Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **511,** 3768–3784 (2022) https://doi.org/10.1093/mnras/stac239
Advance Access publication 2022 January 29

# MATRYOSHKA : halo model emulator for the galaxy power spectrum

Jamie Donald-McCann [1]★, Florian Beutler [2], Kazuya Koyama [1] and Minas Karamanis [2]

[1]*Institute of Cosmology and Gravitation, University of Portsmouth, Dennis Sciama Building, Portsmouth PO1 3FX, UK*
[2]*Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK*

Accepted 2022 January 23. Received 2022 January 17; in original form 2021 October 13

## ABSTRACT

We present MATRYOSHKA, a suite of neural-network-based emulators and accompanying PYTHON package that have been developed with the goal of producing fast and accurate predictions of the non-linear galaxy power spectrum. The suite of emulators consists of four linear component emulators, from which fast linear predictions of the power spectrum can be made, allowing all non-linearities to be included in predictions from a non-linear boost component emulator. The linear component emulators include an emulator for the matter transfer function that produces predictions in ∼0.0004 s, with an error of <0.08 per cent (at $1\sigma$ level) on scales $10^{-4} < k < 10^{1}\ h\ \mathrm{Mpc}^{-1}$. In this paper, we demonstrate MATRYOSHKA by training the non-linear boost component emulator with analytic training data calculated with Halofit, which has been designed to replicate training data that would be generated using numerical simulations. Combining all the component emulator predictions we achieve an accuracy of <0.75 per cent (at $1\sigma$ level) when predicting the real space non-linear galaxy power spectrum on scales $0.0025 < k < 1\ h\ \mathrm{Mpc}^{-1}$. We use MATRYOSHKA to investigate the impact of the analysis set-up on cosmological constraints by conducting several full shape analyses of the real-space galaxy power spectrum. Specifically we investigate the impact of the minimum scale (or $k_{\mathrm{max}}$), finding an improvement of ∼1.8× in the constraint on $\sigma_8$ by pushing $k_{\mathrm{max}}$ from 0.25 to 0.85 $h\ \mathrm{Mpc}^{-1}$, highlighting the potential gains when using clustering emulators such as MATRYOSHKA in cosmological analyses.

**Key words:** methods: data analysis – cosmological parameters – large-scale structure of Universe.

## 1 INTRODUCTION

The next generation of galaxy surveys such as the Dark Energy Spectroscopic Instrument (DESI; DESI Collaboration et al. 2013, 2016) and *Euclid* (Laureijs et al. 2011) will map the spatial distribution of galaxies with unprecedented accuracy. Studying the distribution of galaxies through their two-point clustering statistics has proven to be a valuable tool to understand the expansion history and matter content of the Universe and the nature of dark energy (Cole et al. 2005; Percival et al. 2007; Alam et al. 2017, 2021). Traditionally these analyses have focused on linear or mildly non-linear scales, and put constraints on a cosmological model through scaling parameters such as $f\sigma_8$ (avoiding the need to recompute the full shape (FS) of the power spectrum for each cosmology considered). These choices are motivated by difficulties in accurately and efficiently modelling small non-linear scales, however these small scales are where we have the largest statistical power from measurements of galaxy clustering. Producing accurate theoretical predictions on these small non-linear scales will be essential to extract the highest amount of information from upcoming surveys.

To accurately model galaxy clustering on small scales we rely on numerical simulations of a cosmological volume (Dolag et al. 2008; Kuhlen, Vogelsberger & Angulo 2012; Schneider et al. 2016; Vogelsberger et al. 2020). However these numerical simulations come at considerable computational cost, making it impractical using their direct outputs when fitting theoretical models to observed data. Complex perturbative models based on one-loop perturbation theory can provide accurate predictions for galaxy clustering up to $k \sim 0.3\ h\ \mathrm{Mpc}^{-1}$ (Foreman, Perrier & Senatore 2016; Ivanov, Simonovic & Zaldarriaga 2020; Philcox et al. 2020), and thanks to the FFTlog (Hamilton 2000) these predictions can be made with a tractable computational cost (Simonovic et al. 2018). In principle accurate predictions can be made on smaller scales by going to two-loop order. The matter power spectrum can be accurately predicted with two-loop models up to $k \sim 0.6\ h\ \mathrm{Mpc}^{-1}$ (Senatore 2015; Senatore & Zaldarriaga 2015). The increased number of nuisance parameters that comes with going to two-loop, and difficulty in constructing bias models that are accurate on small scales make this challenging in practice.

*Cosmic emulation* is a method that allows for simulation outputs to be used indirectly at a reasonable computation cost. Generally speaking an emulator is a sophisticated interpolation scheme with the ultimate goal of producing fast predictions that accurately reproduce the output of the model they are designed to replace. Emulators require *training* on a given set of example outputs, what is meant by 'training' when discussing an emulator depends on the specifics of the interpolation scheme implemented in the emulator. A popular choice for the interpolation scheme when developing an emulator for galaxy clustering is Gaussian process (GP) regression. GPs are non-parametric and robust against overfitting, but are generally limited to producing scalar outputs. Using GPs to construct an emulator requires methodological choices that accommodate these scalar predictions. There has been great success combining GPs with

★ E-mail: jamie.donald-mccann@port.ac.uk

a dimensionality reduction procedure such as principal component analysis (PCA; Habib et al. 2007; Heitmann et al. 2009; Kwan et al. 2015; Lawrence et al. 2017; Giblin et al. 2019; Nishimichi et al. 2019). When constructing an emulator in this way the hyperparameters of multiple GPs are optimized to predict the weights of the principal components of the training set. Successfully carrying out a PCA on training data that is often noisy and in a high-dimensional parameter space is non-trivial. GPs can be used in isolation when constructing an emulator (Bird et al. 2019; Zhai et al. 2019; Pedersen et al. 2021). In this case individual GPs are trained to predict the value of a summary statistic of interest for different elements of a data vector (avoiding the need to employ PCA), i.e. different $r_{\rm p}$ bins of the projected correlation function as in Zhai et al. (2019). Neural networks (NNs) have proven to be a viable alternative to GPs (Agarwal et al. 2012, 2014; Alsing et al. 2020). NNs are more susceptible to overfitting but are capable of producing vector outputs, and they also scale better than GPs with larger training sets. This capability of producing vector outputs is advantageous for multiple reasons. If a data vector is very large, optimizing and deploying a GP for each element of that data vector can be computationally expensive. In addition to this, producing predictions for individual elements of a data vector ignores any possible correlations between these elements, while producing vector outputs can preserve some of these correlations.

When developing emulators for the non-linear matter power spectrum, it has proven useful to emulate a *non-linear boost* rather than emulating the matter power spectrum directly (see e.g. Euclid Collaboration et al. 2019, 2021; Angulo et al. 2021). This non-linear boost is given by

$$B(k) = \frac{P_{\rm nl}(k)}{P_{\rm L}(k)}, \tag{1}$$

with $P_{\rm nl}(k)$ being the non-linear power spectrum that would be measured from a simulation, and $P_{\rm L}(k)$ is a prediction of the power spectrum coming from linear theory. The major benefit of emulating this non-linear boost rather than the power spectrum itself is that the boost has a simpler functional form than the power spectrum. We expect $B(k) \approx 1$ on large scales where linear theory holds. The simpler functional form of the boost (and consequently lower dynamic range on large scales) leads to a higher prediction accuracy. A downside of this *boosting* method is that the linear theory predictions can often create a bottleneck. The prediction of the boost coming from the emulator is generally orders of magnitude faster than the linear theory prediction. Several recent works have applied the idea of emulation to predictions coming from linear theory (Aricò, Angulo & Zennaro 2021; Mootoovaloo et al. 2022; Spurio Mancini et al. 2022). These linear theory emulators have uses beyond the boosting method mentioned above as they can also be used to speed up any cosmological analysis that requires a linear theory prediction for the power spectrum. In Aricò et al. (2021) for example, they emulate the linear matter power spectrum prediction to be used in a Lagrangian perturbation theory model for galaxy clustering.

For this work we aim to apply the boosting method mentioned above to the galaxy power spectrum. We also apply emulation to the linear theory predictions, allowing us to exploit the gain in prediction accuracy coming from the boosting method, whilst maintaining the fastest possible prediction speed. The paper is divided as follows. Section 2 describes the suite of emulators that make up MATRYOSHKA. The tests we conduct to evaluate the prediction accuracy of each of the component emulators are outlined in Section 3. Section 4 describes several mock power spectrum FS analyses that are designed to assess how the obtained level of prediction accuracy impacts parameter constraints when using MATRYOSHKA to do model fitting, and how the level of constraint is impacted by the minimum scale included in the analysis. Section 5 contains short discussions about how the boost predicted by the emulator can be used to absorb complex small-scale physics that is difficult to model analytically. We conclude in Section 6.

## 2 EMULATOR DESIGN

MATRYOSHKA is made up of a suite of emulators, predictions from each of these emulators are combined to make a prediction for the non-linear galaxy power spectrum. The goal of MATRYOSHKA is to exploit the gain in accuracy that comes from emulating the non-linear boost rather than the power spectrum directly, whilst maintaining the maximum possible prediction speed by also emulating several quantities that significantly increase the speed of the linear theory predictions for the galaxy power spectrum (hereafter the base model).

This section will describe the design of each of the emulators that make up MATRYOSHKA; the emulated components of the base model are outlined in Section 2.1, Section 2.1.2 describes the parameter space covered by the base model and how the training data for the base model can be focused on a particular suite of simulations, and the details of the galaxy halo connection model and non-linear boost emulator are covered in Section 2.2.

### 2.1 Base model

For our base model we use the halo model (HM) framework (Cooray & Sheth 2002; Murray et al. 2021). The HM is a very popular framework that has been used extensively when making predictions for galaxy clustering. We use the HM as it will allow us to very easily produce equivalent predictions from the base model and from a halo catalogue coming from a numerical simulation.

In the HM framework the clustering of galaxies is split into two regimes, describing the small- and large-scale clustering, respectively,

$$P(k) = P_{\rm 1h}(k) + P_{\rm 2h}(k). \tag{2}$$

In the context of galaxy clustering these two terms are given by

$$P_{\rm gg,\,1h}(k) = \int n(M) \frac{\langle N(N-1)|M\rangle}{n_{\rm g}^2} |u_{\rm g}(k|M)|^2 \,{\rm d}M \tag{3}$$

and

$$P_{\rm gg,\,2h}(k) = P_{\rm L}(k) \left[\int n(M) b_{\rm h}(M) \frac{\langle N|M\rangle}{n_{\rm g}} u_{\rm g}(k|M) \,{\rm d}M\right]^2, \tag{4}$$

with $n(M)$ being the halo mass function, $\langle N|M\rangle$ and $\langle N(N-1)|M\rangle$ being the expected number of galaxies and galaxy pairs, respectively, for a given halo mass $M$, $b_{\rm h}(M)$ being the halo bias, $u_{\rm g}(k, M)$ being the profile that satellite galaxies follow within a halo (in Fourier space), and $P_{\rm L}(k)$ being the linear matter power spectrum. Throughout this work we assume that the satellite galaxies directly follow the Navarro, Frenk & White (1996, hereafter NFW) profile of their host halo. The NFW profile has the form

$$\rho(r|M) \propto \frac{1}{r\frac{c(M)}{r_{\rm h}(M)}\left[1 + r\frac{c(M)}{r_{\rm h}(M)}\right]^2}, \tag{5}$$

where $r_{\rm h}(M)$ is the radius of a halo with mass $M$ assuming spherical haloes, and $c(M)$ is the halo concentration–mass relation. With the use of fitting functions for $n(M)$, $b_{\rm h}(M)$, and $c(M)$ (Duffy et al. 2008), both terms in equation (2) can be calculated analytically at a relatively

small, but non-negligible computational cost. The non-linear boost component emulator (discussed in Section 2.2) produces predictions in ∼0.0004 s.[1] For comparison this is ∼200× faster than the linear power spectrum calculation needed for equation (4). If we were not to emulate the base model, the prediction time of the non-linear galaxy power spectrum would be dominated by the prediction time of the base model. Hence we are motivated to emulate several components of equations (3) and (4).

### 2.1.1 Base model component emulation

We emulate four quantities that allow us to greatly increase the speed of the base model predictions. Those are the matter transfer function $T(k)$, the mass variance $\sigma(M)$, the logarithmic derivative of the mass variance $\frac{\mathrm{d}\ln\sigma(M)}{\mathrm{d}\ln M} \equiv \mathcal{S}(M)$, and the linear growth function $D(z)$. To achieve the best accuracy for a prediction of the power spectrum from the HM, $T(k)$ is normally calculated using a Boltzmann code such as CAMB (Lewis, Challinor & Lasenby 2000) or CLASS (Lesgourgues 2011). There are cheaper analytic alternatives, such as Eisenstein & Hu (1998), but the accuracy on large scales from these cheaper alternatives is not high enough. When developing a boosting emulator it is important that there is good agreement on large scales between the base model prediction and the prediction coming from the numerical simulation, as this is what gives the small dynamic range for $B(k)$ on large scales. The transfer function enters equations (3) and (4) in multiple terms, such as the matter power spectrum:

$$P_L(k) = A_s k^{n_s} T^2(k), \tag{6}$$

where $A_s$ is the amplitude of the primordial power spectrum, and $n_s$ is the spectral index. $T(k)$ also enters indirectly in the halo mass function $n(M)$ and halo bias $b_h(M)$ via the mass variance $\sigma(M)$, given by

$$\sigma^2(M) = \frac{1}{2\pi^2}\int_0^\infty k^2 P_L(k) W^2(kM)\,\mathrm{d}k, \tag{7}$$

and the logarithmic derivative $\mathcal{S}(M)$, given by

$$\mathcal{S}(M) = \frac{3}{2\pi^2 r^4 \sigma^2(M)}\int_0^\infty \frac{\mathrm{d}W^2(kM)}{\mathrm{d}M}\frac{P_L(k)}{k^2}\,\mathrm{d}k. \tag{8}$$

Although we could just emulate $T(k)$ and evaluate equations (7) and (8) directly using the $T(k)$ emulator predictions, these calculations add non-negligible time to the base model prediction, so we decide to emulate $\sigma(M)$ and $\mathcal{S}(M)$ in addition to $T(k)$. In the equations above $W(kM)$ is the Fourier transform of the window function, throughout this work we use a top-hat window function of the form

$$W(kM) = 3\frac{\sin(kM) - kM\cos(kM)}{(kM)^3}. \tag{9}$$

For our base model we use a Tinker et al. (2008) halo mass function, with the form

$$n(M) = \frac{\rho_0}{M^2} f_n[\sigma(M)]\,|\mathcal{S}(M)|, \tag{10}$$

where the function $f_n[\sigma(M)]$ is given by

$$f_n[\sigma(M)] = A\left[\left(\frac{b}{\sigma(M)}\right)^a + 1\right]\exp\left(-\frac{c}{\sigma(M)^2}\right), \tag{11}$$

and the coefficients of the function above are calibrated against simulations in Tinker et al. (2008). For the base model we use a Tinker et al. (2010) halo bias with the form

$$b_h(M) = 1 - A\frac{\nu^a}{\nu^a + \delta_c^a} + B\nu^b + C\nu^c, \tag{12}$$

where $\nu = \delta_c/\sigma(M)$, $\delta_c = 1.686$, and as with equation (11), the coefficients of equation (12) are calibrated against simulations in Tinker et al. (2010).

To avoid including the redshift $z$ as an extra input parameter for the linear component emulators, we include all redshift dependence through the growth factor $D(z)$: $P(k, z) \propto D^2(z)$, $\sigma^2(M, z) \propto D^2(z)$, and $\mathcal{S}(M) \propto D^2(z)/D^2(z)$. This means we can emulate $T(k)$, $\sigma(M)$, and $\mathcal{S}(M)$ at redshift zero, and include all redshift dependence with an emulator for $D(z)$. These four component emulators are what make up the *emulated* base model.

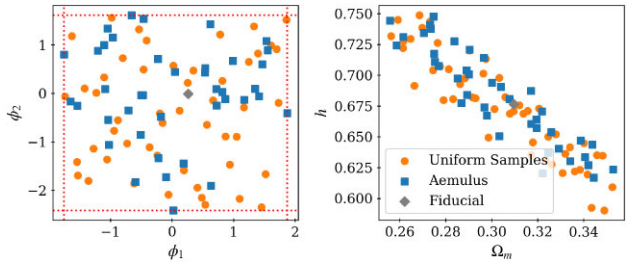


**Figure 1.** Visualization of the procedure of focusing the MATRYOSHKA training data on a suite of simulations. The left-hand panel shows the 40 Aemulus training cosmologies (blue squares) and our fiducial cosmology (grey diamond) in the uncorrelated latent space. The orange circles show 50 random samples in the latent space, and the red dotted lines show the boundaries in the latent space defined by the extreme values of the Aemulus suite. The right-hand panel shows how these 50 samples are distributed in the cosmological parameter space.

### 2.1.2 Parameter space and training data

For this work we focus on the context where the base model will be trained to be used alongside a suite of numerical simulations, such as the Abacus Cosmos (Garrison et al. 2018), Aemulus (DeRose et al. 2019), Dark Quest (Nishimichi et al. 2019), or Quijote (Villaescusa-Navarro et al. 2020) simulations. We choose to consider the Aemulus simulations. This publicly available simulation suite totals 75 dark matter only simulations, each with a volume of $(1.05\ \mathrm{Gpc}\ h^{-1})^3$. 40 of these simulations form a training set that has already been used to successfully train an emulator for the correlation function in redshift space (Zhai et al. 2019). This emulator has recently been used to measure the growth rate from the extended Baryon Oscillation Spectroscopic Survey (eBOSS) luminous red galaxy (LRG) sample (Chapman et al. 2021). These 40 simulations sample a seven-dimensional cosmological parameter space, with parameters $\{\Omega_m, \Omega_b, \sigma_8, h, n_s, N_{\mathrm{eff}}, w_0\}$. The samples are selected to uniformly cover a $4\sigma$ region on these seven parameters coming from analysis of cosmic microwave background (CMB), baryon acoustic oscillations (BAO), and supernovae measurements (for details see section 2 of DeRose et al. 2019). To focus our training data for our base model on the Aemulus simulations, we calculate the covariance of these 40 training simulations, we then use the Cholesky decomposition $\Sigma = \boldsymbol{L}\boldsymbol{L}^{\mathrm{H}}$to decompose the covariance matrix into the lower triangle and corresponding conjugate transpose (indicated by the superscript H). This lower triangle is used to transform the samples that make up the Aemulus training set into an uncorrelated latent space, shown in Fig. 1. We generate 10 000 samples uniformly in each dimension of

[1]This prediction time and all others referred to in this paper are based on predictions made on a laptop with a 2.7 GHz Quad-Core Intel Core i7 CPU.

this latent space, with intervals defined by the latent space samples corresponding to the Aemulus simulations. The lower triangle $L$ can then be used to transform these latent space samples back into the cosmological parameter space. Sampling the parameter space in this way allows us to focus only on the region where we will have simulated training data to combine with our base model, and not extend the training space (at the determent of prediction accuracy) to regions that will not be covered by simulations. It should be noted that in the case where the simulations uniformly cover the parameter space, such as the Dark Quest simulations, this procedure is not necessary. All of the 1D and 2D projections of the cosmological parameter space are shown in Fig. 2. We can see that the 10 000 samples cover the region sampled by the Aemulus simulations whilst minimizing sampling in regions where there are no simulations. Fig. 2 also shows that our fiducial cosmology [based on the most recent *Planck* $\Lambda$ cold dark matter ($\Lambda$CDM) TT, TE, EE+lowE+lensing+BAO analysis, see Section 4.1] that will be used in the various analysis tests in Section 4 is located roughly at the centre of this seven-dimensional parameter space.

The 10 000 samples are split into training, validation, and test subsamples (with 6400, 1400, and 2000 samples, respectively). The components of the emulated base model are trained using the training subsample, the validation subsample is used for model selection, and the prediction accuracy of the emulated base model is tested using the test subsample. Transfer functions and growth factors are calculated for the 10 000 samples using CLASS as implemented in the PYTHON package NBODYKIT (Hand et al. 2018), for 400 logarithmically spaced $k$-bins in the interval $10^{-4} < k < 10\ h\ \mathrm{Mpc}^{-1}$ and at 200 linearly spaced redshifts in the interval $0 < z < 2$. These transfer functions are then used to calculate $\sigma^2(M)$ and $\mathcal{S}(M)$ with the PYTHON package HMF (Murray, Power & Robotham 2013).

## 2.2 Non-linear boost emulator

### 2.2.1 Halo occupation distribution model

A key ingredient when calculating the galaxy power spectrum in the HM framework is the halo occupation distribution (HOD). A HOD model is a probabilistic model that describes the probability of a dark matter halo of mass $M$ hosting $N$ galaxies of a given type. For this work we use the popular Zheng et al. (2009) model. This model is described by five parameters and is split into two terms describing the occupation of central and satellite galaxies separately, where the expected central occupation is modelled as a smoothed step function:

$$\langle N_{\rm cen}|M\rangle = \frac{1}{2}\mathrm{erfc}\left[\frac{\ln M_{\rm cut} - \ln M}{\sqrt{2}\sigma}\right], \tag{13}$$

and the expected number of satellite galaxies is modelled as a power law:

$$\langle N_{\rm sat}|M\rangle = \begin{cases} 0 & \text{if } M < \kappa M_{\rm cut}\,, \\ \left(\frac{M-\kappa M_{\rm cut}}{M_1}\right)^{\alpha} & \text{if } M > \kappa M_{\rm cut}\,. \end{cases} \tag{14}$$

$M_{\rm cut}$ defines the minimum mass for a halo to host a central galaxy, $\sigma$ defines to what extent the central step function is smoothed, the product $\kappa M_{\rm cut}$ defines the minimum mass for a halo to host a satellite, $M_1$ defines the typical mass for a halo to host a satellite, and $\alpha$ defines how the expected number of galaxies increases with mass. We also impose the condition that a halo cannot host a satellite galaxy without first hosting a central galaxy, such that the expected total occupation is given by

$$\langle N|M\rangle = \langle N_{\rm cen}|M\rangle(1 + \langle N_{\rm sat}|M\rangle). \tag{15}$$

In order to calculate $P_{\rm gg,\,1h}(k)$ (equation 3) we need to compute the expected number of pairs for a given halo mass $\langle N(N-1)|M\rangle$. As shown in section 3.1 of Zheng et al. (2005) $\langle N(N-1)|M\rangle$ can be written as

$$\langle N(N-1)|M\rangle = 2\langle N_{\rm cen}N_{\rm sat}|M\rangle + \langle N_{\rm sat}(N_{\rm sat}-1)|M\rangle. \tag{16}$$

Under the assumption that the number of satellite galaxies follows a Poisson distribution, we can write that $\langle N_{\rm sat}(N_{\rm sat}-1)|M\rangle = \langle N_{\rm sat}|M\rangle^2$, and following Miyatake et al. (2021) we approximate $\langle N_{\rm cen}N_{\rm sat}|M\rangle = \langle N_{\rm cen}|M\rangle\langle N_{\rm sat}|M\rangle$ under the central condition. This is a simple HOD model, and for this work this is the HOD model used in the emulated base model, in addition to being used to generate non-linear power spectra that will train the boost emulator. It should be noted that there is no requirement for the galaxy halo connection model to be identical for emulated base model and the non-linear boost component emulator. The only requirement is that it is possible to relate the two models such that power spectra predictions produced by the base model and those coming from the simulation agree on large scales (see Section 5.1).

In Section 4, we conduct a series of mock power spectrum FS analyses for a Baryon Oscillation Spectroscopic Survey (BOSS) CMASS (Dawson et al. 2012) style power spectrum, as such we define the extent of the HOD parameter space to be the same as that from Kwan et al. (2015). This parameter space was designed to cover the HOD models from the analyses of BOSS CMASS galaxies by White et al. (2011). The ranges of the five HOD parameters are given in Table 1.

### 2.2.2 Analytic training spectra

For this work we aim to introduce MATRYOSHKA, and demonstrate how the MATRYOSHKA emulated base model can be focused on a suite of simulations to be used alongside the non-linear boost component emulator that has been trained on data coming from that same suite of simulations. Generating this training data from simulations is cheaper than running the simulations themselves, but still comes at considerable computational cost. With this in mind we opt to generate the training data for the non-linear boost component emulator with Halofit (Takahashi et al. 2012). Non-linear training data are calculated via equation (2), with $P_{\rm L}$ in equation (4) being replaced with the non-linear matter power spectrum, with non-linearities introduced via Halofit. This allows us to very quickly generate data for the non-linear boost component emulator, and demonstrate MATRYOSHKA. We take steps to replicate the scenario where simulated training data are used, such as introducing noise into the Halofit training data that would be present in simulated training data, and only generating training data for the 40 Aemulus training cosmologies. In future work we will train this non-linear boost component emulator directly on simulated training data.

Following a similar procedure to Zhai et al. (2019) we sample the HOD parameter space 50 times for each cosmology, resulting in 2000 training samples for the non-linear boost component emulator. When calculating power spectra associated with these cosmological and HOD parameters we only include scales that would be accessible from a simulation box. The smallest $k$-mode is defined by the fundamental mode of the simulation box,

$$k_{\rm fund} = \frac{2\pi}{L_{\rm box}}, \tag{17}$$

with $L_{\rm box}$ being the length of one side of the simulation box. For a simulation from the Aemulus suite with $L_{\rm box} = 1.05$ Gpc $h^{-1}$ we have $k_{\rm fund} \approx 6 \times 10^{-3}\ h\ \mathrm{Mpc}^{-1}$. To determine the highest $k$-mode that we want to emulate, we consider the smallest scales

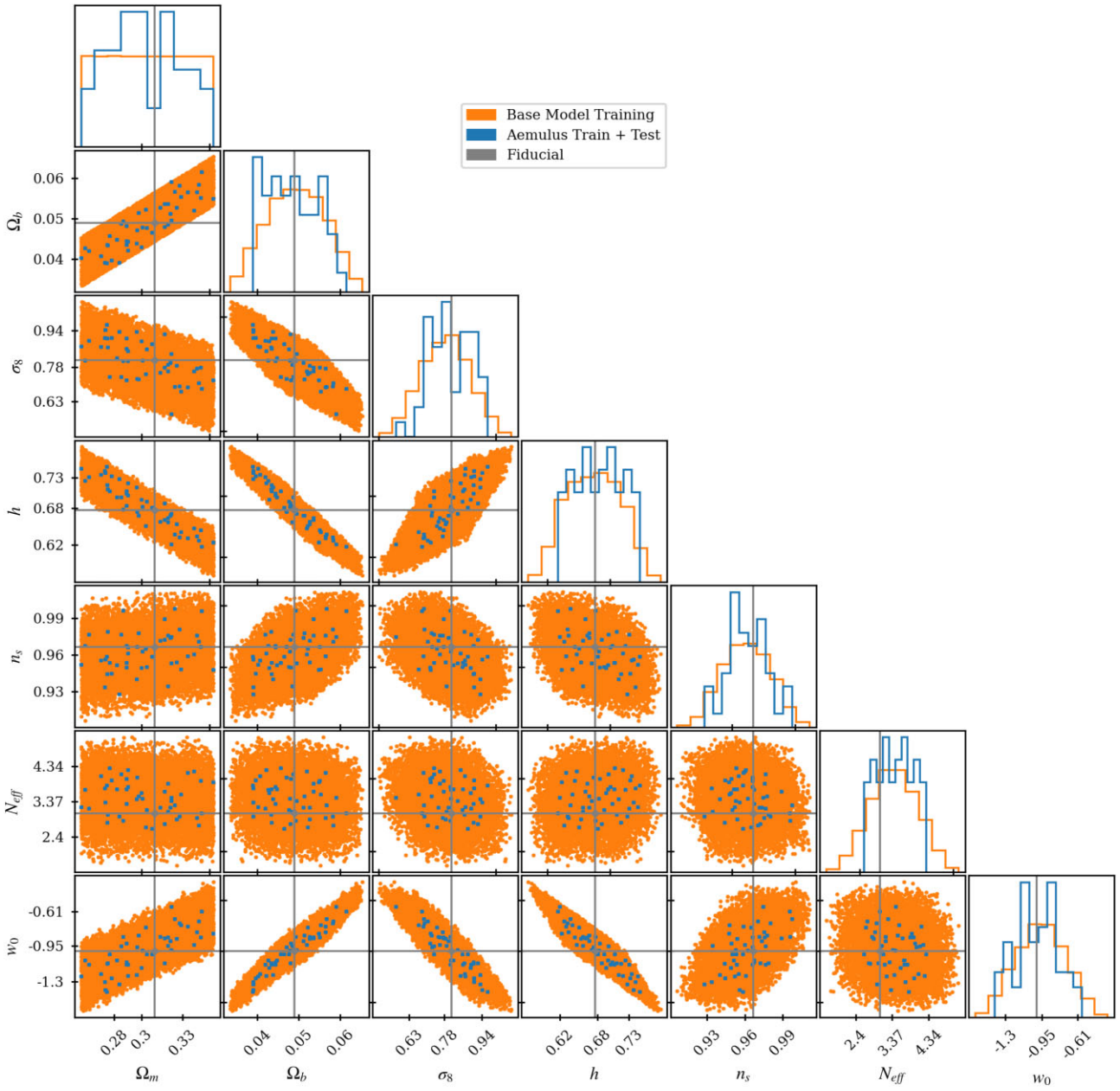


**Figure 2.** 1D and 2D projections of the training space for the cosmological parameters of our model. The blue points and histograms show the Aemulus training and test cosmologies. The sampling of these training cosmologies is influenced by results from CMB, BAO, and supernova experiments (see section 2 of DeRose et al. 2019). The orange points and histograms show the training data for the base model of MATRYOSHKA. This training data has been generated to cover the same region of the parameter space as the Aemulus training cosmologies by uniformly sampling from an uncorrelated latent space defined by the Aemulus training cosmologies (see Section 2.1.2). The grey point and solid lines show the location of the cosmology used to generate the mock power spectrum for the FS analyses described in Section 4.

that can be accurately represented by a dark matter only (DMO) $N$-body simulation. Many works have studied the impact of baryonic effects on the dark matter power spectrum, by comparing the dark matter power spectrum measured from a DMO simulation and hydrodynamic counterpart (Schneider et al. 2019; Aricò et al. 2020; Debackere, Schaye & Hoekstra 2020), and although there is some disagreement between different hydrodynamical codes and baryonic feedback models as to how strong the impact is on the dark matter power spectrum, there is general agreement that for scales where $k \gtrsim 1\ h\ \mathrm{Mpc}^{-1}$ the impact on the dark matter power spectrum is >1 per cent. With this in mind we consider $k \approx 1\ h\ \mathrm{Mpc}^{-1}$ the smallest possible scale that can be accurately modelled without including baryonic effects in the simulation and thus the smallest scale we want to emulate. With these limitations that would come from simulated training data in mind we limit the analytic training data coming from Halofit to 127 $k$-values from $0.012 \lesssim k \lesssim 1.152\ h\ \mathrm{Mpc}^{-1}$.[2]

[2]These values correspond to bin centres from a hypothetical power spectrum measurement covering $k_{\mathrm{fund.}} < k < k_{\mathrm{Nyq.}}/2$, where $k_{\mathrm{Nyq.}} = N_{\mathrm{mesh}}\pi/L_{\mathrm{box}}$ and $N_{\mathrm{mesh}} = 1024$ such that the smallest emulated scale covers at least $k_{\mathrm{Nyq.}}/2$.

**Table 1.** The ranges for each of the HOD parameters used to train the non-linear boost emulator, along with the parameters used to calculate the mock observations for the fiducial FS analysis in Section 4. The extent of the HOD parameter space matches that of Kwan et al. (2015) and is designed to cover the results of White et al. (2011). These results from White et al. (2011) also define our fiducial HOD parameters.

| Parameter | Range | Fiducial value |
|---|---|---|
| $\log M_{\rm cut}$ | [12.9, 13.78] | 13.04 |
| $\log M_1$ | [13.5, 14.7] | 14.05 |
| $\sigma$ | [0.5, 1.2] | 0.94 |
| $\kappa$ | [0.5, 1.5] | 0.93 |
| $\alpha$ | [0.5, 1.5] | 0.97 |

As mentioned above simulated training data come with noise. One source of noise is the sample variance of the simulations in the suite. There are methods to reduce the impact of this sample variance on the training data. In the case of the Aemulus simulations the initial conditions for each of the training simulations are generated with different random seeds, which prevent the emulator learning the noise coming from a specific set of initial conditions. Running phase-matched simulations effectively removed this sample variance (Angulo & Pontzen 2016; Chuang et al. 2019; Klypin, Prada & Byun 2020), however this method doubles the computational cost of producing the training simulations without increasing the density of the sampling in the training space. It is not clear if a suite of phase-matched simulations would outperform a suite with random phases but with twice the sampling density. Another source of noise in simulated training data comes from the procedure used to populate the simulation with galaxies. Populating simulated dark matter haloes according to an HOD is a random process, as such multiple realizations of the same HOD will result in variation of the measured power spectrum. To try and remove some of this HOD realization noise, multiple realizations are normally generated and the power spectrum from each of these realization is averaged. This averaging will remove some but not all of the HOD realization noise. To approximate this leftover noise we take the average of 10 random draws from a multivariate Gaussian with mean being the smooth non-linear power spectrum calculated with Halofit and covariance given by

$$C_{ii} = \frac{(2\pi)^3}{V}\left[\frac{2(P(k_i)+n_{\rm g}^{-1})^2}{4\pi k_i^2 dk}\right]. \tag{18}$$

### 2.3 Neural networks as emulators

A neural network (NN) is a machine learning algorithm structured into *layers* and *nodes*. The nodes take numeric values corresponding to the weighted sum of all the nodes in the previous layer. In the case where we have two layers, with $i$ and $j$ nodes, respectively, each node of the second layer will take the value given by

$$y_j = \sum_i W_{i,j}\, x_i. \tag{19}$$

In the equation above $W_{i,j}$ are the weights connecting the nodes of the first layer $x_i$ to the $j$th node of the second layer. From equation (19) it is clear that a NN structured in this way will only be able to learn functions that correspond to a linear combination of the inputs. To enable the NN to learn non-linear functions of the inputs *activation functions* are applied to each node

$$y_j = f\left(\sum_i W_{i,j}\, x_i\right), \tag{20}$$

where $f(\cdots)$ is the non-linear activation function. The non-linearities introduced by these activation functions allow NNs to become *universal approximators* and make them a good choice when constructing an emulator. When we train a NN given a set of input–output pairs, we adjust the weights $\boldsymbol{W} = \{W_0, \ldots, W_n\}$ such that the output of the NN most closely matches the target function output.

The number of input parameters, layers, nodes in each layer, and outputs define the *architecture* of the NN. For this work we use NNs with a relatively simple architecture. A schematic visualizing the architecture of the transfer function component of the emulated base model described in Section 2.1 is shown in Fig. 3. We can summarize this architecture in a compressed way by writing 5:300:300:300, this allows us to immediately see that the NN has five input parameters, two hidden layers with 300 nodes each, and an output of 300 nodes. All components of the emulated base model share a very similar architecture to the transfer function, which are given in Table 2. The non-linear boost component emulator has an architecture of 12:200:200:127. The hidden layers in all the components of MATRYOSHKA have the rectified linear unit (ReLU) activation applied to their nodes, whilst the input and output layers have no activation applied. These are not unique architectures. They were hand-tuned to achieve sub-per cent accuracy from the MATRYOSHKA predictions on all scales considered (for a discussion on accuracy requirements, see Appendix B). Whilst aiming for sub-per cent accuracy we also try to maintain the simple architectures with a small number of layers and nodes. Simple NNs are less susceptible to overfitting and are more computationally efficient when producing predictions. Starting with a minimal architecture, a single hidden layer with 50 nodes, the numbers of nodes and/or layers is increased until the increase has minimal or negative impact on the optimized value for the loss function. Grid searches or even more sophisticated methods of architecture selection could result in higher prediction accuracy, however they were not necessary to achieve the desired level of accuracy for this work.

We apply pre-processing to the training data for each of the component emulators. The pre-processors all have the form

$$f(x,\theta_i) = \frac{g(x,\theta_i)-\beta(x)}{\alpha(x)}, \tag{21}$$

where $f(x, \theta_i)$ is the processed target function for the $i$th sample in the training set, $g(x, \theta_i)$ is the unprocessed target function, and $\alpha(x)$ and $\beta(x)$ are functions that summarize the training data. With $\alpha(x_i) = \max\{g(x_i,\theta)\} - \beta(x_i)$ and $\beta(x_i) = \min\{g(x_i,\theta)\}$. The result of this pre-processing is that all the training data now lie in the interval [0, 1] for all target functions and at all scales. This has the effect that all scales contribute equally to the loss function, meaning that no scale is preferentially recovered as a consequence of the magnitude of the target function at that scale. The pre-processing also improves stability of the training procedure. Having all scales lying in the interval [0, 1] ensures that the weights do not get too large during training.

When training a NN we would like to optimize the weights $\boldsymbol{W}$ in equation (20) such that we have sufficient prediction accuracy for all possible $\boldsymbol{\theta}$ sampled from the region of our parameter space covered by the training data. However NNs can suffer from *overfitting* when trained on finite data sets (this is particularly a problem for small data sets). Overfitting occurs when the NN is learning details of the training set that are not general, resulting in high prediction accuracy

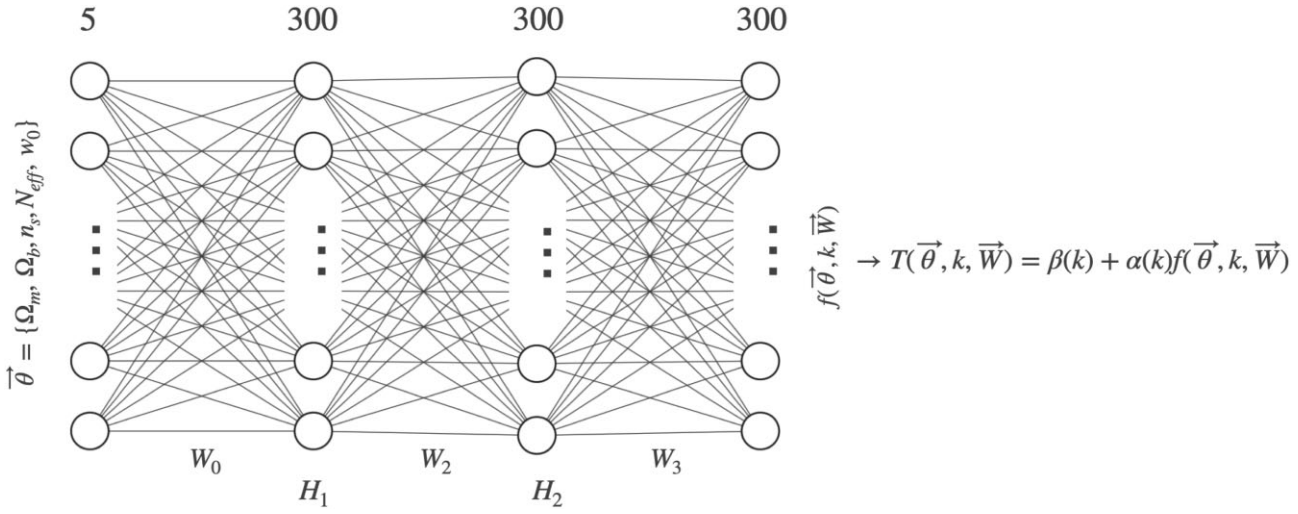


**Figure 3.** Schematic of the $T(k)$ component of the base model. The size of the input and output layers depends on the component being emulated, otherwise all components in the base model have a similar architecture (all base model component architectures are given in Table 2). The nodes in both hidden layers $H_1$ and $H_2$ have the ReLU activation function applied, while there is no activation applied to the input or output layers. The functions $\alpha(k)$ and $\beta(k)$ post-process the NN output for it to be interpreted as $T(k)$.

**Table 2.** Defining the ranges for each of the parameters considered when constructing the base model emulators, along with the parameters used to calculate the mock observations for the fiducial FS analysis in Section 4. We also indicate which parameters are used by which emulator(s) and the architecture of each base model component emulator.

| Parameter | Range | Fiducial value |
|---|---|---|
| $h$ | [0.570, 0.780] | 0.6766 |
| $\Omega_{\rm m}$ | [0.256, 0.353] | 0.30966 |
| $\Omega_{\rm b}$ | [0.0334, 0.0653] | 0.04897 |
| $w_0$ | [−1.58, −0.322] | −1 |
| $N_{\rm eff}$ | [1.61, 5.14] | 3.046 |
| $\sigma_8$ | [0.502, 1.07] | 0.8102 |
| $n_{\rm s}$ | [0.906, 1.01] | 0.9665 |

| Emulator | Parameters | Architecture |
|---|---|---|
| $T(k)$ | $\{\Omega_{\rm m}, \Omega_{\rm b}, h, N_{\rm eff}, w_0\}$ | 5:300:300:300 |
| $D(z)$ | $\{\Omega_{\rm m}, \Omega_{\rm b}, h, N_{\rm eff}, w_0\}$ | 5:200:200:200 |
| $\sigma(M)$ | $\{\Omega_{\rm m}, \Omega_{\rm b}, h, N_{\rm eff}, w_0, n_{\rm s}, \sigma_8\}$ | 7:200:200:500 |
| $\mathcal{S}(M)$ | $\{\Omega_{\rm m}, \Omega_{\rm b}, h, N_{\rm eff}, w_0, n_{\rm s}, \sigma_8\}$ | 7:200:200:500 |

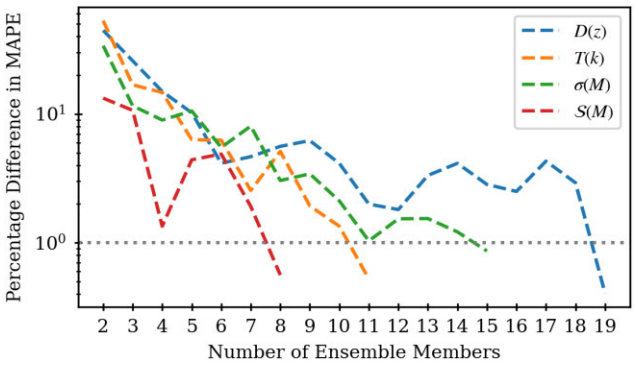


**Figure 4.** Plot showing the percentage difference in the mean absolute percentage error (MAPE) with increasing number of ensemble members when making predictions on the validation set for the base model component emulators. The grey dotted line shows the 1 per cent level.

when making predictions on the training set but poor accuracy when making predictions on unseen data. *Ensembling* is a widely used technique in machine learning to mitigate against the problem of overfitting and improve prediction accuracy on unseen data. To train a basic ensemble of NNs, each NN is trained on the same training data but with randomly initialized weights. When predictions are made using this basic ensemble the predictions from each ensemble member are averaged. An ensemble was trained in this way in Agarwal et al. (2014) when developing an emulator for the dark matter power spectrum. We employ the same ensembling method for this work. For the ensembles that form the emulated base model components we train ensemble members until the addition of new members results is $<1$ per cent change in the prediction accuracy when making predictions on the validation set. Fig. 4 shows how the percentage change in the mean absolute percentage error (MAPE) with increasing numbers of ensemble members for the base model component emulators. The grey dotted line shows the 1 per cent level and the coloured dashed lines show the change in MAPE. We can see that there is some noise present in these coloured lines. This is due to the random initialization of the NN weights, i.e. some NNs will improve the prediction accuracy more than others. To account for this we generate each ensemble multiple times and use the one that performs best on the validation set. When training the ensemble for the non-linear boost component emulator we do not have a validation set so we cannot employ the same procedure as it is important that the test set is not used at any stage of training or ensemble selection to allow us to determine the generalization error of the emulator. From Fig. 4 we can see that a majority of the improvement comes from the first ∼10 ensemble members. With this in mind we train 10 ensemble members for the non-linear boost component emulator.

All the NNs are constructed with TensorFlow (Abadi et al. 2016; TensorFlow Developers 2021). They are all optimized with a mean squared error loss function and Adam optimizer (Kingma & Ba 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$, and a learning rate of 0.013 (the values for $\beta_1$, $\beta_2$, and $\epsilon$ are the TensorFlow defaults, while the value for the learning rate was hand-tuned to achieve sub-per cent accuracy as with the NN architectures). We train each NN for 1000

epochs or until there is a persistent (more than 20 epochs) plateau in the loss function.

## 3 EMULATOR ACCURACY

### 3.1 Base model

To test the accuracy of the emulated components of the base model we use the test subsample, make predictions for each sample in the test set and compare the predictions to the results of calculating the components analytically. These comparisons are shown in Fig. 5, where we show the percentage errors for each of the components of the base model. The green and blue regions show the 68 per cent and 95 per cent confidence intervals (CIs), respectively. We can see that these shaded regions are all centred on 0 per cent, indicating that all component emulators produce unbiased predictions on average. We can also see that all component emulators are producing predictions with sub-per cent accuracy for almost all 2000 samples in the test set (the prediction accuracy is better than 0.1 per cent for the 68 per cent CI). In the top two panels of Fig. 5 there is a sample that performs considerably worse than the others. This sample represents a very extreme cosmology (with $w_0 \approx -0.40$), we can see that even in this extreme case the highest level of prediction error from the transfer function component emulator is ≈1.25 per cent.

The top panel of Fig. 5 clearly shows the highest levels of error in predictions around the BAO scale ($k \sim 0.2\ h\ \mathrm{Mpc}^{-1}$), where the BAO wiggles make these scales of $T(k)$ particularly difficult to predict. A higher level of accuracy could be achieved on these scales by increasing the number of $k$-modes around these scales, however this would mean that these scales have a greater contribution to the loss function and as such the weights of the NN would be optimized to preferentially recover these scales. For this work we aim to obtain a prediction accuracy of $< 1$ per cent at 68 per cent CI (see Appendix B on accuracy requirements) on the non-linear galaxy power spectrum from MATRYOSHKA. As is shown in Section 3 this is achievable without exploring this solution and therefore we leave it to future works.

Fig. 6 shows the response of the transfer function to various parameters of our cosmological model, when calculating the transfer function with CLASS and making a prediction for the transfer function with the emulator. Fig. 6 shows that the emulator well recovers the response of the transfer function for all the relevant parameters of our model, however there is some indication of a slight bias in the prediction of the transfer function on small scales for increasingly negative values of $w_0$. This bias is $\ll 1$ per cent and only occurs for extreme values of $w_0$, so we do not expect this small bias to have a significant impact when using the emulator to make predictions for less extreme cosmologies.

### 3.2 Non-linear boost

To test the performance of the non-linear boost emulator we calculate non-linear boosts for the seven Aemulus test cosmologies. As with the training data we generate power spectra for 50 different HOD parameter sets for each cosmology, which results in 350 test samples. We generate two versions of this test set: the first contains the same level of noise as the training data and the second contains no noise. The reasons for generating these two versions of the same test set is clear when looking at Fig. 7, which shows percentage error on the predictions from the non-linear boost component emulator. The green shaded region shows the prediction error (68 per cent CI) on the noise-free test set and the green dashed line shows the prediction

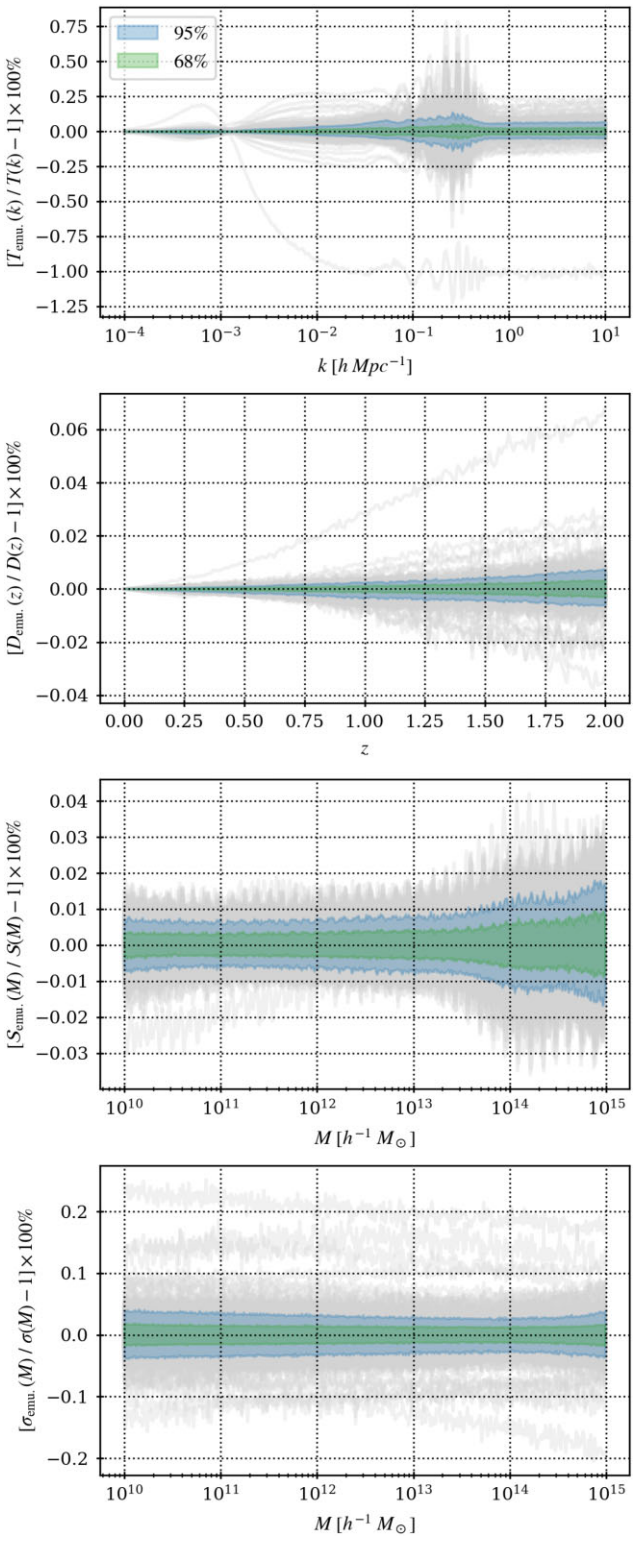


**Figure 5.** Percentage errors on the test set for each of the components of the base model. The panels from top to bottom correspond to the transfer function, growth factor, mass variance, and logarithmic derivative of the mass variance. The grey lines show the percentage error for each sample in the test set, the shaded regions show the 95 per cent and 68 per cent CIs.

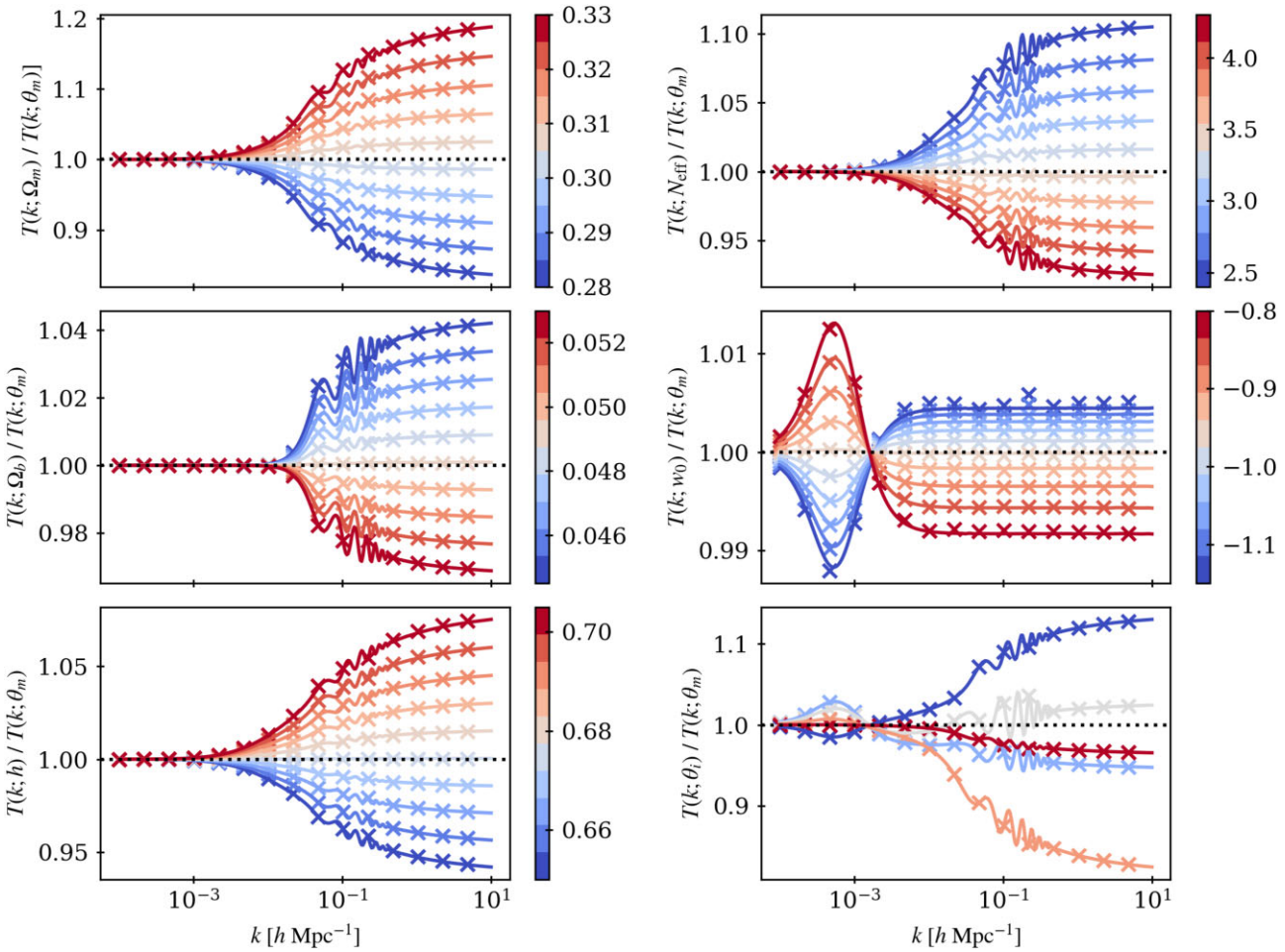


**Figure 6.** Plots showing the response of the transfer function to relevant parameters of our cosmological model. All panels except the bottom right show the effect of varying a single parameter of the model, the bottom right-hand panel shows five random parameter sets. In all panels the ratio is with respect to the transfer function calculated for parameters that correspond to the mean of the training space shown in Fig. 2. The solid lines show responses of transfer functions calculated with CLASS, the crosses show the emulator predictions for the same parameters. In all except the bottom right-hand panel the lines and crosses are coloured by the value of the parameter that is being varied, in the bottom right-hand panel the colour corresponds to the index of the test set from which the parameter sets were randomly selected.

error on the noisy test set. We can see that for $k \gtrsim 0.8\ h\ {\rm Mpc}^{-1}$ the green dashed line and shaded region agree while on larger scales we can see that the prediction error calculated using the noisy test set very closely follows the noise level of the noisy test set. This indicates that the calculated prediction error is dominated by the noise on these scales, which is confirmed by looking at the prediction error on the noise-free test set on the same scales. The NNs are able to predict the non-linear boost at higher accuracy than the noise level of the training data because the noise is random and therefore the noise averages across the training set. When using a simulation suite it is not possible to generate a noise-free test set. The test simulations in the Aemulus suite do however have multiple realizations, such that the noise level in the test set generated from these simulations is lower than that of the training set allowing for an accurate assessment of the prediction accuracy.

From Fig. 7 we can see that the non-linear boost component emulator has $\lesssim$0.5 per cent prediction error from $0.01 \lesssim k \lesssim 1\ h\ {\rm Mpc}^{-1}$. The level of prediction error is largest on the smallest scales. This is where the dynamic range of the non-linear boost is largest. This prediction error gets smaller going to larger scales, however there is a spike in prediction error at $k \sim 0.1\ h\ {\rm Mpc}^{-1}$. This spike in prediction error is a result of the noise level of the training set having an impact on the NNs ability to learn the non-linear boost on these scales. To reduce the impact of this when producing predictions for the non-linear galaxy power spectrum (by combing the predictions of the boost with the base model prediction), the predictions of the non-linear boost can be smoothly 'stitched' with those that we expect from linear theory (that being $B(k) \approx 1$ on large scales) as was done in Kobayashi et al. (2020). We are able to produce predictions for the non-linear galaxy power spectrum with MATRYOSHKA with $<1$ per cent error (at 68 per cent CI) without exploring this solution, as such we leave this for future works.

We can examine the relative contributions from the emulated base model and the non-linear boost component emulator to the prediction error on the non-linear galaxy power spectrum. Fig. 8 compares the $1\sigma$ prediction errors on the non-linear galaxy power spectrum to those on: the non-linear boost, the emulated base model (which is calculated with predictions from all the base model component emulators), and the linear matter power spectrum (which is calculated with predictions from the transfer function component emulator). We can immediately see that the prediction error from the non-linear boost is dominating on all scales. We can also see that on small scales

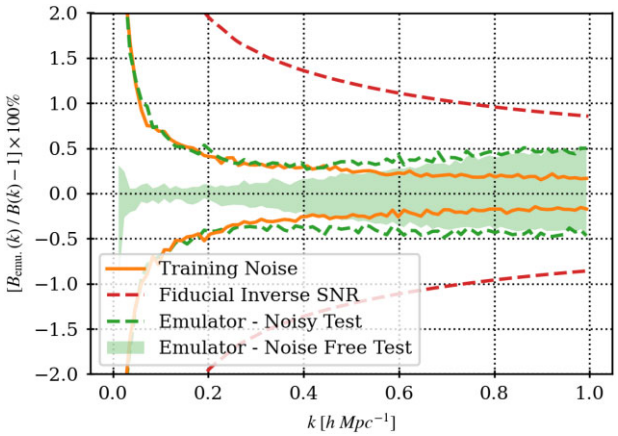


**Figure 7.** Prediction error from the non-linear boost component emulator compared to the statistical error of our fiducial mock and the noise level of the training data. The green shaded region represents the true $1\sigma$ prediction error of the emulator, the green dashed line represents the $1\sigma$ prediction error measured when evaluating the emulator predictions with a noisy test data. The orange solid line shows the $1\sigma$ noise level in the non-linear boost training data, and the red dashed line shows the inverse signal-to-noise ratio (SNR) of our fiducial mock observation in per cent.

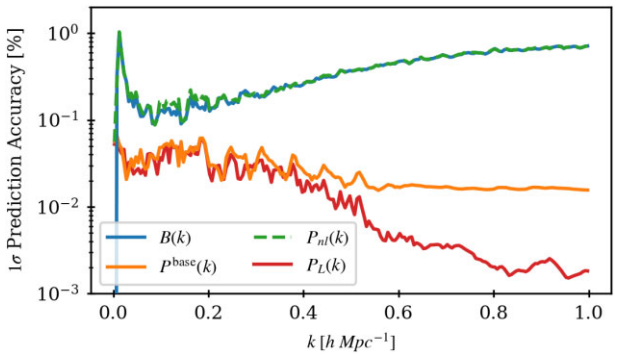


**Figure 8.** Plot showing the relative contributions to the prediction error on the non-linear galaxy power spectrum $P_{\rm nl}(k)$. The prediction error from each of the component emulators contributes to the overall prediction error on $P_{\rm nl}(k)$. For simplicity we have only shown the error coming from the non-linear boost $B(k)$ component emulator, the error in the base model $P^{\rm base}(k)$ (which is calculated with predictions from all the base model component emulators), and the error in the prediction of the linear matter power spectrum $P_{\rm L}(k)$ (which is calculated with predictions from the transfer function component emulator).

the contribution to the error from the linear power spectrum (and thus the transfer function) is lower than the emulated base model. This is to be expected as on these small scales the one-halo term (equation 3) is dominating, and the prediction errors from the other base model component errors are more significant.

The prediction accuracy of the non-linear boost component emulator (the dominating component of the MATRYOSHKA prediction error) is compared to the inverse signal-to-noise ratio (SNR) of our fiducial mock in Fig. 7 [with a volume of (1 Gpc $h^{-1}$)$^3$ and number density of $\sim 6 \times 10^{-4}$ (Mpc$^{-1}$ $h$)$^3$]. We can see that the prediction error from the non-linear boost component emulator is significantly lower than the statistical error of our fiducial mock on all scales considered. This implies that the achieved level of prediction accuracy is high enough to produce predictions for the power spectrum that are consistent with the truth within the statistical errors of our mock analyses set-up (see Appendix B for a discussion on how the required prediction accuracy depends on sample considered).

Fig. 9 shows the response of the non-linear galaxy power spectrum to the cosmological and HOD parameters of our model. We can see that the response is generally well recovered by MATRYOSHKA, particularly on large scales, however the response is not well recovered on small scales for all parameters. This is most apparent in the response to $w_0$, where we can see that the response is under predicted on small scales. This effect is greater the further the value of $w_0$ deviates from $-1$. It should be noted that the response of the non-linear galaxy power spectrum to $w_0$ is very small. The difference between the power spectra for $w_0 = -1.15$ and $-0.8$ is $\sim$2 per cent. This response is considerably smaller than any of the other parameters. The result of this is that the response to $w_0$ has the least impact on the loss function. This effect is exaggerated by the noise in the training data for the non-linear boost component emulator. The $1\sigma$ noise level at $k = 0.8\ h$ Mpc$^{-1}$ is $\sim$0.2 per cent, which is similar to the difference between power spectra calculated with $w_0 = -1.15$ and $-1.11$.

## 4 MOCK FULL SHAPE ANALYSES

In this section, we test MATRYOSHKA by doing several full shape (FS) analyses of a mock observed power spectrum. These analyses are summarized in Table 3. These mock analyses aim to verify that we can recover unbiased constraints on the input cosmology when conducting a FS analysis with MATRYOSHKA. They also allow us to investigate the potential gain in constraining power by including smaller scales when conducting a FS analysis of the power spectrum.

### 4.1 Mock observation

We produce a mock observed power spectrum designed to approximate the power spectrum of BOSS CMASS galaxies, with non-linearities coming from Halofit as with the non-linear training data (see Section 2.2.2). The cosmological and HOD parameters corresponding to this mock observation are shown in Tables 2 and 1, respectively. The cosmological parameters come from the most recent *Planck* ΛCDM TT, TE, EE+lowE+lensing+BAO analysis (table 2 in Planck Collaboration VI 2020, henceforth *Planck* 2018). The HOD parameters are the best-fitting parameters that result from the small-scale clustering analysis of BOSS CMASS galaxies conducted by White et al. (2011). The number density associated with this mock observation is $\sim 6 \times 10^{-4}$ (Mpc$^{-1}$ $h$)$^3$. It should be noted that this number density is slightly greater than the observed CMASS number density. This value corresponds to the number density calculated using the *Planck* 2018 cosmological parameters and White et al. (2011) best-fitting HOD parameters with the equation

$$n_{\rm g} = \int n(M)\langle N|M\rangle \, {\rm d}M. \tag{22}$$

We include linearly spaced $k$-bins covering the range $0.0025 < k < 0.85\ h$ Mpc$^{-1}$ with $\Delta k = 0.005\ h$ Mpc$^{-1}$. These scales are selected such that the $k$-bins included in our fiducial analysis (with $k_{\rm max} = 0.25\ h$ Mpc$^{-1}$) match those from Ivanov et al. (2020), which used the perturbation-theory-based Effective Field Theory of Large Scale Structure (EFTofLSS) approach to analyse multipoles of the power spectra of BOSS galaxies.

Fig. 10 shows our mock observation with grey points. We calculate an uncertainty for this mock observation using equation (18) with

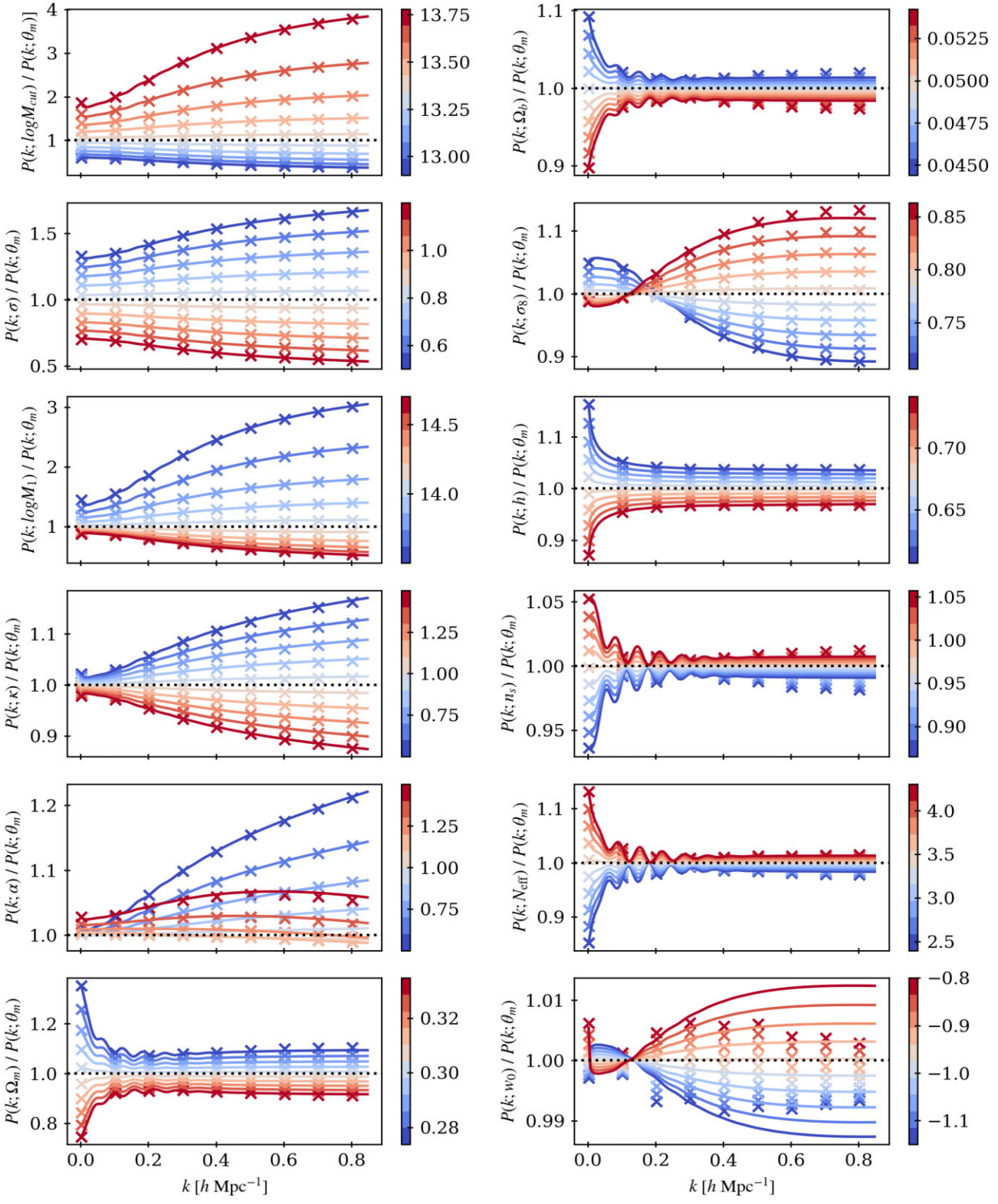


**Figure 9.** Similar to Fig. 6, but for the response of the non-linear galaxy power spectrum to the cosmological and HOD parameters of our model.

a volume of (1 Gpc $h^{-1}$)$^3$. This uncertainty is shown with the grey shaded region in Fig. 10. The MATRYOSHKA prediction for the fiducial parameters is shown with the orange solid line. We can see that the MATRYOSHKA prediction only becomes distinguishable from the truth at very small scales ($k \gtrsim 0.5\,h\,\mathrm{Mpc}^{-1}$), but is still consistent with the truth at the $1\sigma$ level. The FS analyses that follow will determine to what extent this small error in the prediction of the power spectrum on small scales impacts the constrained cosmology.

**Table 3.** Summarizing the different FS analyses described in Section 4. Unless otherwise stated all analysis set-ups have $k_{\rm max} = 0.025$ ($h$ Mpc$^{-1}$).

| Set-up | $k_{\rm max}$ ($h$ Mpc$^{-1}$) | Notes |
|---|---|---|
| Fiducial | 0.25 | Our fiducial FS analysis. Designed to mimic the scales used in an EFTofLSS style analysis. |
| Including number density | 0.25 | Repeats of our fiducial analysis with the inclusion of the number density in the likelihood as in equation (25). |
| Small scales | [0.45–0.85] | Repeats of our fiducial analysis with increased values of $k_{\rm max}$. |
| Fixed HOD | [0.25–0.85] | Repeats of our fiducial analysis with all HOD parameters fixed to the truth, and with the increased values of $k_{\rm max}$ of the small-scales analyses. |

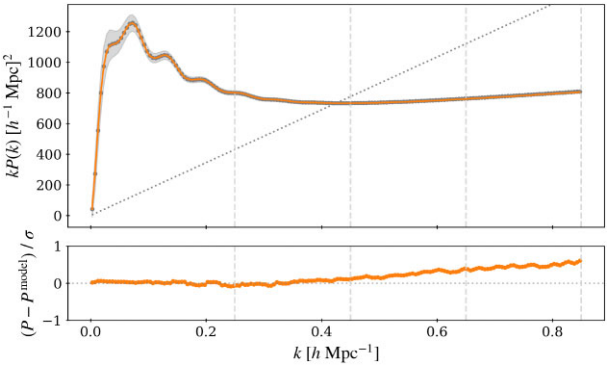


**Figure 10.** The top panel shows the mock CMASS power spectrum described in Section 4.1 (grey points and shaded region), as well as the MATRYOSHKA prediction for the fiducial parameters (solid orange line). The vertical dashed lines show the $k_{\rm max}$ values of the FS analyses. The dotted line shows the shot noise level of our mock observation. The bottom panel shows the normalized residuals comparing our mock observation to the MATRYOSHKA prediction.

### 4.2 Fiducial analysis

For our fiducial analysis we fit four out of the five HOD parameters of our model ($\log M_{\rm cut}$, $\sigma$, $\log M_1$, and $\alpha$) and five out of the seven cosmological parameters ($\Omega_{\rm m}$, $\sigma_8$, $h$, $n_{\rm s}$, and $w_0$). We fix $\kappa$ to its true value as it is not well constrained by the real space power spectrum for the scales considered in our fiducial analysis (or any of the FS analyses that follow). The purpose of the analyses of this section is to verify that unbiased cosmology can be recovered, as such fixing $\kappa$ to the truth in this way does not influence any conclusions we draw. We also fix $\Omega_{\rm b}$ and $N_{\rm eff}$ to their true values. We do not expect to get competitive constraints on these parameters from the real-space power spectrum. It is common practice to use a very tight prior on $\Omega_{\rm b}$ informed by big bang nucleosynthesis, and to fix $N_{\rm eff} = 3.046$ to align with standard model predictions.

We use Markov chain Monte Carlo (MCMC) sampling to calculate the posterior distributions of HOD and cosmological parameters. We define a Gaussian likelihood with the form

$$\ln \mathcal{L}(d|\theta, \phi) = -\frac{1}{2}(P - \tilde{P})^{\rm T}\mathbf{C}^{-1}(P - \tilde{P}), \tag{23}$$

where $P$ is the mock observed galaxy power spectrum, $\mathbf{C}$ is the Gaussian covariance matrix calculated using equation (18) shown by the grey shaded region in Fig. 10, and $\tilde{P}$ is the emulated galaxy power spectrum. We do not include any information about the galaxy number density in the likelihood for our fiducial analysis, however the number density is very sensitive to the HOD parameters. We explore the impact of including number density information in the likelihood in Section 4.3.

We use flat priors on all of the free HOD parameters with ranges equivalent to the extent of the HOD training space (see Table 1). For the cosmological parameters we use a multivariate Gaussian prior with mean and covariance defined by the training samples for the emulated base model shown in Fig. 2. This is a very wide prior, as mentioned in Section 2.1.2, which covers a $4\sigma$ region coming from previous CMB, BAO, and supernovae analyses. The use of this multivariate Gaussian prior on the cosmological parameters is necessary to assign low probability to areas of the parameter space that have not been sampled with training data, as the predictions from the emulators will not be accurate in these regions of the parameter space. MCMC sampling is done using ZEUS (Karamanis, Beutler & Peacock 2021), ZEUS uses ensemble slice sampling, a method that is robust when sampling from challenging distributions that is often the case for HOD parameters. Convergence of MCMC chains is discussed in Appendix A.

The posterior distributions calculated from our fiducial analysis are shown in Fig. 11 with blue filled contours. We can see that the true cosmological parameters are recovered within $1\sigma$, verifying that the obtained level of prediction accuracy from MATRYOSHKA is sufficient to return unbiased cosmology for our mock. We also show the marginalized posterior distributions for the effective galaxy bias $b_{\rm eff.}$ in Fig. 11. $b_{\rm eff.}$ is not a free parameter but can be calculated from the cosmological and HOD parameters with the equation

$$b_{\rm eff.} = \frac{1}{n_{\rm g}}\int n(M) b_{\rm h}(M) \langle N|M\rangle. \tag{24}$$

We can see there is a strong, and expected, degeneracy between $b_{\rm eff.}$ and cosmological parameters that primarily impact the amplitude of the power spectrum such as $\sigma_8$. The effective bias is sensitive to the HOD parameters, which are more tightly constrained by small scales. Therefore we expect to see an improved constraint on the cosmological parameters when including smaller scales. This improvement is coming from the increase in statistical power, along with the improved constraint on the HOD parameters, and thus the effective bias.

### 4.3 Impact of number density

In our fiducial analysis we do not include any information about the number density in our likelihood. The number density is sensitive to the HOD parameters. Works such has Zhou et al. (2021) and Lange et al. (2022) have included information about the number density via an extra term in the likelihood, such that

$$\ln \mathcal{L}(d|\theta, \phi) = -\frac{1}{2}\left[(P - \tilde{P})^{\rm T}\mathbf{C}^{-1}(P - \tilde{P}) + \frac{(n_{\rm g} - \tilde{n_{\rm g}})^2}{\sigma^2_{n_{\rm g}}}\right], \tag{25}$$

where $P$, $\tilde{P}$, and $\mathbf{C}$ are the same as in equation (23), $n_{\rm g}$ is the observed number density, $\tilde{n}_{\rm g}$ is the number density predicted by the model that can be calculated via equation (22), and $\sigma_{n_{\rm g}}$ is the uncertainty on the observed number density. Miyatake et al. (2021) noted very little change to inferred cosmological parameters by including information about the number density when analysing projected clustering and weak lensing of a CMASS-like sample. To investigate the impact of

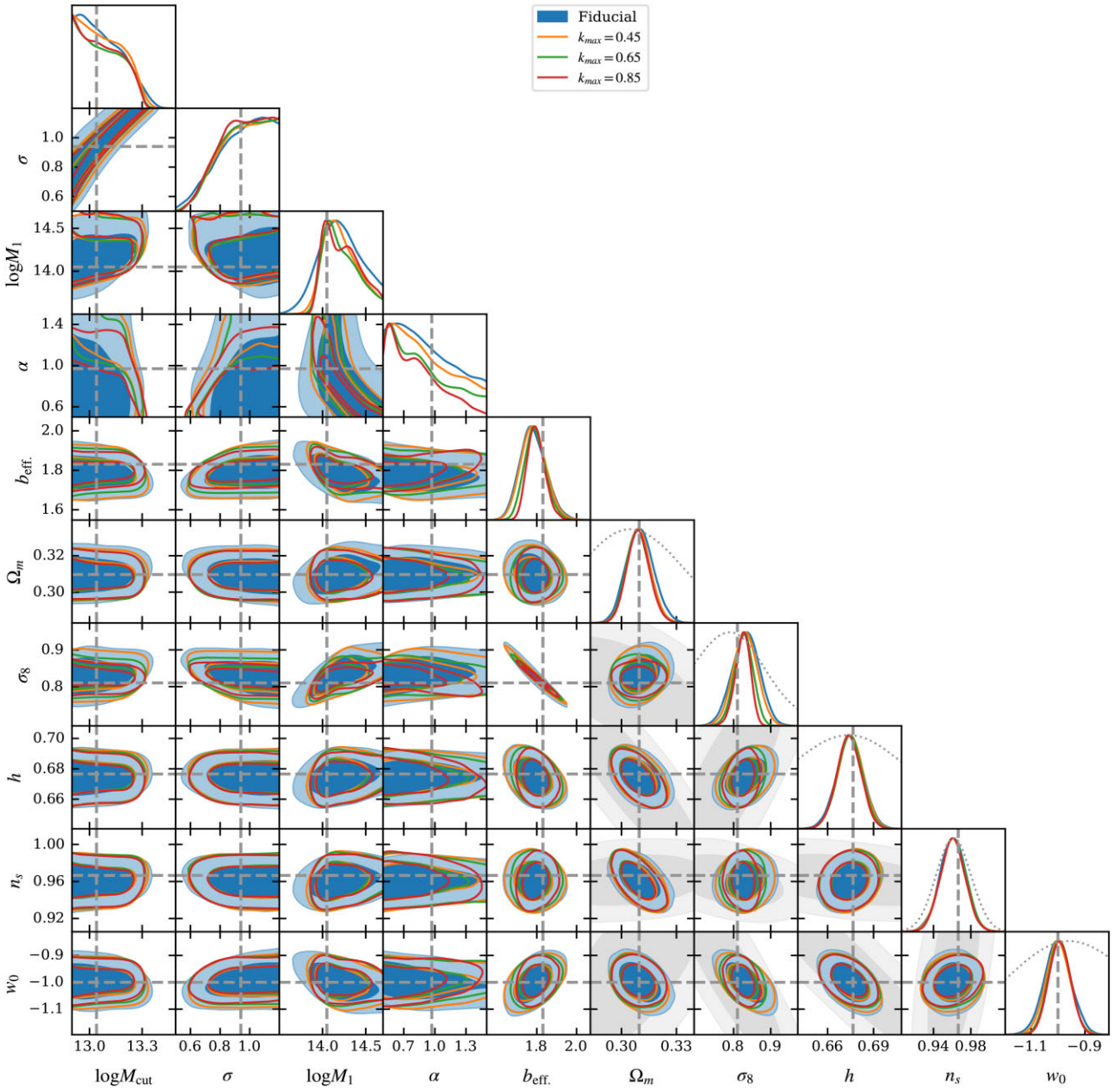


**Figure 11.** Marginalized 1D and 2D posterior distributions resulting from the FS analyses described in Sections 4.2 and 4.4. The two contour levels in the off-diagonal panels represent the $1\sigma$ and $2\sigma$ regions. The grey dashed horizontal and vertical lines show the true cosmological and HOD parameters. The grey contours in the off-diagonal panels and grey dotted lines in the diagonal panels show the multivariate prior on the cosmological parameters. All model parameters not shown are fixed to the truth.

the number density when doing a FS analysis of the power spectrum, we rerun our fiducial analysis with $\sigma_{n_{\rm g}} = [0.1, 0.05, 0.01]n_{\rm g}$. The decreasing values of $\sigma_{n_{\rm g}}$ have a similar impact to placing tighter priors on the HOD parameters.

The posterior distributions of the HOD parameters resulting from these analyses are shown in Fig. 12, alongside the results from the fiducial analysis for comparison. The cosmological parameters are not shown as the difference between the inferred cosmology from these analyses is minimal (the marginalized 1D posteriors for the cosmological parameters are shown in Fig. 13 for reference), however we do show the marginalized posteriors for the number density predicted by our model. We can see that even for our fiducial analysis the true value of $n_{\rm g}$ is well recovered. We can also see that even a relatively high value of $\sigma_{n_{\rm g}}$ significantly improves the constraint on some of the HOD parameters, particularly $\log M_{\rm cut}$. This is to be expected as $\log M_{\rm cut}$ directly controls the number of central galaxies (which make up the majority of a CMASS-like sample), and thus the number density.

### 4.4 Increasing $k_{\rm max}$

To investigate the impact of the minimum scale included when conducting a FS analysis of the power spectrum on the constraint on the cosmological and HOD parameters, we rerun our fiducial analysis pushing the value of $k_{\rm max}$ to $[0.45, 0.65, 0.85]\,h\,{\rm Mpc}^{-1}$. The

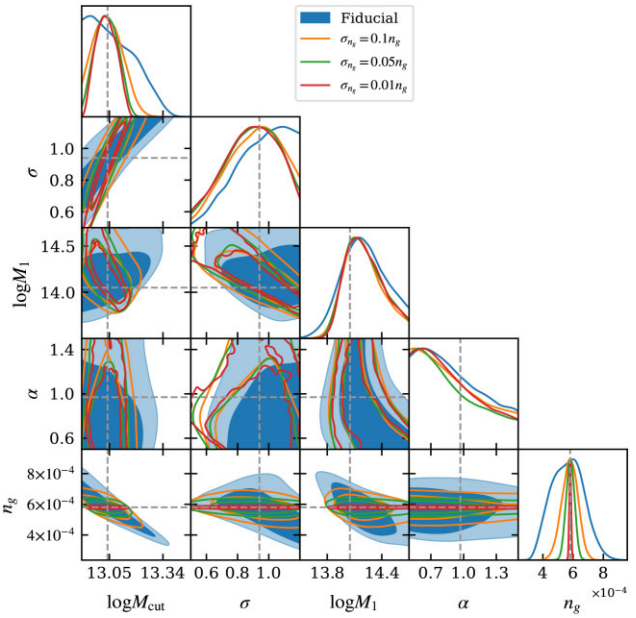


**Figure 12.** Similar to Fig. 11. The blue filled contours are the same as in Fig. 11. The empty contours show the results of the number density analyses described in Section 4.3. Each of the empty contours shows the results of a FS analysis over the same scales of the fiducial analysis with different levels of assumed uncertainty on the number density from the mock. Only the HOD parameter posteriors are shown here as the impact on the cosmological parameters is negligible, see Fig. 13 for the 1D cosmological posteriors.

1D and 2D marginalized posteriors resulting from these analyses are shown in Figs 11 and 13. As expected we see significant improvement in the constraint on all cosmological parameters by including smaller scales. This is particularly true for $\Omega_{\rm m}$ and $\sigma_8$. Our fiducial analysis results in a ∼8.7 per cent (∼4.8 per cent) constraint on $\sigma_8$ ($\Omega_{\rm m}$). Pushing the minimum scale to $k_{\rm max} = 0.85\ h\ {\rm Mpc}^{-1}$ results in a ∼4.9 per cent (∼3.9 per cent) constraint on $\sigma_8$ ($\Omega_{\rm m}$), which represents a ∼1.8× (∼1.2×) improvement. This improvement is coming from the higher statistical power of smaller scales in addition to the improved constraint on the HOD parameters that arises from the increasing magnitude of the response of the power spectrum to the HOD parameters shown in Fig. 9.

We can see from Figs 11 and 13 that there is a small ($<1\sigma$) bias in the median values (blue squares in Fig. 13) of the marginalized posteriors from these analyses. The origin of this bias is likely the small error in the emulator predictions of the power spectrum. This effect is exaggerated by degeneracies between model parameters. For example the median value for $\sigma_8$ is overpredicted whilst $b_{\rm eff.}$ is underpredicted. Improving the constraint on $b_{\rm eff.}$ by including smaller scales (or fixing the HOD as described in Section 4.5) reduces the observed bias in $b_{\rm eff.}$ and $\sigma_8$.

### 4.5 Fixed HOD

To investigate to what level the cosmological parameter constraints are degraded by fitting the HOD parameters, we rerun our fiducial analysis with all HOD parameters fixed to the truth, and for the same $k_{\rm max}$ values used in Section 4.4. The results of these fixed HOD analyses are shown in Figs 13 and 14. The orange contours show the results that cover the same scales as our fiducial analysis. We can see that compared to the results from the fiducial analysis (blue filled contours) there is a significant increase in the constraint on all cosmological parameters even when the same scales are considered. The improvement on the constraint on $\sigma_8$ is ∼2.0×, which is already larger than the improvement from pushing to $k_{\rm max} = 0.85\ h\ {\rm Mpc}^{-1}$ in Section 4.4. This result is expected and demonstrates just how much fitting the HOD parameters degrades the constraint on the cosmological parameters, and highlights that accurate prior information about the HOD coming from small-scale clustering studies (such as Zheng, Coil & Zehavi 2007; White et al. 2011; Beutler et al. 2013; Parejko et al. 2013; Zhai et al. 2017) can greatly improve constraints on cosmology coming from a FS analysis of the power spectrum in the HM framework. We can also see that increasing $k_{\rm max}$ compared to our fiducial analysis results in further improvement to the constraint on cosmology. Pushing the fixed HOD analysis to $k_{\rm max} = 0.45$ results in a ∼4.8× improvement to the constraint on $\sigma_8$, however we notice that the improvement including scales smaller than this is much less significant. The dotted line in Fig. 10 shows the shot noise of our mock observation. We can see that when $k \sim 0.4\ h\ {\rm Mpc}^{-1}$ the shot noise is the same magnitude as our clustering signal. As such, increasing $k_{\rm max}$ to values $\lesssim 0.4\ h\ {\rm Mpc}^{-1}$ will result in a higher SNR, however increasing $k_{\rm max}$ to values $\lesssim 0.4\ h\ {\rm Mpc}^{-1}$ will not. The reason we see continued gain when pushing $k_{\rm max} \gtrsim 0.4\ h\ {\rm Mpc}^{-1}$ for the analyses of Section 4.4 but not these fixed HOD analyses is due to the scale dependence of the response to the HOD parameters shown in Fig. 9. We can see that increasing $k_{\rm max}$ increases sensitivity to all the HOD parameters, however for the cosmological parameters there is no scale dependence beyond $k_{\rm max} \gtrsim 0.4\ h\ {\rm Mpc}^{-1}$.

## 5 DISCUSSION

### 5.1 Inclusion of additional effects through $B(k)$

The base model of MATRYOSHKA uses fitting functions for quantities such as the halo mass function and the concentration–mass relation. Furthermore, for this work we are ignoring redshift-space distortions (RSD) and effects such as halo exclusion. The base model of MATRYOSHKA has been designed to be used alongside simulations. Effects that cause $B(k)$ to differ significantly from unity on large scales will need to be included in the base model in future work. On the other hand, effects that predominately impact small scales (that can be challenging to model analytically in some cases) do not to necessarily be included as they can effectively be absorbed into the prediction of $B(k)$. For example, RSD can be modelled on large scales with a Kaiser factor

$$P_{\rm s}^{\rm base}(k,\mu) = P(k)(1+f\mu^2)^2, \tag{26}$$

while the impact of RSD on small scales is more challenging to model analytically. To make sure the small-scale RSD is included in the MATRYOSHKA predictions without modelling them analytically, we can decompose $P_{\rm s}^{\rm base}(k,\mu)$ into multipoles,

$$P_\ell^{\rm base}(k) = \frac{2\ell+1}{2}\int_{-1}^{1} L_\ell(\mu) P_{\rm s}^{\rm base}(k,\mu), \tag{27}$$

with $L_\ell$ being the $\ell$th-order Legendre polynomial. We can then create non-linear boost component emulators for each multipole $B_\ell(k) = P_\ell(k)/P_\ell^{\rm base}(k)$. The effect of RSD on small scales will already be included in the simulated $P_\ell(k)$ such that the neglected small-scale RSD in the base model is captured by the prediction of the boost $B_\ell(k)$,

$$B_\ell(k) = B_{\ell,{\rm NL}}(k) + c_\ell(k), \tag{28}$$

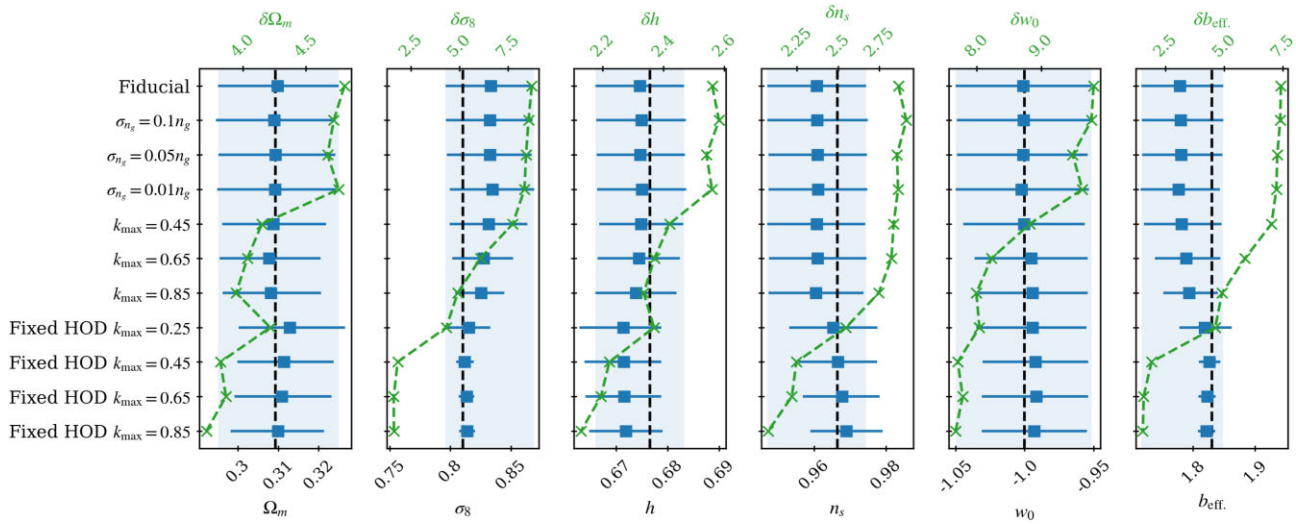


**Figure 13.** Per cent constraint and marginalized 1D posteriors for each of the cosmological parameters considered in the FS analyses of Section 4. The green crosses and green dashed line shows the per cent constrain and corresponds to the top $x$-axis. The blue points and error bars show the median and $1\sigma$ region of the marginalized posteriors and correspond to the bottom $x$-axis. The blue shaded region shows the width of our fiducial analysis, and the black vertical dashed line shows the location of the truth.

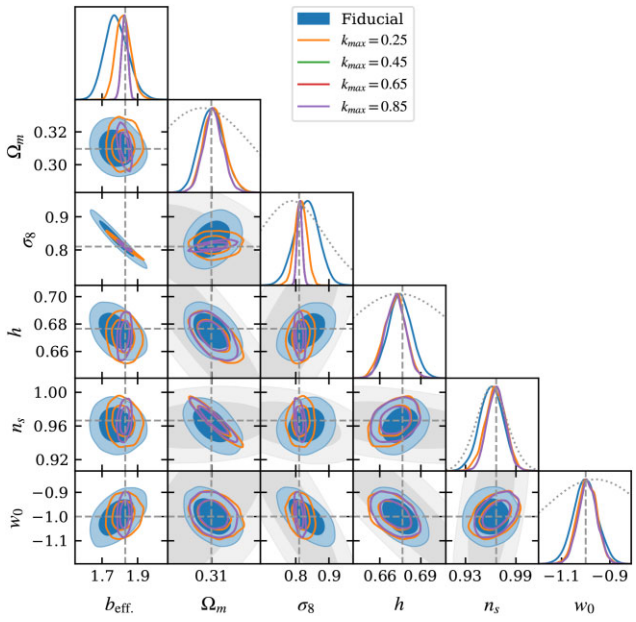


**Figure 14.** Similar to Fig. 11. The blue filled contours are the same as in Fig. 11. The empty contours show the results of the fixed HOD analyses described in Section 4.5, where only the five cosmological parameters shown are allowed to vary (all other model parameters are fixed to the truth).

where $B_{\ell,\,\rm NL}(k)$ would be the scale-dependent non-linear boost, and $c_\ell(k)$ includes corrections to the base model that accommodate the neglected effects that predominately impact small scales. It should be noted that in order to obtain the best possible prediction accuracy we would want to keep $B_\ell(k)$, and thus $c_\ell(k)$, as small as possible. Exactly what small-scale effects to include in the base model will depend on the given application of MATRYOSHKA and is beyond the scope of this work.

### 5.2 MATRYOSHKA PYTHON package

Alongside this paper we also publish the MATRYOSHKA PYTHON package.[3] This package includes all the weights for the NNs discussed in this paper, allowing them to be used without any requirement of retraining. The PYTHON package has been developed such that the component emulators can be used in isolation. For example the transfer function component emulator can be loaded with `matryoshka.emulator.Transfer()`, and predictions can then be made with the `.emu_predict()` method. The transfer function emulator makes predictions in ∼0.0004 s, and a non-linear galaxy power spectrum prediction can be made in ∼0.1 s (this is ∼3× faster than a transfer function prediction that can be made using CLASS with the accuracy settings implemented in NBODYKIT). It should be noted that although we have focused on training MATRYOSHKA to be used alongside the Aemulus simulations, it is simple to retrain any of the base model component emulators based on different parameters spaces. Functions to generate training samples and retrain the component emulators will be provided in the MATRYOSHKA PYTHON package.

Many of the HM functions in MATRYOSHKA are modified versions of those from the PYTHON package HALOMOD (Murray et al. 2021). In most cases these functions have been modified to allow MATRYOSHKA to make batch predictions more easily.

## 6 CONCLUSIONS

We have introduced MATRYOSHKA, a suite of NN-based emulators and PYTHON package, that aims to produce fast and accurate predictions for the non-linear galaxy power spectrum. The suite of emulators consists of a non-linear boost component emulator along with four base model component emulators, allowing us to rapidly produce linear predictions of the galaxy power spectrum to be combined with predictions of a non-linear boost.

[3] https://matryoshka-emu.readthedocs.io/en/latest/

In this paper, we have demonstrated how the base model component emulators can be trained to be used alongside a suite of numerical simulations, those being the Aemulus simulations. When trained on 6400 samples coming from the same 7D $w$CDM parameter space of the Aemulus simulations all base model component emulators have a MAPE $< 0.02$ per cent[4] at 68 per cent CI (for scale-dependent error of the base model component emulators, see Fig. 5). The component emulators are capable of producing very fast predictions. The transfer function component emulator is capable of producing predictions in $\sim$0.0004 s (the prediction time of the other component emulators is similar). This is $\sim 200\times$ faster than CLASS with the accuracy setting implemented via NBODYKIT. This speed up is more modest than that reported by other NN-based linear emulators such as Aricò et al. (2021) or Spurio Mancini et al. (2022). Although it is difficult to quantitatively compare the prediction speed of these emulators with MATRYOSHKA without using consistent hardware and accuracy settings for Boltzmann codes that the predictions are being compared to. It is however likely that these other linear emulators produce faster predictions because the use of ensembling to reduce generalization error in MATRYOSHKA does increase prediction time. Alternatives to ensembling will be explored in future works. The pre-trained transfer function emulator (and the other component emulators) is available in the MATRYOSHKA repository https://github.com/JDonaldM/Matryoshka

Using MATRYOSHKA we investigated the potential gain in constraining power by including small non-linear scales in a FS analysis of the galaxy power spectrum. To carry out this investigation we trained the non-linear boost component emulator with non-linear boosts calculated using Halofit. We approximate the scenario of using the Aemulus suite by only generating data for the 40 Aemulus training cosmologies and introducing simulation like noise into our training set. This non-linear boost component emulator returns predictions with a MAPE $< 0.25$ per cent at 68 per cent CI when evaluated with a test set produced for the Aemulus test samples (for scale-dependent error from the non-linear boost component emulator, see Fig. 7). Using MATRYOSHKA we conducted a series of FS analyses with different analysis set-ups on a mock power spectrum with underlying cosmology and HOD coming from *Planck* 2018 and White et al. (2011), respectively. From these analyses we estimate a $\sim 1.8\times$ improvement in the constraint on $\sigma_8$ by increasing $k_{\rm max}$ from 0.25 to 0.85 $h$ Mpc$^{-1}$. This increases to the $\sim 4.8\times$ improvement when the underlying HOD is known. These results highlight the potential gain in understanding we can achieve by using emulators in cosmological analyses, as well as motivating further studies of galaxy formation physics that will inform us about appropriate HOD parameters.

## ACKNOWLEDGEMENTS

KK is supported the UK STFC grant ST/S000550/1. He was also supported by the European Research Council under the European Union's Horizon 2020 Framework Programme (grant agreement no. 646702 'CosTesGrav'). FB has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Framework Programme (grant agreement no. 853291, 'FutureLSS'). FB is a Royal Society University Research Fellow. JD-M was supported by a STFC studentship.

[4]This scalar version of the MAPE is calculated by taking the average prediction error of the test set, and then averaging this across all scales predicted by the component emulators.

## DATA AVAILABILITY

All training, validation, and test data for the MATRYOSHKA base model component emulators are available in the MATRYOSHKA repository (validation data were not produced for the non-linear boost component emulator as such it is only available for the base model components). All weights for the NNs are also available in the repository. The Aemulus training and test samples can be found at https://github.com/zxzhai/emulator

## REFERENCES

Abadi M. et al., 2016, preprint (arXiv:1603.04467)
Agarwal S., Abdalla F. B., Feldman H. A., Lahav O., Thomas S. A., 2012, MNRAS, 424, 1409
Agarwal S., Abdalla F. B., Feldman H. A., Lahav O., Thomas S. A., 2014, MNRAS, 439, 2102
Alam S. et al., 2017, MNRAS, 470, 2617
Alam S. et al., 2021, Phys. Rev. D, 103, 083533
Alsing J. et al., 2020, ApJS, 249, 5
Angulo R. E., Pontzen A., 2016, MNRAS, 462, L1
Angulo R. E., Zennaro M., Contreras S., Aricò G., Pellejero-Ibañez M., Stücker J., 2021, MNRAS, 507, 5869
Aricò G., Angulo R. E., Hernández-Monteagudo C., Contreras S., Zennaro M., Pellejero-Ibañez M., Rosas-Guevara Y., 2020, MNRAS, 495, 4800
Aricò G., Angulo R. E., Zennaro M., 2021, preprint (arXiv:2104.14568)
Beutler F. et al., 2013, MNRAS, 429, 3604
Bird S., Rogers K. K., Peiris H. V., Verde L., Font-Ribera A., Pontzen A., 2019, J. Cosmol. Astropart. Phys., 02, 050
Chapman M. J. et al., 2021, preprint (arXiv:2106.14961)
Chuang C.-H. et al., 2019, MNRAS, 487, 48
Cole S. et al., 2005, MNRAS, 362, 505
Cooray A., Sheth R., 2002, Phys. Rep., 372, 1
Dawson K. S. et al., 2012, AJ, 145, 10
Debackere S. N. B., Schaye J., Hoekstra H., 2020, MNRAS, 492, 2285
DeRose J. et al., 2019, ApJ, 875, 69
DESI Collaboration et al., 2013, preprint (arXiv:1308.0847)
DESI Collaboration et al., 2016, preprint (arXiv:1611.00036)
Dolag K., Borgani S., Schindler S., Diaferio A., Bykov A. M., 2008, Space Sci. Rev., 134, 229
Duffy A. R., Schaye J., Kay S. T., Vecchia C. D., 2008, MNRAS, 390, L64
Eisenstein D. J., Hu W., 1998, ApJ, 496, 605
Euclid Collaboration et al., 2019, MNRAS, 484, 5509
Euclid Collaboration et al., 2021, MNRAS, 505, 2840
Foreman S., Perrier H., Senatore L., 2016, J. Cosmol. Astropart. Phys., 05, 027
Garrison L. H., Eisenstein D. J., Ferrer D., Tinker J. L., Pinto P. A., Weinberg D. H., 2018, ApJS, 236, 43
Giblin B., Cataneo M., Moews B., Heymans C., 2019, MNRAS, 490, 4826
Habib S., Heitmann K., Higdon D., Nakhleh C., Williams B., 2007, Phys. Rev. D, 76, 083503
Hamilton A. J. S., 2000, MNRAS, 312, 257
Hand N., Feng Y., Beutler F., Li Y., Modi C., Seljak U., Slepian Z., 2018, AJ, 156, 160
Heitmann K., Higdon D., White M., Habib S., Williams B. J., Lawrence E., Wagner C., 2009, ApJ, 705, 156
Ivanov M. M., Simonovic M., Zaldarriaga M., 2020, J. Cosmol. Astropart. Phys., 05, 042
Karamanis M., Beutler F., Peacock J. A., 2021, MNRAS, 508, 3589
Kingma D. P., Ba J., 2014, preprint (arXiv:1412.6980)
Klypin A., Prada F., Byun J., 2020, MNRAS, 496, 3862
Kobayashi Y., Nishimichi T., Takada M., Takahashi R., Osato K., 2020, Phys. Rev. D, 102, 063504
Kuhlen M., Vogelsberger M., Angulo R., 2012, Phys. Dark Universe, 1, 50
Kwan J., Heitmann K., Habib S., Padmanabhan N., Finkel H., Frontiere N., Pope A., 2015, ApJ, 810, 35

Lange J. U., Hearin A. P., Leauthaud A., van den Bosch F. C., Guo H., DeRose J., 2022, MNRAS, 509, 1779
Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
Lawrence E. et al., 2017, ApJ, 847, 50
Lesgourgues J., 2011, preprint (arXiv:1104.2932)
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
Miyatake H. et al., 2021, preprint (arXiv:2101.00113)
Mootoovaloo A., Jaffe A. H., Heavens A. F., Leclercq F., 2022, Astron. Comput., 38, 100508
Murray S. G., Power C., Robotham A. S. G., 2013, Astron. Comput., 3, 23
Murray S. G., Diemer B., Chen Z., Neuhold A. G., Schnapp M. A., Peruzzi T., Blevins D., Engelman T., 2021, Astron. Comput., 36, 100487
Navarro J. F., Frenk C. S., White S. D. M., 1996, ApJ, 462, 563( NFW)
Nishimichi T. et al., 2019, ApJ, 884, 29
Parejko J. K. et al., 2013, MNRAS, 429, 98
Pedersen C., Font-Ribera A., Rogers K. K., McDonald P., Peiris H. V., Pontzen A., Slosar A., 2021, J. Cosmol. Astropart. Phys., 05, 033
Percival W. J., Cole S., Eisenstein D. J., Nichol R. C., Peacock J. A., Pope A. C., Szalay A. S., 2007, MNRAS, 381, 1053
Philcox O. H. E., Ivanov M. M., Simonovic M., Zaldarriaga M., 2020, J. Cosmol. Astropart. Phys., 05, 032
Planck Collaboration VI, 2020, A&A, 641, A6( *Planck* 2018)
Schneider A. et al., 2016, J. Cosmol. Astropart. Phys., 04, 047
Schneider A., Teyssier R., Stadel J., Chisari N. E., Le Brun A. M. C., Amara A., Refregier A., 2019, J. Cosmol. Astropart. Phys., 03, 020
Senatore L., 2015, J. Cosmol. Astropart. Phys., 11, 007
Senatore L., Zaldarriaga M., 2015, J. Cosmol. Astropart. Phys., 02, 013
Simonovic M., Baldauf T., Zaldarriaga M., Carrasco J. J., Kollmeier J. A., 2018, J. Cosmol. Astropart. Phys., 04, 030
Spurio Mancini A., Piras D., Alsing J., Joachimi B., Hobson M. P., 2022, MNRAS, in press (arXiv:2106.03846)
Takahashi R., Sato M., Nishimichi T., Taruya A., Oguri M., 2012, ApJ, 761, 152
TensorFlow Developers, 2021, TensorFlow (v2.6.0). Zenodo(https://doi.org/10.5281/zenodo.5181671)
Tinker J. L., Kravtsov A. V., Klypin A., Abazajian K., Warren M. S., Yepes G., Gottlober S., Holz D. E., 2008, ApJ, 688, 709
Tinker J. L., Robertson B. E., Kravtsov A. V., Klypin A., Warren M. S., Yepes G., Gottlober S., 2010, ApJ, 724, 878
Villaescusa-Navarro F. et al., 2020, ApJS, 250, 2
Vogelsberger M., Marinacci F., Torrey P., Puchwein E., 2020, Nat. Rev. Phys., 2, 42
White M. et al., 2011, ApJ, 728, 126
Zhai Z. et al., 2017, ApJ, 848, 76
Zhai Z. et al., 2019, ApJ, 874, 95
Zheng Z. et al., 2005, ApJ, 633, 791
Zheng Z., Coil A. L., Zehavi I., 2007, ApJ, 667, 760
Zheng Z., Zehavi I., Eisenstein D. J., Weinberg D. H., Jing Y. P., 2009, ApJ, 707, 554
Zhou R. et al., 2021, MNRAS, 501, 3309

## APPENDIX A: MONITORING CONVERGENCE OF MCMC CHAINS

For this work we run our chains in two stages: a burn-in stage and a dense sampling stage. The burn-in stage uses the minimum number of walkers required by ZEUS (2× the number of dimensions). A small number of walkers are used here to allow for the walkers to move larger distances during each step of the chain, which results in a shorter burn-in period. Every 200 steps we estimate the integrated autocorrelation time (IAT). We consider the burn-in stage complete once the total number of steps exceeds 5× the IAT and our estimate of the IAT has changed by <1 per cent compared to the previous estimate. For the dense sampling stage we increase the number of walkers by a factor of 6, so for our fiducial analysis this increases the number of walkers from 16 to 108. We run this dense sampling stage for 10× the IAT estimate from the burn-in stage. The burn-in stage is discarded and the dense sampling stage is used for inference. Running the chains in this way allows us to exploit the rapid predictions from MATRYOSHKA whilst avoiding the long burn-in period that comes with using a large number of walkers with ensemble sampling.

## APPENDIX B: ACCURACY REQUIREMENTS

When developing an emulator we want to be able to produce predictions with errors that are smaller than the statistical errors of our observation, allowing us to return unbiased constraints on the parameters of interest within the statistical error. With this in mind, the statistical error of a given observation defines the accuracy requirement of the emulator predictions. For this work we have demonstrated training the non-linear boost component emulator with the goal of conducting a series of FS analyses of a CMASS-like power spectrum with a sample volume of 1 (Gpc $h^{-1}$)$^3$. Fig. B1 shows how the achieved level of prediction accuracy from MATRYOSHKA in this work compares to the statistical error from a CMASS-like sample with a volume of 1 (Gpc $h^{-1}$)$^3$ (solid blue line). We can see that for $0.01 \lesssim k \lesssim 1\ h\,\mathrm{Mpc}^{-1}$ the statistical error is larger than the prediction error, however if we double the volume of the sample (shown by the solid orange line), this is no longer the case. In Section 4, we show that the achieved level of prediction accuracy is sufficient to return unbiased cosmological parameters when analysing a CMASS-like power spectrum up to $k = 0.85\ h\ \mathrm{Mpc}^{-1}$ with a sample volume of 1 (Gpc $h^{-1}$)$^3$. This unlikely to be the case for a sample volume of 2 (Gpc $h^{-1}$)$^3$ as the emulator error is going to be larger than the statistical error on those scales as seen in Fig. B1.

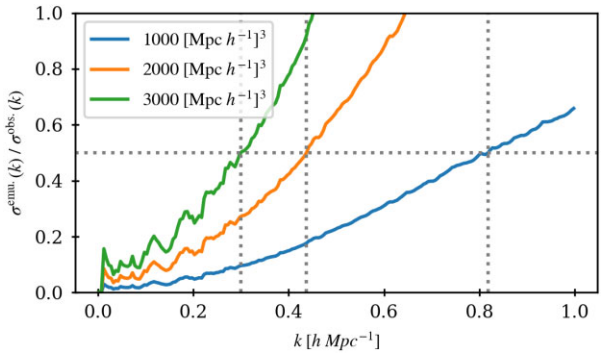


**Figure B1.** Ratio of the emulator error and statistical error for a CMASS-like sample with various sample volumes. The statistical errors are calculated using equation (18). The horizontal and vertical dotted grey lines indicate the scale at which the ratio of error equals 0.5.

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