Bayesian emulator optimisation for cosmology: application to the Lyman-alpha forest

Journal of Cosmology and Astroparticle Physics
An IOP and SISSA journal

Bayesian emulator optimisation for cosmology: application to the Lyman-alpha forest

Keir K. Rogers,[a,1] **Hiranya V. Peiris,**[b,a] **Andrew Pontzen,**[b] **Simeon Bird,**[c] **Licia Verde**[d,e] **and Andreu Font-Ribera**[b]

[a]Oskar Klein Centre for Cosmoparticle Physics, Stockholm University, AlbaNova, Stockholm SE-106 91, Sweden
[b]Department of Physics & Astronomy, University College London, Gower Street, London WC1E 6BT, U.K.
[c]Department of Physics & Astronomy, University of California, Riverside, 900 University Avenue, Riverside, CA 92521, U.S.A.
[d]Institut de Ciències del Cosmos, University of Barcelona, ICCUB, Barcelona 08028, Spain
[e]Institució Catalana de Recerca i Estudis Avançats, Passeig Lluís Companys 23, Barcelona 08010, Spain

E-mail: keir.rogers@fysik.su.se, h.peiris@ucl.ac.uk, a.pontzen@ucl.ac.uk, sbird@ucr.edu, liciaverde@icc.ub.edu, a.font@ucl.ac.uk

Received December 31, 2018
Accepted February 4, 2019
Published February 14, 2019

[1]Corresponding author.

https://doi.org/10.1088/1475-7516/2019/02/031

JCAP02(2019)031

Abstract. The Lyman-alpha forest provides strong constraints on both cosmological parameters and intergalactic medium astrophysics, which are forecast to improve further with the next generation of surveys including eBOSS and DESI. As is generic in cosmological inference, extracting this information requires a likelihood to be computed throughout a high-dimensional parameter space. Evaluating the likelihood requires a robust and accurate mapping between the parameters and observables, in this case the 1D flux power spectrum. Cosmological simulations enable such a mapping, but due to computational time constraints can only be evaluated at a handful of sample points; "emulators" are designed to interpolate between these. The problem then reduces to placing the sample points such that an accurate mapping is obtained while minimising the number of expensive simulations required. To address this, we introduce an emulation procedure that employs Bayesian optimisation of the training set for a Gaussian process interpolation scheme. Starting with a Latin hypercube sampling (other schemes with good space-filling properties can be used), we iteratively augment the training set with extra simulations at new parameter positions which balance the need to reduce interpolation error while focussing on regions of high likelihood. We show that smaller emulator error from the Bayesian optimisation propagates to smaller widths on the posterior distribution. Even with fewer simulations than a Latin hypercube, Bayesian optimisation shrinks the 95% credible volume by 90% and, e.g., the 1σ error on the amplitude of small-scale primordial fluctuations by 38%. This is the first demonstration of Bayesian optimisation applied to large-scale structure emulation, and we anticipate the technique will generalise to many other probes such as galaxy clustering, weak lensing and 21cm.

Keywords: cosmological parameters from LSS, cosmological simulations, Lyman alpha forest

ArXiv ePrint: 1812.04631

JCAP02(2019)031

Contents

JCAP02(2019)031

1 Introduction

The cosmic large-scale structure informs us about the late-time evolution of the Universe as well as bearing imprints of the primordial fluctuations; the most accurate modelling of this epoch requires numerical simulation. For example, the Lyman-alpha forest is simultaneously sensitive to a wide range of cosmological and astrophysical parameters. It is sourced by the (mildly) non-linear gas in the intergalactic medium (IGM), meaning that forward modelling requires the computationally-expensive calculation of a cosmological hydrodynamical simulation. This requires one to estimate the likelihood function with *millions* of samples (to adequately sample the parameter space by Markov chain Monte Carlo methods) while only being able to compute *tens* of cosmological simulations.

This challenge is worth overcoming because of the unique range of scales (hundreds of Mpc to sub-Mpc) and redshifts ($2 < z < 5$) at which the Lyman-alpha forest is observed. This allows the forest to put tight limits on the presence of additional cosmological components like massive neutrinos or non-cold dark matter, and deviations from a power-law primordial power spectrum (on small scales) [1–6]; and (on large scales) measure the cosmological expansion rate [7–12] and geometry [13–16] at a redshift before dark energy came to dominate the energy contents of the Universe. The Lyman-alpha forest is also sensitive to the thermal history (temperature and density) of the IGM from the end of hydrogen reionisation through the

reionisation of helium, e.g., refs. [17, 18]. These constraints are forecast [19] to tighten further with ongoing and future spectroscopic surveys, e.g., the extended Baryon Oscillation Sky Survey (eBOSS) [20] and the Dark Energy Spectroscopic Instrument (DESI) [21, 22]. It may also be possible to learn about the sources of ionising radiation on large scales [23]. However, in order to learn about either cosmology or astrophysics, it is necessary to marginalise over the uncertainty in the other. It follows that in order to estimate the likelihood function of a given dataset, it is necessary to simultaneously vary multiple parameters (e.g., in this study, we consider a model with seven parameters in total). This puts even further strain on the small number (approximately 50–100; e.g., ref. [18] run a suite of ~ 70) of cosmological simulations that can be reasonably run in the available computing time.

The solution lies in *emulating* the outputs of simulations. This is a form of interpolation, meaning that a small number of forward simulations can be used to predict simulation outputs throughout parameter space. Emulators have found use in various branches of science, wherever forward modelling is computationally expensive due e.g., either to complex non-linearities (e.g., fluid dynamics problems in engineering [24–27]) or an extremely large number of elements that require calculation (e.g., the simulation of microbial communities in biology [28]).

Indeed, emulators have found use in modelling the cosmological large-scale structure, e.g., modelling the small-scale non-linear matter power spectrum [29]; the galaxy power spectrum [30, 31]; galaxy weak lensing peak counts and power spectrum [32, 33]; the 21cm power spectrum [34]; or the halo mass function [35]. Ref. [18] emulate the one-dimensional (1D) Lyman-alpha forest flux power spectrum for three thermal parameters and the mean flux in their inference on the IGM thermal history (for a fixed cosmology). Also, ref. [36] interpolate the small-scale ($0.005\,\mathrm{s\,km^{-1}} \leq k_{||} \leq 0.08\,\mathrm{s\,km^{-1}}$), high-redshift ($4.2 \leq z \leq 5$) 1D flux power spectrum using the "ordinary kriging" method (e.g., ref. [37]) for the study of non-cold dark matter models.

The standard approach for the interpolation of Lyman-alpha forest observables (as used in refs. [38–41]) is to make use of quadratic polynomial interpolation using a second-order Taylor expansion around a fiducial simulation. However, most importantly for statistical inference, this method gives no theoretical error estimate in the interpolation. Instead, errors are estimated empirically from test simulations and a global worst-case error is assigned. In this work and a companion article [42], we use Gaussian processes to model the 1D flux power spectrum for a full set of cosmological and astrophysical parameters. A Gaussian process is a stochastic process where any finite subset forms a multivariate Gaussian distribution (see ref. [43] for a review). The Gaussian process provides a principled theoretical estimate of the uncertainty in interpolated simulation values, eliminating any need to resort to worst-case empirical estimates.

The key practical element is the construction of the training dataset for optimising the Gaussian process. In this study, we investigate Bayesian optimisation [44–48] as a method to decide where in parameter space to run training simulations. In cosmology, ref. [49] used Bayesian optimisation in likelihood-free inference from Joint Light-curve Analysis supernovae data [50] in order to gain more precise posterior distributions with fewer forward simulations than existing methods. The approach iteratively uses knowledge about the approximate likelihood function and the regions of parameter space where there is greatest uncertainty. The principle of Bayesian optimisation is to propose new training samples balancing *exploration* of the prior volume where the current uncertainty in the optimised function is highest with *exploitation* of previous iterations of the emulator revealing the most interesting (for

JCAP02(2019)031

us, high-likelihood) regions. Bayesian optimisation has been developed as a technique to find the optima of functions in as few evaluations as possible; it achieves this by using prior information in the manner set out above. This naturally dovetails into Gaussian process emulation, which provides a robust estimate of uncertainty in predictions across the parameter space [51]. The details of the balance between exploration and exploitation are encoded in the acquisition function used to determine future proposals, of which many examples have been developed (we use a novel expansion of the GP-UCB acquisition function) [52–56]. We also show how to propose multiple training samples simultaneously (batch acquisition).

The key point here is that cosmologists are mostly interested in accurately characterising the peak of the posterior distribution (i.e., the $\sim$ 95%–99% credible region). Indeed, the standard approach to sampling the posterior distribution — Markov chain Monte Carlo (MCMC) methods — is optimised for this purpose. Therefore, we do not actually require (and often cannot afford) uniform accuracy in the modelling of cosmological observables across the prior volume. Bayesian optimisation solves this problem by concentrating available resources in regions of high posterior probability using informed Bayesian decision making.

Bayesian optimisation can help in our inference problem, where we specify the likelihood function but where model evaluations are extremely computationally expensive and the parameter space is high-dimensional — as well as many other inference problems in cosmology with computationally expensive forward simulations and many parameters. This is compared to the "brute-force" method of simply increasing the number of simulations in the uniform Latin hypercube sampling scheme described in our companion paper [42]. This could "waste" samples on the edges of the prior volume. The informed decision-making of the Bayesian optimisation can ultimately lead to the robust statistical inference necessary to exploit the full potential of ongoing and future spectroscopic surveys such as eBOSS and DESI — as well as improve the performance of many other cosmological emulators.

The article is organised as follows. In section 2.1, we review the use of Gaussian processes to emulate the 1D Lyman-alpha forest flux power spectrum. We detail our use of Bayesian optimisation in section 2.2. We present our main results in section 3, discuss them in section 4 and draw our final conclusions in section 5.

2 Method

2.1 Gaussian process emulator

The full details of our Gaussian process emulator are given in our companion paper [42]; here we summarise the key points.

2.1.1 Simulated training data

The data vector which we emulate (see section 2.1.2) and from which we calculate our likelihood function (see section 2.1.3) is the 1D Lyman-alpha forest flux power spectrum $P^{\rm 1D}(\boldsymbol{\theta}; k_{||}, z)$ — line-of-sight fluctuations of transmitted flux in quasar spectra. It is a function of line-of-sight wavenumber $k_{||}$, redshift z and cosmological and astrophysical model parameters $\boldsymbol{\theta}$. In order to evaluate accurate theoretical predictions for $P^{\rm 1D}$, it is necessary to run a cosmological hydrodynamical simulation for each $\boldsymbol{\theta}$. For this, we use the publicly-available code `MP-Gadget`[1] [57], itself derived from the public `GADGET-2` code [58, 59], in

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

JCAP02(2019)031

Model parameter	Prior range
A_s	$[1.5 \times 10^{-9}, 2.8 \times 10^{-9}]$
n_s	$[0.9, 0.99]$
h	$[0.65, 0.75]$
H_A	$[0.6, 1.4]$
H_S	$[-0.4, 0.4]$
τ_0	$[0.75, 1.25]$
$\mathrm{d}\tau_0$	$[-0.25, 0.25]$

Table 1. The ranges of the uniform prior on our model parameters.

order to evolve 256^3 particles each of dark matter and gas in a $(40\,h^{-1}\,\mathrm{Mpc})^3$ box[2] from a starting redshift $z = 99$ to the redshifts at which the Lyman-alpha forest is observed in BOSS Data Release 9 (DR9) [40, 60, 61], $2.2 \leq z \leq 4.2$. From the simulated particle data, we then generate mock quasar spectra containing only the Lyman-alpha absorption line and calculate the 1D flux power spectrum using `fake_spectra` [62]. We use the same 35 $k_{||}$ and 11 z bins as used in BOSS DR9.

The model parameters $\boldsymbol{\theta}$ we elect to use are:

- three cosmological: the amplitude A_s and scalar spectral index n_s of the primordial power spectrum with a pivot scale $k_\mathrm{pivot} = \frac{2\pi}{8}\,\mathrm{Mpc}^{-1}$ corresponding to the scales probed by the 1D flux power spectrum, and the dimensionless Hubble parameter[3] h;
- two astrophysical: the heating amplitude H_A and slope H_S of the (over)density Δ - dependent rescaling of photo-heating rates[4] $\epsilon = H_\mathrm{A}\epsilon_0\Delta^{H_\mathrm{S}}$;
- two for the mean flux, a multiplicative correction to the amplitude τ_0 and an additive correction to the slope $\mathrm{d}\tau_0$ of the empirical redshift dependence of the mean optical depth as measured by ref. [64].

We use a uniform prior (in the evaluation of our posterior distribution of model parameters; see section 2.1.3) on $\boldsymbol{\theta}$ as given in table 1 (the limits differ slightly from our companion paper [42]). More details about our parameterisation and possible alternatives are discussed in our companion paper [42].

Before considering Bayesian optimisation (see section 2.2), the basic set-up of our emulator is to generate training data (the sample points from which we emulate) on a Latin hypercube sampling of the full prior volume. We calculate $P^\mathrm{1D}(\boldsymbol{\theta}; k_{||}, z)$ as described above at a certain number N_Latin (our base emulator uses 21 training samples) of values of $\boldsymbol{\theta}$, as sampled

[2]Our hydrodynamical simulations lack the size or resolution for a robust comparison to real data, but they are sufficient for the explication of this method. We do not expect any complication in applying this method with fully-converged simulations; the only difference is that simulated flux power spectra will be more precisely determined. In particular, the Gaussian process model and Bayesian optimisation method that we prove here will be equally applicable to these converged simulations.

[3]Note that we otherwise fix the (physical) total matter density $\Omega_\mathrm{m}h^2 = 0.1327$ and baryon density $\Omega_\mathrm{b} = 0.0483$. Indeed, the vast majority of our apparent constraining power on h in section 3 arises from the inverse scaling of Ω_m and its effect on the linear growth factor.

[4]This effectively allows us to modify the temperature-density relationship of the IGM gas [63].

JCAP02(2019)031

by the Latin hypercube. A Latin hypercube is a random sampling scheme with good space-filling properties (low discrepancy). It sub-divides the prior volume along each axis into $N_{\rm Latin}$ equal sub-spaces. It then randomly distributes $N_{\rm Latin}$ samples under the constraint that each sub-space is sampled once. We randomly generate many different Latin hypercubes and choose the one that maximises the minimum Euclidean distance between different samples.

The number of simulations necessary in the base Latin hypercube emulator for accurate estimation of the likelihood function is a non-trivial choice dependent on the number of model parameters, the size of the prior volume and the particular correlation structure of the parameter space — i.e., quite particular to the problem at hand. In our testing, however, we found that if the density of training samples in the full prior volume $N_{\rm Latin}/V_{\rm prior}$ is insufficient, the resulting large error in the interpolation (same order of magnitude as the "measurement" error) causes spurious bias and/or multi-modality in the approximate likelihood function (see sections 2.1.2 and 2.1.3). Our base emulator of 21 simulations gives a good first approximation of the likelihood for this particular context (with the emulator-to-measurement error ratio $\sim 20\%$); see section 4 and our companion paper [42] for further discussion.

Because the mean flux can be adjusted in the post-processing of hydrodynamical simulations (with comparatively negligible computational cost), its parameters are handled differently. Although we sample in the likelihood function the two parameters $[\tau_0, {\rm d}\tau_0]$ described above, we actually emulate the mean optical depth separately in each redshift bin. This means that the $N_{\rm Latin}$ training samples from above are only distributed in the five cosmological and astrophysical parameters. Then, at each of these training points, the mean optical depth in each redshift bin is sampled uniformly from a prior range translated from the prior ranges on $[\tau_0, {\rm d}\tau_0]$. We use 10 mean flux samples per redshift bin; this number can in effect be arbitrarily increased though tests show no improvement in the accuracy of Gaussian process modelling from doing so.[5]

Finally, we anticipate sections 2.1.3 and 3 and note that we test our method on simulated rather than real data because we wish to compare our results to a known truth. We therefore generate a mock data vector $\boldsymbol{d} \equiv P^{\rm 1D}(\boldsymbol{\theta}_{\mathbf{true}}; k_{||}, z)$ by the same process as our theory evaluations described above, where $[\tau_0, {\rm d}\tau_0, A_{\rm s}, n_{\rm s}, h, H_{\rm A}, H_{\rm S}] = [0.95, 0, 2.24 \times 10^{-9}, 0.974, 0.685, 1.09, 0.0509]$.

2.1.2 Gaussian process emulation as interpolation

In section 2.1.3, we will want to estimate our likelihood function by evaluating $P^{\rm 1D}(\boldsymbol{\theta}; k_{||}, z)$ for very many ($\sim 10^6$) different values of $\boldsymbol{\theta}$, but we have only $N_{\rm Latin} = 21$ model evaluations. We want to find a flexible model to *interpolate* $P^{\rm 1D}(\boldsymbol{\theta}; k_{||}, z)$ between the evaluations we have and to have a robust estimate of the uncertainty in this interpolation so that we can include it in the statistical model. We achieve this by modelling the simulation output as a Gaussian process (a stochastic process where any finite sub-set forms a multivariate Gaussian distribution):

$$P^{\rm 1D}(\boldsymbol{\theta}) \sim \mathcal{N}(0, K(\boldsymbol{\theta}, \boldsymbol{\theta'}; \boldsymbol{\psi})). \tag{2.1}$$

Here, we have dropped the dependence on $k_{||}$ and z; each z bin is emulated separately and correlations between $k_{||}$ bins are not modelled. The zero mean condition is approximated by normalising the flux power spectra by the median value in the training set. It is then necessary to specify a form for the covariance $K(\boldsymbol{\theta}, \boldsymbol{\theta'}; \boldsymbol{\psi})$ between two points in parameter

[5] Our treatment of the mean flux is somewhat akin to the concept of "fast" and "slow" parameters in MCMC sampling [65].

JCAP02(2019)031

space $\boldsymbol{\theta}$ and $\boldsymbol{\theta'}$; note however that this is a much more general specification than in traditional interpolation methods where the functional form for $P^{\rm 1D}(\boldsymbol{\theta})$ must be given. We use a linear combination of a squared exponential (or radial basis function; RBF) and a linear kernel: $K(\boldsymbol{\theta}, \boldsymbol{\theta'}; \boldsymbol{\psi}) = \sigma^2_{\rm RBF} \exp\left(-\frac{(\boldsymbol{\theta}-\boldsymbol{\theta'})^2}{2l^2}\right) + \sigma^2_{\rm linear}\boldsymbol{\theta}.\boldsymbol{\theta'}$. This gives three hyper-parameters $\boldsymbol{\psi}$: two variances for the squared exponential ($\sigma^2_{\rm RBF}$) and the linear ($\sigma^2_{\rm linear}$) kernels and a length-scale l for the squared exponential. We optimise these hyper-parameters (or "train" the emulator) by maximising the (Gaussian) marginal likelihood of the training data.

Once the emulator has been trained, we can use the Gaussian process to interpolate $P^{\rm 1D}(\boldsymbol{\theta})$ at arbitrary values of $\boldsymbol{\theta}$. For this, we calculate the posterior predictive distribution of simulation output $P^{\rm 1D}(\boldsymbol{\theta^*})$ conditional on the training data $P^{\rm 1D}(\boldsymbol{\theta}_i)$, for i in $\{1, \ldots, N\}$, where N is the number of training samples:

$$p(P^{\rm 1D}(\boldsymbol{\theta^*})|P^{\rm 1D}(\boldsymbol{\theta}_i), \boldsymbol{\theta}_i, \boldsymbol{\theta^*}) \sim \mathcal{N}(K_* K^{-1} P^{\rm 1D}(\boldsymbol{\theta}_i), K_{**} - K_* K^{-1} K_*^{\rm T}). \tag{2.2}$$

Here, $K_* = K(\boldsymbol{\theta^*}, \boldsymbol{\theta}_i; \boldsymbol{\psi})$ and $K_{**} = K(\boldsymbol{\theta^*}, \boldsymbol{\theta^*}; \boldsymbol{\psi})$. Since we have determined the full distribution of interpolated flux power spectra, we have a robust estimate of the error in our interpolation as given by the variance term in eq. (2.2). Note that this variance is independent of the training output, being only dependent on the locations in parameter space and number of training samples; this will be of use in section 2.2.4.

2.1.3 Likelihood function and Markov chain Monte Carlo sampling

We now have all the pieces to construct the likelihood function and perform inference. We use a simple Gaussian likelihood function.[6] The mean is inferred by our trained Gaussian process $\boldsymbol{\mu}(\boldsymbol{\theta^*}) = K_* K^{-1} P^{\rm 1D}(\boldsymbol{\theta}_i)$ [eq. (2.2)]. The covariance matrix adds in quadrature the "data" covariance matrix as estimated by BOSS DR9 for their real data and the diagonal "emulator" covariance matrix, using the variance as inferred by the trained Gaussian process $\boldsymbol{\sigma}^2(\boldsymbol{\theta^*}) = K_{**} - K_* K^{-1} K_*^{\rm T}$ [eq. (2.2)]. In this way, the statistical model accounts for the uncertainty in the theoretical predictions (emulation) and we are able to test our method using realistic BOSS errors. Note that the Gaussian process is not currently modelling correlations between different scale bins. In our companion paper [42], these are conservatively estimated as being maximally correlated (though uncorrelated across redshifts). Both approaches amount to approximations in the likelihood function; in future work, explicit modelling of these correlations by the Gaussian process can be investigated. These likelihood approximations do not change our main results on demonstrating the effect of Bayesian emulator optimisation for a given likelihood function. Having constructed the likelihood function, we then estimate the posterior probability distribution for our mock data $\boldsymbol{d}$ by MCMC sampling using the `emcee` package [66].

2.2 Bayesian optimisation

For Gaussian process emulators (as with machine learning problems generically), the construction of the training dataset is critical. In section 2.1.1, we detailed the use of Latin hypercube sampling to construct the training dataset. Latin hypercubes have good space-filling properties ensuring that the prior volume is well sampled. However, this is not necessarily the most efficient use of the limited resources available. In optimisation problems, such as the estimation of a posterior probability distribution where we are only interested in the peak

[6] In principle, a non-Gaussian likelihood function could be used if necessary as long as a robust statistical model can be constructed which propagates uncertainty from the emulator.

JCAP02(2019)031

and surrounding credible region of the distribution, evenly sampling the full prior volume may waste training samples on the edges of that volume. The idea of Bayesian optimisation is to build up the training dataset actively and iteratively using, at each stage, the information we have gained from previous iterations of the emulator. This is expressed in the acquisition function (section 2.2.1). Bayesian optimisation proceeds by iteratively proposing new training data points at the maximum of the acquisition function (plus a small random displacement; section 2.2.3), which is continually updated as the emulator is re-trained on the expanded training dataset. This procedure can be modified to allow "batches" of training data to be acquired simultaneously (section 2.2.4), which may be preferred depending on the availability of computational resources.

2.2.1 Acquisition function

At each iteration, the next training simulation is run at the parameters of the maximum of the acquisition function (plus a small random displacement; see below). This function should peak where uncertainty in the emulated function is high so that it is better characterised. It should also peak where the objective function is high so that training samples accumulate where it matters. This manifests in a function which increases with the Gaussian process uncertainty or variance (exploration) and with the objective function itself (exploitation).

We use the Gaussian process upper confidence bound (GP-UCB) acquisition function [52–55]. This is a weighted linear combination of exploitation and exploration terms, usually $\boldsymbol{\mu}(\boldsymbol{\theta}) + \alpha\boldsymbol{\sigma}(\boldsymbol{\theta})$, where $\boldsymbol{\mu}(\boldsymbol{\theta})$ and $\boldsymbol{\sigma}(\boldsymbol{\theta})$ are respectively the mean and standard deviation of the posterior predictive distribution as given by the Gaussian process (see eq. (2.2)). Here, $\boldsymbol{\theta}$ is the parameter vector and α is a hyper-parameter, which can be optimised to give minimum regret[7] in the decisions made (ref. [67] calculate optimal values for various Gaussian process covariance kernels when the objective is the emulated function). A subtlety arises because in this study, we are not emulating the function which must be optimised (the posterior), but rather the flux power spectrum, which, along with its uncertainty, goes into the likelihood function (see section 2.1.3). We therefore use a modified form of the acquisition function:

$$\mathcal{A}(\widetilde{\boldsymbol{\theta}}) = \mathcal{P}(\widetilde{\boldsymbol{\theta}}|\boldsymbol{d}) + \alpha\boldsymbol{\sigma}^{\mathrm{T}}(\widetilde{\boldsymbol{\theta}})\Sigma^{-1}\boldsymbol{\sigma}(\widetilde{\boldsymbol{\theta}}), \tag{2.3}$$

where $\mathcal{P}(\widetilde{\boldsymbol{\theta}}|\boldsymbol{d})$ is the logarithm of the posterior probability distribution given data $\boldsymbol{d}$ and Σ is the data covariance matrix (see section 2.1.3). This function exploits the current best estimate of the posterior (by the first term) but explores the parameter space where this estimate is most uncertain (by the second term). The second term is constructed by estimating the uncertainty on the (log) posterior as $\frac{1}{2}|\mathcal{P}(\boldsymbol{\mu}(\widetilde{\boldsymbol{\theta}}) + \boldsymbol{\sigma}(\widetilde{\boldsymbol{\theta}})) - \mathcal{P}(\boldsymbol{\mu}(\widetilde{\boldsymbol{\theta}}) - \boldsymbol{\sigma}(\widetilde{\boldsymbol{\theta}}))|$.

A further subtlety is that although the mean flux parameters $[\tau_0, \mathrm{d}\tau_0]$ are emulated, an arbitrary number of training samples can be constructed for these parameters since their effect on the flux power spectrum is estimated in the post-processing of hydrodynamical simulations (section 2.1.1). Therefore, no Bayesian optimisation is necessary for the training set in these dimensions. It follows that the posterior in eq. (2.3) is marginalised over the mean flux parameters and that $\widetilde{\boldsymbol{\theta}}$ only lists the remaining cosmological and astrophysical parameters (see section 2.1.1). For the hyperparameter α, we use the optimised form as given by refs. [67, 68]. Here, we simplify their notation to $\alpha = 0.97\sqrt{\nu}$ and ν linearly decreases

[7] I.e., $\lim_{T\to\infty} \frac{R_T}{T} = 0$, where the cumulative regret $R_T = \sum_{t=1}^{T}[f_{\max} - f(\boldsymbol{\theta}_t)]$, where $f_{\max}$ is the true maximum of the objective function f and $\boldsymbol{\theta}_t$ are the positions of acquisition function proposals for the training set. Minimising regret puts a lower limit on the convergence rate of finding the true optimum.

JCAP02(2019)031

from a starting value of 1 to 0.4 by convergence as the size of the training set increases in order to reflect the increasing confidence in the emulator. Since $\nu \sim 1$, this approximates to about a 1σ uncertainty in the log posterior.

Gaussian processes are unsuited to extrapolation and so the emulator error increases sharply at the edges of the prior volume. This can spuriously dominate the acquisition function (we manifestly do not want to add samples on the perimeter of the prior volume). Therefore, we apply a uniform prior when finding the maximum of the acquisition function which excludes the outer 5 % of parameter space in each dimension (approximating the convex hull formed by the initial training samples).

Finally, following e.g., refs. [49, 69], once the maximum of the acquisition function has been found, the final proposal for a new training sample is a small random displacement away from the maximum. This helps to ensure that the same position in parameter space cannot be proposed more than once, which can in principle be the case especially when the emulator is (near-)converged. It also helps to explore the credible region around the peak of the posterior distribution (to the extent desired by the user, usually the 95% credible region), bearing in mind that the optimisation procedure should not only find the peak of the distribution but should correctly characterise the region around it. Therefore, this random displacement can be drawn from a Gaussian distribution with a full-width-at-half-maximum set by the current estimation of the posterior contours (most simply approximated by using the desired number (e.g., two) of sigma from the 1D marginalised distributions; see ref. [56] for discussion and comparison of deterministic and stochastic acquisition rules). In general, these "exploration" terms should be tuned to the size of the credible region which is required to be well estimated.

2.2.2 Initial Latin hypercube

We initiate the Bayesian optimisation with a Latin hypercube (see section 2.1.1; other sampling schemes with good space-filling properties, e.g., a Sobol sequence [70], can be used) on the full prior volume. The size of this initial hypercube should not be so small as to characterise the emulated function poorly. As mentioned in section 2.1.1, when the density of training samples (for the Latin hypercube) in the prior volume $N_{\rm Latin}/V_{\rm prior}$ is too low, the emulated function is characterised so poorly that the likelihood function, in propagating emulator error, is biased. In testing with an initial hypercube of only nine simulations, the convergence rate of the Bayesian optimisation from this initialisation became such that it was more efficient to run an initial hypercube with more simulations. Equally, the size of the initial hypercube should not be too large. This would waste samples on the edges of the prior volume and negate the power of Bayesian optimisation to propose training samples efficiently. Ultimately, the best size for the initial hypercube is a trade-off and we show the results of our experiments (our initial hypercube has 21 simulations) in section 3, with discussion in section 4.

2.2.3 Serial optimisation

Serial Bayesian optimisation proceeds by proposing and running training simulations one-by-one. After each training simulation has been added to the training dataset, the emulator is re-trained and the acquisition function re-evaluated. The procedure can be summarised by these steps:

1. Construct the initial training dataset by running hydrodynamical simulations (see section 2.1.1 for details) at parameters sampled by a Latin hypercube on the full prior volume (section 2.2.2). (Also see our companion paper [42] for more discussion about using Latin hypercubes.)

JCAP02(2019)031

2. Train the Gaussian process emulator for the flux power spectrum (section 2.1.2) and then evaluate the posterior probability distribution for the given data (section 2.1.3).

3. Evaluate the acquisition function (eq. (2.3)) and find its maximum in order to propose the location of the next training simulation (plus a small random displacement; section 2.2.1).

4. Run this "refinement" simulation, re-train the emulator using the optimised and expanded training set and then re-evaluate the posterior distribution.

5. Repeat the previous *two* steps (3 and 4) until the desired number of optimisation steps have been executed.

6. Optimisation can continue until successive estimations of the posterior (practically, summary statistics like the mean and 1σ limits can be used) are seen to sufficiently converge (to some specified tolerance like a fraction, e.g., 0.2, of a sigma).

7. Performance can be checked by cross-validation and/or a test suite of fiducial simulations (bearing in mind that the emulator is optimised only for the true parameters of the data but that tests with fiducial simulations will still inform us about performance in relevant areas of parameter space).

Figure 1 illustrates two example iterations of the serial optimisation method. It shows the procedure for a toy (cubic) function. It demonstrates how Bayesian optimisation efficiently proposes training samples in order to better characterise the emulated function and therefore the true likelihood function. In the actual case, the emulated function is the 1D flux power spectrum and there are six model parameters (three cosmological, two astrophysical and the mean flux at each redshift; see section 2.1.1).

2.2.4 Batch optimisation

Batch Bayesian optimisation proceeds as in the serial case (section 2.2.3) except that at each optimisation step, multiple training samples are proposed and evaluated simultaneously in a single batch. This may be preferred depending on the particular allocation of computational resources for running hydrodynamical simulations. Within each batch, decisions are in fact still made in series but without running the simulations from earlier decisions or re-training the emulator until the set of batch proposals is completed. The positions in parameter space of new proposals from earlier on in the batch can be added to the training set in order to help inform later decisions within the same batch. This will correctly update the expected error distribution of the emulator. We can do this because the variance of a Gaussian process is independent of the training output (it is only a function of the Gaussian process covariance kernel; see section 2.1.2 and eq. (2.2)). Thus, in turn, before each proposal in a given batch, the second "exploration" term in the acquisition function (eq. (2.3)) can be updated and a new maximum found. The disadvantage is that, even so, proposals later on in a given batch are less informed about the true objective function (for us, the posterior distribution) than they would be in the equivalent serial case.

Once all the proposals in a given batch have been made and the simulations have finished running, the emulator can be re-trained and the acquisition function fully re-evaluated. A new batch can be started as necessary. For maximum efficiency of the Bayesian optimisation, the procedure should be as serial as possible so that each proposal is as informed as it can be;

JCAP02(2019)031

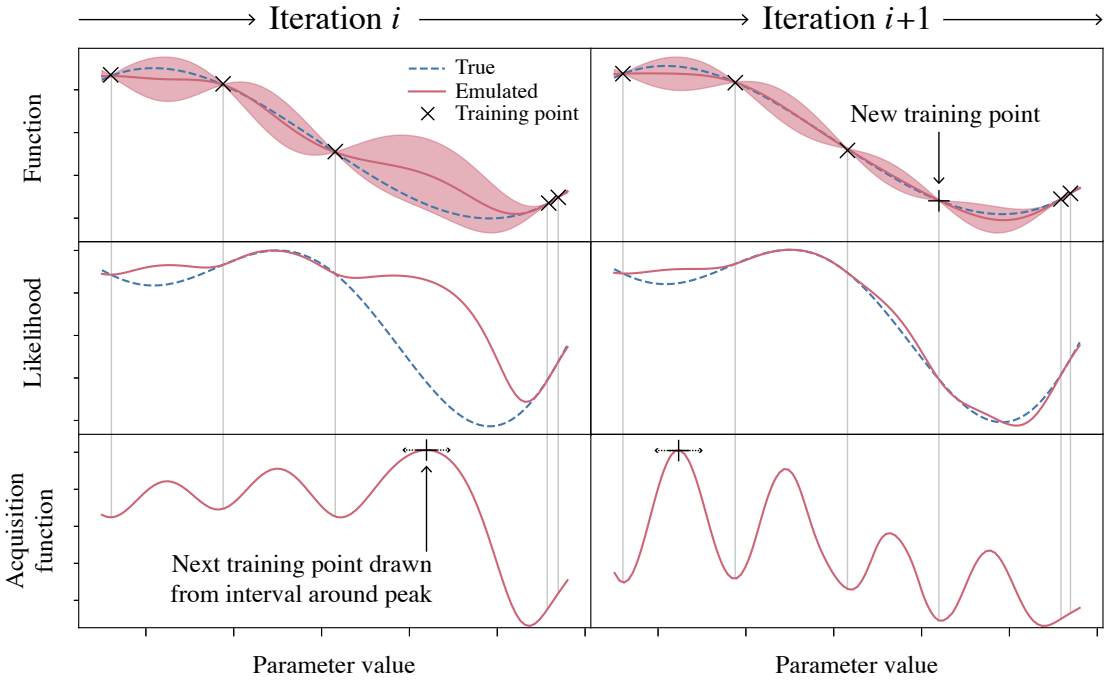

Figure 1. An illustration of (*from left to right*) two example successive iterations of the serial Bayesian optimisation (section 2.2.3) for a toy function. *From top to bottom*, we show the emulated function (the truth is a cubic function); the (Gaussian) likelihood function; and the acquisition function. The blue dotted lines show the truth and the red solid lines show the emulated estimation of the toy function (the mean as inferred by the Gaussian process model; see section 2.1.2) in the top panels and the likelihood propagating the emulator error (as described in section 2.1.3) in the middle panels. The red band shows the $\pm 1\sigma$ error on the emulated function as inferred by the Gaussian process. The training data (with which the Gaussian process is optimised) are indicated by the black crosses. At the end of each iteration, the next training sample is proposed at the maximum of the acquisition function (plus a small random displacement; see section 2.2.1). Bayesian optimisation efficiently proposes new training samples in order to better characterise the emulated function and therefore the true likelihood function.

i.e., the batch size should be as small as the allocation of computational resources reasonably allows. In our testing in section 3, we use a batch size of three. The procedure can be summarised by these steps:

1. As with serial optimisation (section 2.2.3), the procedure starts with an initial Latin hypercube training set, which is used to train a first iteration of the emulator (see section 2.2.2).

2. The first proposal of each optimisation batch is made using the maximum of the full acquisition function as given by eq. (2.3) (see section 2.2.1).

3. Subsequent proposals in each batch use the acquisition function, partially updated with the new Gaussian process standard deviation $\boldsymbol{\sigma}(\boldsymbol{\theta})$ including the effect of proposals made previously in the same batch. This is achieved by adding to the training set the positions in parameter space of earlier proposals from the same batch without re-

JCAP02(2019)031

training the emulator. This exploits the fact that the variance of a Gaussian process is independent of training output.

4. Within each batch, repeat the above step (3) until the desired number of samples per batch has been chosen.

5. Simultaneously run all the "refinement" simulations of the batch, re-train the emulator using the optimised and expanded training set (including its simulation output) and then re-evaluate the posterior distribution.

6. Repeat the previous *four* steps (2 to 5) until the desired number of optimisation batches have been executed.

7. The same convergence and cross-validation tests can be carried out as in the serial case (see section 2.2.3).

3 Results

3.1 Serial optimisation

Figure 2 shows the results of serial Bayesian optimisation (section 2.2.3) on the (1D and 2D marginalised) posterior probability distribution (in the filled coloured contours) of our model parameters $\boldsymbol{\theta}$ given our mock data $\boldsymbol{d}$ (the true parameters $\boldsymbol{\theta}_{\mathbf{true}}$ of which are indicated by the gray dotted lines).[8] The coloured crosses indicate the projected positions in parameter space of our training simulations. Note that the mean flux parameters $[\mathrm{d}\tau_0, \tau_0]$ do not form part of the Latin hypercube and are treated differently (see section 2.1.1 for more details). Most importantly, these parameters are adjusted in the post-processing of hydrodynamical simulations and hence it is not necessary to employ Bayesian optimisation in these axes. Otherwise, our initial Latin hypercube (section 2.2.2; in gray; consisting of 21 simulations) fills the full prior volume with random samples, uniformly in projection on each parameter axis. By employing the procedure detailed in section 2.2.3, five optimisation samples were chosen (until the inferred posterior distributions were seen to converge as confirmed by subsequent optimisation steps by the tests and details set out in section 2.2.3). The first three of these are coloured in red and the final two in blue. Thanks to the Bayesian optimisation exploiting our knowledge of the approximate posterior distribution as inferred using previous iterations of the emulator, these optimisation samples are concentrated in the most important region of parameter space (the 95% credible region of the posterior distribution). This region is explored by including the emulator error $\boldsymbol{\sigma}(\boldsymbol{\theta})$ in the Bayesian optimisation acquisition function (eq. (2.3)) and the stochastic element in the final acquisition (see section 2.2.1). Note that although the projection makes some of the hypercube samples seem to appear in the peak of the posterior, it is the optimisation samples that are actually located in the central posterior volume.

It can be seen in figure 2 that reduced emulator error after Bayesian optimisation propagates to reducing the widths of the marginalised posterior distributions, e.g., the 1σ error on A_s reduces by 38%. The effect of the Bayesian optimisation is to reduce the emulator error in the central posterior volume. For example, the emulator error at $\boldsymbol{\theta}_{\mathbf{true}}$ is reduced by

[8] Note that, in general, our inferred posterior distributions differ from our companion paper [42] in their width due to different approximations in the likelihood function (see section 2.1.3). However, this does not change our main conclusions on the effect of Bayesian emulator optimisation.

JCAP02(2019)031

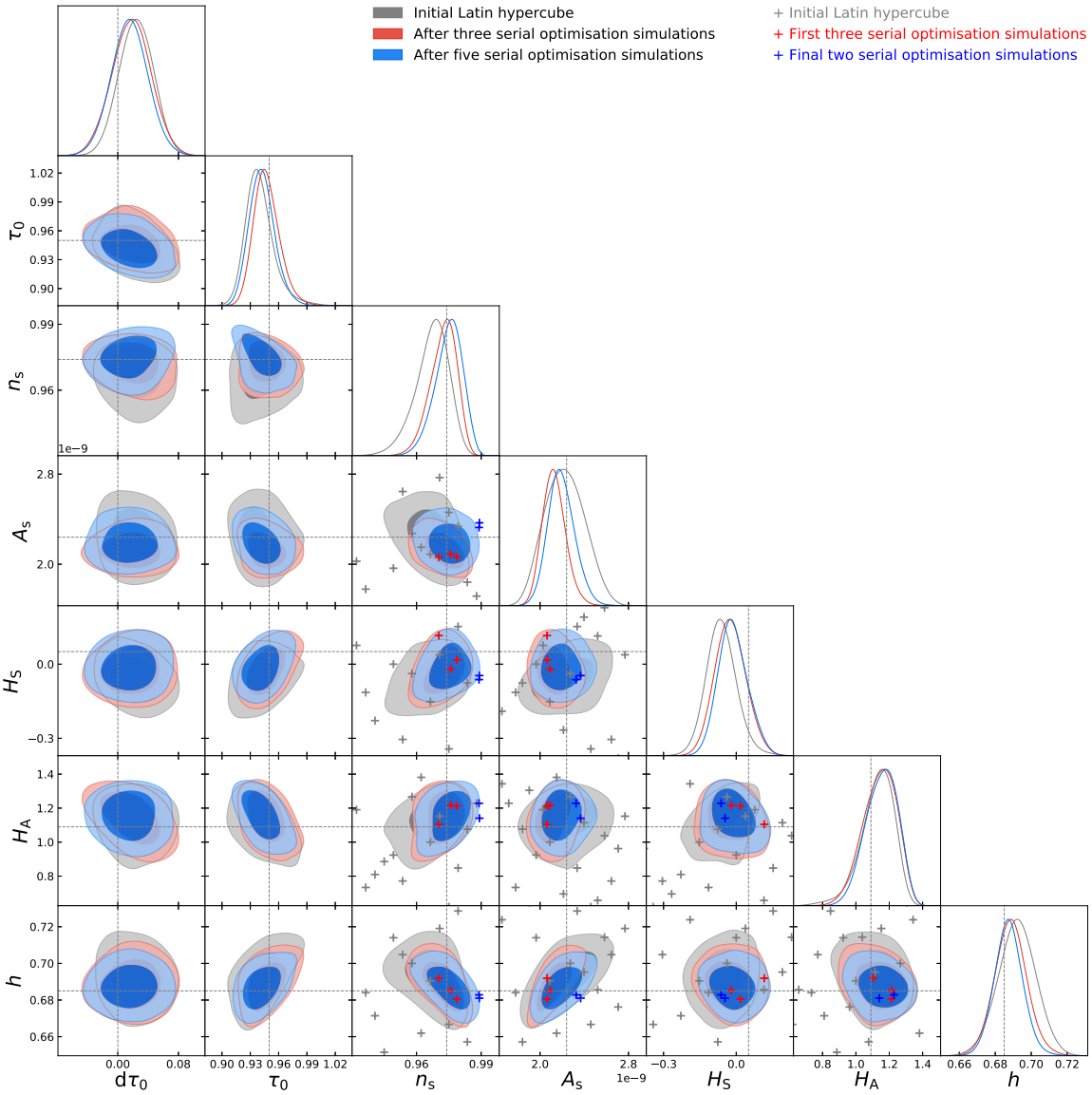

Figure 2. The 1D and 2D marginalised posterior probability distributions (see section 2.1.3) for the seven model parameters $\boldsymbol{\theta}$ (section 2.1.1) given the mock data vector $\boldsymbol{d}$. The gray, red and blue contours use an emulator trained respectively on the initial Latin hypercube of 21 samples; the initial hypercube plus three serial Bayesian optimisation simulations; and the initial hypercube plus five serial optimisation simulations. The darker and lighter shaded contours show respectively the 68% and 95% credible regions. The gray dotted lines indicate the true model parameter values $\boldsymbol{\theta}_{\mathbf{true}}$ of our mock data vector. The gray, red and blue crosses indicate the projected positions in parameter space of our training samples respectively for the initial Latin hypercube; the first three serial optimisation simulations; and the final two serial optimisation simulations. Note that we do not show the training samples for the mean flux parameters $[\mathrm{d}\tau_0, \tau_0]$, which do not form part of our Latin hypercube or Bayesian optimisation and are treated differently (see section 2.1.1 for details).

61% (averaged over all the $k_{||}$ and z bins of the data vector) after the five serial optimisation steps; meaning that the ratio of emulator error to "data" error (the diagonal elements of the

JCAP02(2019)031

data covariance matrix; see section 2.1.3), again averaged over all bins, goes from 16% for the initial hypercube to 6% after optimisation. By reducing this ratio, the more accurate emulator provides a more accurate estimation of the posterior distribution (see eq. (2.3) and section 2.2.1). Figure 1 shows schematically how this effect works. In general, when the error in the emulated function is high, the likelihood at those parameter values, in propagating the error, is inflated. This is because the increased uncertainty means that there is more probability than otherwise that the data are drawn from that region of parameter space. The general effect is to broaden the peak of the likelihood function. However, the addition of new training points better characterises the interpolated function and therefore the true likelihood. (Spurious biases, rather than simple broadening of the peak, can also arise in the limit of the emulator-to-data error ratio being too large, as is discussed in section 4.2.)

3.2 Batch optimisation

Figure 3 shows the results of batch Bayesian optimisation (section 2.2.4) on the posterior probability distribution of our model parameters $\boldsymbol{\theta}$ given our mock data $\boldsymbol{d}$; again, the coloured crosses indicate the projected positions in parameter space of our training simulations. The difference from above is that now optimisation samples are proposed in batches of three. The results and explanations are very similar to the serial case (see above): the optimisation samples are concentrated in the central posterior volume exploiting our knowledge of the posterior; the emulator-to-data error ratio is reduced to the same extent; and the widths of the marginalised posterior distributions shrink. The main difference is that the batch optimisation takes longer to converge (after three batches each of three simulations). This makes sense since the second and third proposals in each batch are less informed than the equivalent serial proposal because the emulator is only re-trained after each batch of three has finished running in parallel. We will discuss the benefits and disadvantages of batch acquisition in section 4.

3.3 Comparison to a Latin hypercube

Figure 4 compares the results of Bayesian optimisation with a Latin hypercube of 30 simulations (our initial hypercube from above from which we optimise has 21 simulations). For a fair comparison, we construct this larger Latin hypercube (still spanning the full prior volume) using the initial Latin hypercube as a sub-set of its samples; the extra nine training points are indicated by the red crosses in figure 4. It can be seen in figure 4 that the smaller emulator error from Bayesian optimisation with respect to the Latin hypercube propagates to reducing the size of the 68% and 95% credible regions of the posterior distribution. Indeed, the full volume of these regions reduces by 90% and, e.g., the 1σ error on A_s reduces by 38%. Although the Latin hypercube has a larger training set than our serial Bayesian optimisation example (which has 26 simulations in total), because the samples of the larger hypercube are spread throughout the prior space, the Latin hypercube actually has larger emulator error in the central posterior volume. The large Latin hypercube has $\sim 20\%$ greater emulator error at $\boldsymbol{\theta}_\mathbf{true}$. A more accurate emulation of the flux power spectrum means more accurate estimation of the posterior probability distribution (see above, section 4 and figure 1 for explanation). This in particular means that the weakening of parameter constraints from uncertainty in forward modelling (emulation) can be reduced.

JCAP02(2019)031

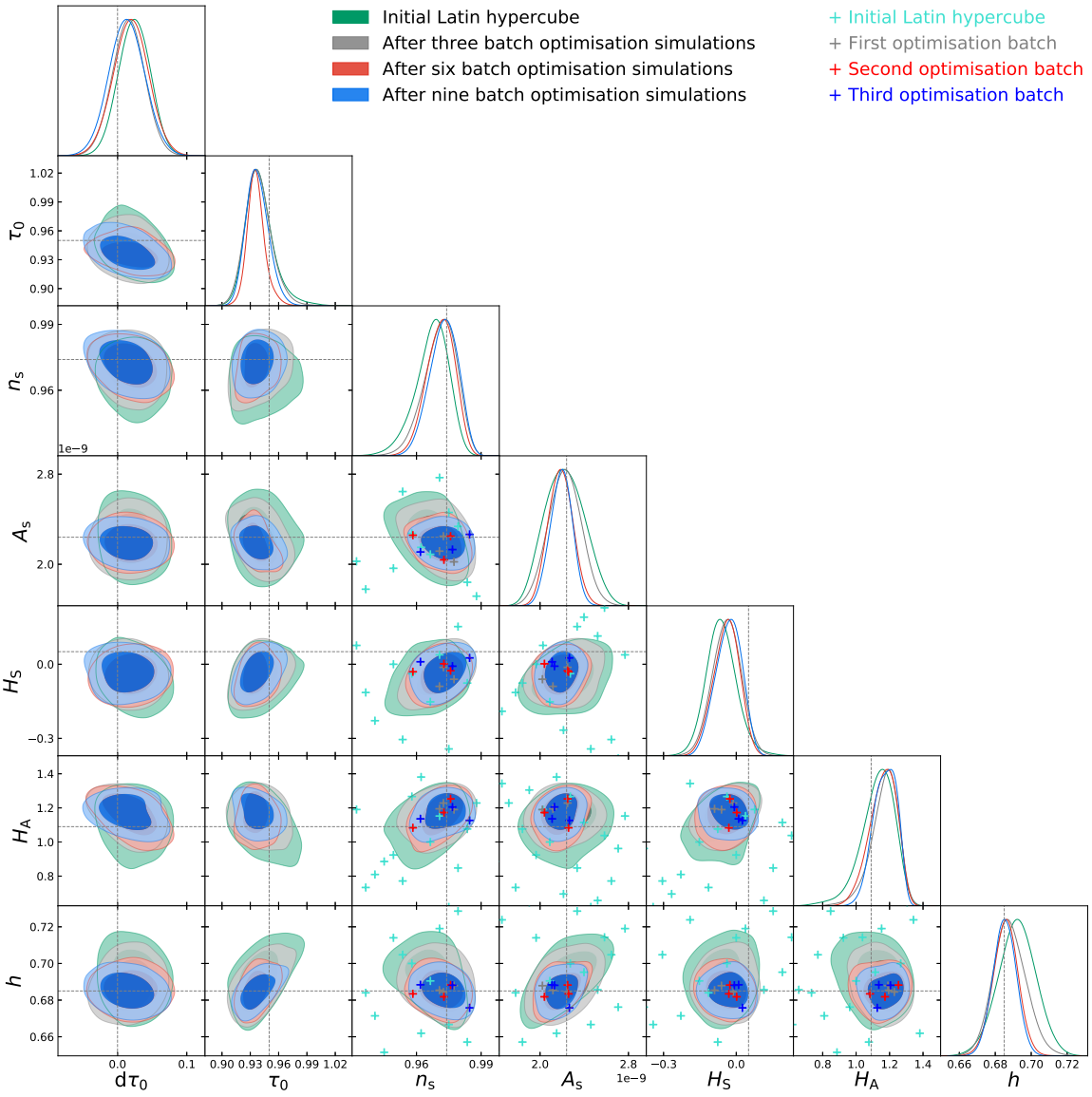

Figure 3. As figure 2 except showing the results of batch Bayesian optimisation.

4 Discussion

Figures 2 and 3 demonstrate the ability of Bayesian optimisation to determine more accurately the posterior distribution in the central, high-probability parameter space. It achieves this by concentrating training samples for the Gaussian process emulator of the 1D Lyman-alpha forest flux power spectrum (section 2.1) in the parameter space corresponding to the central, high-probability region of the posterior distribution. As a consequence, the error on the emulated flux power spectrum is more than halved (e.g., a 61% reduction when averaged over all power spectrum bins at $\boldsymbol{\theta}_{\mathbf{true}}$, the true model parameters of our mock data vector). More accurate estimation of the flux power spectrum means more accurate estimation of the posterior probability distribution. This is because, in the likelihood function (see section 2.1.3), the theory prediction has smaller interpolation error and in the covariance matrix,

JCAP02(2019)031

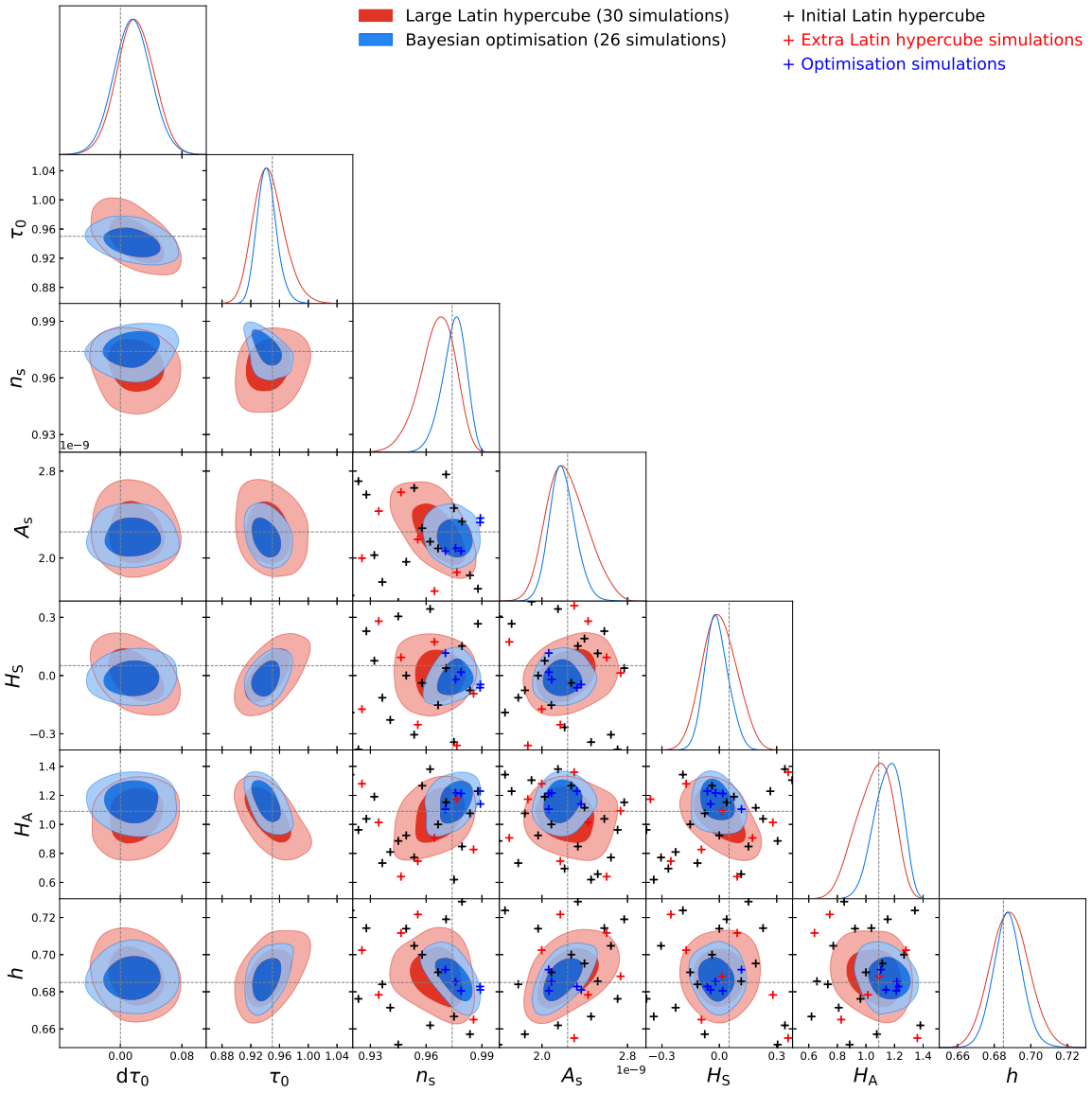

Figure 4. As figure 2 except comparing the results of Bayesian optimisation (with a total number of 26 training simulations) to a Latin hypercube (with 30 training simulations).

the ratio of emulator-to-data error is reduced. Figure 1 demonstrates how this more accurate determination of the "true" (zero emulator error) likelihood in general manifests in reducing the width of the peak. This is because areas of parameter space with high (interpolation) uncertainty in the flux power spectrum have increased possibility than otherwise that the data are drawn from there.

We also consider how well the Gaussian process estimates interpolation error. There is detailed discussion on this matter in our companion paper [42], where the estimated error distribution is compared to the true error distribution as evaluated at a suite of test simulations. The general tendency is for the Gaussian process to overestimate the emulator error moderately. This will tend to have a conservative effect on parameter estimation as it will tend to broaden the peak of the posterior distribution more than necessary.

JCAP02(2019)031

4.1 Comparison between serial and batch acquisition

The difference between the two cases in figures 2 and 3 is in respectively the serial (section 2.2.3) and batch (section 2.2.4) acquisition. The batch optimisation takes longer to converge in its posterior distribution (requiring four more simulations). This is because training sample proposals are less informed than in the equivalent serial case (after the first proposal in each batch). The scaling of our simulation code with the number of computing cores means that, in this case, the serial acquisition was marginally more efficient in terms of overall computing time. However, we have not yet carried out detailed tests of how the convergence rate scales with batch size. Therefore, in future uses of this method, it will probably still be preferable to use batch acquisition in order to make efficient use of the available computational resources. The size of the batch to use can be determined by balancing the specifics of the distribution of computational power, the optimal load balancing of the forward modelling code and the decrease in the convergence rate from batch acquisition. A potential approach is to start with serial optimisation until the estimation of posterior distributions for the validation set is unbiased (e.g., the truth is recovered to 68% credibility) and then to continue with batch optimisation once exploration of the 95% credible region is required.

4.2 The importance of the initial sampling density

In our examples, the initial Latin hypercube (section 2.2.2), from which we optimise, has 21 training simulations. We found this to be a sufficient number to give a reasonable first estimate of the posterior distribution. We found in testing with an initial hypercube of nine simulations (with emulator error on the same order of magnitude as the "measurement" error) that using fewer simulations in the initial hypercube (or, more specifically, having a lower density of initial training samples in the prior volume $N_{\rm Latin}/V_{\rm prior}$) can lead to significant bias and/or spurious multi-modality in the inferred posterior distribution using this initial training set. (In particular, using Bayesian optimisation from this initial set proved less efficient than simply starting from a hypercube with more simulations and a better characterisation of the objective function — the likelihood.) Figure 1 helps to qualitatively understand this phenomenon. Where the emulator error is high, the likelihood function is inflated taking account of the uncertainty in model prediction. If the error is sufficiently high, it can form local maxima (i.e., multi-modality) in the likelihood function, which are not otherwise present in the limit of zero emulator error.

However, the precise threshold on $N_{\rm Latin}/V_{\rm prior}$ to keep the bias at a level below 2σ (by reducing the ratio of emulator to data error) has not been determined. It is additionally dependent on the particular correlation structure of the model parameter space (as estimated by the optimal Gaussian process hyper-parameter values; see section 2.1.2), i.e., the extent to which any particular training sample can predict model values elsewhere in the prior volume. In particular, each correlation length of the parameter volume should be sampled at least once. It follows that this is a non-trivial, problem-dependent decision. It is possible to construct Latin hypercubes with more simulations using an existing Latin hypercube as a subset of the samples, as we demonstrate in section 3. This problem of determining the size of the initial Latin hypercube could be tackled by iteratively increasing its size until the spurious biases are removed. In future uses of this method with more precise datasets, in order to maintain the same level of final emulator-to-data error ratio ($\sim 10\%$), it will be necessary to reduce the emulator error further. This may require a higher density of training samples, especially in the central posterior volume using Bayesian optimisation, but also in the initial training set (in order to avoid the spurious biases discussed above). Future testing

JCAP02(2019)031

of the method with alternative model parameterisations and data vectors, with different optimal Gaussian process hyper-parameter values, (e.g., the 3D flux power spectrum; see our companion paper [42] for more discussion) will address these important issues further.

4.3 Comparison to a Latin hypercube

Figure 4 demonstrates the power of Bayesian optimisation compared to "brute-force" Latin hypercube sampling. Following the explanations from above, smaller emulator error reduces the size of the 68% and 95% credible regions of the posterior volume by 90%, with, e.g., a 38% reduction in the 1σ error on A_s, for the posterior using the Bayesian optimisation emulator compared to the Latin hypercube with four more simulations. Because the emulator error varies as a function of parameter value, there is no simple way to estimate how it affects the shape of the posterior distribution. The extreme limit of this with few training points and high emulator error leads to the spurious bias and multi-modality in the likelihood function discussed in section 4.2 (rather than just simply broadening the posterior). Ultimately, the most robust method is the Bayesian optimisation which will concentrate training samples at the peak of the posterior. This will smoothly lower the emulator error at the peak of the posterior in order to determine more accurately the 95% credible volume.

The details of optimal batch size and the size of the initial training set will be explored further in future work and is expected to be specific to the particular survey (data errors) and distribution of computational resources available. Our results (see also our companion paper [42]) on BOSS DR9 — the current state-of-the-art in terms of large-scale survey 1D flux power spectrum — mock data have shown that Bayesian optimisation can lead to smaller errors propagating through to the final model parameter estimation. This is achieved with fewer simulations than simply running a Latin hypercube with more simulations.

5 Conclusions

We have investigated Bayesian emulator optimisation — iterative addition of extra simulation samples to an existing emulator of the Lyman-alpha forest 1D flux power spectrum. We found that this method produces converged posterior parameter constraints with 15% fewer simulations than a single-step Latin hypercube. Bayesian optimisation of the training set reduces the error (compared to brute-force Latin hypercube sampling) in the required interpolation (emulation) of simulated flux power spectra. This propagates to more accurate inference of the posterior distribution. In our companion paper [42], we showed how Gaussian processes can be used to robustly interpolate between a training set of simulations so that the likelihood function can be evaluated. Gaussian processes give a principled estimate of the error in this interpolation which can be propagated to the final inference.

However, the construction of the training set is essential, especially considering how few simulations are available. We found that building the training set using Bayesian optimisation concentrates training samples in areas of high posterior probability (i.e., the 95% credible region), where we most care about correctly inferring the posterior distribution. This is akin to the standard MCMC approaches to sampling posterior distributions, which focus on characterising the peak of the distribution accurately. Bayesian optimisation achieves this by iteratively proposing training samples, exploiting knowledge of the approximate posterior from previous iterations of the emulator and exploring the prior volume where our current characterisation of the posterior is most uncertain. We initiated the procedure with a Latin hypercube training set; we found that if this initial set has too low a density of samples in the

JCAP02(2019)031

prior volume, the emulator error is high enough to bias the inferred posterior distribution. We explored two types of acquisition for the Bayesian optimisation: one where proposals are made serially and the decisions are maximally informed; and one where proposals are made in batches and simulations are run in parallel. The latter can make more efficient use of computational resources at the expense of some increase in the number of optimisation steps until convergence.

We found that both Bayesian optimisation acquisition methods can give smaller emulator error in interpolated flux power spectra in areas of high posterior probability than by using a Latin hypercube to construct the training dataset. Despite having four more simulations in the training set, the emulator using a large Latin hypercube had $\sim 20\%$ larger error at the true parameters of our mock data than by using Bayesian optimisation to concentrate training samples around the peak of the posterior. This reduced emulator error propagates to reduced widths on the inferred posterior distribution, with the volume of the 68% and 95% credible regions shrunk by an order of magnitude and, e.g., a 38% reduction in the 1σ error on the amplitude of the small-scale primordial fluctuations ($k_{\rm pivot} = \frac{2\pi}{8}\,{\rm Mpc}^{-1}$).

This is the first demonstration of Bayesian optimisation applied to large-scale structure emulation. Having tested the efficacy of Bayesian emulator optimisation on mock data with BOSS DR9 data covariance, this study outlines a new methodology for the cosmological and astrophysical parameter estimation from ongoing and future spectroscopic surveys like eBOSS and DESI. Furthermore, we anticipate that these methods will be of benefit to the many other emulators of the large-scale structure (e.g., galaxy clustering, weak lensing and 21cm).

Acknowledgments

The authors thank Jens Jasche, Florent Leclercq, Chris Pedersen and Risa Wechsler for valuable discussions. This work was performed in part at the Aspen Center for Physics, which is supported by National Science Foundation (NSF) grant PHY-1607611. KKR was supported by the Science Research Council (VR) of Sweden. HVP was partially supported by the European Research Council (ERC) under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement number 306478-CosmicDawn, and the research project grant "Fundamental Physics from Cosmological Surveys" funded by the Swedish Research Council (VR) under Dnr 2017-04212. AP was supported by the Royal Society. HVP and AP were also partially supported by a grant from the Simons Foundation. SB was supported by NSF grant AST-1817256. LV was supported by the European Union's Horizon 2020 research and innovation programme ERC (BePreSySe, grant agreement 725327) and Spanish MINECO projects AYA2014-58747-P AEI/FEDER, UE, and MDM-2014-0369 of ICCUB (Unidad de Excelencia Maria de Maeztu). AFR was supported by a Science and Technology Facilities Council (STFC) Ernest Rutherford Fellowship, grant reference ST/N003853/1. HVP, AP and AFR were further supported by STFC Consolidated Grant number ST/R000476/1. This work was partially enabled by funding from the University College London (UCL) Cosmoparticle Initiative.

References

[1] SDSS collaboration, *Cosmological parameter analysis including SDSS Ly-α forest and galaxy bias: constraints on the primordial spectrum of fluctuations, neutrino mass and dark energy*, *Phys. Rev.* **D 71** (2005) 103515 [astro-ph/0407372] [INSPIRE].

JCAP02(2019)031

[2] N. Palanque-Delabrouille et al., *Neutrino masses and cosmology with Lyman-α forest power spectrum*, *JCAP* **11** (2015) 011 [arXiv:1506.05976] [INSPIRE].

[3] V. Iršič et al., *New constraints on the free-streaming of warm dark matter from intermediate and small scale Lyman-α forest data*, *Phys. Rev.* **D 96** (2017) 023522 [arXiv:1702.01764] [INSPIRE].

[4] C. Yèche, N. Palanque-Delabrouille, J. Baur and H. du Mas des Bourboux, *Constraints on neutrino masses from Lyman-α forest power spectrum with BOSS and XQ-100*, *JCAP* **06** (2017) 047 [arXiv:1702.03314] [INSPIRE].

[5] V. Iršič, M. Viel, M.G. Haehnelt, J.S. Bolton and G.D. Becker, *First constraints on fuzzy dark matter from Lyman-α forest data and hydrodynamical simulations*, *Phys. Rev. Lett.* **119** (2017) 031302 [arXiv:1703.04683] [INSPIRE].

[6] E. Armengaud, N. Palanque-Delabrouille, C. Yèche, D.J.E. Marsh and J. Baur, *Constraining the mass of light bosonic dark matter using SDSS Lyman-α forest*, *Mon. Not. Roy. Astron. Soc.* **471** (2017) 4606 [arXiv:1703.09126] [INSPIRE].

[7] A. Slosar et al., *The Lyman-α forest in three dimensions: measurements of large scale flux correlations from BOSS 1st-year data*, *JCAP* **09** (2011) 001 [arXiv:1104.5244] [INSPIRE].

[8] N.G. Busca et al., *Baryon Acoustic Oscillations in the Ly-α forest of BOSS quasars*, *Astron. Astrophys.* **552** (2013) A96 [arXiv:1211.2616] [INSPIRE].

[9] D. Kirkby et al., *Fitting methods for Baryon Acoustic Oscillations in the Lyman-α forest fluctuations in BOSS data release* 9, *JCAP* **03** (2013) 024 [arXiv:1301.3456] [INSPIRE].

[10] A. Slosar et al., *Measurement of Baryon Acoustic Oscillations in the Lyman-α forest fluctuations in BOSS data release* 9, *JCAP* **04** (2013) 026 [arXiv:1301.3459] [INSPIRE].

[11] BOSS collaboration, *Baryon Acoustic Oscillations in the Lyα forest of BOSS DR*11 *quasars*, *Astron. Astrophys.* **574** (2015) A59 [arXiv:1404.1801] [INSPIRE].

[12] J.E. Bautista et al., *Measurement of Baryon Acoustic Oscillation correlations at* $z = 2.3$ *with SDSS DR*12 *Lyα-forests*, *Astron. Astrophys.* **603** (2017) A12 [arXiv:1702.00176] [INSPIRE].

[13] C. Alcock and B. Paczynski, *An evolution free test for non-zero cosmological constant*, *Nature* **281** (1979) 358 [INSPIRE].

[14] L. Hui, A. Stebbins and S. Burles, *A geometrical test of the cosmological energy contents using the Lyman-α forest*, *Astrophys. J.* **511** (1999) L5 [astro-ph/9807190] [INSPIRE].

[15] P. McDonald and J. Miralda-Escude, *Measuring the cosmological geometry from the Lyman α forest along parallel lines of sight*, *Astrophys. J.* **518** (1999) 24 [astro-ph/9807137] [INSPIRE].

[16] P. McDonald, *Toward a measurement of the cosmological geometry at* $Z \sim 2$*: predicting Lyman-α forest correlation in three dimensions and the potential of future data sets*, *Astrophys. J.* **585** (2003) 34 [astro-ph/0108064] [INSPIRE].

[17] M.G. Haehnelt and M. Steinmetz, *Probing the thermal history of the intergalactic medium with Lyman-α absorption lines*, *Mon. Not. Roy. Astron. Soc.* **298** (1998) 21 [astro-ph/9706296] [INSPIRE].

[18] M. Walther, J. Oñorbe, J.F. Hennawi and Z. Lukić, *New constraints on IGM thermal evolution from the Lyα forest power spectrum*, arXiv:1808.04367 [INSPIRE].

[19] A. Font-Ribera, P. McDonald, N. Mostek, B.A. Reid, H.-J. Seo and A. Slosar, *DESI and other dark energy experiments in the era of neutrino mass measurements*, *JCAP* **05** (2014) 023 [arXiv:1308.4164] [INSPIRE].

[20] K.S. Dawson et al., *The SDSS-IV extended Baryon Oscillation Spectroscopic Survey: overview and early data*, *Astron. J.* **151** (2016) 44 [arXiv:1508.04473] [INSPIRE].

JCAP02(2019)031

[21] DESI collaboration, *The DESI experiment part I: science, targeting and survey design*, arXiv:1611.00036 [INSPIRE].

[22] DESI collaboration, *The DESI experiment part II: instrument design*, arXiv:1611.00037 [INSPIRE].

[23] A. Pontzen, S. Bird, H. Peiris and L. Verde, *Constraints on ionising photon production from the large-scale Lyman-α forest*, *Astrophys. J.* **792** (2014) L34 [arXiv:1407.6367] [INSPIRE].

[24] J. Sacks, W.J. Welch, T.J. Mitchell and H.P. Wynn, *Design and analysis of computer experiments*, *Statist. Sci.* **4** (1989) 409.

[25] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan and P.K. Tucker, *Surrogate-based analysis and optimization*, *Progr. Aerospace Sci.* **41** (2005) 1.

[26] A. Keane, A. Forrester and A. Sobester, *Engineering design via surrogate modelling: a practical guide*, American Institute of Aeronautics and Astronautics Inc., Wiley, U.S.A. (2008).

[27] A.I. Forrester and A.J. Keane, *Recent advances in surrogate-based optimization*, *Progr. Aerospace Sci.* **45** (2009) 50.

[28] O. Oyebamiji, D. Wilkinson, B. Li, P. Jayathilake, P. Zuliani and T. Curtis, *Bayesian emulation and calibration of an individual-based model of microbial communities*, *J. Comput. Sci.* **30** (2019) 194.

[29] K. Heitmann, D. Higdon, M. White, S. Habib, B.J. Williams and C. Wagner, *The Coyote universe II: cosmological models and precision emulation of the nonlinear matter power spectrum*, *Astrophys. J.* **705** (2009) 156 [arXiv:0902.0429] [INSPIRE].

[30] J. Kwan et al., *Cosmic emulation: fast predictions for the galaxy power spectrum*, *Astrophys. J.* **810** (2015) 35 [arXiv:1311.6444] [INSPIRE].

[31] Z. Zhai et al., *The Aemulus project III: emulation of the galaxy correlation function*, arXiv:1804.05867 [INSPIRE].

[32] J. Liu, A. Petri, Z. Haiman, L. Hui, J.M. Kratochvil and M. May, *Cosmology constraints from the weak lensing peak counts and the power spectrum in CFHTLenS data*, *Phys. Rev.* **D 91** (2015) 063507 [arXiv:1412.0757] [INSPIRE].

[33] A. Petri, J. Liu, Z. Haiman, M. May, L. Hui and J.M. Kratochvil, *Emulating the CFHTLenS weak lensing data: cosmological constraints from moments and Minkowski functionals*, *Phys. Rev.* **D 91** (2015) 103511 [arXiv:1503.06214] [INSPIRE].

[34] W.D. Jennings, C.A. Watkinson, F.B. Abdalla and J.D. McEwen, *Evaluating machine learning techniques for predicting power spectra from reionization simulations*, *Mon. Not. Roy. Astron. Soc.* **483** (2019) 2907 [arXiv:1811.09141] [INSPIRE].

[35] T. McClintock et al., *The Aemulus project II: emulating the halo mass function*, arXiv:1804.05866 [INSPIRE].

[36] R. Murgia, V. Iršič and M. Viel, *Novel constraints on noncold, nonthermal dark matter from Lyman-α forest data*, *Phys. Rev.* **D 98** (2018) 083540 [arXiv:1806.08371] [INSPIRE].

[37] R. Webster and M.A. Oliver, *Geostatistics for environmental scientists*, Wiley, U.S.A. (2007).

[38] M. Viel and M.G. Haehnelt, *Cosmological and astrophysical parameters from the SDSS flux power spectrum and hydrodynamical simulations of the Lyman-α forest*, *Mon. Not. Roy. Astron. Soc.* **365** (2006) 231 [astro-ph/0508177] [INSPIRE].

[39] S. Bird, H.V. Peiris, M. Viel and L. Verde, *Minimally parametric power spectrum reconstruction from the Lyman-α forest*, *Mon. Not. Roy. Astron. Soc.* **413** (2011) 1717 [arXiv:1010.1519] [INSPIRE].

[40] N. Palanque-Delabrouille et al., *The one-dimensional Ly-α forest power spectrum from BOSS*, *Astron. Astrophys.* **559** (2013) A85 [arXiv:1306.5896] [INSPIRE].

JCAP02(2019)031

[41] N. Palanque-Delabrouille et al., *Constraint on neutrino masses from SDSS-III/BOSS Lyα forest and other cosmological probes*, *JCAP* **02** (2015) 045 [arXiv:1410.7244] [INSPIRE].

[42] S. Bird, K.K. Rogers, H.V. Peiris, L. Verde, A. Font-Ribera and A. Pontzen, *An emulator for the Lyman-α forest*, arXiv:1812.04654 [INSPIRE].

[43] C.E. Rasmussen and C.K.I. Williams, *Gaussian processes for machine learning*, MIT Press, U.S.A. (2005).

[44] H.J. Kushner, *A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise*, *J. Basic Eng.* **86** (1964) 97.

[45] J. Mockus, V. Tiesis and A. Zilinskas, *Bayesian methods for seeking the extremum*, chapter in *Toward global optimization*, volume 2, North-Holland, The Netherlands (1978).

[46] J. Mockus, *Application of Bayesian approach to numerical methods of global and stochastic optimization*, *J. Global Optim.* **4** (1994) 347.

[47] M.C. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, *Biometrika* **87** (2000) 1.

[48] M.C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, *J. Roy. Statist. Soc.* **B 63** (2001) 425.

[49] F. Leclercq, *Bayesian optimization for likelihood-free cosmological inference*, *Phys. Rev.* **D 98** (2018) 063511 [arXiv:1805.07152] [INSPIRE].

[50] SDSS collaboration, *Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples*, *Astron. Astrophys.* **568** (2014) A22 [arXiv:1401.4064] [INSPIRE].

[51] M. Locatelli, *Bayesian algorithms for one-dimensional global optimization*, *J. Global Optim.* **10** (1997) 57.

[52] D.D. Cox and S. John, *SDO: a statistical method for global optimization*, in *Multidisciplinary design optimization: state-of-the-art*, (1997), pg. 315.

[53] P. Auer, *Using confidence bounds for exploitation-exploration trade-offs*, *J. Machine Learn. Res.* **3** (2002) 397.

[54] P. Auer, N. Cesa-Bianchi and P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, *Machine Learn.* **47** (2002) 235.

[55] V. Dani, T.P. Hayes and S.M. Kakade, *Stochastic linear optimization under bandit feedback*, (2008).

[56] M. Järvenpää, M.U. Gutmann, A. Pleska, A. Vehtari and P. Marttinen, *Efficient acquisition rules for model-based approximate Bayesian computation*, arXiv:1704.00520.

[57] Y. Feng, S. Bird, L. Anderson, A. Font-Ribera and C. Pedersen, *MP-gadget/MP-gadget: a tag for getting a DOI*, October 2018 [Zenodo].

[58] V. Springel, N. Yoshida and S.D.M. White, *GADGET: a code for collisionless and gasdynamical cosmological simulations*, *New Astron.* **6** (2001) 79 [astro-ph/0003162] [INSPIRE].

[59] V. Springel, *The cosmological simulation code GADGET-2*, *Mon. Not. Roy. Astron. Soc.* **364** (2005) 1105 [astro-ph/0505010] [INSPIRE].

[60] SDSS collaboration, *SDSS-III: massive spectroscopic surveys of the distant universe, the Milky Way galaxy and extra-solar planetary systems*, *Astron. J.* **142** (2011) 72 [arXiv:1101.1529] [INSPIRE].

[61] BOSS collaboration, *The Baryon Oscillation Spectroscopic Survey of SDSS-III*, *Astron. J.* **145** (2013) 10 [arXiv:1208.0022] [INSPIRE].

JCAP02(2019)031

[62] S. Bird, *FSFE: Fake Spectra Flux Extractor*, Astrophysics Source Code Library, October 2017 [ascl:1710.012].

[63] J.S. Bolton, M. Viel, T.S. Kim, M.G. Haehnelt and R.F. Carswell, *Possible evidence for an inverted temperature-density relation in the intergalactic medium from the flux distribution of the Lyman-α forest*, *Mon. Not. Roy. Astron. Soc.* **386** (2008) 1131 [arXiv:0711.2064] [INSPIRE].

[64] T.S. Kim, J.S. Bolton, M. Viel, M.G. Haehnelt and R.F. Carswell, *An improved measurement of the flux distribution of the Ly-α forest in QSO absorption spectra: the effect of continuum fitting, metal contamination and noise properties*, *Mon. Not. Roy. Astron. Soc.* **382** (2007) 1657 [arXiv:0711.1862] [INSPIRE].

[65] A. Lewis and S. Bridle, *Cosmological parameters from CMB and other data: a Monte Carlo approach*, *Phys. Rev.* **D 66** (2002) 103511 [astro-ph/0205436] [INSPIRE].

[66] D. Foreman-Mackey, D.W. Hogg, D. Lang and J. Goodman, *emcee: the MCMC hammer*, *Publ. Astron. Soc. Pac.* **125** (2013) 306 [arXiv:1202.3665] [INSPIRE].

[67] N. Srinivas, A. Krause, S.M. Kakade and M. Seeger, *Gaussian process optimization in the bandit setting: no regret and experimental design*, *IEEE Trans. Inform. Theory* **58** (2012) 3250 [arXiv:0912.3995].

[68] E. Brochu, V.M. Cora and N. de Freitas, *A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*, arXiv:1012.2599.

[69] M.U. Gutmann and J. Corander, *Bayesian optimization for likelihood-free inference of simulator-based statistical models*, arXiv:1501.03291.

[70] I. Sobol', *On the distribution of points in a cube and the approximate evaluation of integrals*, *Zh. Vychisl. Mat. Mat. Fiz.* **7** (1967) 784 [*U.S.S.R. Comput. Math. Math. Phys.* **7** (1967) 86].

JCAP02(2019)031