Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **517,** 3200–3211 (2022) https://doi.org/10.1093/mnras/stac2435
Advance Access publication 2022 August 29

A multifidelity emulator for the Lyman-α forest flux power spectrum

M. A. Fernandez ,★ Ming-Feng Ho and Simeon Bird

Department of Physics and Astronomy, University of California Riverside, 900 University Ave, Riverside, CA 92521, USA

Accepted 2022 August 25. Received 2022 August 19; in original form 2022 July 13

ABSTRACT

In this work, we extend our recently developed multifidelity emulation technique to the simulated Lyman-α forest flux power spectrum. Multifidelity emulation allows interpolation of simulation outputs between cosmological parameters using many cheap low-fidelity simulations and a few expensive high-fidelity simulations. Using a test suite of small-box (30 Mpc h^{-1}) simulations, we show that multifidelity emulation is able to reproduce the Lyman-α forest flux power spectrum well, achieving an average accuracy when compared to a test suite of 0.8 per cent. We further show that it has a substantially increased accuracy over single-fidelity emulators, constructed using either the high- or low-fidelity simulations only. In particular, it allows the extension of an existing simulation suite to smaller scales and higher redshifts.

Key words: methods: numerical – methods: statistical – cosmology: theory.

1 INTRODUCTION

The modern and future test bed for cosmology lies in small scales and non-linear structures. Cosmological analyses exploit observations of these scales to explore questions such as the nature of dark matter, the total neutrino mass, and the thermal history of the intergalactic medium (IGM). One of the most powerful probes of small-scale structure is the Lyman-α forest, a series of absorption features in the spectrum of quasars (Gunn & Peterson 1965; Theuns et al. 1998; McDonald et al. 2000, 2006; Hui et al. 2001; Viel et al. 2002; Fan et al. 2006; Viel & Haehnelt 2006). Numerical simulations are required to analyse these observations as they probe the non-linear regime. As these simulations are expensive, cosmologists build emulators (Heitmann et al. 2006, 2009; Habib et al. 2007), which interpolate a summary statistic (in this case the 1D Lyman-α forest flux power spectrum) between simulation outputs at different cosmological parameters.

A recent development in cosmology is the application of multi-fidelity emulators, which allow simulations with different particle loads, and thus costs, to be combined together (Ho, Bird & Shelton 2022). Here, we adapt the multifidelity emulation technique to the Lyman-α forest 1D flux power spectrum. Multifidelity emulation is especially useful in this context because the Lyman-α forest probes a range of redshifts, and is sensitive to smaller scales, which require higher resolution simulations, at higher redshifts (Bolton & Becker 2009). The models developed here will allow a single emulator to target the wide range of scales probed by the Lyman-α forest, which would otherwise require a computationally infeasible number of very large simulations (Borde et al. 2014).

The Lyman-α forest is the result of overlapping neutral hydrogen absorption profiles in the spectra from distant luminous quasars, processed through the expansion of the Universe (Gunn & Peterson 1965). As light travels from the quasar, it passes through neutral hydrogen gas of varying densities. In the rest frame of those neutral hydrogen islands, light that has been redshifted close to the Lyman-α transition at 1215.67 Å will be absorbed and the rest transmitted. This is repeated as the light continues to intersect more neutral hydrogen islands on its path towards us, the observers. The result is a quasar transmission spectrum containing an overlapping field of absorption features that provides a proxy to the dark matter density along that sightline (Croft et al. 1998).

The densities probed by the Lyman-α forest, from redshift 2 to 5, are $\sim$1–100 times the cosmological mean density. At these densities, stellar winds and star formation effects are negligible, although black hole feedback is important (Viel, Schaye & Booth 2013b; Chabanier et al. 2020). The densities along with the range of scales accessed have made the Lyman-α forest popular in cosmological studies, including constraining the thermal history of the IGM and thus reionization (Bolton et al. 2008, 2014; Nasir, Bolton & Becker 2016; Boera et al. 2019; Wu et al. 2019; Gaikwad et al. 2021; Villasenor et al. 2021), constraining cosmological parameters including the neutrino mass (Viel, Haehnelt & Springel 2004; McDonald et al. 2005; Seljak et al. 2005; Seljak, Slosar & McDonald 2006; Viel, Haehnelt & Lewis 2006; Palanque-Delabrouille et al. 2020; Garny et al. 2021), and testing alternatives to cold dark matter (Viel et al. 2005, 2013a; Iršič et al. 2017a; Palanque-Delabrouille et al. 2020; Garzilli et al. 2021; Rogers & Peiris 2021b).

The Lyman-α forest 1D flux power spectrum is the most commonly used summary statistic for Lyman-α forest spectra. It probes small-scale structure by measuring the two-point Fourier-space correlation between neutral hydrogen absorption within a sightline (Croft et al. 1998).

Current observational measurements of the Lyman-α forest flux power spectrum come from either a lower resolution, larger sample survey (Sloan Digital Sky Survey, SDSS; Chabanier et al. 2019) or various higher resolution, smaller sample surveys (Iršič et al. 2017b; Day, Tytler & Kambalur 2019; Karaçaylı et al. 2022). In Chabanier et al. (2019), the flux power spectrum constructed from Baryon Oscillation Spectroscopic Survey (BOSS) and extended BOSS

★ E-mail: mfern027@ucr.edu

(eBOSS) spectra accesses redshifts from $z = 2.2$ to 4.6 (6 per cent and 18 per cent average uncertainty, respectively) and scales from $k \approx 0.001$ to 0.02 s km^{-1} (6 per cent and 14 per cent average uncertainty, respectively). The small sample, higher resolution surveys generally access a similar redshift range, but shift both the largest and smallest scales to higher k. For example, in Karaçaylı et al. (2022) (their conservative results), using spectra from multiple surveys (XQ-100, KODIAQ, and SQUAD), they access redshifts from $z = 2$ to 4.6 (7 per cent and 27 per cent average uncertainty, respectively) and scales from $k \approx 0.005$ to 0.1 s km^{-1} (12 per cent and 9 per cent average uncertainty, respectively).

The Dark Energy Spectroscopic Instrument (DESI) will soon report its first-year results. Ultimately, it will increase the number of Lyman-α quasar spectra by a factor of 4 over SDSS. This corresponds to ∼50 quasars per square degree and a total of 7×10^5 quasars over the 14 000 deg^2 survey footprint (DESI Collaboration et al. 2016). In addition, DESI is expected to measure the 1D flux power spectrum at smaller scales ($k < 0.035$ s km^{-1}) and higher redshifts ($z > 4.6$) than SDSS, achieving an accuracy of the order of a few per cent (Valluri et al. 2022).

Extracting cosmological information from these observations will require simulations that follow the distribution of gas at relevant densities and on relevant scales. For the Lyman-α forest, box sizes of at least 100 Mpc h^{-1} and mean particle spacing of 100/3072 ≈ 0.03 Mpc h^{-1} are necessary (Borde et al. 2014). Earlier work has focused on methods that can reduce the cost of such simulations. Borde et al. (2014) used a splicing technique to produce high-resolution, large-volume outputs from three sets of less computationally intensive simulations: low resolution, large volume; high resolution, small volume; and low resolution, small volume. Lukić et al. (2015) explored the use of Richardson extrapolation to enhance output resolution, in addition to testing the splicing technique.

Parameter inference tasks, such as a direct Markov chain Monte Carlo analysis, require ∼10^5–10^6 model evaluations, indicating the number of simulations required by a naive approach. Even using techniques such as splicing, this is computationally infeasible. However, using a significantly reduced number of simulations (∼30), an emulator can be constructed that effectively interpolates between this smaller set of simulations. In addition to the Lyman-α forest, emulators have been used extensively in cosmology for studying the matter power spectrum (Heitmann et al. 2009, 2014; Lawrence et al. 2017; Giblin et al. 2019; Aricò et al. 2021; Euclid Collaboration et al. 2021; Giri & Schneider 2021), weak lensing (Harnois-Déraps, Giblin & Joachimi 2019; Davies et al. 2021), the halo mass function (McClintock et al. 2019; Nishimichi et al. 2019; Bocquet et al. 2020), and the 21-cm signal (Kern et al. 2017; Cohen et al. 2020; Bevins et al. 2021; Bye, Portillo & Fialkov 2022). Still, the computational resources required to run ∼30 simulations with the requisite volume and resolution are highly restrictive, especially for the Lyman-α forest.

Here, we use modern machine learning techniques to alleviate the computational resource cost associated with constructing an emulator. Specifically, we are concerned with using machine learning to predict high-resolution simulation outputs to a high degree of accuracy, while running a minimal number of high-resolution simulations. One machine learning method that is suited to this task is a Gaussian process (GP) emulator. GPs (Rasmussen & Williams 2006) are a means of interpolating between the simulation outputs, providing function prediction in a Bayesian framework. Essentially, a distribution of functions is learned through training on simulations, and the mean (best estimate) and variance (interpolation error) of the output can be returned for arbitrary simulation inputs. While other interpolation methods are possible, GPs have many benefits: the inherent quantification of prediction uncertainty, the option to incorporate prior knowledge, and the ability to interpolate within high-dimensional parameter space.

Previous uses of GP emulators for the Lyman-α forest have been shown to be effective at predicting summary statistics (Bird et al. 2019; Rogers et al. 2019; Pedersen et al. 2021; Rogers & Peiris 2021a, b; Walther et al. 2021). Bird et al. (2019) self-consistently showed that the predicted flux power from their GP emulator (trained with 21 simulations) agreed to within 1–2 per cent of the corresponding simulation flux power spectrum. These GP emulators still require a substantial computational cost, as the full simulation suite must be run with sufficient volumes and resolutions for the Lyman-α forest.

Recently, Ho et al. (2022) implemented a multifidelity GP emulator (Kennedy & O'Hagan 2000) for the matter power spectrum. Here, we combine and expand on the methods outlined in Bird et al. (2019) and Ho et al. (2022), to produce a multifidelity GP emulator for the Lyman-α forest flux power spectrum. In our multifidelity model, the training simulations are split into two fidelities: a large sample of low-resolution simulations (low fidelity, LF) and a small subset of these simulations run at higher resolution (high fidelity, HF). Note that we use fidelity and resolution interchangeably throughout this work. Using these two training sets, the multifidelity emulator is trained to predict the 1D flux power spectrum that would be output by a *high*-resolution simulation for arbitrary cosmological and astrophysical parameters.

A multifidelity emulator allows us to replace some of the HF simulations that would be needed in a single-fidelity emulator with LF simulations. This can dramatically reduce the computational cost of constructing an emulator, while retaining predictive power across parameter space. Using this method, emulators can be constructed that make use of the full range of scales and redshifts probed by Lyman-α forest observations. This enables analyses that can jointly constrain thermal, astrophysical, and cosmological parameters.

In our high-resolution simulations, the Lyman-α forest flux power spectrum is converged to ≈ 5 per cent. While the scales we probe in this work are resolved, the box size we use is smaller than required to analyse the full range of scales available in Lyman-α forest data, i.e. we cannot compute a likelihood function using all the real data without larger boxes. We therefore defer a full cosmological likelihood analysis to future work. Our goal is to quantitatively test the accuracy of the emulator output, as compared with the output from a set of testing simulations run at the same resolution as the HF training set, and demonstrate the validity and utility of the multifidelity technique. Specifically, we quantify the accuracy of the single- and multifidelity emulators with respect to true values from the testing simulations, thus determining how effective the multifidelity model is at producing high-resolution outputs at minimal computational cost.

2 SIMULATIONS

Simulations were performed using MP-GADGET[1], an *N*-body and smoothed particle hydrodynamics (SPH) code built on the solid base of GADGET-3 (last described in Springel 2005). MP-GADGET has been substantially modified to include shared-memory parallelism using OpenMP, together with many other algorithmic improvements and new subgrid models, as described in Bird et al. (2020, 2022) and Ni et al. (2022). The initial power spectrum and transfer functions

[1] https://github.com/MP-Gadget/MP-Gadget

are generated with the Boltzmann code CLASS (Lesgourgues 2011). Species-specific initial conditions are generated for baryons and dark matter (Bird et al. 2020; Fernandez, Bird & Upton Sanderbeck 2021). We include radiation in the cosmological background model and assume massless neutrinos.

The physics models largely follow those used for the ASTRID simulation, which are described in Bird et al. (2022). Primary sources for these models, as well as changes from Bird et al. (2022), are described below.

We use a cubic kernel for our density estimator (rather than a quintic kernel). For these simulations, the cubic kernel, in addition to running faster, produced a neutral hydrogen column density distribution that was more consistent with observations for column densities between 10^{20} and 10^{22} cm^{-2} (Ho, Bird & Garnett 2021). We continue to use the pressure–entropy formulation of SPH. The smaller SPH kernel increases the noise within galaxies, but has minimal effect on the Lyman-α forest (Bird et al. 2013).

Star formation follows the model of Springel & Hernquist (2003), with our specific implementation as described in Feng et al. (2016). We lower the number of stars produced per gas particle from 4 (used in ASTRID) to 1 (as in Illustris-TNG), which speeds up the simulation without having an effect on the Lyman-α forest.

Black holes follow the model of Ni et al. (2022). We found that for the resolutions used here, the dynamic friction from gas led to a few black holes escaping from their dark matter halo, so we only use dynamic friction from dark matter and stars. The black hole feedback factor, which controls the fraction of luminosity that is converted into thermal energy, is an emulator parameter (BHF), with the associated parameter limits in Fig. 3. The black hole feedback radius is fixed to 3 kpc h^{-1}, selected to be the average black hole feedback radius at the highest tested resolution when using a nearest neighbour distance. To accommodate a lower mass resolution than ASTRID, the minimum stellar mass needed in a halo to seed a black hole was increased to 2×10^8 M$_\odot$, and all black hole seeds start with a mass of 5×10^4 M$_\odot$.

Stellar winds are modelled following Okamoto et al. (2010). The decoupling distance for the winds is increased from 20 to 1 Mpc h^{-1}, which allows the winds to recouple due to density changes rather than travel distance. The density threshold for wind recoupling is set to 10 per cent of the star formation density threshold (which is 57.7 times the critical density). The minimum wind velocity is set to 100 km s^{-1}. Finally, metal return (gas enrichment) is disabled as it is not important for the Lyman-α forest and can be computationally expensive.

Gas is assumed to be in ionization equilibrium with a uniform ultraviolet background using the model of Faucher-Giguère (2020). We boost the temperature of the gas to 15 000 K, the time-step after the gas is reionized, to model impulsive heating during hydrogen reionization from ionization fronts (D'Aloisio et al. 2019).

We implement He II reionization using the model of Upton Sanderbeck & Bird (2020). The input parameters for this model are quasar mean bubble size and variance, redshifts for the start and completion of He II reionization $z_i^{\rm He\,II}$, $z_f^{\rm He\,II}$, and the quasar spectral index α_q (which effectively scales the peak temperature during He II reionization). The quasar bubble size is reduced from the default of ∼30 Mpc, motivated by radiative transfer simulations (McQuinn et al. 2009), to 5 Mpc, due to our small box size.

Simulations are initialized at $z = 99$ and finish at $z = 2$, and use periodic boundaries. Box volume, particle number, and gas particle mass resolution are reported in Table 1. The range given for the gas resolution is due to the varying value of h in our simulation suite. The gas particle mass resolution for our HF simulations does not meet the resolution that Bolton & Becker (2009) recommend to resolve the forest at all redshifts of interest. However, the Lyman-α forest flux power spectrum from our HF simulations is converged to within ≈ 5 per cent of a simulation that does meet the required resolution of Bolton & Becker (2009). We are interested in the performance of the multifidelity GP emulator in learning the mapping from low to high resolution; thus, this slight lack of numerical convergence does not affect our results. Examples of the gas density (at $z = 3.6$) for the two resolutions are shown in the top and bottom panels of Fig. 1.

Lyman-α forest absorption spectra are generated using the fake spectra flux extractor (Bird 2017),[2] described in Bird et al. (2015). We generate 32 000 (seeded) randomly placed skewers for each snapshot, from $z = 5.4$ to 2.0 in increments of $\Delta z = 0.2$. The pixel resolution is set to 10 km s^{-1}. An optical depth threshold of $\tau < 10^6$ is set to eliminate damped Lyman-α systems. Example spectra from one LF and one HF simulation (at $z = 4$) are shown in the middle panel of Fig. 1. Note that the LF simulation is not simply a smoothed version of the HF simulation, as the fine velocity structure of the gas moves the location of the absorption peaks.

These sets of neutral hydrogen absorption spectra are used to construct the Lyman-α forest flux power spectrum for each simulation, at each redshift. The flux power spectrum is defined as $P_F(k) = |L^{-1}\tilde{\delta}_F^2(k)|$, where $\tilde{\delta}_F^2(k)$ is the Fourier transform of the flux excess, $\delta_F(k) = F(k)/\langle F(k)\rangle - 1$, and L is the length of the sightline. The reported flux power spectrum is averaged over all 32 000 spectra.

Fig. 2 shows flux power spectra from a single LF simulation and its HF counterpart, and the ratio of these at several redshifts. While the exact difference between the LF and HF flux power spectra depends on simulation input parameters and redshift, in general the LF differs most from the HF at small scales. The enhanced power on large scales for the LF flux power spectra is consistent with Borde et al. (2014), and is likely due to differences in heating and cooling during H I and He II reionization.

Fig. 3 lists the input parameters that are varied across our suite of simulations, as well as their limits. Two parameters control the primordial power spectrum: n_s is the scalar spectral index (slope) and A_p is the amplitude (see Bird et al. 2019 for more details). Three of the parameters relate to the He II reionization model: $z_i^{\rm He\,II}$ and $z_f^{\rm He\,II}$ are the redshifts for the start and end of He II reionization, and α_q is the quasar spectral index. We vary the Hubble constant through h, and the total matter density through $\Omega_{\rm M}h^2$. One parameter is varied for H I reionization: $z^{\rm H\,I}$ is the midpoint redshift of H I reionization. Finally, we vary one parameter for the black hole model: BHF is the black hole feedback factor, which controls the fraction of luminosity that is converted into thermal energy. Note that our simulations do not contain a kinetic feedback model. However, at $z > 2$, it is expected that the thermal mode dominates. Also shown in Fig. 3 are the LF and HF training samples. Note that the HF samples are a subset of the LF samples. The selection of the HF samples is described in Section 3.2.3.

3 EMULATORS

In Section 3.1, we will briefly review emulation using a GP. In Section 3.2, we will review how the GP emulator can be extended to model simulations with different qualities using a multifidelity emulator, MFEmulator. The multifidelity emulation technique of Kennedy & O'Hagan (2000) will be reviewed in Section 3.2.1. Section 3.2.2 will discuss the differences in multifidelity emulator design between this paper and Ho et al. (2022). Finally, Section 3.2.3

[2] https://github.com/sbird/fake_spectra

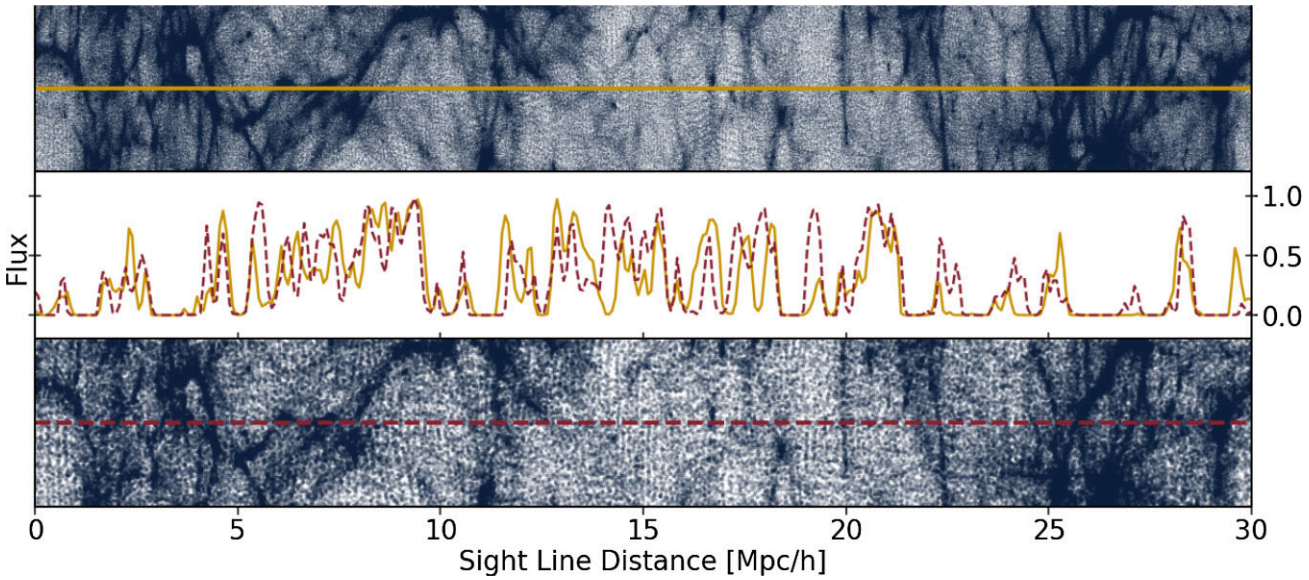

Figure 1. Example Lyman-α forest spectra and corresponding gas density from simulations at redshift 4. The top and bottom panels show the simulated gas surrounding the skewer, which produced the spectra shown in the middle panel. Examples are shown for simulations run at high (top panel, yellow line) and low resolution (bottom panel, red line).

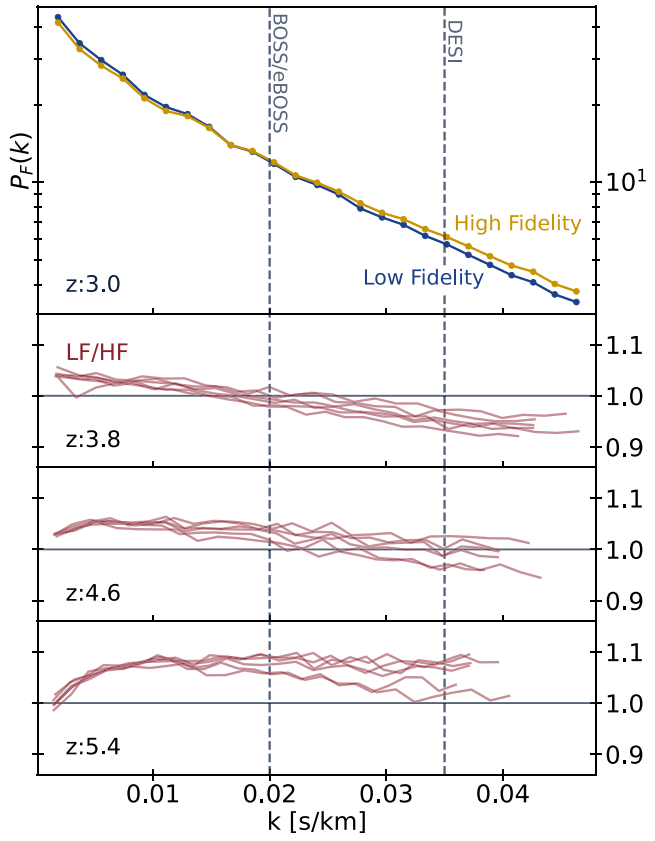

Figure 2. Lyman-α forest flux power spectrum from LF and HF simulations. The top panel shows the flux power spectrum at redshift 3 from an LF simulation (blue) and its HF counterpart (yellow). The lower panels show the ratio of the flux power for these two resolutions, for all LF–HF simulation pairs, at $z = 3.8, 4.6,$ and 5.4. Dashed lines show the highest k probed by BOSS/eBOSS (Chabanier et al. 2019) and the estimated reach for DESI (Valluri et al. 2022).

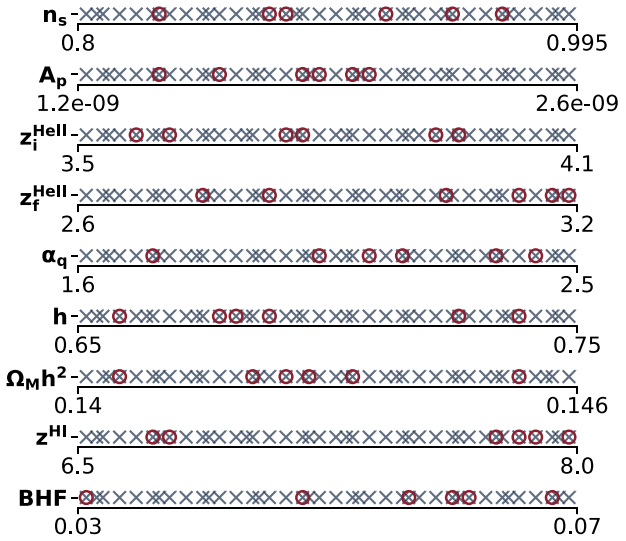

Figure 3. Simulation parameter limits and samples. Parameters for the low-resolution simulations (crosses) were determined by filling a Latin hypercube. Initially, 30 low-resolution samples were generated, then an additional 10 were added while maintaining the Latin hypercube method, hence the non-uniform spacing for the low-resolution samples. The optimal subset of low-resolution simulations is determined (see Section 3.2.3), and this subset is run at higher resolution (shown as red circles).

Table 1. Table of simulation sets.

Simulation	Box volume	$N_{\rm part}$	$M_{\rm gas}$ ($M_\odot\ h^{-1}$)
LF	$(30\ {\rm Mpc}\ h^{-1})^3$	2×256^3	$[1.78, 2.37] \times 10^7$
HF	$(30\ {\rm Mpc}\ h^{-1})^3$	2×512^3	$[2.22, 2.96] \times 10^6$
Test	$(30\ {\rm Mpc}\ h^{-1})^3$	2×512^3	$[2.22, 2.96] \times 10^6$

will outline how we select our training simulations in the parameter space.

3.1 GP emulator

GP regression models (Rasmussen & Williams 2006) have been widely used to build cosmological emulators (Heitmann et al. 2006, 2009; Habib et al. 2007). A GP provides closed-form expressions for predictions. In addition, a GP naturally comes with uncertainty quantification, which is useful when building an inference framework. In the context of emulation, a GP can be seen as a Bayesian prior for the simulation response. It is a prior because the emulator model is chosen to ensure smoothness and monotonicity features of the simulation response *before* data are collected (Santner, Williams & Notz 2003).

Let $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$ be the input cosmologies for the simulator, where d is the dimension of the parameters ($d = 9$ for our emulator). $f(\boldsymbol{\theta})$ is the corresponding output summary statistic. In this work, the summary statistic, $f(\boldsymbol{\theta})$, is the Lyman-α forest flux power spectrum. A GP regression model can be viewed as a prior on the response surface of our simulated Lyman-α forest flux power spectrum:

$$f(\boldsymbol{\theta}) \sim \mathcal{GP}(\mu(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')), \tag{1}$$

where $\mu(\boldsymbol{\theta}) = \mathbb{E}[f(\boldsymbol{\theta})]$ is the mean function, and $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathrm{Cov}[f(\boldsymbol{\theta}), f(\boldsymbol{\theta}')]$ is the covariance kernel function. In this work, we assume a zero mean function, and we used the same covariance function as in Bird et al. (2019), which will be defined later in this section.

Suppose we run the simulations at n carefully chosen input cosmologies, $\mathcal{D} = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_n\}$, and we generate the corresponding Lyman-α forest flux power spectrum for each simulation, $\boldsymbol{y} = \{f(\boldsymbol{\theta}_1), \cdots, f(\boldsymbol{\theta}_n)\}$. Conditioning on this training data, we can get the predictive distribution of f at a new input cosmology $\boldsymbol{\theta}$ through the closed-form expression

$$f(\boldsymbol{\theta}) \mid \boldsymbol{y}, \mathcal{D} \sim \mathcal{N}(\mu_n(\boldsymbol{\theta}), \sigma_n^2(\boldsymbol{\theta})), \tag{2}$$

where the mean and variance are

$$\begin{aligned} \mu_n(\boldsymbol{\theta}) &= \boldsymbol{k}(\boldsymbol{\theta}, \mathcal{D})^\intercal \mathbf{K}(\mathcal{D})^{-1} \boldsymbol{y}; \\ \sigma_n^2(\boldsymbol{\theta}) &= k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \boldsymbol{k}(\boldsymbol{\theta}, \mathcal{D})^\intercal \mathbf{K}(\mathcal{D})^{-1} \boldsymbol{k}(\boldsymbol{\theta}, \mathcal{D}). \end{aligned} \tag{3}$$

The vector $\boldsymbol{k}(\boldsymbol{\theta}, \mathcal{D}) = [k(\boldsymbol{\theta}, \boldsymbol{\theta}_1), \cdots, k(\boldsymbol{\theta}, \boldsymbol{\theta}_n)]$ represents the covariance between the new input cosmology, $\boldsymbol{\theta}$, and the training data. The matrix $\mathbf{K}(\mathcal{D})$ is the covariance for the training data.

For the covariance kernel function, we choose the same kernel as in Bird et al. (2019), which is a combination of a linear kernel and a radial basis kernel (RBF):

$$\begin{aligned} k(\boldsymbol{\theta}, \boldsymbol{\theta}'; \sigma_0, \boldsymbol{l}, \boldsymbol{\sigma}) &= k_{\mathrm{RBF}}(\boldsymbol{\theta}, \boldsymbol{\theta}'; \sigma_0, \boldsymbol{l}) + k_{\mathrm{LIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}'; \boldsymbol{\sigma}) \\ &= \sigma_0^2 \exp\left(\sum_{i=1}^{d} -\frac{(\boldsymbol{\theta}_i - \boldsymbol{\theta}_i')^2}{2 l_i^2}\right) + \sum_{i=1}^{d} \sigma_i^2 \boldsymbol{\theta}_i \boldsymbol{\theta}_i', \end{aligned} \tag{4}$$

where σ_0^2 and $\boldsymbol{\sigma}^2$ are the variance hyperparameters for the RBF kernel and the linear kernel, respectively. $\boldsymbol{l}$ is the length-scale parameter that controls the smoothness of the GP function. We applied automatic relevance determination for both linear and RBF kernels. That is, we assign one length-scale l_i (variance σ_i) hyperparameter for each input dimension i for the RBF and linear kernels. This allows the GP to dynamically learn the scale over which each input dimension varies, which corresponds to the degree of sensitivity of the flux power spectrum to the input parameter.

Although we do not explicitly write in the notation, $f(\boldsymbol{\theta})$ is a single-valued output. Since our target summary statistic is a vector, we model each k bin of the flux power spectrum with a separate GP. The primary reason for this choice is that the correlation between the LF and HF flux power spectrum changes depending on the scale considered. The multifidelity method can only capture this scale dependence if we model each scale separately.

3.2 Multifidelity emulation

We first introduce the Kennedy–O'Hagan model (KO model; Kennedy & O'Hagan 2000) in Section 3.2.1. Section 3.2.2 describes the changes we have made to adapt the model from Ho et al. (2022) to the Lyman-α forest. Finally, the strategy we employ for choosing parameters at which to generate HF training simulations is described in Section 3.2.3.

3.2.1 KO method

The KO model (Kennedy & O'Hagan 2000) was first introduced to model a sequence of computer codes with increasing fidelity. For simplicity, we assume that there are only two fidelities: low-fidelity (LF) simulations with low resolution and high-fidelity (HF) simulations with high resolution.

We define $\{\boldsymbol{y}_{\mathrm{LF}}, \boldsymbol{y}_{\mathrm{HF}}\}$ as the Lyman-α forest flux power spectra in the training set. $\boldsymbol{y}_{\mathrm{LF}} = \{f_{\mathrm{LF}}(\boldsymbol{\theta}_i^{\mathrm{LF}})\}_{i=1}^{n_{\mathrm{LF}}}$ and $\boldsymbol{y}_{\mathrm{HF}} = \{f_{\mathrm{HF}}(\boldsymbol{\theta}_i^{\mathrm{HF}})\}_{i=1}^{n_{\mathrm{HF}}}$, where n_{LF} and n_{HF} are the number of simulations in the low- and high-fidelity training sets. We use the KO method to model Lyman-α forest flux power spectra from different fidelities:

$$f_{\mathrm{HF}}(\boldsymbol{\theta}) = \rho \cdot f_{\mathrm{LF}}(\boldsymbol{\theta}) + \delta(\boldsymbol{\theta}), \tag{5}$$

where ρ is a trainable parameter describing a multiplicative correction between the LF and HF Lyman-α forest flux power spectra. $\delta(\boldsymbol{\theta})$ is a GP independent of $f_{\mathrm{LF}}(\boldsymbol{\theta})$, describing an additive correction between fidelities. In other words, equation (5) assumes that the HF Lyman-α forest flux power can be decomposed as the LF flux power multiplied by a correction parameter, ρ, and an additive bias function $\delta(\boldsymbol{\theta})$.

As mentioned in Ho et al. (2022), the ρ parameter has to be scale-dependent (a function of k) to model the well-known fact that small scales are less well resolved in smaller simulations. Here, we use the same method as Ho et al. (2022) and assume that equation (5) is a single-output GP model. We assign a KO model to each k bin of the data.[3] In this way, we can model ρ as a function of k, as shown in Fig. 4.

We also assign KO models for each redshift. As shown in Fig. 4, ρ is a non-trivial function of both k and z, so we cannot simply use an emulator trained on one redshift to apply on another redshift.[4] We note that it is possible to assume a smooth function to model $\rho(k, z)$. However, validating $\rho(k, z)$ is out of scope for this paper. In practice, observational data are conditioned on a specific redshift, so training emulators on separate redshifts is sufficient for cosmology inference.

Fig. 4 shows that ρ stays close to unity at large scales for most of the redshifts. At small scales, however, different redshifts require different values of ρ. At the middle redshifts ($3 \leq z < 5$), ρ has a

[3]We can easily get the same set of k bins for LF and HF by using the same spectral resolution for both simulations.

[4]See Pedersen et al. (2021) for a Lyman-α forest emulator that uses a single GP for all redshifts, and achieves sub-per cent accuracy, albeit with some ambiguity between model parameters and redshift.

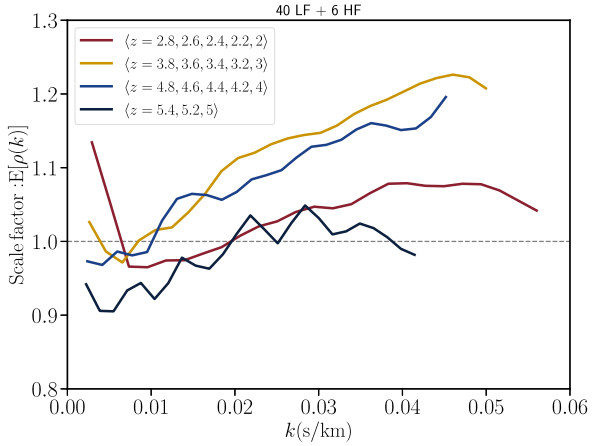

Figure 4. The scale parameter, ρ, of the KO model (equation 5) as a function of k. Different colours represent different redshifts. We separate the redshifts into four bins, $2 \leq z < 3$, $3 \leq z < 4$, $4 \leq z < 5$, and $5 \leq z \leq 5.4$. Within each redshift bin, we average over fixed k modes, which are linearly spaced between the maximum k and minimum k at the given redshift range. This emulator used 40 LF and 6 HF for training.

large positive deviation from unity. At the low redshifts ($2 \leq z < 3$), ρ has a moderate deviation towards $\rho > 1$. The only exception is at the high redshifts ($5 \leq z \leq 5.4$), which stays close to $\rho = 1$ for all scales. This indicates that the correction to the Lyman-α forest flux power due to the resolution of the simulation varies with redshift, depending on the overdensity probed by the forest.

3.2.2 *Model differences from Ho et al. (2022)*

Here, we highlight the ways in which we have adapted the model from Ho et al. (2022), which emulated the non-linear matter power spectrum at $z = 0$–2, to the Lyman-α forest at $z = 2$–5.4. Since the redshift range is larger in this work, we employed a new strategy to select the optimal HF training set by averaging over the interpolation loss for all redshift bins. We will describe the strategy in detail in Section 3.2.3.

In Ho et al. (2022), we outlined two multifidelity methods: a linear multifidelity emulator [the KO model or the autoregression model (AR1)] and a non-linear multifidelity emulator (non-linear autoregressive GP or NARGP; Perdikaris et al. 2017). However, we found that NARGP requires more HF training simulations for the Lyman-α forest flux power than AR1, perhaps due to the wide range of redshifts used. We use the KO model for our main results, and describe the NARGP results in Appendix A.

In this work, instead of emulating logarithm scaled powers, we adopted the mean-flux normalization strategy proposed in Bird et al. (2019). We normalize all flux power spectra in the training set by the median spectrum:

$$\begin{aligned} \boldsymbol{y}_{\rm LF} &\leftarrow \frac{\boldsymbol{y}_{\rm LF}}{\text{median}_i(\boldsymbol{y}_{\rm LF})} - 1; \\ \boldsymbol{y}_{\rm HF} &\leftarrow \frac{\boldsymbol{y}_{\rm HF}}{\text{median}_i(\boldsymbol{y}_{\rm LF})} - 1. \end{aligned} \tag{6}$$

The index i refers to one of the spectra in the training set, $\boldsymbol{y}_{\rm LF} = \{f_{\rm LF}(\boldsymbol{\theta}_i^{\rm LF})\}_{i=1}^{n_{\rm LF}}$. Equation (6) ensures that the training sample distribution is close to having a zero mean, matching the prior of the GP emulator. We found that in practice this normalization makes training the emulator substantially easier. Note that we normalize the HF training set using the same LF median spectrum. As the HF training set is small, the median spectrum estimate for HF is noisy, and so using it for normalization may introduce some unwanted training bias.

3.2.3 *Sampling strategy for HF simulations*

The KO model approach can be seen as a Bayesian way to correct an emulator from LF to HF. Thus, if $\boldsymbol{y}$ is the HF Lyman-α forest flux power spectrum used for training, and $\boldsymbol{\theta}$ is the corresponding input parameters:

$$\begin{aligned} \boldsymbol{y} &= f_{\rm LF}(\boldsymbol{\theta}) + (f_{\rm HF}(\boldsymbol{\theta}) - f_{\rm LF}(\boldsymbol{\theta})) \\ &= f_{\rm LF}(\boldsymbol{\theta}) + \text{error}(\boldsymbol{\theta}). \end{aligned} \tag{7}$$

The emulation accuracy will be directly affected by how well an autoregressive construction can model error($\boldsymbol{\theta}$). Usually, a large set of LF simulations are used as training data for $f_{\rm LF}(\boldsymbol{\theta})$ because they can be obtained cheaply. The quality of training data for error($\boldsymbol{\theta}$) = $f_{\rm HF}(\boldsymbol{\theta}) - f_{\rm LF}(\boldsymbol{\theta})$ thus relies on the choice of HF simulations.

In Ho et al. (2022), we proposed an optimization strategy to select HF training simulations. A low-fidelity only emulator (LFEmu)[5] is trained on a subset of LF training simulations. The posterior means of the trained LFEmu are used to calculate the emulation errors from the remaining LF samples in the Latin hypercube sampling (LHS). By minimizing the emulation errors of LFEmu, we can grid search for the optimal set of cosmologies that best interpolates the parameter space using a small number of training simulations. Assuming LFEmu is correlated with HFEmu, we can use the selected optimal cosmologies as inputs for the HF training set. By ensuring the HF training set achieves a good interpolation, we mitigate emulation errors for the multifidelity emulator.

In practice, we employ a three-stage procedure for building a multifidelity emulator:

(i) Prepare LF simulation suite.
(ii) Prepare HF simulation suite. This is done by using LFEmu to find the set of cosmologies that minimizes the interpolation loss.
(iii) Build `MFEmulator`. If the accuracy is not enough, go back to stage 1 or 2 to run more training simulations.

For stage (ii), to avoid wasting computational resources running more LF simulations, we directly use the LF simulation suite in stage (i) to build and validate the LFEmu. Thus, the cosmologies chosen for the HF set are a subset of the LF simulation LHS, which fulfils the nested training data set design suggested in Kennedy & O'Hagan (2000). The benefit of using a nested data structure, $\boldsymbol{\theta}_{\rm HF} \subseteq \boldsymbol{\theta}_{\rm LF}$, is that we can directly compute posterior means from the LF training set for cosmologies $\boldsymbol{\theta}_{\rm HF}$, without any interpolation in LF.

We note that it is possible to train a `MFEmulator` without using the LF simulations to optimize the HF points. However, if the selection of HF points is suboptimal (i.e. can barely interpolate in the prior volume), then the `MFEmulator` accuracy will be suboptimal. This is because the error(θ) cannot be decomposed into an autoregressive structure easily.

To find the optimal HF training set across the full redshift range, $z = 2$–5.4, we train an LFEmu for each redshift and get the validation loss (we used mean squared errors). We sum up the validation loss for all redshifts and find a subset of cosmologies that minimizes the summed validation loss.

In Fig. 5, we summarize the above-described procedure in a

[5]In a similar way, we call a high-fidelity only emulator, HFEmu.

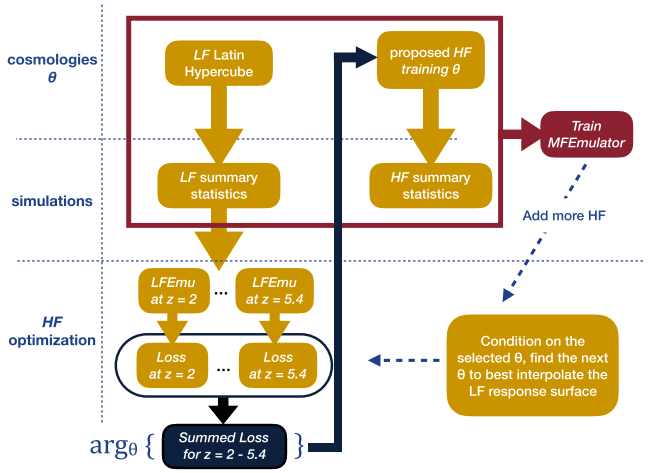

Figure 5. A flowchart for training a multifidelity emulator. We start with an LF set of simulations. We then select a subset of LF points to train an emulator, LFEmu, at each redshift. The validation loss for each LFEmu is computed using the rest of the LF simulations as a validation set. Finally, we sum up the validation losses at each redshift, and use the summed loss to propose a set of cosmologies $\boldsymbol{\theta}_{\rm HF}$ that can best minimize the summed loss.

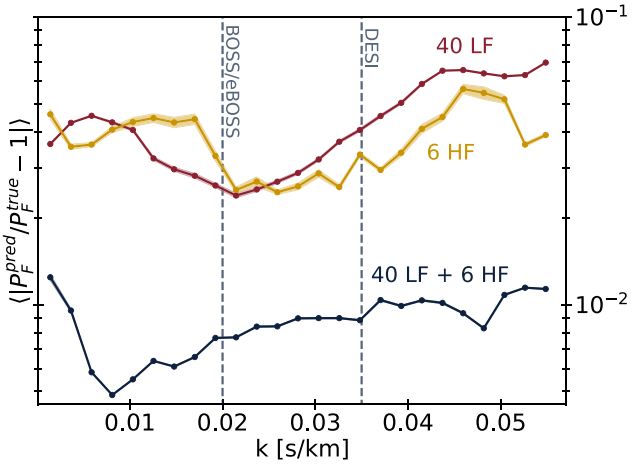

Figure 6. Comparing the prediction error as a function of (linearly binned) wavevector for single- and multifidelity emulators. This is the mean error across all redshifts and 10 test simulations. The shaded regions are the variance in the prediction error. Dashed lines show the highest k probed by BOSS/eBOSS (Chabanier et al. 2019) and the estimated reach for DESI (Valluri et al. 2022).

flowchart. For a formal description, we refer to Ho et al. (2022). The only difference between the proposed procedure in Fig. 5 and the procedure in Ho et al. (2022) is that we optimize $\boldsymbol{\theta}_{\rm HF}$ across redshifts $z = 2$–5.4 in this work. Thus, we have an additional step to sum the LFEmu validation loss across redshifts.

Training an LFEmu on all possible LF subsets is computationally intensive. To reduce costs, we employed the greedy optimization strategy from Ho et al. (2022). We first explored all possible subsets for three design points within the LF LHS. For the optimal four design points, instead of exploring all possible subsets, we grew the subset one point at a time, fixing the previously chosen optimal three HF points. In the same line of thought, we grew the subset to six optimal design points for HF training cosmologies. Our final simulation suite of 40 LF and 6 HF samples, along with parameter limits, is shown in Fig. 3.

4 RESULTS AND DISCUSSION

Using the flux power spectra from our LF and HF simulations, we train single-fidelity (one LF only and one HF only) and multifidelity emulators. These trained emulators are used to predict the flux power spectrum output for a set of 10 simulation input parameters. We then compare these predictions to the corresponding testing simulations that were run at the same resolution as the HF simulations (see Table 1).

4.1 Emulator accuracy

In the following, we only show results for the emulators that use the full available set of training simulations (40 LF and 6 HF). We have verified that using all available training simulations leads to the most accurate emulator. Section 4.2 shows how emulator accuracy degrades when a smaller subset of the available simulations is used.

Using the full set of available simulations, the mean prediction error for the multifidelity emulator is $\langle |P_F^{\rm pred}/P_F^{\rm true} - 1| \rangle \approx 0.8$ per cent (averaging across all scales, redshifts, and testing simulation outputs). For the LF single-fidelity emulator, the mean prediction error is ≈ 4 per cent. This is not unexpected; there are real differences between the flux power spectra output by the low- and high-resolution simulations that are not being captured with this method. The 4 per cent error may seem quantitatively quite good, considering the simpler methodology and reduced resource cost. However, there is no indication that the error could be reduced further with additional simulations (see Section 4.2).

For the HF single-fidelity emulator, the mean prediction error is ≈ 3 per cent. This is likely limited by the sample size of the training set (six simulations), leading to increased errors when making predictions for inputs that are far away from the training samples. It is important to note that the HF samples are selected to optimize the multifidelity emulator, rather than as an independent emulator (i.e. as a Latin hypercube sample). There is some indication that prior information from the LF training samples provides useful information about the best areas of parameter space to sample. To test this, we split our testing set (10 simulations, same resolution as the HF samples) into training and testing sets, then train all 210 combinations of 6 samples, and predict the outputs for the remaining 4 samples. The error range from this exercise is 2.5–11.5 per cent (5.5 per cent mean error, 1.5 per cent standard deviation). Though not a direct comparison, the 3 per cent error we obtain from the HF single-fidelity emulator compares favourably with this, indicating that the HF samples selected are an improvement over using an LHS scheme.

Fig. 6 shows the mean prediction error, averaged over all redshifts and 10 test simulation outputs, as a function of wavevector k. In this, and the following figures, the shaded region around the curves is the variance in prediction error ($|P_F^{\rm pred}/P_F^{\rm true} - 1|$), to give a sense of how much the error varies beyond the mean. The multifidelity emulator outperforms the single-fidelity emulators at all scales, with an error between 0.5 and 1.5 per cent. The LF (HF) single-fidelity emulator has error between 2 and 7 per cent (2–6 per cent).

Both single-fidelity emulators and the multifidelity emulator trend towards higher error for small scales. The LF emulator dips 1–2 per cent around $k \approx 0.02$ s km^{-1}. The dip occurs on scales at which the low-resolution flux power spectra go from overestimating to underestimating the high-resolution power (see Fig. 2, for $z \leq 4.6$). The uptick in the multifidelity emulator error for the largest

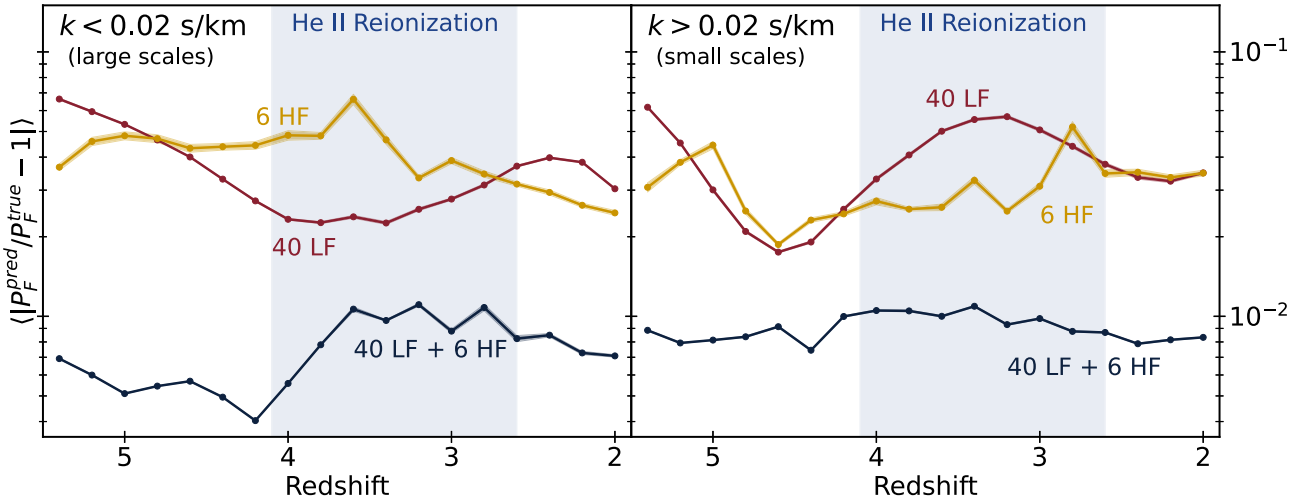

Figure 7. Comparing the prediction error as a function of redshift and scale for single- and multifidelity emulators. This is mean error across all 10 test simulations. The blue shaded region shows the extent of helium reionization in our simulations (see parameter limits in Fig. 3).

k bins is also present in the HF emulator, indicating that there is a scarcity of large-scale modes available in the emulator training.

Fig. 7 shows how the emulators perform as a function of redshift and scale. In the following, we define small scales as $k > 0.02$ s km^{-1} and large scales as $k \leq 0.02$ s km^{-1} (divided at the smallest scale accessible by BOSS/eBOSS data; Chabanier et al. 2019). On large scales (Fig. 7, left-hand panel), the LF emulator error decreases with time until $z \approx 4$, and then slowly increases. This trend is because, as can be seen in Fig. 2 (left of the BOSS/eBOSS dashed line), the low-resolution flux power comes into better agreement with the high-resolution flux power as it nears $z \approx 4$. We also found that the LF simulations do not cool as efficiently as the HF simulations after He II reionization, likely leading to the rise in error for $z \leq 3.2$.

On small scales (Fig. 7, right-hand panel), the LF emulator error is more variable. The dip in error seen around $z \approx 4.6$ occurs as the low-resolution flux power crosses from overestimating to underestimating the high-resolution flux power (Fig. 2, right of the BOSS/eBOSS dashed line). The subsequent rise in error is due to the loss of small-scale power and consequent underestimation of the flux power spectrum in the LF simulations.

The trends in redshift and scale seen in the LF emulator performance are not due to interpolation error, but due to the different numerical resolution of the two simulation fidelities, since this emulator is not predicting the flux power at the higher resolution. Some differences are connected to temperature differences between the LF and HF simulations. For low densities (∼1 times the mean density), the LF simulations are colder than the HF simulations at high redshift, but come into better agreement leading up to He II reionization, after which they diverge from the HF simulations again. For higher densities (1–100 times the mean density), the LF simulations are once again colder than the HF simulations at high redshift, but at lower redshift they are too hot (with a crossover at $z \approx 4.6$). As higher redshifts probe lower densities, the error initially decreases with redshift, before rising again towards the lowest redshifts.

On both large and small scales, the HF emulator errors are around 3.5 per cent, dominated by sampling variance. During He II reionization, the HF emulator has more variation in error, which is probably exacerbated by our small box sizes. The increased variation during He II reionization further indicates that the primary source of error for the HF emulator is the sample size of the training set.

On small scales, the multifidelity emulator error is insensitive to redshift and small (0.9 per cent). On large scales, the multifidelity emulator error slightly decreases until $z = 4.2$, and then increases with the onset of He II reionization, before flattening again. The trend is also more variable during He II reionization, indicating that emulator finds it more difficult to learn the mapping during this process. However, the multifidelity emulator still outperforms the single-fidelity emulators, with an error between ≈ 0.4 and 1 per cent.

4.2 Emulator runtime

While we have shown that the multifidelity emulator outperforms the single-fidelity emulators presented here, it still remains to show that it is more computationally cost efficient. We could, for example, add more training simulations to our single-fidelity HF emulator and get a similarly accurate high-resolution emulator. However, the computational cost would increase significantly. By comparing the total emulator runtime to prediction error, we can determine the choice that balances computational cost and accuracy. In practice, the important question is to determine the computational cost at which a given emulation technique can achieve a desired accuracy. The computational cost of training the emulators is subdominant ($\mathcal{O}(1)$ cpu-hours) to running the training simulations, so in the following, we only consider the runtime for the simulation suites.

Fig. 8 shows the mean prediction error (averaged over all redshifts and test outputs) as a function of the number of simulations used in the training set, for small and large scales (as defined in Section 4 and Fig. 7). The solid lines show prediction errors for multifidelity emulators trained using six HF and a varying number of LF simulations. The small- and large-scale errors flatten out after ≈30 LF simulations are used in the training set. The LF simulations allow the emulator to determine how the flux power spectrum depends on the cosmological input parameters, and so this indicates that 30 LF simulations are needed to explore our nine-parameter space. Other emulators range from using ≈6 simulations per parameter (e.g. McClintock et al. 2019; Ho et al. 2022) to 30 per parameter (e.g. Euclid Collaboration et al. 2021). The three to four simulations per parameter required here is unusually low, perhaps because the input parameters affect the flux power spectrum close to linearly in much of parameter space.

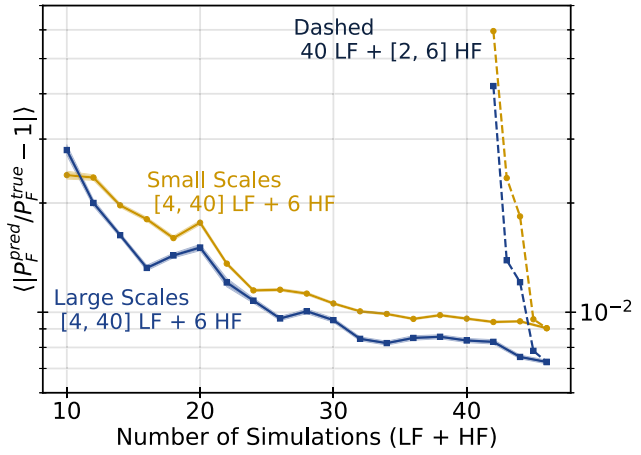

Figure 8. Emulator prediction error as a function of the number of simulations used in training the emulator. This is the mean error across all redshifts and 10 test simulation outputs. The prediction error is broken down into large ($k < 0.02\,\mathrm{s\,km^{-1}}$) and small ($k > 0.02\,\mathrm{s\,km^{-1}}$) scales, as in Fig. 7. The solid lines show how the average error depends on the number of LF simulations, while the dashed lines show how the average error depends on the number of HF simulations once all LF simulations are included.

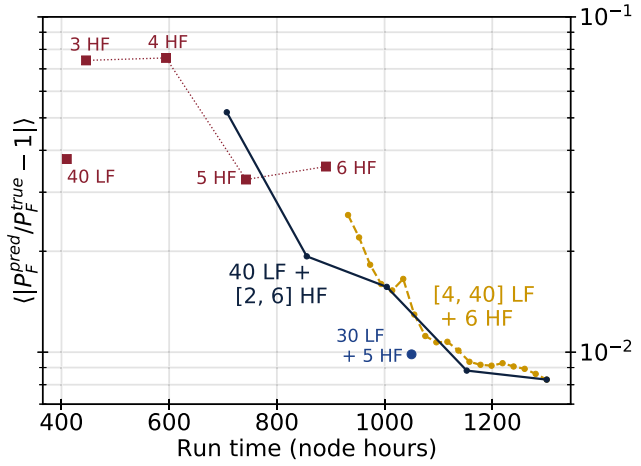

Figure 9. Prediction error as a function of total training simulation computational cost. This is the mean error for 10 test simulations over all redshifts and scales. The solid line shows the prediction error when changing the number of HF simulations used in training the multifidelity emulator. Also shown are the single-fidelity emulators (red squares) and the multifidelity emulator trend with a varying number of LF training simulations (yellow, dashed).

Dashed lines show prediction errors for multifidelity emulators trained using 40 LF and a varying number of HF simulations. Adding extra HF simulations to the training set has a larger impact than adding LF simulations. The addition of each HF simulation generally improves the emulator accuracy for small scales more than for large scales. This is as expected, since the main purpose of the HF simulations is to learn the mapping from low- to high-resolution output, with small scales being more resolved in the HF simulations.

Fig. 9 shows the emulator prediction errors as a function of the total runtime (cost of running the training simulations). All simulations were run on the Frontera supercomputer at the Texas Advanced Computing Center. The cost is divided between the LF training simulations, which cost ≈10 node hours each, and the HF training simulations, which cost ≈150 node hours each.

The dashed trend shows the same emulators as Fig. 8, but no longer divided by scale. Qualitatively, it looks the same as both the large-scale and small-scale results from the previous figure. The error is flat after ≈30 LF training simulations, indicating that a similar accuracy can be achieved using the multifidelity emulator with ≈30 rather than 40 LF simulations. The most efficient 1 per cent error emulator in this study is a multifidelity emulator using 30 LF and 5 HF training simulations (the cost for this was ≈1050 node hours). The most accurate emulator is the 40 LF, 6 HF multifidelity emulator, with error 0.8 per cent and cost ≈1300 node hours.

The dotted line (squares) shows the error and runtime for the single-fidelity emulators. Following from the six HF single-fidelity emulator result to the dashed (yellow) line, it can be seen that the addition of just a few LF training simulations quickly improves the accuracy. It can also be seen that in terms of computational cost, the multifidelity emulator is more efficient.[6] Note that the HF training simulations are not selected to optimize a single-fidelity emulator, but instead are selected to optimize the multifidelity emulator. They thus use prior information provided by the LF training simulations and so perform better than a naive Latin hypercube construction of an HF emulator using six training samples. Our multifidelity scheme is thus an even larger improvement on a single-fidelity model than Fig. 9 suggests.

The solid line shows the error and runtime for the multifidelity emulator trained using 40 LF simulations and 2–6 HF simulations. The point on the solid line corresponding to 40 LF, 2 HF has a similar cost, but slightly worse performance than the 5 HF single-fidelity result. Adding a third HF training sample decreases the error more for the multifidelity emulator (error for 40 LF, 3 HF emulator) than it does for the single-fidelity emulator (error marked 6 HF). Adding a sixth HF simulation to the 40 LF, 5 HF multifidelity emulator produces a relatively small improvement in error, perhaps indicating that stochasticity in the simulations due to our relatively small box size is beginning to dominate interpolation error.

5 CONCLUSIONS

In this work, we developed and tested a multifidelity emulator for the simulated Lyman-α forest flux power spectrum. Emulators address the growing computational demands of simulations, which must be run at increasingly high resolutions to allow analysis of the increasing quality and quantity of observational data. Here, we use a GP-based emulator that addresses this demand by, in a Bayesian framework, training an interpolating function to predict the output (Lyman-α forest flux power spectrum) for a given input (simulation input parameters). Relatively few simulations are required to accurately predict across the span of input parameter space, making emulators especially useful for parameter inference problems.

The multifidelity framework allows a further reduction in computational cost by dividing the emulator training samples into multiple (in our case two) fidelities. The LF (low-resolution) training samples allow the emulator to learn how the outputs depend on input parameters. The HF (high-resolution) training samples correct numerical errors in the LF emulator with a (parameter-dependent) mapping from LF to HF. Thus, the emulator can be trained with a large sample of LF training simulations and a small subset of HF training simulations.

[6] At least for errors less than 4 per cent, the approximate amount by which the LF simulations fail to be converged.

Our training suite included 40 low-resolution hydrodynamical simulations (30 Mpc h^{-1} simulation box length, 256^3 particles) and 6 high-resolution hydrodynamical simulations (30 Mpc h^{-1} simulation box length, 512^3 particles). Using the Lyman-α forest flux power spectrum extracted from these simulations, we trained single- and multifidelity emulators to predict the high-resolution flux power spectrum. 10 independent simulations were run to test the prediction accuracy of the trained emulators.

In summary, the multifidelity emulator:

(i) Modelled a redshift range 5.4–2 (in 18 redshift bins with $\Delta z = 0.2$), on scales ranging from $k = 1.4 \times 10^{-3}$ to 5.7×10^{-2} s km^{-1} (in 25 bins).

(ii) Achieved sub-1 per cent error on most scales and redshifts when averaged over 10 test simulations.

(iii) Reached an average error of 0.8 per cent.

(iv) Achieved 1 per cent average error most cost efficiently using a training set with 30 low-resolution and 5 high-resolution simulations.

The low-resolution single-fidelity emulator (4 per cent average error) predicts the low-resolution flux power, so it is limited by real differences between the output of the two resolutions. The high-resolution single-fidelity emulator (3 per cent average error) is limited by the small number of training samples. It is likely that the average error for the high-resolution single-fidelity emulator could be improved to match the multifidelity emulator performance with the addition of more training simulations. However, the high-resolution single-fidelity emulator quickly increases in computational cost with additional samples, and we expect it would thus be more expensive than our multifidelity emulator.

Some important caveats to our results are that the Lyman-α forest is converged at the ≈ 5 per cent level in our high-resolution simulations, and the box size is small. In a forthcoming work, a model that uses two different box sizes (rather than two different resolutions) to construct a multifidelity emulator will be developed and tested. While there is no direct evidence to suggest that changing the resolution or box size would significantly enhance or diminish the accuracy of the emulators presented here, it still remains to be tested on simulations with higher resolution and larger box sizes. In a forthcoming work, we test the multifidelity framework on larger box size, higher resolution simulations, and use this multifidelity emulator for cosmological inference.

ACKNOWLEDGEMENTS

MAF is supported by a National Science Foundation Graduate Research Fellowship under grant no. DGE-1326120. MFH is supported by a National Aeronautics and Space Administration FINESST under grant no. ASTRO20-0022. SB is supported by NSF grant AST-1817256.

Computing resources were provided by Frontera LRAC AST21005. The authors acknowledge the Frontera computing project at the Texas Advanced Computing Center (TACC) for providing HPC and storage resources that have contributed to the research results reported within this paper. Frontera is made possible by National Science Foundation award OAC-1818253. URL: http://www.tacc.utexas.edu

DATA AVAILABILITY

Flux power spectra generated from the low resolution, high resolution, and testing sets are available at https://github.com/mafern/MFEmulatorLyaData. HDF5 and plain text (appropriate for multifidelity emulation) formats are available. Select single- and multifidelity emulator predictions for the 10 testing simulations are also available from the same repository. The spectra underlying the flux power are available upon request.

REFERENCES

Aricò G., Angulo R. E., Contreras S., Ondaro-Mallea L., Pellejero-Ibañez M., Zennaro M., 2021, MNRAS, 506, 4070
Bevins H. T. J., Handley W. J., Fialkov A., de Lera Acedo E., Javid K., 2021, MNRAS, 508, 2923
Bird S., 2017, Astrophysics Source Code Library, record ascl:1710.012
Bird S., Vogelsberger M., Sijacki D., Zaldarriaga M., Springel V., Hernquist L., 2013, MNRAS, 429, 3341
Bird S., Haehnelt M., Neeleman M., Genel S., Vogelsberger M., Hernquist L., 2015, MNRAS, 447, 1834
Bird S., Rogers K. K., Peiris H. V., Verde L., Font-Ribera A., Pontzen A., 2019, J. Cosmol. Astropart. Phys., 2019, 050
Bird S., Feng Y., Pedersen C., Font-Ribera A., 2020, J. Cosmol. Astropart. Phys., 2020, 002
Bird S., Ni Y., Di Matteo T., Croft R., Feng Y., Chen N., 2022, MNRAS, 512, 3703
Bocquet S., Heitmann K., Habib S., Lawrence E., Uram T., Frontiere N., Pope A., Finkel H., 2020, ApJ, 901, 5
Boera E., Becker G. D., Bolton J. S., Nasir F., 2019, ApJ, 872, 101
Bolton J. S., Becker G. D., 2009, MNRAS, 398, L26
Bolton J. S., Viel M., Kim T. S., Haehnelt M. G., Carswell R. F., 2008, MNRAS, 386, 1131
Bolton J. S., Becker G. D., Haehnelt M. G., Viel M., 2014, MNRAS, 438, 2499
Borde A., Palanque-Delabrouille N., Rossi G., Viel M., Bolton J. S., Yèche C., LeGoff J.-M., Rich J., 2014, J. Cosmol. Astropart. Phys., 2014, 005
Bye C. H., Portillo S. K. N., Fialkov A., 2022, ApJ, 930, 79
Chabanier S. et al., 2019, J. Cosmol. Astropart. Phys., 2019, 017
Chabanier S., Bournaud F., Dubois Y., Palanque-Delabrouille N., Yèche C., Armengaud E., Peirani S., Beckmann R., 2020, MNRAS, 495, 1825
Cohen A., Fialkov A., Barkana R., Monsalve R. A., 2020, MNRAS, 495, 4845
Croft R. A. C., Weinberg D. H., Katz N., Hernquist L., 1998, ApJ, 495, 44
D'Aloisio A., McQuinn M., Maupin O., Davies F. B., Trac H., Fuller S., Upton Sanderbeck P. R., 2019, ApJ, 874, 154
Damianou A., Lawrence N. D., 2013, in Carvalho C. M., Ravikumar P., eds, Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, vol. 31. Deep Gaussian Processes. PMLR, Scottsdale, AZ, p. 207
Davies C. T., Cautun M., Giblin B., Li B., Harnois-Déraps J., Cai Y.-C., 2021, MNRAS, 507, 2267
Day A., Tytler D., Kambalur B., 2019, MNRAS, 489, 2536
DESI Collaboration et al., 2016, preprint (arXiv:1611.00036)
Euclid Collaboration et al., 2021, MNRAS, 505, 2840
Fan X. et al., 2006, AJ, 132, 117
Faucher-Giguère C.-A., 2020, MNRAS, 493, 1614
Feng Y., Di-Matteo T., Croft R. A., Bird S., Battaglia N., Wilkins S., 2016, MNRAS, 455, 2778
Fernandez M. A., Bird S., Upton Sanderbeck P., 2021, MNRAS, 503, 1668
Gaikwad P., Srianand R., Haehnelt M. G., Choudhury T. R., 2021, MNRAS, 506, 4389
Garny M., Konstandin T., Sagunski L., Viel M., 2021, J. Cosmol. Astropart. Phys., 2021, 049
Garzilli A., Magalich A., Ruchayskiy O., Boyarsky A., 2021, MNRAS, 502, 2356
Giblin B., Cataneo M., Moews B., Heymans C., 2019, MNRAS, 490, 4826
Giri S. K., Schneider A., 2021, J. Cosmol. Astropart. Phys., 2021, 046
Gunn J. E., Peterson B. A., 1965, ApJ, 142, 1633
Habib S., Heitmann K., Higdon D., Nakhleh C., Williams B., 2007, Phys. Rev. D, 76, 083503
Harnois-Déraps J., Giblin B., Joachimi B., 2019, A&A, 631, A160

Heitmann K., Higdon D., Nakhleh C., Habib S., 2006, ApJ, 646, L1
Heitmann K., Higdon D., White M., Habib S., Williams B. J., Lawrence E., Wagner C., 2009, ApJ, 705, 156
Heitmann K., Lawrence E., Kwan J., Habib S., Higdon D., 2014, ApJ, 780, 111
Ho M.-F., Bird S., Garnett R., 2021, MNRAS, 507, 704
Ho M.-F., Bird S., Shelton C. R., 2022, MNRAS, 509, 2551
Hui L., Burles S., Seljak U., Rutledge R. E., Magnier E., Tytler D., 2001, ApJ, 552, 15
Iršič V. et al., 2017a, Phys. Rev. D, 96, 023522
Iršič V. et al., 2017b, MNRAS, 466, 4332
Karaçaylı N. G. et al., 2022, MNRAS, 509, 2842
Kennedy M., O'Hagan A., 2000, Biometrika, 87, 1
Kern N. S., Liu A., Parsons A. R., Mesinger A., Greig B., 2017, ApJ, 848, 23
Lawrence E. et al., 2017, ApJ, 847, 50
Lesgourgues J., 2011, preprint (arXiv:1104.2932)
Lukić Z., Stark C. W., Nugent P., White M., Meiksin A. A., Almgren A., 2015, MNRAS, 446, 3697
McClintock T. et al., 2019, ApJ, 872, 53
McDonald P., Miralda-Escudé J., Rauch M., Sargent W. L. W., Barlow T. A., Cen R., Ostriker J. P., 2000, ApJ, 543, 1
McDonald P. et al., 2005, ApJ, 635, 761
McDonald P. et al., 2006, ApJS, 163, 80
McQuinn M., Lidz A., Zaldarriaga M., Hernquist L., Hopkins P. F., Dutta S., Faucher-Giguère C.-A., 2009, ApJ, 694, 842
Nasir F., Bolton J. S., Becker G. D., 2016, MNRAS, 463, 2335
Ni Y. et al., 2022, MNRAS, 513, 670
Nishimichi T. et al., 2019, ApJ, 884, 29
Okamoto T., Frenk C. S., Jenkins A., Theuns T., 2010, MNRAS, 406, 208
Palanque-Delabrouille N., Yèche C., Schöneberg N., Lesgourgues J., Walther M., Chabanier S., Armengaud E., 2020, J. Cosmol. Astropart. Phys., 2020, 038
Pedersen C., Font-Ribera A., Rogers K. K., McDonald P., Peiris H. V., Pontzen A., Slosar A., 2021, J. Cosmol. Astropart. Phys., 2021, 033
Perdikaris P., Raissi M., Damianou A., Lawrence N. D., Karniadakis G. E., 2017, Proc. R. Soc. A, 473, 20160751
Rasmussen C. E., Williams C. K. I., 2006, Gaussian Processes for Machine Learning. MIT Press, Cambridge, Massachusetts
Rogers K. K., Peiris H. V., 2021a, Phys. Rev. D, 103, 043526
Rogers K. K., Peiris H. V., 2021b, Phys. Rev. Lett., 126, 071302
Rogers K. K., Peiris H. V., Pontzen A., Bird S., Verde L., Font-Ribera A., 2019, J. Cosmol. Astropart. Phys., 2019, 031
Santner T. J., Williams B. J., Notz W. I., 2003, Springer Series in Statistics, The Design and Analysis of Computer Experiments. Springer, New York, NY
Seljak U. et al., 2005, Phys. Rev. D, 71, 103515
Seljak U., Slosar A., McDonald P., 2006, J. Cosmol. Astropart. Phys., 2006, 014
Springel V., 2005, MNRAS, 364, 1105
Springel V., Hernquist L., 2003, MNRAS, 339, 289
Theuns T., Leonard A., Efstathiou G., Pearce F. R., Thomas P. A., 1998, MNRAS, 301, 478
Upton Sanderbeck P., Bird S., 2020, MNRAS, 496, 4372
Valluri M. et al., 2022, preprint (arXiv:2203.07491)
Viel M., Haehnelt M. G., 2006, MNRAS, 365, 231
Viel M., Matarrese S., Mo H. J., Haehnelt M. G., Theuns T., 2002, MNRAS, 329, 848
Viel M., Haehnelt M. G., Springel V., 2004, MNRAS, 354, 684
Viel M., Lesgourgues J., Haehnelt M. G., Matarrese S., Riotto A., 2005, Phys. Rev. D, 71, 063534
Viel M., Haehnelt M. G., Lewis A., 2006, MNRAS, 370, L51
Viel M., Becker G. D., Bolton J. S., Haehnelt M. G., 2013a, Phys. Rev. D, 88, 043502
Viel M., Schaye J., Booth C. M., 2013b, MNRAS, 429, 1734
Villasenor B., Robertson B., Madau P., Schneider E., 2022, APJ,933, 59
Walther M., Armengaud E., Ravoux C., Palanque-Delabrouille N., Yèche C., Lukić Z., 2021, J. Cosmol. Astropart. Phys., 2021, 059
Wu X., McQuinn M., Kannan R., D'Aloisio A., Bird S., Marinacci F., Davé R., Hernquist L., 2019, MNRAS, 490, 3177

APPENDIX A: NON-LINEAR MULTIFIDELITY EMULATOR

In the main text, we have explored the effectiveness of a linear multifidelity emulator (the KO model, or AR1). In the linear model, the mapping from LF to HF is $f_{\rm HF}(\boldsymbol{\theta}) = \rho\, f_{\rm LF}(\boldsymbol{\theta}) + \delta(\boldsymbol{\theta})$, where $f_{\rm HF}$ and $f_{\rm LF}$ are the emulator predictions at those resolutions, and ρ is independent of the input parameters $\boldsymbol{\theta}$.

Here, we compare those results with the results using a non-linear multifidelity emulator (NARGP). In the non-linear multifidelity model, proposed by Perdikaris et al. (2017), the mapping is a function of both the LF output and the input parameters. We model this as

$$f_{\rm HF}(\boldsymbol{\theta}) = \rho(\boldsymbol{\theta},\, \tilde{f}_{\rm LF}(\boldsymbol{\theta})) + \delta(\boldsymbol{\theta}),$$

such that ρ depends on both the input parameters and LF posterior output. The LF outputs, as is the case with the linear model, are median normalized such that the assumption on the GP of zero mean is more reasonable, $\tilde{f}_{\rm LF}(\boldsymbol{\theta}) = f_{\rm LF}(\boldsymbol{\theta})/\mu_{\rm LF} - 1$.

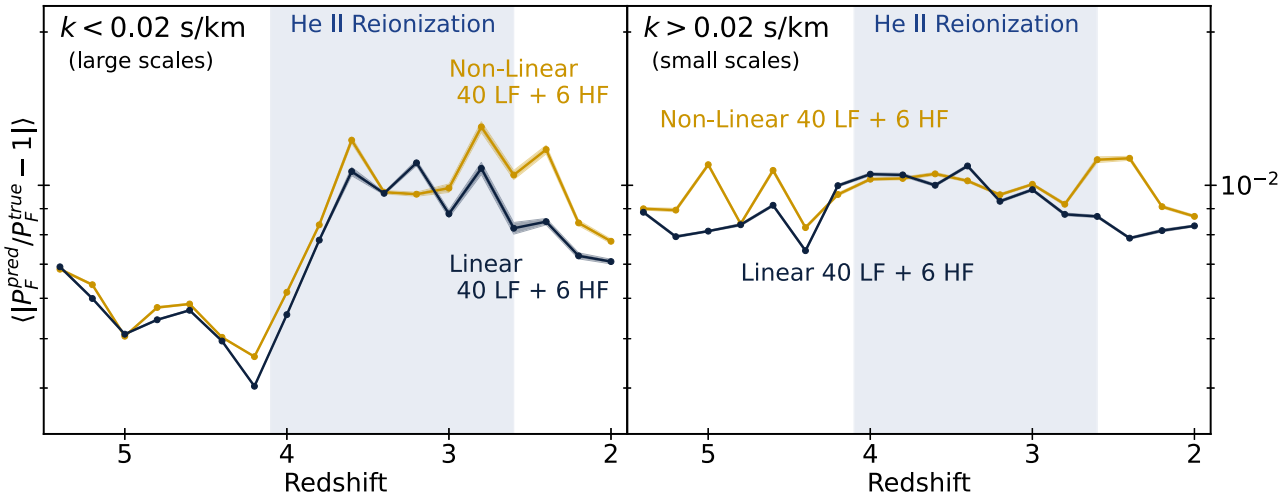

Figure A1. Comparing the prediction error as a function of redshift and scale for linear and non-linear multifidelity emulators. This is mean error across all 10 test simulations. The shaded region shows the extent of helium reionization in our simulations (see parameter limits in Fig. 3).

Following Perdikaris et al. (2017), ρ is modelled as a GP with input from both input cosmologies for HF, $\boldsymbol{\theta}_{\rm HF}$, and the output from LF, $\tilde{f}_{\rm LF}(\boldsymbol{\theta})$. The NARGP construction results in a deep GP model (Damianou & Lawrence 2013). We follow the approximation in Perdikaris et al. (2017) and replace $\tilde{f}_{\rm LF}(\boldsymbol{\theta})$ with its posterior distribution. Thus, the training reduces to training two regular GPs recursively.

In Fig. A1, we show the prediction errors separated into small and large scales, as a function of redshift for both the linear and non-linear multifidelity emulators. They perform similarly, with the linear emulator being more accurate at low redshifts on all scales, and high redshifts for small scales. The difference between the average error for the linear and non-linear models (over all scales and redshifts) is 0.08 per cent. This is in contrast to emulation of the matter power spectrum in Ho et al. (2022), where the non-linear model outperformed the linear model.

It is worth noting that the non-linear model agrees closely with the linear model when using the full suite of training simulations, but lags behind the linear model when using fewer HF training simulations. For example, the difference in the average error between linear and non-linear models using 40 LF and 3 HF is ≈ 2 per cent (1.9 per cent error for linear, 3.7 per cent error for non-linear). When using four HF, the difference is ≈ 1 per cent (1.5 per cent, 2.5 per cent), and when using five HF the difference is ≈ 0.4 per cent (0.9 per cent, 1.3 per cent). While differences in the effectiveness of the non-linear model may be due to the quantity being emulated (matter power versus flux power), one likely reason for the difference is the number of input parameters. In Ho et al. (2022), five input parameters are used, while in this work we use nine. The non-linear model uses the posterior of the LF output, which requires Monte Carlo sampling. It is possible that the additional dimensions degrade the performance of the Monte Carlo integration, and thus the performance of the non-linear model. One other reason may be the larger number of hyperparameters that need to be optimized in the training.

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