Draft version May 16, 2017
Preprint typeset using LATEX style AASTeX6 v. 1.0

EMULATING SIMULATIONS OF COSMIC DAWN FOR 21 cm POWER SPECTRUM CONSTRAINTS ON COSMOLOGY, REIONIZATION, AND X-RAY HEATING

Nicholas S. Kern[1,⋆], Adrian Liu[1,†], Aaron R. Parsons[1], Andrei Mesinger[2], Bradley Greig[2]

[1]Department of Astronomy and Radio Astronomy Laboratory, University of California Berkeley, Berkeley, CA 94720, USA
[2]Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126 Pisa, Italy

ABSTRACT

Current and upcoming radio interferometric experiments are aiming to make a statistical characterization of the high-redshift 21 cm fluctuation signal spanning the hydrogen reionization and X-ray heating epochs of the universe. However, connecting 21 cm statistics to underlying physical parameters is complicated by the theoretical challenge of modeling the relevant physics at computational speeds quick enough to enable exploration of the high dimensional and weakly constrained parameter space. In this work, we use machine learning algorithms to build a fast emulator that mimics expensive simulations of the 21 cm signal across a wide parameter space to high precision. We embed our emulator within a Markov-Chain Monte Carlo framework, enabling it to explore the posterior distribution over a large number of model parameters, including those that govern the Epoch of Reionization, the Epoch of X-ray Heating, and cosmology. As a worked example, we use our emulator to present an updated parameter constraint forecast for the Hydrogen Epoch of Reionization Array experiment, showing that its characterization of a fiducial 21 cm power spectrum will considerably narrow the allowed parameter space of reionization and heating parameters, and could help strengthen *Planck*'s constraints on σ_8. We provide both our generalized emulator code and its implementation specifically for 21 cm parameter constraints as publicly available software.

1. INTRODUCTION

Cosmic Dawn is a fundamental milestone in our universe's history, and marks the era when the first generation of stars and galaxies formed, ending the Dark Ages that followed recombination. These first luminous sources eventually reionized the neutral hydrogen filling the Intergalactic Medium (IGM) during the Epoch of Reionization (EoR). For all of their implications on the formation and evolution of the first galaxies and compact objects, the EoR and Cosmic Dawn remain a relatively unexplored portion of our universe's history. However, in recent years there have been significant observational advances in our understanding of this epoch. These include Cosmic Microwave Background (CMB) measurements that constrain the timing of reionization (Planck Collaboration et al. 2016; Zahn et al. 2012; Mesinger et al. 2012); direct measurements of the bright end of the ultraviolet luminosity function of galaxies up to $z \sim 10$, which constrain some of the sources of reionization (Bouwens et al. 2015; Finkelstein et al. 2015; Livermore et al. 2017), and Lyman-α absorption studies that place limits on the end of reionization (Fan et al. 2006; Becker et al. 2015; McGreer et al. 2015).

Another promising class of probes are radio interferometer intensity mapping experiments targeting the 21 cm hyperfine transition from neutral hydrogen (Hogan & Rees 1979; Scott & Rees 1990; Madau et al. 1997; Tozzi et al. 2000). Such experiments aim to tomographically map out the distribution, thermal state and ionization state of neutral hydrogen in the IGM throughout Cosmic Dawn, and are potentially the only *direct* probes of the epochs relevant to the formation of the first generations of stars, galaxies, stellar-mass black holes, supernovae, and quasars. For reviews of 21 cm cosmology, see e.g. Furlanetto et al. (2006); Morales & Wyithe (2010); Pritchard & Loeb (2012); Loeb & Furlanetto (2013); Mesinger (2016). While 21 cm cosmology faces formidable observational challenges, recent years have seen significant advances toward resolving issues of optimal array design (Beardsley et al. 2012; Parsons et al. 2012b; Greig et al. 2015; Dillon & Parsons 2016), internal systematics (Ewall-Wice et al. 2016c; Barry et al. 2016; Patil et al. 2016; Ewall-Wice et al. 2016a), and astrophysical foreground mitigation (Datta et al. 2010; Morales et al. 2012; Vedantham et al. 2012; Parsons et al. 2012b; Trott et al. 2012; Chapman et al. 2012, 2013; Thyagarajan et al. 2013; Pober et al. 2013a; Liu et al. 2014a,b; Switzer & Liu 2014; Wolz et al. 2014; Moore et al. 2015; Thyagarajan et al. 2015a,b; Asad et al. 2015; Chapman et al. 2016; Pober et al. 2016; Kohn et al. 2016; Liu & Parsons 2016). Increasingly competitive upper limits have been placed on the red-

[⋆] nkern@berkeley.edu

[†] Hubble Fellow

shifted 21 cm signal, using instruments such as the Donald C. Backer Precision Array for Probing the Epoch of Reionization (PAPER; Parsons et al. 2014; Jacobs et al. 2015; Ali et al. 2015), the Giant Metrewave Radio Telescope (GMRT; Paciga et al. 2013), the Murchinson Widefield Array (MWA; Dillon et al. 2014, 2015; Ewall-Wice et al. 2016c; Beardsley et al. 2016), and the Low Frequency Array (LOFAR; Vedantham et al. 2015; Patil et al. 2017). Many of these upper limits are stringent enough to be scientifically interesting, and have typically ruled out extremely cold reionization scenarios (Parsons et al. 2014; Pober et al. 2015; Greig et al. 2016). As these experiments continue to be expanded and second-generation experiments, such as the Hydrogen Epoch of Reionization Array[1] (HERA; DeBoer et al. 2017) and the Square Kilometer Array (SKA; Koopmans et al. 2015), begin commissioning and data processing, a first positive detection of the cosmological 21 cm signal will soon be within reach.

Following a first detection, instruments such as HERA are expected to make high signal-to-noise measurements of the spatial power spectrum of 21 cm brightness temperature fluctuations. Previous studies have shown that such measurements would place stringent constraints on parameters governing the EoR and Epoch of X-ray Heating (EoH) (Pober et al. 2014; Liu & Parsons 2016; Ewall-Wice et al. 2016b), as well as on fundamental cosmological parameters when jointly fit with *Planck* data (McQuinn et al. 2006; Mao et al. 2008; Barger et al. 2009; Clesse et al. 2012; Liu et al. 2016). However, most of these forecasting studies have been limited in at least one of two ways: they have either bypassed full parameter space explorations by employing the Fisher Matrix formalism, or they have relied on simplified parameterizations of the 21 cm signal that may not be appropriate for describing real observations. Thus far, the only method capable of systematically exploring the EoR parameter space is `21CMMC` (Greig & Mesinger 2015), which combined an optimized version of the semi-numerical simulation `21cmFAST` (Mesinger et al. 2011) with an MCMC sampler. This was used to connect upper limits from PAPER to theoretical models (Greig et al. 2016) and to synthesize constraints set by complementary EoR probes (Greig & Mesinger 2017b). However, these studies were limited to $z < 10$, because at higher redshifts the inhomogenous heating of the IGM by X-rays becomes important (Kuhlen et al. 2006; Pritchard & Furlanetto 2007; Warszawski et al. 2009; Mesinger et al. 2013; Pacucci et al. 2014; Fialkov et al. 2014a,b; Fialkov & Barkana 2014; Ghara et al. 2015), and computing it slows down the simulation runtime considerably. As a quantitative illustration, consider `21cmFAST`, which takes $\sim$ 24 hours to run on a single core when computing IGM heating. A parameter constraint analysis with 100 MCMC-chains each evaluated for 10^3 steps would take 3 years to run on a 100-core computing cluster, rendering it intractable. This is a problem that must be solved in order for 21 cm measurements to place rigorous constraints on theoretical models.

One solution is to optimize the simulations to make them run faster. This was in fact recently accomplished for `21cmFAST` by Greig & Mesinger (2017a), who were able to MCMC `21cmFAST` over EoR and EoH parameters; however, with the inclusion of cosmological parameters this is pushed out of the realm of feasibility. Furthermore, even with detailed optimization, more sophisticated numerical simulations are unlikely to be feasible for MCMC in the near future. Faced with this daunting challenge, one approach is to abandon MCMC parameter fitting altogether. This was explored recently by Shimabukuro & Semelin (2017), who showed that promising results could be obtained using artificial neural networks. If one desires detailed information on constraint uncertainties and parameter degeneracies, however, one must turn to an MCMC framework.

Another solution to the aforementioned problem is to use machine learning algorithms to build surrogate models for the behavior of the expensive simulation. The collection of surrogate models, called an emulator, mimics the simulation across the space of its input parameters. After training the emulator over a pre-computed training set, one can discard the simulation entirely and use the emulator in the MCMC sampler to produce parameter constraints. The speed of the emulator depends on the complexity of the surrogate models, but it is generally many orders of magnitude faster to evaluate than the original simulation. This technique is known as emulation, and has recently taken hold in the astrophysics literature to produce parameter constraints with expensive simulations. Examples include emulation of N-body simulations of the matter power spectrum (Heitmann et al. 2006; Habib et al. 2007; Heitmann et al. 2009; Schneider et al. 2011), simulations of the Cosmic Microwave Background power spectrum (Fendt & Wandelt 2007; Aslanyan et al. 2015), simulations of weak lensing (Petri et al. 2015), stellar spectra libraries (Czekala et al. 2015), and numerical relativity waveforms (Field et al. 2014). In short, emulators allow us to produce parameter constraints with simulations that are otherwise unusable for such purposes. Another crucial benefit of emulators is their repeatability: once we have put in the computational resources and time to build the training set, if we change our measurement covariance matrix or add more data to our observations, re-running the MCMC chains with an emulator for updated fits is extremely quick. Even for semi-numerical simulations that are brought into the realm of MCMC-feasibility via optimization techniques, having to repeat an MCMC analysis many times may be computationally prohibitive.

In preparation for observations from upcoming 21 cm experiments, we have built a fast and accurate emulator for simulations of Cosmic Dawn. Embedding it within an MCMC framework, we present updated forecasts on

[1] http://reionization.org/

the constraints that a 21 cm power spectrum experiment like HERA will place on EoR & EoH astrophysical parameters and now also include ΛCDM base cosmological parameters. It is important to note that the emulator algorithm we present here is not tied to any specific model of Cosmic Dawn. Although we will proceed using a particular simulation of Cosmic Dawn, we could in principle repeat these calculations using an entire suite of various simulations with only minor changes to our procedure. We provide a generalized implementation of our emulator algorithm in a publicly-available Python package called `emupy`.[2] This base package can be used to emulate any dataset and is not specific to 21 cm cosmology. We also provide our implementation of the emulator code specific to 21 cm —including our 21 cm likelihood function, sensitivity forecasts and simulation training sets— in a publicly-available Python package called `pycape`.[3]

The rest of this paper is organized as follows. In Section 2 we provide a detailed overview of our emulator algorithm. In Section 3 we discuss our Cosmic Dawn simulation and its model parameterization. In Section 4 we discuss observational systematics for the upcoming HERA experiment and forecast the ability of HERA to constrain astrophysical and cosmological parameters, and in Section 5 we provide performance benchmarks for further validation of the emulator algorithm. We summarize our conclusions in Section 6.

2. BUILDING THE EMULATOR

At the most basic level, emulation is a combination of three major steps: (i) building a training set, (ii) regressing for analytic functions that mimic the training set data and (iii) evaluating those functions at desired interpolation points and accounting for interpolation error. The emulator itself is then just the collection of these functions, which describe the overall behavior of our simulation. To produce parameter constraints, we simply substitute the simulation with the emulator in our likelihood function, attach it to our MCMC sampler, and let the sampler explore the posterior distribution across our model parameter space. In the following sections, we describe the various steps that go into building such an emulator, which allows us to produce parameter constraints using simulations that would otherwise be either too computationally expensive or take too long to run iteratively.

2.1. *Training Set Design*

To emulate the behavior of a simulation, we first require a training set of simulation outputs spanning our N dimensional parameter space, with each sample corresponding to a unique choice of parameter values $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_N\}$. These constitute the tunable model parameters of our simulation, which for example could be cosmological parameters like σ_8 or H_0. Deciding on a training set "design" is the way in which we optimize our emulation, using as few samples as possible to cover the parameter space while retaining a pre-determined level of fidelity dictated by one's science goal. This is particularly crucial for computationally expensive simulations because the construction of the training set will be the most dominant source of overhead.

Previous work applying emulators to astrophysical problems have experimented with different sampling designs. Promising techniques have included variants of space filling designs like the Latin-Hypercube (LH) design, which seek to maximize the minimum distance between any two points and to produce uniform sampling densities when projected onto any one dimension. Most studies have focused on rectangular LH designs, which have been shown to work well for Gaussian Process-based emulators (Heitmann et al. 2009). However, such studies have generally been limited to no more than five cosmological parameters, which furthermore are already reasonably constrained by prior datasets. In contrast, our present application will involve eleven parameters, six of which are effectively unconstrained.

Of particular concern is the "curse of dimensionality", or the fact that a parameter space volume depends exponentially on its dimensionality. If we would like to evenly sample the parameter space with our training samples, then the number of samples we need to generate will also depend exponentially on the dimensionality of the space. One way around this is to impose a prior on our parameter space, specifically a Gaussian or spherical top-hat prior. This allows us to ignore sampling in the corners of the hypercube where the prior distribution has very small probability. In low dimensional spaces, cutting corners only marginally helps us; in two dimensions, for example, the area of a square is only $4/\pi$ greater than the area of a circumscribed circle. In ten dimensions, however, the hypervolume of a hypercube is 400 times that of its circumscribed hypersphere. In eleven dimensions this increases to over a factor of 1000. This means that if we choose to restrict ourselves to a hypersphere instead of a hypercube in an eleven dimensional space, we have reduced the hypervolume we need to cover with the training set by over three orders of magnitude. Schneider et al. (2011) investigated the benefits of this technique, specifically applying the Fisher Matrix formalism to inform the size of the hypersphere, which they call Latin-Hypercube Sampling Fisher Sphere (LHSFS). This technique works well in the limit that we already have relatively good prior distributions on our parameters. For parameters that are highly unconstrained, building a training set spanning the entire parameter range may be infeasible and we may have to turn to gradient descent algorithms to locate areas of high probability before we emulate.

For the parameter constraint forecast we present in Section 4.3, we start with a large but coarse rectan-

[2] https://github.com/nkern/emupy

[3] https://github.com/nkern/pycape

gular LH design spanning a wide range in parameter values. We emulate at a highly approximate level and use the MCMC sampler to roughly locate the region of high probability in parameter space. We then lay down additional training samples from a wide but spherical Gaussian distribution located in the preferred region to further refine our estimate of the maximum a posteriori (MAP) point. With a refined MAP estimate, we lay down a final, highly clustered training set drawn from a combination of a Gaussian distribution and spherical LH design to emulate at a higher precision and characterize the posterior distribution about the MAP (Section 4.4). The extent of this spherical training set is informed by a Fisher Matrix forecast of the approximate bounds of the posterior distribution around the estimated MAP.

We should note that emulator studies have typically focused on building a training set by iterating an extremely expensive numerical simulation on the order of $10-100$ times, thus allowing for the exploration of a handful of model parameters. This is indeed one of the primary benefits of emulators: they allow us to MCMC with simulations that by themselves would be unthinkable to use for direct MCMC even under optimistic optimization schemes. However, as we have discussed above, our present application requires the exploration of more than just a few handful of parameters. And since one's ability to explore high dimensional spaces hinges on the ability to generate a large number of training samples, it is often impractical to train on expensive numerical simulations. Semi-numerical simulations provide a more practical alternative for generating training samples, possibly at the expense of self-consistently including all the "correct" physics. For the study of Cosmic Dawn and EoR, though, we are faced with a situation where the dominant uncertainty on the power spectrum comes not from differences between semi-numerical and full numerical simulations, but in the vast range of the highly unconstrained model parameter space (c.f., Mesinger et al. 2013). Therefore, in this work, we focus on emulating a semi-numerical simulation (specifically `21cmFAST`) so that we can produce a larger training set and push to higher dimensional spaces (>10) compared to what has previously been explored with emulators.

2.2. *Data Compression*

After constructing a training set, our next task is to decide on which simulation data products to emulate over the high dimensional parameter space. Let us define each number that our simulation outputs as a single datum called d, which in our case will be the 21 cm power spectrum, Δ^2_{21}, at a specific k mode and a specific redshift z.[4] Because the power spectra are non-negative quantities, we will hereafter work with the log-transformed data. For example, we might choose our first data element as $\mathrm{d}_1 = \ln\Delta^2_{21}(k = 0.1$ h Mpc$^{-1}, z = 10.0)$. We then take all n simulation outputs we would like to emulate and concatenate them into a single column vector,

$$\mathbf{d} = \begin{pmatrix} \mathrm{d}_1 \\ \mathrm{d}_2 \\ \vdots \\ \mathrm{d}_n \end{pmatrix}, \tag{1}$$

which we call a data vector. Suppose our training set consists of m_{tr} samples scattered across parameter space, each having its own data vector. Hereafter, we will index individual data vectors across the training samples $\{1, 2, \ldots, m_{\mathrm{tr}}\}$ with upper index j such that the data vector from the j^{th} training sample is identified as $\mathbf{d}^j$, located in parameter space at point $\boldsymbol{\theta}^j$. We will also index individual data elements across the data outputs $\{1, 2, \ldots, n\}$ with lower index i, such that the i^{th} data output is identified as d_i. The i^{th} data output from the j^{th} training sample is therefore uniquely identified as d_i^j.

Under the standard emulator algorithm, each data output, d_i, requires its own emulating function or predictive model. If we are only interested in a handful of outputs, then constructing an emulating function for each data output (i.e., direct emulation) is typically not hard. However, we may wish to emulate upwards of hundreds of data outputs, say for example the 21 cm power spectrum at dozens of k modes over dozens of individual redshifts, in which case this process becomes extremely complex. One way we can reduce this complexity is to compress our data. Instead of performing an element-by-element emulation of the data vectors, we may take advantage of the fact that different components of a data vector will tend to be correlated. For example, with the smoothness of most power spectra, neighboring k and z bins will be highly correlated (example 21 cm power spectra are shown in Figure 3). There are thus fewer independent degrees of freedom than there are components in a data vector. This is the idea behind data compression techniques such as Principal Component Analysis (PCA), which seeks to construct a set of principal components (PCs) that linearly sum to equal our data. Transforming to a new basis of independent modes thus constitutes a form of information compression, reducing the number of data points that must be emulated. Hereafter we will use the term principal component and eigenmode interchangeably.

To construct the principal components, we begin by taking the covariance of our training data, since it captures the typical ways in which the data vary over the parameter space. We also center the data (i.e., subtract the mean) and rescale the data (i.e., divide by a constant) such that the covariance is given by

$$\mathbf{C} \equiv \left\langle \mathbf{R}^{-1}\left(\mathbf{d}-\overline{\mathbf{d}}\right)\left(\mathbf{d}-\overline{\mathbf{d}}\right)^T\mathbf{R}^{-1}\right\rangle \tag{2}$$

where $\overline{\mathbf{d}}$ is a vector containing the average of each data output across the training set, $\mathbf{R}$ is a diagonal $n \times n$

[4] See Equation 15 for a formal definition of Δ^2_{21}.

matrix containing our scaling constants, and the outer angle brackets $\langle \dots \rangle$ represent an average over all $m_{\rm tr}$ samples in the training set. The principal components are then found by performing an eigen decomposition of the covariance matrix, given as

$$\mathbf{C}\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Lambda}, \tag{3}$$

where $\boldsymbol{\Phi}$ is an $n \times n$ matrix with each column representing one of the n orthogonal eigenmodes (or principal components), and $\boldsymbol{\Lambda}$ is a diagonal matrix containing their corresponding eigenvalues. We can think of the eigenmode matrix $\boldsymbol{\Phi}$ as a linear transformation from the basis of our centered and scaled data to a more optimal basis, given as

$$\mathbf{w}^j = \boldsymbol{\Phi}^T \left[\mathbf{R}^{-1}(\mathbf{d}^j - \overline{\mathbf{d}}) \right], \tag{4}$$

where $\mathbf{w}$ is our data expressed in the new basis. This basis partitions our data into mutually exclusive, uncorrelated modes. Indeed, the covariance of our data in this basis is

$$\langle \mathbf{w}\mathbf{w}^T \rangle = \boldsymbol{\Lambda}, \tag{5}$$

i.e., our eigenvalue matrix from before, which is diagonal. We can rearrange Equation 4 into an expression for our original data vector, given as

$$\mathbf{d}^j = \overline{\mathbf{d}} + \mathbf{R}\boldsymbol{\Phi}\mathbf{w}^j, \tag{6}$$

where because $\boldsymbol{\Phi}$ is real and unitary, its inverse is equal to its transpose. This gives us insight as to why the $\mathbf{w}$ vectors—the data expressed in the new basis—are called the eigenmode weights: to reconstruct our original data, we need to multiply our eigenmode matrix by an appropriate set of weights, $\mathbf{w}$, and then undo our initial scaling and centering.

Although we have expressed our data in a new basis, we have not yet compressed the data because the length of $\mathbf{w}^j$, like $\mathbf{d}^j$, is n, meaning we are still using n numbers to describe our data. However, one benefit of working in our new basis is that we need not use all n eigenmodes to reconstruct our data vector. If we column-sort the n eigenmodes in $\boldsymbol{\Phi}$ by their eigenvalues, keep those with the top M eigenvalues and truncate the rest, we can approximately recover our original data vector as

$$\mathbf{d}^j \approx \overline{\mathbf{d}} + \mathbf{R}\boldsymbol{\Phi}\mathbf{w}^j, \tag{7}$$

where $\boldsymbol{\Phi}$ is now defined as the $n \times M$ truncated eigenmode matrix, and $\mathbf{w}^j$ is now defined as the length-M column vector where we have similarly sorted and then truncated the weights corresponding to the truncated eigenmodes. *Hereafter, we will use* $\boldsymbol{\Phi}$ *and* $\mathbf{w}$ *to exclusively mean the eigenmode matrix and weight vector respectively after truncation.* Because we are now expressing our data with M numbers where $M < n$, we have compressed our data by a factor of n/M. The precision of this approximation depends on the inherent complexity of the training set and the number of eigenmodes we choose to keep. For our use-case, we typically achieve percent-level precision with an order-of-magnitude of compression ($n/M \sim 10$).

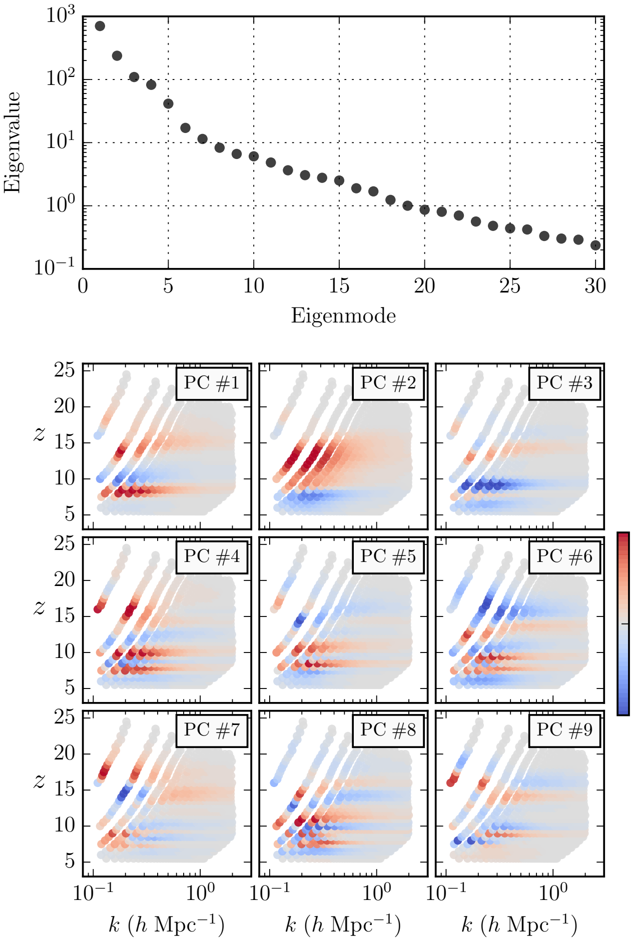

Figure 1. **Top**: Scree plot showing the eigenvalues of thirty principal components formed from training data of $\ln \Delta^2_{21}$. **Bottom**: The first nine principal components of the power spectrum data at each unique k-z combination. The color scale is artificially normalized to [-1, 1] for easier comparison.

In the case where our scaling matrix, $\mathbf{R}$, is the identity matrix, the formalism described above is the standard Principal Component Analysis (PCA) or Karhunen-Loève Transform (KLT). This means that PCA and KLT operate directly on the data covariance matrix formed from our unscaled data. However, not all of the k modes of our power spectrum data will be measured to the same fidelity by our experiment. For the k modes where our experiment will deliver higher precision measurements, our data compression technique should also yield higher precision data reconstructions. To do this, we can incorporate a non-identity scaling matrix, $\mathbf{R}$, which can take an arbitrary form such that we produce eigenmodes that are desirable for the given task at hand. A natural choice would be to use the noise (or, in general, the experimental errors) of our instrument. This has the effect of downweighting portions of the data where our measurements will have minimal influence due to larger experimental errors, and conversely upweights the parts of the data with the smallest experimental errors. In the context of our worked example, we also include a whitening term in our scaling matrix, σ_{d}, which is the standard deviation of the unscaled and centered data. After experimenting with various scaling matrices, we find a scaling matrix of $\mathrm{R}_{ij} = \delta_{ij}\sigma_{\mathrm{d}}^{i}[\sigma_i/\exp(\overline{\mathrm{d}}_i)]^{1/2}$ to work well, where δ_{ij} is the Kronecker delta, σ are the observational errors, and $\exp(\overline{\mathrm{d}})$ is the average of the training set data, expressed in linear (not logarithmic) space.

An example set of principal components formed from our training data is shown in Figure 1, where we display the first nine principal components (eigenmodes) of the log, centered and scaled Δ^2_{21} training data. We discuss the simulation used to generate this training data in Section 3. The amplitude of the PCs have been artificially normalized to unity for easier comparison. We find in general that at a particular redshift, an individual PC tends to be smooth and positively correlated along k, and at a particular k shows negative and positive correlations across redshift. This is a reflection of the underlying smoothness of the power spectra across k, and the fact that physical processes such as reionization, X-ray heating and Lyman-α coupling tend to produce redshift-dependent peaks and troughs in the power spectrum (see e.g., Figure 3). The reason why the PCs lose strength at high k is because our rescaling matrix $\mathbf{R}$ downweights our data covariance matrix at high k. As we will see in Section 4.1, the bulk of a 21 cm experiment's sensitivity to the power spectrum is located at lower k.

2.3. *Gaussian Process Regression*

For the purposes of emulation, we are not interested in merely reconstructing our original training set data at their corresponding points in parameter space $\boldsymbol{\theta}^j$, but are interested in constructing a prediction of a *new* data vector, $\mathbf{d}^{\mathrm{new}}$, at a new position in our parameter space, $\boldsymbol{\theta}^{\mathrm{new}}$. We can construct a prediction of the data vector at this new point in parameter space by evaluating Equation 7 with $\mathbf{w}^{\mathrm{new}}$; however, we do not know this weight vector a priori. To estimate it at any point in parameter space, we require a predictive function for each element of $\mathbf{w}$ spanning the entire parameter space. Put plainly, we need to interpolate $\mathbf{w}$ over our parameter space. To do this, we adopt a Gaussian Process (GP) model, which is a highly flexible and non-parametric regressor.

A GP is fully specified by its mean function and covariance kernel. The GP mean function can be thought of as the global trend of the data, while its covariance kernel describes the correlated random Gaussian fluctuations about the mean trend. In practice, because we center our data about zero before constructing the principal components and their weights (Equation 2), we set our mean function to be identically zero. For the covariance kernel we employ a standard squared-exponential kernel, which is fully stationary, infinitely differentiable, produces smooth realizations of a correlated random Gaussian field, and is given in multivariate form as

$$k(\boldsymbol{\theta},\boldsymbol{\theta}'|\mathbf{L}) = \sigma_{\mathrm{A}}^2 \cdot \exp\left[-\frac{1}{2}\left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)^T \mathbf{L}^{-2}\left(\boldsymbol{\theta}-\boldsymbol{\theta}'\right)\right], \tag{8}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ denote two position vectors in our parameter space, $\mathbf{L}$ is a diagonal matrix containing the characteristic scale length of correlations ℓ across each parameter, and σ_{A} is the characteristic amplitude of the covariance. $\mathbf{L}$ is a tunable hyperparameter of the kernel function that must be selected *a priori*. We discuss how we make these choices in Section 2.3.1. We set $\sigma_{\mathrm{A}} = 1$ and therefore it is not a hyperparameter of our kernel. For this to be valid, we must scale the eigenmode weight training data to have variance of unity.

In our case, we have multiple GP regressors—one for each component of the eigenmode weight vector. Consider for example the weight for the first eigenmode. Suppose we group the training data for this weight into a vector $\mathbf{y}^{\mathrm{tr}}$, such that $y_j^{\mathrm{tr}} \equiv w_1^j/\lambda_1^{1/2}$, where λ_i is the variance of weight element w_i from Equation 5. Dividing by the standard deviation ensures that the variance of the weights are unity, and therefore allows us to set $\sigma_{\mathrm{A}} = 1$. If we define an $m_{\mathrm{tr}} \times m_{\mathrm{tr}}$ matrix $\mathbf{K}_1^{\mathrm{tr\text{-}tr}}$ such that $(\mathrm{K}_1^{\mathrm{tr\text{-}tr}})_{ij} \equiv k(\boldsymbol{\theta}_i^{\mathrm{tr}},\boldsymbol{\theta}_j^{\mathrm{tr}}|\mathbf{L}_1)$, then the GP prediction for the weight at point $\boldsymbol{\theta}^{\mathrm{new}}$ is given by

$$w_1^{\mathrm{new}} = \lambda_1^{1/2}(\mathbf{k}_1^{\mathrm{new\text{-}tr}})^T\left[\mathbf{K}_1^{\mathrm{tr\text{-}tr}} + \sigma_n^2\mathbf{I}\right]^{-1}\mathbf{y}^{\mathrm{tr}}, \tag{9}$$

where $\mathbf{k}_1^{\mathrm{new\text{-}tr}}$ is a length-m_{tr} vector defined analogously to $\mathbf{K}_1^{\mathrm{tr\text{-}tr}}$, i.e., $(\mathrm{k}_1^{\mathrm{new\text{-}tr}})_i \equiv k(\boldsymbol{\theta}^{\mathrm{new}},\boldsymbol{\theta}_i^{\mathrm{tr}}|\mathrm{L}_1)$, $\mathbf{L}_1$ is the matrix containing the hyperparameters chosen a priori for the input training data and the subscript 1 specifies that the input training data are the weights of the first PC mode, w_1. The variance about this prediction is then given by[5]

$$\gamma_1^{\mathrm{new}} = 1 - (\mathbf{k}_1^{\mathrm{new\text{-}tr}})^T\left(\mathbf{K}_1^{\mathrm{tr\text{-}tr}} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}_1^{\mathrm{new\text{-}tr}}, \tag{10}$$

where $\mathbf{I}$ is the identity matrix, and σ_n^2 is the variance of random Gaussian noise possibly corrupting the training data from their underlying distribution and is a hyperparameter of the GP (Rasmussen & Williams 2006).

Evaluating Equation 9 for each PC weight yields a set of predicted weights that come together to form the vector $\mathbf{w}^{\mathrm{new}}$. This may then be inserted into Equation 7 to yield predictions for the quantities we desire. Similarly, evaluating Equation 10 for each PC weight and stacking them into a vector $\boldsymbol{\gamma}^{\mathrm{new}}$, we may propagate our GP's uncertainty on $\mathbf{w}^{\mathrm{new}}$ into an emulator covariance $\boldsymbol{\Sigma}_{\mathrm{E}}$,

[5] In principle, one may perform a GP estimate over several points in parameter space at once. Equation 9 then predicts an entire vector of w_1^{new} values simultaneously, and Equation 10 generalizes to a full covariance matrix. Here we do not employ such a formalism since an MCMC chain explores parameter space one point at a time.

which describes the uncertainty on the unlogged[6] emulator predictions $\exp(\mathbf{d}^{\rm new})$, and is given by

$$(\Sigma_{\rm E})_{ij} = \sum_k^M \exp(\mathrm{d}_i^{\rm new})\exp(\mathrm{d}_j^{\rm new})\Phi_{ik}\Phi_{jk}\gamma_k^{\rm new}, \quad (11)$$

where in deriving this expression we have assumed that the emulator errors are small. Importantly, note that because $\gamma_k^{\rm new}$ depends on $\boldsymbol{\theta}^{\rm new}$, the same is true for $\boldsymbol{\Sigma}_{\rm E}$. This is to be expected. For instance, one would intuitively expect the emulator error to be larger towards the edge of our training region than at the center of it. In practice, it is helpful to complement estimates of emulator from Equation 11 with empirical estimates derived from cross validation. Essentially, one takes a set of simulation evaluations not in the training set and compares the emulator's prediction at those points in parameter space against the true simulation output. We further discuss these considerations and how the estimated emulator error $\boldsymbol{\Sigma}_{\rm E}$ comes into our parameter constraints when we lay out the construction of our likelihood function in Section 4.2.

So far we have been working towards constructing a set of GP models for each PC mode, each of which is a predictive function spanning the entire parameter space and uses all of the training data. A different regression strategy is called the Learn-As-You-Go method (Aslanyan et al. 2015). In this method, one takes a small subset of the training data immediately surrounding the point-of-interest, $\boldsymbol{\theta}^{\rm new}$, in order to construct localized predictive functions, which then get thrown away after the prediction is made. This is desirable when the training set becomes exceedingly large ($m_{\rm tr} \gtrsim 10^4$ samples), because the computational cost of GP regression naively scales as $m_{\rm tr}^3$. This is the regression strategy we adopt in our initial broad parameter space search in Section 4.3.

Our emulator algorithm in `emupy` relies on base code from the Gaussian Process module in the publicly-available Python package Sci-Kit Learn,[7] which has an optimized implementation of Equation 9 and Equation 10 (Pedregosa et al. 2012).

2.3.1. *GP Hyperparameter Solution*

The problem we have yet to address is how to select the proper set of hyperparameters for our GP kernel function, in particular the characteristic scale length of correlations ℓ across each model parameter. We can do this through a model selection analysis, where we seek to find $\mathbf{L}$ such that the marginal likelihood of the training data given the model hyperparameters is maximized. From Rasmussen & Williams (2006), the GP log-marginal likelihood for a single PC mode is given (up to a constant) by

$$\ln \mathcal{L}_{\rm M}(\mathbf{y}^{\rm tr}|\mathbf{L}) \propto -\frac{1}{2}(\mathbf{y}^{\rm tr})^T(\mathbf{K}^{\text{tr-tr}})^{-1}\mathbf{y}^{\rm tr} - \frac{1}{2}\det(\mathbf{K}^{\text{tr-tr}}), \quad (12)$$

where $\mathbf{K}^{\text{tr-tr}}$ has the same definition as in Equation 9, and thus carries with it a dependence on $\boldsymbol{\theta}^{\rm tr}$ and $\mathbf{L}$. In principle, one could also simultaneously vary the assumed noise variance (σ_n^2) of the target data as an additional hyperparameter and jointly fit for the combination of $[\mathbf{L}, \sigma_n^2]$. To find these optimal hyperparameters, we can use a gradient descent algorithm to explore the hyperparameter parameter space of $\mathbf{L}$ and σ_n^2 until we find a combination that maximizes $\ln \mathcal{L}_{\rm ML}$. When working with training data directly from simulations, we would expect σ_n^2 to be minimal; we are not dealing with any observational or instrumental systematics that might introduce uncertainty into their underlying values. Depending on the simulation, there may be numerical noise or artifacts that introduce excess noise or outlier points into the target data, which may skew the resultant best-fit for $\mathbf{L}$ or break the hyperparameter regression entirely. This can be alleviated by keeping σ_n^2 as a free parameter and fitting for it and $\mathbf{L}$ jointly.

In our eleven dimensional space, this calculation can become exceedingly slow when the number of samples in our training set exceeds ten thousand. In our initial broad parameter space exploration (Section 4.3), for example, performing a hyperparameter gradient descent with all 15,000 samples is not attempted. To solve for the hyperparameters, we thus build a separate training set that slices the parameter space along each parameter and lays down samples along that parameter while holding all others constant. We then take this training set slice and train a 1D GP and fit for the optimal ℓ of that parameter by maximizing the marginal likelihood. We repeat this for each parameter and then form our $\mathbf{L}$ matrix by inserting our calculated ℓ along its diagonal. This is a method of constraining each dimension's ℓ individually, in contrast to the previous method of constraining ℓ across all dimensions jointly. While this is a more approximate method, it is computationally much faster.

In order to construct a fully hierarchical model, we should in principle not be selecting a single set of hyperparameters for our GP fit, but instead should be marginalizing over all allowed hyperparameter combinations informed by the marginal likelihood. That is to say, we should fold the uncertainty on the optimal choice of ℓ into our uncertainty on our predicted $\mathbf{w}^{\rm new}$ and thus our predicted $\mathbf{d}^{\rm new}$. In theory this would be ideal, but in practice, this quickly becomes computationally infeasible. This is because the time it takes to train a GP and make interpolation predictions naively scales as the number of training samples $m_{\rm tr}$ cubed. Optimized implementations, such as the one in Sci-Kit Learn, achieve better scaling for low $m_{\rm tr}$, but for large $m_{\rm tr}$ this efficiency quickly drops, to the point where hav-

[6] Recall that in Equation 1 we defined the data vector to be the *logarithm* of the original quantities we wished to emulate.

[7] `http://scikit-learn.org/`

ing to marginalize over the hyperparameters to make a single prediction at a single point in parameter space can take upwards of minutes, if not hours, which begins to approach the run time of our original simulation. Furthermore, all of these concerns are exponentially exacerbated in high dimensional spaces. However, in the limit of a high training set sampling density this difference becomes small, which is to say that our marginal likelihood becomes narrow as a function of the hyperparameters. Lastly, and most importantly, we can always turn to diagnose the accuracy of our emulator (and calibrate out its failures) by cross validating it against a separate set of simulation evaluations. In doing so, we can ensure that the emulator is accurate within the space enclosed by our training set.

2.3.2. *GP Cross Validation*

Emulators are approximations to the data products of interest, and as such we need to be able to assess their performance if we are to trust the parameter constraints we produce with them. As discussed above, this can be accomplished empirically via cross validation (CV). In this paper, rather than computing extra simulation outputs to serve as CV samples, we elect to perform k-fold cross validation. In k-fold cross validation, we take a subset of our training samples and separate them out, train our emulator on the remaining samples, cross validate over the separated set, and then repeat this k times. This ensures we are not training on the cross validation samples and also means we do not have to use extra computational resources generating new samples.

We use two error metrics to quantify the emulator performance. The first metric is the absolute fractional error of the emulator prediction over the cross validation data, expressed as $\epsilon_{\rm abs} = ([\Delta^2_{21}]_{\rm E} - [\Delta^2_{21}]_{\rm CV})/[\Delta^2_{21}]_{\rm CV}$. This gives us a sense of the absolute precision of our emulator. However, not all k modes contribute equally to our constraining power. Because our 21 cm experiment will measure some k modes to significantly higher signal-to-noise (S/N) than other k modes, we want to confirm that our emulator can at least emulate at a precision better than the S/N of our experiment, and is therefore not the dominant source of error at those k modes. The second metric we use is then the offset between the emulator and the CV, divided by the error bars of our 21 cm experiment, given by $\epsilon_{\rm obs} \equiv ([\Delta^2_{21}]_{\rm E} - [\Delta^2_{21}]_{\rm CV})/\sigma_{\rm S}$, where $\sigma_{\rm S}$ are the 21 cm power spectrum error bars of our experiment. For this, we use the projected sensitivity error bars of the HERA experiment, which we discuss in detail in Section 4.1, and is shown in Figure 4.

Cross validation leaves us with an error distribution of the CV samples at each unique k and z. Applying our error metrics, we are left with two sets of error distributions for the emulated power spectra, $\epsilon_{\rm abs}(k,z)$ and $\epsilon_{\rm obs}(k,z)$. To quantify their characteristic widths, we calculate their robust standard deviations, $\sigma_{\rm abs}$ and $\sigma_{\rm obs}$ respectively, using the bi-weight method of Beers et al. (1990). We show an example of these standard deviations in Figure 2, which demonstrates the emulator's ability to recover the 21 cm power spectra having trained it on the training set described in Section 4.4. Here, we take 2×10^3 of the center-most samples of the 5×10^3-sample training set and perform 5-fold cross validation on them. The top subplot shows the absolute error metric $\sigma_{\rm abs}$, and demonstrates our ability to emulate at $\leq 5\%$ precision for the majority of the power spectra, and $\leq 10\%$ for almost all of the power spectra. The bottom subplot shows the observational error metric $\sigma_{\rm obs}$, and demonstrates that we can emulate to an average precision that is well below the observational error bars of a HERA-like experiment for virtually all k modes, keeping in mind that the highest S/N k modes for 21 cm experiments are at low-k and low-z for $z \gtrsim 6$. The inset shows the underlying error distribution and its robust standard deviation for one of the power spectra data output. Note that the the distribution of points on the k-z plane that are shown in Figure 2 are not

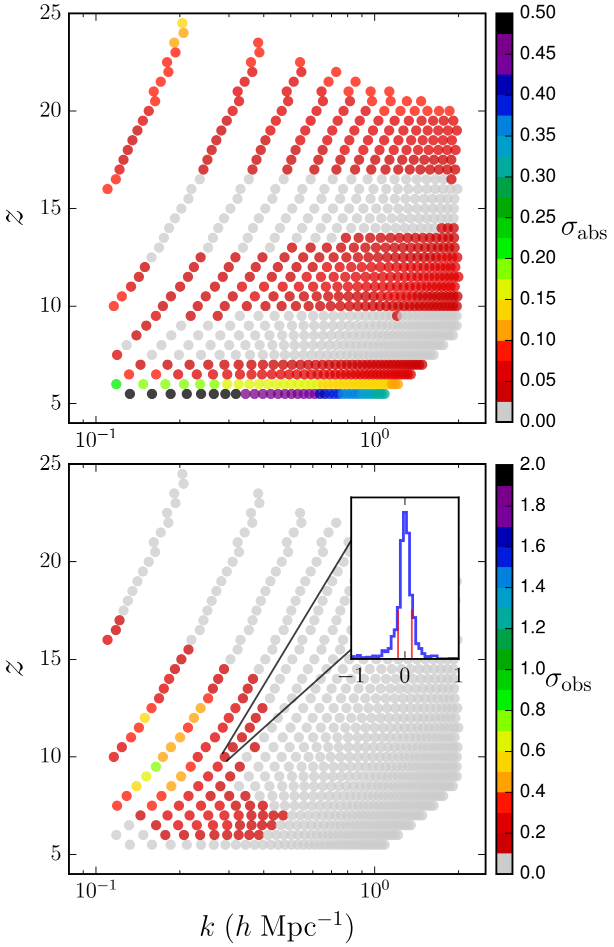

Figure 2. **Top**: Standard deviation of the absolute fractional emulator error ($\sigma_{\rm abs}$) with respect to the CV set. Grey color indicates an emulator precision of $\leq 2.5\%$. **Bottom**: Standard deviation of the offset between emulator prediction and CV data, divided by the experimental errors ($\sigma_{\rm obs}$). The grey color over the majority of the data signifies we can recover the data to $\leq 10\%$ relative to the experimental error bars. **Inset**: Error distribution $\epsilon_{\rm obs}$ for a data output, with its robust standard deviation marked as vertical bars.

determined by the emulator; indeed, one can easily emulate the power spectra at different values of k and z. Instead, these points were chosen to match the observational survey parameters and our choice of binning along our observational bandpass. We discuss such survey parameters in more detail in Section 4.1.

The observational error metric is of course dependent on the chosen 21 cm experiment and its power spectrum sensitivity. This particular emulator design, for example, may not be precise enough to emulate within the error bars of a futuristic experiment. If we need to boost our emulator's precision, we can do so to an almost arbitrary level by simply generating more training samples and packing the space more densely. The limiting factors of this is the need to generate an additional number of training samples which is unknown a priori, and the intrinsic $m_{\rm tr}^3$ scaling of the Gaussian Process regressor. However, with sufficient computational resources and novel emulation strategies like Learn-As-You-Go, increasing the emulator's precision to match an arbitrary experimental precision is in principle feasible.

3. CHOOSING A MODEL FOR COSMIC DAWN

Having described the core features of our emulator, we will now focus on a specific model of the 21 cm signal so that we may build a training set. To accurately describe the large-scale correlation statistics of the cosmological 21 cm signal, we need large-volume simulations with box lengths $L > 200$ Mpc (Barkana & Loeb 2004; Trac & Gnedin 2011; Iliev et al. 2014). Compared to the physical sizes of ionizing photon sources and sinks at the galactic scale of kpc and smaller, it is clear that in order to directly simulate reionization one needs to resolve size scales that span many orders of magnitude. This has made direct simulations of reionization a computationally formidable task; current state-of-the-art high resolution hydrodynamic plus self-consistent radiative transfer codes are extremely expensive and only reach up to tens of Mpc in box length. As a consequence, less numerically rigorous but dramatically cheaper semi-analytic approaches have made more progress in exploring the EoR parameter space. One such code is the semi-numerical simulation `21cmFAST` (Mesinger & Furlanetto 2007; Mesinger et al. 2011), which we use in this work to build our training sets. For the following, we use the publicly available `21cmFAST_v1.12`.[8] However, we again emphasize that the idea of emulating simulations of Cosmic Dawn is not one that is tied to `21cmFAST`; indeed, one could easily perform similar calculations to the one in this paper with other semi-numerical simulations, such as those described in Geil & Wyithe (2008), Choudhury et al. (2009), Thomas et al. (2009), Santos et al. (2010), Battaglia et al. (2013), and Kulkarni et al. (2016), or with numerical simulations, such as those described in Mellema et al. (2006), Zahn et al. (2007), Baek et al. (2009), Trac & Gnedin (2011), Iliev et al. (2014), Gnedin (2014), Ross et al. (2017), Kaurov & Gnedin (2016), and Das et al. (2017).

The relevant observable our simulation needs to predict is the 21 cm brightness temperature at radio frequencies. Specifically, it is the 21 cm brightness temperature offset from the background CMB brightness temperature. Because the 21 cm signal is a line transition, its fluctuations across frequency encode redshift information while fluctuations across the sky encode angular information. We can therefore recover three-dimensional information of the IGM structure and thermal state with the 21 cm line. For a parcel of gas at a specific location on the sky with a redshift z corresponding to a redshifted 21 cm frequency of ν, the 21 cm brightness temperature offset can be written as

$$\delta T_{\rm b}(\nu) \approx 9 x_{\rm HI}(1+\delta)(1+z)^{\frac{1}{2}}\left(1-\frac{T_\gamma}{T_{\rm S}}\right)\left(\frac{H(z)}{dv_r/dr}\right)\,{\rm mK} \tag{13}$$

where $x_{\rm HI}$ is the hydrogen neutral fraction, δ the baryon overdensity, T_γ the CMB background temperature, $T_{\rm S}$ the neutral hydrogen hyperfine "spin" temperature (Wouthuysen 1952; Field 1958; Furlanetto et al. 2006), $H(z)$ is the Hubble parameter, and dv_r/dr is the line of sight proper velocity gradient. All of the parameters on the right hand side have both frequency and angular dependence on the sky, except for the Hubble parameter with only frequency dependence. To make a prediction of the 21 cm brightness temperature field, we therefore need an underlying cosmology, a prescription for the matter density field, the hydrogen ionization fraction field and in certain cases the hyperfine spin temperature field. Some models choose to make the assumption that the spin temperature greatly exceeds the photon temperature ($T_{\rm S} \gg T_\gamma$), in which case the 21 cm temperature is insensitive to $T_{\rm S}$ and we need not compute it. This is sometimes assumed to be true during the late stages of reionization ($z < 10$) when the IGM gas temperature has been sufficiently heated. This is complicated by the fact that certain EoR scenarios of reionization, dubbed"cold reionization," predict inefficient IGM heating and therefore the $T_{\rm S} \gg T_\gamma$ assumption breaks down (Mesinger et al. 2014; Cohen et al. 2016; Mirocha et al. 2017). Furthermore, recent work shows that even if the spin temperature is saturated with respect to the photon temperature at $z < 10$, efficient IGM heating can still leave an imprint on EoR and will bias astrophysical constraints that neglect it (Greig & Mesinger 2017a).

`21cmFAST` generates a density field by evolving an initial Gaussian random field to low redshifts via the Zel'dovich approximation. `21cmFAST` does not account for baryonic hydrodynamics and thus makes the implicit assumption that the baryons track the dark matter. From the evolved density field, hydrogen ionization fields are calculated using the excursion set theory of Furlanetto et al. (2004). In this formalism, the density

[8] https://github.com/andreimesinger/21cmFAST

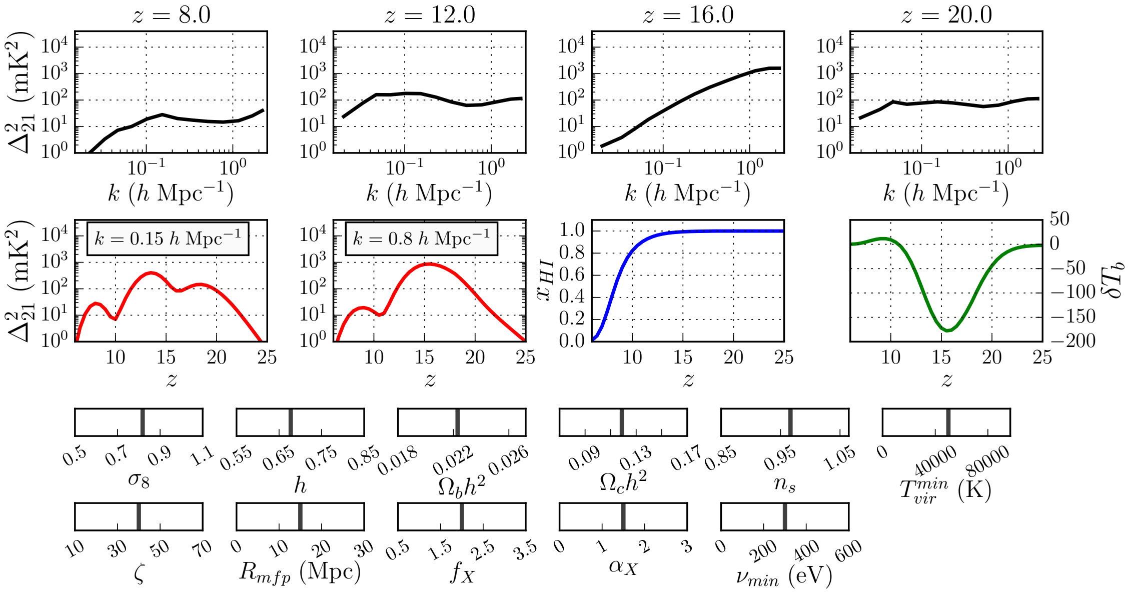

Figure 3. Data products from the semi-numerical EoR simulation `21cmFAST`. From top to bottom, left to right we show the 21 cm dimensional power spectra as a function of wavenumber k at various redshifts, the power spectra redshift evolution at a specific k-mode, the hydrogen neutral fraction redshift evolution and the global signal redshift evolution. The parameters on bottom show the choice of model parameters for this specific realization. For building intuition, a movie showing the behavior of these outputs to variations in the model parameters can be found at http://w.astro.berkeley.edu/~nkern/images/ps_movie.mp4.

field is smoothed to a comoving radius scale, $R_{\rm mfp}$, and the central cell about the smoothing is considered ionized if

$$f_{\rm coll}(\boldsymbol{x}, z, R) \geq \zeta^{-1}, \qquad (14)$$

where $f_{\rm coll}$ is the fraction of matter that has collapsed into gravitationally bound structure at position $\boldsymbol{x}$, redshift z and with smoothing scale R. $f_{\rm coll}$ is computed via the hybrid prescription in Barkana & Loeb (2004). ζ is an ionization efficiency parameter for star forming halos (see Section 3.1.1 below for a description). This process is iterated on smaller smoothing scales until either the cell becomes ionized or the cell resolution is reached. The initial smoothing scale, $R_{\rm mfp}$, can be thought of as the mean-free path of photons through ionized regions, which accounts for unresolved sub-halos with pockets of neutral hydrogen that act as ionization sinks. Detailed studies have compared `21cmFAST` against more accurate RT simulations and have shown that their ionization fields and 21 cm power spectra during the EoR ($z < 10$) are consistent at the $\sim 20\%$ level (Mesinger et al. 2011; Zahn et al. 2011).

`21cmFAST` can also calculate the kinetic gas temperature and spin temperature fields. The spin temperature couples to either the background photon CMB temperature or to the IGM kinetic gas temperature. The latter can occur via hydrogen collisional coupling, as is thought to occur early on ($z \geq 20$), or via the Wouthuysen–Field effect where Lyman-α pumping couples the spin temperature to the Lyman-α color temperature, which closely traces the kinetic gas temperature due to the high scattering cross section of Lyman-α photons with neutral hydrogen (Furlanetto 2006). In order to calculate the IGM gas kinetic temperature one must track inhomogeneous IGM heating, which is thought to predominately occur by X-rays. To track this, `21cmFAST` integrates the angle-averaged specific X-ray emissivity across a light cone and across X-ray frequencies for each cell. X-ray production, due to either high-mass X-ray binaries or a hot Interstellar Medium (ISM) phase, is assumed to be tied to star formation. While the power spectra during X-ray heating from a fiducial `21cmFAST` realization roughly agree with the trends seen from numerical simulations (Mesinger et al. 2011; Ross et al. 2017), a detailed comparison of `21cmFAST` against numerical simulations of X-ray heating has not been made. Such comparisons are necessary to test the accuracy of the X-ray treatment in `21cmFAST`, and could help calibrate or better inform the semi-analytics therein. For the time being, we accept `21cmFAST`'s treatment of the X-ray heating for intuitive purposes. For a more detailed description of the semi-numerics inside `21cmFAST`, see Mesinger et al. (2011, 2013).

To build our training sets, the simulation runs have box lengths L = 400 Mpc. We sample the Gaussian initial conditions for the density field from a 800^3-voxel grid, which then get smoothed onto a 200^3-voxel grid to track its evolution via the Zel'dovich approximation and to compute the relevant 21 cm fields. This lower resolution grid corresponds to a cell resolution of 2 Mpc. For comparison, the minimum length-scale that an experiment like HERA is expected to be sensitive to is around 5 Mpc.

The data products that `21cmFAST` produces are 3D box outputs of cosmological fields, from which we can construct the 21 cm power spectrum, the average 21 cm brightness temperature offset from the CMB (also referred to as the global signal or monopole signal), the average hydrogen neutral fraction and the integrated electron scattering optical depth. Figure 3 shows an example of these data products from a fiducial 21cmFAST realization. We could also construct higher order statistical probes from the box-outputs, such as three- or four-point functions, which in principle carry useful information because the ionization fields are non-Gaussian; however, for this study we focus on the power spectrum as our observable. Future work will focus on synthesizing other EoR probes within the parameter estimation framework presented here.

The 21 cm power spectrum is defined as $\Delta^2_{21}(k) = (k^3/2\pi^2)P_{21}(k)$, with $P_{21}(k)$ being defined as

$$\langle\widetilde{\delta T_{\rm b}}(\boldsymbol{k}_1)\widetilde{\delta T_{\rm b}}^*(\boldsymbol{k}_2)\rangle = (2\pi)^3\delta^D(\boldsymbol{k}_1-\boldsymbol{k}_2)P_{21}(k_1), \quad (15)$$

where $\langle\dots\rangle$ denotes an ensemble average, $\widetilde{\delta T_{\rm b}}(\boldsymbol{k})$ is the spatial Fourier transform of $\delta T_{\rm b}(\boldsymbol{x})$, δ^D is the Dirac delta function, $\boldsymbol{k}$ is the spatial Fourier wavevector, and $k\equiv|\boldsymbol{k}|$. Because we constructed the power spectrum by taking the spatial Fourier transform of $\delta T_{\rm b}$, Δ^2_{21} carries units of mK^2. This is the statistic 21 cm interferometric experiments like HERA are aiming to measure (among other quantities), and this is the 21 cm statistic we will focus on in this paper.

3.1. *Model Parameterization*

We adopt an eleven parameter model within 21cmFAST to characterize the variability of $\delta T_{\rm b}$ across the reionization and X-ray heating epochs. This consists of six astrophysical parameters that describe the production and propagation of UV and X-ray photons, and five cosmological parameters that influence the underlying density field and velocity fields of our Universe.

3.1.1. *EoR Parameters: $\zeta, T_{vir}^{min}, R_{mfp}$*

The production rate of UV photons is governed by the ionization efficiency of star-forming galaxies, ζ, which can be expressed as

$$\zeta = 30\left(\frac{f_{\rm esc}}{0.15}\right)\left(\frac{f_\star}{0.1}\right)\left(\frac{N_\gamma}{4000}\right)\left(\frac{2}{1+n_{\rm rec}}\right), \quad (16)$$

where $f_{\rm esc}$ is the fraction of produced UV photons that escape the galaxy, $f_\star$ is the fraction of collapsed gas in stars, N_γ is the number of ionizing photons produced per stellar baryon and $n_{\rm rec}$ is the average number of times a hydrogen atom in the IGM recombines. The splitting of ζ into these four constituent parameters is merely for clarity: the numerics of 21cmFAST respond only to a change in ζ, regardless of what sub-parameter sourced that change. These sub-parameters are therefore completely degenerate with each other in the way they affect reionization in 21cmFAST. Previous works have explored how to parameterize the mass and redshift evolution of ζ (Greig & Mesinger 2015; Sun & Furlanetto 2016), and this will certainly be a feature to incorporate into this framework for future studies. For the time being, we assume ζ to be a constant for intuitive purposes, similar to previous work. Some of the fiducial values for the terms in Equation 16 are physically motivated—$N_\gamma\sim 4000$ is the expectation from spectral models of Population II stars (Barkana & Loeb 2005), and both $f_\star$ and $f_{\rm esc}$ are thought to lie within a few factors of 0.1 (Kuhlen & Faucher-Giguère 2012; Robertson et al. 2015; Paardekooper et al. 2015; Xu et al. 2016; Sun & Furlanetto 2016)—however, these are not strongly constrained at high redshifts and are particularly unconstrained for low-mass halos.

Baryonic matter must cool in order for it to condense and allow for star formation. This can occur through radiative cooling from molecular hydrogen, although this is easily photodissociated by Lyman-Werner photons from stellar feedback (Haiman et al. 1997). Other cooling pathways exist, but in general, low mass mini-halos are thought to have poor star formation efficiencies due to stellar feedback (Haiman et al. 2000). We can parameterize the lower limit on halo mass for efficient star formation as a minimum halo virial temperature, $T_{\rm vir}^{\rm min}$ (K). Here we adopt a fiducial $T_{\rm vir}^{\rm min}$ of 5×10^4 K, above the atomic line cooling threshold of 10^4 K (Barkana & Loeb 2002). In practice, this parameter has the effect of stopping the excursion set formalism for a cell smoothed on scale R if its mass is less than the minimum mass set by $T_{\rm vir}^{\rm min}$.

As ionizing photons escape star forming galaxies and propagate through their local HII region, they are expected to encounter pockets of neutral hydrogen in highly shielded sub-structures (Lyman-limit systems). Without explicitly resolving these ionization sinks, we can parameterize their effect on ionizing photons escaping a galaxy by setting an effective mean-free path through HII regions for UV photons, $R_{\rm mfp}$. In practice, this sets the maximum bubble size around ionization sources, and is the initial smoothing scale for the excursion set (as discussed above in Section 3). Motivated by subgrid modeling of inhomogeneous recombinations (Sobacchi & Mesinger 2014), we adopt a fiducial value of $R_{\rm mfp} = 15$ Mpc.

3.1.2. *X-ray Spectral Parameters: f_X, α_X, ν_{min}*

The sensitivity of the 21 cm power spectrum to cosmic X-rays during the IGM heating epoch may allow us to constrain the spectral properties of the X-ray generating sources. These are theorized to come predominately from either High Mass X-ray Binaries (HMXB) or a hot Interstellar Medium (ISM) component in galaxies heated by supernovae. In 21cmFAST, the X-ray source emissivity is proportional to

$$\epsilon_X(\nu)\propto f_X\left(\frac{\nu}{\nu_{\rm min}}\right)^{-\alpha_X}, \quad (17)$$

where f_X is the X-ray efficiency parameter (an overall normalization), α_X is the spectral slope parameter, and $\nu_{\rm min}$ is the obscuration frequency cutoff parameter, below which we take the X-ray emissivity to be zero due to ISM absorption. High-resolution hydrodynamic simulations of the X-ray opacity within the ISM have found that such a power-law model is a reasonable approximation of the emergent X-ray spectrum from the first galaxies (Das et al. 2017). Our fiducial choice of $f_X = 1$ corresponds to an average of 0.1 X-ray photons produced per stellar baryon. HMXB spectra have typical spectral slopes α_X of roughly unity, while a hot ISM component tends to have a spectral slope of roughly 3 (Mineo et al. 2012). Our fiducial choice of $\alpha_X = 1.5$ straddles these expectations. The obscuration cutoff frequency, $\nu_{\rm min}$, parameterizes the X-ray optical depth of the ISM in the first galaxies and is dependent on their column densities and metallicities. We choose a fiducial value of $\nu_{\rm min} = 0.3\,{\rm keV}$, consistent with previous theoretical work (Pacucci et al. 2014; Ewall-Wice et al. 2016b). Because the model assumes the X-ray production comes from star forming halos, the EoR parameter $T_{\rm vir}^{\rm min}$ also affects the spatial distribution of X-ray sources, and is therefore also implicitly an X-ray heating parameter. For a more detailed description of the X-ray numerics in `21cmFAST`, see Mesinger et al. (2013).

3.1.3. *Cosmological Parameters*

A previous study utilizing the Fisher Matrix approach found that even though cosmological parameters have precise constraints from other cosmological probes such as the *Planck* satellite, their residual uncertainties introduce a non-negligible effect on the 21 cm power spectrum and thus degrade the constraints one can place on astrophysical parameters using 21 cm measurements (Liu et al. 2016). Stated another way, by excluding cosmological parameters from a joint fit, we would be falsely *overconstraining* the astrophysical parameters. Additionally, besides their degradation of astrophysical parameter constraints, we would also like to be able to constrain cosmology with the rich amount of information the 21 cm signal provides us. We pick $\{\sigma_8, H_0, \Omega_b h^2, \Omega_c h^2, n_s\}$ as our cosmological parameter set. This particular parameterization is selected to match the current `21cmFAST` cosmological inputs and is done merely for convenience. It may be worth investigating in future work if other ΛCDM parameterizations are more suitable for 21 cm constraints. In terms of `21cmFAST`, all of the chosen parameters play a role in setting the initial conditions for the density field, and $\Omega_b h^2$, $\Omega_c h^2$ and H_0 are furthermore directly related to the definition of the 21 cm brightness temperature (Equation 13). Some of these cosmological parameters also play a role in transforming our observed coordinates on the sky into cosmological distance coordinates (see Equation 18). While we do not include these effects into this study, a complete analysis would require such a consideration, which will be addressed in future work. Our fiducial values for the cosmological parameters are (σ_8, h, $\Omega_b h^2$, $\Omega_c h^2$, n_s) = (0.8159, 0.6774, 0.0223, 0.1188, 0.9667), which are consistent with recent *Planck* results (Planck Collaboration et al. 2016). Because σ_8 and H_0 are not directly constrained by *Planck* but are derived parameters in their ΛCDM parameterization, we use the `CAMB` code (Lewis et al. 2000) to map the parameter degeneracies of A_s (the normalization of the primordial perturbation power) and $\theta_{\rm MC}$ (the `CosmoMC` code approximation for the angular size of the sound horizon at recombination; Lewis & Bridle 2002) onto that of σ_8 and H_0 respectively.

4. FORECASTED HERA331 CONSTRAINTS

Here we forecast the ability of the HERA experiment to constrain the eleven parameter model described above. Before we present the parameter constraints, however, we must discuss how we construct our likelihood function and account for how errors in the emulator prediction can affect the final likelihood statistic. We begin by creating a mock observation for the HERA experiment and accounting for known systematics like bright foreground contamination.

4.1. *Interferometer Sensitivity Model*

To create a mock 21 cm power spectrum observation, we run `21cmFAST` with "true" model parameters set to the fiducial values described in Section 3.1. The initial conditions of the density field are generated with a different random seed than what was used to construct the training set realizations.

Uncertainty in the 21 cm power spectrum at the EoR comes from three main sources, (i) thermal noise of the instrument, (ii) uncertainty in foreground subtraction, and (iii) sampling (or cosmic) variance of our survey. For portions of Fourier space that are clean of foregrounds, the variance of the power spectrum at an individual $\boldsymbol{k}$ mode from the two remaining sources of uncertainty can be written as

$$\sigma^2(\boldsymbol{k}) \approx \left[X^2 Y \frac{k^3}{2\pi^2}\frac{\Omega'}{2t}T_{\rm sys}^2 + \widehat{\Delta_{21}^2}(\boldsymbol{k})\right]^2, \qquad (18)$$

where the first term is the thermal noise ($k = |\boldsymbol{k}|$), and the second term is the sampling variance uncertainty at each individual $\boldsymbol{k}$ mode (Pober et al. 2013b). In the first term, X^2Y are scalars converting angles on the sky and frequency spacings to transverse and longitudinal distances in h Mpc^{-1}, and Ω' is the ratio of the square of the solid angle of the primary beam divided by the solid angle of the square of the primary beam (Parsons et al. 2014). The total amount of integration time on a particular k mode is t, and $T_{\rm sys}$ is the system temperature taken to be the sum of a receiver temperature at 100 K and a sky temperature at $60\lambda^{2.55}$ K, where λ has units of meters (Parsons et al. 2014). To compute the variance on the 1D power spectrum, Var[$\Delta_{21}^2(k)$], from the above variances on the 2D power spectrum, we bin into annuli of constant scalar k and add the variances reciprocally (Pober et al. 2013b).

We perform these calculations with the public Python

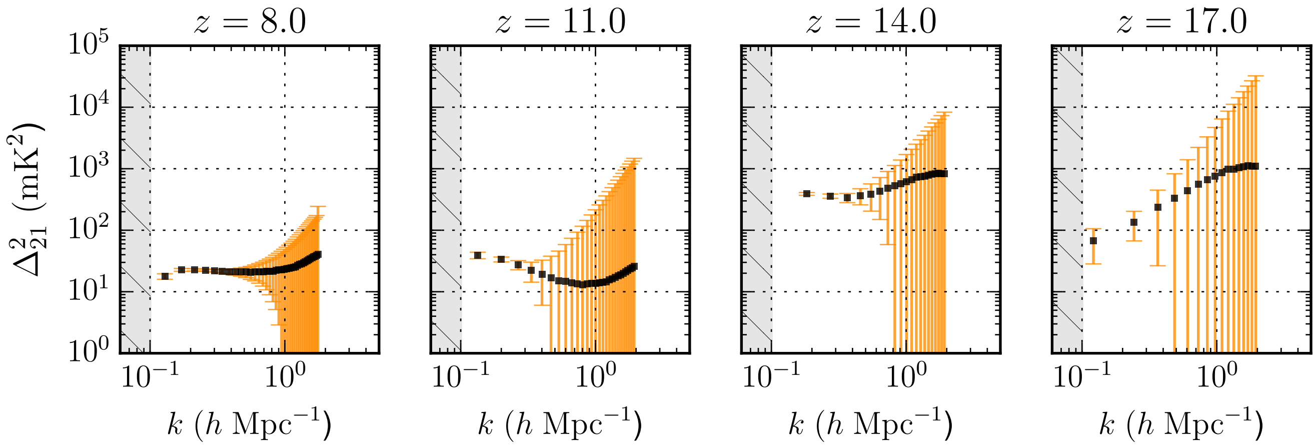

Figure 4. A mock observation of the 21 cm power spectrum created from an underlying "truth" realization of `21cmFAST` with error bars corresponding to the projected sensitivity of the HERA331 experiment after a single observing season. The grey-hatched region to the left denotes inaccessibility due to foreground dominated k modes. Although we display only four redshifts, the entire mock observation contains the 21 cm power spectrum from $5 < z < 25$ in steps of $\Delta z = 0.5$.

package `21cmSense`,[9] which takes as input a specification of the interferometer design and survey parameters (see Parsons et al. 2012a; Pober et al. 2014). We assume a HERA-like instrument with a compact, hexagonal array of 331 dishes that each span 14-m in diameter (Dillon & Parsons 2016; DeBoer et al. 2017). We further assume the observations are conducted for 6 hours per day spanning a 180 day season for a total of 1080 observing hours. Within an instrumental bandpass spanning 50-250 MHz, we construct power spectra from $5 < z < 25$ in co-evolution redshift bins of $\Delta z = 0.5$. We also adopt the set of "moderate" foreground assumptions defined in `21cmSense`. This assumes that in a cylindrical Fourier space decomposed into wavenumbers perpendicular ($k_\perp$) and parallel ($k_\|$) to the observational line-of-sight, the foreground contaminants are limited to a characteristic "foreground wedge" at low $k_\|$ and high $k_\perp$(see e.g., Datta et al. 2010; Morales et al. 2012; Trott et al. 2012; Parsons et al. 2012b; Pober et al. 2013a; Liu et al. 2014a,b). One then pursues a strategy of foreground avoidance (rather than explicit subtraction) under the approximation that outside the foreground wedge there is a foreground-free "EoR window". To be conservative, we impose an additional buffer above the foreground wedge of $k_\| = 0.1$ h Mpc^{-1} to control for foreground leakage. For more detailed discussions of foreground avoidance and subtraction techniques for 21 cm interferometers, we defer the reader to Pober et al. (2014). Figure 4 shows the resulting mock observation produced by applying the above calculations to our truth `21cmFAST` realization.

4.2. *Constructing the Likelihood*

The likelihood function describes the ability of our observations to constrain the model parameters. This could in principle contain data from multiple observable probes of reionization. For this study, we focus solely on the likelihood function for the 21 cm power spectrum, but future work will investigate extending this formalism to incorporate other EoR probes. Our 21 cm power spectrum likelihood function can be written up to a constant as

$$\ln \mathcal{L}(\mathbf{y}|\boldsymbol{\theta}) \propto -\frac{1}{2}(\mathbf{y}-\mathbf{d})^T \boldsymbol{\Sigma}_{\rm S}^{-1}(\mathbf{y}-\mathbf{d}), \tag{19}$$

where $\boldsymbol{\Sigma}_{\rm S}$ is a diagonal covariance matrix containing the observational survey error bars (including both thermal noise and sample variance as described in Section 4.1), $\mathbf{y}$ are our observations of the 21 cm power spectrum spanning a redshift range of $5 < z < 25$ and scale range $0.1 < k < 2$ h Mpc^{-1}, and $\mathbf{d}$ are the model predictions evaluated at some point in parameter space $\boldsymbol{\theta}$. In Greig & Mesinger (2015), who similarly investigated parameter constraints with `21cmFAST`, the authors add an additional 20% error on the sampled 21 cm power spectra along the diagonal of their likelihood covariance matrix to account for the ~ 10's of percent difference in the power spectra between `21cmFAST` and detailed numerical simulations (Mesinger et al. 2011; Zahn et al. 2011). In this work, we do not include this term because we do not claim that our constraints with `21cmFAST` are representative of the constraints that a numerical simulation might place. Rather, because we are using `21cmFAST` as our model, we have implicitly performed model selection up-front, and can only place quantifiable constraints on the parameters of this model. The $\boldsymbol{\Sigma}_{\rm S}$ term in our likelihood function therefore contains only the variance of the uncorrelated survey sensitivity (calculated in Section 4.1) along its diagonal.

If we were directly using our simulation to generate our model vectors $\mathbf{d}$, then this is indeed the likelihood function we would seek to minimize to produce our parameter constraints. However, in the regime where the simulation is either too computationally expensive or

[9] https://github.com/jpober/21cmSense

too slow to directly evaluate, we replace it with its emulated prediction $\mathbf{d}_{\rm E}$. The emulated prediction is an approximation, and can be related to the true simulation output as

$$\mathbf{d} = \mathbf{d}_{\rm E} - \boldsymbol{\delta}, \tag{20}$$

where $\boldsymbol{\delta}$ is offset of the emulator from the simulation. Naturally, we do not know $\boldsymbol{\delta}$ any more than we know $\mathbf{d}$ without directly evaluating the simulation; however, treating $\boldsymbol{\delta}$ as a random variable, we do have an estimate of its probability distribution given to us by either our Gaussian Process fit or by cross validation.

If we treat $\boldsymbol{\delta}$ as a Gaussian random variable, then its probability distribution is given as

$$\boldsymbol{\delta} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\rm E}), \tag{21}$$

where $\boldsymbol{\Sigma}_{\rm E}$ is the covariance matrix describing the uncertainty on $\mathbf{d}_{\rm E}$, and in principle can contain both uncorrelated errors along its diagonal terms and correlated errors in its off-diagonal terms. This variable can be estimated by using the output of the Gaussian Process, or it can be empirically calculated from cross validation. In the context of our worked example, we will always defer to calculating this variable empirically via cross validation. This means that $\boldsymbol{\Sigma}_{\rm E}$ is constant throughout the parameter space.

We can, and should, account for the fact that errors in our emulator's model predictions will induce errors into our likelihood. Writing out the likelihood in terms of Equation 20, we see that

$$\ln \mathcal{L}(\mathbf{y}|\boldsymbol{\theta}) \propto -\frac{1}{2}(\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta})^T \boldsymbol{\Sigma}_{\rm S}^{-1}(\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta}). \tag{22}$$

If we do not know the precise value of $\boldsymbol{\delta}$, we can propagate its uncertainty into the likelihood by marginalizing over its possible values. The derivation is given in Appendix A, and the result is that Equation 22 can be cast as

$$\ln \mathcal{L}(\mathbf{y}|\boldsymbol{\theta}) \propto -\frac{1}{2}(\mathbf{y} - \mathbf{d}_{\rm E})^T \left(\boldsymbol{\Sigma}_{\rm S} + \boldsymbol{\Sigma}_{\rm E}\right)^{-1} (\mathbf{y} - \mathbf{d}_{\rm E}), \tag{23}$$

where the marginalization process has left with a larger effective covariance matrix that is the direct sum of the error matrices of our two sources of error (observational error and emulator error). The inflated covariance matrix means we will recover broader constraints than if we had not included the emulator error budget. This is actually a desirable trait; it is better to recover broader constraints and be more confident that the truth parameters fall within those constraints rather than bias ourselves into falsely over-constraining regions of parameter space. In Section 5, we provide a test case for our emulator to see if it can indeed inform us when there has been a failure, such as the training set missing the location of some of the "true" parameters of a mock observation.

The likelihood function defined above will be what we use to directly constrain our model parameters of interest. Because this function is non-analytic, we use a Markov-Chain Monte Carlo (MCMC) sampling algorithm to find this function's peak and characterize its topology. There are a number of samplers that are well suited for this task. Our emulator employs `emcee`, the ensemble sampler algorithm from Foreman-Mackey et al. (2013), which is itself based on the affine-invariant sampling algorithm described in Goodman & Weare (2010).

4.3. *Broad Parameter Space Search*

To produce parameter constraints with an emulator, we must first construct a training set spanning the regions of parameter space we would like our MCMC sampler to explore. Due to the finite size of any training set, we need to set hard limits *a priori* on the breadth of the training set in parameter space. Our prior distribution on the model parameters is a straightforward way to make this choice. The astrophysical parameters of EoR and EoH, however, are highly unconstrained and in some cases span multiple orders of magnitude. In order to fully explore this vast parameter space with the emulator, we are left with a few options: (i) we could construct a sparse and wide training set, emulate at a highly approximate level, MCMC for the posterior distribution and then repopulate the parameter space with more training samples in the region of high probability and repeat, or (ii) use a gradient descent method to locate the general location of maximum probability. Both require direct evaluations of the simulation, but the former can be done in batch runs on a cluster and the latter is a sequential, iterative process (although it is typically not as computationally demanding as a full MCMC). For this work, we choose the former, and construct a wide rectangular training set from a Latin-Hypercube design consisting of 15×10^3 points. For one parameter in particular, f_X, we do not cover the entire range of its currently allowed values. In order to exhaustively explore the prior range of f_X, one might consider performing an initial gradient descent technique to localize its value. Because gradient descent algorithms are common in the scientific literature, we do not perform this test and assume we have already localized the value of f_X to within the extent of our initial training set, or assume we are comfortable adopting a prior on f_X spanning the width of our initial training set.

We use this initial training set to solve for an estimate of the hyperparameters for our Gaussian Process kernel as detailed in Section 2.3.1. With a training set consisting of over 10^4 points, we do not solve for a global predictive function of the eigenmode weights, but use a variant of the Learn-As-You-Go algorithm described in Section 2.3 for emulation. We k-fold cross validate on the training set and find that we can emulate the power spectra to accuracies ranging in the 50%-100% level depending on the redshift and k mode. While this is by no means "high-precision" emulation (and will pale in comparison to the precision achieved in our final em-

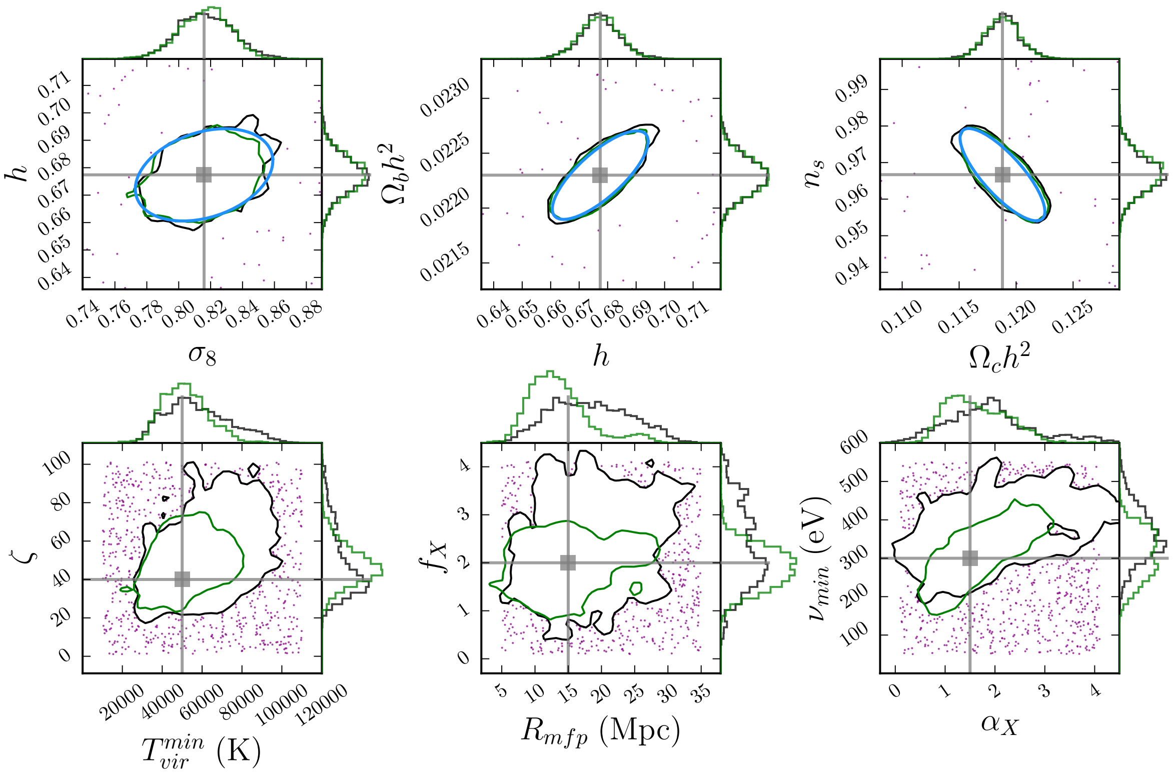

Figure 5. Posterior constraints for our initial parameter space exploration. The black contours represent 95% posterior credibility after emulating over our rectangular LH-design training set (shown as purple points). The green contours represent 95% posterior credibility after emulating over the LH training set plus a second, spherical training set populated within the contours of the initial constraints. The blue ellipses over the cosmological parameters show the 95% probability contour of our *Planck* prior distribution. The grey square shows the true underlying parameters of the observation. The histograms adjacent to the contour plots show the marginalized posterior distribution across each model parameter.

ulator runs for producing the ultimate parameter constraints), it is enough to refine our rough estimate of the location of the MAP point. We incorporate these projected emulator errors into our likelihood as described in Section 4.2. We adopt flat priors over the astrophysical parameters and covarying priors on the cosmological parameters representative of the *Planck* base TT+TE+EE+low-ℓ constraint, whose covariance matrix can be found in the `CosmoMC` code (Lewis & Bridle 2002).

We show the results of our initial parameter space search in Figure 5, where the black contours represent the 95% credible region of our constraint and the histograms show the posterior distribution marginalized across all other model parameters. The purple points in Figure 5 show samples from the initial LH training set, demarcating its hard bounds. The blue contours on the cosmological parameters show the 95% credible region of the prior distribution, showing that at this level of emulator precision the posterior distribution across the cosmology is dominated by the strong *Planck* prior. Even while emulating to a highly approximate level, we find that we can recover a rough and unbiased localization of the underlying truth parameter values. After localization, we can choose to further refine the density of our training set to produce better estimates of the MAP point and ensure we are converging upon the underlying true parameters. To do this, we extend the training set with an extra 5,000 samples sampled from a spherical multivariate Gaussian located near the truth parameters with a size similar to the width of the posterior distribution. The 95% credible region of the parameter constraints produced using an updated training set of 20,000 samples is shown in Figure 5 as the green contours, which shows an improvement in the MAP localization.

4.4. *Posterior Characterization*

With a reasonable estimate of the MAP location in hand, we now construct a dense training set so that we may emulate to higher precision. To do this, we build another training set consisting of 5000 samples from a packed Gaussian distribution and 500 samples from a LHSFS design (see Section 2.1) with a location near the truth parameters and size similar to the posterior distribution found in Section 4.3. To assess the accuracy of the emulator trained over this training set, we 5-fold cross validate over a subset of the data in the core of the training set. The results can be found in Figure 2, which shows we can emulate the power spectra to an

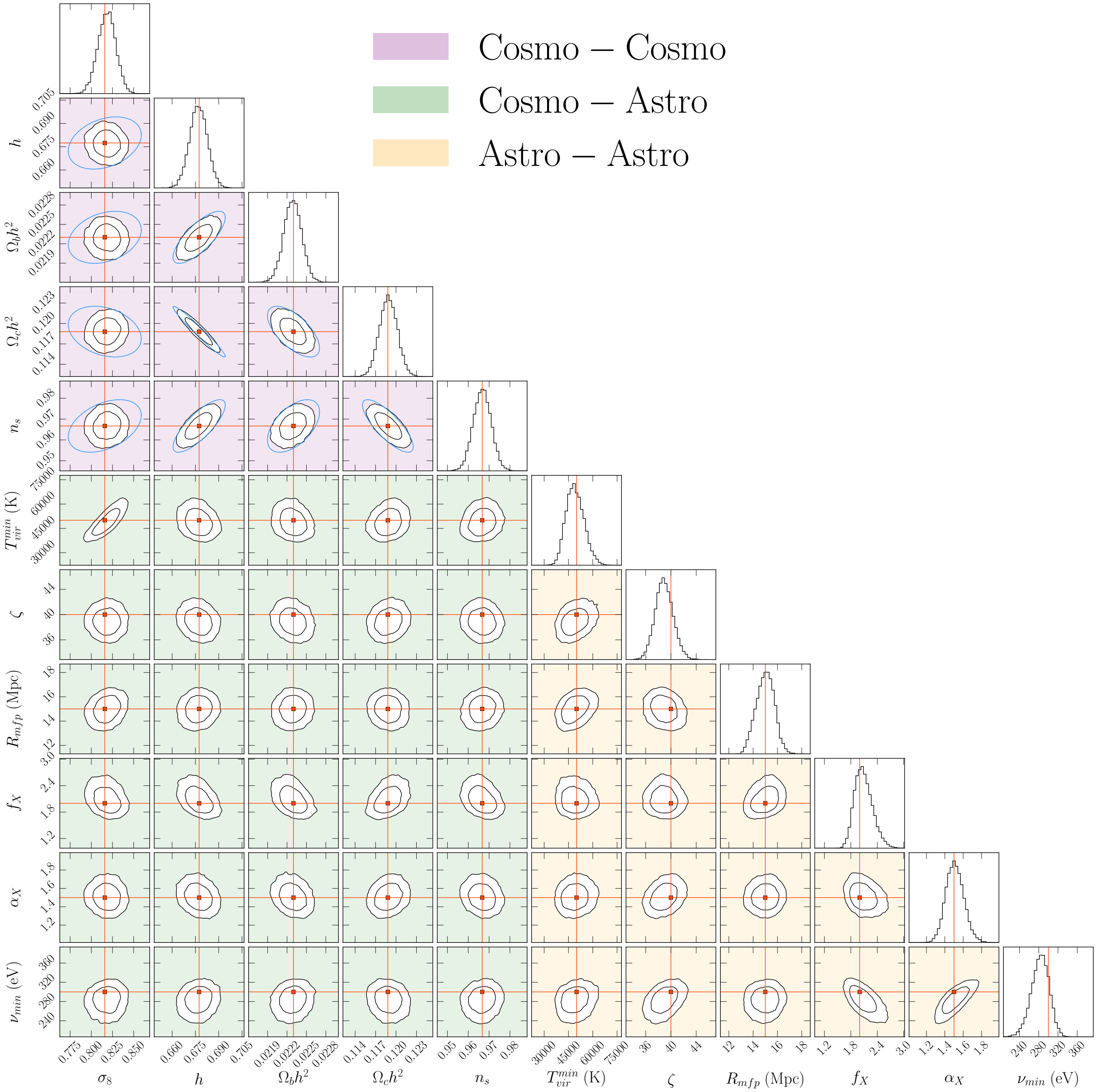

Figure 6. The joint posterior distribution of the eleven-parameter model, showing the 68% and 95% credible regions of the pairwise covariances (off-diagonal) and their marginalized distribution across each model parameter (diagonal). Purple-shaded boxes represent pairwise covariances between cosmological parameters; green-shaded boxes represent cosmological-astrophysical covariances, and yellow-shaded boxes represent astrophysical covariances. Blue contours on the cosmological covariances indicate the 95% credible region of the adopted prior distribution consistent with *Planck*. The underlying true parameters of the observation are marked as red squares with crosshairs.

accuracy of ∼ 10% for most of the data. More importantly, however, Figure 2 shows that the emulator error is always lower than the inherent observational survey error, and for the majority of the data is considerably lower. Nonetheless, we account for these projected emulator errors by adding them in quadrature with the survey error bars as described in Section 2.3.2. Our MCMC run setup involves 300 chains each run for ∼5,000 steps, yielding over 10^6 posterior samples. On a MacPro Desktop computer, this entire calculation takes ∼ 12 hours and utilizes ∼ 10 GB of memory.

The final characterization of the posterior distribution is found in Figure 6, where we show its marginalized pairwise covariances between all eleven model parameters and its fully marginalized distributions along the diagonal. With the exception of σ_8, the cosmological constraints are mostly a reflection of the strong *Planck* prior distribution (shown as blue contours). Compared to previous EoR forecasts of Pober et al. (2014); Ewall-Wice et al. (2016b); Greig et al. (2016), the strength of the EoR parameter degeneracies are weakened due to the inclusion of cosmological physics that washes out part of the covariance structure. This importance is exemplified by the strong degeneracy between the amplitude of clustering, σ_8, and the minimum virial temperature, $T_{\rm vir}^{\rm min}$. At a particular redshift, an increase in σ_8 increases the number of collapsed dark matter halos. At the same time, an increase in $T_{\rm vir}^{\rm min}$ suppresses the number of collapsed halos that can form stars, meaning they balance each other out in terms of their effect on the number of star forming halos present at any particular redshift. This degeneracy on the overall timing of EoR between these parameters is clearly seen in the animation tied to Figure 3 (see caption).

Compared to the recent work of Greig & Mesinger (2017a), who performed a full MCMC over EoR and EoH parameters with `21cmFAST`, our constraints are slightly stronger. This could be for a couple of reasons, including (i) the fact that they add an additional 20% modeling error onto their sampled power spectra and (ii) their choice of utilizing power spectra across 8 redshifts when fitting the mock observation, compared to our utilization of power spectra across 37 different redshifts.

The posterior distributions for each parameter marginalized across all others are shown in Figure 7, where they are compared against their input prior distributions. We see that the HERA331 experiment, with a moderate foreground avoidance scheme, will nominally place strong constraints on the EoR and EoH parameters of `21cmFAST` with respect to our currently limited prior information. For the cosmological parameters, the HERA likelihood alone is considerably weaker than the *Planck* prior; however, we can see that a HERA likelihood combined with a *Planck* prior can help strengthen constraints on certain cosmological parameters. Because 21 cm experiments are particularly sensitive to the location of the redshift peaks of the 21 cm signal,[10] parameters like σ_8, which control the overall clustering and thus affect the timing of reionization, are more easily constrained. Going further, Liu et al. (2016) showed that one can produce improved CMB cosmological parameter constraints by using 21 cm data to constrain the prior range of τ, which is a CMB nuisance parameter that is strongly degenerate with σ_8 and thus degrades its constraining power. Our 21 cm power spectrum constraint on σ_8 shown above does not include this additional improvement one can achieve by jointly fitting 21 cm and CMB data, which is currently being explored.

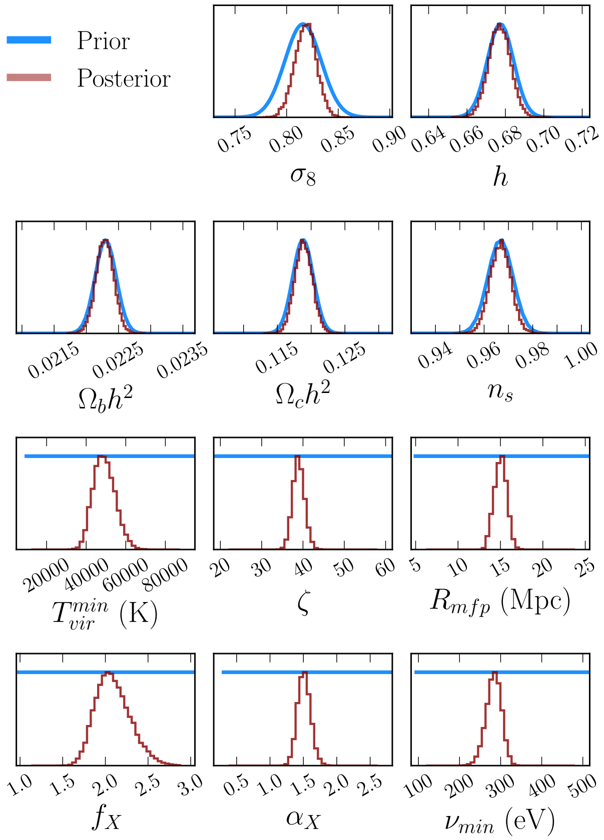

Figure 7. The posterior distribution of Figure 6 for each model parameter marginalized across all other parameters, compared against the adopted prior distributions. We adopt priors on the cosmological parameters consistent with *Planck* constraints, and adopt flat priors across the astrophysical parameters. We find that HERA will be able to produce $\sim 10\%$ level constraints on the astrophysical parameters and will help strengthen constraints on σ_8.

5. DISCUSSION

Here we discuss performance tests that help to further validate the efficacy of the emulator algorithm. We address the issue of what happens when the underlying true parameters lie at the edges or outside of the hard bounds of our training set, and make a direct comparison of the constraints produced by our emulator and a traditional brute-force MCMC algorithm.

5.1. *Comparison Against Direct MCMC*

An important performance test of the emulator algorithm is to compare its parameter constraints against the constraints produced by brute-force, where we directly call the simulation in our MCMC sampler. Of course we cannot do this for the simulation we would like to use—hence the need for the emulator—but we can do this if we use a smaller and faster simulation. For this test, we still use `21cmFAST` but only generate the power spectra at $z = \{8, 9, 10\}$ and ignore the spin temperature contribution to Δ^2_{21}, which drastically speeds up the simulation. In addition, we use a smaller simulation box-size and use a coarser resolution which yields additional speed-ups. We also restrict ourselves to varying only the three EoR astrophysics parameters described in Section 3.1.1, meaning we achieve faster MCMC con-

[10] Strong peaks and dips in the z evolution of Δ^2_{21} mean that slight deviations along z produce large deviations in Δ^2_{21}.

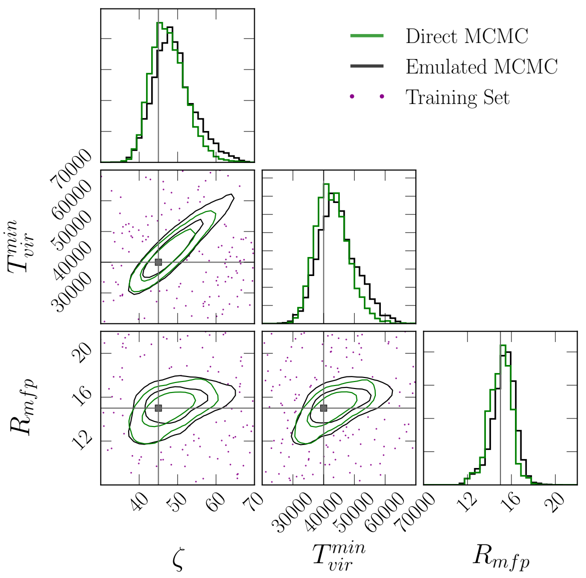

Figure 8. Emulator performance test comparing the constraints from the emulator (black) against brute-force constraints which directly evaluate the simulation (green). Both are able to produce unbiased constraints on the underlying "truth" parameters of the mock observation (square). The training set samples used to construct the emulator are shown in the background (purple points).

vergence. Using a coarser resolution and ignoring spin temperature fluctuations means the simulation is less accurate, but for the purposes of this test the simulation accuracy is irrelevant: we merely want to gauge if the emulator induces significant bias into constraints that we would otherwise produce by directly using the simulation.

Our mock observation is constructed using a realization of `21cmFAST` with fiducial EoR parameters (ζ, $T_{\rm vir}^{\rm min}$, $R_{\rm mfp}$) = (45, 4×10^3 K, 15 Mpc), and with the same fiducial cosmological parameters of Section 4.4. We place error bars over the fiducial realization using the same prescription of that described in Section 4.1, corresponding to the nominal sensitivity projections for the HERA331 experiment under "moderate" foreground avoidance. Similar to Section 4.1, we fit the power spectra across $0.1 \le k \le 2\,h\,{\rm Mpc}^{-1}$ in our MCMC likelihood function calls.

The result of the test is shown in Figure 8, where we plot the emulator and brute-force posterior constraints, as well as the training set samples used to construct the emulator. We find that the emulator constraints are in excellent agreement with the constraints achieved by brute-force. In the case where the emulator constraints slightly deviate from the brute-force constraints (in this case high ζ and high $T_{\rm vir}^{\rm min}$), the emulator deviations are conservative relative to the brute-force contours. In other words, the emulator constraints are always equal to or broader than the brute-force constraints, and do not falsely over-constrain the parameter space or induce systematic bias into the recovered MAP.

5.2. *Training Set Miscentering*

The ability of the emulator to produce reliable parameter constraints hinges principally on the assumption that the true parameter values lie within the bounds of the training set. If this is not the case, the emulator cannot make accurate predictions of the simulation behavior and is making a best guess based on extrapolation. In the case that emulator errors are not accounted for, this can lead to artificial truncation of the posterior distribution and create a false, over-constraining of the parameter space. This was observed to be problematic for a small number of figures in the 2015 *Planck* papers. Though the underlying cosmological constraints were unaffected, some illustrative plots employed an emulator-based method that seemed to be in tension with a more accurate direct MCMC method because the underlying parameters lay outside of the emulator's training set (Addison et al. 2016). It is therefore crucial to be able to assess if our training set encompasses the underlying truth parameters or if the training set has been miscentered. If the emulator can alert us when this is the case, we can repopulate a new training set in a different location and have greater confidence that the emulator is not falsely constraining the parameter space due to the finite width of the training set.

Given our method in Section 4 for localizing the parameter space via a sequence of training sets that iteratively converge upon the general location of the underlying true parameters, it is natural to ask, what if we made our final, compact training set a little too compact and missed the underlying MAP? How can we assess if this is the case, and if so, where do we populate the new training set? The most straightforward answer is to look at the posterior constraints compared to the width of the training set: if the posterior constraints run-up against the edge of the training set significantly, this may be an indication that we need to move the training set in that direction.

We perform such a test using our final compact training set and shift the position of mock observation's underlying truth parameters to the edges of the training set for parameters ζ and f_X: two particularly unconstrained parameters. Figure 9 shows the result, demonstrating the emulator's ability to shift the posterior contours when it senses that the MAP lies at the edge of the training set. In this case, we would know to generate more training samples near the region of high probability and retrain our emulator.

6. CONCLUSIONS

The next generation of 21 cm radio interferometric experiments with raw sensitivities at fiducial EoR levels are currently being built. The next few years will likely see either a detection and characterization of the 21 cm power spectrum or strong upper limits. However, interpreting these observations and connecting them to the high dimensional and weakly constrained parameter space of Cosmic Dawn is not straightforward. This is in

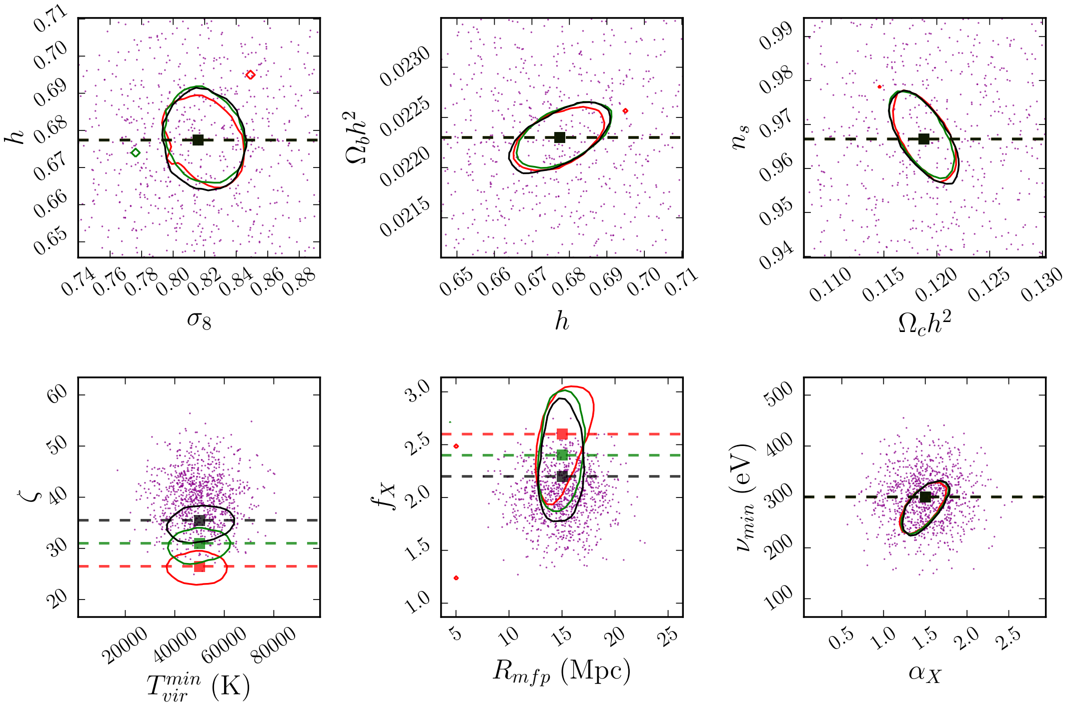

Figure 9. 95% credible regions of the posterior distribution while moving the true parameters of the mock observation away from the center of the training set, demonstrating the ability of the emulator to recover unbiased MAP constraints even when the training set does not directly overlap with the underlying "truth" parameters.

part because the relevant physics spans many order of magnitude in dynamic range and is non-linear, making detailed simulations computationally expensive, slow, and not conducive for iteration within an MCMC framework. While semi-numerical approaches have made progress exploring this parameter space, even they can have difficulty achieving speeds quick enough for MCMC techniques.

One way to address this challenge is to build an emulator for the simulation, which mimics the simulation output and is generally many orders of magnitude faster to evaluate. Emulators yield a few crucial advantages for parameter constraints over a direct MCMC approach. First, after the overhead of building the training set and training the emulator, running a parameter constraint analysis with an emulator is extremely cheap and quick. This feature is beneficial for MCMC repeatability: if we change our instrumental or survey covariance matrix, add more data to our observations, discover a bug in our data analysis pipeline, or find that a particular MCMC sampler is not exploring the parameter space properly, we would need to completely re-run these chains. Without an emulator, this could become a computationally prohibitive cost even for the most optimized semi-numerical simulations.

However, emulators also have their challenges. Most importantly, emulators are dependent on a training set, which will invariably have a finite size. This means we must *a priori* select a finite range over which our emulator is valid. This choice can be particularly hard to make for parameters that are highly unconstrained. This can be overcome by prefacing emulation with a gradient descent algorithm to localize parameters that are particularly unconstrained, or by incorporating prior information on these parameters from other probes.

In preparation for analyzing data sets from current and upcoming 21 cm experiments, we have built a fast emulator that can mimic simulations of Cosmic Dawn to high precision. We review the emulator algorithm present in our publicly available `emupy` and `pycape` packages, and discuss techniques for data compression, Gaussian Process regression and cross validation. We then apply our emulator to a simulation of Cosmic Dawn and demonstrate its ability to take a mock observation of the 21 cm signal and produce constraints on fundamental astrophysical and cosmological parameters. We find that a characterization of the 21 cm power spectrum from the upcoming Hydrogen Epoch of Reionization Array (HERA) will considerably narrow the allowed parameter space of reionization and X-ray heating parameters, and could help strengthen the constraints on σ_8 already set by *Planck*. The forecast presented in this work used an MCMC setup with 300 chains, each run for 5,000 steps, which took $\sim$ 12 hours on a MacPro desktop. While this forecast utilized a specific simulation of Cosmic Dawn, the emulator package and the parameter constraint framework outlined in this work are entirely independent: we could in principle emulate a whole suite Cosmic Dawn simulations ranging in their numerical implementations with only minor changes to the procedure outlined in this work. Although in this work we focus solely on the constraining power of the 21 cm power spectrum, the emulator framework can also be used to incorporate information from other probes of reionization, such as the hydrogen neutral fraction, averaged brightness temperature, electron scattering optical depth, galaxy clustering statistics and higher order statistics probes of the 21 cm field. Future work synthesizing these observables under the emulator framework will address this, enabling 21 cm intensity mapping efforts to live up to their theoretical promise of constraining a wide breadth of astrophysical and cosmological physics.

The authors would like to thank Grigor Aslanyan, Michael Betancourt, Josh Dillon, Danny Goldstein, Raul Monsalve, Danny Jacobs, Uroš Seljak, and Martin White for helpful discussions. AM and BG acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 638809: AIDA). AL, ARP, and NSK acknowledge support from the University of California Office of the President Multicampus Research Programs and Initiatives through award MR-15-328388, as well as from NSF CAREER award No. 1352519, NSF AST grant No.1129258, NSF AST grant No. 1410719, and NSF AST grant No. 1440343. AL acknowledges support for this work by NASA through Hubble Fellowship grant #HST-HF2-51363.001-A awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., for NASA, under contract NAS5-26555. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

APPENDIX

A. EMULATOR ERROR PROPAGATION

In this Appendix, we derive Equation 23 from Section 4.2 for propagating emulator interpolation error into our final likelihood function. We start by assuming the emulator prediction $\mathbf{d}_{\rm E}$ is offset from the true simulation output $\mathbf{d}$ by some amount $\boldsymbol{\delta}$, such that $\mathbf{d} \equiv \mathbf{d}_{\rm E} - \boldsymbol{\delta}$. In practice we do not know $\boldsymbol{\delta}$ precisely, but we do have an estimate of its possible values, given to us by our Gaussian Process fit (see Equation 11). In particular, we have an estimate of its probability distribution, modeled as a zero-mean Gaussian distribution with covariance given by $\boldsymbol{\Sigma}_{\rm E}$ (Equation 21), which will act as our prior distribution on $\boldsymbol{\delta}$.

Our likelihood function $\mathcal{L}$ is given by

$$\ln \mathcal{L}(\mathbf{y}|\boldsymbol{\theta}) \propto -\frac{1}{2}(\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta})^T \boldsymbol{\Sigma}_{\rm S}^{-1} (\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta}),$$

where $\mathbf{y}$ is a vector containing the observations, $\boldsymbol{\Sigma}_{\rm S}$ a matrix containing the survey error bars, and both $\mathbf{d}$ and $\boldsymbol{\delta}$ are functions of the model parameters $\boldsymbol{\theta}$. Multiplying this likelihood with a Gaussian prior on $\boldsymbol{\delta}$ yields the posterior distribution $\mathcal{P}$ given by

$$\begin{aligned}
\ln \mathcal{P} \propto & -\frac{1}{2}(\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta})^T \boldsymbol{\Sigma}_{\rm S}^{-1} (\mathbf{y} - \mathbf{d}_{\rm E} + \boldsymbol{\delta}) - \frac{1}{2}\boldsymbol{\delta}^T \boldsymbol{\Sigma}_{\rm E}^{-1} \boldsymbol{\delta} \\
= & -\frac{1}{2}(\mathbf{y} - \mathbf{d}_{\rm E})^T \boldsymbol{\Sigma}_{\rm S}^{-1} (\mathbf{y} - \mathbf{d}_{\rm E}) \\
& - \frac{1}{2}\boldsymbol{\delta}^T \left(\boldsymbol{\Sigma}_{\rm S}^{-1} + \boldsymbol{\Sigma}_{\rm E}^{-1}\right) \boldsymbol{\delta} + (\mathbf{y} - \mathbf{d}_{\rm E})^T \, \boldsymbol{\Sigma}_{\rm S}^{-1} \boldsymbol{\delta}
\end{aligned}$$

where in the second expression we factored out a term that is independent of $\boldsymbol{\delta}$.

We can account for $\boldsymbol{\delta}$'s influence on $\mathcal{L}$ by marginalizing (i.e. integrating) over it. To do so, we make use of the identity

$$\int \exp\left[-\frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}\cdot\mathbf{x}\right] d^n\mathbf{x} = \sqrt{\frac{(2\pi)^n}{\det \mathbf{A}}} \exp\left[\frac{1}{2}\mathbf{b}^T \mathbf{A}^{-1} \mathbf{b}\right], \tag{A1}$$

where $\mathbf{A}$ is an $n \times n$ real, symmetric matrix, and $\mathbf{b}$ and $\mathbf{x}$ are both vectors of length n. The resulting posterior distribution becomes

$$\begin{aligned}
\ln \mathcal{P} \propto & -\frac{1}{2}\left(\mathbf{y} - \mathbf{d}_{\rm E}\right)^T \times \\
& \left[\boldsymbol{\Sigma}_{\rm S}^{-1} - \boldsymbol{\Sigma}_{\rm S}^{-1}\left(\boldsymbol{\Sigma}_{\rm S}^{-1} + \boldsymbol{\Sigma}_{\rm E}^{-1}\right)^{-1} \boldsymbol{\Sigma}_{\rm S}^{-1}\right] \left(\mathbf{y} - \mathbf{d}_{\rm E}\right),
\end{aligned}$$

which can be simplified using identity

$$\left(\mathbf{A} + \mathbf{B}\right)^{-1} \equiv \mathbf{A}^{-1} + \mathbf{A}^{-1}\left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} \mathbf{A}^{-1}$$

to give

$$\ln \mathcal{P} \propto -\frac{1}{2}\left(\mathbf{y} - \mathbf{d}_{\rm E}\right)^T \left(\boldsymbol{\Sigma}_{\rm S} + \boldsymbol{\Sigma}_{\rm E}\right)^{-1} \left(\mathbf{y} - \mathbf{d}_{\rm E}\right). \tag{A2}$$

In other words, the emulator error covariance and the observational (or survey) error covariance simply add to form a new effective covariance that allows $\mathcal{L}$ to account for emulator error fluctuations in $\mathbf{d}_{\rm E}$. This result can also be reached by expressing $(\mathbf{y} - \mathbf{d}_{\rm E})$ as the sum of random variables $(\mathbf{y} - \mathbf{d})$ and $\boldsymbol{\delta}$, which we can think of as the convolution of two Gaussian distributions. If we assume each are normally distributed random variables with covariance $\boldsymbol{\Sigma}_{\rm S}$ and $\boldsymbol{\Sigma}_{\rm E}$ respectively, then the probability distribution of their sum is equivalent to the convolution of their individual probability distributions. The convolution theorem then tells us that the variance of the normal distribution describing their sum is just the sum of their individual variances, or $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\rm S} + \boldsymbol{\Sigma}_{\rm E}$.

REFERENCES

Addison, G., Huang, Y., Watts, D., Bennett, C., Halpern, M., Hinshaw, G., & Weiland, J. 2016, ApJ, 818, 132

Ali, Z., et al. 2015, ApJ, 809, 61

Asad, K., et al. 2015, MNRAS, 451, 3709

Aslanyan, G., Easther, R., & Price, L. 2015, JCAP, 9, 5

Baek, S., Di Matteo, P., Semelin, B., Combes, F., & Revaz, Y. 2009, A&A, 495, 389

Barger, V., Gao, Y., Mao, Y., & Marfatia, D. 2009, Phys. Lett. B, 673, 173

Barkana, R., & Loeb, A. 2002, ApJ, 578, 1

—. 2004, ApJ, 609, 474

—. 2005, ApJ, 626, 1

Barry, N., Hazelton, B., Sullivan, I., Morales, M., & Pober, J. 2016, MNRAS, 461, 3135

Battaglia, N., Trac, H., Cen, R., & Loeb, A. 2013, ApJ, 776, 81

Beardsley, A., et al. 2012, MNRAS, 425, 1781

—. 2016, ApJ, 833, 102

Becker, G., Bolton, J., Madau, P., Pettini, M., Ryan-Weber, E., & Venemans, B. 2015, MNRAS, 447, 3402

Beers, T., Flynn, K., & Gebhardt, K. 1990, AJ, 100, 32

Bouwens, R., et al. 2015, ApJ, 803, 34

Chapman, E., Zaroubi, S., Abdalla, F. B., Dulwich, F., Jelić, V., & Mort, B. 2016, MNRAS, 458, 2928

Chapman, E., et al. 2012, MNRAS, 423, 2518

—. 2013, MNRAS, 429, 165

Choudhury, T., Haehnelt, M., & Regan, J. 2009, MNRAS, 394, 960

Clesse, S., Lopez-Honorez, L., Ringeval, C., Tashiro, H., & Tytgat, M. 2012, PhRvD, 86, 123506

Cohen, A., Fialkov, A., Barkana, R., & Lotem, M. 2016, ArXiv e-prints: 1609.02312

Czekala, I., Andrews, S., Mandel, K., Hogg, D., & Green, G. 2015, ApJ, 812, 128

Das, A., Mesinger, A., Pallottini, A., Ferrara, A., & Wise, J. 2017, ArXiv e-prints: 1702.00409

Datta, A., Bowman, J., & Carilli, C. 2010, ApJ, 724, 526

DeBoer, D., et al. 2017, PASP, 129, 45001

Dillon, J., & Parsons, A. 2016, ApJ, 826, 181

Dillon, J., et al. 2014, PhRvD, 89, 23002

—. 2015, PhRvD, 91, 123011

Ewall-Wice, A., Dillon, J., Liu, A., & Hewitt, J. 2016a, ArXiv e-prints: 1610.02689

Ewall-Wice, A., Hewitt, J., Mesinger, A., Dillon, J., Liu, A., & Pober, J. 2016b, MNRAS, 458, 2710

Ewall-Wice, A., et al. 2016c, MNRAS, 460, 4320

Fan, X., et al. 2006, AJ, 132, 117

Fendt, W., & Wandelt, B. 2007, ApJ, 654, 2

Fialkov, A., & Barkana, R. 2014, MNRAS, 445, 213

Fialkov, A., Barkana, R., Pinhas, A., & Visbal, E. 2014a, MNRAS, 437, L36

Fialkov, A., Barkana, R., & Visbal, E. 2014b, Nature, 506, 197

Field, G. 1958, Proc. IRE, 46, 240

Field, S., Galley, C., Hesthaven, J., Kaye, J., & Tiglio, M. 2014, Phys. Rev. X, 4, 31006

Finkelstein, S., et al. 2015, ApJ, 810, 71

Foreman-Mackey, D., Hogg, D., Lang, D., & Goodman, J. 2013, PASP, 125, 306

Furlanetto, S. 2006, MNRAS, 371, 867

Furlanetto, S., McQuinn, M., & Hernquist, L. 2006, MNRAS, 365, 115

Furlanetto, S., Zaldarriaga, M., & Hernquist, L. 2004, ApJ, 613, 1

Geil, P., & Wyithe, J. 2008, MNRAS, 386, 1683

Ghara, R., Choudhury, T., & Datta, K. 2015, MNRAS, 447, 1806

Gnedin, N. 2014, ApJ, 793, 29

Goodman, J., & Weare, J. 2010, Comm. App. Math. Comp. Sci., 5

Greig, B., & Mesinger, A. 2015, MNRAS, 449, 4246

Greig, B., & Mesinger, A. 2017a, ArXiv e-prints: 1705.03471

—. 2017b, MNRAS, 465, 4838

Greig, B., Mesinger, A., & Koopmans, L. 2015, ArXiv e-prints: 1509.03312

Greig, B., Mesinger, A., & Pober, J. 2016, MNRAS, 455, 4295

Habib, S., Heitmann, K., Higdon, D., Nakhleh, C., & Williams, B. 2007, PhRvD, 76, 83503

Haiman, Z., Abel, T., & Rees, M. 2000, ApJ, 534, 11

Haiman, Z., Rees, M., & Loeb, A. 1997, ApJ, 476, 458

Heitmann, K., Higdon, D., Nakhleh, C., & Habib, S. 2006, ApJL, 646, L1

Heitmann, K., Higdon, D., White, M., Habib, S., Williams, B., Lawrence, E., & Wagner, C. 2009, ApJ, 705, 156

Hogan, C., & Rees, M. 1979, MNRAS, 188, 791

Iliev, I., Mellema, G., Ahn, K., Shapiro, P., Mao, Y., & Pen, U.-L. 2014, MNRAS, 439, 725

Jacobs, D., et al. 2015, ApJ, 801, 51

Kaurov, A., & Gnedin, N. 2016, ApJ, 824, 114

Kohn, S., et al. 2016, ApJ, 823, 88

Koopmans, L., et al. 2015, Adv. Astrophys. with Sq. Km. Array, 1

Kuhlen, M., & Faucher-Giguère, C.-A. 2012, MNRAS, 423, 862

Kuhlen, M., Madau, P., & Montgomery, R. 2006, ApJL, 637, L1

Kulkarni, G., Choudhury, T., Puchwein, E., & Haehnelt, M. 2016, MNRAS, 463, 2583

Lewis, A., & Bridle, S. 2002, PhRvD, D66, 103511

Lewis, A., Challinor, A., & Lasenby, A. 2000, ApJ, 538, 473

Liu, A., & Parsons, A. 2016, MNRAS, 457, 1864

Liu, A., Parsons, A., & Trott, C. 2014a, PhRvD, 90, 23018

—. 2014b, PhRvD, 90, 23019

Liu, A., Pritchard, J., Allison, R., Parsons, A., Seljak, U., & Sherwin, B. 2016, PhRvD, 93, 43013

Livermore, R., Finkelstein, S., & Lotz, J. 2017, ApJ, 835, 113

Loeb, A., & Furlanetto, S. R. 2013, The First Galaxies In The Universe (Princeton University Press, Princeton, NJ)

Madau, P., Meiksin, A., & Rees, M. 1997, ApJ, 475, 429

Mao, Y., Tegmark, M., McQuinn, M., Zaldarriaga, M., & Zahn, O. 2008, PhRvD, 78, 23529

McGreer, I., Mesinger, A., & D'Odorico, V. 2015, MNRAS, 447, 499

McQuinn, M., Zahn, O., Zaldarriaga, M., Hernquist, L., & Furlanetto, S. 2006, ApJ, 653, 815

Mellema, G., Iliev, I., Pen, U.-L., & Shapiro, P. 2006, MNRAS, 372, 679

Mesinger, A. 2016, Underst. Epoch Cosm. Reionization Challenges Prog., 423

Mesinger, A., Ewall-Wice, A., & Hewitt, J. 2014, MNRAS, 439, 3262

Mesinger, A., Ferrara, A., & Spiegel, D. 2013, MNRAS, 431, 621

Mesinger, A., & Furlanetto, S. 2007, ApJ, 669, 663

Mesinger, A., Furlanetto, S., & Cen, R. 2011, MNRAS, 411, 955

Mesinger, A., McQuinn, M., & Spergel, D. 2012, MNRAS, 422, 1403

Mineo, S., Gilfanov, M., & Sunyaev, R. 2012, MNRAS, 419, 2095

Mirocha, J., Furlanetto, S., & Sun, G. 2017, MNRAS, 464, 1365

Moore, D., et al. 2015, ArXiv e-prints: 1502.05072

Morales, M., Hazelton, B., Sullivan, I., & Beardsley, A. 2012, ApJ, 752, 137

Morales, M., & Wyithe, J. 2010, ARA&A, 48, 127

Paardekooper, J.-P., Khochfar, S., & Dalla Vecchia, C. 2015, MNRAS, 451, 2544

Paciga, G., et al. 2013, MNRAS, 433, 639

Pacucci, F., Mesinger, A., Mineo, S., & Ferrara, A. 2014, MNRAS, 443, 678

Parsons, A., Pober, J., Aguirre, J., Carilli, C., Jacobs, D., & Moore, D. 2012a, ApJ, 756, 165

Parsons, A., Pober, J., McQuinn, M., Jacobs, D., & Aguirre, J. 2012b, ApJ, 753, 81

Parsons, A., et al. 2014, ApJ, 788, 106

Patil, A., et al. 2016, MNRAS, 463, 4317

—. 2017, ApJ, 838, 65

Pedregosa, F., et al. 2012, ArXiv e-prints: 1201.0490

Petri, A., Liu, J., Haiman, Z., May, M., Hui, L., & Kratochvil, J. 2015, PhRvD, 91, 103511

Planck Collaboration et al. 2016, A&A, 594, A13

Pober, J., et al. 2013a, ApJL, 768, L36

—. 2013b, AJ, 145, 65

—. 2014, ApJ, 782, 66

—. 2015, ApJ, 809, 62

—. 2016, ApJ, 819, 8

Pritchard, J., & Furlanetto, S. 2007, MNRAS, 376, 1680

Pritchard, J., & Loeb, A. 2012, Reports Prog. Phys., 75, 86901

Rasmussen, C. E., & Williams, C. K. I. 2006, Gaussian processes for machine learning (MIT Press), 248

Robertson, B., Ellis, R., Furlanetto, S., & Dunlop, J. 2015, ApJL, 802, L19

Ross, H. E., Dixon, K. L., Iliev, I. T., & Mellema, G. 2017, MNRAS, 468, 3785

Santos, M., Ferramacho, L., Silva, M., Amblard, A., & Cooray, A. 2010, MNRAS, 406, 2421

Schneider, M., Holm, Ó., & Knox, L. 2011, ApJ, 728, 137

Scott, D., & Rees, M. 1990, MNRAS, 247, 510

Shimabukuro, H., & Semelin, B. 2017, MNRAS, 468, 3869

Sobacchi, E., & Mesinger, A. 2014, MNRAS, 440, 1662

Sun, G., & Furlanetto, S. 2016, MNRAS, 460, 417

Switzer, E., & Liu, A. 2014, ApJ, 793, 102

Thomas, R., et al. 2009, MNRAS, 393, 32

Thyagarajan, N., et al. 2013, ApJ, 776, 6

—. 2015a, ApJL, 807, L28

—. 2015b, ApJ, 804, 14

Tozzi, P., Madau, P., Meiksin, A., & Rees, M. 2000, ApJ, 528, 597

Trac, H., & Gnedin, N. 2011, Adv. Sci. Lett., 4, 228

Trott, C., Wayth, R., & Tingay, S. 2012, ApJ, 757, 101

Vedantham, H., Udaya Shankar, N., & Subrahmanyan, R. 2012, ApJ, 745, 176

Vedantham, H., et al. 2015, MNRAS, 450, 2291

Warszawski, L., Geil, P., & Wyithe, J. 2009, MNRAS, 396, 1106

Wolz, L., Abdalla, F. B., Blake, C., Shaw, J. R., Chapman, E., & Rawlings, S. 2014, MNRAS, 441, 3271

Wouthuysen, S. 1952, AJ, 57, 31

Xu, H., Wise, J., Norman, M., Ahn, K., & O'Shea, B. 2016, ApJ, 833, 84

Zahn, O., Lidz, A., McQuinn, M., Dutta, S., Hernquist, L., Zaldarriaga, M., & Furlanetto, S. 2007, ApJ, 654, 12

Zahn, O., Mesinger, A., McQuinn, M., Trac, H., Cen, R., & Hernquist, L. 2011, MNRAS, 414, 727

Zahn, O., et al. 2012, ApJ, 756, 65