Draft version January 11, 2021
Typeset using LaTeX **twocolumn** style in AASTeX62

Cosmic Inference: Constraining Parameters With Observations and a Highly Limited Number of Simulations

Timur Takhtaganov,[1] Zarija Lukić,[1] Juliane Müller,[1] and Dmitriy Morozov[1]

[1] *Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

Submitted to the Astrophysical Journal

ABSTRACT

Cosmological probes pose an inverse problem where the measurement result is obtained through observations, and the objective is to infer values of model parameters that characterize the underlying physical system — our universe, from these observations and theoretical forward-modeling. The only way to accurately forward-model physical behavior on small scales is via expensive numerical simulations, which are further "emulated" due to their high cost. Emulators are commonly built with a set of simulations covering the parameter space with Latin hypercube sampling and an interpolation procedure; the aim is to establish an approximately constant prediction error across the hypercube. In this paper, we provide a description of a novel statistical framework for obtaining accurate parameter constraints. The proposed framework uses multi-output Gaussian process emulators that are adaptively constructed using Bayesian optimization methods with the goal of maintaining a low emulation error in the region of the hypercube preferred by the observational data. In this paper, we compare several approaches for constructing multi-output emulators that enable us to take possible inter-output correlations into account while maintaining the efficiency needed for inference. Using a Lyα forest flux power spectrum, we demonstrate that our adaptive approach requires considerably fewer — by a factor of a few in the Lyα $P(k)$ case considered here — simulations compared to the emulation based on Latin hypercube sampling, and that the method is more robust in reconstructing parameters and their Bayesian credible intervals.

Keywords: Cosmological parameters; Intergalactic medium; Computational methods

1. INTRODUCTION

The field of cosmology has rapidly progressed in the past few decades, going from a largely qualitative picture of the hot Big-Bang to the now well-tested standard model of cosmology. This relatively simple model describes current observations at a few-percent level using only six parameters (Planck Collaboration et al. 2018). While this has been a great success — driven by a deluge of observations — questions still remain about the nature of dark matter and dark energy, primordial fluctuations relating to the inflation in the early universe, and the mass of neutrino particles. To make further progress in answering these questions, new ground- and space-based observational missions will be carried out, probing the highly nonlinear scales of cosmic structure. Planned wide-field sky surveys such as the Dark Energy Spectroscopic Instrument (DESI Collaboration et al. 2016), the Large Synoptic Survey Telescope (LSST Dark Energy Science Collaboration 2012), the Wide Field Infrared Survey Telescope (WFIRST, Spergel et al. 2015), and Euclid (Refregier et al. 2010) will provide precision measurements of cosmological statistics such as weak-lensing shear correlations, cluster abundance, and the distribution of galaxies, quasars and Lyα absorption lines. Inferring values of the physical model parameters using observations of the mentioned sky surveys is a problem that belongs to the class of inverse problems in statistics.

The application of Markov chain Monte Carlo (MCMC, Metropolis et al. (1953); Gelman et al. (2013)) or similar Bayesian methods requires hundreds of thousands to even millions of forward-model evaluations in order to determine the posterior probabilities of the considered parameters. When modeling the highly nonlinear regime of the structure formation in the universe, each such evaluation is a high-performance computing simulation costing more than 10^5 CPU hours. Most perturbation theory methods break down at about the scale of $x \lesssim 65\,h^{-1}$Mpc (Carlson et al. 2009). The most recent works have pushed this scale down to $\sim 10\,h^{-1}$Mpc (d'Amico et al. 2020; Ivanov et al. 2020; Chen et al. 2020), but this is still between one and two orders of

Corresponding author: Zarija Lukić
zarija@lbl.gov

magnitude larger than what is needed for Lyα studies. Cosmological simulations that from first principles numerically evolve the density field are therefore essential for the analysis and scientific inference of the future observational data sets.

While it may seem at first that this cost makes the inference computationally unfeasible, it is in fact possible to efficiently sample the parameter space with a dramatically reduced number of simulations, provided that certain smoothness conditions are satisfied. This is achieved through the development of cosmological emulators, that is, computationally cheap surrogate models of expensive cosmological simulations. The pioneering work on these techniques in cosmology was the cosmic calibration effort (Heitmann et al. 2006; Habib et al. 2007), resulting in a 1% accurate matter power spectrum emulator (Heitmann et al. 2010). Later works have expanded the range of validity of this nonlinear matter power spectrum in terms of the k and redshift coverage, and they also increased the number of cosmological parameters (Lawrence et al. 2017; Euclid Collaboration et al. 2019). In addition, this emulation technology proliferated into modeling many other statistics and observables, including gravitational lensing quantities (Liu et al. 2015; Petri et al. 2015; Wibking et al. 2020), the galaxy halo occupation model (Kwan et al. 2015), the halo mass function (McClintock et al. 2018), the galaxy correlation function (Zhai et al. 2018), the 1D Lyα flux power spectrum (Walther et al. 2019), and the 21cm power spectrum (Jennings et al. 2019). More generally, outside the field of astrophysics, there is a large body of work on emulators and designs for large-scale computer experiments, see, e.g. Haaland & Qian (2011) and Kleijnen (2015).

In this work we are not concerned with building an emulator for the cosmological simulation models that is accurate over the entire prior range of the input parameter values. We instead focus on constructing an emulator in an adaptive fashion by preferentially selecting the inputs for the simulation that are more likely to result in the values of the output that are consistent with the observational data. By building up our emulator in this sequential way, we strive to avoid performing unnecessary simulations that would be needed to have a globally accurate surrogate. To find an optimal point in parameter space for running the simulation, we use Bayesian optimization techniques (Mockus 1994; Kennedy & O'Hagan 2001; Mockus et al. 2014; Leclercq 2018), specifically developed to efficiently determine the global optima of functions (we also refer to Shahriari et al. (2016) or Frazier (2018) for recent pedagogical surveys of Bayesian optimization). A similar Bayesian optimization for the construction of an emulator of the 1D Lyα flux power spectrum has recently been considered in Rogers et al. (2019). The authors subsequently applied their method to the problem of the Lyα forest in Bird et al. (2019). The difference between their work and our here is that we use a different acquisition function and a different problem parametrization. In practice, we do expect the two approaches to result in similar computational efficiency and we consider our work as complementary to that of Rogers et al. (2019).

The execution of such an iterative workflow can be efficiently executed on high-performance computing platforms using the system described in Lohrmann et al. (2017). Briefly, as the workflow requires exploration of the parameter space via simulation trials, each of these simulations becomes a job managed by a parallel scheduler. This approach relies on Henson (Morozov & Lukić 2016), a cooperative multitasking system for the in situ execution of loosely coupled codes.

Our treatment of multi-output emulation is different from the previous approach of Habib et al. (2007), which relied on dimension-reducing techniques. Instead of approximating the power spectrum in the basis obtained from a principal component decomposition of the simulator's covariance structure, we assume a simple separable form for the covariance of the power spectrum as a vector-valued function (similarly to the approach of Conti & O'Hagan (2010)). This allows us to start with a small number of training inputs for the initial emulator construction and to iteratively refine the initial design. Additionally, the separable structure of the covariance function allows us to perform training and prediction with the emulator using Kronecker products of small matrices; this makes it efficient.

In this work, we use a 1D Lyα flux power spectrum as an output quantity of interest. Following reionization that occurs around redshift $z \sim 8$, the diffuse gas in the intergalatic medium (IGM) is predominantly photoionized, but the small residual fraction of the neutral hydrogen gives rise to Lyα absorption that is observed in spectra of distant quasar sightlines (for a recent review, see McQuinn (2016)). This so-called Lyα forest is the premier probe of the IGM and cosmic structure formation at redshifts $2 \lesssim z \lesssim 6$. As Lyα absorption at $z \sim 3$ is sensitive to gas at around the cosmic mean and at redshifts $z \geq 4$ even to the underdense gas in void regions of the universe (Lukić et al. 2015), complex and poorly understood physical processes related to galaxy formation are expected to play only a minor role in determining its structure (Desjacques et al. 2006; Kollmeier et al. 2006). Forward-modeling the structure of the IGM for a given cosmological and reionization scenario is thus a theoretically well-posed problem, but it requires expensive cosmological hydrodynamical simulations. The 1D power spectrum is a summary statistic of the Lyα flux field that measures the Fourier-space analog of two-point correlations in flux absorption along lines of sight to quasars. This statistic can be sensibly used to measure cosmological parameters (Seljak et al. 2006; Palanque-Delabrouille et al. 2020), constrain the neutrino sector (Palanque-Delabrouille et al. 2015; Rossi et al. 2015; Yèche et al. 2017), probe exotic dark matter

models (Armengaud et al. 2017; Iršič et al. 2017; Rogers & Peiris 2020), or measure the thermal properties of the IGM (Boera et al. 2019; Walther et al. 2019). Here, we focus on parameters describing the thermal state of the IGM, similarly as in Walther et al. (2019). However, there is nothing specific to the Lyα forest probe, particular data set or simulations in our inference formalism, thus the method we present here can straightforwardly be applied to other cosmological probes as well.

The outline of the paper is as follows. In Section 2 we describe the details of the forward model for the Lyα power spectrum. In addition to hydrodynamical simulations, we also use an approximate model for postprocessing the thermal state of the IGM which is described in Appendix A. In Section 3 we provide a high-level overview of our main approach to inferring cosmological parameters from measurement data. First, we state the general Bayesian inference problem, and following Bilionis & Zabaras (2014), we show how it can be reformulated using a Gaussian process (GP) emulator as a Bayesian surrogate of the forward model. Next, we provide an outline of the adaptive algorithm developed in Takhtaganov & Müller (2018), which we use to construct a GP emulator iteratively. The details of the GP emulator construction and a comparison of the approaches to modeling interactions between emulator outputs are provided in Section 4, as well as in the Appendix B. Results of applying this method on Viel et al. (2013) data and inferring thermal parameters of the IGM are given in Section 5. Finally, we present our conclusions in Section 6.

2. FORWARD MODEL

In this paper, we analyze different ways of inferring the model parameters using flux power spectrum observations. To this end, it is necessary to model the growth of cosmological structure and the thermal evolution of the IGM on scales far smaller (down to $\mathcal{O}(10h^{-1}\text{kpc})$) than those described by the linear perturbation theory. Cosmological hydrodynamical simulations with atomic cooling and UV heating are the only method capable of modeling this process at the percent level accuracy (for approximate methods, see Sorini et al. 2016; Lochhaas et al. 2016, and references therein). Unfortunately, such simulations are computationally very expensive, $\sim 10^5$ CPU hours or more. It is therefore desirable to also have a "reduced" model, which we can evaluate for a large number of points in the chosen parameter space, even if not as accurate as the full simulation model. In the following we will first review our "direct" simulation model, and in Section A we will present the approximate model based on post-processing the simulation's instantaneous temperature-density relation and the mean flux.

2.1. *Simulations*

The hydrodynamical simulations we use in this paper are part of the THERMAL suite of Nyx simulations (Walther et al. 2019) consisting of 75 models, each in $L = 20h^{-1}$Mpc box with 1024^3 Eulerian cells and 1024^3 dark matter particles. The Nyx code (Almgren et al. 2013) follows the evolution of dark matter modeled as self-gravitating Lagrangian particles, and baryons modeled as an ideal gas on a set of rectangular Cartesian grids. The Eulerian gas dynamics equations are solved using a second-order accurate piecewise parabolic method (PPM) to accurately capture shocks. Besides solving for gravity and the Euler equations, we also include the main physical processes relevant for the Lyα forest. We consider the chemistry of the gas as having a primordial composition of hydrogen and helium, include inverse Compton cooling off the microwave background and keep track of the net loss of thermal energy resulting from atomic collisional processes (Lukić et al. 2015). All cells are assumed to be optically thin to ionizing radiation, and radiative feedback is accounted for via a spatially uniform, time-varying UV background radiation given to the code as a list of photoionization and photoheating rates (Haardt & Madau 2012).

This type of simulations is used as a forward model in virtually any recent inference work using Lyman alpha power spectrum (Boera et al. 2019; Walther et al. 2019; Palanque-Delabrouille et al. 2020; Rogers & Peiris 2020). Effects of inhomogeneous reionization are neglected, both temperature and UV background fluctuations. We do expect future simulations to start adopting models with fluctuations (Oñorbe et al. 2019) to achieve better accuracy of the power spectrum, especially on larger scales. High column density absorbers ($N_{H_I} \gtrsim 10^{17}\text{cm}^{-2}$), which broaden absorption lines with characteristic damping wings (Rogers et al. 2018), are also not modelled in these simulations. Note that these percent-level details regarding the accuracy of the physics forward model are not affecting any of our conclusions.

Thermal histories are generated in a similar way as in Becker et al. (2011) through rescaling the photoheating by a density dependent factor: $\epsilon = A\Delta^B\epsilon_{\text{hm12}}$. Here, $\Delta = \rho_b/\bar{\rho}_b$ is the baryon overdensity, ϵ_{hm12} are the heating rates tabulated in Haardt & Madau (2012) while A and B are free parameters adjusted to obtain different thermal histories. Note that while this approach makes it straightforward to change instantaneous density-temperature relation in the simulation, changing the pressure smoothing scale is more difficult as it represents an integral of (an unknown) function of temperature across cosmic time. We will return to this point later in Section 5.

We choose mock sightlines, or "skewers", crossing the domain parallel to one of the axes of the simulation grid and piercing the cell centers. Computationally, this is the most efficient approach. This choice of rays avoids explicit ray-casting and any interpolation of the cell-centered data, which introduce other numerical and periodicity issues. As a result, from an N^3 cell simulation,

we obtain up to N^2 mock spectra, each spectrum having N pixels. We calculate the optical depth, τ, by convolving neutral hydrogen in each pixel along the skewer with the profile for a resonant line scattering and assuming Doppler shift for velocity (for details, see Lukić et al. 2015). We compute this optical depth at a fixed redshift, meaning we do not account for the speed of light when we cast rays in the simulation; we use the gas thermodynamical state at a single cosmic time. The simulated skewers are therefore not meant to globally mock observed Lyα forest spectra, but they do recover the statistics of the flux in a narrow redshift window, which is what we need for this work. We have neglected instrumental noise and metal contamination in simulated skewers, but this will not be relevant for the conclusions of this paper.

2.2. *Model parameters*

Lyman-α forest simulations include both cosmological parameters as well as astrophysical parameters needed to model the thermal state of the gas, which is significantly affected by hydrogen and helium reionizations. Our main goal is to test and improve the parameter sampling scheme and the emulation method used for constraining the parameters; in order to reduce the computational expense, in this work we will focus our attention on the set of "standard" parameters, $\{T_0, \gamma, \lambda_P, \bar{F}\}$, describing the thermal state of the IGM. We keep the cosmological parameters fixed and based on the Planck Collaboration et al. (2014) flat ΛCDM model with $h = 0.67$, $\Omega_m = 0.32$, $\Omega_b h^2 = 0.022312$, $n_s = 0.96$, $\sigma_8 = 0.8288$.

The values for thermal parameters T_0 and γ are obtained from the simulation by approximating the temperature-density relation as the power law:

$$T = T_0 \Delta^{\gamma - 1}\,, \tag{1}$$

and finding the best fit using a linear least squares method in $\log T$ – $\log \Delta$ (Lukić et al. 2015). Therefore, T_0 parametrizes the temperature at mean density in the universe, while γ is the slope of temperature-density relation, expected to asymptote $\gamma \approx 1.6$ long after reionization ends. To determine the pressure smoothing scale, λ_P in post-processing. , we fit the cutoff in the power spectrum of the real-space Lyα flux, as described in Kulkarni et al. (2015). Real-space Lyα flux is calculated using actual density and temperature at each cell in the simulation, but omitting all redshift-space effects such as peculiar velocities and thermal broadening.

In section 5.3 we will demonstrate our adaptive GP approach on the problem of inferring the three parameters $\boldsymbol{\theta} = (F, T_0, \gamma)$. There, we use as the forward model the post-processing of hydrodynamical simulations as the forward model, which is described in Appendix A. We use only three out of four parameters, as there is no good way to model λ_P in post-processing since this parameter depends on the integrated thermal history rather than one moment in time. The advantage of using this approximate model is that it allows us to evaluate our model at different points in parameter space at a very low computational cost. While this in principle already demonstrates the efficiency of our new sampling scheme, the worry of conclusions being affected by leaving out λ_P parameter can be legitimately raised.

To alleviate that worry and confirm that demonstrated behavior of our adaptive GP emulator is not qualitatively different when parameter space changes, in Section 5.4 we apply our technique on the problem where a forward model consist of hydrodynamical simulations and a full set of $\{T_0, \gamma, \lambda_P, \bar{F}\}$ parameters. For computational efficiency, we use an existing THERMAL suite of Nyx simulations[1]. This means we are not evaluating the forward model at any chosen point, which would be the case in practical application of our method. Importantly, the conclusions from this experiment are fully consistent with experiment presented in Section 5.3.

3. ADAPTIVE CONSTRUCTION OF GAUSSIAN PROCESS EMULATORS FOR BAYESIAN INFERENCE

In this section we outline our main approach to the adaptive construction of the GP emulators. Our approach is designed to solve the specific problem at hand, which is inferring the parameters of interest that serve as input into the forward model of the power spectrum from the observational data. The main ingredient of our approach is a so-called "acquisition" function that guides the selection of the training inputs for the emulator. This acquisition function arises from the form of the likelihood for the measurement data. Thus, before explaining the acquisition process we start by providing a general framework.

We denote the parameters of interest by $\boldsymbol{\theta} \in \mathbb{R}^p$. We denote the vector of observations by $\mathbf{d} = (d_1, \ldots, d_q)^T$, where each d_i represents a measured value of the Lyα forest flux power spectrum at a certain value of the wavenumber k. The outputs of the forward model of the power spectrum for a given $\boldsymbol{\theta}$ will be thought of as a q-dimensional vector $\mathbf{P}(\boldsymbol{\theta})$ (more on this in Section 4). We will work under the assumption of the Gaussian measurement noise with zero mean and known covariance $\boldsymbol{\Sigma}_E$. With this specification of the measurement noise we formulate the likelihood function for the observational data which depends on the value of the forward model at $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta}|\mathbf{d}) = \mathcal{N}_q(\mathbf{d} - \mathbf{P}(\boldsymbol{\theta}) \,|\, \mathbf{0}_q, \boldsymbol{\Sigma}_E). \tag{2}$$

Assuming the Bayesian framework, we model the prior information about the parameters $\boldsymbol{\theta}$ as a known distribution $p(\boldsymbol{\theta})$. Given the prior and the observed measurements $\mathbf{d}$, the solution of the inverse problem is the

[1] http://thermal.joseonorbe.com/

posterior density obtained by applying Bayes' rule:

$$p(\boldsymbol{\theta}|\mathbf{d}) \propto L(\boldsymbol{\theta}|\mathbf{d})p(\boldsymbol{\theta}). \tag{3}$$

This posterior density can, in principle, be explored with MCMC methods. However, since evaluating the likelihood function $L(\boldsymbol{\theta}|\mathbf{d})$ requires evaluating the forward model $\mathbf{P}(\boldsymbol{\theta})$, the direct application of MCMC methods for the current application (or any expensive-to-evaluate forward model) is infeasible. This difficulty can be circumvented by using a surrogate model, such as a Gaussian process emulator, in place of the forward model. A Gaussian process is fully specified by its mean and covariance functions. The mean is commonly set to zero, while the covariance function describes statistical relation between data points and is reflection of the prior beliefs. This Bayesian nature of the Gaussian processes makes them particularly suitable for our framework. Next, we provide a formal review of GP emulators leaving the details out until Section 4 and outline our adaptive approach to sequentially adding training inputs for the emulator.

Suppose that we have collected a set of evaluations of the forward model $\mathbf{P}(\boldsymbol{\theta})$ at n input points:

$$\mathcal{D} = \{\boldsymbol{\theta}^{(j)}, \mathbf{P}(\boldsymbol{\theta}^{(j)})\}_{j=1}^{n}. \tag{4}$$

The information in $\mathcal{D}$ can be used to obtain a surrogate model specified by a random variable $\mathbf{P}^{GP}$ with a predictive distribution conditioned on the input $\boldsymbol{\theta}$ and data $\mathcal{D}$:

$$p\big(\mathbf{P}^{GP} \,|\, \boldsymbol{\theta}, \mathcal{D}\big) = \int p\big(\mathbf{P}^{GP} \,|\, \boldsymbol{\theta}, \mathcal{D}, \boldsymbol{\psi}\big) p(\boldsymbol{\psi} \,|\, \mathcal{D}) d\boldsymbol{\psi}. \tag{5}$$

Here $\boldsymbol{\psi}$ denotes the *hyperparameters* of the predictive model, $p\big(\mathbf{P}^{GP} \,|\, \boldsymbol{\theta}, \mathcal{D}, \boldsymbol{\psi}\big)$ is the predictive distribution of the assumed model given the hyperparameters, and $p(\boldsymbol{\psi}|\mathcal{D})$ is the posterior distribution of $\boldsymbol{\psi}$ given data $\mathcal{D}$. Hyperparameters are thus free parameters which allow for customization ("training") of the Gaussian process for the particular problem. Together with the choice of the covariance function, they define a specific *Gaussian process model* (this will be detailed in Section 4.1). In this work, we use the Bayesian view on model selection, where the optimal hyperparameters are determined by maximizing the probability of this model given the data. Specifically, we use the maximum likelihood estimate (MLE) of the hyperparameters when training the surrogate model $\mathbf{P}^{GP}$. More generally, one could apply MCMC for obtaining $p(\boldsymbol{\psi}|\mathcal{D})$, see details in Appendix B.

The solution of the inverse problem with a limited number of forward model evaluations can now be formulated using the likelihood of the observational data evaluated using the surrogate model. This *$\mathcal{D}$-restricted likelihood* (similarly to Bilionis & Zabaras (2014)) is defined as follows:

$$L(\boldsymbol{\theta} \,|\, \mathbf{d}, \mathcal{D}) = \int L\big(\boldsymbol{\theta} \,|\, \mathbf{d}, \mathbf{P}^{GP}\big) p\big(\mathbf{P}^{GP} \,|\, \boldsymbol{\theta}, \mathcal{D}\big) d\mathbf{P}^{GP}, \tag{6}$$

where $L\big(\boldsymbol{\theta} \,|\, \mathbf{d}, \mathbf{P}^{GP}\big) = \mathcal{N}_q\big(\mathbf{d} - \mathbf{P}^{GP}(\boldsymbol{\theta}) \,|\, \mathbf{0}_q, \boldsymbol{\Sigma}_E\big)$. Once we have an approximation of the distribution $p(\boldsymbol{\psi}|\mathcal{D})$, e.g., ψ_{MLE} when using MLE approach, we can integrate the product of the two Gaussians and obtain an approximate formula for the likelihood $L(\boldsymbol{\theta} \,|\, \mathbf{d}, \mathcal{D})$, see Takhtaganov & Müller (2018) and Appendix B for details:

$$L(\boldsymbol{\theta} \,|\, \mathbf{d}, \mathcal{D}) \approx \exp\left[-\frac{g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}_{MLE})}{2}\right] \tag{7}$$

Evaluations of the approximate likelihood $L(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D})$ involve computing the following *misfit function* between the observational data and the predictions of the GP emulator

$$\begin{aligned} g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) =& (\mathbf{d} - \mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}))^T (\boldsymbol{\Sigma}_E + \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}))^{-1} \\ & \times (\mathbf{d} - \mathbf{m}(\boldsymbol{\theta}, \mathcal{D}, \boldsymbol{\psi})) \end{aligned} \tag{8}$$

for the estimate $\boldsymbol{\psi} = \boldsymbol{\psi}_{MLE}$. In (8), we denote by $\mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ the mean vector of the GP emulator evaluated at $\boldsymbol{\theta}$, and by $\boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ its predictive covariance. This misfit function captures the discrepancy between the observed values of the power spectrum and the predicted values in the norm weighted by the measurement error and the uncertainty of the emulator. Note that for the inputs $\boldsymbol{\theta}$ in the training set $\mathcal{D}$, the mean of the GP emulator $\mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ coincides with the values of the power spectrum $\mathbf{P}(\boldsymbol{\theta})$, and the covariance $\boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ vanishes, and thus the exact value of the misfit (and hence the likelihood) is known.

We use the misfit function (8) to inform our choice of the candidate inputs to add to the dataset $\mathcal{D}$ in order to improve the GP surrogate. The "improvement" we are looking for is to make the approximate likelihood $L(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D})$ more accurately resemble the "true" likelihood $L(\boldsymbol{\theta}|\mathbf{d})$. The overall accuracy of the GP emulator over the support of the prior is unimportant.

Our adaptive approach to extending the training set $\mathcal{D}$ is based on an acquisition function commonly used in Bayesian optimization—the so-called "expected improvement" (EI) criterion (Jones et al. 1998). In our version of the EI criterion, we look for an input $\boldsymbol{\theta}$ that provides the largest *expected improvement in fit* with expectation taken with respect to the posterior distribution of the hyperparameters $p(\boldsymbol{\psi}|\mathcal{D})$:

$$\begin{aligned} \mathcal{I}(\boldsymbol{\theta}) \equiv & \int [g_{min} - g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})]^+ p(\boldsymbol{\psi}|\mathcal{D}) d\boldsymbol{\psi} \\ & \approx [g_{min} - g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}_{MLE})]^+. \end{aligned} \tag{9}$$

Here g_{min} denotes the smallest misfit to the measurement data for the points in the current training set $\mathcal{D}$, and $[\,\cdot\,]^+$ takes the positive part of the difference: $[\,\cdot\,] \equiv \max\{\,\cdot\,, 0\}$. This formulation allows us to balance the exploration and the exploitation of the GP emulator

in an iterative search for a new training input to maximize the *expected improvement in fit* function $\mathcal{I}(\boldsymbol{\theta})$, i.e., we search for the input $\boldsymbol{\theta}$ that provides the largest improvement in fit to the measurement data under the current GP model, conditioned on the misfit being smaller than the current best value for the points in the training set. The outline of the algorithm is given in Algorithm 1. For more details see Takhtaganov & Müller (2018).

Algorithm 1 Adaptive construction of GP emulators

Input: Initial design $\{\boldsymbol{\theta}^{(j)}\}_{j=1}^{n}$, threshold value ϵ_{thresh}, search space $\mathcal{X}_\theta$, maximum allowed number of forward model evaluations s_{max}.

Output: Adaptive design $\mathcal{D}$.

1: Evaluate $\mathbf{P}(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \{\boldsymbol{\theta}^{(j)}\}_{j=1}^{n}$ to obtain $\mathcal{D} = \{\boldsymbol{\theta}^{(j)}, \mathbf{P}(\boldsymbol{\theta}^{(j)})\}_{j=1}^{n}$.
2: **for** s from 1 to s_{max} **do**
3: train the GP model using current design;
4: update the current best fit value g_{min};
5: maximize the expected improvement in fit function $\mathcal{I}(\boldsymbol{\theta})$ over the search space $\mathcal{X}_\theta$, and let $\boldsymbol{\theta}^s = \arg\max_{\boldsymbol{\theta}\in\mathcal{X}_\theta} \mathcal{I}(\boldsymbol{\theta})$;
6: **if** $\mathcal{I}(\boldsymbol{\theta}^s) < \epsilon_{thresh} \cdot g_{min}$ **then**
7: **break**
8: **end if**
9: evaluate $\mathbf{P}$ at $\boldsymbol{\theta}^s$ and augment the training set: $\mathcal{D} = \mathcal{D} \cup \{\boldsymbol{\theta}^s, \mathbf{P}(\boldsymbol{\theta}^s)\}$;
10: **end for**
11: **return** $\mathcal{D}$.

For our numerical experiments, we take the search space $\mathcal{X}_\theta$ to be the support of the prior $p(\boldsymbol{\theta})$, and we set the threshold value ϵ_{thresh} to be 1%. We solve the auxiliary optimization problem in Step 5 by using multi-start gradient-based optimization, see Takhtaganov & Müller (2018) for details. We set the allowed number of simulations s_{max} to a large number so that the effective termination condition is the one on line 6. In practice, s_{max} is dictated by the simulation budget. In the next section, we discuss the details of the construction of the GP emulators for modeling the Lyα forest power spectrum.

4. GAUSSIAN PROCESS EMULATORS FOR THE LYα FLUX POWER SPECTRUM

We model the power spectrum $P(k, \boldsymbol{\theta})$ as a multi-output Gaussian process with outputs corresponding to the fixed values of the wavenumber k. Furthermore, we assume a separable structure of the kernel function, meaning that it can be formulated as a product of a kernel function for the input space $\boldsymbol{\theta}$ alone, and a kernel function that encodes the interactions between the outputs k (Alvarez et al. 2012, Section 4).

In the following subsections we discuss in details the construction of multi-output GP emulators for the modeling of the power spectrum $P(k, \boldsymbol{\theta})$. To our knowledge, a similar comparison of multi-output emulators has not been done in the current context. Through a detailed numerical comparison, we arrive at the two preferred approaches that are used in the rest of the paper. Our choices are motivated by considerations of efficiency, accuracy, and correct modeling of correlation between the outputs.

The preferred approach will inherently be application dependent. We work here on the Lyman α flux power spectrum, but we emphasize that in a different setting, with much stronger correlations, there would be a bigger difference between different approaches. The methodology that we describe in this section can thus be used to diagnose when to apply a given approach, and is meant to serve as a practical guide for the application specialists. Readers who are not practitioners themselves could skip this section and proceed to results presented in Section 5.

4.1. *Gaussian process emulators*

We will treat $P(k, \boldsymbol{\theta})$ as a function from $\mathcal{X}_k \times \mathcal{X}_\theta$ to $\mathbb{R}^q$, where $\mathcal{X}_k \subset \mathbb{R}$, $\mathcal{X}_\theta \subset \mathbb{R}^p$, and q is the number of values of the wavenumber k for which we have observations. For a given vector of input parameters $\boldsymbol{\theta}$, we treat the output of the simulation code as a vector $\mathbf{P}(\boldsymbol{\theta}) \in \mathbb{R}^q$ of the values of the power spectrum at fixed values of k.

Similarly to Conti & O'Hagan (2010), we model $\mathbf{P}(\cdot)$ as a q-dimensional separable Gaussian process:

$$\mathbf{P}^{GP}(\cdot)|\boldsymbol{\psi}, \boldsymbol{\Sigma}_k \sim \mathcal{N}_q(\boldsymbol{\mu}(\cdot), c(\cdot,\cdot\,;\boldsymbol{\psi})\boldsymbol{\Sigma}_k), \tag{10}$$

conditional on hyperparameters $\boldsymbol{\psi}$ of the correlation function $c : \mathcal{X}_\theta \times \mathcal{X}_\theta \times \mathcal{X}_\psi \to \mathbb{R}$, and symmetric positive-definite matrix $\boldsymbol{\Sigma}_k \in \mathbb{R}^{q\times q}$. This means that for any two inputs $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$, we have $\mathbb{E}\left[\mathbf{P}^{GP}(\boldsymbol{\theta}^{(i)}) \,|\, \boldsymbol{\psi}, \boldsymbol{\Sigma}_k\right] = \boldsymbol{\mu}(\boldsymbol{\theta}^{(i)})$, $i = 1, 2$, and $\mathbb{C}\mathrm{ov}\left[\mathbf{P}^{GP}(\boldsymbol{\theta}^{(1)}), \mathbf{P}^{GP}(\boldsymbol{\theta}^{(2)}) \,|\, \boldsymbol{\psi}, \boldsymbol{\Sigma}_k\right] = c(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}; \boldsymbol{\psi})\boldsymbol{\Sigma}_k$. As indicated by several studies, e.g., Chen et al. (2016), the introduction of the regression term $\boldsymbol{\mu}(\cdot)$ does not generally affect the performance of the predictive model, and, in some cases, might have an adverse effect. In our case, adequate results were obtained by simply setting $\boldsymbol{\mu}(\boldsymbol{\theta}) \equiv 0$. Furthermore, we set the covariance function $c(\cdot,\cdot\,;\boldsymbol{\psi})$ to be squared-exponential with $p+1$ hyperparameters $\boldsymbol{\psi} = (\sigma_c, \ell_1, \ldots, \ell_p)^T$:

$$c(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}; \boldsymbol{\psi}) = \sigma_c^2 \exp\left(-\sum_{i=1}^{p} \frac{\left(\theta_i^{(1)} - \theta_i^{(2)}\right)^2}{2\ell_i^2}\right). \tag{11}$$

Note that the choice of the covariance function here is purely empirical and does not affect the forthcoming methodology.

Our treatment of the inter-output covariance matrix $\boldsymbol{\Sigma}_k$ differs from that in Conti & O'Hagan (2010) and Bilionis et al. (2013). There, the authors assume a weak non-informative prior on the matrix $\boldsymbol{\Sigma}_k$ and integrate it

out of the predictive posterior distribution. Instead, we study four different approaches for treating interactions between outputs, summarized in the following Section 4.2.

4.2. *Approaches for dealing with multi-output models*

1. First, we test a naive approach that emulates each output separately with a single-output GP. We refer to this approach as (MS) as it corresponds to the MS (many single-output) emulator in Conti & O'Hagan (2010). This approach has an increased computational cost (which could be alleviated by training in parallel) compared to training only one GP as the following two approaches do. Also, this approach ignores any dependencies between the outputs.

2. In our second approach (IND), we treat the outputs as independent given the hyperparameters of the covariance function $c(\cdot, \cdot\,; \boldsymbol{\psi})$. This approach has been considered in Bilionis & Zabaras (2014) and Takhtaganov & Müller (2018). It leads to a simple and efficient implementation of a multi-output emulator with a diagonal $\boldsymbol{\Sigma}_k$, see Appendix B for details.

3. Our third approach (COR) assumes that correlations between different outputs are non-zero but are still independent of the parameter $\boldsymbol{\theta}$. In this case, we fix the correlation matrix a priori and use it to obtain $\boldsymbol{\Sigma}_k$ by rescaling by the training variances. This approach requires specifying the inter-output correlation matrix. In terms of computational efficiency, this approach still allows us to use the Kronecker product structure of the training covariances (see Appendix B) and is as efficient as using the diagonal covariance in approach IND.

4. Our final approach (INP) is related to Approach COR, but it is computationally more demanding. Here, we treat k as another input dimension into the GP model with associated covariance kernel being again a squared-exponential. The main computational cost is associated with inverting the training covariance matrices, which become q times larger due to the addition of another input dimension. Conceptually, however, this approach is similar to Approach COR with $\boldsymbol{\Sigma}_k$ having entries specified by the squared-exponential kernel for a fixed (learned) value of the length-scale hyperparameter ℓ_{p+1} associated with the k input dimension. The difference to Approach COR is that the inter-output correlation matrix now depends on the training data. As our experiments demonstrate, this additional flexibility provides no discernible advantage. This approach is referred to as TI (time-input) emulator in Conti & O'Hagan (2010) where an extra dimension is time rather than k.

All of our approaches utilize matrix-valued kernels that fall into the category of *separable* kernels, see (Alvarez et al. 2012, Section 4) for an overview. Specifically, our Approaches IND, COR, and INP are examples of the so-called *intrinsic coregionalization model* or ICM (Alvarez et al. 2012, Section 4.2). The ICM approach allows for an efficient implementation of GP-based regression and inference that exploits the properties of the Kronecker product of the covariance matrix, see Appendix B for details. For the adaptive algorithm, efficiency is important for solving the auxiliary optimization problem for the expected improvement in fit function $\mathcal{I}(\boldsymbol{\theta})$. Solving this optimization problem requires multiple restarts as the $\mathcal{I}(\boldsymbol{\theta})$ function is highly multi-modal, leading to a large number of evaluations of the GP emulator predictions and their gradients. For the GP-based inference, having an efficient emulator allows us to carry out MCMC sampling of the posterior with minimum computational effort. As the following numerical study of the considered approaches suggests, in our application it is reasonable to expect that the outputs corresponding to different values of the wavenumber k have similar properties with respect to the parameters $\boldsymbol{\theta}$, therefore, our choice of the separable form of the kernel is justified.

4.3. *Numerical study of multi-output approaches*

In this section we compare the predictive performance of the different multi-output Gaussian process emulators introduced in Section 4.2 using an approximate model of the power spectrum $P(k, \boldsymbol{\theta})$ described in Section A. To obtain a better picture of the dependence of the results on the choice of design inputs, we build emulators using 10 to 30 inputs arranged in a Latin Hypercube Design (LHD) where the minimum distance between the points has been maximized—the so-called maximin LHD (see, e.g., Johnson et al. (1990)). For each design we perform multiple experiments. For each experiment we generate a large test set consisting of 500 input-output pairs for computing various measures of predictive accuracy. In order to avoid an unnecessary cost associated even with the post-processing procedure of Section A, we precompute the power spectrum on a dense grid in $\mathcal{X}_\theta$ using our automatized Henson system (Morozov & Lukić 2016; Lohrmann et al. 2017) and interpolate the outputs using tri-linear interpolation to obtain the continuous approximation of the power spectrum in $\mathcal{X}_\theta$. We have confirmed that the $P(k, \boldsymbol{\theta})$ error associated with this interpolation is negligible. Specifically, we take a grid of 10^3 input parameters $\boldsymbol{\theta} = (F, T_0, \gamma)$ covering the box

$$\mathcal{X}_\theta = [0.2, 0.5] \times [3 \times 10^3 K, 3 \times 10^4 K] \times [1.0, 2.0] \; . \quad (12)$$

We restrict our attention to the redshift of $z = 4.2$ and $q = 8$ outputs corresponding to the following k values $\{3.26 \times 10^{-3}, 6.51 \times 10^{-3}, 9.77 \times 10^{-3}, 1.63 \times 10^{-2}, 2.28 \times$

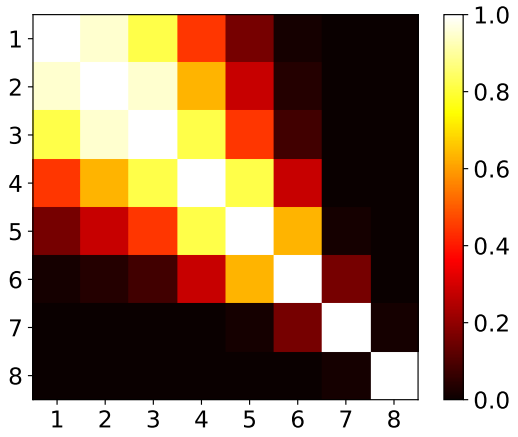


Figure 1. Correlation matrix between outputs (k-wavemodes) 1 through 8 for the COR emulator.

$10^{-2}, 3.26\times10^{-2}, 5.21\times10^{-2}, 8.14\times10^{-2}\}\mathrm{km}^{-1}\mathrm{s}$, which cover the range of values as the Viel et al. (2013) measurements that we use later. In the following we simply number these k outputs from 1 to 8.

We construct the four multi-output GP emulators (MS, IND, COR, and INP) using fixed LHDs with 10, 20, and 30 points in $\mathcal{X}_\theta$. In each case, the training output values are normalized (see Appendix B). We fit the hyperparameters of the covariance function using the maximum likelihood approach (MLE in Appendix B). Recall that the COR emulator requires a fixed output correlation matrix $\mathbf{\Sigma}_k$. We obtain an estimate of this matrix by first using the INP emulator built with LHD 20 design. Upon training of the INP emulator we obtain an estimate of the length scale for its k (output) variable. By plugging-in this estimate into the one-dimensional squared-exponential kernel and evaluating the kernel for the values of k that we consider, we get a desired estimate of $\mathbf{\Sigma}_k$. This estimate remains fixed for all experiments with the COR emulator, see Figure 1.

To test the predictive performance of the emulators we use a test set consisting of $N = 500$ points in a randomized LHD. We use the following measures of predictive accuracy.

1. Standardized mean squared error (SMSE): this is the mean-squared prediction error scaled by the variance of the test data for each output $j = 1, \ldots, q$:

$$\mathrm{SMSE}_j = \frac{\sum_{i=1}^{N} \left(m_j(\boldsymbol{\theta}^{(i)}, \mathcal{D}) - P_j(\boldsymbol{\theta}^{(i)})\right)^2}{\sum_{i=1}^{N} \left(\overline{P}_j - P_j(\boldsymbol{\theta}^{(i)})\right)^2}, \quad (13)$$

where $m_j(\boldsymbol{\theta}, \mathcal{D})$ is the j-th component of the vector of predictive means $\mathbf{m}(\boldsymbol{\theta}, \mathcal{D})$ of the GP model, and $\overline{P}_j$ is the mean of the test values $P_j(\boldsymbol{\theta}^{(i)})$, $i = 1, \ldots, N$.

2. Credible interval percentage (CIP), also known as coverage probability: the percentage of the $100\alpha\%$ credible intervals that contain the true test value. For an emulator that provides adequate estimates of the uncertainty about its predictions, this value should be close to α. We plot the CIP against $\alpha \in [0, 1]$ and look for deviations from the straight line. This statistic can only be plotted for each output separately.

3. Squared Mahalanobis distance (SMD) between the predicted and the test outputs at a test point i:

$$\mathrm{SMD}_i = \left(\mathbf{P}(\boldsymbol{\theta}^{(i)}) - \mathbf{m}(\boldsymbol{\theta}^{(i)}, \mathcal{D})\right) \mathbf{\Sigma}_{GP}(\boldsymbol{\theta}^{(i)}, \mathcal{D})^{-1} \times \left(\mathbf{P}(\boldsymbol{\theta}^{(i)}) - \mathbf{m}(\boldsymbol{\theta}^{(i)}, \mathcal{D})\right), \quad (14)$$

where $\mathbf{m}(\boldsymbol{\theta}, \mathcal{D})$ and $\mathbf{\Sigma}_{GP}(\boldsymbol{\theta}, \mathcal{D})$ are the predictive mean and the predictive covariance of the multi-output GP emulator, respectively. According to the multivariate normal theory, this distance should be distributed as χ^2_q for all test points. A discrepancy between the distribution of distances for the emulator and a reference distribution indicates a misspecification of the covariance structure between the outputs.

Figure 2 reports the SMSE estimates for the three LHDs. In each case we repeat the experiment 20 times, meaning that we generate 20 training and 20 test sets for each of the three designs and use them to train and test each of the four emulators. We observe a noticeable spread of the estimates between the experiments with the same number of design points regardless of the chosen emulator or output index. Increasing the size of the training design generally improves accuracy but does not necessarily combat the variation in results for different experiments. All four emulators demonstrate similar output-marginal accuracy, and the difference in errors for different outputs is low, with the exception of output 8, which has consistently higher relative errors for all four emulators. The COR and the INP emulators appear to have a similar spread of the error values indicating that fixing the output covariance $\mathbf{\Sigma}_k$ a priori rather than allowing it to depend on training data has little effect on accuracy. The MS and the IND emulators achieve smaller errors, but also have relatively larger spreads between experiments, which suggests higher dependence on the design than in the cases of COR and INP emulators.

While there is little that separates different multi-output approaches in terms of per-output SMSE, the results for CIP and SMD provide a more complete picture.

Figure 3 reports the CIP estimates for the three LHDs and for selected output indices (1, 4, and 7, in the top,

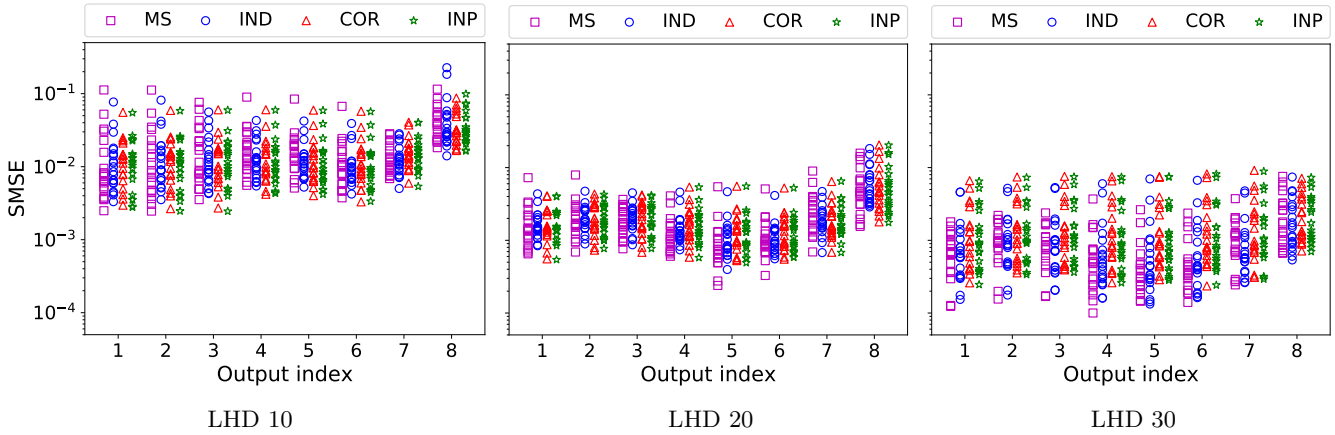


Figure 2. Standardized mean-squared errors (SMSE) for the four multi-output emulators (MS, IND, COR, INP) for three different LHDs (LHD 10, 20, and 30). Test errors are computed with 500 points. Each experiment is repeated 20 times.

middle, and bottom rows, respectively). Here we report the CIP values averaged over the 20 experiments. In general, we observe that all four emulators underestimate the uncertainty in their predictions, i.e., they are over-confident in their predictions. This trend becomes more pronounced with growing training design size. We view this as an artifact of the MLE approach to GP training, see discussion in (Takhtaganov & Müller 2018, Section 3). Among the emulators, the MS emulator exhibits the worst performance and is most over-confident in its predictions in all cases. The other emulators compensate for the shortcomings of the MLE approach by accounting for correlations between the outputs. The IND emulator does it in a least effective way, and hence performs worse than the COR and the INP emulators. The COR emulator on average comes closest to providing accurate uncertainty estimates.

Figure 4 shows the box plots of the distributions of the squared Mahalanobis distances for the test points for the three designs. We only show the results of a single experiment for each design. The reference distribution χ^2_8 has mean of 8 and variance of 16. We observe that although all of the emulators fail to correctly model the posterior covariance (a direct consequence of underestimating the predictive uncertainty), the IND, COR, and INP emulators all do better than MS with the last two coming closest to the reference. If we look at the means of the distributions, then for LHD 10, the average values over the 20 experiments are 54, 38, 27, and 29 for the MS, IND, COR, and INP emulators, respectively. The results shown represent the general trend that, on average, COR and INP emulators tend to represent the predictive covariance better.

In terms of the wall-clock time required for training and evaluating the four emulators, the most time-consuming emulator to train is INP for which the required training time grows exponentially with the size of the training set. The next most time-consuming emulator to train is MS, however, it can be easily trained and evaluated in parallel since there is no overlap between the outputs. The time for training IND and COR emulators is approximately the same and is considerably less than for the other two since only a single GP needs to be trained.

Based on the performed experiments, in the following we choose to work with IND and COR emulators due to their good performance on the three test statistics as well as good computational efficiency. The question of how to obtain the correlation matrix for the COR emulator still remains open. In our numerical experiments we did not observe significant differences in the emulator performance with the small changes to the correlation matrix $\boldsymbol{\Sigma}_k$. In particular, the estimates of $\boldsymbol{\Sigma}_k$ obtained with the INP emulator were similar for all experiments. It appears that the variability in the results due to the training designs outweighs the variability from using approximate correlations.

As mentioned previously, an alternative way of treating the separable form of the covariance is to assume a "non-informative" prior on $\boldsymbol{\Sigma}_k$ as in Conti & O'Hagan (2010) which allows analytical integration of this matrix out of the predictive distribution. In addition, if the mean $\boldsymbol{\mu}(\cdot)$ is taken to be a generalized linear model with a flat prior, it can also be treated analytically. We deviated from the approach of Conti & O'Hagan (2010) for the sake of a simpler and more interpretable model.

5. NUMERICAL STUDY OF INFERENCE WITH ADAPTIVE GP EMULATORS

In this section we evaluate the performance of the adaptive construction of the GP emulators introduced in Section 3. First, we solve a synthetic inference problem for the same set of three parameters $\boldsymbol{\theta} = (F, T_0, \gamma)$. For this task, we generate synthetic measurement data with the post-processing model of Section A and corrupt it with noise. Similarly to Section 4.3, we use a trilinear interpolation of the outputs of the post-processing model as the forward model for the GP construction and inference. We run Algorithm 1 and use the con-

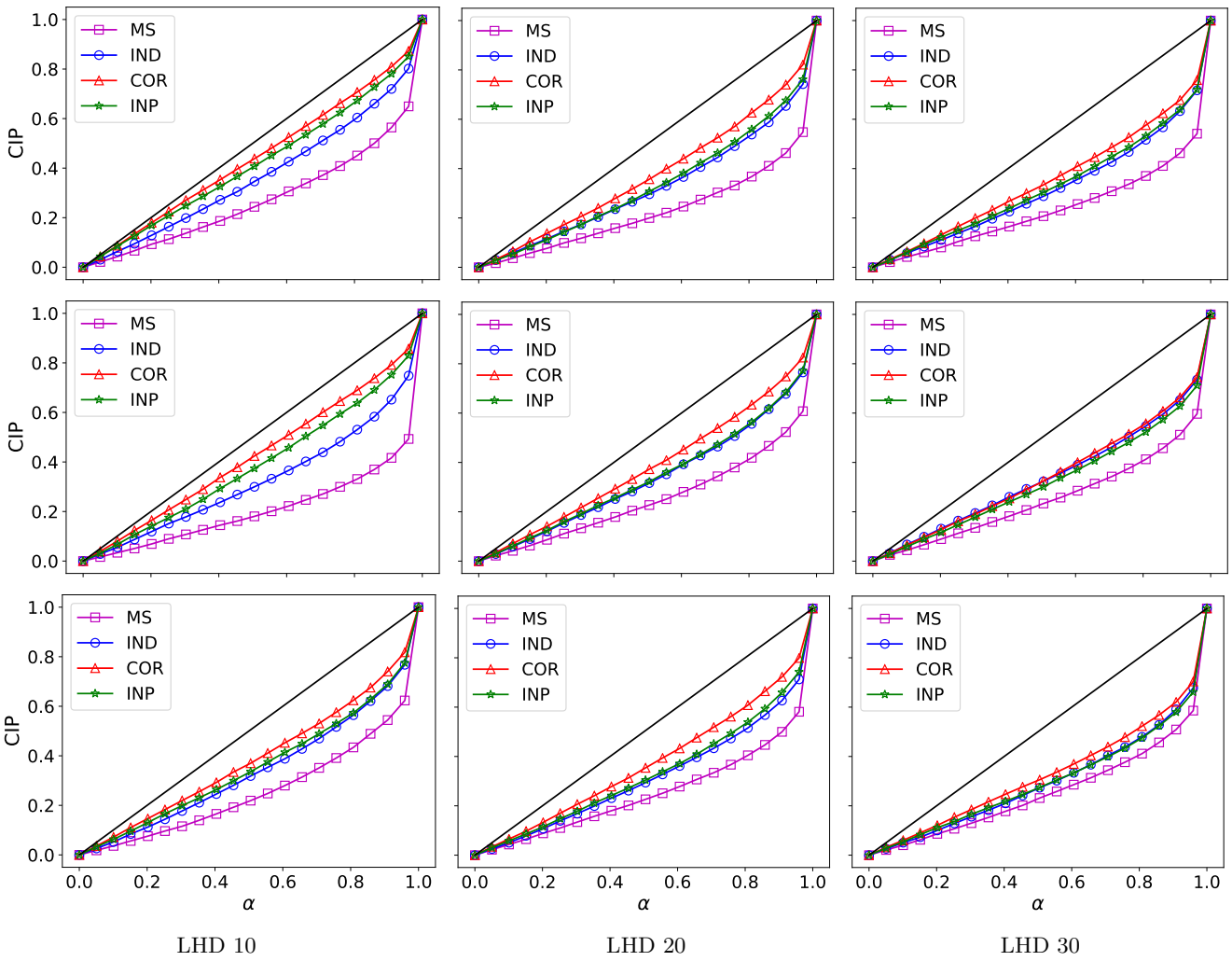
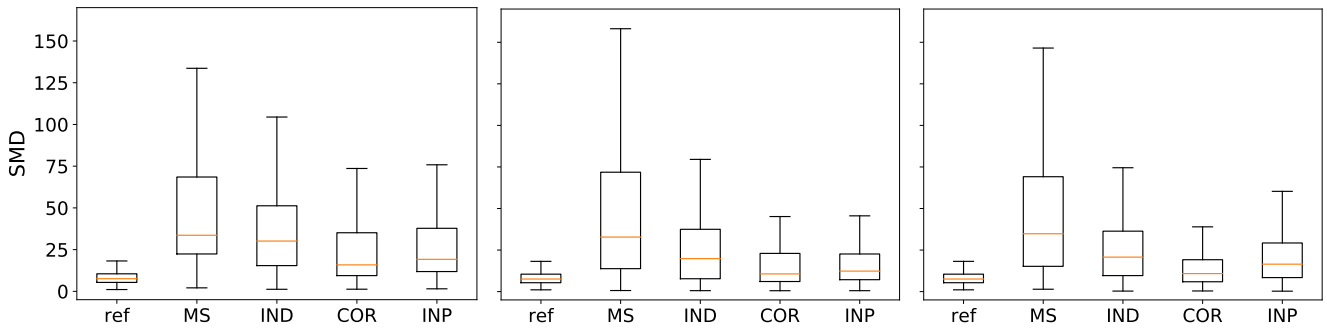


Figure 3. Percentage of credible intervals (CIP) containing the true test values for the four multi-output emulators (MS, IND, COR, INP) for three different LHDs based on 500 test points and averaged over 20 experiments. Top row - output 1, middle row - output 4, bottom row - output 7. The closer the colored graphs are to the black line, the better.

structed GP emulator to obtain the posterior of the parameters $\boldsymbol{\theta}$ given the measurements. We compare the results obtained with the adaptive approach to those obtained using the GP emulators built using fixed design (see Section 5.1). Having a relatively inexpensive forward model, we can obtain a reference posterior using the "true" likelihood $L(\boldsymbol{\theta}|\mathbf{d})$, which allows us to obtain quantitative measures of the quality of the GP-based posteriors.

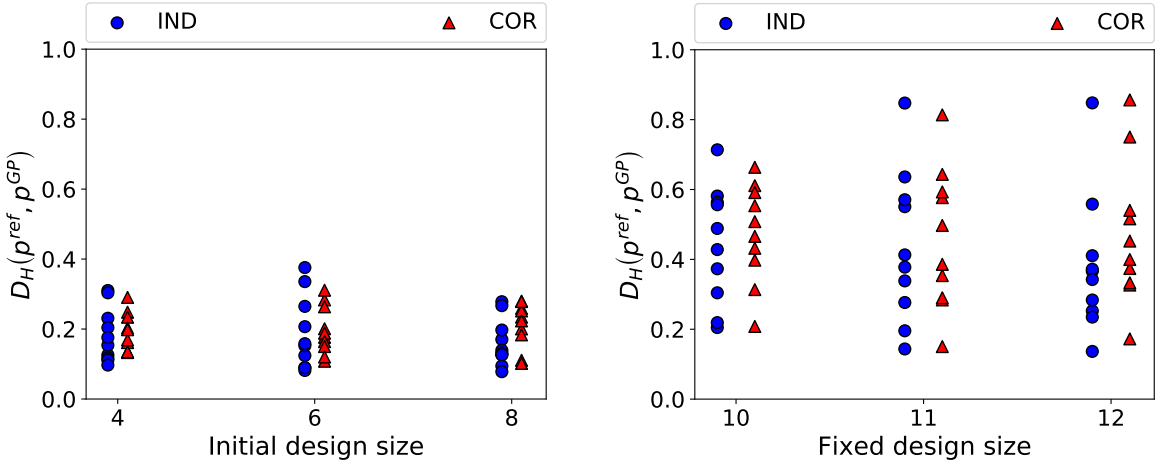
Following this detailed study, we use the adaptive algorithm to obtain parameter posteriors using observational data from Viel et al. (2013) and using the post-processing model A as the forward model. These results are reported in Section 5.3.

Finally, we use a simplistic version of our adaptive approach to construct posteriors for an extended set of parameters, namely $\boldsymbol{\theta} = (F, T_0, \gamma, \lambda_P)$, using the same Viel data and the THERMAL suite of Nyx simulations. Here, we restrict the search space $\mathcal{X}_\theta$ to the 75 points for the parameters (T_0, γ, λ_P) in this dataset and 40 values of F giving us a total of 3,000 possible values of $\boldsymbol{\theta}$. We use our adaptive algorithm to select a small subset of the points for constructing a GP emulator. We compare the results obtained with this restricted version of our algorithm to those in Walther et al. (2019) where all 75 points for (T_0, γ, λ_P) and several values of F were used for the GP construction. Note that we do *not* expect to obtain identical results, as, besides differences in implementation (Walther et al. (2019) build a GP emulator using the PCA-based approach of Habib et al. (2007), see below), we also use only Viel et al. (2013) subset of measurement data for the given redshift. The reason for this approach is our focus on the inference method. However, even using a "restricted" version of our adaptive algorithm, we are able to effectively constrain the



LHD 10 LHD 20 LHD 30

Figure 4. Distribution of the squared Mahalanobis distances for the four multi-output emulators (MS, IND, COR, INP) for three different LHDs computed with 500 test points. The reference distribution (ref) is χ^2_8. A single representative experiment is chosen for each of the three designs. The closer distributions are to the reference the better.



Adaptive designs. Fixed designs.

Figure 5. Hellinger distances $D_H\left(p^{ref}, p^{GP}\right)$ for the GP-based posteriors. While fixed-design-based emulators can occasionally produce good quality posteriors, the inconsistency of the results makes them a poor choice compared to adaptive designs.

parameters using only a fraction of the available inputs and simulation results.

5.1. *State-of-the-art data-agnostic approach*

Construction of emulators is often done using space-filling LHDs, such as maximin and orthogonal designs Moon et al. (2011), or sliced LHDs Qian (2012). Some of those space-filling LHDs are also used by the cosmological community for selecting the training points for the construction of emulators. Typically, the number of training points is selected a priori and remains fixed. For example, in the first such work in cosmology (Habib et al. 2007), the space-filling LHD with 128 points is selected for the problem of inferring 5 cosmological parameters from matter power spectrum measurements. The authors employ a PCA-based approach to constructing multi-output emulators. Specifically, they use a singular value decomposition (SVD) of the simulations at the training points specified by the LHD. The weights of the SVD are then modeled as independent GP emulators.

A similar approach is taken in Walther et al. (2019) for recovering thermal parameters of the IGM using the Lyα flux power spectrum. However, the training points do not exactly form an LHD, as one of the parameters, λ_P, is difficult to independently vary in a way that is not correlated with the other thermal state parameters T_0 and γ. This is because λ_P probes the integrated thermal history which is smooth for each individual physical model of heating and cooling of the IGM during and after reionization process. Of course, in principle one could generate models with abruptly changing instantaneous temperature such that the pressure smoothing does not have enough time to adjust, but we lack physical motivation for such models. In addition, one of parameters—the mean flux of the Lyα forest—is not part of the LHD as it can be easily rescaled in the post-

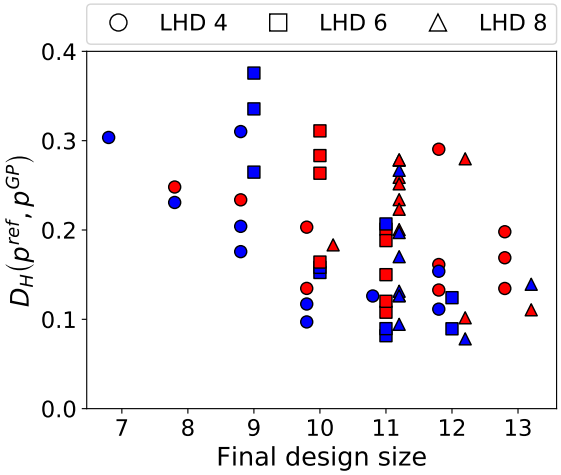


Figure 6. Hellinger distances for the posteriors obtained with the adaptive designs ordered by the size of the final designs. In blue - IND, in red - COR approaches. Different markers correspond to different initial design sizes.

processing, thus its sampling does not require running additional expensive simulations.

5.2. *Comparison of the adaptive method and the data-agnostic approach*

We continue to use the post-processing model described in Section A as the true "forward model" (with the same $q = 8$ outputs). We first generate the mock measurement by evaluating the model at a fixed $\boldsymbol{\theta}_{true} = (0.275, 1.245 \times 10^4 K, 1.6)^T$, and we corrupt the resulting measurement vector with noise from a multivariate Gaussian distribution $\mathcal{N}_q(\mathbf{0}_q, \sigma_e^2 \mathbf{I}_q)$ with $\sigma_e = 0.01$. This level of measurement noise is consistent with the observational data (Viel et al. 2013) that we use later.

In order to analyze how the size of the initial design for the adaptive algorithm influences the obtained solution, we perform experiments with 4, 6, or 8 design points in the three-dimensional parameter space $\boldsymbol{\theta} = (F, T_0, \gamma)$. In each iteration s of Algorithm 1, we construct a GP surrogate using the current selection of design points, we solve the auxiliary optimization problem to maximize the expected improvement in fit function, and augment the design set with a single point that provides the largest predicted improvement. We iterate until the relative expected improvement drops below a 1% threshold.

For each selection of the size of the initial training design, we run our algorithm 10 times, each time selecting new initial design points using a maximin LHD. For each of the initial designs, we run the algorithm with two emulator choices: IND and COR (see Section 4.2). For the COR emulator the inter-output correlation matrix $\boldsymbol{\Sigma}_k$ is taken to be the same as the one used in Section 4.3 (see Figure 1). On average the final designs contain 10–11 inputs regardless of the number of points in the initial design or the emulator choice.

We perform MCMC sampling of the posterior using the integrated likelihood based on the constructed GP surrogates (see Appendix B for details). To obtain a quantitative measure of the quality of the obtained posteriors, we compute the Hellinger distance between the GP-based posteriors $p^{GP}(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D})$ and the reference posterior $p^{ref}(\boldsymbol{\theta}|\mathbf{d})$ obtained by a direct MCMC sampling with the "true" likelihood function, i.e., the likelihood of the measurement data that uses evaluations of our (post-processing) forward model. The Hellinger distance is a metric for evaluating differences between two probability distributions and is a probabilistic analogue of the Euclidean distance. It can be related to other commonly used distance measures, such as total variation distance and Kullback-Leibler divergence, see, e.g., Dashti & Stuart (2016). It has also been recently studied in the context of posteriors obtained with Gaussian process emulators (Stuart & Teckentrup 2018). The Hellinger distance between p^{ref} and p^{GP} is defined as follows:

$$D_H(p^{ref}, p^{GP}) = \left(\frac{1}{2} \int_{\mathcal{X}_\theta} \left(\sqrt{p^{ref}(\boldsymbol{\theta}|\mathbf{d})} - \sqrt{p^{GP}(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D})} \right)^2 d\boldsymbol{\theta} \right)^{1/2}. \tag{15}$$

To compute $D_H(p^{ref}, p^{GP})$ we approximate the densities p^{ref} and p^{GP} by fitting kernel-density estimates (KDEs) with Gaussian kernels to the generated samples from the respective posteriors and discretize the integral in equation (15) using 3-dimensional Sobol' sequence with 10^4 points. The results for the posteriors obtained with the adaptive GPs (10 runs for each initial design) are presented in Figure 5 on the left.

We compare the posteriors obtained with the adaptive approach to the posteriors obtained by training GP models wtih fixed maximin LHDs. Here, we fix the design sizes to be 10, 11, and 12. As in the adaptive case, we train the GP emulators using both IND and COR approaches, and we repeat each experiment 10 times. The results for the posteriors obtained with the fixed designs are shown in Figure 5 on the right.

The comparison of the results in Figure 5 demonstrates the superiority of the adaptive approach. The adaptive approach is able to achieve results that are closer to the reference posterior in the Hellinger distance, and often with fewer design points. Furthermore, the results for the adaptive approach are less spread out, and thus making it more robust and consistent.

In Figure 6, we re-plot the data from Figure 5 for the adaptive cases, and we show the final design sizes on the x-axis. As we can see, in most cases, the final design size is either 10, 11, or 12. There is no significant difference in the results for different initial design sizes. However, there appears to be more variability in the final design sizes when the initial design contains only 4 points (LHD

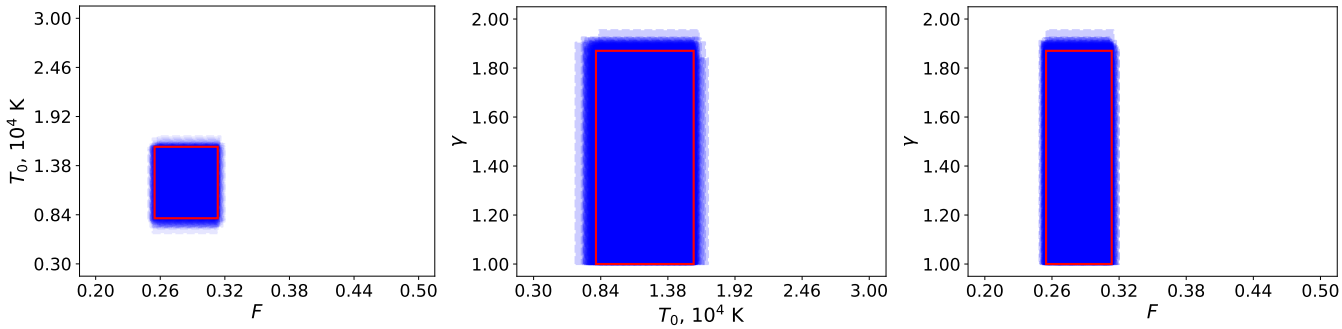


Figure 7. 95% HPD intervals for the marginal posteriors obtained with adaptive designs (blue rectangles). We plot intervals for two parameters at a time and overlay results from 60 experiments. The red rectangle represents the HPD intervals of the marginal distributions of the reference posterior. We observe a good correspondence for the intervals of obtained posteriors with those of the reference.

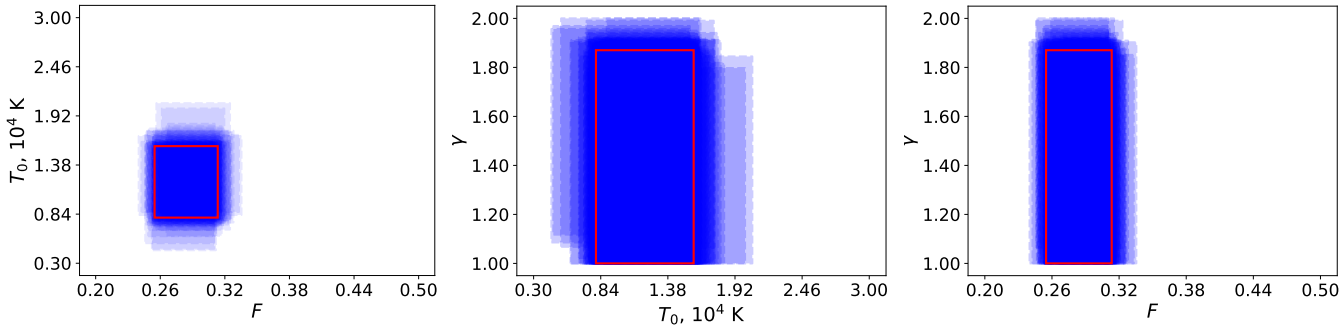


Figure 8. 95% HPD intervals for the marginal posteriors obtained with fixed designs (blue rectangles). We plot intervals for two parameters at a time and overlay results from 60 experiments. The red rectangle represents the HPD intervals of the marginal distributions of the reference posterior. We observe a noticeably larger scatter in the results between different experiments than in adaptive designs case presented in Figure 7.

4)—the final designs have between 7 to 13 points. We also observe a small trend of decreasing distance values as the final design size increases from 7 to 10. However, beyond 10 there is no significant difference in the results. The COR emulator appears less likely to terminate with a design consisting of less than 10 points, but in terms of the distance values, COR does not outperform the IND approach.

Results for the Hellinger distances confirm that there is little difference between the IND and the COR approaches for our current application. As far as the choice of the initial design size, starting with smaller designs (4 to 6 points for the current three-dimensional problem) leads to fewer forward model evaluations without compromising the quality of the result.

Additionally, we compare the posteriors obtained with the adaptive algorithm and with fixed LHDs by looking at the 95% highest posterior density (HPD) intervals for the marginal posteriors for each parameter, i.e., shortest intervals containing 95% of the marginal posteriors for each parameter. This provides a visual representation of the spread of the obtained distributions. We plot the intervals for two parameters at a time and overlay results from all 60 experiments (2 approaches $\times$ 3 initial designs $\times$ 10 random realizations) in Figures 7 (adaptive cases) and 8 (fixed cases). We observe a good correspondence between the HPD intervals for the adaptive designs. For the fixed designs there is much more variation in the spreads of the posterior estimates. This comparison reinforces the conclusion that the results obtained with fixed designs are inconsistent, and, therefore, less reliable.

5.3. *Results for the adaptive GP with post-processing model and Viel data*

In this section we apply our adaptive GP approach to the problem of inferring the same three parameters $\boldsymbol{\theta} = (F, T_0, \gamma)$ using the post-processing model of Section A as the forward model of the power spectrum, and data from Viel et al. (2013) for the redshift of $z = 4.2$.

The measurement data consists of seven values of the power spectrum for $k = \{5.01 \times 10^{-3}, 7.95 \times 10^{-3}, 1.26 \times 10^{-2}, 1.99 \times 10^{-2}, 3.16 \times 10^{-2}, 5.01 \times 10^{-2}, 7.95 \times$

$10^{-2}\}\mathrm{km}^{-1}\mathrm{s}$ with errorbars that we treat as $\pm 1\sigma_k$. The measurement noise covariance, thus, has a diagonal form $\mathbf{\Sigma}_E = \mathrm{diag}[\sigma_1^2, \ldots, \sigma_7^2]$. We take a uniform (flat) prior $p(\boldsymbol{\theta})$ on all three parameters defined over the same box $\mathcal{X}_\theta$ as in Section 4.3:

$$\mathcal{X}_\theta = [0.2, 0.5] \times [3 \times 10^3 K, 3 \times 10^4 K] \times [1.0, 2.0]. \quad (16)$$

We use the COR emulator with the same $\mathbf{\Sigma}_k$ as in Section 4.3 and initialize Algorithm 1 with a maximin LHD with 4 points. The stopping threshold for the algorithm is again set to 1%.

Figure 9 shows the design points at different iterations of the adaptive algorithm. The first figure shows the initial 4 points arranged in maximin LHD, the figure in the middle has three additional points after three iterations of the adaptive algorithm, and the last figure shows the final design upon termination.

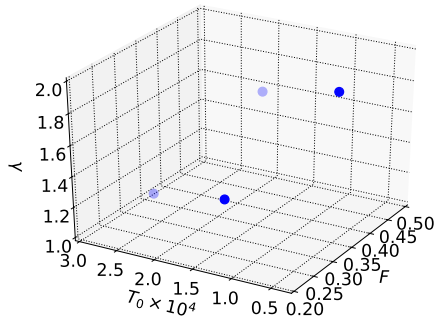
Figure 10 shows iteration history of the algorithm with iteration $s = 1$ corresponding to the initial design with four points. The blue line in this figure shows the value of the best misfit for the points in the training set in each iteration s, g^s_{min}, scaled by the best misfit value obtained upon convergence, g^{best}_{min}. A point with a better misfit value is not obtained in every iteration, e.g., in iterations 2 and 3 we have the same g_{min} as initially. The points added to the design in these iterations serve the purpose of decreasing the overall uncertainty of the GP. The new point added to the training set $\mathcal{D}$ after iteration $s = 3$ provides a reduction in g_{min} for the first time (see red dot in the middle figure in Figure 9). The red dotted line in Figure 10 shows one minus the relative expected improvement in each iteration, i.e., $1 - \mathcal{I}(\boldsymbol{\theta}^s)/g^s_{min}$. As the algorithm progresses, we expect both lines to approach 1.

Figure 11 shows the power spectrum $\mathbf{P}(\boldsymbol{\theta}^{s=6})$ evaluated at the last $\boldsymbol{\theta}$ added by the algorithm in iteration $s = 6$. This point corresponds to the smallest misfit g^{best}_{min} to the Viel data found by our algorithm. This point is shown as a red dot in the bottom panel of Figure 9.

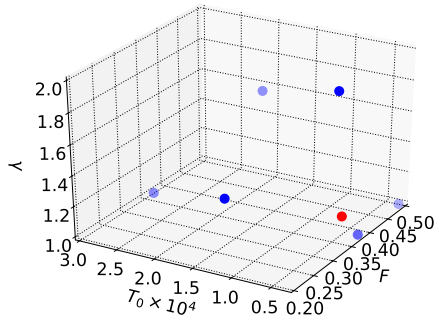
5.4. *Results for the adaptive GP with Nyx simulations and Viel data*

In this section we work with the THERMAL suite of Nyx simulations consisting of 75 models with given parameters (T_0, γ, λ_P) (see Figure 12). Mean flux we treat in post-processing as in virtually all Lyα power spectrum inference works. To be specific, in this work we produce 40 equidistant values for the parameter F in the interval $[0.2, 0.5]$ for each of the 75 thermal models, thereby sampling the 4-dimensional parameter space $\boldsymbol{\theta} = (F, T_0, \gamma, \lambda_P)$.

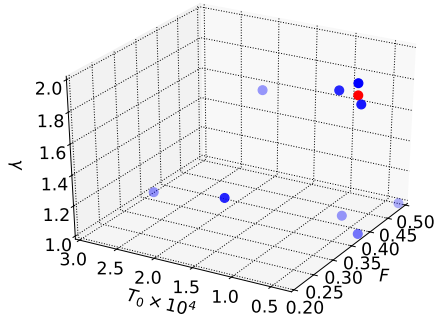
We run a restricted version of Algorithm 1 with the possible selection of $\boldsymbol{\theta}$ limited to the existing thermal models. We start by selecting the first six points randomly, build a GP emulator using the IND approach, and evaluate the expected improvement in fit function



Initial design



Design after iteration $s = 3$



Final design

Figure 9. Evolution of the designs for the adaptive GP using the post-processing model and Viel data. We have achieved desired convergence with 10 evaluations of the forward model.

$\mathcal{I}(\boldsymbol{\theta})$ for the remaining points (using Viel data). We select the point that corresponds to the largest improvement in fit, and iterate until no further improvement can be made. In this regime we do not perform a direct optimization over the parameter space but select the inputs out of the available THERMAL data. Our restricted algorithm terminates after iteration $s = 4$ using a total of 10 design points. Figure 13 shows the plot demonstrating the fit of the $\mathbf{P}(\boldsymbol{\theta}^{s=4})$ corresponding to

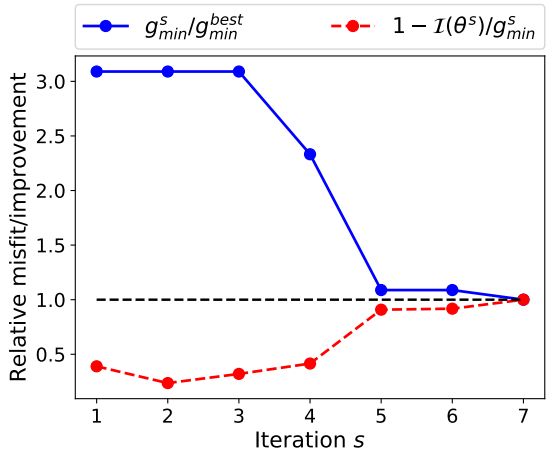


Figure 10. Iteration history of Algorithm 1 applied to post-processing model with Viel data.

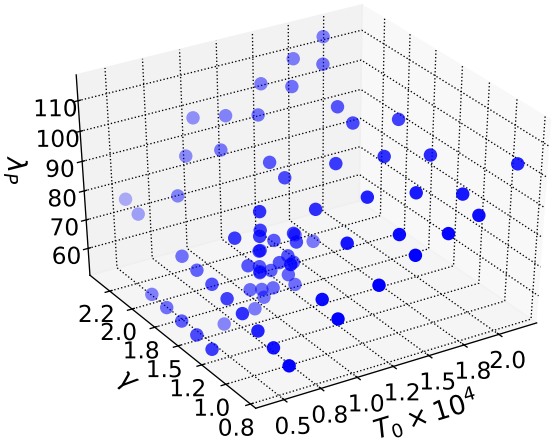


Figure 12. All 75 simulated models in the THERMAL suite.

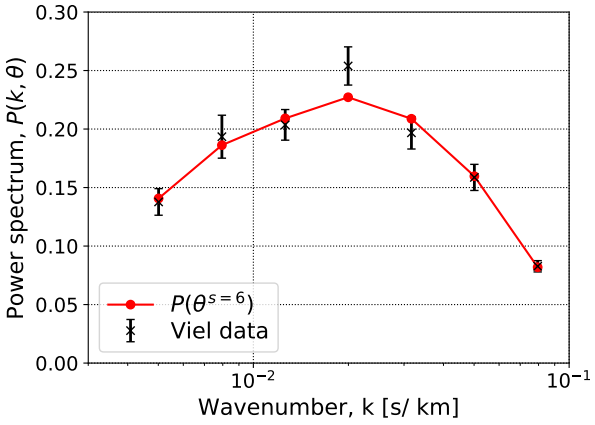


Figure 11. $\mathbf{P}(\boldsymbol{\theta}^{s=6})$ corresponding to the best found misfit for the adaptive GP with post-processing model and Viel data.

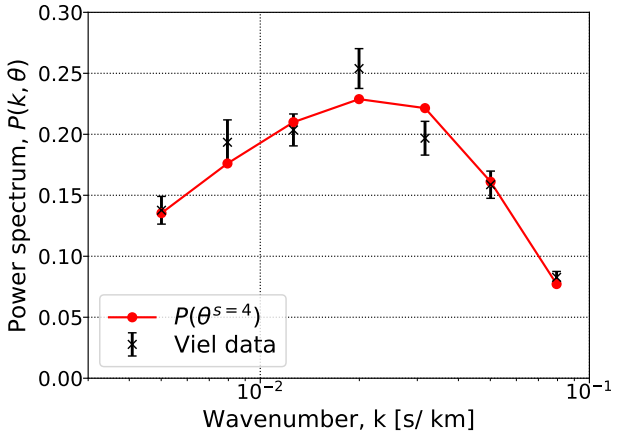


Figure 13. $\mathbf{P}(\boldsymbol{\theta}^{s=4})$ corresponding to the best found misfit value for the adaptive GP with Nyx simulations and Viel data.

the best found misfit value. Note that here we used Nyx simulations as the forward model on a pre-existing set of discrete points to chose from, and we are reaching convergence in similar number of evaluations as in Section 5.3 when we did continuous search for a candidate evaluation of the rescaled model (purely by coincidence, the number of evaluations in both cases was actually identical: 10).

The other important ingredient is the construction of the prior $p(\boldsymbol{\theta})$. We use a flat prior for F, $\log T_0$, γ, and $\log \lambda_P$ in a box constrained by the smallest and the largest values for each parameter. We then truncate this prior to the convex hull of the THERMAL grid points, as is done in Walther et al. (2019). The resulting truncated prior is shown in Figure 14. This truncation is done to avoid GP extrapolation into a region of parameter space where this IGM model cannot produce an answer, for example, in case of very low T_0 but very high pressure smoothing scale (see also discussion in Section 5.1 about the λ_P parameter).

The posterior obtained with this prior using the likelihood based on the adaptively constructed GP is shown in Figure 15. We observe that the resulting posterior is considerably more constrained than the prior, although we want to draw the reader's attention to the poor constraints on parameter γ, which plays very little role at high redshifts (at most!) when the density of Lyα absorbing gas is close to the cosmic mean. Overall, the marginal ranges and central values for the parameters are in good agreement with the ones reported in Walther et al. (2019). Note, however, that we do not use BOSS (Palanque-Delabrouille et al. 2013) measurements here, but only Viel et al. (2013) data, as we want to avoid modeling of correlated Silicon III absorption which is present in the BOSS dataset. We prefer maintaining the forward model as simple as possible

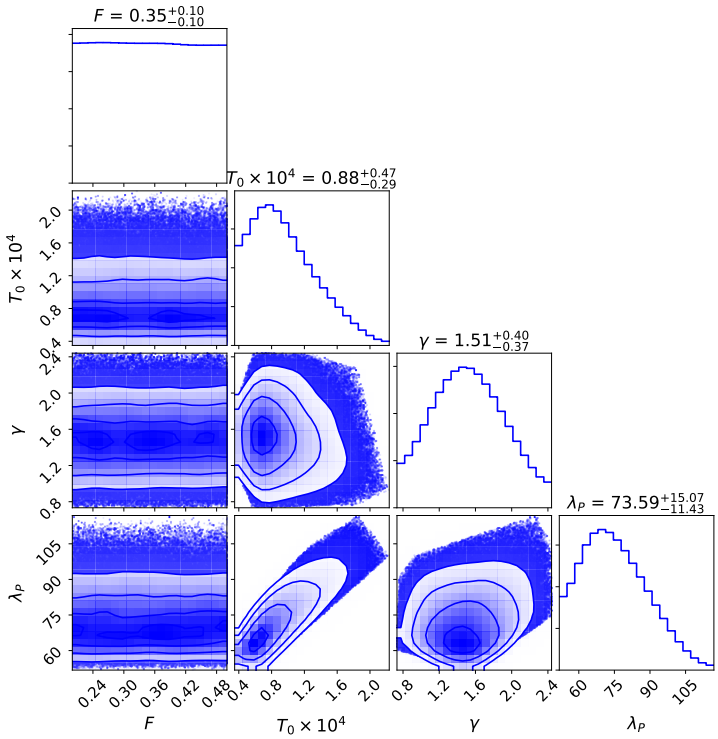


Figure 14. Prior $p(\boldsymbol{\theta})$ for $\boldsymbol{\theta} = (F, T_0, \gamma, \lambda_P)$ à la Walther et al. (2019)

as the goal of this work is testing and improving inference schemes for the Lyα power spectrum. In any case, the fact that our results presented here are completely consistent with Walther et al. (2019) indicates that the BOSS dataset contributes negligible information about the thermal state of the IGM at high redshifts.

6. CONCLUSIONS

In this work we described the use of an adaptive design of GP emulators of the Lyman α flux power spectrum for solving inference problems for the thermal parameters of the IGM. To the best of our knowledge, while GP emulators constructed from a Latin hypercube design are nowadays in common use in the cosmological community, the data-driven adaptive selection of training inputs has been considered only very recently in Rogers et al. (2019) and in our work. In the future, we expect to see wider use of similar techniques in other astrophysical applications where observational data already exists prior to the construction of an emulator.

Our motivation for this work is primarily the reduction of the number of computationally intensive simulations required to build a GP emulator. By prioritizing the regions of the parameter space that are consistent with the measurement data under the predictive model of the emulator, we obtain the desired reduction without sacrificing the quality of the parameter posteriors. A numerical study that we performed on a problem with an approximate model of the Lyman α forest power spectrum and with synthetic measurement data demonstrated that our adaptive approach obtains consistently good approximations of the parameter posterior and outperforms a similar-size fixed design approach based on maximin Latin hypercube designs.

We provided a complete framework for building multi-output GP emulators that predict the power spectrum at the pre-selected modes k. Our numerical study demonstrates that the resulting multi-output emulators that either treat outputs as conditionally independent given the hyperparameters (IND) or explicitly model linear correlations between the outputs (COR) are effective and computationally efficient. Furthermore, our approaches allow us to train emulators using only highly limited number of training inputs, which in turn enables the adaptive selection of additional inputs.

The initial results obtained with our adaptive approach are encouraging. Specifically, for the problem of inferring three thermal parameters of the IGM and mean flux using measurements of the power spectrum at seven values of k our approach (constrained to the 75 available Nyx THERMAL simulations) required simulation outputs for only 10 input values to constrain the parameters to the same level of accuracy as in Walther et al. (2019) that used substantially larger number of simulations.

Finally, we want to emphasize that we do not consider the "classical" parameterization of $\boldsymbol{\theta} = (F, T_0, \gamma, \lambda_P)$ to be the best for modeling the state of the IGM, but we nevertheless perform this type of analysis as it is straightforward to make comparisons of our results with

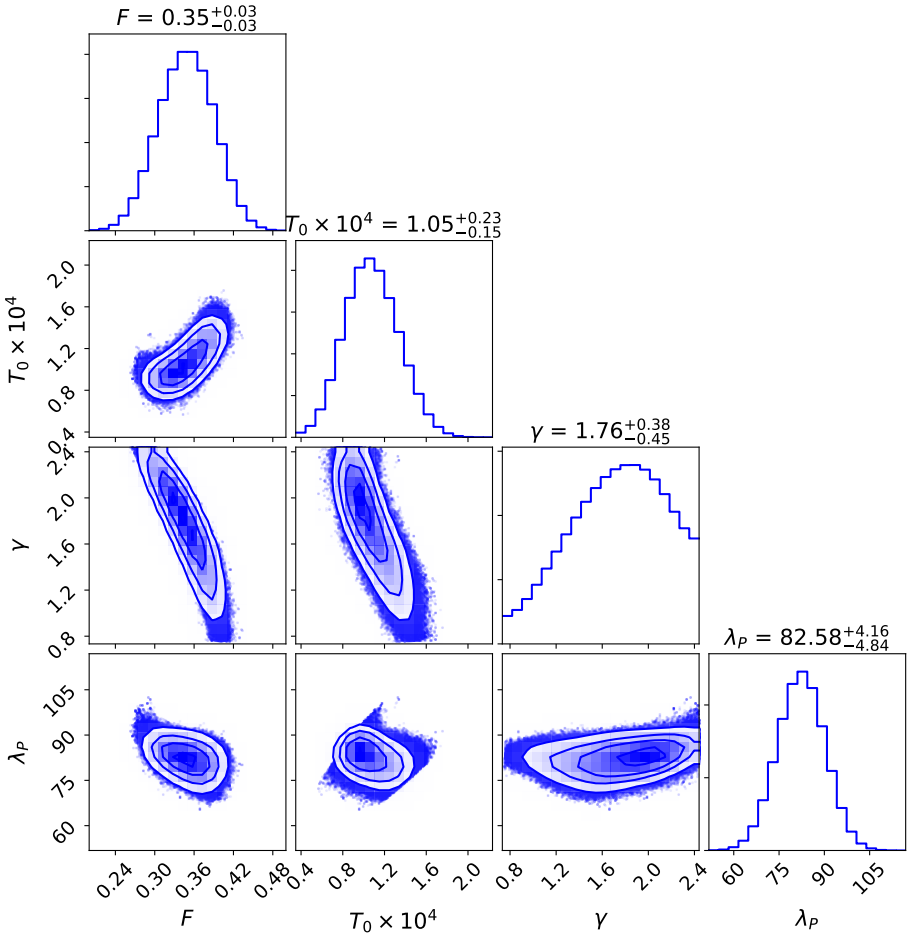


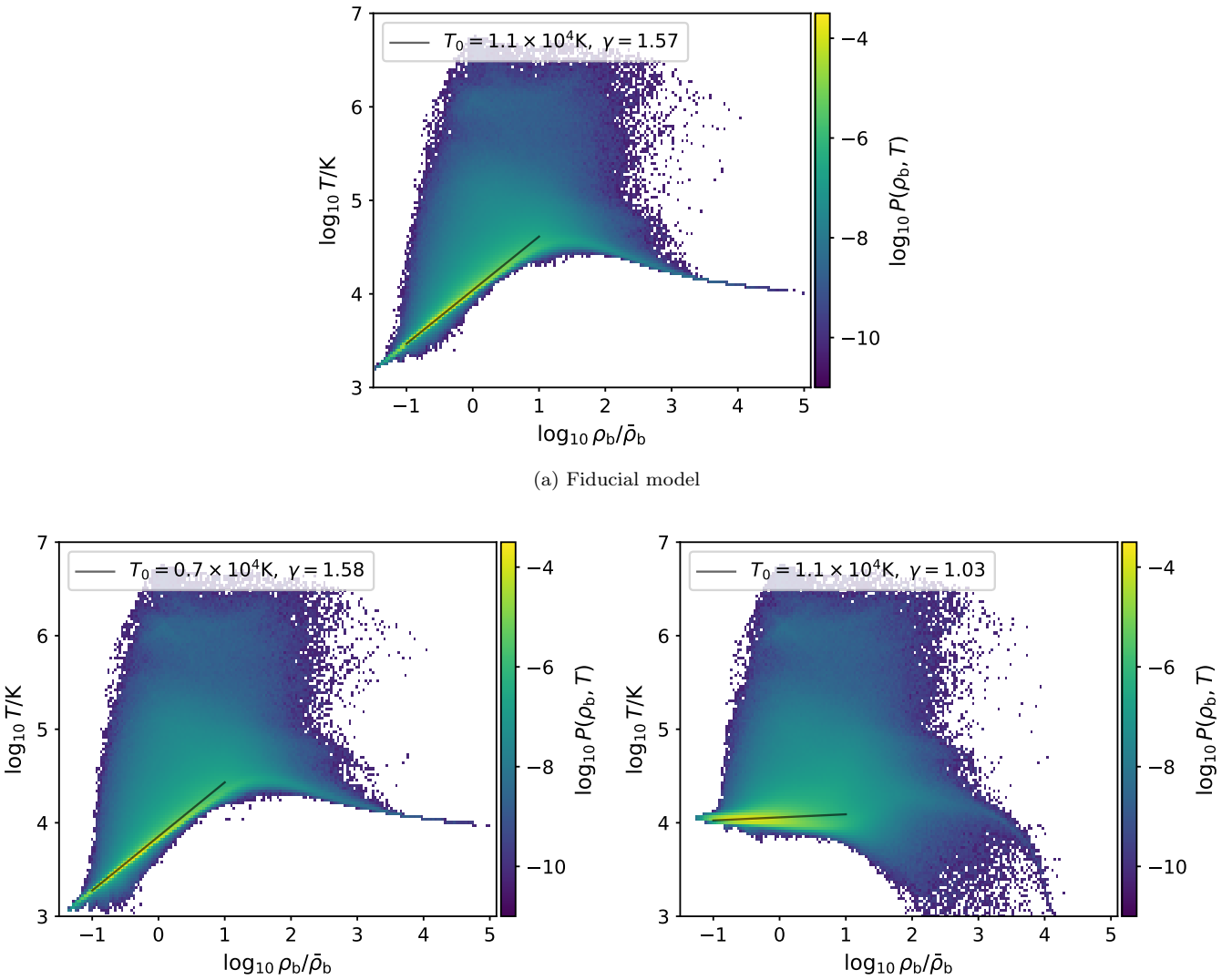
Figure 15. 1D and 2D marginal posteriors for $\boldsymbol{\theta} = (F, T_0, \gamma, \lambda_P)$ obtained with a restricted version of Algorithm 1 using Nyx simulation and Viel data. Note that we apply smoothing to the plots of the marginal histograms which makes them look more Gaussian. The numbers above 1D histograms report 50%-quantiles of the marginal distributions plus/minus differences between 84%- and 50%-quantiles and 50%- and 16%-quantiles.

previous works. While these parameters have intuitive physical meaning in describing the thermodynamical state of the IGM, there are several practical problems with them. First, they are *output* rather than *input* parameters which brings significant difficulties with implementations of sampling and iterative emulation procedure. Second, these 4 parameters are parameterizing *each time snapshot* instead of *the physical model* itself. For that reason, we consider models which parameterize the time and duration of the reionization as well as associated heat input (Oñorbe et al. 2019) as better and we will be using those in future works.

Authors are grateful to Jose Oñorbe for making his Nyx simulations available to us, as well as for providing helpful comments and insights. We thank Joe Hennawi and members of the Enigma group[2] at UC Santa Barbara for insightful suggestions and discussions. This research used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract no. DE-AC02-05CH11231. This work made extensive use of the NASA Astrophysics Data System and of the astro-ph preprint archive at arXiv.org.

Software: Nyx (Almgren et al. 2013)

[2] http://enigma.physics.ucsb.edu/



(a) Fiducial model

(b) Low-T_0 model

(c) Low-γ model

Figure 16. The density-temperature distribution of gas (volume-weighted histogram) in three Nyx simulations: (a) "fiducial" one, (b) simulation with lower T_0 value than in the fiducial one, and (3) simulation with lower γ value. Other than one parameter differing, simulations have the same all other parameters, including pressure smoothing scale at this redshift, λ_P.

APPENDIX

A. RESCALING OF THERMAL PARAMETERS

Parameter space considered in this work consists of four "standard" parameters, $\{T_0, \gamma, \lambda_P, \bar{F}\}$, describing the thermal state of the IGM. In this appendix we specify on rescaling of those parameters, which is as model used in Section 5.3. The advantage of this approximate model is that it does not require producing new simulation for every new evaluated point.

In photo-ionization equilibrium, the mean flux of the Lyα forest is proportional to the fraction of neutral hydrogen, and is thus degenerate with the amplitude of the assumed UV background. Therefore, mean flux can be rescaled by finding the constant multiplier of the optical depth of all spectral pixels in the simulation so that the mean flux matches the desired value: $\bar{F} = \langle \exp(-A\tau_{i,j,k}) \rangle$. For accuracy considerations of rescaling the mean flux, we refer the reader to Lukić et al. (2015).

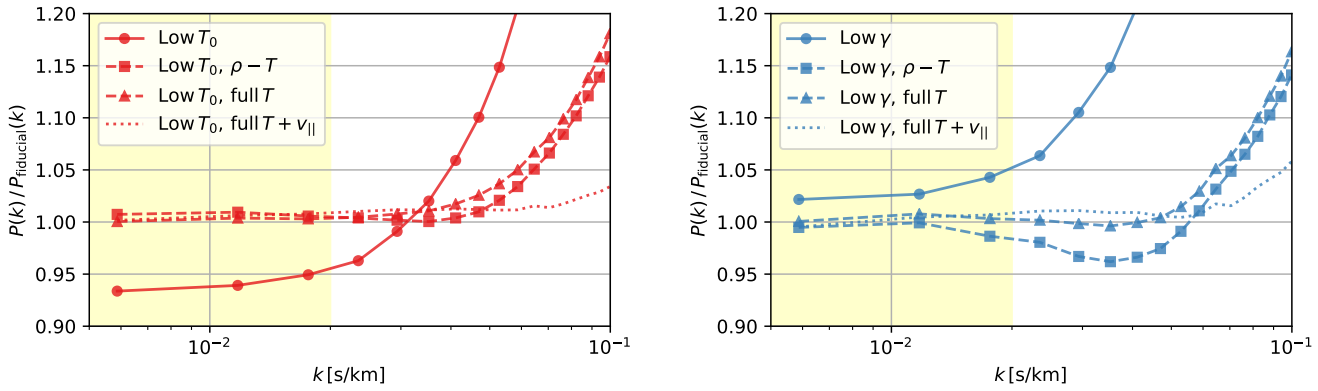


Figure 17. Power spectra ratios showing the accuracy of different post-processing approaches for rescaling the instantaneous temperature in simulations. Solid lines show ratios of low-T_0 (left panel) and low-γ (right panel) simulations with respect to the fiducial simulation. Dashed lines lines show same ratios after rescaling simulations to match fiducial one's T_0–γ relation assuming all the gas is exactly on the power law (squares) and accounting for the scatter in T_0–γ (triangles). Dotted lines additionally take line of sight velocity from the fiducial simulation demonstrating that most of the remaining rescaling error is coming from gas elements moving at different speeds. Note that over the BOSS/eBOSS/DESI range of k (yellow region), this rescaling shows good accuracy.

Modifying any of the other three thermal parameter commonly requires running a new simulation (see, e.g. Walther et al. (2019)). While modifying λ_P (3D pressure smoothing) is inherently difficult due to its dependence on the whole thermal history, we can hope to be also able to modify the instantaneous temperature – the one that determines the recombination rate and the thermal broadening (1D smoothing). That way, we can generate different values of $\{T_0$, γ, $\bar{F}\}$, without the need to re-run expensive simulations. To test this, we use three simulations which have the same cosmological and numerical parameters, and yield the same pressure smoothing parameter $\lambda_p \approx 68$kpc at redshift $z = 3$. Temperature-density diagrams for these simulations are shown in Figure 16. The fiducial simulation has $T_0 = 1.1 \times 10^4$K and $\gamma = 1.57$; the "low-T_0" simulation differs from fiducial only in that $T_0 = 7 \times 10^3 K$, while the "low-γ" simulation has $\gamma = 1.03$ and all other parameters the same as the fiducial model.

In Figure 17 we show power spectra ratios of low-T_0 and low-γ simulations with respect to the fiducial model. Mean flux is matched in all cases shown. Solid lines (with circles) are ratios of unscaled simulations, and we can see they are significantly different over the k range covered by data ($k \lesssim 0.08\,\text{km}^{-1}s$). Two dashed lines with square and triangle points are models where temperature-density relation has been rescaled to the fiducial one without and with accounting for the scatter in the $T - \rho$. We notice the significant improvement in power spectrum, and we can also see that scatter in $T - \rho$, as expected from optically thin models, does not play a significant role, although it help in a case of radical change in γ parameter as seen in the right panel of Figure 17. Finally, dashed line represent the case where we both rescale temperature-density relation, and use line of sight velocity from the fiducial simulation. This is not a practical solution, as we wouldn't know these velocities when rescaling a simulation to a target $T - \rho$ relation, but we want to show that differences in velocity account for most of the remaining error in this rescaling procedure.

While our approximate, post-processing model does not recover power spectrum at a percent accurate level over the whole range of available data, it is sufficiently accurate for experiments conducted in Section 4. The essential requirements there are to know the "true" answer for a given model, and to be able to evaluate the model a large number of times. Note also that this rescaling procedure is loosing accuracy at high-k end which is important for thermal constraints and interpreting $P(k)$ from high resolution spectra, but over the k range relevant to BOSS/eBOSS/DESI observations ($k \lesssim 0.02\,\text{km}^{-1}s$), the achieved accuracy is ≈1%, which should suffice for many studies.

B. TRAINING GP EMULATORS

Let $\boldsymbol{\theta}^{(j)}$, $j = 1, \ldots, n_{train}$, represent training inputs with $\mathbf{P}\big(\boldsymbol{\theta}^{(j)}\big)$ being a q-vector of outputs corresponding to a particular input. Denote by $\mathbf{y}_i$ the n_{train}-vector of the normalized values of the i-th output, $i = 1, \ldots, q$, defined as follows

$$\mathbf{y}_i = \left(\widehat{P}_i\big(\boldsymbol{\theta}^{(1)}\big), \ldots, \widehat{P}_i\big(\boldsymbol{\theta}^{(n_{train})}\big)\right)^T, \tag{B1}$$

where

$$\widehat{P}_i(\boldsymbol{\theta}) = \frac{P_i(\boldsymbol{\theta}) - m_i}{\mathbb{V}_i^{1/2}} \tag{B2}$$

with

$$m_i = \frac{1}{n_{train}} \sum_{j=1}^{n_{train}} P_i\big(\boldsymbol{\theta}^{(j)}\big), \quad \mathbb{V}_i = \frac{1}{n_{train}} \sum_{j=1}^{n_{train}} \Big(P_i\big(\boldsymbol{\theta}^{(j)}\big) - m_i\Big)^2. \tag{B3}$$

The normalized training outputs together form an output matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_q] \in \mathbb{R}^{n_{train} \times q}$. Finally, the vectorized form of $\mathbf{Y}$ is obtained by stacking the normalized training outputs into a $(n_{train} \cdot q)$-vector $\overline{\mathbf{y}} = \mathrm{vec}(\mathbf{Y})$.

The set of all training inputs we denote by $\boldsymbol{\theta}_{train} = \{\boldsymbol{\theta}^{(j)}, j = 1, \ldots, n_{train}\}$, and the combined set of training inputs and outputs, or training data, by $\mathcal{D} = \{\boldsymbol{\theta}_{train}, \mathbf{Y}\}$.

Training a GP emulator requires specifying the hyperparameters $\boldsymbol{\psi}$ of its kernel. These are characterised by the posterior distribution (note that conditioning on $\boldsymbol{\Sigma}_k$ is implicit but not necessary since it's fixed)

$$p(\boldsymbol{\psi} \,|\, \mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(\mathcal{D})}, \tag{B4}$$

where

$$p(\mathcal{D}|\boldsymbol{\psi}) = p(\mathbf{Y}|\boldsymbol{\theta}_{train}, \boldsymbol{\psi}) = \mathcal{N}_{n_{train} \times q}(\mathbf{Y}|\boldsymbol{\mu}^{norm}(\boldsymbol{\theta}_{train}), c(\boldsymbol{\theta}_{train}, \boldsymbol{\theta}_{train}; \boldsymbol{\psi}), \boldsymbol{\Sigma}_k^{norm}) \tag{B5}$$

is the likelihood of the training data $\mathcal{D}$ under the matrix-normal distribution defining the Gaussian process (see Section 4) and $p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\psi}) p(\boldsymbol{\psi}) d\boldsymbol{\psi}$ is referred to as *evidence*. Note that $\boldsymbol{\mu}^{norm}$ is a normalized version of the mean function obtained by applying the linear transformation (B2), and $\boldsymbol{\Sigma}_k^{norm}$ is the inter-output correlation matrix. In order to somewhat simplify this notation let us denote the covariance matrix for the training inputs by $\mathbf{C}_\psi = c(\boldsymbol{\theta}_{train}, \boldsymbol{\theta}_{train}; \boldsymbol{\psi})$. Also, recall that we take $\boldsymbol{\mu}(\cdot) \equiv 0$. Thus, we have

$$p(\mathbf{Y}|\boldsymbol{\theta}_{train}, \boldsymbol{\psi}) = \mathcal{N}_{n_{train} \times q}(\mathbf{Y}|\mathbf{0}_{n_{train} \times q}, \mathbf{C}_\psi, \boldsymbol{\Sigma}_k^{norm}). \tag{B6}$$

In a vectorized form we can express the likelihood above as a regular multivariate normal density

$$p(\overline{\mathbf{y}}|\boldsymbol{\theta}_{train}, \boldsymbol{\psi}) = \mathcal{N}_{n_{train} \cdot q}(\overline{\mathbf{y}}|\mathbf{0}_{n_{train} \cdot q}, \boldsymbol{\Sigma}_k^{norm} \otimes \mathbf{C}_\psi). \tag{B7}$$

How do we obtain the hyper-posterior (B4)? Since no analytical form for this posterior exist, we describe it via a *particle approximation* (Bilionis & Zabaras 2016, Section 2.6). That is we approximate the hyper-posterior with a weighted sum of Dirac delta functions centered at samples $\boldsymbol{\psi}^{(j)}$:

$$p(\boldsymbol{\psi}|\mathcal{D}) \approx \sum_{j=1}^{n_\psi} w^{(j)} \delta(\boldsymbol{\psi} - \boldsymbol{\psi}^{(j)}) \tag{B8}$$

with weights $w^{(j)} \geq 0$ and $\sum_{j=1}^{n_\psi} w^{(j)} = 1$.

One way to obtain such a particle approximation is by maximizing the likelihood of the data given by (B7). This leads to a single-particle approximation

$$p(\boldsymbol{\psi}|\mathcal{D}) \approx \delta(\boldsymbol{\psi} - \boldsymbol{\psi}^*_{MLE}), \tag{B9}$$

where

$$\boldsymbol{\psi}^*_{MLE} = \underset{\boldsymbol{\psi} \in \mathcal{X}_\psi}{\arg\max}\, p(\mathcal{D}|\boldsymbol{\psi}) \tag{B10}$$

is the maximum likelihood estimator (MLE) of the hyperparameter vector. In the case of a flat prior on the hyperparameters this estimator coincides with a maximum a posteriori (MAP) estimator. MLE approach is convenient to work with, since the covariance matrix $\mathbf{C}_\psi$ needs to be only formed once, however, it might lead to somewhat over-confident estimates of predictive uncertainties of the GP emulator . In the case of a sharply peaked likelihood $p(\mathcal{D}|\boldsymbol{\psi})$ the MLE estimator can be sufficient. Another way of obtaining the particle approximation of the hyper-posterior is by sampling it using MCMC techniques. This way provides a more complete picture of the hyper-posterior, albeit at an additional computational cost.

Whether using MLE or MCMC approach to obtaining hyper-posterior $p(\boldsymbol{\psi}|\mathcal{D})$, we need to be able to evaluate the logarithm of the likelihood function (B7) (note that by applying logarithm we preserve the order relation and obtain a

better-behaved function). In the following we derive the expression for the log-likelihood of the data and explain how it can be efficiently computed.

Let $\mathbf{\Sigma}_{tot} = \mathbf{\Sigma}_k^{norm} \otimes \mathbf{C}_\psi$ and let $a_{i,j}$ denote the entries of $(\mathbf{\Sigma}_k^{norm})^{-1}$, then

$$\begin{aligned}
\log p(\overline{\mathbf{y}} \,|\, \boldsymbol{\theta}_{train}, \boldsymbol{\psi}) &= -\frac{1}{2}\overline{\mathbf{y}}^T \mathbf{\Sigma}_{tot}^{-1}\overline{\mathbf{y}} - \frac{1}{2}\log|\mathbf{\Sigma}_{tot}| - \frac{n_{train}\cdot q}{2}\log(2\pi) \\
&= -\frac{1}{2}\sum_{i=1}^{q}\sum_{j=1}^{q} a_{i,k}\mathbf{y}_i^T\mathbf{C}_\psi^{-1}\mathbf{y}_j - \frac{1}{2}\big(n_{train}\log|\mathbf{\Sigma}_k^{norm}| + q\log|\mathbf{C}_\psi|\big) \\
&\quad - \frac{n_{train}\cdot q}{2}\log(2\pi) \\
&= -\frac{1}{2}\mathrm{tr}((\mathbf{\Sigma}_k^{norm})^{-1}\mathbf{Y}^T\mathbf{C}_\psi^{-1}\mathbf{Y}) - \frac{1}{2}\big(n_{train}\log|\mathbf{\Sigma}_k^{norm}| + q\log|\mathbf{C}_\psi|\big) \\
&\quad - \frac{n_{train}\cdot q}{2}\log(2\pi).
\end{aligned} \tag{B11}$$

In our implementation we first compute the Cholesky decomposition of the input covariance

$$\mathbf{C}_\psi = \mathbf{L}\mathbf{L}^T, \tag{B12}$$

and let

$$\mathbf{A} = \mathbf{L}^T \backslash (\mathbf{L} \backslash \mathbf{Y}), \tag{B13}$$

then compute

$$\mathbf{B} = \mathbf{Y}^T\mathbf{A}, \tag{B14}$$

and set

$$\mathbf{D} = \mathbf{S}^T \backslash (\mathbf{S} \backslash \mathbf{B}), \tag{B15}$$

where $\mathbf{\Sigma}_k^{norm} = \mathbf{S}\mathbf{S}^T$ is the Cholesky decomposition of the output correlation matrix. Then

$$\log p(\overline{\mathbf{y}} \,|\, \boldsymbol{\theta}_{train}, \boldsymbol{\psi}) = -\frac{1}{2}\left(\mathrm{tr}(\mathbf{D}) + 2n_{train}\sum_{i=1}^{q}\log(\mathbf{S}_{i,i}) + 2q\sum_{i=1}^{n_{train}}\log(\mathbf{L}_{i,i}) + qn_{train}\log(2\pi)\right). \tag{B16}$$

The expression above can be further simplified in certain cases. For example, for the IND emulator that treats outputs as independent given the $\mathbf{C}_\psi$ matrix, the output correlation matrix in a unitary matrix $\mathbf{\Sigma}_k^{norm} = \mathbf{I}_q$, and the computation of $\mathrm{tr}(\mathbf{D})$ does not require cross-terms $\mathbf{y}_i\mathbf{C}_\psi^{-1}\mathbf{y}_j$ for $i \neq j$. In the case of IND emulator, the log-likelihood thus simplifies to:

$$\log p(\overline{\mathbf{y}} \,|\, \boldsymbol{\theta}_{train}, \boldsymbol{\psi}) = -\frac{1}{2}\left(\mathrm{tr}(\mathbf{B}) + 2q\sum_{i=1}^{n_{train}}\log(\mathbf{L}_{i,i}) + qn_{train}\log(2\pi)\right). \tag{B17}$$

In our implementation, the optimal hyper-parameter values $\boldsymbol{\psi}^*_{MLE}$ are obtained by maximizing the above log-likelihood using a multi-start strategy with an quasi-Newton iterative nonlinear optimizer such as Sequential Least Squares Programming (SLSQP) or Limited memory Broyden-Fletcher-Goldfarb-Shannon (L-BFGS-B)[3].

C. OBTAINING PREDICTIONS

In order to obtain a prediction for an un-tried input $\boldsymbol{\theta}$, we apply the standard GP formulas obtained by conditioning on the data $\mathcal{D}$. Furthermore, by exploiting the Kronecker product structure of the covariance, we can apply standard GP formulas to each output separately. Indeed, as shown in Bonilla et al. (2008),

$$\begin{aligned}
\mathbf{m}^{norm}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) &= (\mathbf{\Sigma}_k^{norm} \otimes \mathbf{c}_\psi)^T(\mathbf{\Sigma}_k^{norm} \otimes \mathbf{C}_\psi)^{-1}\overline{\mathbf{y}} \\
&= ((\mathbf{\Sigma}_k^{norm})^T \otimes \mathbf{c}_\psi^T)((\mathbf{\Sigma}_k^{norm})^{-1} \otimes \mathbf{C}_\psi^{-1})\overline{\mathbf{y}} \\
&= ((\mathbf{\Sigma}_k^{norm}(\mathbf{\Sigma}_k^{norm})^{-1}) \otimes (\mathbf{c}_\psi^T\mathbf{C}_\psi^{-1}))\overline{\mathbf{y}} \\
&= (\mathbf{c}_\psi^T\mathbf{C}_\psi^{-1}\mathbf{y}_1, \ldots, \mathbf{c}_\psi^T\mathbf{C}_\psi^{-1}\mathbf{y}_q)^T \\
&= (m(\boldsymbol{\theta}; \mathcal{D}_1, \boldsymbol{\psi}), \ldots, m(\boldsymbol{\theta}; \mathcal{D}_q, \boldsymbol{\psi}))^T \in \mathbb{R}^q,
\end{aligned} \tag{C18}$$

[3] Convenient implementations of both algorithms exist in Python's SciPy library.

where superscript *norm* indicates that this is the predictive mean of the GP fitted to the normalized outputs, and $\mathbf{c}_\psi = c(\boldsymbol{\theta}, \boldsymbol{\theta}_{train}; \boldsymbol{\psi}) \in \mathbb{R}^{n_{train}}$. For the predictive covariance we get

$$\begin{aligned}\boldsymbol{\Sigma}_{GP}^{norm}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) &= c(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\psi})\boldsymbol{\Sigma}_k^{norm} - (\boldsymbol{\Sigma}_k^{norm} \otimes \mathbf{c}_\psi)^T((\boldsymbol{\Sigma}_k^{norm})^{-1} \otimes \mathbf{C}_\psi^{-1})(\boldsymbol{\Sigma}_k^{norm} \otimes \mathbf{c}_\psi) \\ &= (c(\boldsymbol{\theta}, \boldsymbol{\theta}; \boldsymbol{\psi}) - \mathbf{c}_\psi^T \mathbf{C}_\psi^{-1} \mathbf{c}_\psi)\boldsymbol{\Sigma}_k^{norm} = \mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})\boldsymbol{\Sigma}_k^{norm}. \end{aligned} \tag{C19}$$

Upon re-scaling we obtain:

$$\mathbf{P}^{GP}(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\psi}, \boldsymbol{\Sigma}_k \sim \mathcal{N}_q(\mathbf{P}^{GP}(\boldsymbol{\theta})|\mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}), \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})) \tag{C20}$$

with

$$\mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) = (\mathbb{V}_1^{1/2}\, m(\boldsymbol{\theta}; \mathcal{D}_1, \boldsymbol{\psi}) + m_1, \ldots, \mathbb{V}_q^{1/2}\, m(\boldsymbol{\theta}; \mathcal{D}_q, \boldsymbol{\psi}) + m_q)^T, \tag{C21}$$

and

$$\begin{aligned}\boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) &= \mathbf{V}^{1/2}\boldsymbol{\Sigma}_{GP}^{norm}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})\mathbf{V}^{1/2} \\ &= \mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})(\mathbf{V}^{1/2}\boldsymbol{\Sigma}_k^{norm}\mathbf{V}^{1/2}) \\ &= \mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})\boldsymbol{\Sigma}_k, \end{aligned} \tag{C22}$$

where $\mathbf{V} = \mathrm{diag}[\mathbb{V}_1, \ldots, \mathbb{V}_q] \in \mathbb{R}^{q \times q}$. Finally, integrating out the hyperparameters $\boldsymbol{\psi}$ (recall the particle approximation (B8)) we obtain

$$\mathbf{P}^{GP}(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\Sigma}_k \sim \sum_{j=1}^{n_\psi} w^{(j)} \mathcal{N}_q\big(\mathbf{P}^{GP}(\boldsymbol{\theta})|\mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}^{(j)}), \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}^{(j)})\big) \tag{C23}$$

D. INFERENCE USING GP EMULATORS

Suppose now that we are given a vector of observations $\mathbf{d} \in \mathbb{R}^q$ and a distribution of the measurement noise $\mathcal{N}_q(\mathbf{0}_q, \boldsymbol{\Sigma}_E)$ with a known covariance $\boldsymbol{\Sigma}_E$. Upon substituting the true response $\mathbf{P}(\cdot)$ with the GP emulator $\mathbf{P}^{GP}(\cdot)$ in the likelihood of the measurement data, and integrating with respect to the GP distribution (C23), we obtain (see Takhtaganov & Müller (2018) for details) the so-called $\mathcal{D}$-restricted likelihood

$$L(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D}) = \sum_{j=1}^{n_\psi} \frac{s^{(j)}}{n_\psi} \exp\bigg[- \frac{g\big(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}^{(j)}\big)}{2} \bigg], \tag{D24}$$

where $s^{(j)} = (2\pi)^{-q/2}|\boldsymbol{\Sigma}_E + \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}^{(j)})|^{-1/2}$, and $g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ is a data misfit function defined as

$$g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) = (\mathbf{d} - \mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}))^T(\boldsymbol{\Sigma}_E + \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}))^{-1}(\mathbf{d} - \mathbf{m}(\boldsymbol{\theta}; \mathcal{D}, \psi)). \tag{D25}$$

When performing inference the likelihood $L(\boldsymbol{\theta}|\mathbf{d}, \mathcal{D})$ needs to be repeatedly evaluated for different values of $\boldsymbol{\theta}$. Instead of using the Cholesky factorization of the matrix appearing in the definition of $g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$, we compute the misfit efficiently as follows.

First, we cover the case of homoscedastic measurement noise, i.e., when $\boldsymbol{\Sigma}_E = \sigma_E^2 \mathbf{I}_q$. Denote the matrix appearing in $g(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})$ as

$$\boldsymbol{\Sigma}_{lik}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) = \boldsymbol{\Sigma}_E + \boldsymbol{\Sigma}_{GP}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}). \tag{D26}$$

Plugging-in $\boldsymbol{\Sigma}_E$ and $\boldsymbol{\Sigma}_{GP}$ we get

$$\boldsymbol{\Sigma}_{lik}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) = \mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})\bigg(\boldsymbol{\Sigma}_k + \frac{\sigma_E^2}{\mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})}\mathbf{I}_q\bigg). \tag{D27}$$

We have the sum of a symmetric matrix and a constant times the identity matrix. In this case, the inverse of $\boldsymbol{\Sigma}_{lik}$ can be efficiently computed using the eigendecomposition of the $\boldsymbol{\Sigma}_k$ matrix. Let

$$\boldsymbol{\Sigma}_k = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T, \quad \text{with } \mathbf{Q}^{-1} = \mathbf{Q}^T,\ \boldsymbol{\Lambda} = \mathrm{diag}[\lambda_1, \ldots, \lambda_q]. \tag{D28}$$

Then

$$\boldsymbol{\Sigma}_{lik}^{-1}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi}) = \frac{1}{\mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})}\mathbf{Q}\bigg[\boldsymbol{\Lambda} + \frac{\sigma_E^2}{\mathbb{V}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\psi})}\mathbf{I}_q\bigg]^{-1}\mathbf{Q}^T. \tag{D29}$$

In order to compute the misfit function $g(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})$ we first compute

$$\mathbf{v} = \mathbf{Q}^T(\mathbf{d} - \mathbf{m}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})) \in \mathbb{R}^q, \tag{D30}$$

then

$$\begin{aligned} g(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi}) &= \frac{1}{\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})}\mathbf{v}^T\mathbf{D}^{-1}\mathbf{v} \\ &= \sum_{i=1}^{q}\frac{1}{\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})\lambda_i + \sigma_E^2}v_i^2, \end{aligned} \tag{D31}$$

where $\mathbf{D} = \boldsymbol{\Lambda} + (\sigma_E^2/\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})\mathbf{I}_q$ is a diagonal matrix. Thus, for each $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ the computation of the data misfit function requires $\mathcal{O}(q^2)$ operations.

For a general $\boldsymbol{\Sigma}_E$, we use the generalized eigendecomposition

$$\boldsymbol{\Sigma}_E\mathbf{U} = \boldsymbol{\Sigma}_k\mathbf{U}\boldsymbol{\Lambda}, \tag{D32}$$

which leads to the following form for the inverse of $\boldsymbol{\Sigma}_{lik}$:

$$\boldsymbol{\Sigma}_{lik}^{-1}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi}) = \frac{1}{\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})}\mathbf{U}\left[\mathbf{I}_q + \frac{1}{\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi})}\boldsymbol{\Lambda}\right]^{-1}\mathbf{U}^T. \tag{D33}$$

Then

$$g(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi}) = \sum_{i=1}^{q}\frac{1}{\mathbb{V}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi}) + \lambda_i}v_i^2, \tag{D34}$$

with $\mathbf{v} = \mathbf{U}^T(\mathbf{d} - \mathbf{m}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\psi}))$.

REFERENCES

Almgren, A. S., Bell, J. B., Lijewski, M. J., Lukić, Z., & Van Andel, E. 2013, ApJ, 765, 39

Alvarez, M. A., Rosasco, L., Lawrence, N. D., et al. 2012, Foundations and Trends® in Machine Learning, 4, 195

Armengaud, E., Palanque-Delabrouille, N., Yèche, C., Marsh, D. J. E., & Baur, J. 2017, MNRAS, 471, 4606

Becker, G. D., Bolton, J. S., Haehnelt, M. G., & Sargent, W. L. W. 2011, MNRAS, 410, 1096

Bilionis, I., & Zabaras, N. 2014, Inverse Problems, 30, 015004, 32. https://doi.org/10.1088/0266-5611/30/1/015004

—. 2016, Handbook of Uncertainty Quantification, 1

Bilionis, I., Zabaras, N., Konomi, B. A., & Lin, G. 2013, J. Comput. Phys., 241, 212. https://doi.org/10.1016/j.jcp.2013.01.011

Bird, S., Rogers, K. K., Peiris, H. V., et al. 2019, JCAP, 2019, 050

Boera, E., Becker, G. D., Bolton, J. S., & Nasir, F. 2019, ApJ, 872, 101

Bonilla, E. V., Chai, K. M., & Williams, C. 2008, in Advances in neural information processing systems, 153–160

Carlson, J., White, M., & Padmanabhan, N. 2009, Phys. Rev. D, 80, 043531. https://link.aps.org/doi/10.1103/PhysRevD.80.043531

Chen, H., Loeppky, J. L., Sacks, J., & Welch, W. J. 2016, Statist. Sci., 31, 40. https://doi.org/10.1214/15-STS531

Chen, S.-F., Vlah, Z., & White, M. 2020, JCAP, 2020, 062

Conti, S., & O'Hagan, A. 2010, J. Statist. Plann. Inference, 140, 640. https://doi.org/10.1016/j.jspi.2009.08.006

d'Amico, G., Gleyzes, J., Kokron, N., et al. 2020, JCAP, 2020, 005

Dashti, M., & Stuart, A. M. 2016, Handbook of Uncertainty Quantification, 1

DESI Collaboration, Aghamousa, A., Aguilar, J., et al. 2016, arXiv e-prints, arXiv:1611.00036

Desjacques, V., Haehnelt, M. G., & Nusser, A. 2006, MNRAS, 367, L74

Euclid Collaboration, Knabenhans, M., Stadel, J., et al. 2019, MNRAS, 484, 5509

Frazier, P. 2018, arXiv:1807.02811

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. 2013, Bayesian Data Analysis, 3rd edn. (Chapman and Hall/CRC)

Haaland, B., & Qian, P. Z. G. 2011, Ann. Statist., 39, 2974. https://doi.org/10.1214/11-AOS929

Haardt, F., & Madau, P. 2012, ApJ, 746, 125

Habib, S., Heitmann, K., Higdon, D., Nakhleh, C., & Williams, B. 2007, Physical Review D, 76, 083503

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

Heitmann, K., White, M., Wagner, C., Habib, S., & Higdon, D. 2010, ApJ, 715, 104

Iršič, V., Viel, M., Haehnelt, M. G., Bolton, J. S., & Becker, G. D. 2017, PhRvL, 119, 031302

Ivanov, M. M., Simonović, M., & Zaldarriaga, M. 2020, JCAP, 2020, 042

Jennings, W. D., Watkinson, C. A., Abdalla, F. B., & McEwen, J. D. 2019, MNRAS, 483, 2907

Johnson, M. E., Moore, L. M., & Ylvisaker, D. 1990, Journal of statistical planning and inference, 26, 131

Jones, D. R., Schonlau, M., & Welch, W. J. 1998, J. Global Optim., 13, 455, workshop on Global Optimization (Trier, 1997). https://doi.org/10.1023/A:1008306431147

Kennedy, M. C., & O'Hagan, A. 2001, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63, 425

Kleijnen, J. P. 2015, in International Workshop on Simulation, Springer, 3–22

Kollmeier, J. A., Miralda-Escudé, J., Cen, R., & Ostriker, J. P. 2006, ApJ, 638, 52

Kulkarni, G., Hennawi, J. F., Oñorbe, J., Rorai, A., & Springel, V. 2015, ApJ, 812, 30

Kwan, J., Heitmann, K., Habib, S., et al. 2015, ApJ, 810, 35

Lawrence, E., Heitmann, K., Kwan, J., et al. 2017, ApJ, 847, 50

Leclercq, F. 2018, PhRvD, 98, 063511

Liu, J., Petri, A., Haiman, Z., et al. 2015, PhRvD, 91, 063507

Lochhaas, C., Weinberg, D. H., Peirani, S., et al. 2016, MNRAS, 461, 4353

Lohrmann, E., Lukić, Z., Morozov, D., & Müller, J. 2017, in Workshop on Job Scheduling Strategies for Parallel Processing, Springer, 122–131

LSST Dark Energy Science Collaboration. 2012, arXiv e-prints, arXiv:1211.0310

Lukić, Z., Stark, C. W., Nugent, P., et al. 2015, MNRAS, 446, 3697

McClintock, T., Rozo, E., Becker, M. R., et al. 2018, arXiv e-prints, arXiv:1804.05866

McQuinn, M. 2016, Annual Review of Astronomy and Astrophysics, 54, 313

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, Journal of Chemical Physics, 21, 1087

Mockus, J. 1994, Journal of Global Optimization, 4, 347

Mockus, J., Tiesis, V., & Zilinskas, A. 2014, The application of Bayesian methods for seeking the extremum, Vol. 2 (North-Holand), 117–129

Moon, H., Dean, A., & Santner, T. 2011, Journal of Statistical Theory and Practice, 5, 81

Morozov, D., & Lukić, Z. 2016, in Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing, ACM, 285–288

Oñorbe, J., Davies, F. B., Lukić, et al. 2019, MNRAS, 486, 4075

Palanque-Delabrouille, N., Yèche, C., Schöneberg, N., et al. 2020, JCAP, 2020, 038

Palanque-Delabrouille, N., Yèche, C., Borde, A., et al. 2013, A&A, 559, A85

Palanque-Delabrouille, N., Yèche, C., Baur, J., et al. 2015, Journal of Cosmology and Astro-Particle Physics, 2015, 011

Petri, A., Liu, J., Haiman, Z., et al. 2015, PhRvD, 91, 103511

Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2014, A&A, 571, A16

Planck Collaboration, Aghanim, N., Akrami, Y., et al. 2018, arXiv e-prints, arXiv:1807.06209

Qian, P. Z. 2012, Journal of the American Statistical Association, 107, 393

Refregier, A., Amara, A., Kitching, T. D., et al. 2010, arXiv e-prints, arXiv:1001.0061

Rogers, K. K., Bird, S., Peiris, H. V., et al. 2018, MNRAS, 474, 3032

Rogers, K. K., & Peiris, H. V. 2020, arXiv e-prints, arXiv:2007.12705

Rogers, K. K., Peiris, H. V., Pontzen, A., et al. 2019, Journal of Cosmology and Astroparticle Physics, 2019, 031

Rossi, G., Yèche, C., Palanque-Delabrouille, N., & Lesgourgues, J. 2015, PhRvD, 92, 063505

Seljak, U., Slosar, A., & McDonald, P. 2006, Journal of Cosmology and Astro-Particle Physics, 2006, 014

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. 2016, Proceedings of the IEEE, 104, 148

Sorini, D., Oñorbe, J., Lukić, Z., & Hennawi, J. F. 2016, ApJ, 827, 97

Spergel, D., Gehrels, N., Baltay, C., et al. 2015, arXiv e-prints, arXiv:1503.03757

Stuart, A. M., & Teckentrup, A. L. 2018, Math. Comp., 87, 721. https://doi.org/10.1090/mcom/3244

Takhtaganov, T., & Müller, J. 2018, arXiv:1809.10784. https://arxiv.org/pdf/1809.10784

Viel, M., Becker, G. D., Bolton, J. S., & Haehnelt, M. G. 2013, Physical Review D, 88, 043502

Walther, M., Oñorbe, J., Hennawi, J. F., & Lukić, Z. 2019, ApJ, 872, 13

Wibking, B. D., Weinberg, D. H., Salcedo, A. N., et al. 2020, MNRAS, 492, 2872

Yèche, C., Palanque-Delabrouille, N., Baur, J., & du Mas des Bourboux, H. 2017, JCAP, 2017, 047

Zhai, Z., Tinker, J. L., Becker, M. R., et al. 2018, arXiv e-prints, arXiv:1804.05867