brightness temperature generated by \textit{zreion} or \textit{21cmFAST}, we apply two instrumental artifacts: a maximum angular resolution and a “wedge filter”, which will be present in upcoming data generated by the HERA or the...
SKA. For the present study, we adopt the design parameters of HERA in this work.

The angular resolution of radio arrays is determined by the longest baselines:

$$p = \frac{k u_{\text{max}}}{D_{c2}},$$

where $D_{c}$ is the comoving distance along the line of sight and $u_{\text{max}}$ is the maximum baseline vector. The longest baseline for HERA is $\sim 870$ m (DeBoer et al. 2017), which corresponds to $k_{l,\text{max}} \sim 0.5 \ h^{-1}\text{Mpc}$ at $z = 8$. To include this effect in our snapshot data, we set all the modes with $k_{l} \geq k_{l,\text{max}}$ to be 0 in Fourier space.

Another observation effect is foreground contamination, which is present in all interferometer data. Within the target wavelength range, the foreground radio signal, which mainly comes from the synchrotron radiation in the Milky Way, overwhelms the EoR signal by several orders of magnitudes (Morales & Wyithe 2010). Since the galactic synchrotron radiation roughly follows a power law, it is intrinsically smooth in Fourier space. Naïvely, the contamination should be limited to small $k_{l}$ modes. However, due to the interferometer’s chromatic response, the foreground signal scatters into high $k_{l}$ modes and creates a “wedge” in the Fourier space (Parsons et al. 2012). The slope of the wedge $m(z)$ is a function of the baseline length and cosmology, and can be expressed as (Thyagarajan et al. 2015):

$$m(z) = \frac{k_{l}}{D_{c} f_{21} H(z)} = \frac{\lambda D_{c} f_{21} H(z)}{c^{2}(1 + z)^{2}},$$

where $\lambda$ is the 21 cm signal wavelength at the given redshift $z$, $D_{c}$ is the comoving distance, $f_{21}$ is the rest-frame 21 cm signal frequency, and $H$ is the Hubble constant. Note that the expression in Equation (8) accounts for maximal data contamination; physically, the bright foreground contamination extends down to the horizon of the interferometer beam. To include this effect, we transform the snapshots into Fourier space and set all the Fourier modes where $k_{l} \geq m(z)k_{l}$ to be 0. For the redshift values of interest here, $m \sim 3$, which leads to removing a significant amount of data from the input snapshots.

Figure 1. A comparison of zreion (top) and 21cmFAST (bottom) simulations. The columns show a two-dimensional snapshot at $z = 6$ (left), $z = 7$ (center), and $z = 15$ (right). Because the same density field is used for the two simulations, the differences are due primarily to the means by which the ionization and brightness temperature fields are computed. Note that although the small scales of the images are different, especially at relatively low redshift values, the larger scale features are generally comparable.
Table 1

| Parameter                  | Unit         | Fiducial Value | σ      | Minimum Value | Maximum Value |
|----------------------------|--------------|----------------|--------|---------------|---------------|
| \( \log(T_{\text{vir}}) \) | K            | 5.10           | 0.15   | ...           | ...           |
| \( \zeta \)                |              | 100            | 25     | ...           | ...           |
| \( \log(M_{\text{cut}}/M_\odot) \) | ...         | 8.7            | 0.3    | ...           | ...           |
| \( E_0 \)                  | keV          | 500            | 50     | ...           | ...           |
| \( \alpha \)               |              | 1.0            | 0.3    | ...           | ...           |
| \( \tau_s \)               |              | 0.5            | 0.1    | ...           | ...           |
| \( \log(L_{\text{vir}}) \) | erg s\(^{-1}\) M\(_{\odot}\) yr | 40      | 0.5    | ...           | ...           |

Note. For the 21cmFAST snapshots, we randomly choose a value from a Gaussian distribution for each parameter characterized by the mean (“fiducial”) value and σ value specified in the table. For the zreion snapshots, we randomly choose each parameter from a uniform distribution between the ranges of values. Choosing the model parameters in this way leads to ionization histories that are comparable, which leads to comparable values of \( \tau \).

The way we generate snapshots is modeled after La Plante & Ntampaka (2019). For both zreion and 21cmFAST, we vary model parameters given in Table 1 to generate new simulations. For 21cmFAST, we use a Gaussian distribution for each parameter with the mean value listed in the third column and the standard deviation present in the fourth column. For zreion, we randomly change \( \{ \zeta; k_0; \alpha \} \) with the minimum and maximum values shown in the fifth and sixth columns. Then, we generate brightness temperature boxes at a given redshift. After this we apply filters to include the observation effects mentioned above. We pick the central two-dimension slab in the box to represent the signal at a specific redshift. One slab lies in the plane of the sky, spanning 2 \( h^{-1}\)Gpc with 512 pixels on each side. The axis along the line of sight spans \( \sim48.8 \, h^{-1}\)Mpc. Over such relatively short distance the redshift evolution could be ignored, so the slab could be considered comoving and we do not include light cone effect (La Plante et al. 2014). We generate 30 slices that extend over the redshift range \( 6 \leq z \leq 15 \) and use them as a single image fed into the CNNs. Finally, we normalized each snapshots among \( 0 \sim 1 \). As a result, each image has \( 512 \times 512 \times 30 \) pixels. We generate 1000 snapshots for zreion and 21cmFAST, respectively. For both of the two methods, we keep the underlying cosmological parameters fixed, though we explore the impact of changing these parameters in Appendix.

Figure 3 shows the reionization history for several snapshots from zreion and 21cmFAST, which are randomly chosen from the data set. As can be seen, the histories produced by the two methods are qualitatively similar, and the large variances of parameter space produce a wide range of reionization histories. Despite the large-scale agreement between the two simulations, there are nevertheless several differences. In general, the reionization histories of zreion are slightly briefer than 21cmFAST (e.g., the slope of the blue lines in the figure are mostly steeper than the orange ones). As discussed above in Section 2.1, it is difficult to generate zreion simulations with very extended reionization histories. As such, there are several 21cmFAST samples with large duration that cannot be matched despite widely varying the zreion parameters. Nevertheless, at the level of these global one-point statistics, the two methods can produce similar reionization histories, as well as the two-point level shown in Figures 1 and 2.

In our case, the redshift axis serves as the color channels of the CNNs inputs. This is in contrast to other works, such as Gillet et al. (2018), where the authors use two-dimensional
input with one axis representing evolution along the line-of-sight (LOS). The latter approach may be easier for CNNs to capture the map-space features along the redshift/LOS axis and regress on global parameters such as reionization history and optical depth, because the redshift evolution is included explicitly as one of the axes of the image. However, actual observation data are difficult to reshape into such a two-dimensional format. Furthermore, CNNs conventionally deal with square-shaped images. For these reasons, we choose to use color channels to represent the redshift axis. Figure 4 shows a comparison between a typical input zreion sample and a 21cmFAST sample, which share similar reionization history and optical depth. In contrast to Figure 1, these snapshots have had the wedge filter applied. Since we remove the $k = 0$ mode to preclude referencing an absolute scale, Figure 4 shows variation about the mean temperature of the map. Visually, the simulations outputs of 21cmFAST and zreion are qualitatively comparable. However, there are still important
Figure 5. The pixel distribution of one typical zreion snapshot (left) and one 21cmFAST snapshot (right) used in the respective training sets. These plots show the fluctuation of the brightness temperature $\delta T_T$ normalized to be between 0 and 1. The x-axis corresponds to the temperature value and the y-axis corresponds to the redshift value. The solid black lines are the variance as a function of redshift. The dash black lines label the mean. See Section 2.4 for further discussion.

distinctions, such as the snapshots from 21cmFAST having larger fluctuation scale.

Figure 5 shows the normalized pixel values distribution for each slice in a zreion snapshot and a 21cmFAST snapshot, which is a more quantitative comparison between the two methods. As mentioned above, there are 30 two-dimensional slices in each snapshot representing 30 different redshift values. Since we have remove the $k = 0$ mode, what we show here is essentially the distribution of fluctuations, which are encoded by the color. The solid black line indicates the variance in brightness temperature for a given redshift value, with the dashed line serving as a reference for the mean value. At the beginning of reionization, most of the simulated space is unionized so pixels are concentrated around the mean value. When reionization is partially underway, the variance increases because the ionization state of the volume starts to show large variations. Near the end of reionization, space becomes homogeneous again because most of the volume is ionized. As can be seen, the snapshots produced by zreion has a variance that is larger near the midpoint of reionization, and is smaller at the end. Conversely, for 21cmFAST snapshots the pixel variance stays relatively large at the low redshift.

3. Method

Convolutional neural networks (CNNs; Fukushima & Miyake 1988; LeCun et al. 1999; Krizhevsky et al. 2012) are a family of machine learning algorithms used widely for image recognition tasks. In general, they use a series of convolution filters to capture local features in image-space among all the points in the data. Multiple convolutional layers stacked together allow the network to extract subtle and sophisticated information, such as the non-Gaussian components in the EoR maps. Additionally, convolution layers typically have one or two orders of magnitude fewer parameters compared with dense layers (Li et al. 2019). The CNNs used in this work consist of several convolution layers followed by a few dense layers, which are used to analyze the extracted image information and to regress on $\tau$. This is a typical architecture for predicting parameters from the images, and is similar to the ones in Simonyan & Zisserman (2014), Gillet et al. (2018), and La Plante & Ntampaka (2019).

Following common practice, we use $3 \times 3$ filters for convolutional layers and pair them with $2 \times 2$ max pooling layers with a stride of 2, which are used to cut down the number of parameters (Riesenhuber & Poggio 1999). Between each convolutional and max pooling layer, batch normalization (Ioffe & Szegedy 2015) is adopted to provide regularization. A global average pooling layer is placed to connect convolution layers and the dense layers, which effectively flattens the data output by the convolutional layers. For all of the dense layers except for the final one, we use the “leaky rectified linear unit” (leaky ReLU) as an activation function (Maas et al. 2013). A 20% dropout is adopted to avoid overfitting (Srivastava et al. 2014). All the layers are initialized by the “he normal” technique (He et al. 2015). Each network is trained with a batch size of 32 and a learning rate of $10^{-5}$, and trained to minimize the mean squared error (MSE). The CNN architectures in this work are implemented using the Keras framework (Chollet 2015) with the TensorFlow as the backend (Abadi et al. 2016).

In principle, $\tau$ as a cosmological parameter depends primarily on the reionization history of the universe, and as such is a model-independent parameter. Nevertheless, the differences between zreion and 21cmFAST may lead to different results when using the different snapshots as input training data. In order to understand that impact of underlying models on the machine learning networks, we train three CNNs with zreion, 21cmFAST, and mixed snapshots as training data sets to regress the value of $\tau$, and label them as CNN I to III.
Table 2

| Data set     | CNN I     | CNN II    | CNN III   |
|--------------|-----------|-----------|-----------|
|              | 21cmFAST  | zreion    | mixed     |
| Convolution  | 64 filters| 64 filters| 64 filters|
| layers       | 128 filters| 128 filters| 128 filters|
| Dense layers | 256 filters| 256 filters| 256 filters|
|              | 512 filters| 512 filters| 512 filters|
| Dense layers | 400 neurons| 400 neurons| 400 neurons|
|              | 100 neurons| 50 neurons | 100 neurons|
|              | 20 neurons | 20 neurons | 20 neurons |
| Output layer | 1 neuron  | 1 neuron  | 1 neuron  |

Note. We do not show the max pooling layers and dropout layers here. Full architectures are described in Section 3.

1. CNN trained and validated exclusively on 21cmFAST data (CNN I)
2. CNN trained and validated exclusively on zreion data (CNN II)
3. CNN trained and validated on mixed data (including both zreion and 21cmFAST data) (CNN III)

We make the distinction between “training data”, which are used during the training process to adjust the weights and biases in the network; “validation data”, which are used during the training process for monitoring metrics such as the value of the loss function; and “test data”, which are only used for evaluation after the network has been trained. In the results that follow, we show the performance of the networks on testing data.

An optimization process was carried out by a random grid search to decide on the hyperparameters for CNNs I, II, and III. We varied the number of convolutional layers, the number of filters in the convolutional layers, the number of dense layers, and the number of neurons in these dense layers. We train these networks using a reduced number of epochs to evaluate the validation loss accurately. We identify the “best” architectures, which are shown in Table 2, as the ones that have the best performance on validation data, i.e., have the smallest validation loss, and reinitialize the network to train with more epochs. Here only the loss on the trained model is considered, i.e., 21cmFAST for CNN I, zreion for CNN II and mixed data for CNN III.

If the networks are generally independent of these two semi-numeric reionization models, we expect any CNN we mentioned above, no matter which data set trained on, would work equally well when predicting on test data generated by either zreion or 21cmFAST. If, instead, the CNN does depend on the model used for generating snapshot data, then there would be a clear difference between the prediction results on these two methods for those networks trained on only one of them. In that case, the CNNs are implicitly able to distinguish between zreion and 21cmFAST, even though they are being used to predict on a model-independent parameter \( \tau \). Thus, such a result would indicate that these two methods have distinct features that do not generalize, and furthermore that CNN-based methods are sensitive to these differences during training. This result is not ideal because when CNN-based inference methods are to be deployed on actual telescope data, the underlying physics of reionization will certainly be more complicated than either semi-numeric model individually. In order to achieve robust predictions on astrophysical or cosmological parameters, we would like to train models that perform equally well on the two methods. To achieve such even-handed performance, we need to provide the networks with sufficient motivation to work well on both zreion and 21cmFAST, i.e., to train the networks on data from both models. To demonstrate this, we train the CNN III with a data set that contains equal number of zreion and 21cmFAST snapshots, which is supposed to include all the information represented by these two methods.

4. Results

In this section, we present the prediction results of training our CNN networks described in Section 3. We show the architectures that produce the best predictions for the three CNNs in Table 2. Figure 6 plots the prediction results made by CNN I, which is trained on snapshots from 21cmFAST, on the data from two methods. These data correspond to a testing data set generated by each of these methods, which were not included in the training or validation data. The relative error shown in the figures are calculated by \( |\tau_{\text{true}} - \tau_{\text{predicted}}|/\tau_{\text{true}} \). As can be seen, the network could only produce good predictions on the models they trained on, for which the errors are within 5%. For the other model, the prediction error ranges from a few percent to over 20%, with a large bias at high-value end. Figure 7 shows the prediction for CNN II, which is trained exclusively on zreion. Analogously, this network works well on zreion but has large errors on 21cmFAST. Following this, the results of CNN III are shown in Figure 8, which is trained on a mixed data set containing snapshots from both zreion and 21cmFAST. The errors for this network are all within 5%, with no obvious difference between the two methods.

5. Discussion

5.1. CNN’s Behavior on Two Models

The results shown in Section 4 indicate that although the snapshots generated by two reionization models are very similar by eye, they are nevertheless sufficiently different that inhibit CNN from performing equally well on the both models.
In this section, we compare \textit{zreion} and \textit{21cmFAST} based on the CNN’s behavior by analyzing the loss and comparing their architectures in Table 2. We show the MSE loss on \textit{zreion} and \textit{21cmFAST} respectively during the training for CNNs I, II, and III in Figure 9. During the early stages, the loss on \textit{zreion} and \textit{21cmFAST} are both decreasing, which implies that the networks are initially identifying features that are common to the two models. With additional training epochs,
the loss value for \textit{zreion} and \textit{21cmFAST} starts to deviate, which appear for all the three networks. The different behavior on the two methods indicates that at a certain point, the network starts using specific features from the trained model that are not shared by the other for making predictions. If the goal of the CNN training is to produce a generalized network that performs equally well on \textit{21cmFAST} and \textit{zreion}, it may be possible to use a simpler architecture, or stop training with a relatively small number of epochs. However, based on the fact that the CNN performs poorly on the untrained model and there is a large improvement on the trained one after the point where the loss starts to diverge, we assume the networks could not produce good predictions on neither of two models at that time.

One interesting conclusion is that \textit{21cmFAST} is easier for CNNs to capture features and produce good predictions. In Table 2, it can be seen that the CNN trained on \textit{21cmFAST} has a simpler architecture than that trained on \textit{zreion}. CNN I, which has three convolution layers with 64, 128, and 256 filters, respectively, could produce similarly good results on the training simulation model with CNN II, which has one more layer with 512 filters. What is more, Figure 9 shows that for the network trained on the mixed data set, the loss on \textit{21cmFAST} and the orange line is the loss on \textit{zreion}.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{image1.png}
\caption{Regression results for CNN III, which is trained on both \textit{zreion} and \textit{21cmFAST} data. The orange dots are predictions from an independent \textit{21cmFAST} test data set, and blue dots are from an independent \textit{zreion} test data set.}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{image2.png}
\caption{The MSE loss values for CNN I, II, III on \textit{zreion}, \textit{21cmFAST} data respectively during the training. For all the subplots, the blue line is the loss on \textit{21cmFAST} and the orange line is the loss on \textit{zreion}.}
\end{figure}
decreases faster and ends up smaller than that on zreion. For CNN III, it takes the network about 1500 epochs to give the similar small loss on zreion which is achieved on 21cmFAST at only 1000 epochs. This result is slightly non-intuitive because based on how convolution layers work, we might assume that CNNs would be more sensitive to features implemented in Fourier space.

We also notice different behavior between CNN I on zreion and CNN II on 21cmFAST. When the network is trained on 21cmFAST (the left column in Figure 9), the loss on zreion stays at the same level from some point. While for the network trained on zreion (the center column in Figure 9), it overfits on 21cmFAST data. This might be caused by the different architecture complexity since zreion data need more convolution layers. To investigate this, we use the architecture of CNN I to train zreion and the architecture of CNN II to train 21cmFAST, whose losses are shown in Figure 10. The networks cannot learn further on zreion even with a more complex architecture, and the overfitting on the left plot is not obvious, which rule out the possibility of network complexity causing the discrepancy.

The above analysis supports the idea that these two models share some common features while have some exclusive characteristics, and the general parts are not enough to allow CNNs to make accurate prediction of $\tau$. The distinction on loss might indicate that the relationship between 21 cm map and resulting value of $\tau$ in 21cmFAST is more straightforward than that in zreion. This also implies that machine learning might be a potential method to compare different simulation models and draw comparisons between them.

5.2. Uncertainty Qualification by Monte Carlo Dropout

In this section, we use Monte Carlo Dropout to estimate the uncertainties of CNNs’ results on different semi-numeric methods. Dropout was first proposed as a method to solve the overfitting problem and traditionally is not used during model prediction. However, dropout can also be used as a Bayesian approximation because mathematically neural networks with dropout are equivalent to approximate variational inference in the deep Gaussian process and further serve as an effective method to estimate the uncertainty (Brach et al. 2020; Nair et al. 2020). Such a technique is called Monte Carlo dropout (Gal & Ghahramani 2016), in which a stochastic process is performed for each input sample because dropout effectively introduces random variables in the networks. As a result, we get an ensemble of predictions for the same input data and the standard deviation of these predictions is an estimate of the uncertainty. There are two main sources of uncertainties: aleatoric uncertainties, which represent the inherent property of the data set, and epistemic uncertainties, which correspond to an inadequate understanding of the underlying models (Abdar et al. 2021). The uncertainty caused by the different semi-analytic models falls under epistemic uncertainty, which we seek to quantify here.

We pick 100 zreion snapshots and 100 zreion snapshots from the test data set, and use CNN I, II, III to make 100 predictions on each of the sample, whose standard deviations are plot in Figure 11. It can be seen that when the networks trained exclusively on one simulation method, the uncertainty on the untrained model would be larger across the whole range. And for CNN III, the one which was trained on a mixed data set, the uncertainties for both zreion and 21cmFAST are small. We also show the averaged standard
deviation on the 100 samples ($\sigma_{21\text{cmFAST}}$ and $\sigma_{\text{zreion}}$) for three networks in Table 3. We point out that since the these CNNs networks have different architectures, the uncertainties within the same row should not be compared directly.

### 5.3. Larger Convolution Filters

For studying EoR, large-scale information in 21 cm mapping is scientifically interesting. In a CNN network, deeper convolution layers, between which a pooling operation is performed, deal with larger parts in the original picture, and in principle, we could use deep convolution layers to extract large-scale information. However, small convolution filters still emphasize more about small-scale features. As mentioned before, we use $3 \times 3$ filters to train the networks, which equal to $11.7 \times 11.7 \, h^{-1}\text{Mpc}$ comoving sizes for the first layer. Here we try larger filters, $11 \times 11$ filters, which correspond to $43.0 \times 43.0 \, h^{-1}\text{Mpc}$. Another motivation for using larger filters is that although we have demonstrated CNNs treat 21cmFAST and zreion differently, they might be more similar on larger scales. Training with large filters might give us a network that performs equally well on both of the two methods even when trained on only one of them.

One downside with this approach is that a large convolution filter tends to increase the volatility of loss function. As a result, we have to decrease the learning rate, which requires additional training epochs. We use a learning rate of $10^{-6}$ with the same architecture of CNN II shown in Table 2, and train exclusively on zreion data. Crucially, we change all the convolution filters from $3 \times 3$ to $11 \times 11$.

We plot the loss values for 4000 epochs on zreion and 21cmFAST snapshots in Figure 12, which indicates there is no obvious improvement after 1500 epochs and the loss bottoms out at $10^{-2}$ for both of the two models. We also show the predictions for the trained networks on snapshots from zreion and 21cmFAST. As can be seen, the scatter in the predictions in larger than that in Figure 8. Meanwhile for CNN II, whose loss is shown in the center column of Figure 9, the loss value on zreion keeps going down and ends up less than $10^{-3}$. Hence, with larger convolution filters, the gap between zreion and 21cmFAST becomes narrower, while the performance is worse than that of CNNs with smaller convolution filters. When we use $5 \times 5$ filters, the predictions are better but the difference between two models is still apparent. This implies that for CNNs, zreion and 21cmFAST do share large-scale features, which are better captured by larger convolutional filters. Accordingly, larger filters may be used to generate predictions that are more similar between the two methods, at the cost of worse performance on both of them.

### 5.4. Dependence on Redshift

The comparison in Section 2.3 shows that the difference between 21cmFAST and zreion is more obvious at low redshift. In this section we investigate whether this difference causes the discrepancy in networks’ behavior and if this redshift range plays a more important role for CNNs to make decisions.

To visualize the importance of different redshift values, we swap out certain slices in a snapshot and replace them with the...
counterpart of the sample from the other model. To be more specific, we pick several \((z_{\text{reion}}, 21\text{cmFAST})\) pairs from test data sets and two snapshots in each pair have a similar reionization midpoint \(z_{50}\), reionization duration \(\Delta z\), and optical depth \(\tau\). This selection ensures the different results after swapping mostly come from the substituted slices, rather than distinctions in the ionization history. There are 30 slices in a single snapshot, representing redshift \(6 < z < 15\), and we replace six adjacent slices of them each time, which correspond to data around a given redshift. By feeding these swapped snapshots into CNNs and examining how the prediction values evolve, we could have an idea about which redshift range outweighs others for networks to decide optical depth. In Figure 13, we show that there are different \(z_{50} - \tau\) relations for \(z_{\text{reion}}\) and 21cmFAST. Accordingly, the requirement of using similar \(z_{50}\) and \(\tau\) means we are only allowed to pick samples with \(\tau \sim 0.055\).

In the left panel of Figure 14, we pick five pairs of snapshots and use CNN II to make predictions on these original \(z_{\text{reion}}\) snapshots and 21cmFAST snapshots, whose averaged values are marked by the black dashed line and red dashed line, respectively. We note that here the average could serve as a
marker of all the initial predictions since all the $\tau_{\text{true}}$ are close to 0.055, which makes the picked zreion or 21cmFAST samples have similar $\tau_{\text{pred}}$. Then, we show predicted values for the five swapped 21cmFAST snapshots, which have six adjacent slices from zreion snapshots, which corresponds to data at different redshift values. The predictions are made by CNN II. Right: the predictions for CNN I on five swapped zreion snapshots with six adjacent slices from 21cmFAST data. Each pair of zreion and 21cmFAST samples have similar reionization histories and $\tau$. The black dashed lines mark the average value of predictions on the original zreion samples and the red dashed lines are the average value of predictions on the original 21cmFAST samples.

As a comparison, we zeroing the same adjacent slices in the snapshots and plot the evolution of predicted values in Figure 15, where the red/black dashed lines still indicate the average initial predicted values. The deviation from the original predictions is much larger than that shown in Figure 14, which implies the network learn much information from the other model although this is not sufficient to make accurate regression. When the slices at low redshift are set to zero, the deviation from the “original” predictions is generally larger than that on at high redshift, which again supports the idea that low-redshift data is more crucial for the networks to predict $\tau$.

As another test of the networks' dependency on each redshift range, we train CNNs exclusively on low redshift part ($z < 8$, 11 slices) or high redshift part ($z > 8$, 19 slices) of the snapshots, whose results are plot in Figure 16. Here we use the
exact same architecture, learning rate, and number of training epochs for both CNN I and CNN II to train 21cmFAST and zreion data, so the only thing changed is the input data shape. Figure 16 shows that when we only include low-redshift data in the training, the networks produce almost the same results with those in Figures 6 and 7. On the other hand, when only high-redshift data are used, the performance gets worse. This suggests a similar conclusion as for Figure 14, which is that the information at low redshift plays a more important role for the CNNs to make the regression for both semi-numeric models used.

5.5. Visualizing CNN Feature Extraction

Understanding the CNN’s interpretation of input data might provide some physical insight into the question itself. There are several methods that have been developed for this purpose, one of which is known as activation maximization (Mahendran & Vedaldi 2015; La Plante & Ntampaka 2019; Billings et al. 2021). This technique transforms a random input image into one that maximizes the response of the chosen neuron through gradient ascent. The output result indicates which features are important to the network in order to discriminate between different parameter values. In this work, we use another way to visualize feature extraction called the saliency map (Tong et al. 2010; Jiang et al. 2013), in which we calculate how much the prediction value for a trained network varies when each pixel in a given image is perturbed. In other words, this map visualizes the gradient of the prediction with respect to the changes in the pixel of the input data. Similar to the activation maximization technique, saliency maps were first proposed to analyze classification in CNNs, showing the key features a network

---

**Figure 16.** The regression results on zreion and 21cmFAST when the training data set includes information only at low redshift values ($6 < z < 8$, 11 slices) or high redshift values ($8 < z < 15$, 19 slices). The upper two plots are results for networks trained exclusively on zreion data, which have the same architecture with CNN II except for the input data shape, and the lower two plots are results for networks trained exclusively on 21cmFAST data, which have the same architecture with CNN I except for the input size.
Figure 17. The saliency maps of CNN I, II, III, and the corresponding snapshots. The first row show the central slice ($z \sim 9$) of 21cmFAST snapshot (panel a) and zreion snapshot (panel b) used to generate the saliency maps. The panels (c)/(e)/(g) are the saliency maps on the 21cmFAST data made by CNN I/II/III, respectively, and the panels (d)/(f)/(g) are the maps on the zreion snapshot made by CNN I/II/III. The saliency is normalized for each map and the maximum values of the given pixel across all the channels are coded by the color. We apply 10% Gaussian noise to the original snapshots for 50 samples and average the gradients to obtain these saliency maps.
uses to identify a specific class. In our regression case, the map indicates the differences between the snapshots with large and small \( \tau \) values captured by the CNNs.

We calculated the saliency maps for a \( z_{\text{reion}} \) and 21cmFAST snapshot using CNN I, II and III, which are shown in Figure 17. Panel (a) is the central slice (\( \sim 9 \)) of the 21cmFAST snapshot used to generated the maps in (c), (e), (g), and panel (b) is the central slice of the \( z_{\text{reion}} \) snapshot for maps (d), (f), (h). The \( (\tau_{\text{true}}, \tau_{\text{CNNI}}, \tau_{\text{CNNII}}, \tau_{\text{CNNIII}}) \) for the 21cmFAST and \( z_{\text{reion}} \) samples are (0.067, 0.065, 0.074, 0.066) and (0.053, 0.054, 0.053, 0.052), respectively. The second/third/fourth rows are saliency maps for CNN I/II/III, respectively. To improve the maps’ behavior, we use the technique of “SmoothGrad” (Smilkov et al. 2017) to reduce the visual noise. We add Gaussian noise \( \mathcal{N}(0, \sigma^2) \) on the original snapshots and calculate the average gradients of 50 samples, where \( \sigma = 0.1(x_{\text{max}} - x_{\text{min}}) \) and the x is the pixel values in the input data. The saliency map is normalized to range from 0 to 1 and the color is coded by the maximum values of the given pixel across all the channels.

The gradient of CNN I (panels (c) and (d)) is generally larger than that of others, and the distribution of power is more isotropic and accentuates more small-scale features. The maps for CNN II (panels (e) and (f)) have large-scale patchy structure, and the contrasts of large and small scales are more obvious, which also appears in the maps for CNN III (panels (g) and (h)). It is clear to see that the same snapshot produces distinct saliency maps for each network, which implies the individual networks use different sets of spatial features to make their determination. Interestingly, there is no obvious correlation between the four plots within a column: the voids and patches that appear in snapshots do not coincide with those that appear in saliency maps, ruling out the possibility that the structures in the maps correspond to the positions of ionized bubbles in snapshots.

Furthermore, the saliency map for a given network identifies different features for \( z_{\text{reion}} \) and 21cmFAST. Since the saliency map works better for samples with accurate predictions, we concentrate our analysis on panels (g) and (h), which are the saliency maps for CNN III on the two methods. Although the sizes of features are similar in (g) and (h), the contrasts between patches and voids are different, which indicate that even for the CNN trained on a mixed data set, it can still implicitly distinguish 21cmFAST and \( z_{\text{reion}} \). One possible interpretation of this result is that the good performance of CNN III is due to learning the relevant details for both simulations, rather than the shared or common features. If this is the case, then making a more general network might be possible by using adversarial techniques to train a network that is agnostic to the particular details. We save such investigations for future work.

6. Conclusion

In this work, we investigate the extent to which training CNNs for reionization applications depends on the underlying semi-numeric model used to generate input snapshots. We use two semi-numeric methods: 21cmFAST and \( z_{\text{reion}} \), and find that CNNs are indeed sensitive to the input model. In general, a network trained on data from one model does not make accurate predictions for data generated by the other one, even though the parameter \( \tau \) is in principle model independent. We find that using a training data set that contains snapshots from both simulation methods leads to satisfactory results when training, though the CNN still implicitly can tell the difference between the snapshots from the two models, as suggested by the saliency maps used to understand the networks’ behavior.

For future analysis, we can make use of adversarial techniques for reducing the dependence of the trained networks on features unique to a single simulation. For example, by using a discriminator network which attempts to distinguish whether particular input data was generated by 21cmFAST or \( z_{\text{reion}} \), it may be possible to train a network that is not overly reliant on particular details for one simulation over another. Such an approach may be a way to build a “general” network that does not over-learn the details of any one network and does not require an arbitrarily large set of semi-numeric models to yield robust estimates. Such a general network is essential for making predictions on model-independent parameters, and can help ensure that results are not biased.

Although the particular applications in this work use CNNs and map-space data to investigate the dependence on different semi-numeric models, the distinctions are also be present in Fourier space, and hence appear in the power spectrum as well (see Figure 2). When interpreting the results of upcoming experiments like HERA or the SKA, it is important to understand the impact that the particular simulation framework used to model the signal affects the systematic errors associated with the inferred parameters. As such, using a variety of simulation frameworks, like the analysis performed in The HERA Collaboration et al. (2021b), is essential for building confidence in the ultimate results.

We thank James Aguirre, Adrian Liu, Jordan Mirocha, Julian Muñoz, Michelle Ntampaka, and Jonathan Pober for insightful comments and useful suggestions on this work, and we thank Emma Bloomfield for invaluable feedback on the manuscript. P.L. acknowledges support from the Berkeley Center for Cosmological Physics. This material is based upon work supported by the National Science Foundation under grant Nos. 1636646 and 1836019 and institutional support from the HERA collaboration partners. This research is funded in part by the Gordon and Betty Moore Foundation through grant GBMF5212 to the Massachusetts Institute of Technology. This work used the Extreme Science and Engineering Discovery
Environment (XSEDE), which is supported by National Science Foundation grant No. ACI-1548562 (Towns et al. 2014). Specifically, it used the Bridges-2 system, which is supported by NSF award No. ACI-1445606, at the Pittsburgh Supercomputing Center (PSC, Nystrom et al. 2015).

Appendix

dependence on cosmological parameters

Because there are finite errors associated with the underlying cosmological parameters, we investigate the extent to which our CNNs depend on these errors. To do this, we generate 200 21cmFAST input snapshots with different cosmological parameters \((H_0, \Omega_m, \Omega_b, \sigma_8, n_s)\), whose ranges are listed in Table 4. We sample each parameter according to Gaussian distributions, whose fiducial set is the is the one used in the training data set and variances are consistent with the errors in Planck Collaboration et al. (2016). 100 snapshots have the same astrophysical parameters while the other 100 snapshots have different astrophysical parameters. We use CNN-I, which is trained on our fiducial cosmology model, to predict optical depth on them and present our results in Figure 18.

As can be seen, the optical depth is mainly determined by astrophysical parameters, since it is primarily determined on the ionization fraction as a function of redshift \(x_i(z)\), and only weakly depends on the cosmological parameters. Figure 18 shows that the snapshots with the same astrophysical parameters but varied cosmological parameters (green points) fall into a relatively narrow range of \(\tau\) values, while those with different astrophysics span a much broader range. We also see that CNNs still produce good results even when the underlying cosmological parameters differ. The predictions show slightly more scatter than that on snapshots with the same fiducial cosmology, but are still close to the true values. Thus, the additional error associated with changing cosmological parameters is sub-dominant to the error associated with changing the astrophysical parameters. Therefore, for the real data, whose cosmology model might have some deviation from the one used in our training data set, our CNNs should still perform well.

![Figure 18. The predictions of CNN I on 21cmFAST snapshots in the test data set (blue dots), i.e., with the fiducial cosmology model and different astrophysical parameters, the 21cmFAST snapshots with the different cosmological and astrophysical parameters (orange dots), and the 21cmFAST snapshots with the different cosmological while the same astrophysical parameters (green dots).](image-url)
