Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **534,** 1299–1316 (2024)
Advance Access publication 2024 September 17

https://doi.org/10.1093/mnras/stae2153

Deep learning the intergalactic medium using Lyman-alpha forest at $4 \leq z \leq 5$

Fahad Nasir,[1★] Prakash Gaikwad,[1] Frederick B. Davies,[1] James S. Bolton,[2] Ewald Puchwein[3] and Sarah E. I. Bosman[1,4]

[1] *Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany*
[2] *School of Physics and Astronomy, University of Nottingham, University Park, Nottingham NG7 2RD, UK*
[3] *Leibniz-Institut für Astrophysik Potsdam, An der Sternwarte 16, D-14482 Potsdam, Germany*
[4] *Institute for Theoretical Physics, Heidelberg University, Philosophenweg 12, D-69120 Heidelberg, Germany*

Accepted 2024 September 13. Received 2024 September 13; in original form 2024 April 8

ABSTRACT

Unveiling the thermal history of the intergalactic medium (IGM) at $4 \leq z \leq 5$ holds the potential to reveal early onset He II reionization or lingering thermal fluctuations from H I reionization. We set out to reconstruct the IGM gas properties along simulated Lyman-alpha (Lyα) forest data on pixel-by-pixel basis, employing deep neural networks. Our approach leverages the Sherwood-Relics simulation suite, consisting of diverse thermal histories, to generate mock spectra. Our convolutional and residual networks with likelihood metric predict the Lyα optical depth-weighted density or temperature for each pixel in the Lyα forest skewer. We find that our network can successfully reproduce IGM conditions with high fidelity across range of instrumental signal-to-noise ratio. These predictions are subsequently translated into the temperature–density plane, facilitating the derivation of reliable constraints on thermal parameters. This allows us to estimate temperature at mean cosmic density, T_0, with 1σ confidence, $\delta T_0 \lesssim 1000$ K, using only one 20 h^{-1} cMpc sightline ($\Delta z \simeq 0.04$) with a typical reionization history. Existing studies utilize redshift path-length comparable to $\Delta z \simeq 4$ for similar constraints. We can also provide more stringent constraints on the slope (1σ confidence interval, $\delta\gamma \lesssim 0.1$) of the IGM temperature–density relation as compared to other traditional approaches. We test the reconstruction on a single high signal-to-noise observed spectrum (20 h^{-1} cMpc segment) and recover thermal parameters consistent with current measurements. This machine learning approach has the potential to provide accurate yet robust measurements of IGM thermal history at the redshifts in question.

Key words: methods: numerical – – .

1 INTRODUCTION

Accurately understanding the thermal history of the intergalactic medium (IGM) stands as a fundamental goal in astronomy, as it offers crucial insights into the timing and duration of the H I and He II reionization epochs. One of the most robust methods for probing this thermal history is through the examination of Lyα absorption, commonly known as the Lyα forest, observed in the spectra of quasars (Becker, Bolton & Lidz 2015). This approach is grounded in the concept of photoheating, where the IGM temperature increases due to H I and He II ionizing photons (Miralda-Escudé & Rees 1994). The Doppler motions of the heated IGM are then imprinted on to the absorption lines of the Lyα forest.

Leveraging on this idea, the literature has explored various statistics that demonstrate a range of sensitivity to the thermal state. Studies exploit Lyα flux power spectrum suppression on small scales (Zaldarriaga, Hui & Tegmark 2001; Croft et al. 2002; Zaroubi et al. 2006; Viel et al. 2013; Boera et al. 2019; Walther et al. 2019), the Lyα line width distributions (Haehnelt & Steinmetz 1998; Ricotti, Gnedin & Shull 2000; Schaye et al. 2000; McDonald et al. 2001; Bolton et al. 2012, 2014; Rudie, Steidel & Pettini 2012; Hiss et al. 2018; Telikova, Shternin & Balashev 2019), probability distribution of Lyα flux (Lidz et al. 2006; Bolton et al. 2008; Calura et al. 2012; Lee et al. 2015), curvature of Lyα flux (Becker et al. 2011; Boera et al. 2014, 2016; Padmanabhan, Srianand & Choudhury 2015), statistics of wavelet amplitudes (Meiksin 2000; Theuns, Schaye & Haehnelt 2000; Zaldarriaga 2002; Lidz et al. 2010; Garzilli et al. 2012; Wolfson et al. 2021), utilizing the entire b–$N_{\rm H\,I}$ distribution (Hiss et al. 2019; Hu et al. 2023), and combining an ensemble of statistics (Gaikwad et al. 2021). None the less, despite recent progress, the pursuit of more precise methods persists, especially in the redshift range of $4 \leq z \leq 5$, which can provide insights into the early stages of He II reionization or any residual heating effects from H I reionization.

The common theme of the previous studies has been to constraint the thermal parameters, namely the normalization (T_0) and slope (γ) of the expected power-law relating temperature (T) and normalized cosmic overdensity, Δ, of the IGM aftermath of reionization, $T = T_0\Delta^{\gamma-1}$ (Hui & Gnedin 1997; McQuinn & Upton Sanderbeck 2016). While this approach has been valuable, it primarily offers a statistical description of IGM gas conditions. The richness of information present in the forest goes beyond what these thermal parameters can

[★] E-mail: nasir@mpia.de

capture. Moreover, they are typically provided for the entire data set at a given redshift. The methods are not sensitive enough to provide constraints for individual sightlines or to capture the variations within due to thermal fluctuations at $z \gtrsim 5$ (D'Aloisio, McQuinn & Trac 2015; Keating, Puchwein & Haehnelt 2018; Gaikwad et al. 2020). We aim to harness deep learning methods that can somewhat avoid the loss of information inherent to the standard summary statistics.

Recently, we have witnessed a notable surge in studies exploring deep learning (for review, see Goodfellow, Bengio & Courville 2016) in the context of quasar spectra. To highlight a few, generating quasar (missing) spectra based on its properties (Eilers et al. 2022), generating Lyα forest for large surveys using only N-body simulation (Harrington et al. 2022), predicting Lyα optical depth using transmitted flux (Huang, Croft & Arora 2021), reconstructing the IGM temperatures on large scales around quasars (Wang, Croft & Shaw 2022), and using Lyα forest spectra to obtain constraints on IGM thermal parameters using ideal mock spectra at $z = 2.2$ (Nayak et al. 2024).

Our primary objective is to utilize deep learning techniques for the precise reconstruction of IGM densities and temperatures, pixel by pixel, within the Lyα forest. By the aid of this method, the thermal parameters can also be inferred from the reconstructed conditions. By virtue of the sheer number of data points at our disposal, we can infer thermal parameters even for a single (20 h^{-1} cMpc in this paper) sightline. Consequently, the constraints we obtain can be many times more powerful and accurate in comparison to traditional approaches. Moreover, since this approach strives to reconstruct the entire conditions along a given sightline, it naturally extends its capability to estimate thermal fluctuations that persist after the completion of H I reionization or due to patchy He II reionization. However, we leave the analysis of inhomogeneous reionization models to future work, which would require an extensive suite of simulations for the training. We assume an IGM with a uniform thermal state for this paper.

There are few key considerations before we adapt deep learning methods for our purpose. We think that it is crucial to incorporate uncertainties, in particular for this reconstruction as actual conditions of IGM are unknown a priori. Quantifying the uncertainties due to lack of usable flux or due to parameter degeneracy becomes important, in particular at higher redshifts where the Lyα forest is more opaque. Furthermore, the optimal design of a neural network demands careful attention. A systematic grid search encompassing all relevant hyperparameters allows us to harness the full potential of these techniques. Owing to their versatility and complexity, deep learning methods are susceptible to overfitting issues. This implies that network performance may significantly deteriorate when applied to different but related data sets. Consequently, we take necessary precautions to mitigate overfitting, such as rigorously testing performance on a dedicated test data set using NYX code.

In this work, we address and resolve the aforementioned challenges. We build a neural network that maximizes the likelihood of a given data set with uncertainty estimates as a natural outcome. We also mitigate the effect of overfitting by carefully analysing predictions on a test data set. Furthermore, we conduct a grid search to optimize the network architecture and all associated hyperparameters for training.

Our paper is structured as follows: In Section 2, we provide details of the cosmological simulations employed in this study. Our focus is simulation outputs at redshifts $z = 4$–5. Section 3 delves into the fundamental aspects of the neural network and finding an optimal network configuration. Section 4 presents our core findings. We present the reconstructions along example Lyα skewers. We also show recovery of thermal parameters through various models, using different signal-to-noise (S/N). We also present a reconstruction of a 20 h^{-1} cMpc segment on observational spectrum at $z = 4$. In Section 5, we offer our concluding remarks. We present a comparison of Lyα optically depth-weighted and real-space thermal parameter distributions in the Appendix.

Table 1. The summary of simulations used in this work taken from Sherwood-Relics suite. The columns represent name, H I photoheating rescaling factors, and the redshift of reionization (the redshift where the global neutral fraction reaches values below 10^{-3}), $z_{\rm re}$. All simulations have box size length of 20 h^{-1} cMpc with 1024^3 gas and dark matter particles. The last row shows the runs that are used only for testing purposes.

Model	H I photoheating	$z_{\rm re}$
FIDUCIAL/COLD/HOT	1, 0.5, 2	6.0
ZR525/COLD/HOT	1, 0.5, 2	5.4
ZR675/COLD/HOT	1, 0.5, 2	6.7
ZR750/COLD/HOT	1, 0.5, 2	7.4
G10/G14/G16	Different γ	6.0
NYX-EARLY/NYX-LATE	–	9.7, 6.66

2 HYDRODYNAMICS SIMULATIONS

The neural network relies on appropriate data sets for training, validation, and testing purposes. We use a subset of cosmological simulations from the Sherwood-Relics[1] suite, based on the Sherwood simulations (Bolton et al. 2017). These simulations were performed using a modified version of P-GADGET3 code, an updated version of the P-GADGET2 (Springel 2005). In this section, we will briefly describe the key features of these simulations that form our training, validation, and test data sets. For details, we refer the reader to Puchwein et al. (2023). These simulations are particularly designed for Lyα forest absorption spectra with various reionization and thermal histories. In this work, we restrict ourselves to runs with a standard cold dark matter cosmology. We summarize the runs we use in Table 1 (first five rows).

We use runs with a 20 h^{-1} cMpc box length on each side evolved with 1024^3 dark matter and gas particles. The simulations employ a spatially uniform but time-varying ultraviolet background (UVB; Puchwein et al. 2019), and vary the redshift of reionization, $z_{\rm re}$, and the thermal parameters, namely T_0 and γ. This is achieved by rescaling and shifting the photoionization and photoheating rates. There are 12 simulations on the $z_{\rm re}$–T_0 grid (first row through the fourth row in Table 1). Additionally, there are three runs for $\gamma =$ 1.0, 1.4, and 1.6 fixed at $z = 4$ (fifth row in Table 1). To estimate T_0 and γ for a given simulation run, we calculate the median $\log T$ on $\log \Delta = -0.4$ to 0.2 bins with a size of 0.1, and fit a line through these binned values. Our method is very similar to the one described in Gaikwad et al. (2017).

Finally, to verify the robustness and generalization of our network predictions, we include two more hydrodynamical simulations using the NYX code (Almgren et al. 2013) for testing purposes only. All runs have a box size of 20 h^{-1} cMpc and were evolved with 1024^3 gas and dark matter particles, using rescaled versions of the Haardt & Madau (2012) UVB. The runs mimic very early and relatively late reionization histories as summarized in the last row of Table 1. The reionization and thermal histories of these runs do not closely match

[1] https://www.nottingham.ac.uk/astronomy/sherwood-relics/

any of our training runs from Sherwood-Relics. We refer the reader to Oñorbe, Hennawi & Lukić (2017) for more details.

We extract 20 h^{-1} cMpc long 5000 skewers from each simulation running parallel to the box axes while tracing various quantities at $z = [4,\ 4.4,\ 5]$. We rescale the Lyα optical depths to match the mean flux $\langle F \rangle = [0.4255,\ 0.3216,\ 0.135]$, at each redshift that is taken from Lyα forest measurements (Becker & Bolton 2013; Bosman et al. 2018). We convolve the spectra with Gaussian line profile with full width at half-maximum (FWHM) =6 km s^{-1} to mimic the resolution of Very Large Telescope - Ultraviolet and Visual Echelle Spectrograph (VLT-UVES) instrument. The pixel scale for our mock spectra is ∼2.45 km s^{-1} at $z = 4$. Note that we add Gaussian-distributed (or observational) noise with a fixed S/N during the training phase.

We form pairs of Lyα flux and the corresponding logarithm of the Lyα optical depth-weighted density and temperature ($\log \Delta_\tau$, $\log T_\tau$) skewers for our data set. It is infeasible to recover the real-space quantities ($\log \Delta$ and $\log T$) from a velocity-space Lyα flux skewer, as they are simply washed out by small-scale peculiar velocities of the absorbers due to structure formation. We use a Gaussian line profile (instead of a Voigt profile) to obtain $\log \Delta_\tau$ and $\log T_\tau$ skewers. We found that in some rare sightlines, extremely high densities can affect the weighted quantity along the significant length of the skewer due to extended damping wing. The Lyα forest is evaluated in the standard way using the Voigt line profile. We form one data set by aggregating all skewers from the Sherwood suite (rows one through five in Table 1). The NYX-EARLY and NYX-LATE runs are only utilized at the testing stage.

3 NEURAL NETWORK

3.1 Metric for network

A network learns weights and biases during training using a gradient-based optimization method. The main objective is that once the network is trained, the errors between actual and predicted quantities are minimized. This is solely judged on a metric and conventionally it is either mean absolute error or mean squared error. However, as we want a handle on the uncertainties on predictions, we chose the negative logarithm of the Gaussian likelihood, $-\log \mathcal{L}$, as our metric:

$$-\log \mathcal{L} = \frac{1}{N} \sum_i \left[(Y_i - Y_{i,\mathrm{pred.}})^2/\sigma^2_{i,\mathrm{pred.}} + \log \left(\frac{1}{\sigma^2_{i,\mathrm{pred.}}} \right) \right]. \quad (1)$$

Here, the sum runs over all the pixels. The metric is normalized by the total skewers in a training or validation split, N. The Y_i, $Y_{i,\mathrm{pred.}}$, and $\sigma^2_{i,\mathrm{pred.}}$ are actual mean, predicted mean, and predicted standard deviation of the desired quantity, respectively.

The underlying assumption for our metric is that the predicted quantity follows a Gaussian distribution at every pixel along the sightline. Furthermore, we treat each pixel as independent and ignore any covariances at the training phase. This is to make training feasible and avoid predicting large covariance matrices that quickly become impractical for training a neural network. We later tackle this problem during the prediction step. We will train two independent networks for $\log \Delta_\tau$ and $\log T_\tau$, therefore two values for metric (i.e. $-\log \mathcal{L}$) without taking into consideration the correlations between quantities during training.

3.2 Building deeper networks with building blocks

Before we tackle the problem of designing an optimal network, we will delve into the basic building blocks for our networks. Three basic processing layers that are central to our networks are shown in Fig. 1. We will refer to them as convolutional (top), residual (middle), and dense (bottom) layers, respectively.

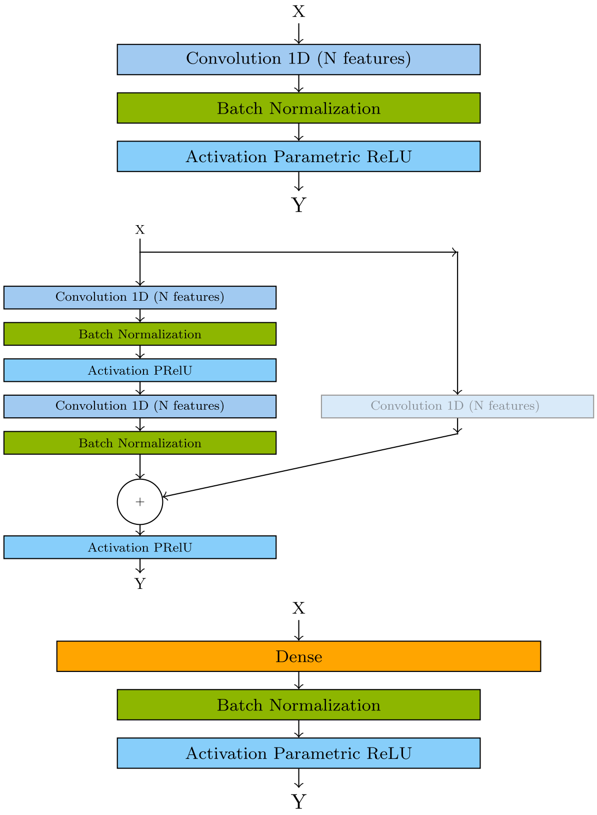
X
Convolution 1D (N features)
Batch Normalization
Activation Parametric ReLU
Y
X
Convolution 1D (N features)
Batch Normalization
Activation PReIU
Convolution 1D (N features)
Batch Normalization
Convolution 1D (N features)
+
Activation PReIU
Y
X
Dense
Batch Normalization
Activation Parametric ReLU
Y

Figure 1. The schematic for the basic layers that forms the networks used in this work. *Top*: The convolutional layer extracts *N* features using 3 pixel-wide convolutional kernels. *Middle*: The residual layer implements two stacked convolutional layers with a skipping connection. The input, X, is fed directly to the last layer before activation. We stack several convolutional or residual layers to form one block, which are placed in sequence to form either ConvNet or ResNet. *Bottom*: The dense layer performs dot product between the input and the weight matrix. All layers share the batch normalization and parametric rectilinear unit as activation that is an elementwise operation. The dense layer is the last layer in our network.

The main objective of the convolutional layer (for introduction, see O'Shea & Nash 2015) is to extract a fixed number of features through a set of convolution kernels. We fix the kernel to be 3 pixels (we have also done trials with 5 and 7) typical for such networks. Note that although the kernel size is set, the subsequent convolutional layer followed with pooling aids the networks to learn complex features on larger scales. The batch normalization layer standardizes samples of the current batch (a subsample of the training data set) during training and encourages faster learning (Ioffe & Szegedy 2015). For inhomogeneous data sets, the network can be exposed to a biased batch, which can slow down the convergence of the metric during training. This layer helps to mitigate this effect. The activation layer introduces non-linearity into the network using the Parametric Rectified Linear Unit (PReLU) function. This function retains input when it is positive but scales it with a trainable factor when negative. PReLU is a piecewise linear function that has the advantage of being cost-effective to compute.

The underlying idea of the residual layer is to skip connections between two consecutive convolutional layers by directly adding input to output before activation. Note that the inputs cannot simply

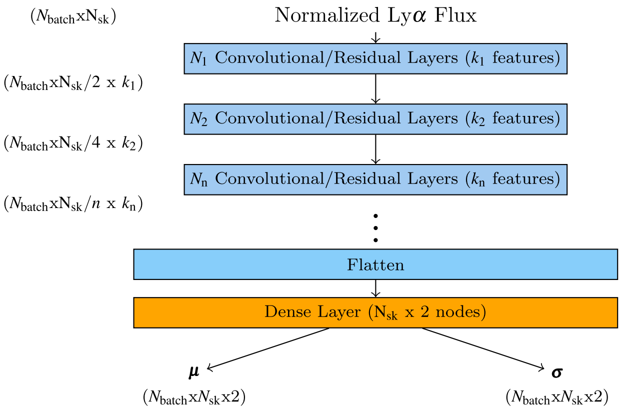
($N_{\rm batch}$xN$_{\rm sk}$)
Normalized Lyα Flux
N_1 Convolutional/Residual Layers (k_1 features)
($N_{\rm batch}$xN$_{\rm sk}/2$ x k_1)
N_2 Convolutional/Residual Layers (k_2 features)
($N_{\rm batch}$xN$_{\rm sk}/4$ x k_2)
$N_{\rm n}$ Convolutional/Residual Layers ($k_{\rm n}$ features)
($N_{\rm batch}$xN$_{\rm sk}/n$ x $k_{\rm n}$)
Flatten
Dense Layer (N$_{\rm sk}$ x 2 nodes)
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
($N_{\rm batch}$x$N_{\rm sk}$x2)
($N_{\rm batch}$x$N_{\rm sk}$x2)

Figure 2. The general schematic of our ConvNet or ResNet. Each stage is a stack of a certain number, N_i, of convolutional or residual layers with a fixed number of features, k_i. Each is processed with maximum pooling of two pixels. The exact number of stages, layers at each stage, and number of features are determined using hyperparameter tuning. Here, $N_{\rm sk}$ is the number of pixels in a skewer and $N_{\rm batch}$ is number of skewers taken for each batch referred to as batch size. The output format at each stage is shown on the left. The stage for each network is dense layer.

be added when both can have a different number of features. Therefore, we process it with a simple convolutional layer to extract the same number of features before adding it to the output. This strategy of skipping layers has two advantages. If adding more layers is useful for the network, the gradients (calculated for error back-propagation) would be non-vanishing in deeper layers. If the additional layer has a negligible effect, the network can simply skip over them, which saves time during training and does not affect network performance.

The first layer of the dense layer implements a matrix multiplication between the weight matrix (learned during training) and the input. The dense layer has the same two last layers as the convolutional layer. The size of our dense layer is twice the number of pixels in an Lyα skewer, as the network predicts both the mean and standard deviation for each pixel. The basic difference between the convolutional/residual and dense processing layers is the scale of features impacting the immediate output. The former extracts localized features, while the latter takes advantage of full connectivity between inputs.

We show the general schematic of our two networks, convolutional network (ConvNet) and residual network (ResNet), in Fig. 2. These networks are essentially one-dimensional implementation of ResNet (He et al. 2015) using TENSORFLOW/KERAS (Chollet et al. 2015). The network performs processing in a total of N stages. Each stage is a stack of N_i convolutional or residual layers, where the number of features, k_i, is kept fixed. The maximum pooling of two pixels is performed at the end of each stage in ConvNet and ResNet. This scaling is shown on the left side of the schematic as downsampling of each $N_{\rm sk}$, the number of pixels in a skewer. We increase the total number of convolutional/residual layers and the features in each stage as we go deeper into the network. This is to tackle the increasingly complex features at later stages. This progressive act of pooling/convolution and increasing features at each stage transforms data from sample to feature space. Both networks appropriately format (flatten layer) the outputs and feed them to a dense layer as a final processing stage. This layer outputs the mean and sigma of $\log \Delta_\tau$ or $\log T_\tau$ for each pixel of a given skewer, i.e. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Note that the dense layer is similar to a conventional weight matrix multiplication but twice the number of weights. The output now represents distribution rather than the actual predictions.

3.3 Training a network

Now, we focus our attention to details involving training a ConvNet or ResNet. In Section 3.4, we will discuss how hyperparameters are tuned to find an optimal network architecture at $z = 4$ through 5.

We do an 80–20–20 per cent split of the original data set to separate the training, validation, and test skewers. We have also done trials with 70–30 and 50–50 splits for train and validation. All cases show very similar training histories but standard split shows best convergence. It is ensured that the splits have proportionally the same number of skewers from each simulation run. Before training, splits are standardized. This is achieved by subtracting the mean value and dividing it by the standard deviation of each quantity (i.e. Lyα flux, $\log \Delta_\tau$, and $\log T_\tau$) of the training split. We use the same values to transform to or back when needed. This is a standard practice and helps to improve speed, stability, and convergence during training.

At the training/validation steps, we periodically shift the skewers to random locations taking into account the periodic boundary conditions of the simulation. This is equivalent to changing the starting position of a skewer along one axis. Later, we add Gaussian-distributed noise with the fixed signal-to-noise pixel. The noise realization is generated during the training and validation steps. We do this procedure to overcome any overfitting problems, which arise due to the limited number of skewers extracted along the box axes or because of a fixed noise realization.

The training and validation are performed in batches, the number of which is a hyperparameter. Once the network has seen all the examples, it is known as an epoch. The skewers in the entire data set are randomized at the start of each epoch. The metric is calculated first for training and later for validation split over all batches, with one value for each batch. The loss at each epoch is then given by a simple sum of the loss from all batches. This way we obtain training and validation losses for each epoch. We also add sum of squares of learnable weights with factor of 10^{-4} to loss function for regularization purposes. During training, we half the current learning rate if the metric for validation split does not improve for 10 consecutive epochs. We run training for 150 epochs, which is enough to reach convergence. Typically, the metric for validation split asymptotes before reaching 100 epochs. All training/validation sessions were run on four A100 Nvidia Graphics Processing Unit (GPU) nodes with a typical completion time of under half an hour.

3.4 Finding optimal networks

The main hyperparameters we considered to search for our optimal networks are the type of architecture (ConvNet/ResNet), as well as the total number of stages, layers, and features at each stage. We impose a few restrictions in this grid search to avoid the network from becoming overly complex. We limit the total stages between 1 and 6. The number of layers at the first stage is either 1 or 2 for ResNet and 2–4 for ConvNet. It is then kept the same or doubled in subsequent stages, but never allowed to increase above 4 for ResNet and 8 for ConvNet. The reason for different criteria is to ensure that the largest possible network has the same complexity. The number of features at the first block is chosen from [2, 4, 8, 16, 32]. It is either kept the same or doubled for each subsequent stage.

In addition, there are also hyperparameters related to the training that are needed to be optimized. These are $N_{\rm batch}$ and the learning rate,

Table 2. The hyperparameters related to our networks used in this work at $z = 4.0$ through 5.0. The columns represent layers at each stage, features at each stage (progressively increasing in power of two), learning rate ($l_{\rm r}$), and batch size ($N_{\rm batch}$). All networks are ResNet in architecture.

z	Layers	Features	$l_{\rm r}$	$N_{\rm batch}$
		log $\boldsymbol{\Delta_\tau}$		
4.0	4, 4, 4, 4, 4	32, 32, 64, 128, 256	2.0×10^{-3}	1024
4.4	3, 4, 4, 4, 4	16, 32, 32, 64, 128	6.1×10^{-3}	1024
5.0	2, 4, 4, 4	16, 32, 64, 128	1.5×10^{-3}	1024
		log $\boldsymbol{T_\tau}$		
4.0	4, 4, 4, 4, 4, 4	8, 16, 32, 32, 32, 64	8.3×10^{-3}	1024
4.4	3, 3, 3, 4, 4, 4	16, 32, 32, 64, 128, 256	1.0×10^{-2}	2048
5.0	3, 3, 3, 4, 4, 4	32, 32, 32, 32, 64, 128	1.5×10^{-2}	4096

$l_{\rm r}$. The weights and biases of the network are updated at the end of every batch. A small batch typically results in noisy gradient estimation (calculated during error propagation), and therefore translates into fluctuations in parametric space. A large batch gives an averaged gradient and the metric takes longer to reach optimal values. The gradient of weights and biases are adjusted through $l_{\rm r}$. A higher rate may introduce noise and stop the network from converging. On the other hand, smaller rates might make the network take way too long to reach convergence. We sampled $l_{\rm r}$ in log space across 10^{-4}–0.5. This is an initial learning rate; however, we half its current value when the metric does not improve after 10 epochs. The batch size range in powers of two is 1024–8192. The large batch size is chosen to run the training on four Nvidia A100 GPUs in parallel. Usually, there is a correlation between $N_{\rm batch}$ and $l_{\rm r}$; therefore, both parameters need to be tuned together.

We use OPTUNA,[2] a PYTHON-based Application Programming Interface (API) to search for optimal hyperparameters. We use the default tree-structured Parzen estimator provided by OPTUNA to sample the parameter space. We ran 100 training and validation trials for log T_τ and log Δ_τ separately at each redshift. We are expecting each quantity to have a network with different complexity. For each trial, we assemble a network with hyperparameters suggested by OPTUNA. We train the network (procedure outlined in Section 3.3) and minimize our metric. We train for 100 epochs (50 epochs less than actual training) and keep the minimum value attained for –log $\mathcal{L}$ for the validation split at the end of each trial. The metric and optimizer (i.e. Adam) are same for hyperparameters grid search and training. The code used for data preparation training and grid search is also available online at repository.[3]

We summarized the optimal set of hyperparameters of this grid search in Table 2. It is clear that a much smaller network extracting fewer features is preferable for log Δ_τ as compared to log T_τ. ResNet is the preferred network for all cases. Note that the ResNet has twice the complexity as compared to ConvNet with same number of stages, layers, and features. We have found that at higher redshift the network performance degrades that can be primarily attributed to low transmitted Lyα flux. Although there can be some impact on network architecture and (or) hyperparameters with the noise assumed for the mock spectra, we chose S/N = 50 pixel^{-1} as fiducial value.

3.5 Predictions

For our results, we only utilized predictions from test data sets from each model (1000 sightlines per model). The network outputs mean and standard deviations at each pixel of the Lyα skewer for log Δ_τ and log T_τ. We have shown the residual distributions for each simulation in Fig 3. This plot highlights any bias and skewness of the predicted distributions among models with different thermal parameters.

By sampling these distributions, we first obtain an initial estimate of T_0 and γ. However, using uncorrelated single pixel Gaussian to reproduce realizations gives us unconstrained estimate and unrealistically small confidence intervals. Therefore, we need to estimate a true error model that we could sample to measure T_0 and γ and build their respective confidence intervals over many realizations.

We first obtain residuals of concatenated log Δ_τ and log T_τ, as $(Y_{\rm con} - \boldsymbol{\mu}_{\rm con})/\boldsymbol{\sigma}_{\rm con}$, where $Y_{\rm con}$ is the actual quantity, and $\boldsymbol{\mu}_{\rm con}$ and $\boldsymbol{\sigma}_{\rm con}$ are the predictions for the mean and standard deviation, respectively. Later, we congregate 40 sets of randomly shifted residuals to make the matrix smooth. This amounts to 40 000 skewers in total. Finally, we obtain the residual correlation matrix, $\sum$. We repeat this process for each model and at each redshift. In Fig. 4, we have shown the correlation matrix for FIDUCIAL model at $z = 4$. It is evident that there is a significant correlation between residual pixels for log T_τ (bottom-right panel) at small scale. The log T_τ and log Δ_τ residuals are weakly cross-correlated. Finally, we generate joint realizations of log Δ_τ–log T_τ skewers by simply sampling the multivariate Gaussian, $\boldsymbol{\mu}_{\rm con} + \mathcal{N}(0, \sum)\boldsymbol{\sigma}_{\rm con}$.

To determine which correlation matrix to use for a given sightline, we use our initial estimates of uncorrelated T_0–γ distributions. We use the correlation matrix of the model with the least Euclidean distance in T_0–γ plane using our initial estimates. However, we found no noticeable differences even when we obtain realizations using only the FIDUCIAL correlation matrix. In practice, the confidence intervals obtained from a large number of realizations using this procedure and the ones directly from network predictions are extremely similar. However, individual realizations of a given skewer can differ significantly. We obtain 1000 realizations for each skewer.

To estimate T_0 and γ on log Δ_τ–log T_τ distribution (for an actual or predicted realization), we make a small modification to our previous method. We employ an additional step of removing values that correspond to saturated pixels in the flux before fitting a line through the median log T_τ points in the desired log Δ_τ bins. We defined saturated pixels as when the flux is less than the 1σ noise level. For each sightline, we use all 1000 log Δ_τ–log T_τ distribution realizations (not same as skewers) and estimate T_0–γ distribution. This provides us with joint T_0–γ distributions for any given 20 h^{-1} cMpc skewer.

Thelog Δ_τ also shows a slight bias towards high densities; however, for log T_τ it is most noticeable with different thermal histories. For instance, the distributions for HOT and COLD are biased low and high, respectively. The same is true for models with different redshift of reionization. G10 shows a significant tail, which can be attributed to relatively flat and insensitive log T_τ along the Lyα flux skewer. Secondly, we are limited by training examples, as most of them are at $\gamma \simeq 1.2$–1.5. Although the predicted distributions for individual simulations can be biased, their covering fraction, $\sigma_{\rm cov}$, still remains above the expected 68 per cent.

3.6 Lyα flux through the network

In this section, we will examine the flow of the Lyα flux through different layers of a simplistic network to build intuition into the reconstruction process. In order to make the outputs easier to

[2] https://optuna.readthedocs.io/en/stable/
[3] https://github.com/nicenustian/bh2igm

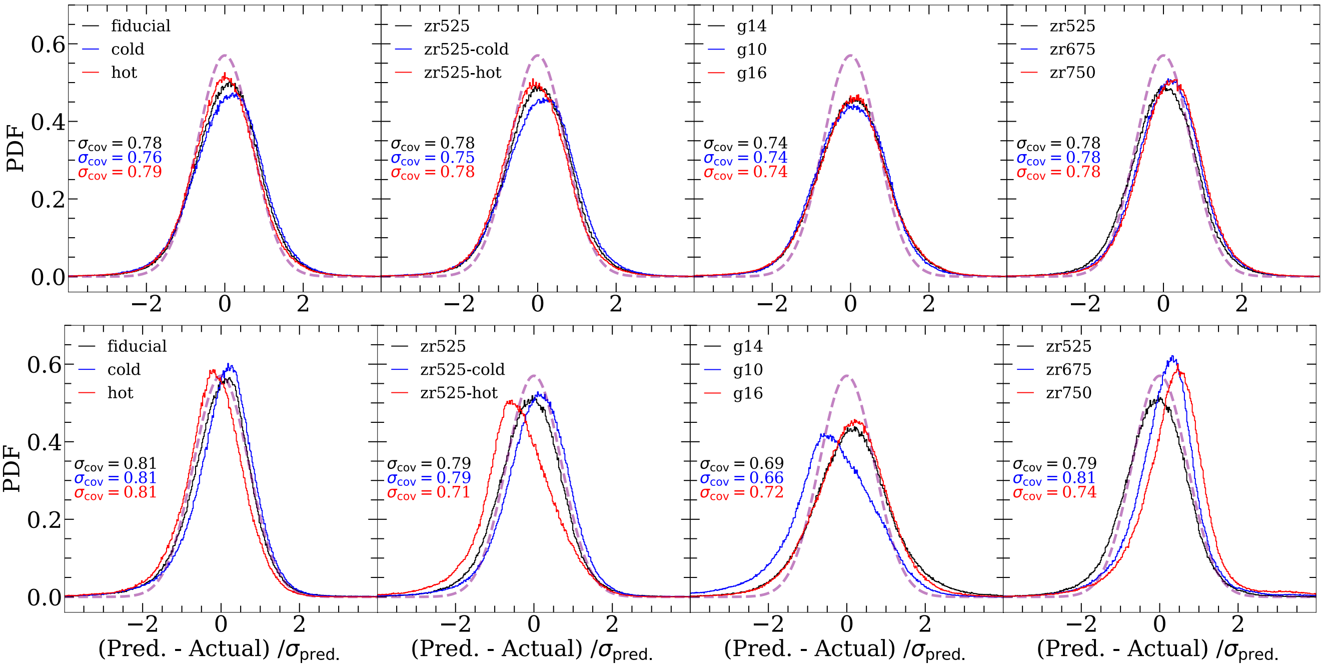

Figure 3. The residual distributions of predicted $\log \Delta_\tau$ (top row) and $\log T_\tau$ (bottom row) along with the respective $\sigma_{\rm cov}$ for the simulations shown in the legends. The dashed purple curve shows a Gaussian distribution with zero mean and unit variance to highlight any skewness or bias.

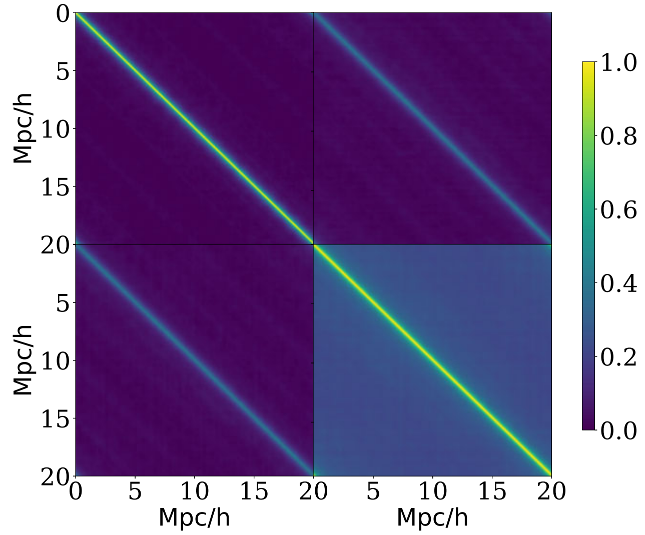

Figure 4. The residual correlation matrix of concatenated $\log \Delta_\tau$ and $\log T_\tau$ for FIDUCIAL model at $z = 4$. To determine a well-behaved matrix, 40 randomly shifted skewers from the model were stacked together. This amounts to 40 000 skewers. The top-left panel represents $\log \Delta_\tau$, while the bottom-right panel represents residual of $\log T_\tau$. The rest show the cross matrix between $\log \Delta_\tau$ and $\log T_\tau$.

visualize at each stage, we utilize an elementary architecture. The network has two convolutional layers, extracting four and eight features, respectively, forming a two-stage ConvNet. We train the network using $N_{\rm batch} = 32$ and $l_{\rm r} = 10^{-4}$. We utilize the same data set as our primary results for training and validation.

Fig. 5 shows the propagation of a normalized Lyα forest skewer at $z = 4$ through different stages. The purpose is to illustrate how the subtle differences in Lyα flux between HOT and COLD models with different T_0 are translated into $\log T_\tau$ predictions.

The first convolution layer extracts four features directly from the normalized flux. The output is extracted by convolving the normalized flux using 3 pixel-wide kernels, which emphasizes the sharp features in the Lyα flux of the COLD model. Notice that the output pixels can be below zero, which is only possible with PReLU activation. Allowing the neurons to fire even when the output is negative is crucial to fully utilize the dynamic range of normalized Lyα flux pixels. At the second stage of convolution (third row), the trend is even more pronounced. Notice that each output skewer at the second stage is evaluated by combining all the feature skewers of the previous stage (in this case, four) through one convolutional kernel.

The fourth panel shows the output at the dense layer. The feature skewers are finally transformed into predictions, which are distributions for each pixel. The network only outputs the parameters of the distributions that are the mean, $\boldsymbol{\mu}$ (dashed curves), and standard deviation, $\boldsymbol{\sigma}$. The predicted distributions are transformed back into their original units by using the mean and standard deviations from the training split. We obtain 1σ confidence intervals shown as light shaded regions.

The higher density regions show relatively less Lyα transmission that translates into higher uncertainty in $\log T_\tau$ or vice versa. The actual $\log T_\tau$ (solid line) falls mostly within the predicted confidence intervals (light grey region). It is expected that the actual quantity should fall within the predicted light shaded contours at least 68 per cent of the time for the entire data set. It is evident that the network sometimes fails to predict the right confidence intervals, specifically for saturated pixels. This gives rise to the modest tails in the residual distributions.

4 RESULTS

In this section, we will discuss in detail the predictions, primarily focused at $z = 4$ and with $\mathrm{S/N} = 50$ pixel^{-1} for noise (Sections 4.2–4.4). Our main goal is to recover the IGM $\log \Delta_\tau$ and $\log T_\tau$ along the sightlines for our models with varying thermal parameters. This

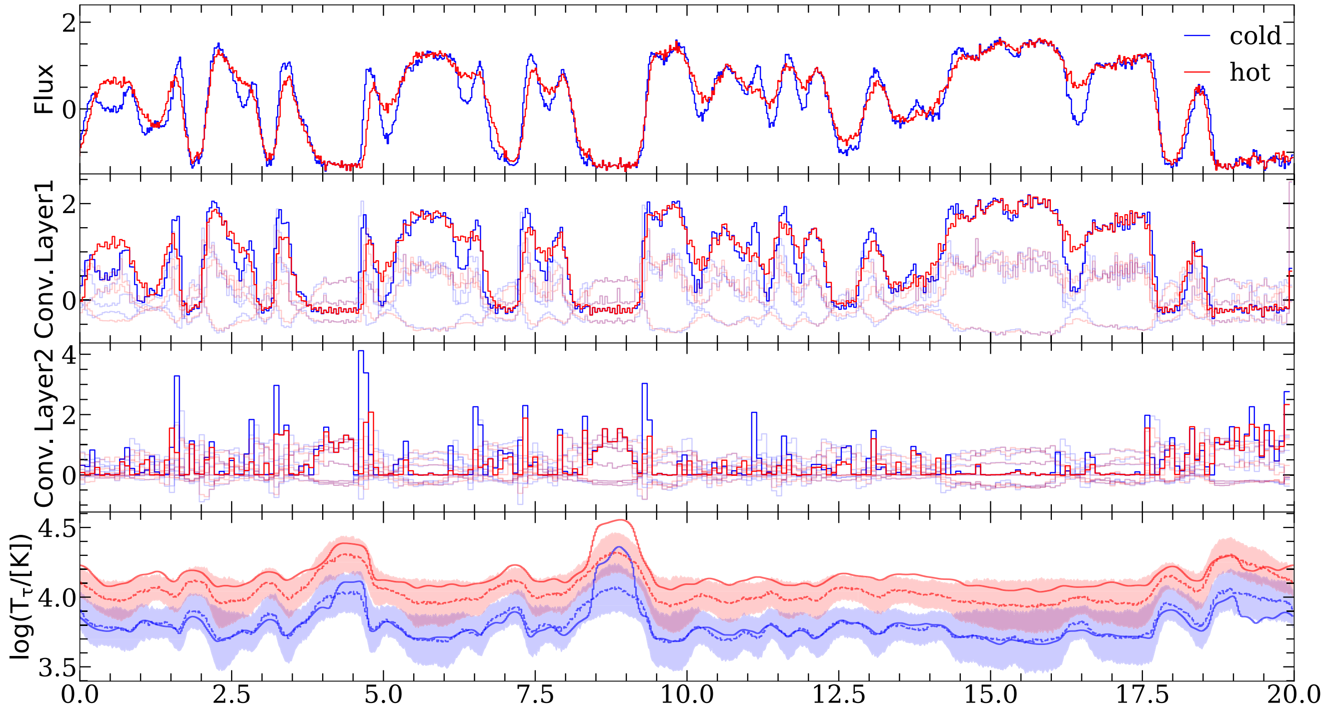

Figure 5. The flow of the example skewers at $z = 4$ from HOT and COLD models through the various layers of trained ConvNet discussed in Section 3.6. The first row is the normalized Lyα forest skewer input to network. The second and third rows show the four and eight features, respectively. The second convolutional layer extracts each feature by combining all the skewers from first layer. We highlight the most prominent feature skewers for clarity. The output curves at the dense layer (fourth row) show predicted mean (dashed) and 1σ confidence intervals as shaded regions, at each pixel of input skewers. The solid curves show the actual $\log T_\tau$.

ultimately leads us to constrain the thermal parameters (i.e. T_0 and γ). We also extend our analysis for spectra treated with different S/N (Section 4.4). Later, we will extend the results up to redshift $z = 5.0$ in Section 4.5. In the final section (Section 4.6), we will show predictions for a segment taken from an observational spectrum and establish that the method can provide reasonable but powerful constraints on thermal parameters.

4.1 Predictions along the sightlines

Figs 6 and 7 show the predictions of example skewers from simulations highlighting the impact of varying thermal parameters at $z = 4$ with S/N = 50 pixel^{-1}. It is evident that the actual quantities (solid curves) lie mostly within the predicted 1σ along the skewers.

The 1σ intervals are smaller in regions with significant Lyα flux transmission and largest in the saturated parts. The network is unable to predict the right confidence intervals for saturated regions (for example, at 5 h^{-1} cMpc in the first panel). This shows that the network is not overfitting the training data set. This essentially limits the predicting power primarily to underdense IGM gas, which is not saturated at $z = 4$. The prediction for $\log \Delta_\tau$ has very narrow confidence intervals as compared to $\log T_\tau$. This is mainly because $\log \Delta_\tau$ reconstruction is very localized and is impacted by very local features extracted from Lyα forest. The prediction of $\log T_\tau$ at a given pixel depends on the Lyα pixels on several scales. The small uncertainties on $\log \Delta_\tau$ predictions make it potentially a method to constrain cosmological models that can impact the IGM densities on smaller scales such as warm dark matter (Iršič et al. 2017, 2024; Villasenor et al. 2023).

Varying T_0 impacts the small-scale IGM densities due to pressure smoothing (Hui & Gnedin 1997; Peeples et al. 2010; Nasir, Bolton & Becker 2016). It is clear that the predictions also capture faithfully $\log \Delta_\tau$ along the sightlines (see top panel of Fig. 6). The differences in $\log \Delta_\tau$ skewers are partly due to the difference in the Lyα Doppler broadening between models varying T_0. The COLD has noticeably more structure as compared to HOT. The impact is subtle but smaller uncertainties help to reliably capture this in $\log \Delta_\tau$ predictions. The models with different T_0 also have slightly different slope of T_0–γ relation where COLD (HOT) is steeper (shallower) (see Section 4.4). The predicted pixel distributions for $\log T_\tau$ along the skewers remarkably trace the underlying temperatures with the right confidence intervals (except for saturated regions). The predicted T_0 values for the example skewers are predicted within a few hundred Kelvins of their actual values with 1σ confidence intervals of $\delta T_0 \lesssim 1000$ K for most cases.

The predicted $\log T_\tau$ for models with varying γ (second panel, bottom row) can also capture the trend of actual $\log T_\tau$ within predicted confidence intervals. The Lyα flux for G10 by and large is insensitive to variations in $\log T_\tau$. However, the actual $\log T_\tau$ still lies within narrow confidence intervals. The $z_{\rm re}$ parameter has a very fine imprint on the flux that simply translates into larger uncertainties on $\log T_\tau$ predictions. However, $\log \Delta_\tau$ is well constrained with similar uncertainties as compared to rest of the models. The subtle differences among these models are essentially due to Jeans smoothing in the gas.

Lastly, we have shown our test models NYX-LATE and NYX-EARLY in the bottom panel of Fig. 7. There is one point worth reiterating that these models were run with entirely different hydrodynamical code. The predictions are obtained with exactly the same method with frozen network weights. Qualitatively, the predictions are similar to those from the Sherwood runs. However, one difference is predicted that $\log T_\tau$ does not strongly correlate with $\log \Delta_\tau$, which results

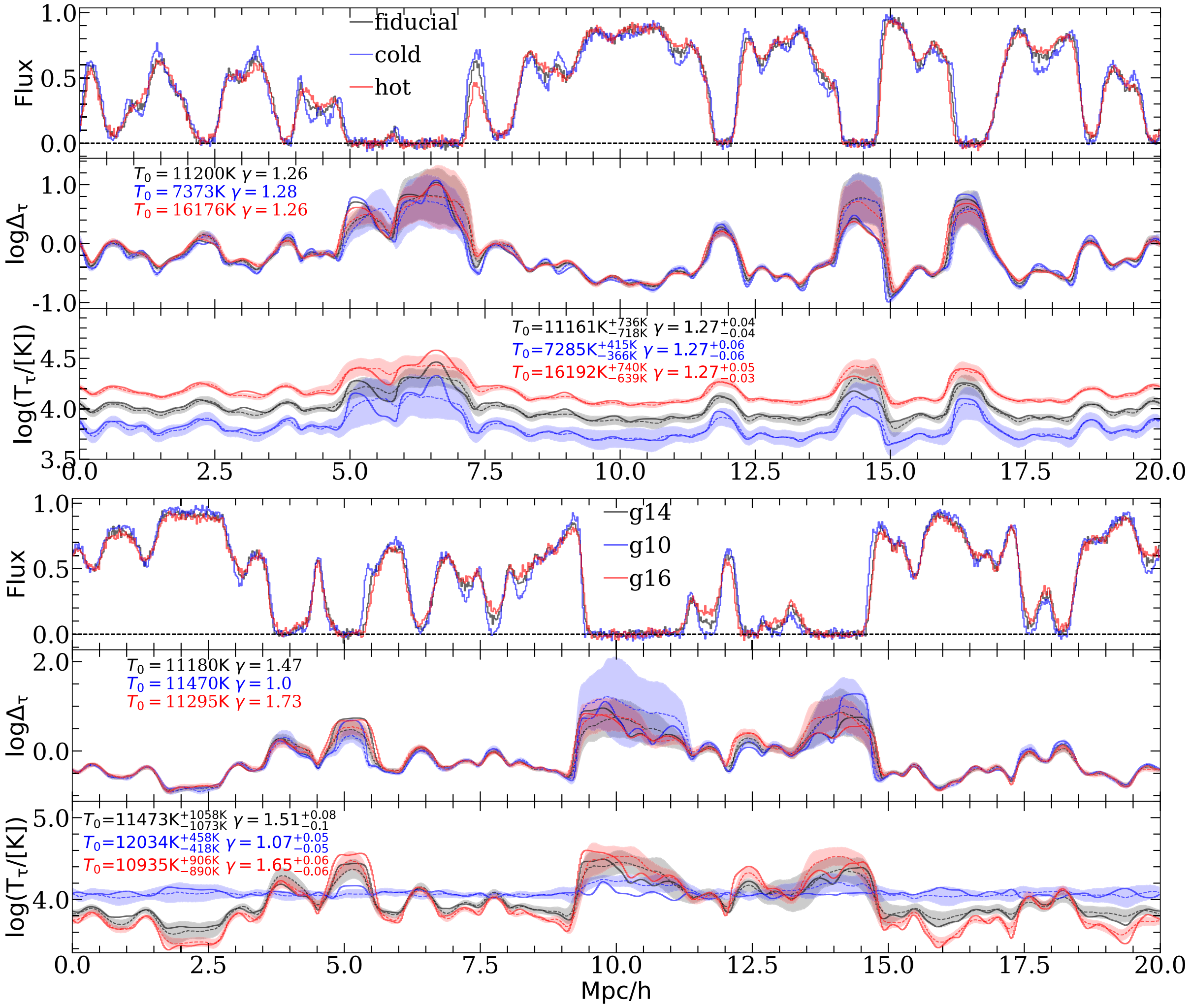

Figure 6. The predicted $\log\Delta_\tau$ and $\log T_\tau$ along example skewers for the simulations varying T_0 (top panel) and γ (bottom panel) at $z = 4$ with S/N = 50 pixel^{-1}. Actual quantities are shown as solid curves overlaid with predicted mean (dashed curves) along with 1σ confidence intervals as shaded regions. The actual T_0 and γ are shown in second row of every panel, while estimated in every third row along with 1σ confidence interval.

in systematically underestimated γ value. We will later discuss this issue in Section 4.3.

4.2 $\log\Delta_\tau$ and $\log T_\tau$ distributions

To examine the range of IGM conditions probed by our reconstruction method, we examine the actual (solid curves) and predicted (dashed curves) probability density functions (PDFs) in Fig. 8. We have overlaid the scatter in the distributions over 20 h^{-1} cMpc skewers as the 1σ shaded region in Fig. 8. Overall, there is an excellent agreement and no noticeable bias between the predicted and actual distributions. Despite some expected scatter among the 20 h^{-1} cMpc skewers, the distributions can capture the broad trends in densities and temperatures probed by the Lyα forest at $z = 4$. It is worth mentioning that the step-like feature at the high $\log\Delta_\tau$ and high $\log T_\tau$ is statistical, simply due to lack of pixels.

The $\log\Delta_\tau$ distributions are very similar for all models and are in agreement (see top row of Fig. 8). The sharp drop in the distribution at mean cosmic density, $\log\Delta_\tau \gtrsim 0$, suggests that forest is mostly sensitive to underdense gas at $z = 4$. The exact densities can slightly differ based on the simulation thermal parameters and history. This can be seen in models with different γ (second last column) showing subtle differences at the high-density tail end. The actual distributions show a small excess at $\log\Delta_\tau \simeq 0.2$. This is primarily due to Lyα flux saturation and consequently losing sensitivity at high densities, which ultimately degrades the reconstruction accuracy. The median of the predicted distributions ranges from $\log\Delta_\tau = -0.33$ to -0.37, very closely following the actual range from -0.32 to -0.38.

Broadly, the predicted $\log T_\tau$ distributions agree very well with a few exceptions. The G10 (third panel) is slightly narrow and exhibits a subtle offset. The ZR525 is almost indiscernible from ZR750. The NYX-LATE model shows a bimodal distribution at the high $\log T_\tau$ end. As the shaded regions show the variations you would expect from a 20 h^{-1} cMpc sightline, it is evident that we can reliably estimate the IGM conditions with one sightline. We can expect a systematic bias that depends on the thermal parameters of model. However, this bias is typically small as compared to predicted confidence interval. It is worth noting that the predicted $\log T_\tau$ distributions are generally broader than the truth, but this is expected given the predicted scatter.

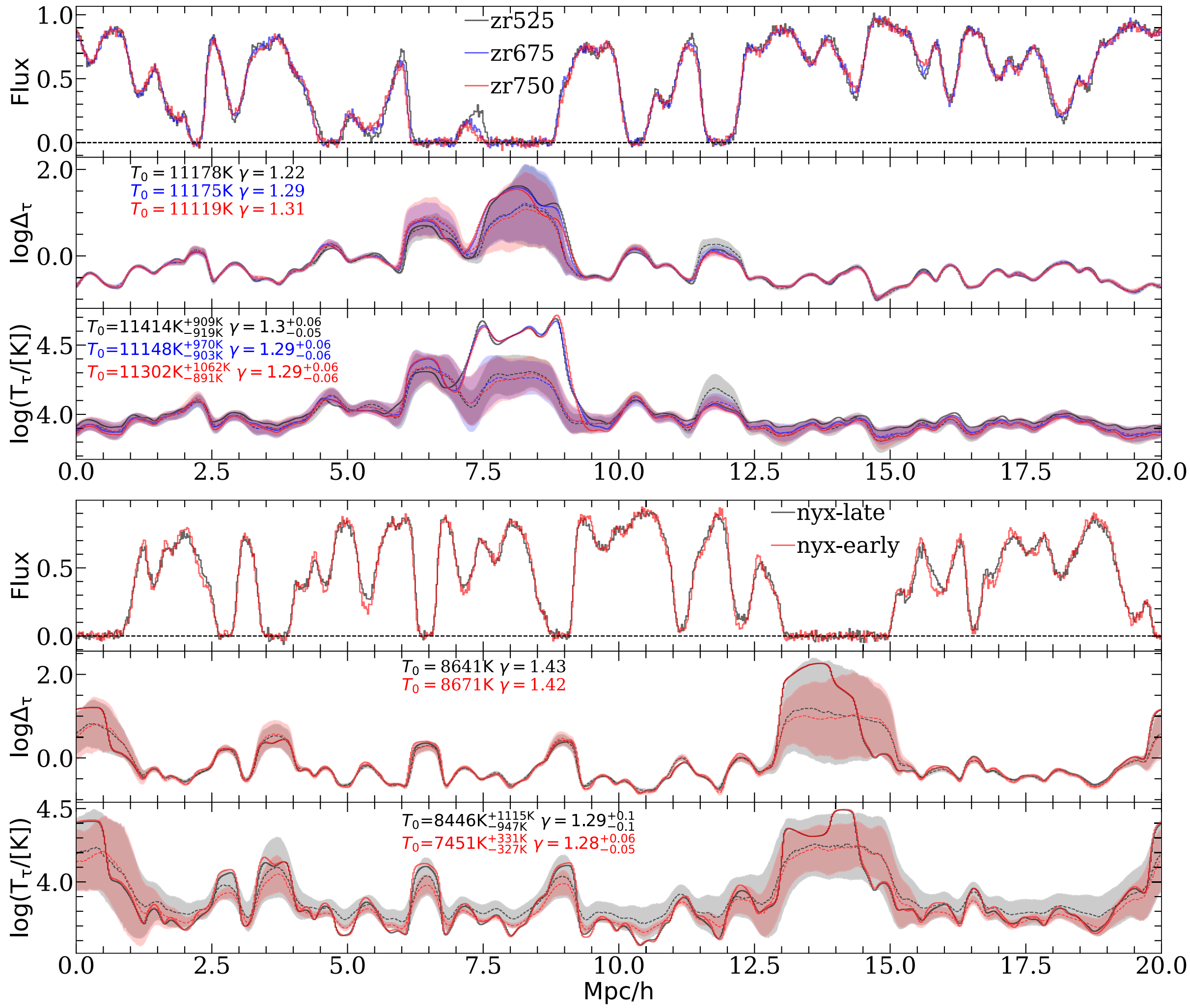

Figure 7. Same as Fig. 6 but now for runs varying redshift of reionization (top panel) and test models (bottom panel).

4.3 log T_τ–log Δ_τ plane

The characterization of IGM by thermal parameters (i.e. T_0 and γ) although useful is an oversimplification. It is a fit to the complex two-dimensional (2D) distribution of temperature and density. It is highly non-trivial to recover the entire log Δ_τ–log T_τ distribution and typically only thermal parameters are provided as a statistical insight. However, owing to our reconstruction method, we can show the log Δ_τ–log T_τ plane and examine the detailed 2D distributions for models with varying thermal parameters.

In Fig. 9, we have shown the predicted log Δ_τ–log T_τ distributions, overlaid with 68th and 95th percentiles for predicted (broad) and actual (narrow) as contours. The median log T_τ on log Δ_τ bins (bin size of 0.1) are shown as dashed curves. The estimated T_0 and γ are also shown and appropriately coloured. By comparing the 2σ contours, it is evident that the predicted distributions are broader than the actual distributions. A typical T_0 is within few hundred Kelvins of the actual value, indicating that we can reliably estimate thermal parameters across various models.

The median log T_τ shown as dashed curves broadly follows the trend but shows some noticeable deviations at low (log $\Delta_\tau \lesssim -0.4$) and high densities (log $\Delta_\tau \gtrsim 0.4$). Therefore, in order to have minimal biases on the thermal parameter estimates, we fit a line through bins ranging from log $\Delta_\tau = -0.4$ to 0.2 after removing the saturated pixels. The distributions follow the trend we expect from the temperature–density relation in simulations evolved with non-equilibrium codes such as Sherwood-Relics (see Puchwein et al. 2015). In general, we can reconstruct the deviations from simple power law at lower densities.

The test simulations (NYX-LATE and NYX-EARLY) shown in last column exhibit a very narrow and rather steeper distribution. Recall that these simulations were evolved assuming photoionization equilibrium and therefore do not show any deviations from power law at lower densities. Our predictions fail to capture these differences at the lower density end as highlighted by the median log T_τ curves shown in red. Another reason for these deviations is that we do not have any simulation run that is relatively steep, $\gamma \simeq 1.4$, but colder, $T_0 \simeq 8800$ K. Therefore, predictions show a rather lower γ similar to our colder models such as COLD and ZR525-COLD. These results can be improved with a more comprehensive model grid sampling the T_0–γ space.

4.4 T_0–γ distributions

We now proceed to show our main results of T_0–γ distributions for models. Recall that we have 1000 log Δ_τ–log T_τ realizations for

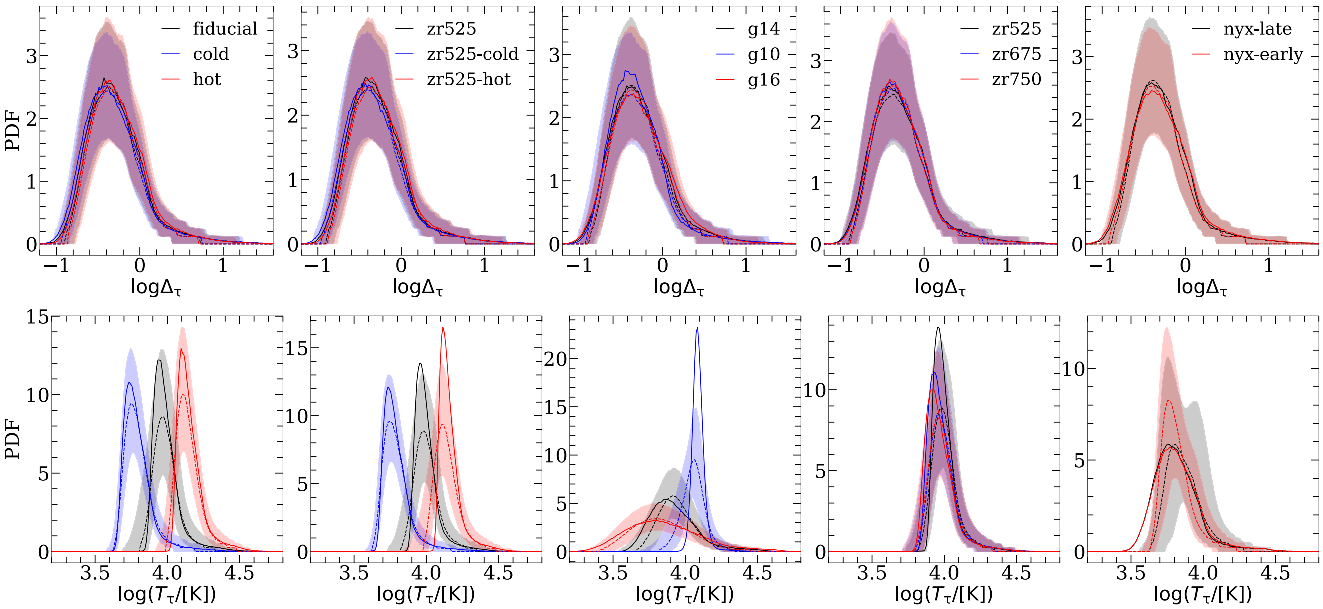

Figure 8. The log Δ_τ (top row) and log T_τ (bottom row) distributions for 20 h^{-1} cMpc skewers. The mean for actual and predicted quantities are shown as solid and dashed curves, respectively. The shaded regions represent the 1σ scatter over realizations (see Section 3.5 for details).

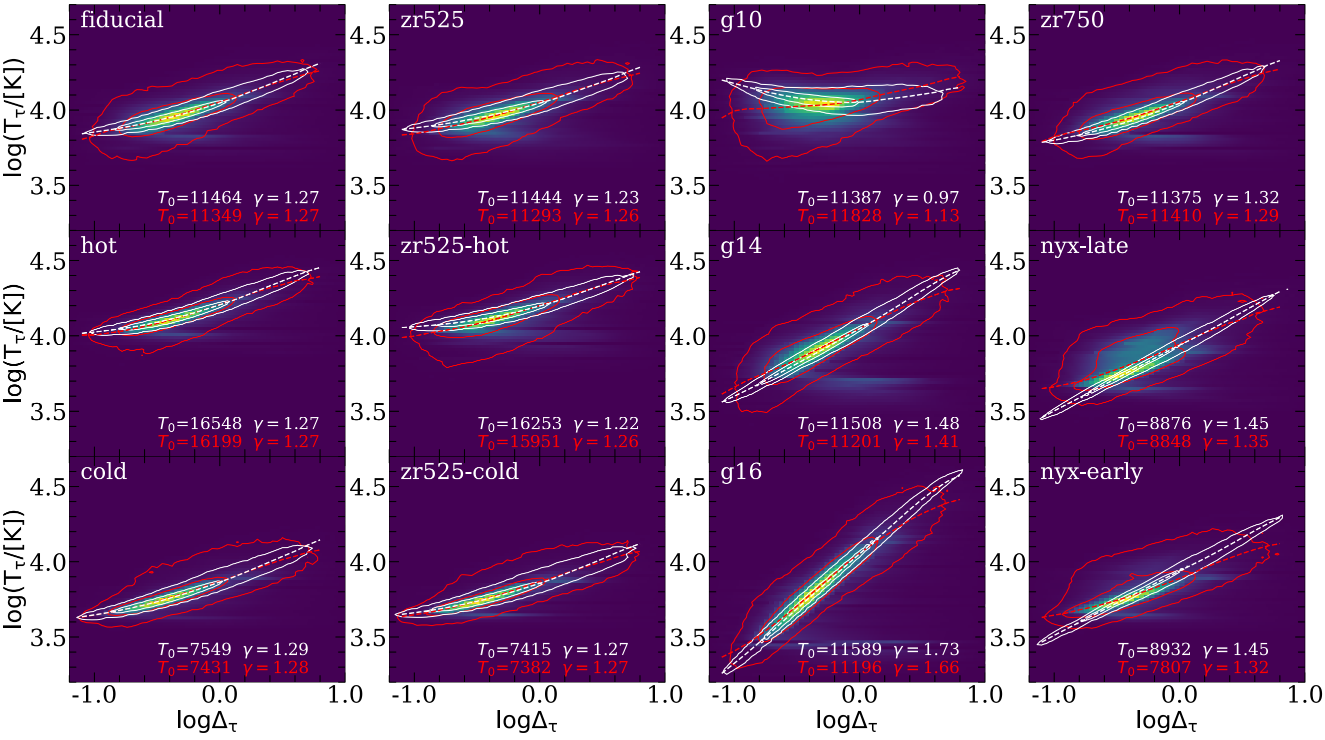

Figure 9. The predicted log Δ_τ–log T_τ distributions for the models shown in legends. The contours encapsulate the central 68th and 95th percentile interval for predicted and actual quantities. The estimated T_0 and γ for the entire skewers are also shown in legends. The dashed curves represent the median log T_τ on log Δ_τ bins.

each 20 h^{-1} cMpc skewer and their estimates for T_0–γ using our test split. In total, we have 40 000 T_0–γ data points for each model. We have shown the T_0–γ distributions (using all data points) using different S/N (first through third rows) in Fig. 10. We add zero-centred Gaussian noise with a desired S/N during training/validation, as we have discussed in Section 3.3. The contours cover 68th and 95th percentiles of data points. The medians of actual and predicted values are shown as cross and plus symbols, respectively. The predicted T_0 and γ along with 1σ confidence intervals for each model are shown in legends.

It is evident that the predicted distributions are in good agreement with actual values (cross symbols) lying mostly within predicted

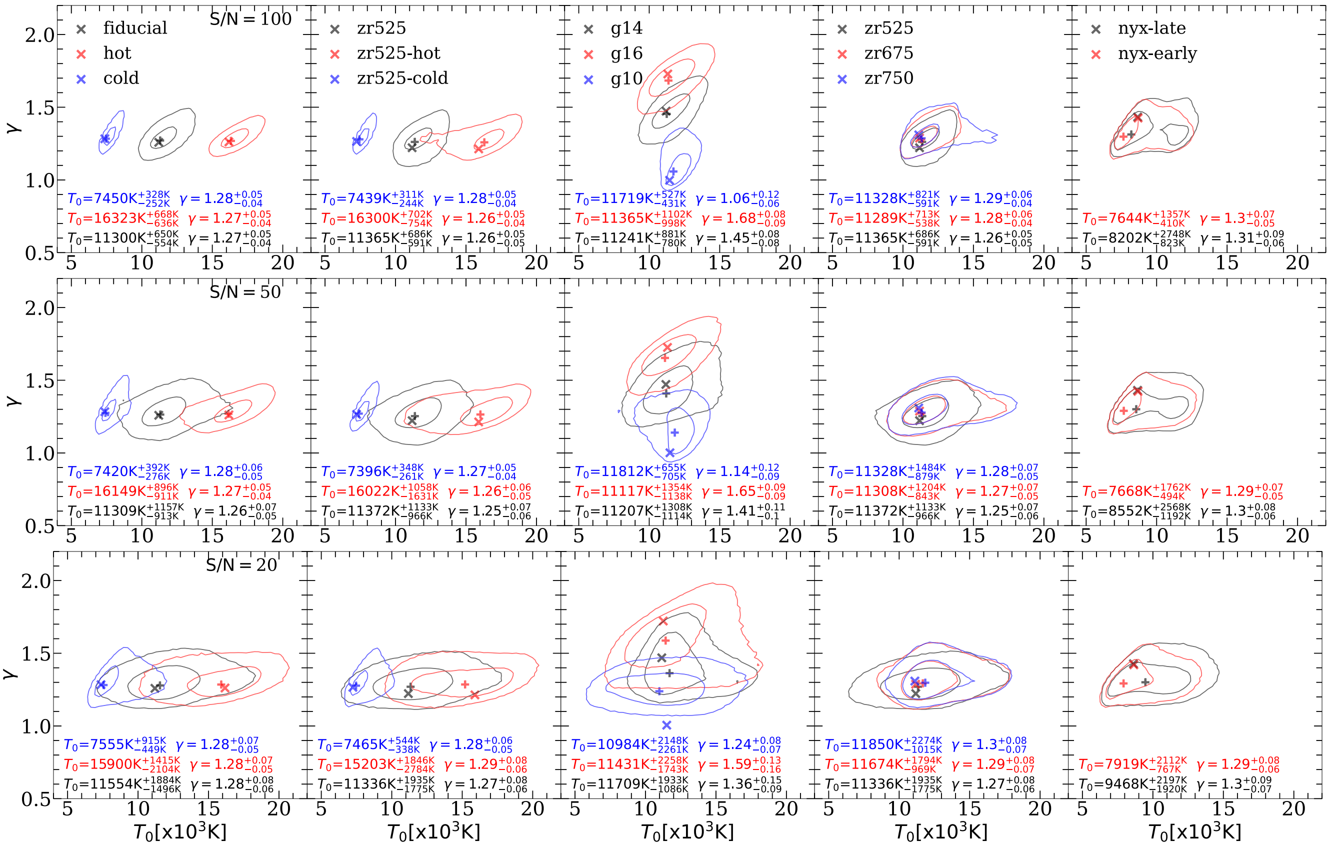

Figure 10. The T_0–γ distributions with different S/N (shown in legends) at $z = 4$. Each row represents sightlines that are post-processed with added Gaussian noise with S/N = 100, 50, and 20 pixel^{-1}, respectively. The distributions are obtained by estimating T_0–γ for each realization of 20 h^{-1} cMpc sightlines (1000 realizations for each skewer). The contours encapsulate the central 68th and 95th percentiles of the points. The estimated T_0 and γ along with their 1σ confidence intervals are shown in the legends. The plus and cross symbols mark the median of actual and predicted values for entire model.

68th percentile. The T_0 predictions are very constraining with typical uncertainties $\delta T_0 \simeq 250$–2500 K. The γ estimate also shows promise with 1σ uncertainties at $\delta\sigma \simeq 0.04$–0.15.

As expected, the constraining power degrades with decreased S/N. We can constrain our typical model with varying T_0 with 2σ confidence interval at S/N = 100 (except ZR525-HOT), while at S/N = 20, it drops to 1σ. The constraints for models varying γ are less stringent. For example, for S/N = 100 case, the G10 is constrained with 2σ, while G14 and G16 with only 1σ confidence. For NYX models, there is a systematic bias, with γ underpredicted by ∼0.1. We have already discussed this issue in Section 4.3.

The uncertainties on thermal parameters also depend on the model. By comparing models for S/N = 50 shown in second row, FIDUCIAL has uncertainties at $\delta T_0 \sim 1000$ K, while COLD has significantly lower $\delta T_0 \sim 400$ K. Same is true for ZR525-COLD and ZR525-HOT. Notice that the uncertainties are very similar for models with different γ (similar T_0) as shown in third column. The subtle differences in γ for different $z_{\rm re}$ models can be seen in fourth column. Relatively late reionization ZR525 tends to have small values for γ as compared to early model ZR750, mostly due to IGM adiabatic cooling. The estimated median γ hints about this evolution, although the uncertainties remain quite large.

So far, we have discussed the recovery of T_0–γ in the context of individual skewers. We can significantly improve these constraints if we consider combining several 20 h^{-1} cMpc segments. For this, we combine their log Δ_τ–log T_τ realizations first and later we estimate the T_0–γ by fitting a line to the binned log Δ_τ–log T_τ as before. To obtain the T_0–γ distributions, we simply draw 10 000 times over any 5 or 10 skewers with repetition and estimate T_0–γ. The resulting distributions for both cases are shown in Fig. 11. The redshift path-length for 5 (10) skewers is $\Delta z \simeq 0.2(0.4)$ with fixed S/N = 50 pixel^{-1}.

As expected, the constraints get tighter with additional skewers, by comparing the T_0–γ distributions shown in Fig. 10 (middle row) (S/N = 50 case) and Fig. 11. For example, for models varying T_0 (first panel), the uncertainties are now reduced up to 50 per cent for $\Delta z \simeq 0.2$ case. They are further reduced by ∼25 per cent for $\Delta z \simeq 0.4$, which we expect with the increase in path-length according to central limit theorem. Furthermore, the increase in path-length does not result in any noticeable bias among models except for G10.

Overall, the T_0–γ constraints provided by our method are more powerful as compared to the Lyα forest flux power spectrum. Using our approach, a *single* high-resolution 20 h^{-1} cMpc segment of the Lyα forest can provide constraints on IGM temperature with uncertainties $\delta T_0 \simeq 1000$ K with a typically thermal history. This method potentially provides below $\delta T_0 \simeq 500$ K constraints with a redshift path of $\Delta z \simeq 0.4$, which is 10 times lower than existing studies. In addition, the thermal parameter recovery for models varying γ is also very encouraging, although a slight bias should be accounted for, in case of extreme γ. A single skewer reconstruct can give below $\delta\gamma \sim 0.1$, which is usually achieved for considerable sized data sets using flux power spectrum studies in the literature.

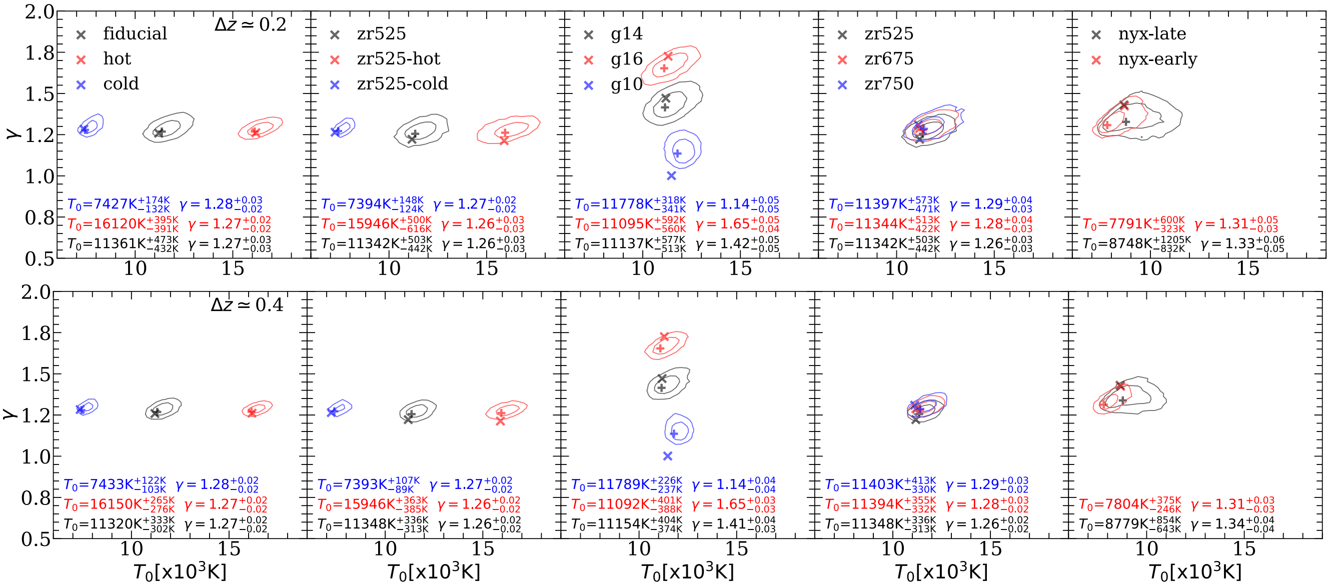

Figure 11. Same as Fig. 10 but now distributions are obtained by reiterating 100 000 times over 5 (top) or 10 (bottom) $\log \Delta_\tau$–$\log T_\tau$ realizations with replacement from different skewers and estimating a joint T_0–γ. This corresponds to total redshift path-lengths of $\Delta z \simeq 0.2$ (top) and $\Delta z \simeq 0.4$ (bottom). The S/N is kept fixed at 50 pixel^{-1}.

4.5 Redshift evolution

Until now, we have shown results at a fixed redshift $z = 4.0$. To pose constraints on the thermal history of IGM, we need to extend this method to higher redshifts. As the Lyα forest flux is the only input to our neural networks, it is likely that the performance could degrade due to significant drop in mean transmitted flux by $z = 5.0$. In this section, we will present our results at $z = 4.4$ and 5.0.

We show the T_0–γ distributions at $z = 4.4$ (top row) and $z = 5.0$ (bottom row) in Fig. 12. It is evident that the drop in mean flux impacts the constraints on T_0 and γ. Overall, there is a trend of rise in uncertainties with higher redshift. All distributions tend to become broader, which is most noticeable for models with varying γ, i.e. G10, G14, and G16. By comparing the 95th percentile between redshifts, most of the distributions become broader in the T_0 direction. The increase in δT_0 can be up to ∼50 per cent. Notice that due to the significantly broader distributions, models with different thermal parameters tend to overlap reducing the constraining power of the predictions using only single 20 h^{-1} cMpc skewer.

4.6 Observational sightline

So far, we have tested our method with mock spectra with different S/N and at different redshifts. In order to put method to practice and determine that any instrumental effects would not compromise our results, we take a 20 h^{-1} cMpc segment from quasar J021043, which is part of SQUAD DR1[4] survey. The details of the reduction can be found in Murphy et al. (2019). The spectrum is observed using VLT-UVES instrument that has resolution of FWHM ∼ 6 km s^{-1} with average S/N = 20 pixel^{-1} over the skewer. The emission redshift of quasar is $z = 4.65$. The spectrum is continua-normalized and has bias regions removed for quasar proximity effect.

To determine the appropriate noise realizations for the mock spectra of this observational spectra, we determine a noise model by using the noise vector from sightline. As the noise is correlated with the transmitted flux level, we calculate median S/N in flux bins with bin size of 0.01. Later, we add zero-centred Gaussian noise with the determined S/N based on the Lyα flux of mock spectra during training/validation. To remove the dependence on periodicity of skewers, we modified our training/validation by masking 16 pixels at left edge after shifting the skewer as before.

The predictions for $\log \Delta_\tau$ (middle) and $\log T_\tau$ (bottom) for the Lyα forest segment (top) overlaid with its noise vector are shown in Fig. 13. The mean (dashed curves) along with 1σ confidence intervals as shaded regions are shown. The segment has a mean flux of $\langle F \rangle = 0.55$ and S/N ≃ 20 pixel^{-1}. The estimates for the thermal parameters are $T_0 = 8270\ \mathrm{K}^{+1467\ \mathrm{K}}_{-1036\ \mathrm{K}}$ and $\gamma = 1.5^{+0.2}_{-0.15}$. The recovered value for T_0 is very similar to our COLD model. We want to reiterate the fact that only single 20 h^{-1} cMpc skewer realizations are used for estimating any T_0–γ distribution. One skewer corresponds to a redshift path-length of $\Delta z \simeq 0.04$ at $z = 4$. For reference, the existing measurements at $4 \le z \le 5$ utilized redshift path-length of $\Delta z \simeq 4$ or 6, typically using 15 or more high-resolution quasar spectra (Boera et al. 2019; Walther et al. 2019). The measurements we have obtained from this example are also consistent with earlier measurements using summary statistics of the Lyα forest (Becker et al. 2011; Boera et al. 2019; Walther et al. 2019b). However, we do not intend to present this result as a measurement but rather a way to validate the method. In future studies, we plan to apply this method to a more comprehensive quasar data set at $z = 4$–5 to measure the IGM thermal history in unprecedented detail.

5 CONCLUSIONS

We have established that reconstruction of IGM gas conditions using neural networks offers significant advantage over traditional summary statistics. The method helps us to transform the Lyα transmitted flux directly to (optical depth-weighted) gas densities and temperatures. This pixel-by-pixel reconstruction enables the mapping of the entire $\log T_\tau$–$\log \Delta_\tau$ plane. This is not possible

[4] https://github.com/MTMurphy77/UVES_SQUAD_DR1

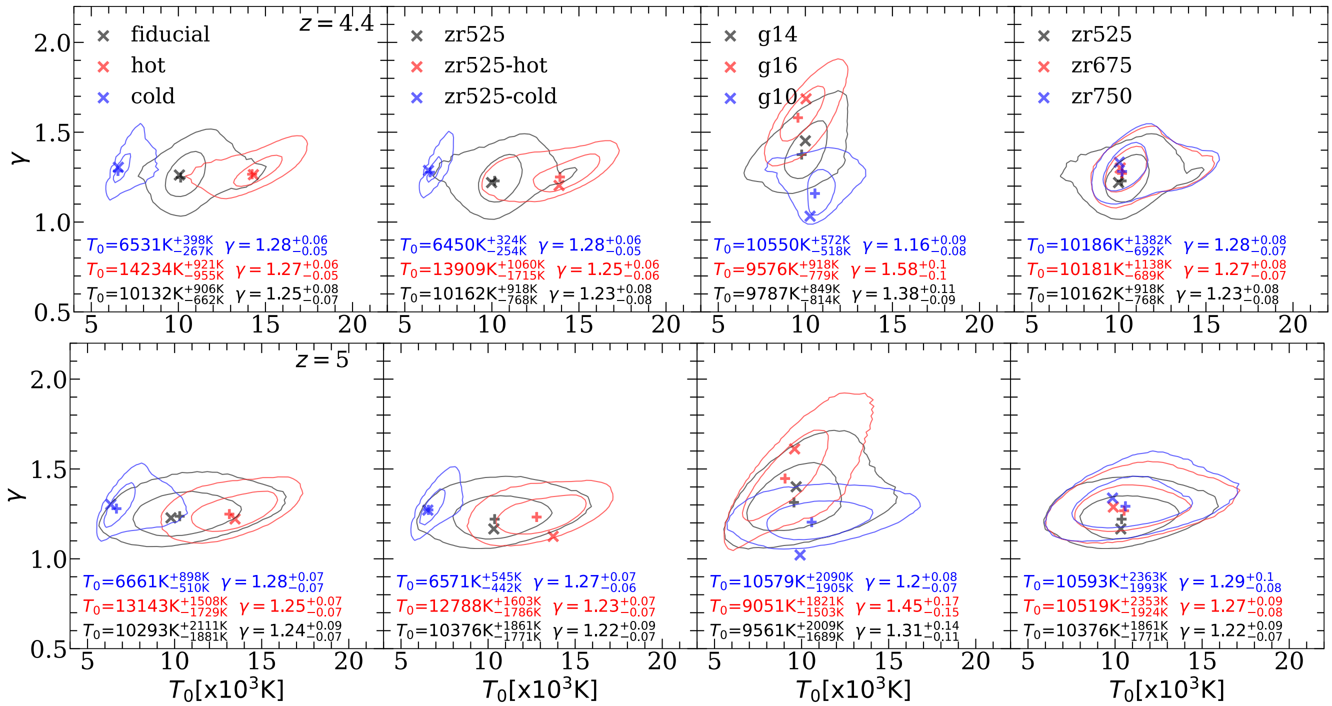

Figure 12. Same as Fig. 10 but at $z = 4.4$ (top), and $z = 5.0$ (bottom) with S/N = 50 pixel^{-1}.

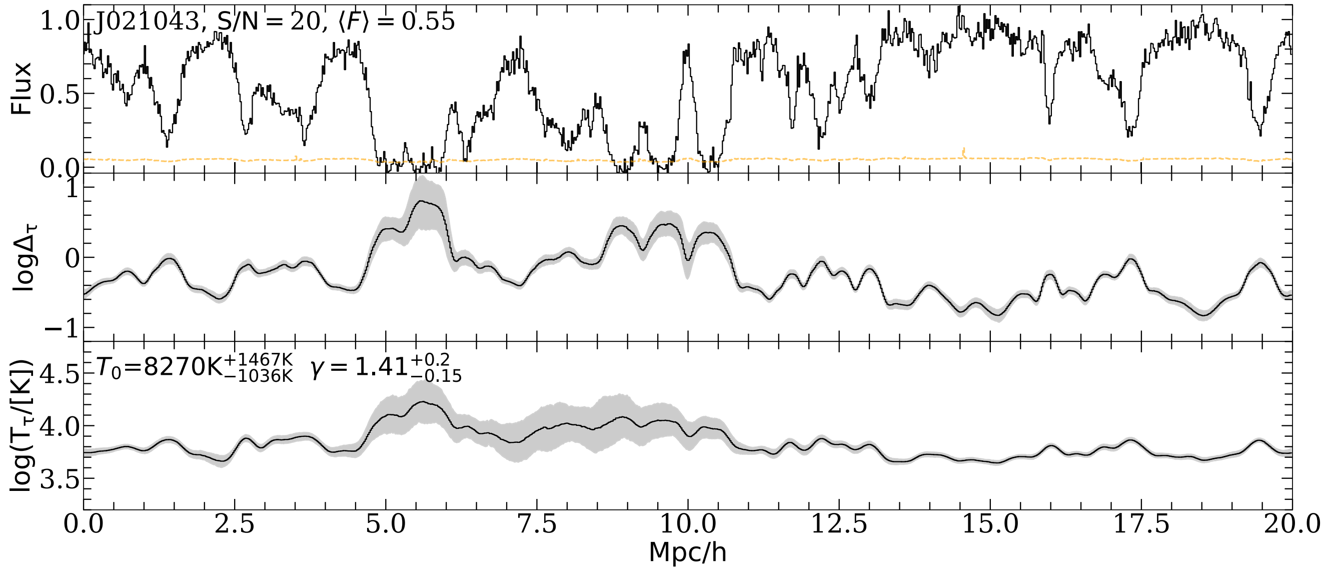

Figure 13. The predictions for IGM gas conditions of quasar J021043. This is 20 h^{-1} cMpc segment centred at $z = 4$, overlaid with noise vector. The predictions for $\log \Delta_\tau$ and $\log T_\tau$ are shown in middle and bottom panels, respectively. The predicted mean (dashed curves) along with 1σ confidence intervals are shown. The estimated T_0 and γ, with 1σ confidence interval, are also shown in the last row.

with traditional methods that only recover thermal parameters using statistics such as the Lyα flux power spectrum. We have shown that only one 20 h^{-1} cMpc segment of the Lyα forest from a single quasar can deliver constraints comparable to moderately sized data sets usually employed for such studies. We have seen that our method can provide fairly robust constraints on thermal parameters even in the presence of significant instrumental noise. We can also perform a reasonable reconstruction with test data set from NYX simulations by our trained neural network with frozen weights. In addition, the method can be extended up to redshift $z = 5.0$, providing valuable insight into the thermal evolution of IGM. However, we expect the performance to get worse with mostly saturated spectra; therefore, it might require a significant change in the current architecture. The technique can also be pushed towards lower redshifts $z \lesssim 4$, until most of the flux is at the continuum level.

Neural networks utilize quite complex feature-space transformation to convert Lyα flux to the IGM conditions. This can potentially make inference somewhat more model-dependent than traditional methods. For instance, traditional methods cannot capture the small deviation from a power law in the temperature–density plane using the statistical thermal parameters T_0 and γ. As our method can reconstruct the entire plane, it requires the mock spectra to be a realistic representation of observations. This includes incorporating all the physical processes that can impact the IGM conditions such as non-equilibrium photoionization. An important aspect of training large neural networks is to come up with clever solutions to overfitting problems. We found that training can be sensitive to noise realization, skewers with correlated density structures, and more importantly the network architecture. We have taken appropriate measures to overcome these problems by adopting strategies for overfitting by adding noise during training stage, constructing realistic mock data sets, and performing a grid search over the hyperparameters of the network.

In the future, we plan to provide thermal parameter constraints using observational spectra at $4 \le z \le 5$. We have shown a glimpse of IGM gas conditions' reconstruction from a real spectrum in Section 4.6. This unique approach enables insight into the IGM using individual 20 h^{-1} cMpc segments contrary to current methods that require averaging together a much larger volume of the Universe. Another possible implication of our method is to perform reconstruction for models with thermal fluctuations at $z = 5.0$. This method would require grid of inhomogeneous reionization simulations on much larger scales. Potentially, we can see evidence of excess scatter in the distribution of recovered thermal parameters along individual sightlines.

ACKNOWLEDGEMENTS

We thank Jose Onorbe for sharing the NYX simulations and useful discussion. The Sherwood-Relics simulations and their post-processing were performed using the Curie supercomputer at the Tre Grand Centre de Calcul (TGCC), and the DiRAC Data Analytic system at the University of Cambridge, operated by the University of Cambridge High Performance Computing Service on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). This equipment was funded by BIS National E-infrastructure capital grant (ST/K001590/1), STFC capital grants ST/H008861/1 and ST/H00887X/1, and STFC DiRAC Operations grant ST/K00333X/1. DiRAC is part of the National E-Infrastructure. Computations in this work were also performed using the CALX machines at IoA. Support by ERC Advanced Grant 320596 'The Emergence of Structure During the Epoch of Reionization' is gratefully acknowledged.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

Almgren A. S., Bell J. B., Lijewski M. J., Lukić Z., Van Andel E., 2013, ApJ, 765, 39
Becker G. D., Bolton J. S., 2013, MNRAS, 436, 1023
Becker G. D., Bolton J. S., Haehnelt M. G., Sargent W. L. W., 2011, MNRAS, 410, 1096
Becker G. D., Bolton J. S., Lidz A., 2015, Publ. Astron. Soc. Aust., 32, e045
Boera E., Murphy M. T., Becker G. D., Bolton J. S., 2014, MNRAS, 441, 1916
Boera E., Murphy M. T., Becker G. D., Bolton J. S., 2016, MNRAS, 456, L79
Boera E., Becker G. D., Bolton J. S., Nasir F., 2019, ApJ, 872, 101
Bolton J. S., Viel M., Kim T. S., Haehnelt M. G., Carswell R. F., 2008, MNRAS, 386, 1131
Bolton J. S., Becker G. D., Raskutti S., Wyithe J. S. B., Haehnelt M. G., Sargent W. L. W., 2012, MNRAS, 419, 2880
Bolton J. S., Becker G. D., Haehnelt M. G., Viel M., 2014, MNRAS, 438, 2499
Bolton J. S., Puchwein E., Sijacki D., Haehnelt M. G., Kim T.-S., Meiksin A., Regan J. A., Viel M., 2017, MNRAS, 464, 897
Bosman S. E. I., Fan X., Jiang L., Reed S., Matsuoka Y., Becker G., Haehnelt M., 2018, MNRAS, 479, 1055
Calura F., Tescari E., D'Odorico V., Viel M., Cristiani S., Kim T.-S., Bolton J. S., 2012, MNRAS, 422, 3019
Chollet F. et al., 2015, Keras, available at https://github.com/fchollet/keras
Croft R. A. C., Weinberg D. H., Bolte M., Burles S., Hernquist L., Katz N., Kirkman D., Tytler D., 2002, ApJ, 581, 20
D'Aloisio A., McQuinn M., Trac H., 2015, ApJ, 813, L38
Eilers A.-C., Hogg D. W., Schölkopf B., Foreman-Mackey D., Davies F. B., Schindler J.-T., 2022, ApJ, 938, 17
Gaikwad P., Srianand R., Choudhury T. R., Khaire V., 2017, MNRAS, 467, 3172
Gaikwad P. et al., 2020, MNRAS, 494, 5091
Gaikwad P., Srianand R., Haehnelt M. G., Choudhury T. R., 2021, MNRAS, 506, 4389
Garzilli A., Bolton J. S., Kim T.-S., Leach S., Viel M., 2012, MNRAS, 424, 1723
Goodfellow I. J., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA, USA
Haardt F., Madau P., 2012, ApJ, 746, 125
Haehnelt M. G., Steinmetz M., 1998, MNRAS, 298, L21
Harrington P., Mustafa M., Dornfest M., Horowitz B., Lukić Z., 2022, ApJ, 929, 160
He K., Zhang X., Ren S., Sun J., 2015, preprint (arXiv:1512.03385)
Hiss H., Walther M., Hennawi J. F., Oñorbe J., O'Meara J. M., Rorai A., Lukić Z., 2018, ApJ, 865, 42
Hiss H., Walther M., Oñorbe J., Hennawi J. F., 2019, ApJ, 876, 71
Hu T., Khaire V., Hennawi J. F., Tripp T. M., Oñorbe J., Walther M., Lukic Z., 2023, preprint (arXiv:2311.17895)
Huang L., Croft R. A. C., Arora H., 2021, MNRAS, 506, 5212
Hui L., Gnedin N. Y., 1997, MNRAS, 292, 27
Ioffe S., Szegedy C., 2015, preprint (arXiv:1502.03167)
Iršič V. et al., 2017, Phys. Rev. D, 96, 023522
Iršič V. et al., 2024, Phys. Rev. D, 109, 043511
Keating L. C., Puchwein E., Haehnelt M. G., 2018, MNRAS, 477, 5501
Lee K.-G. et al., 2015, ApJ, 799, 196
Lidz A., Heitmann K., Hui L., Habib S., Rauch M., Sargent W. L. W., 2006, ApJ, 638, 27
Lidz A., Faucher-Giguère C.-A., Dall'Aglio A., McQuinn M., Fechner C., Zaldarriaga M., Hernquist L., Dutta S., 2010, ApJ, 718, 199
McDonald P., Miralda-Escudé J., Rauch M., Sargent W. L. W., Barlow T. A., Cen R., 2001, ApJ, 562, 52
McQuinn M., Upton Sanderbeck P. R., 2016, MNRAS, 456, 47
Meiksin A., 2000, MNRAS, 314, 566
Miralda-Escudé J., Rees M. J., 1994, MNRAS, 266, 343
Murphy M. T., Kacprzak G. G., Savorgnan G. A. D., Carswell R. F., 2019, MNRAS, 482, 3458
Nasir F., Bolton J. S., Becker G. D., 2016, MNRAS, 463, 2335
Nayak P., Walther M., Gruen D., Adiraju S., 2024, A&A, 689, A153
Oñorbe J., Hennawi J. F., Lukić Z., 2017, ApJ, 837, 106
O'Shea K., Nash R., 2015, preprint (arXiv:1511.08458)
Padmanabhan H., Srianand R., Choudhury T. R., 2015, MNRAS, 450, L29
Peeples M. S., Weinberg D. H., Davé R., Fardal M. A., Katz N., 2010, MNRAS, 404, 1281
Puchwein E., Bolton J. S., Haehnelt M. G., Madau P., Becker G. D., Haardt F., 2015, MNRAS, 450, 4081
Puchwein E., Haardt F., Haehnelt M. G., Madau P., 2019, MNRAS, 485, 47
Puchwein E. et al., 2023, MNRAS, 519, 6162
Ricotti M., Gnedin N. Y., Shull J. M., 2000, ApJ, 534, 41
Rudie G. C., Steidel C. C., Pettini M., 2012, ApJ, 757, L30

Schaye J., Theuns T., Rauch M., Efstathiou G., Sargent W. L. W., 2000, MNRAS, 318, 817
Springel V., 2005, MNRAS, 364, 1105
Telikova K. N., Shternin P. S., Balashev S. A., 2019, ApJ, 887, 205
Theuns T., Schaye J., Haehnelt M. G., 2000, MNRAS, 315, 600
Viel M., Becker G. D., Bolton J. S., Haehnelt M. G., 2013, Phys. Rev. D, 88, 043502
Villasenor B., Robertson B., Madau P., Schneider E., 2023, Phys. Rev. D, 108, 023502
Walther M., Oñorbe J., Hennawi J. F., Lukić Z., 2019, ApJ, 872, 13
Wang R., Croft R. A. C., Shaw P., 2022, MNRAS, 515, 1568
Wolfson M., Hennawi J. F., Davies F. B., Oñorbe J., Hiss H., Lukić Z., 2021, MNRAS, 508, 5493
Zaldarriaga M., 2002, ApJ, 564, 153
Zaldarriaga M., Hui L., Tegmark M., 2001, ApJ, 557, 519
Zaroubi S., Viel M., Nusser A., Haehnelt M., Kim T.-S., 2006, MNRAS, 369, 734

APPENDIX A: COMPARISON WITH REAL-SPACE DISTRIBUTIONS

In this paper, we have chosen to work with optical depth-weighted quantities and have determined thermal parameters using predicted $\log \Delta_\tau$–$\log T_\tau$ distribution realizations. In actuality, the weighted quantities are just a proxy for real-space distribution, i.e. $\log T$–$\log \Delta$. So, a comparison of density, temperature, and T_0–γ distributions using real-space and optical depth-weighed quantities is presented in this section. We have summarized the results in Figs A1 and A2.

In Fig. A1, we have compared real-space (red dotted), Lyα optical depth-weighted (black dashed), and predicted (blue solid) quantities for density (top row) and temperature (bottom row) distributions. The shaded regions represent the 1σ scatter over 20 h^{-1} cMpc skewers, which is appropriately coloured.

As expected, the IGM density distributions (comparing between curves across panels in first row) are very similar. Furthermore, the real-space, optical depth-weighted, and predicted quantities for given model (comparing curves in single panel) also closely match, although the distributions of optical depth-weighted density (blue curves) have a slight tail at the high-density end, $\log \Delta_\tau \simeq 0.2$. The reason is that the act of optical depth-weighting shifts slightly underdense gas into mild overdensities. A comparison of the shaded regions suggests that all quantities exhibit a very similar scatter over 20 h^{-1} cMpc skewers.

The real-space (red) and optical depth-weighted (black) temperature (bottom row in Fig. A1) also indicate a very similar trend, where the latter has tail at the higher temperature end. In addition, there is significantly more scatter at around median temperatures in the optical depth-weighted case for the reasons discussed before. Overall, the predicted distributions are broader than the rest and show significantly more scatter as well. Note that at the low-temperature end the distribution cannot capture the sharp rise for models with relatively lower γ (panels 1 and 2), which gives a noticeable tail. The reason is that models with relatively shallower slope have a small range of temperatures that corresponds to a large range of densities.

Fig. A2 shows a comparison of T_0–γ distributions using $\log \Delta$–$\log T$ (real-space), $\log \Delta_\tau$–$\log T_\tau$ (optically weighted), and realizations of $\log \Delta_\tau$–$\log T_\tau$ plane. The 68th and 95th percentiles of these distributions are shown as dashed, dotted, and solid contours, respectively.

It is obvious that the predicted T_0–γ distributions are broader than the rest. This reassures us that we do not have to incorporate additional uncertainties from our choice of optical depth-weighted quantities into our predictions. Upon close inspection, we can see a subtle bias between actual optical depth-weighted and real-space distributions. The former has slightly lower value of T_0 ($\sim$500 K) for majority of models. Notice that this also causes the predicted distributions to slightly underpredict T_0 for majority of models, which can be seen by comparing the plus symbol with the dashed contours. Despite these subtle biases, the real-space distribution broadly lies within 1σ of the predicted optical depth-weighted distributions.

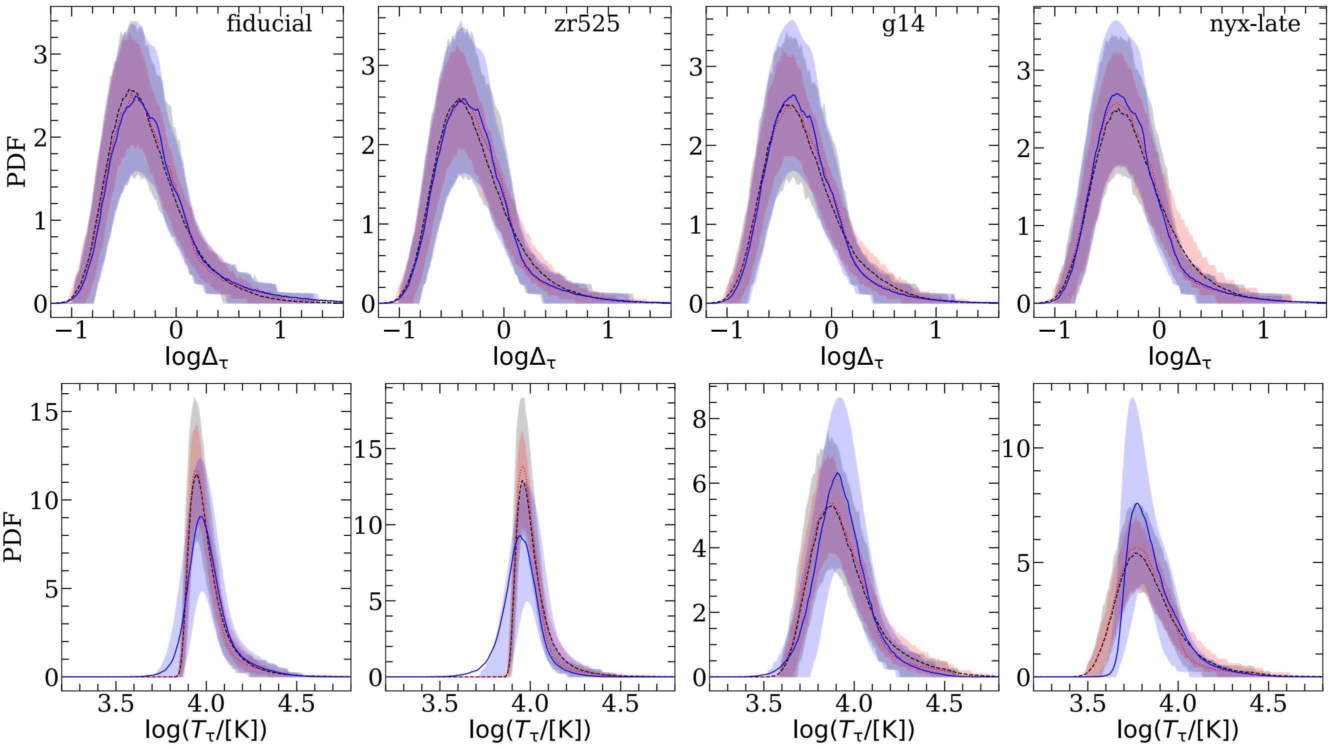

Figure A1. Same as for Fig. 8, but now for selected models shown in legends. Each panel shows real-space (dashed), Lyα optical depth-weighted (dotted), and predicted (solid) quantities.

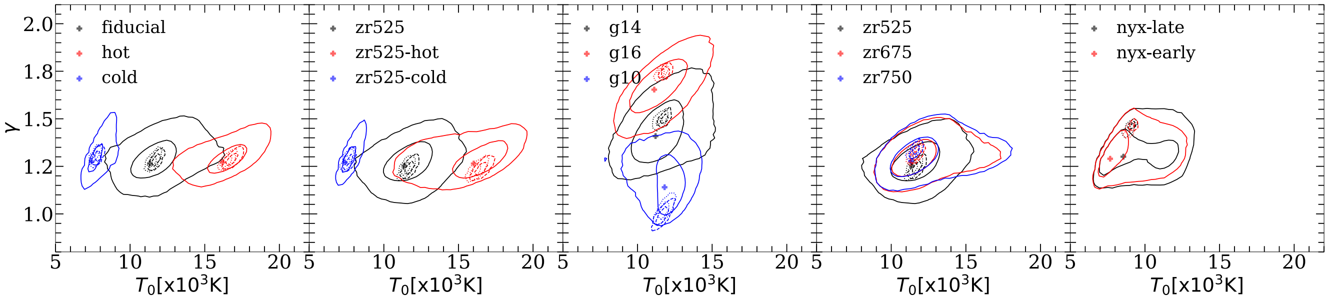

Figure A2. Same as for Fig. 10 but now a comparison between real-space (dashed), Lyα optical depth-weighted (dotted), and predicted (solid) quantities with S/N = 50. The contours enclose 1σ and 2σ scatter over 20 h^{-1} cMpc skewers.

APPENDIX B: MEAN FLUX TESTS

To quantity the impact of uncertainties on the mean flux on our T_0–γ predictions, we rescale Lyα flux to match $\langle F \rangle = 0.468\,05$ and 0.382 95, which is 10 per cent higher and lower than our value for S/N = 50 pixel^{-1} case. These rescaled data sets are used to provide the predictions from network with frozen weights and shown in Fig. B1. Overall, there is no noticeable change; however, there is some subtle change in γ that is underpredicted most noticeably (2–3 per cent) for higher mean flux case.

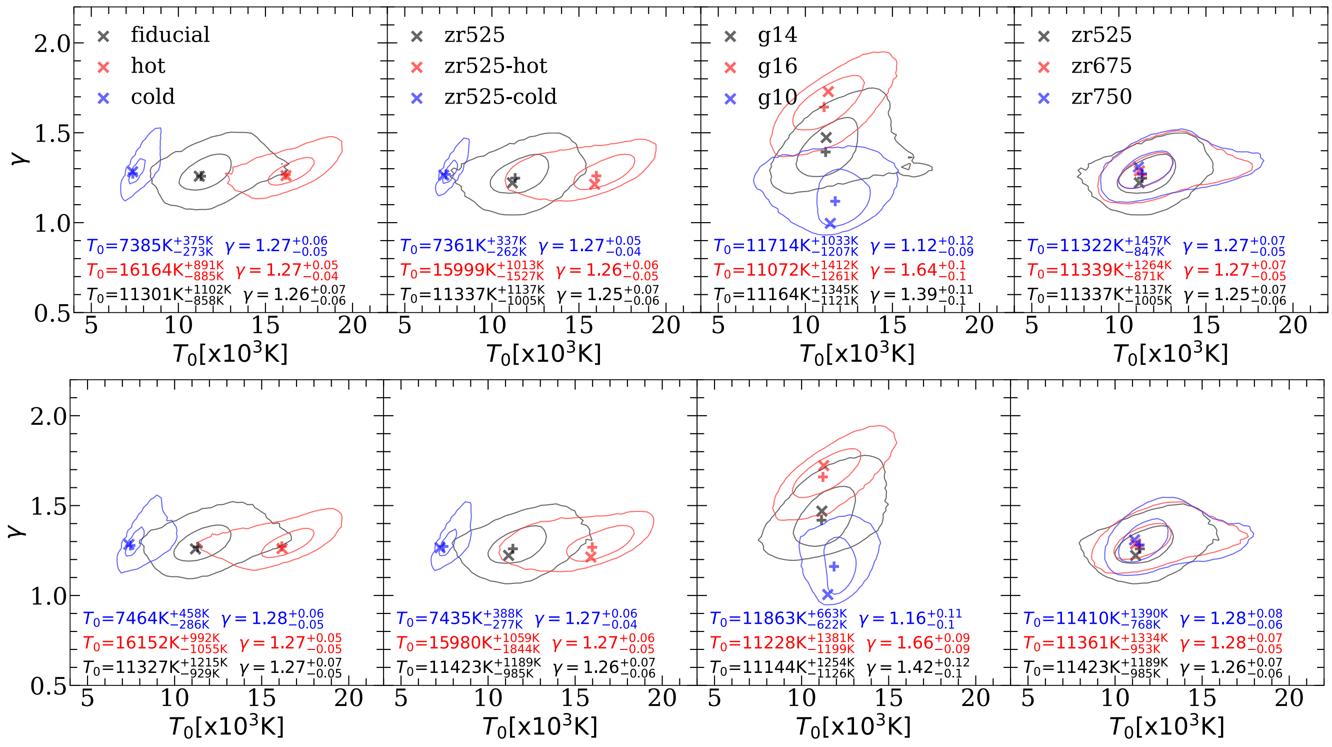

Figure B1. Same as for Fig. 10, but now for Lyα flux matched to mean flux 10 per cent higher (top) and lower (bottom) than fiducial value at $z = 4.0$ with S/N = 50 pixel^{-1}.

APPENDIX C: BOX SIZE TEST

We have taken six more runs from Sherwood-Relics to see the impact of box size on the network predictions with fixed mass resolution. These runs have box length of 40 h^{-1} cMpc and have 2048^3 gas and dark matter particles. We have taken 20 h^{-1} cMpc long skewers from these boxes for this exercise. These skewers are only used at the prediction stage. Our network relies on the periodicity of skewers during training, which is not the case for these skewers taken from 40 h^{-1} cMpc box. Therefore, we slightly modified our training by masking (16 pixels) at start of Lyα skewer during training with our 20 h^{-1} cMpc boxes to break dependence of network on periodicity at boundaries. The masking was done after each skewer was periodically shifted by a random value. We use this slightly modified network to obtain predictions for 40 h^{-1} cMpc runs (model shown in legend) as shown in T_0–γ distributions in Fig. C1. Although the mean values remain largely unchanged, the predicted distributions are in quite broad. This is partly because network cannot rely on periodic boundary and partly due to box size impacting the T_0–γ distribution prediction.

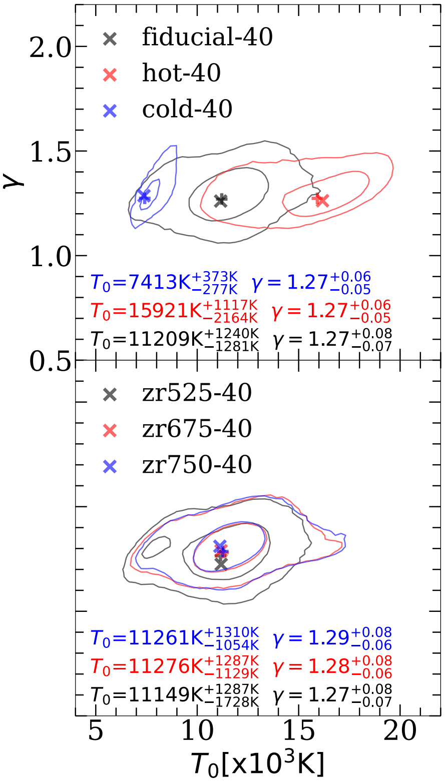

Figure C1. Same as for Fig. 10, but now for 40 h^{-1} cMpc boxes from Sherwood-Relics at $z = 4$ with S/N = 50 pixel^{-1}.

This paper has been typeset from a TEX/LATEX file prepared by the author.