Euclid preparation: IX. EuclidEmulator2 – Power spectrum emulation with massive neutrinos and self-consistent dark energy perturbations

Euclid Collaboration: M. Knabenhans1,2, J. Stadel1,2, D. Potter1,2, J. Dakin1,2, S. Hannestad2,3, T. Tram2,3, S. Marelli3, A. Schneider1,4,5, R. Teyssier1,6, S. Andreon5, N. Auricchio6, C. Baccigalupi7,8,9, A. Balaguera-Antolínez10,11, M. Baldi12,13,14, S. Bardelli6, P. Battaglia15, R. Bender16,17, A. Biviano9,18, C. Bodendorf17, E. Bozzo9, E. Branchini20,21,22, M. Brescia23, C. Burigana14,24,25, R. Cabanac26, S. Camera27,28,29, V. Capobianco29, A. Cappi6,30, C. Carbone31, J. Carretero32, C.S. Carvalho33, R. Casas34,35, S. Casas36, M. Castellano22, G. Castignani37, S. Cavuoti23,38,39, R. Cledassou40, C. Colodro-Conde11, G. Congedo41, C.J. Conselice42, L. Conversi43,44, Y. Copin45, L. Corcione29, J. Coupon19, H.M. Courtois46, A. Da Silva47,48, S. de la Torre49, D. Di Ferdinando14, C.A.J. Duncan50, X. Dupac44, G. Fabbian51, S. Farrens36, P.G. Ferreira50, F. Finelli12,52, M. Frailis9, E. Franceschi6, S. Galeotta9, B. Garilli31, C. Giocoli6,13,14, G. Gozaliasl53,54, J. Graciá-Carpio17, F. Grupp16,17, L. Guzzo5,55,56, W. Holmes57, F. Hormuth58, H. Israel16, K. Jahnke59, E. Keihanen54, S. Ker Micha60, C. Kirkpatrick54, B. Kubik61, M. Kunz62, H. Kurki-Suonio54, S. Ligori29, P. B. Lilje63, I. Lloro64, D. Maino31,55,56, O. Marggraf65, K. Markovic57, N. Martinet49, F. Marulli6,13,14, R. Massey66, N. Mauri13,14, S. Maurogordato30, E. Medinaceli15, M. Meneghetti6, B. Metcalf13,15, G. Meylan37, M. More sco6,13, B. Morin36, L. Moscardini6,13,14, E. Munari9, C. Neissner32, S.M. Niemi67, C. Padilla32, S. Paltani19, F. Pasian9, L. Patrizii14, V. Pettorino36, S. Pires36, G. Polenta68, M. Poncet40, F. Raison17, A. Renz69,70, J. Rhodes57, G. Riccio23, E. Romelli9, M. Roncarelli6,13, R. Saglia16,17, A.G. Sánchez17, D. Sapone71, P. Schneider65, V. Scottez72, A. Secroun60, S. Serrano34,35, C. Sirignano69,70, G. Sirri14, L. Stanc469, F. Sureau36, P. Tallada Crespi73, A.N. Taylor41, M. Tenti14, I. Tereno33,47, R. Toledo-Moreo74, F. Torradeflot73, L. Valenziano6,14, J. Valiviita53,54, T. Vassallo16, M. Viel7,8,9,18, Y. Wang75, N. Welikala81, L. Whittaker76,77, A. Zacchei9, E. Zucca6

Accepted XXX. Received YYY; in original form ZZZ.
2 Euclid Collaboration

ABSTRACT
We present a new, updated version of the EuclidEmulator (called EuclidEmulator2), a fast and accurate predictor for the nonlinear correction of the matter power spectrum. Percent-level accurate emulation is now supported in the eight-dimensional parameter space of $w_0w_a$CDM+$\sum m_\nu$ models between redshift $z = 0$ and $z = 3$ for spatial scales within the range $0.01 \ h \ Mpc^{-1} \leq k \leq 10 \ h \ Mpc^{-1}$. In order to achieve this level of accuracy, we have had to improve the quality of the underlying N-body simulations used as training data: (i) we use self-consistent linear evolution of non-dark matter species such as massive neutrinos, photons, dark energy and the metric field, (ii) we perform the simulations in the so-called N-body gauge, which allows one to interpret the results in the framework of general relativity, (iii) we run over 250 high-resolution simulations with $3000^3$ particles in boxes of $1(h^{-1} \ Gpc)^3$ volumes based on paired-and-fixed initial conditions and (iv) we provide a resolution correction that can be applied to emulated results as a post-processing step in order to drastically reduce systematic biases on small scales due to residual resolution effects in the simulations. We find that the inclusion of the dynamical dark energy parameter $w_a$ significantly increases the complexity and expense of creating the emulator. The high fidelity of EuclidEmulator2 is tested in various comparisons against N-body simulations as well as alternative fast predictors like HALOFIT, HMCode and CosmicEmu. A blind test is successfully performed against the Euclid Flagship v2.0 simulation. Nonlinear correction factors emulated with EuclidEmulator2 are accurate at the level of 1% or better for $0.01 \ h \ Mpc^{-1} \leq k \leq 10 \ h \ Mpc^{-1}$ and $z \leq 3$ compared to high-resolution dark matter only simulations. EuclidEmulator2 is publicly available at https://github.com/miknab/EuclidEmulator2.

Key words: cosmology: cosmological parameters – cosmology: large-scale structure of Universe – methods: numerical – methods: statistical

1 INTRODUCTION
Ongoing and forthcoming cosmological surveys such as DESI$^1$ (DESI Collaboration: Aghamousa et al. 2016), LSST$^2$ (LSST Science Collaboration: Abell et al. 2009), Euclid$^3$ (Laureijs et al. 2011), and WFIRST$^4$ (Akeson et al. 2019) have the potential to shed light on dark energy (DE), dark matter (DM), and the neutrino masses. It has been confirmed by Solar and atmospheric neutrino experiments that neutrinos have finite mass (e.g. Valle 2005; Schwetz et al. 2008), yet, attempts to pin down the total mass and the mass hierarchy of the neutrino flavour states have so far not been successful. The ongoing neutrino experiment KATRIN (Weinheimer 2002; Franken 2008; Wolf 2010) has been launched in order to tighten the neutrino mass bounds with particle physics. The equation of state parameter $w$ describing DE is also poorly constrained. While the current $\Lambda$CDM concordance cosmological model with a value of $w = -1$ is favoured, the error bars coming from Planck data alone are of order 50%. They shrink to ~ 10% if more DE sensitive probes such as cluster counts, weak lensing and supernovae type Ia are additionally considered (Planck Collaboration: Ade et al. 2015). Past and current surveys have a hard time constraining the DE parameter more accurately: Surveys like Planck (Planck Collaboration 2006) probe the cosmic microwave background (CMB) only, which is not very sensitive to DE. On the other hand, several surveys analysed either only spectroscopic probes (e.g. BOSS, Dawson et al. 2013) or only photometric probes (e.g. KiDS-450, Hildebrandt et al. 2017). In contrast, large-scale hybrid photometric and spectroscopic experiments like Euclid (Laureijs et al. 2011) will be able to reduce the error bars on the DE parameters, as the combination of weak lensing with galaxy clustering provides a promising handle on DE phenomena.

A common approach for the inference of cosmological parameters is to use Bayesian techniques. One specific possibility is to sample the likelihood function in a Markov chain Monte Carlo (MCMC) approach, compare the predicted observable (e.g. the power spectrum) to the one actually measured in an observation, and extract the maximum likelihood estimator values for the cosmological parameters. However, this requires a large number of accurate theoretical predictions. For studying the nonlinear regime of cosmic structure formation, N-body simulations, still the most accurate tool available today, are numerically too expensive to be used for Bayesian inference and hence there is a high demand for more efficient methods. While halo models for massive neutrinos are investigated by several research groups (see e.g. Massara et al. 2014 or Hannestad et al. 2020), surrogate models for N-body simulations, so-called emulators, have been shown to be very promising candidates. Several emulators are available already. Examples are FrankenEmu (Heitmann et al. 2009, 2010, 2014), CosmicEmu (Heitmann et al. 2016; Lawrence et al. 2017), the emulators of the Aemulus project (DeRose et al. 2019; McClintock et al. 2019; Zhai et al. 2019), ΝΓενΗαλοφιτ (Smith & Angulo 2019), EuclidEmulator1 (Euclid Collaboration: Knabenhans et al. 2019), the Dark quest emulator (Nishimichi et al. 2019) and BE-HaPPY (Valcini et al. 2019).

In this paper we will introduce an update of EuclidEmulator1. While EuclidEmulator1 was able to efficiently estimate the nonlinear correction (NLC) to the matter power spectrum for $w_0CDM$ cosmologies (with the time-variable DE equation of state parameter $w_a$ set to 0), EuclidEmulator2 can handle cosmologies with dynamical DE and massive neutrinos. In addition to a bigger parameter space, EuclidEmulator2 also pushes the upper limit of the $k$-range to $k_{\text{max}} \sim 10 \ h \ Mpc^{-1}$. The motivation for this is the same as described in Euclid Collaboration: Knabenhans et al. (2019): While clearly the dominant source of uncertainties on such small spatial scales is due to baryons, it is important to have best possible control over the dark-matter physics in this regime in order to avoid additional (and avoidable) uncertainties due to the dark sector. As we describe in Euclid Collaboration: Knabenhans et al. (2019), under certain assumptions it is possible to add baryonic and other corrections as a subsequent step in the pipeline, necessitating that theoretical precision is maintained in the high-$k$ regime of DM clustering.

---

* mischak@physik.uzh.ch
1 www.desi.lbl.gov/category/announcements/
2 www.lsst.org/lsst
3 sci.esa.int/euclid
4 wfirst.gsfc.nasa.gov
The authors follow this strategy and emulate the effect due to baryons on weak lensing observables on top of the underlying nonlinear DM physics.

This paper is structured as follows. In section 2, theoretical aspects regarding massive neutrinos, dynamical DE and the N-body gauge are discussed. In section 3 we introduce our approach to include massive neutrinos in N-body simulations using PKDGRAV3. In section 4 we report on an extensive convergence series that we have performed in order to estimate the uncertainties in the input simulations used for the construction of EuclidEmulator2, considering volume and resolution effects. We will find systematics in this section whose treatment will then be discussed subsequently in section 5. In section 6, we investigate a prototype emulator based on HALOFIT data (as we have already done in Euclid Collaboration: Kranbein et al. 2019). In section 7, insights taken from this prototype emulator are used to construct the fully simulation-based EuclidEmulator2, whose performance is ultimately assessed in section 8. We then gain some insight into parameter degeneracies at the level of the matter power spectrum within the w0wa cold dark matter (CDM)+∑mi models. Our conclusions are found in section 10.

2 THEORETICAL BACKGROUND

2.1 Massive neutrinos

In oscillation experiments studying solar and terrestrial neutrinos, compelling evidence has been found that the three flavor eigenstates of neutrinos (e, µ and τ) can be mixed (Becker-Szendy et al. 1992; Fukuda et al. 1998a,b; Ahmed et al. 2004; DUNE Collaboration: Acquistarci et al. 2015). This implies that neutrinos must have finite mass eigenstates, a fact lying outside of the current standard model of particle physics. While those experiments were able to measure the differences between the squared neutrinos masses, they cannot measure the absolute mass scale. Currently, we only have bounds on the sum of the neutrino masses, see e.g. Particle Data Group: Patrignani et al. (2016). On cosmological scales, light neutrinos are very abundant and hence are expected to have a significant effect on large-scale structure. As we are considering neutrinos of masses smaller than 1 eV, they have become non-relativistic only after the electron-nucleon recombination and hence they have imprinted only a small signal into the cosmic microwave background (CMB). However, neutrinos constitute a fraction of the DM in our Universe and hence the DM power spectrum is the key quantity to look at when trying to constrain light neutrino masses.

For the rest of this paper we are restricting ourselves to three light mass eigenstates, i.e. we neglect the possibility of heavy sterile neutrinos. Additionally, we consider only the case of three degenerate neutrino masses, i.e. we neglect that the measured squared differences for their masses is given by (Particle Data Group: Tanabashi et al. 2018)

\[
\delta m^2_{21} = m_2^2 - m_1^2 = (7.37^{+0.197}_{-0.146}) \times 10^{-5} \text{eV}^2, \quad (1)
\]

\[
\delta m^2_{31} = |m_3^2 - m_1^2| = (2.56^{+0.043}_{-0.037}) \times 10^{-3} \text{eV}^2, \quad (2)
\]

or alternatively

\[
\delta m^2_{32} = |m_3^2 - m_2^2| = (2.54 \pm 0.04) \times 10^{-3} \text{eV}^2, \quad (3)
\]

depending on the considered neutrino hierarchy. This simplification is justified as the difference in the resulting power spectra between a degenerate and a realistic (normal or inverted) mass hierarchy is expected to be well sub-per cent. Neutrinos have a non-vanishing mass while at the same time they have a high velocity dispersion, hence they do not belong to the category of CDM. In this paper we focus on cosmologies where there are non-zero contributions from both cold and non-cold DM particles. Such a CDM + massive neutrino (CDM+∑mi) cosmology also serves as the new Euclid reference cosmology (see Table 2), in contrast to a more standard pure CDM cosmology.

While neutrinos are still relativistic, their free-streaming length is of the size of the Hubble scale. Only after the transition to the non-relativistic phase, the comoving free-streaming scale of neutrinos starts to shrink. As a result, neutrino perturbations get washed out on scales below the free-streaming scale. Due to gravitational interaction, this also suppresses the clustering of DM on small enough scales. The strength of the effect for a specific k-mode depends on both redshift and mass of the neutrinos. But even for small (but finite) neutrino masses we expect a several percent suppression signal in the DM power spectrum due to these effects. The main effect is on the background expansion of the Universe: If only massless neutrinos are considered, ΩDM is identical to ΩCDM. However, in order to maintain spatial flatness, even when a non-vanishing Ωv is present, ΩDM remains unaltered and thus this requires ΩCDM to be decreased accordingly:

\[
\Omega = \Omega_{\text{DE}} + \Omega_{\text{DM}} + \Omega_b + \Omega_{\text{rad}} = \Omega_{\text{DE}} + (\Omega_{\text{CDM}} + \Omega_v) + \Omega_b + \Omega_{\text{rad}}. \quad (4)
\]

The DM density is thus decreased by Ωv (while the DM density parameter does not change). As CDM and neutrinos evolve differently over the history of the Universe, this leads to a suppression of the CDM+baryon power spectrum. This power suppression can help us constrain the sum of the neutrino masses (see e.g. Ichiki et al. 2009; Coulton et al. 2019; Copeland et al. 2020).

Neutrinos at very high redshifts (z ≥ 1000) are still a relativistic species, and hence would primarily contribute to Ωrad. However, our simulations focus on the nonlinear growth of structure at (z ≤ 10), at which time the contribution of massive neutrinos shifts to ΩDM. It is for this reason that we consider Ωv as a contribution to ΩDM in the above. Nonetheless, in our N-body simulations the transition between fully relativistic and non-relativistic neutrinos, including the full distribution function at any given redshift, is accounted for by the CLASS Boltzmann solver. All neutrino effects are self-consistently included in our simulations at the linear level, such that this assignment of Ωv to ΩDM is purely a convenient choice of parameterisation.

2.2 Dynamical dark energy

In the standard ΛCDM cosmology, DE is assumed to be a cosmological constant with a time-independent equation of state parameter given by w ≡ −1. This implies that ρDE(t) = const which can be seen from the following conservation equation:

\[
T^\nu_0\mu = \dot{\rho}_\Lambda + 3H(\rho_\Lambda + p) = 0, \quad (5)
\]

\[
\rho_\Lambda + 3Hp_\Lambda(1 + w) = 0, \quad (6)
\]

\[
\dot{\rho}_\Lambda = 0, \quad (7)
\]

where Tμν is the energy momentum tensor, p the pressure, ρ the density and the over-dot denotes a derivative w.r.t. cosmic time. Although this value is close to the best fitting value from supernova surveys (Betoule et al. 2014), DE with a slightly time-dependent equation of state parameter is not ruled out by the data currently

5 In this paper we use the notation CDM+∑mi in order to avoid confusion with mixed DM (MDM) models with significant contributions from more exotic warm or even hot DM species.
available. The effects of DE perturbations become relevant only on the largest scales (usually at $k \lesssim 0.1 \text{ h Mpc}^{-1}$) as has recently been studied carefully in Dakin et al. (2019b). Nevertheless, the effects of DE on the matter power spectrum can become quite significant, primarily because changes in the DE component have an impact on the scale factor $a(t)$ which in turn affects the power spectrum on all scales.

Here we shall just briefly recap the key aspects of the theory of time-dependent DE relevant in the context of EuclidEmulator2. We shall follow closely the explanations given in Dakin et al. (2019b) where this topic has been reviewed in greater detail. There are two popular ways of describing DE in the setting of an effective theory: the fluid description and the so-called parametrized post-Friedmann (PPF) description (Hu & Sawicki 2007). In the fluid description, DE is considered a fluid with an equation-of-state parameter $w$ (a being the cosmic scale factor) and a constant rest-frame sound-speed $c_s$. We will adopt the widely used parameterisation $w(a) = w_0 + w_1 (1 - a)$. As can be seen in Equation (2.9) in Dakin et al. (2019b), the Euler equation for a DE fluid features a factor $(1 + w)^{-1}$, leading to divergences for cosmologies with a DE equation of state (EoS) parameter that evolves across $1 + w = 0$ over time. This is a manifestation of the fact that such a DE is gravitationally unstable for the lack of additional internal degrees of freedom (Fang et al. 2008). The case $w = -1.0$ is sometimes referred to as the “phantom divide” or “phantom barrier” and models crossing it are called “phantom-crossing” cosmologies. It follows, unfortunately, that the fluid description of DE is not well-suited to describe phantom-crossing cosmologies and yet there is no reason why these models should not be taken into account in our analysis.

As described in Fang et al. (2008), common approaches to deal with this problem are either to ignore the DE perturbations in an ad hoc manner or to simply turn them off which violates the energy-momentum conservation for non-$\Lambda$ models and leads to inconsistencies among the Einstein equations. For dynamical DE models it is hence crucial to thoroughly address the issue of phantom-crossing.

The parameterisation of minimally coupled DE is not a complete system of equations but rather it requires two closure conditions. The relation between density and pressure fluctuations of DE, giving rise to its EoS parameter and sound speed, serves as one of these closure relations in the fluid description. The PPF formalism on the other hand takes a direct relation between the momentum density of DE and that of DM on large scales to close the system of equations (Fang et al. 2008), thus circumventing the divergence of sound speed at phantom-crossing. This describes the DE momentum perturbations on large scales. To describe them on small scales, an effective sound speed $c_\Gamma$ is introduced which is related to the scale below which the DE field becomes sufficiently homogeneous compared to the matter field. From an interpolation between these two limits one can obtain an evolution equation for the potential of DE in its rest frame:

$$\partial_t \Gamma = \frac{\partial_t a}{a} \left[ S \left( 1 + \frac{c_s^2 k^2}{\mathcal{H}^2} \right) - \Gamma \left( 1 + \frac{c_\Gamma^2 k^2}{\mathcal{H}^2} \right) \right],$$

(8)

with $\mathcal{H}$ the conformal Hubble parameter and $\partial_t$ denoting the derivative w.r.t. conformal time $\tau$. $\Gamma$ is related to the rest-frame DE density field $\delta_{DE}^{\text{rest}}$ via the Poisson equation

$$k^2 \Gamma = -4\pi G a^2 \delta_{DE}^{\text{rest}},$$

(9)

$c_\Gamma$ being the effective sound speed and $S$ being defined as

$$S = \frac{4\pi G a^2}{\mathcal{H}} (\rho_{DE} + p_{DE}) \frac{\theta_N^N}{k^2}.$$

(10)

In the last equation, in turn, $\theta_N^N$ denotes the velocity divergence field of all other species than DE in the conformal Newtonian gauge. For a more complete discussion we refer the reader to Dakin et al. (2019b), section 2.1.2.

2.3 General relativity in N-body simulations: The N-body gauge

Traditionally, the formation of large-scale structure is simulated with Newtonian N-body codes. There are two ways to bring the general theory of relativity (GR) into these simulations, in order to study its effects on structure formation. The first option is to replace the Newtonian equations of motion of structure formation by their general relativistic counterpart. However, this is not an easy task, in particular because scientists have optimised N-body codes for Newtonian physics over the last decades. The second option is hence to still use these optimised Newtonian N-body simulation codes which is allowed under certain circumstances. As has been shown in Chisari & Zaldarriaga (2011), Newtonian simulations predict the clustering properties of DM and galaxies with negligible errors even on very large scales if non-relativistic components or relativistic but non-clustering components are modelled and if a proper set of coordinates (i.e. gauge) is chosen. According to Fidler et al. (2016, 2017) it suffices to add a relativistic correction to the Newtonian potential $\phi$ in the Euler and the Poisson equation (for more practical detail see e.g. Dakin et al. 2019b):

$$(\partial_t + \mathcal{H}) \phi_{\text{CDM}+b} = -\nabla \phi + \nabla \phi_{\text{DE}},$$

$$\nabla^2 \phi = 4\pi G a^2 \delta_{\text{tot}},$$

(11)

(12)

where $\tau$ is the conformal time and $\delta_{\text{tot}}$ is the total density perturbation from all species,

$$\delta_{\text{tot}} = \delta \rho_{\text{CDM}} + \delta \rho_b + \delta \rho_{\text{photon}} + \delta \rho_{\nu} + \delta \rho_{DE},$$

(13)

where we note that $\delta \rho_{DE} = 0$ for $w \neq -1$. The GR correction potential $\gamma^{\text{Nb}}$ is built from any other gravitating quantity not already accounted for by $\delta \rho_{\text{tot}}$, i.e. the momentum density, pressure and shear of photons, neutrinos and DE. We further parametrise this as

$$\nabla^2 \gamma^{\text{Nb}} = -4\pi G a^2 \delta \rho_{\text{metric}} \gamma^{\text{Nb}},$$

(14)

where $\delta \rho_{\text{metric}}$ is a fictitious density perturbation, the Newtonian gravity of which implements all general relativistic effects of $\nabla^2 \gamma^{\text{Nb}}$.

Finally, the full effective potential is split into $\phi - \gamma^{\text{Nb}} \equiv \phi_{\text{sim}} + \phi_{\text{lin}}$, with $\nabla^2 \phi_{\text{sim}}$ formally equal to $4\pi G a^2 (\delta \rho_{\text{CDM}}^{\text{DE}} + \delta \rho_{\text{DE}}^{\text{DE}})$ but in the simulation computed through usual N-body (particle) techniques, while

$$\nabla^2 \phi_{\text{lin}} \equiv 4\pi G a^2 \left( \delta \rho_{\text{DE}}^{\text{DE}} + \delta \rho_{\text{DE}}^{\text{DE}} + \delta \rho_{DE}^{\text{DE}} + \delta \rho_{DE}^{\text{DE}} \right) \rho_{DE},$$

(15)

is solved on a grid using Fourier techniques and then applied to the particles as a correction force to the main particle gravity. Notice that $\phi_{\text{lin}}$ is the object called $\phi_{\text{GR}}$ in other publications such as e.g. Tram et al. (2019) or Dakin et al. (2019b,a).

Note that the continuity equation is formally not affected by this additional GR potential $\gamma^{\text{Nb}}$. This then can be considered writing the full general relativistic evolution equations of the N-body simulations in a very special gauge, the so-called $N$-body gauge (indicated by the superscript "Nb"). Doing so, it is possible to apply a gauge transformation of the output in a post-processing step in order to re-obtain general relativistic results in any observationally relevant gauge up to first order.
3 COSMOLOGICAL SIMULATIONS

Cosmic emulators are based on training data (also known as the experimental design) generated by N-body simulations. Unsurprisingly, the quality of any emulator hence crucially depends on the quality of the training data. We invested great efforts into optimising the quality of the training set simulations. The construction of EuclidEmulator2 training data relies mainly on three codes: PKDGRAV3, which is the main simulation code, and CLASS which in turn depends on the Einstein-Boltzmann solver CLASS. All three codes had to be optimised in order to be able to fully self-consistently treat cosmologies with massive neutrinos and dynamical DE. In this section we shall describe how these codes were optimised to generate a suite of simulations used as the experimental design for EuclidEmulator2 and thereby put emphasis on the differences with respect to the generation of the experimental design of EuclidEmulator1.

3.1 Pipeline overview

We shall give a quick overview over the implemented pipeline in order to facilitate the understanding of the steps involved in generating the training data for the emulator. More detail is given for each step in dedicated sections below.

Pre-computation of cosmological quantities: We have designed a pipeline in which all cosmological background and linear quantities, such as the time dependence of the Hubble parameter $H(z)$, are computed before running any simulation. This is done through CLASS in order to take all relevant physics into account (e.g. this approach allows for a fully relativistic treatment). In this step, transfer functions are computed to linear level at many different redshifts for all species contributing to $\delta_{\text{lin}}$, that are later used to provide the small corrective force contributions in the N-body simulations. In the case of EuclidEmulator1, however, we only computed the transfer function at $z = 0$, which we later used for initial condition (IC) generation via the usual back-scaling approach, and used an analytical form for $H(z)$ to compute the background evolution.

Gauge transformation: This step was entirely missing in EuclidEmulator1 where we used the transfer function computed in synchronous gauge to set up the IC's. CLASS, as well as CAMB, compute quantities either in the synchronous or in the conformal Newtonian gauge. However, in order to make Newtonian N-body codes consistent with GR, their input transfer functions have to be mapped to the N-body gauge described in section 2.3. This transformation is performed by the CONCEPT code. The results are then stored into look-up tables inside HDF5 files that can be queried by the N-body code.

N-body simulation: Both the initial condition generation and the actual N-body simulations are performed with PKDGRAV3. Paired-and-fixed (P+F) (Angulo & Pontzen 2016) first order Lagrangian perturbation theory (1LPT) initial conditions are set up at redshift $z = 99$ (see also section 4 for a more in-depth discussion). The nonlinear evolution of DM particles is computed with a binary tree-based fast multi-pole method (FMM). To these tree forces we add a particle-mesh (PM) field for the fluctuations due to massive neutrinos, photons, DE and the metric field in order to study their effect on cosmological structure growth at linear level (for simulations treating neutrinos fully nonlinearly see e.g. Banerjee et al. 2018; Bird et al. 2018). This aspect is one of the key differences between EuclidEmulator1 and EuclidEmulator2.

Post-processing: From the simulations we obtain the power spectra for each realisation of the fixed-IC simulation pair. In order to get the final P+P power spectrum we compute the average of those individual realisations. As ultimately we are primarily interested in the NLC, we compute it w.r.t. the linear theory power spectrum (in our case computed by CLASS) at each redshift. Explicitly, the NLC $B(k, z)$ is the dimension-less quantity defined through the relation

$$P_{\text{nl}}(k, z) = P_{\text{LinTh}}(k, z)B(k, z),$$

where the sub-script nl stands for “nonlinear” and the sub-script LinTh for “linear theory”. Notice that, in contrast, in Euclid Collaboration: Knabenhans et al. (2019) the NLC was defined slightly differently: we computed it via division by the IC power spectrum of the simulation. We also compute an additional factor that corrects for power suppression at small scales caused by resolution effects in the simulations. This factor we shall refer to as the resolution correction factor (RCF) and it will be discussed in section 5.

Emulator construction: The obtained data matrix is then principal component analysed. The emulator then predicts the vector of principal component weights.

3.2 Pre-processing with CONCEPT: Transfer functions

For generation of particle initial conditions inside PKDGRAV3, matter density and velocity transfer functions $\delta_{\text{CDM+baryon}}(a_{\text{ini}}, k)$ and $\theta_{\text{CDM+baryon}}(a_{\text{ini}}, k)$ are required, where $a_{\text{ini}}$ is the scale factor at the start of the simulation. As we seek to carry out the simulation in N-body gauge, this is also the gauge of these transfer functions. Note that, due to the contributions from species other than CDM, we cannot obtain accurate particle velocities from only $\delta_{\text{CDM+baryon}}$.

To keep the simulation in N-body gauge, we need to repeatedly apply the linear GR correction force $-\nabla \phi_{\text{lin}}$. From (15) we see that we are thus in need of $\delta_{\text{photons}}(a, k)$, $\delta_{\text{B}}(a, k)$, $\delta_{\text{DE}}(a, k)$ and $\delta_{\text{metric}}(a, k)$. Obtaining these N-body gauge transfer functions, the CONCEPT code has been run in advance. This is a full N-body code in its own right, with the added trait of very tight integration with the CLASS code. Internally, CONCEPT takes the synchronous gauge output from CLASS, converts it to N-body gauge and uses it for both particle initial conditions and GR corrections. The relevant gauge transformations are

$$\delta_{\text{lin}} = \delta_{\text{lin}} + 3H(1 + w_\alpha)\tilde{\rho}_\alpha \theta_{\text{tot}}^\alpha \frac{k^2}{2},$$

$$\delta_{\text{metric}} = \theta_{\text{tot}} \left( \frac{\tilde{\rho}_\alpha}{2} - 3H\theta_{\text{tot}}^\alpha \frac{k^2}{2} \right),$$

where superscripts ‘Nb’ and ‘s’ indicate the N-body and synchronous gauge respectively. $\alpha$ labels the species, $\tilde{\rho}_\alpha$ is the trace of the metric perturbation in synchronous gauge and $\theta_{\text{tot}}$ is the total velocity divergence of all species.

Though more complicated, the $\theta_{\text{metric}}(a, k)$ transfer function is similarly constructed from various CLASS outputs, some of which are only available in the CLASS version that ships with CONCEPT (see Dakin et al. 2019a for more details). We also note that the computation of the DE pressure perturbation within CLASS, needed for $\delta_{\text{DE}}$, is wrong in the standard version of CLASS, but fixed in the version shipping with CONCEPT (see Dakin et al. 2019b for details).

For use with PKDGRAV3, the so-called CLASS-utility was added to CONCEPT, which saves the N-body gauge transfer functions to an external HDF5 file, which is then read in by PKDGRAV3. In the HDF5 all requested transfer functions (i.e. on top of the CDM+baryon
3.3 Initial condition generation

In order to get the initial conditions for the particle positions, we generate a regular grid of particles which we displace from their grid points using first-order Lagrangian perturbation theory. We imprint an initial power spectrum $P_{\text{ini}}$ with the pairing-and-fixing technique (Angulo & Pontzen 2016), i.e.

$$P(\delta_{i,\text{ini}}, \theta_i) = \frac{1}{2\pi} \delta_D(\delta_{i,\text{ini}} - \sqrt{P_{\text{ini}}}) ,$$

(19)

where $\delta_D$ being the Dirac delta function and the index $i$ labelling the Fourier modes (for more info about how we use the pairing-and-fixing technique to generate initial conditions, please refer to Euclid Collaboration: Knabenhsen et al. 2019). The initial power spectrum is computed based on the CDM+baryon overdensity field $\delta_{\text{CDM+baryon}}$ pre-computed by CLASS+CDM/CEPT at $z_{\text{ini}} = 99$. While PKDGRAV3 has no further use for $\delta_{\text{DM}}(z < z_{\text{ini}})$ we still tabulate these, as they are used when computing the NLC factors. We then use

$$P_{\text{m. ini}}(k, z_{\text{ini}}) = \zeta^2(k) \delta_{\text{m}}^2(k, z_{\text{ini}}) ,$$

(20)

where the $\zeta$-function is defined as

$$\zeta(k) = \pi \sqrt{2 A_0 k^{-3/2}} \left[ \frac{k}{k_s} \right]^{n_s - 1} \exp \left[ \frac{\alpha_s}{4} \ln \left( \frac{k}{k_s} \right)^2 \right] .$$

(21)

Here, $A_0$ is the spectral amplitude, $n_s$ the spectral index, and the running $\alpha_s = 0$ in all cases. The pivot scale is set to its standard Planck value, $k_s = 0.05$ Mpc$^{-1}$. The resulting initial power spectrum is used to displace the particles from their regular grid points.

3.4 Nonlinear evolution

Once the initial condition is generated, the DM particles are evolved by the tree code PKDGRAV3 using FMM and a multi-timestepping approach. For further technical details about the gravity evolution of DM in PKDGRAV3, we refer to Potter et al. (2017).

What is newly introduced in the version of PKDGRAV3 that is used for this work (which is also new compared to the version used to construct EuclidEmulator1) is the interaction with massive neutrinos as well as other linearly evolved species, namely photons, DE, and the metric field. For this, PKDGRAV3 fetches the linear evolution of all these linearly evolved species from the pre-computed CLASS transfer functions at every base time step. Via the associated power spectrum, the transfer function can be converted into a density field that is realised on a grid which ultimately leads to a weak corrective mesh force. This particle-mesh interaction provides an additional gravity source term to the particle-particle interaction such that the DM field is evolved taking the linear species into account.

During the early Universe, it is important to capture the effects of the high-frequency oscillations in the linear fields, particularly in the metric field, otherwise we would see a slight offset of the power spectrum at linear scales. The linear evolution may only be done when the simulation is "time synchronised" which happens at the start of a base time step. One approach would be to take sufficient base time steps to capture this effect at high redshift, but it would be computationally wasteful at lower redshifts. Instead we adopted the following scheme. Each N-body simulation was started at redshift $z_{\text{ini}} = 99$ and evolved in 60 time steps to $z_{\text{fin}} = 10$. From $z_{\text{intra}}$ we continued each simulation down to $z_{\text{ini}} = 0$ in another 100 time steps resulting in a total number of base time steps $n_b = 160$. By employing this approach we achieve agreement between the linear evolution of all particle species and CLASS over the entire simulation starting from $z_{\text{ini}}$ all the way to $z = 0$.

4 CONVERGENCE TESTS

We have performed extensive convergence tests for the power spectrum and the NLC in different dimensions. These convergence test results serve to update those presented in Euclid Collaboration: Knabenhsen et al. 2019).

4.1 Volume

We start by determining the minimal volume required to achieve convergence at the 1% level. We compare a series of box size of edge lengths $L \in \{512, 1024, 2048, 4096\}$ h$^{-1}$ Mpc to a reference volume $V = L^3 = (8192 h^{-1}$ Mpc$)^3$. In this process we fix the resolution parameter to $\ell^{-1} \equiv N/L = 1/3 h$ Mpc$^{-1}$.

As our new pipeline (described in section 3) allows us to recover the linear scales very accurately at all redshifts, we could try to actually perform the convergence comparison directly against linear theory itself. However, as for simulation box sizes $L < 2048 h^{-1}$ Mpc even the smallest $k$ modes are already slightly nonlinear, the finite volume effects overlap with the pre-virialisation dip such that particularly at low redshifts no clear conclusion can be drawn.

For this reason, we performed additionally the volume convergence test based on the NLC factor (i.e. we compare to the NLC factor of the simulation in the $L = 8192 h^{-1}$ Mpc box). The result is shown in Fig. 2. In this test the pre-virialisation dip is cancelled out such

\[\text{Notice that in this paper we use } N \text{ for the number of particles per dimension.}\]
that we are only left with the finite volume effects. We find that a simulation box of \( L = 1024 \, \text{h}^{-1} \, \text{Mpc} \) is converged at the level of \( \sim 1\% \) (the cosmic variance of only a few individual mildly nonlinear \( k \) modes exceeds the 1\% limit). Notice that this result is unchanged if one would perform the comparison to the largest box at the power spectrum level. As the NLC is computed with respect to linear theory, dividing two NLCs by each other leads to cancellation of the linear theory out of the expression such that one is left with a ratio of power spectra only:

\[
\frac{B_L}{B_{L=8192}} = \frac{P_{L,\text{LinTh}}}{P_{L=8192}^{\text{LinTh}}} = \frac{P_{L}}{P_{L=8192}}.
\]

Based on this result we choose the simulation box side length for our simulations to be \( L = 1 \, \text{h}^{-1} \, \text{Gpc} \).

### 4.2 Resolution

For the investigation of the mass resolution required to get convergence of the power spectrum at a desired level, we are interested in the minimal value of the resolution parameter \( \ell^{-1} \) of N-body simulations. In Euclid Collaboration: Knabenhans et al. (2019) we claimed that the power spectra are converged at the level of 1\% up to \( k \sim 5 \, \text{h} \, \text{Mpc}^{-1} \) for \( \ell^{-1} \geq 1.6 \, \text{h} \, \text{Mpc}^{-1} \). We state clearly that this statement was overly optimistic. We underestimated the minimal \( \ell^{-1} \) because we compared to a simulation with \( \ell^{-1} = 4 \, \text{h} \, \text{Mpc}^{-1} \) which at that time was the best we could do. For the present work we were able to double the resolution parameter of our reference simulation to \( \ell^{-1} = 8 \, \text{h} \, \text{Mpc}^{-1} \). In turn, this increases our current estimate of the minimal \( \ell^{-1} \) value required to achieve convergence at the 1\% level.

From Fig. 3 one clearly observes that simulations with \( \ell^{-1} \geq 4 \, \text{h} \, \text{Mpc}^{-1} \) are required to be converged at the 1\% level at \( k = 10 \, \text{h} \, \text{Mpc}^{-1} \) with respect to the \( \ell^{-1} = 8 \, \text{h} \, \text{Mpc}^{-1} \) simulation at \( z = 0 \) and the resolution needs to be even higher if 1\%-convergence at higher redshifts is required. Given the minimal volume found in section 4.1, such high \( \ell^{-1} \) values imply prohibitively large particle numbers for our simulations. Taking our computational budget into account, we decided to run simulations with \( \ell^{-1} = 3 \, \text{h} \, \text{Mpc}^{-1} \) thereby doubling the resolution parameter compared to EuclidEmulator1. This means that our simulations, according to the currently best estimate available, are converged at redshift \( z = 0 \) up to \( k_{2\%} = 9.42 \, \text{h} \, \text{Mpc}^{-1} \) at the level of 2\% and up to \( k_{1\%} = 5 \, \text{h} \, \text{Mpc}^{-1} \) at the level of 1\%. Further results showing how the values for \( k_{1\%} \) and \( k_{2\%} \) evolve with redshift are shown in Table 1.

Notice that the suppression of power due to low mass resolutions is very systematic. We investigate this further in section 5.

Further we summarise that a compromise between the requirements estimated from the convergence tests and our computational budget leads to the following specifications for our N-body simulations. We evolve 3000\(^3\) particles in boxes with a volume of \( 1 \, \text{h}^{-3} \, \text{Gpc}^3 \).

### 4.3 Paired-and-fixed vs. Gaussian random field initial conditions

We also re-evaluate the comparison between \( \text{P+F} \) simulations and simulations based on Gaussian random field (GRF) initial conditions. For this comparison we ran one pair of fixed amplitude simulations and an ensemble of 50 different GRF simulations. Over all redshifts of interest (\( z \leq 3 \)), \( \text{P+F} \) simulations approximate the GRF ensemble average very well on all but the largest scales. However, on all scales the deviation of the \( \text{P+F} \) power spectrum w.r.t. the GRF ensemble average lies well within the bound set by \( \max (\sigma_{\text{P+F}}, 1\%) \). This confirms the findings of Angulo & Pontzen (2016). It needs to be taken into account, though, that in that publication an ensemble of 300 GRF simulations (i.e. six times larger than our ensemble) has been used for comparison. Based on this exploration and the results found already in Euclid Collaboration: Knabenhans et al. (2019) we again use the pairing-and-fixing approach to efficiently reduce cosmic variance.

### 4.4 Quantities not investigated in this convergence series

Quantities we have not tested in this convergence series are the starting redshift, the main time-stepping parameter and the resolution of the PM-grid for the linear species. As we describe below, the softening is related to the mass resolution but its convergence was not tested independently either. We set the softening parameter to default values of \( \text{PKDGRAV3:} \, \epsilon = 1/(50 \, \text{N}) \) and the time-stepping parameter \( \eta = 0.20 \). The number of PM-grid cells for the linear species has been conservatively chosen to be a quarter of the CDM+ b particle number \( N \) per dimension, i.e. we use \( N_{\text{lin}} = 750 \) PM-grid cells per dimension. The ratio \( N/N_{\text{lin}} = 4 \) has already been used in the generation of the Euclid Flagship v2.0 Simulation (EFS2) simulation, where it has been proved to be a more than adequate choice. We also follow the EFS2 simulation for the choice of the initial redshift, \( z_{\text{ini}} = 99 \). Convergence of these quantities has been studied in Schneider et al. (2016). Further, we have not investigated second order Lagrangian perturbation theory (2LPT) ICs for this work but use 1LPT ICs to set up our simulations. While 2LPT is expected to improve the resolution convergence results presented above, \( \text{PKDGRAV3} \) does not yet support 2LPT for multiple fluids.

### 4.5 Results of this convergence analysis

We have identified that one requires simulations with 3000\(^3\) particles and a resolution parameter of \( \ell^{-1} = 3 \, \text{h} \, \text{Mpc}^{-1} \) (corresponding to a minimally resolved mass of \( \sim 3.3 \times 10^9 \, \text{M}_\odot/h \)) and a Nyquist frequency of \( k_{\text{max}} \approx 9.4 \, \text{h} \, \text{Mpc}^{-1} \) in order to achieve satisfactory accuracy on both large and small scales. A simulation of this size and resolution takes a bit more than 2000 node hours (on GPU accelerated nodes\(^7\)).

### 5 RESOLUTION CORRECTION

In Fig. 3 in section 4.2 we show that too low a mass resolution leads to a suppression of power on small scales where the amplitude of this suppression grows both with a growing ratio \( \ell^{-1}/L_{\text{lowRes}} \) and as the redshift \( z \) increases. However, as this effect is very systematic it is possible to correct for it in a post-processing step by compensating the suppression with a resolution correction factor (RCF).

---

\(^7\) The simulations were run on the Piz Daint supercomputer at the Swiss National Scientific Supercomputing Centre (CSCS)
5.1 Cosmology dependence of the resolution-induced power suppression

In general it has to be assumed that the precise shape of the resolution-induced power suppression (and equivalently that of the RCF) depends on cosmology. To test this statement we compute a series of simulations for twenty different cosmologies. We choose these cosmologies to be a subset of the experimental design (ED) of Euclid Emulator 2. Of course, running multiple high-resolution reference runs in large boxes is too expensive, so we perform our test in a smaller box. We run these simulations in boxes of $L = 128 \, h^{-1} \, \text{Mpc}$ side length with $N^3 = 384^3$ and $N^3 = 1024^3$ particles, corresponding to resolution parameters of $\ell^{-1} = 3 \, h^{-1} \, \text{Mpc}$ and $\ell^{-1} = 8 \, h^{-1} \, \text{Mpc}$, respectively. The RCF is then simply defined as:

$$f_{\text{res}}^{3 \rightarrow 8}(k, z; \epsilon) = \frac{p(\ell^{-1} = 3 \, h^{-1} \, \text{Mpc}^{-1})(k, z; \epsilon)}{p(\ell^{-1} = 8 \, h^{-1} \, \text{Mpc}^{-1})(k, z; \epsilon)},$$

(23)

where $\epsilon$ denotes a specific cosmology. We thus compare the same resolutions as we do in the case of the dark-green and yellow curves in Fig. 3. We show the results of this analysis in Fig. 4. In this figure, the two upper panels correspond to the lower panels in Fig. 3. The variability in the ratio of power spectra due to variability in the cosmology is then shown in the lower panels of Fig. 4. As is clearly visible, the cosmology dependence of this ratio is weak, particularly at very low redshifts. Also at high redshifts, the 1σ-standard deviation is considerably smaller than the biases at the same $k$ modes for all tested cosmologies. Accordingly, we may simplify

$$f_{\text{res}}^{3 \rightarrow 8}(k, z; \epsilon) \approx f_{\text{res}}^{3 \rightarrow 8}(\epsilon^*; k, z) := f_{\text{res}}^{3 \rightarrow 8}(k, z),$$

(24)

where $\epsilon^*$ denotes any reasonable cosmology not too different from the cosmologies for which low resolution simulations are run. In our case the Euclid Reference Cosmology, the Planck 2015 best-fit cosmology or the central cosmology of the parameter box defined below in Table 2 would all be viable choices for $\epsilon^*$.

From this we conclude that a cosmology-independent correction factor, although introducing a new source of uncertainty, improves the power spectrum measurement of the N-body simulation by a few percent at high $k$. Of course, in future work this should be improved even further by emulating the cosmology dependence of the resolution correction factor. This would greatly reduce the newly introduced uncertainty while still mostly removing the bias.

We do compute $f_{\text{res}}^{3 \rightarrow 8}(k, z)$ at the Euclid Reference Cosmology as defined in Table 2. The RCF curve is shown in Fig. 4 for different redshifts covering the entire redshift range of interest. Notice that values $f_{\text{res}}^{3 \rightarrow 8}$ for $k < 2\pi/(128 \, h^{-1} \, \text{Mpc})$ are set to unity (see discussion below in section 5.2).

5.2 Dependence on simulation box size

For this approach to be a practical strategy, the RCF must not depend strongly on simulation box sizes. Otherwise, the RCF would itself rely on high-resolution runs in large boxes which are exactly the simulations that are not affordable. Whether this is the case was tested by
This corresponds to lowering the minimally resolved mass by roughly an order of magnitude from \( \sim 3.3 \times 10^9 M_\odot/h \) to \( \sim 1.7 \times 10^8 M_\odot/h \) (corresponding to a Nyquist frequency of \( k_{\text{max}} \sim 25 h \text{ Mpc}^{-1} \)). As we define the NLC for EuclidEmulator2 with respect to linear theory which is not affected by this resolution effect, the very same correction can be applied to those quantities:

\[
B^{\ell^{-1}=8 h \text{ Mpc}^{-1}}(k, z; c) \approx f^{3-8}_{\text{res}}(k, z) B^{\ell^{-1}=3 h \text{ Mpc}^{-1}}(k, z; c). \tag{26}
\]

6 PROJECTION STUDIES USING CLASS-BASED MOCK EMULATORS: TRAINING EUCLIDEMULATOR2

N-body simulations of the matter field in \( w_0w_a \text{CDM} + \sum m_\nu \) cosmologies are expensive, even when the mass resolution is low. It is hence not affordable to run thousands of simulations that would allow for an in-depth analysis of a given emulator that involves (potentially several) training, test and validation sets. In order to develop an understanding of the final emulation error and its dependence on the dimensionality of the parameter space as well as on the size of the experimental design, we construct mock emulators based on HALOFIT (Bird et al. 2012) data. We have followed this strategy already in Euclid Collaboration: Knabenhaus et al. (2019) where it has proved to yield a reliable estimate for the performance of the real, simulation-based emulator.

In this section we first define the parameter space inside which the emulator is constructed. This parameter space is the same for
Figure 4. The resolution-induced power suppression is systematic and can be corrected for. In this plot we show the resolution corrections $f_{\text{res}}^{k=8}(k,z;\epsilon)$ computed from a sample of 20 cosmologies represented by the yellow curves. The newly introduced uncertainty due to variance in cosmology, which we neglect, is merely $0.5\%$ (standard deviation, green shaded region) or lower at $z = 0$ and at the level of $1\%$ at $z = 2$. Given the fact that the bias itself is at the $5\%$ level at higher redshifts, correcting for the resolution has a positive net effect. The curves corresponding to the Euclid Reference Cosmology are overplotted. We shall use this cosmology to compute the RCF which we apply to all other cosmologies.

The HALOFIT-based emulator as for the actual simulation-based EuclidEmulator2. In a next step we (approximately) optimise the hyperparameters of the model in order to use the resulting “architecture” for the computation of learning curves, etc. We shall then apply the findings to the training of the actual, simulation-based EuclidEmulator2.

6.1 Definition of the parameter space

For EuclidEmulator2 we consider CDM models with dynamical DE and including massive neutrinos, often abbreviated as $\omega_0w_a\text{CDM}+\sum m_\nu$ models. More precisely, this means that we parametrise the considered cosmologies via the following eight parameters:

- $\Omega_b$, the total baryon density parameter in the Universe,
- $\Omega_m$, the total matter density parameter in the Universe,
- $\Sigma m_\nu$, the sum of masses of all neutrino families,
- $h$, the dimensionless Hubble parameter,
- $n_s$, the spectral index,
- $w_0$, the time-independent part of the DE EoS parameter,
- $w_a$, the linearly scale factor-dependent part of the DE EoS parameter,
- $A_s$, the spectral amplitude.

The radiation density $\Omega_{\text{rad}}$ is given by the CMB temperature which we fix at 2.7255 K. The DE density $\Omega_{\text{DE}}$ is then inferred from the flatness condition given by Eq. (4) with $\Omega = 1$ on the left-hand side.

We do not want to include any prior knowledge about a most likely cosmology into the construction of the emulator other than the assumption that a flat $\omega_0w_a\text{CDM}+\sum m_\nu$ model is sufficiently accurate in order to describe our Universe. In the context of EuclidEmulator2, we explicitly ignore alternative gravity and other more exotic cosmological models. Emulators for such models have been published by other research groups as e.g. Winther et al. (2019); Giblin et al. (2019). We thus apply flat (uniform) priors to each of the eight input parameters. In order to have a well defined, normalised prior probability distribution function we thus need to define compact intervals along each dimension over which the final emulator will be defined. Mathematically this means that we have to define intervals $[a_i, b_i]$ over which the final parameter box $\Pi$ is the Cartesian product of all intervals:

$$\Pi := \prod_{i=1}^{8} [a_i, b_i]$$

The choice of the interval boundaries is mostly arbitrary and depends mainly on the tasks that will be tackled by the emulator. Without imposing any restrictions, we assume that EuclidEmulator2 will be mostly applied to MCMC searches of the cosmological parameter space to solve the inverse problem of finding the parameter values
best describing our Universe. It is reasonable to assume that these values are not too far away from the current best-fit values as published by modern cosmological experiments. We centre the parameter box for \textit{EuclidEmulator2} around the cosmology “EE2centre” which is defined in Table 2. Notice that this central cosmology is identical to the Euclid Reference Cosmology (up to two decimal places) for all quantities with \( n \) defined in Table 2. Notice that this central cosmology is identical to the Euclid Reference Cosmology (up to two decimal places) for all cosmologies with \( \Omega_m \) is almost identical to the Euclid Reference Cosmology, as is also given here. \( \Omega_{\text{tot}} \) is the same for all cosmologies, corresponding to \( T_{\text{CMB}} = 2.7255 \, \text{K} \).

| \( \Omega_b \) | min | max | EE2centre | Euclid Reference |
|--------|-----|-----|------------|------------------|
| 0.04   | 0.06 | 0.05 | 0.049     |                  |
| \( \Omega_m \) | 0.24 | 0.40 | 0.32      | 0.319            |
| \( \sum m_\nu \) | 0.0 eV | 0.15 eV | 0.075 eV | 0.058 eV        |
| \( n_s \) | 0.92 | 1.00 | 0.96      | 0.96             |
| \( h \) | 0.61 | 0.73 | 0.67      | 0.67             |
| \( w_0 \) | -1.3 | -0.7 | -1.0      | -1.0             |
| \( w_a \) | -0.7 | 0.7  | 0.0       | 0.0              |
| \( A_s \) | \( 1.7 \times 10^{-9} \) | \( 2.5 \times 10^{-9} \) | \( 2.1 \times 10^{-9} \) | \( 2.1 \times 10^{-9} \) |

For validation we create much larger sets than for training. Notice that sampling large sets with LHS is computationally demanding as it scales polynomially with the number of points to sample. Additionally, we are primarily interested in the performance of the emulator inside an axis-aligned ellipsoid inscribed in the parameter box (Table 2). LHS, however, is designed to be space-filling and thus the high computational cost comes with a low efficiency as due to the high dimensionality of the parameter space most sample points lie outside that ellipsoid. We hence decided to sample the validation sets purely and randomly and filter the sample with an ellipsoidal mask. Following this procedure we generate in total 30 validation sets with roughly 1500 sampling points each (resulting in 45 000 validation points).

In addition to these data sets, we have also created data sets for cosmologies organised in a grid of \( 50 \times 50 \) points in each parameter plane, resulting in another 70 000 \textit{HALOFIT} evaluations. These sets are on the one hand used for analysis of the principal component analysis (PCA) eigenvectors (see section 6.4) and, on the other hand, also for validation purposes (see section 6.7 and 6.9).

For each sampled cosmology in all of the sets mentioned above, we run \textit{HALOFIT} and evaluate the matter power spectrum at redshift \( z = 0 \). This result is then divided by the linear matter power spectrum in order to get the NLC factors which ultimately form the data sets of interest.

6.3 Emulation strategy: PCE

We use a supervised regression technique called polynomial chaos expansion (PCE) to emulate the NLC factor. We use the implementation of this method in the MATLAB package \textit{UQLab}\footnote{\url{www.uqlab.com}} (Marelli & Sudret 2014, 2017; Marelli et al. 2017). PCE in its generality is well documented in several publications such as Xiu & Karniadakis (2002); Blatman (2009); Blatman & Sudret (2009, 2010, 2011); Marelli & Sudret (2017, 2018); Torre et al. (2019) and its application to cosmological emulation is discussed in \textit{Euclid Collaboration}: Knabenhans et al. (2019). As a reminder we repeat that we express NLC factors using PCA and PCE with the following emulation equation

\[
B(k,z;c) \approx \mu_{\text{PCA}}(k,z) + \sum_{j=1}^{n_{\text{PCA}}} \sum_{\alpha \in \mathcal{A}^{p,q,r}} \hat{\eta}_{j}\alpha \phi_j(f(c))|\mathcal{P}_j(k,z),
\]

(28)

where \( c \) stands for a vector of the eight cosmological parameters discussed here which is transformed through \( f \) into the standard unit hypercube \([-1,1]^8\). The PCA quantities are the mean \( \mu_{\text{PCA}}(k,z) \) and the eigenvectors \( \mathcal{P}_j(k,z) \). The actual PCE is given by the inner sum, with \( \phi_j \) being the PCE basis functions, \( \hat{\eta}_{j}\alpha \) the coefficients and \( \alpha \) being an element from a multi-index set \( \mathcal{A}^{p,q,r} \).

We shall also stress again that this procedure of combining PCE and PCA is the standard approach for emulating vector-valued quantities with PCE (Blatman & Sudret 2013). This implies, however, that \textit{EuclidEmulator2}, having a target space of \( n_{\text{PCA}} \) dimensions, is actually a conglomerate of \( n_{\text{PCA}} \) single, scalar-valued emulators.

The actual learning algorithm we use is a regularised (i.e. LASSO-type) version of a least-squares minimisation called Least angle regression (LAR), discussed in detail in Efron et al. (2004); Blatman & Sudret (2011). This regression algorithm minimises the loss function
each set of the above mentioned 2500 data points sampled in the weights on the cosmological parameters is sufficiently polynomial. For emulation, we implicitly assume that the dependence of the PC last decade. As we use a polynomial regression method of finite order papers about cosmic emulators employing PCA published over the the cosmological parameters has not been investigated in any of the others). The coefficients of the principal components (also called PCA also used in other emulators based on different techniques than PCE, such as e.g. Heitmann et al. 2010; Nishimichi et al. 2019 and others). The coefficients of the principal components (also called weights or eigenvalues) are thus the quantities that are actually emulated. To our knowledge, the dependence of those coefficients on the cosmological parameters has not been investigated in any of the papers about cosmic emulators employing PCA published over the last decade. As we use a polynomial regression method of finite order for emulation, we implicitly assume that the dependence of the PC weights on the cosmological parameters is sufficiently polynomial. To investigate if this assumption is justified we perform a PCE on each set of the above mentioned 2500 data points sampled in the coordinate planes and plot the resulting first order coefficients wpc1 as heatmaps. The (w0, wa) plane stands out as for w0 + wa → 0 the corresponding first order coefficient grows exponentially (see Fig. 5). From a physical point of view this does not come as a surprise as a cosmology with such a DE EoS is highly exotic as it implies a DE with an almost matter-like nature shortly after the big bang. Such a cosmology is not of interest to us as it is highly unrealistic. We have thus identified a clear non-polynomial dependence in the functions we try to emulate. We mitigate this problem by masking out the critical region. In practice, we train our emulator based on all cosmologies within our training set but exclude the very 19 cosmologies that do not meet the condition: wa < 0.5. (30)

As we will show in section 6.7, this modification of the training data set is crucial for the performance of the emulator (although its size is decreased from 127 to only 108 training cosmologies!). This does, however, not restrict the allowed input parameters of the resulting emulator in any way. Clearly, the generalisation performance of the emulator is considerably worse in the masked region than for cosmologies with wa < 0.5. We have tried other, less aggressive cuts, too (e.g. cutting along w0 + wa < 0.5) but have found that only the cut along wa < 0.5 leads to a satisfactory generalisation performance.

6.4 Principal component analysis

Vanilla PCE can only predict scalars. In order to create a PCE emulator for a non-scalar quantity like NLC, it is thus mandatory to decompose the full signal into a series such that only scalar coefficients have to be emulated. This is done in the standard way using PCA (also used in other emulators based on different techniques than PCE, such as e.g. Heitmann et al. 2010; Nishimichi et al. 2019 and others). The coefficients of the principal components (also called weights or eigenvalues) are thus the quantities that are actually emulated. To our knowledge, the dependence of those coefficients on the cosmological parameters has not been investigated in any of the papers about cosmic emulators employing PCA published over the last decade. As we use a polynomial regression method of finite order for emulation, we implicitly assume that the dependence of the PC weights on the cosmological parameters is sufficiently polynomial. To investigate if this assumption is justified we perform a PCE on each set of the above mentioned 2500 data points sampled in the coordinate planes and plot the resulting first order coefficients wpc1 as heatmaps. The (w0, wa) plane stands out as for w0 + wa → 0 the corresponding first order coefficient grows exponentially (see Fig. 5). From a physical point of view this does not come as a surprise as a cosmology with such a DE EoS is highly exotic as it implies a DE with an almost matter-like nature shortly after the big bang. Such a cosmology is not of interest to us as it is highly unrealistic. We have thus identified a clear non-polynomial dependence in the functions we try to emulate. We mitigate this problem by masking out the critical region. In practice, we train our emulator based on all cosmologies within our training set but exclude the very 19 cosmologies that do not meet the condition: wa < 0.5. (30)

As we will show in section 6.7, this modification of the training data set is crucial for the performance of the emulator (although its size is decreased from 127 to only 108 training cosmologies!). This does, however, not restrict the allowed input parameters of the resulting emulator in any way. Clearly, the generalisation performance of the emulator is considerably worse in the masked region than for cosmologies with wa < 0.5. We have tried other, less aggressive cuts, too (e.g. cutting along w0 + wa < 0.5) but have found that only the cut along wa < 0.5 leads to a satisfactory generalisation performance.

6.5 About error measurements

As in all machine learning (ML) and uncertainty quantification (UQ) tasks, error quantifications play a central role in this work. Ultimately, we are primarily interested in creating an emulator that generalises well in an L1 sense in the cosmological parameter space. To be more precise, we try to minimise the generalisation error of the emulator given by the maximum of the relative mean absolute error:

\[ \epsilon_{\text{maxMAE}} = \max_{k, z} \frac{\left| w_{\text{true}}(k, z) - w_{\text{predicted}}(k, z) \right|}{w_{\text{true}}(k, z)} \] (31)

Notice that the mean indicated by the angle brackets is taken over the cosmologies c defined in the parameter space C given in Table 2. The maximisation, on the other hand, is performed over the non-regressed parameters k and z. This generalisation error will be approximated by a validation error of the form

\[ \epsilon_{\text{maxMAE}} = \max_{k, z} \frac{1}{N_{\text{val}}} \sum_{e \in C_{\text{val}}} \left| w_{\text{true}}(k, z) - w_{\text{predicted}}(k, z) \right| \] (32)

where Cval designates the set of validation cosmologies.

However, since an estimate of the above error requires a validation set which we do not have readily available in all situations, we also often use a cross-validation error metric as an alternative. This is given by the leave-one-out (LOO) error defined as

\[ \epsilon_{\text{LOO}, i} = \frac{1}{n_{\text{red}}} \sum_{i=1}^{n_{\text{red}}} \left( \hat{w}_{\text{true}, j} \left( e_{(i)} \right) - w_{\text{predicted}, j} \left( e_{(i)} \right) \right)^2 \] (33)

where \( n_{\text{red}} \) stands for the eigenvalues of the PCA which are the quantities that are actually emulated in this work (for more details on this refer to Euclid Collaboration: Knabenhs et al. 2019). Each \( w_{j} \) corresponds to the inner sum in Eq. (28) running over the multi-index \( a \). To compute this error one trains a PCE emulator on all training example but the i-th one (indicated by the superscript \( \text{PCE}/i \)). In this very example the emulator is then evaluated (second term in the numerator). The quantity is finally rescaled by the overall variance of the quantity \( w_{j} \). More details on this quantity and how to compute it efficiently can be found in Marelli & Sudret (2017). It shall be emphasised that the subscript \( i \) runs over cosmologies in the training set and no reference to any validation examples is made. Further, this metric measures the emulator performance not in the NLC space but rather in the more abstract, associated principal component space.
(hence the subscript \( j \) which refers to the order of the principal component).

6.6 Hyperparameter optimisation using the Halofit mock data

The hyperparameters of PCE for emulation of non-scalar quantities are given by

- the minimum percentage \( a_{\text{PCA}} \) of explained variance retained in the PCA (strongly related to the number \( n_{\text{PCA}} \) of principal components taken into account),
- the polynomial order \( p \) at which the PCE is truncated,
- the maximum interaction \( r \) (number of factors per monomial in PCE),
- the \( q \)-norm.

Remember that actually there is a separate PCE for each principal component (as described in section 6.3). As a result, the hyperparameters \( p, q \) and \( r \) can be chosen differently for each \( i = 1, \ldots, n_{\text{PCA}} \) and the accuracy parameter \( a \) is the only hyperparameter that has to be chosen globally for obvious reasons. As training a PCE is relatively cheap, there is no need for a sophisticated optimisation algorithm. Rather, we perform a (partially greedy) grid search over the grid given by

\[
a_{\text{PCA}} \in \{0.9, 0.99, \ldots, 0.99999999\},
\]

\[
r \in \{1, 2, \ldots, 5\},
\]

\[
p \in \{2, 3, \ldots, 20\},
\]

\[
q \in \{0.3, 0.35, 0.4, \ldots, 0.9\}.
\]

The small set of low numbers looked at for \( r \) are motivated by the “sparsity-of-effects” principle (Marelli & Sudret 2017). Strictly speaking, there are two steps here. In the first step a vector of \( a_{\text{PCA}} \)-values is created and, in the second step, for each value of \( a_{\text{PCA}} \), a multitude of \((p, q, r)\)-grids is searched (one for each principal component). This is important because the optimal point in the \((p, q, r)\)-grid is chosen based on a different criterion than the optimal \( a_{\text{PCA}} \) value. The former is chosen based on a minimisation of the LOO cross-validation error without ever seeing a validation point. This happens entirely on the level of principal component weights (i.e. eigenvalues of the covariance matrix) and thus this step is performed independently for each principal component. For identification of the (near-)optimal \( a_{\text{PCA}} \) value, in contrast, an emulator is trained for each value of \( a_{\text{PCA}} \) and evaluated on a validation set. We then aim to minimise \( \hat{e}_{\text{maxMAE}} \) as defined in Eq. (32). Notice that, while the latter error is the correct quantity to look at when judging the overall performance of the emulator, this can only be investigated once a separate emulator is trained (and, as well, a separate set of hyperparameters is optimised) for each principal component. It is of paramount importance to understand that we go through this procedure really only to fix the values for \( a_{\text{PCA}} \). As \( p, q \) and \( r \) can be optimised without the need of a separate validation set, they can be optimised once the final emulator is being trained based on actual simulations without the risk of overfitting to a validation set. After evaluating the \( \hat{e}_{\text{maxMAE}} \) for all candidate values of \( a_{\text{PCA}} \), we conclude that

\[
a_{\text{PCA}} = 0.999999
\]

is the optimal (i.e. loss minimising) value for this global hyperparameter. If we continue to increase \( a_{\text{PCA}} \), we find that we are attempting to capture the numerical noise in the simulations, leading to an increase again of \( \hat{e}_{\text{maxMAE}} \).

6.7 Learning curves

Now that we have defined the bounds for all parameters, have created the necessary data sets and optimised the global hyperparameter \( a_{\text{PCA}} \), it is natural to ask the following two questions:

- How does the validation error of the emulator decrease as the number of training examples increases?
- Given a fixed number of training examples, how does the validation error increase as a consequence of adding the two additional dimensions compared to version 1 of EuclidEmulator?

To answer these questions we train emulators for different cosmological models with different dimensionalities, namely \( \Lambda \)CDM (5D), \( w_0 \)CDM (6D, corresponding to EuclidEmulator1), \( w_0w_a \)CDM (7D) and \( w_0w_a \)CDM+\( \sum m_i \) (8D, corresponding to EuclidEmulator2). Each model is trained on the series of training sets mentioned in section 6.2. The emulators are then validated on the 30 ellipsoidal validation sets. In this context it becomes evident why we have produced so many validation sets: It allows one to get statistics on the validation error (namely the standard error of the estimated mean error). Notice that such a representative test is by far beyond what is achievable with simulation data as several tens of thousands of simulations with at least moderate mass resolution would have to be run. The resulting learning curves are plotted in Fig. 6.

While it is easy to achieve validation errors \(< 1\%\) for the 5D and 6D models with only 50 training examples, the complexity of the 7D and 8D models is considerably higher, particularly if we train based on samples in the entire original parameter space without masking out the problematic region in the \((w_0, w_a)\) plane (see the discussion in section 6.4). In this scenario, we would require \( \geq 400 \) training examples to reach accuracies of \( 1\% \) or better. Masking out the region where \( w_0 + w_a \sim 0 \) in the training set, reduces the training set size to only 100 to 200 examples.

Notice that in Fig. 6 we show the error estimate for \( n_{\text{ED}} = 127 \) for the case of \( w_0w_a \)CDM+\( \sum m_i \) cosmologies. This measurement was added in hindsight because it turned out that the actual simulation-based emulator would achieve the target accuracy already with a training set of only 127 simulations selected from an LHS of size 200 (see section 7.2 for a short discussion).

6.8 Application to the training of EuclidEmulator2

We shall now anticipate some training aspects of the actual, simulation-based EuclidEmulator2. We use the same surrogate model in order to train the emulator as for the HALOFIT-based mock emulator discussed above and for EuclidEmulator1, i.e. sparse PCE combined with PCA. The number of principal components is defined through the threshold value for the minimally explained variance in the data set (which is independent of the size of the training set). As this was investigated in section 6.2 and found that \( a_{\text{PCA}} = 0.999999 \) is the optimal value, we can now use the same value for the training of the actual emulator. Having found in the learning curves analysis that a training set of 127 cosmologies is enough to achieve the targeted accuracy, this value for \( a_{\text{PCA}} \) corresponds to retaining \( n_{\text{PCE}} = 14 \) principal components (i.e. the target space of the full vector-valued emulator is 14-dimensional).

Also, we found for the HALOFIT-based mock emulator that setting the maximal interaction \( r_{\text{max}} = 4 \), the maximal polynomial order \( p_{\text{max}} = 20 \), and varying the \( q \)-norm between \( q_{\text{min}} = 0.3 \) and \( q_{\text{max}} = 0.9 \) leads to convergence in the selection of terms in the expansion series. So we recycle this here, too. The parameters \( p \) and \( q \), as well as the actual interaction number \( r \), are optimised individually for
Figure 6. Learning curve for CLASS-based emulators for four different cosmological models. On the y-axis the estimated mean of the validation error distribution over the 30 different validation sets is plotted. The error bars show the standard error of this estimated mean. Notice that we explicitly plot the point at $n_{\text{ED}} = 127$ for the $w_0w_a\text{CDM}+\Sigma m_r$ model as this point corresponds to our actual training set. The learning curves have been computed (i) taking all training examples into account (dashed lines, labelled “uncut”) and (ii) ignoring those training examples with $w_{ai} > 0.5$ (solid lines, labelled “cut”). For the $\Lambda\text{CDM}$ and $w_0\text{CDM}$ models this distinction makes no difference because for those models $w_{ai}$ was fixed to 0.

Valiation examples were sampled randomly inside axis-aligned hyperellipsoid inscribed in the parameter box (see section 6.2 for a description of the data sets).

Table 3. Table with optimal hyperparameter values for all 14 scalar-valued PCEs of EuclidEmulator2. The resulting PCE contains 574 non-trivial terms.

| PC order | $p$ | $r$ | $q$ |
|----------|-----|-----|-----|
| 1        | 3   | 2   | 0.45|
| 2        | 3   | 2   | 0.45|
| 3        | 4   | 2   | 0.5 |
| 4        | 4   | 2   | 0.45|
| 5        | 3   | 2   | 0.45|
| 6        | 4   | 2   | 0.4 |
| 7        | 4   | 2   | 0.5 |
| 8        | 4   | 2   | 0.45|
| 9        | 4   | 2   | 0.5 |
| 10       | 16  | 3   | 0.4 |
| 11       | 4   | 2   | 0.5 |
| 12       | 12  | 4   | 0.5 |
| 13       | 4   | 3   | 0.5 |
| 14       | 15  | 2   | 0.4 |

each principal component and listed in Table 3. The full training process (i.e. optimising the values for $p$, $r$ and $q$ for each principal component and fitting the coefficients) takes only 9 seconds on a usual MacBook pro with a 2.8 GHz Intel Core i7 CPU. As a result we get a PCE with a total of 574 terms (all other coefficients vanish). Notice that this corresponds to a very small number of terms, i.e. an extremely sparse PCE, as in our case there were 3 108 105 terms without sparsification.

6.9 Performance estimation of the mock emulator

From the learning curves presented above we can expect EuclidEmulator2 to be sub-1% accurate at $z = 0$ over the entire $k$ range of interest for the available training set of 108 simulations (being a subset of the originally planned set of 200 training examples). While this is the final goal, it is yet interesting to see how the error evolves as a function of cosmology. For this we evaluated the HALOFIT-based mock emulator on all 70 000 validation cosmologies sampled in the 28 parameter planes of the feature space. The results are 28 error maps that we show in Appendix B. It is clearly visible that the validation error is below 2% (and hence at the same level as the mass resolution-related uncertainty in the simulations at small scales) for the vast majority of cosmologies lying inside the axis-aligned hyperellipsoid inscribed in the parameter box (indicated by a grey ellipse in the error maps). Only for cosmologies with a large value of $w_0 + w_a$ the error grows to ~ 5%. We reiterate, however, that on average over the entire 8D-hyperellipsoid the error drops below 1%. Outside that ellipsoidal region, the errors sometimes exceed the 2% limit. We also designate the $w_a = 0.5$ boundary by a grey dashed line.

7 THE TRAINING SET OF EUCLIDEMULATOR2

7.1 Experimental design: sampling

We sample the points in the parameter space defined in Table 2 using LHS, as we have done already in Euclid Collaboration: Knabenhans et al. (2019). LHS is a very straightforward sampling technique that is widely used and accepted in the cosmological emulator community (Heitmann et al. 2009, 2010, 2014; Nishimichi et al. 2019; DeRose et al. 2019; Gratton & Wilkinson 2019; Rogers et al. 2019) and extensively presented in the statistical sampling literature (McKay et al. 1979; Tang 1993; Liefvendahl & Stocki 2006; Crombecq et al. 2011; Damblin et al. 2013; Sheikholeslami & Razavi 2017; Yang et al. 2017; Garg & Stogner 2017; Swiler et al. 2006). Endowed with an additional optimisation step (we use a distance-based criterion), its main advantage is that it combines good space-filling properties with a high degree of randomness. For an in-depth explanation of the exact steps we go through to generate the sample we refer to Euclid Collaboration: Knabenhans et al. (2019).

We chose to generate a sample with 200 points based on the argument that in Fig. 6 we show that a training set of this size should be large enough to achieve a validation error below 1%, while a set of only 100 examples is expected to just miss this requirement in the 8D parameter box. In fact, we use exactly the same LHS of size $n_{\text{ED}} = 200$ to run the simulations as we used in the investigation of the mock emulators in section 6.

The resulting sample of cosmologies is shown in blue in Fig. 7.

7.2 Experimental design: simulations

As the generation of each training example (i.e. each pair of simulations per cosmology) corresponds to an investment of about 4000 node hours of computation, we tested the performance of PCE-based emulators along the way also before the planned training set of 200 cosmologies was completed. Doing so we noticed that the emulator achieved the targeted sub-percent accuracy when trained on only 127 examples. In this case the sample of course is no longer an LHS but rather resembles a random sampling. Consequently, there was no need to invest more time and effort in running the remaining 73 pairs of simulations. The completed set of 127 simulation pairs used for the training of EuclidEmulator2 is plotted in Fig. 7. Notice that it was for this very reason why we also looked at the sample of size 127 when investigating the HALOFIT-based mock emulator in section 6.7.

From this fact we can learn two important conclusions for future projects:
Figure 7. Distribution of LH-sampled cosmologies represented in the coordinate planes of the parameter space. The blue circles and histograms show the full ED of 200 data points while the green dots indicate the subset of 127 training examples actually used to create EuclidEmulator2. The ranges of each subplot exactly matches that specified in Table 2.

- At least as long as the marginal distributions of sample points along all parameter dimensions do not have regions where there are no sampling points at all, LHS is not a necessity for good performance of PCE as random sampling works fine too.
- When generating an ED is related to large computational costs (and hence to a non-negligible risk of failing to generate a large sample in one go), it is advisable to choose an enrichable sampling technique such as e.g. the LHS-based adaptive response surface method (LHS-ARSM, Wang 2003) or active learning (as e.g. done in Rogers et al. 2019).

We thus use an experimental design of 127 (respectively 108 after applying the cut in $w_d$) P+F simulations randomly sampling the parameter box. Each simulation samples the power spectrum at 613 $k$-modes and 100 time-steps between $z_{\text{int}m} = 10$ and $z_{\text{fin}} = 0$. While all this data are used to compute the emulator, we only allow the user to emulate up to $z_{\text{max}} = 3.0$ because the overall accuracy decreases considerably for higher redshifts (primarily because the underlying simulations have not converged for higher redshifts as can be extrapolated from Fig. 3).

7.3 Post processing: Computation of the NLC

We compute the NLC for each power spectrum by dividing the non-linear power spectrum resulting from the simulation by the linear theory power spectrum computed by CLASS. Notice that this is different from what is done for EuclidEmulator1 where the NLC was computed via a division by the re-scaled power spectrum measured from the simulation particle realisation at the initial condition. For training the emulator, the NLC is converted into log space because we have shown in Euclid Collaboration: Knabenhs et al. (2019) that this improves the generalisation of the emulator. We stress that the resolution correction factor introduced in section 5 is not applied to the training data set such that users of EuclidEmulator2 can decide individually whether they want to apply this correction or not to the emulated result. We compile the NLC data into a data matrix $D_{\text{CDM+b}} \in \mathbb{R}^{n_{\text{ED}} \times n_k}$. This data matrix is then decomposed into its principal component basis $\{\text{PC}_i \mid i \leq n_{\text{PCA}}\}$ where $n_{\text{PCA}}$ denotes the number of principal components taken into account. As a result, the $m$-th row in $D$ can be represented as follows:

$$D_m = \sum_{i=1}^{n_{\text{PCA}}} w_i(c_m) \text{PC}_i(k, z),$$

where the argument $c_m$ of the PC weight stands for the vector of parameters defining the $m$-th cosmology and the arguments of the principal components are the $k$ mode and the redshift. We hence build $n_{\text{PCA}}$ individual training sets defined by

$$T_i = \{w_i(c_m) \mid m \leq n_{\text{ED}}\}, \quad \forall i \in [i, \ldots, n_{\text{PCA}}]$$

that are used to train the $n_{\text{PCA}}$ individual, scalar-valued emulators.
8 EMULATOR PERFORMANCE, ERRORS AND SENSITIVITY TO PARAMETERS

8.1 Sensitivity analysis

As for EuclidEmulator1, we have again performed a Sobol’ analysis to investigate the relative importance of each cosmological parameter on the final NLC. Notice that since the parameterisation of the emulator changed significantly from EuclidEmulator1, it cannot be expected that the Sobol’ indices remain unaltered. Clearly, for EuclidEmulator2, the matter density parameter $\Omega_m$ dominates the behaviour of the resulting NLC. At the same time, to first principal component order, $\sum m_\nu$ and $n_s$ are almost entirely negligible. While in the case of neutrino masses this does not come as a surprise (the effect of massive neutrinos is mostly captured by the linear signal already), one might not have guessed that for the spectral index. This is resolved when looking at the Sobol’ indices of the second principal component where the neutrino mass still has almost no impact at all, while the spectral impact becomes actually the dominant parameter.

The fact that the sum of the neutrino masses is almost entirely negligible when computing the NLC supports our suggestion mentioned in Euclid Collaboration: Knabenhans et al. (2019) that to good approximation one can emulate nonlinear power spectra with massive neutrino cosmology by simply computing the corresponding linear power spectrum and multiplying that by an NLC as produced by EuclidEmulator1, i.e. an NLC that does not know anything about massive neutrinos. The test of this hypothesis is deferred to section 8.2.2 (see Fig. 11 in particular).

8.2 Generalisation Performance of EuclidEmulator

In this section we shall compare EuclidEmulator2 to other fast prediction techniques such as HALOFIT (Bird et al. 2012), HMCcode (Mead et al. 2016), CosmicEmu (Lawrence et al. 2010, 2017), the very recent emulator based on the BACCO simulation project (Angulo et al. 2020), hereafter referred to as the “BACCO-emulator”, and the predecessor EuclidEmulator1 (Euclid Collaboration: Knabenhans et al. 2019) as well as with PKDGRAV3 (Potter et al. 2017; Potter & Stadel 2016; Stadel 2001) simulations. While the comparisons of EuclidEmulator2 with EuclidEmulator1 and with PKDGRAV3, respectively, can be conducted at the NLC-level, all comparisons with HALOFIT, HMCcode and CosmicEmu are performed at the level of the fully nonlinear power spectrum. To this end we multiply the NLC computed by EuclidEmulator2 with a linear power spectrum computed by CLASS for the same cosmological parameters.

We compare each pair of predictors in two ways. On the one hand, we compare them for a set of different cosmologies at redshift $z = 0$, while on the other hand we chose a single cosmology equal to the Euclid Reference Cosmology but with a higher total neutrino mass for comparison at different redshifts $z \leq 2$.

For the comparison between EuclidEmulator2 and PKDGRAV3 we have used a small validation data set containing three validation cosmologies.

In order to compare EuclidEmulator2 to EuclidEmulator1, we primarily focus on two extreme cases: the Euclid Reference Cosmology as defined in Euclid Collaboration: Knabenhans et al. (2019), once with massless and once with massive neutrinos.

For all comparisons with HALOFIT, HMCcode and CosmicEmu in this section, we choose the cosmologies from a set of 291 cosmologies, put together by an LHS of size 200, cosmologies along the coordinate axis and along one of the diagonals. From this set, we filter out all cosmologies that are not accepted by any of the emulators. This results in a set of 84 comparison cosmologies (47 in the case of the comparison to the BACCO-emulator).

8.2.1 Comparison of EuclidEmulator2 and PKDGRAV3 simulations

We start our series of comparisons by checking how well EuclidEmulator2 is able to approximate simulation data. To this end, we generate a validation set of P+F simulations with the same resolution as the training data. The validation set contains only three cosmologies (all unseen by the training process) because the generation of a significantly bigger training set is too expensive. These three cosmologies are all sampled from the ellipsoid inscribed the parameter box with axes given by the limits of each parameter range.
8.2.2 Comparison of EuclidEmulator1 and EuclidEmulator2

It is natural to compare the performance of EuclidEmulator2 with its predecessor EuclidEmulator1 (cf. Fig. 11). As both emulators predict the NLC, we can perform the comparison on this level. In a first step we perform this comparison using the version of the Euclid Reference Cosmology as defined in Euclid Collaboration: Knabenhans et al. (2019), i.e. a cosmology without massive neutrinos.

We observe very good agreement on large scales which is achieved by construction as the variability of the NLC is negligible at these scales. The sub-percent differences at these scales are due to the fact that the simulations volumes of the training simulations underlying both emulators are different.

On intermediate scales around the baryon acoustic oscillations (BAO) one observes a peaky pattern at the level of $\leq 2.5\%$. We show in Fig. 12 that this can be explained by cosmic variance. We reiterate a point already reported in Euclid Collaboration: Knabenhans et al. (2019): For EuclidEmulator1, cosmic variance is strongest not on large but on intermediate scales. This is because on large scales the cosmic variance is not significantly amplified by nonlinear structure formation. As a consequence the residual (after pairing and fixing) cosmic variance drops out because we compute the NLC for EuclidEmulator2 by dividing the nonlinear power spectrum at $z$ by the properly rescaled initial condition. On the other hand, we show in Fig. 2 that for EuclidEmulator2 we choose the volume large enough to render cosmic variance irrelevant. However, on intermediate scales, the residual cosmic variance did not get cancelled out even for EuclidEmulator1 as on these scales it is already non-negligibly amplified by nonlinear evolution. As a result, when comparing EuclidEmulator1 to EuclidEmulator2, one actually divides two signals with oscillatory behaviour on intermediate scales, manifesting itself as oscillations on intermediate scales observed in Fig. 11.

Unsurprisingly, EuclidEmulator1 underestimates power at small scales compared to EuclidEmulator2. This is simply due to too low a mass resolution of the training simulations of EuclidEmulator1. Here we can confidently report that the baseline in Fig. 11 given by EuclidEmulator2 is the (more) correct answer.

Now, as we claimed above that massive neutrinos do not have a significant impact on the NLC we should test this hypothesis. To actually do so, we compared predictions of EuclidEmulator1 and EuclidEmulator2 to each other in Fig. 11: We evaluated each emulator at the respective cosmology in Table 4. The ratio between the two NLC factors is clearly dominated by cosmic sample variance and resolution effects. This suggests that EuclidEmulator1 can indeed be used to estimate the NLC to good approximation for $z < 0.5$. 

Figure 9. Comparison of NLC factors predicted by EuclidEmulator2 and ones computed directly from PKDGRAV3 simulations (averaged over three different cosmologies). The agreement is at the sub-percent level and thus respects the target accuracy.

Figure 10. Comparison of full nonlinear power spectra between EFS2 and resolution-corrected product of a CLASS linear power spectrum times an EuclidEmulator2-emulated NLC at the Euclid Reference Cosmology as defined in Table 2. The agreement is generally very good and even at the 1% level or better at small scales. The oscillations at linear scales are expected as EFS2 is not a P+F simulation. We also show the comparison to EuclidEmulator2 when we do not apply the resolution correction factor, which shows a clear 2% bias at nonlinear scales.
Figure 11. In this figure we compare the NLC prediction of EuclidEmulator2 to that of EuclidEmulator1 for two cosmologies: one for which $\sum m_\nu = 0.0 \, \text{eV}$ (left panel) and $\sum m_\nu = 0.15 \, \text{eV}$ for the other (right panel). $A_s$ is the same in both cosmologies, however the value of $\sigma_8$ differs due to the mass difference in the neutrino sector. It is not surprising that the agreement on the largest scales is nearly perfect for both cosmologies as both emulators return values close to unity irrespective of the cosmology by construction. The oscillations around BAO scales are due to cosmic variance. The vertical, black, dashed lines indicate the two $k$-modes where the curves for EuclidEmulator1 and EuclidEmulator2 deviate the most from each other in Fig. 12. The time-dependent mismatch on small scales is due to the different mass resolutions of the training of the two emulators. A key point is that this plot shows that the agreement between EuclidEmulator1 and EuclidEmulator2 is not only very good for massless neutrino cosmologies where EuclidEmulator1 is supposed to work well by construction, but also for $w_0$CDM+$\sum m_\nu$ models which EuclidEmulator1 was not designed for. This suggests that EuclidEmulator1 is able to predict the nonlinear power spectrum within 1% of accuracy for $z < 0.5$ even if the cosmology contains massive neutrinos (of course, the linear power spectrum must be computed taking the neutrino masses into account).

for $w_0$CDM+$\sum m_\nu$ models. The key point to get the correct answer is to account for the difference in $\Omega_m$. Given a particular $\Omega_{\text{EE}2} = \Omega_0 + \Omega_{\text{CDM}} + \Omega_\nu$ for EuclidEmulator2, one needs to choose $\Omega_{\text{EE}1} = \Omega_0 + \Omega_{\text{CDM}}$ for EuclidEmulator1, such that $\Omega_{\text{EE}2} = \Omega_{\text{EE}1} + \Omega_\nu$. As a result, the value for $\sigma_8$ has to be adjusted accordingly. The value $\sigma_8 = 0.799$, corresponding to $A_s = 2.1 \times 10^{-9}$ at the “EE1” cosmology listed in Table 4, was computed by CLASS.

8.2.3 Comparison to HALOFIT

EuclidEmulator2 is compared to the extension of HALOFIT by Bird et al. (Bird et al. 2012) in Fig. 13. The comparison across multiple cosmologies shows almost perfect agreement for all cosmologies on large scales. This is expected as HALOFIT builds on linear theory as does EuclidEmulator2. On intermediate scales around BAOs we find systematic oscillations which are in agreement with what we have found in the corresponding comparison between HALOFIT by Takahashi et al. (Takahashi et al. 2012) and EuclidEmulator1 (Euclid Collaboration: Knabenhans et al. 2019). While there we attributed those oscillations to HALOFIT’s inability to capture the BAOs correctly, this may play a less relevant role for this version of HALOFIT. Rather, the oscillations may be mostly explained by the higher mass resolutions and smaller simulation box sizes used in Bird et al. (2012) compared to the those used in this work. The fact that on average

Table 4. Mutually corresponding cosmologies in order to approximate the EuclidEmulator2-based NLC with an EuclidEmulator1-based NLC neglecting massive neutrinos. The relative difference of the NLC factors produced with the respective version of EuclidEmulator are shown in Fig. 11, right panel. Notice that $\Omega_{\text{rad}}$ is the same for both cosmologies corresponding to $T_{\text{CMB}} = 2.7255 \, \text{K}$. 

| $\Omega_0$ | $\Omega_m$ | $\sum m_\nu$ | $n_s$ | $h$ | $w_0$ | $w_a$ | $\sigma_8$ | $A_s$ |
|----------|------------|---------------|-------|-----|-------|-------|----------|------|
| 0.049    | 0.3154     | 0.0 eV        | 0.96  | 0.67| -1.0  | 0.0   | 0.799    | -3.1 | 2.1 \times 10^{-9} |
less power is found by EuclidEmulator2 compared to HALOFIT (at the level of roughly 3%) is consistent with the findings presented in Fig. 2 of Bird et al. (2012) where it is reported that PM-based neutrino simulations tend to find less power on intermediate to small scales compared to simulations treating neutrinos as particles. This is also why on small scales we then find an overestimation of power in HALOFIT relative to EuclidEmulator2. The mean including the $1\sigma$-region stays within the 5 to 10% error margin, respecting the bounds published in Takahashi et al. (2012); Bird et al. (2012). The error evolution with redshift looks again very similar to what we have already found for EuclidEmulator1. The systematic oscillations on intermediate scales grow with time while on small scales the disagreement is largest for higher redshifts.

8.2.4 Comparison to HMCode

The comparison of EuclidEmulator2 and HMCode is shown in Fig. 14. In Mead et al. (2016) it is reported that HMCode achieves an accuracy of a few percent for cosmologies with massive neutrinos and dynamical DE. We find an agreement at the few percent level both over all tested cosmologies as well as over all redshifts (see Fig. 14). Independent of redshift and cosmology the agreement on large scales is virtually perfect. This does not come as a surprise as HMCode is built on top of HALOFIT which performs almost perfectly on these scales. Too. Around BAO scales we find a systematic overprediction of power in HMCode relative to EuclidEmulator2 (degrading as $z$ increases) which relaxes again at $k \sim 0.6\, h\, \text{Mpc}^{-1}$. On small scales, however, the variance in the relative difference is quite large (though always within the few percent limit) both as the cosmology varies as well as over the probed redshift range.

8.2.5 Comparison to CosmicEmu

The comparison of EuclidEmulator2 and CosmicEmu is shown in Fig. 15. In Lawrence et al. (2017), they report that for predictions of the 8-parameter model they find an approximation accuracy of 10 to 15% or better. On average over all probed cosmologies, the comparison error is far below that and it is even relatively constant over the entire $k$-range of interest. Even the standard deviation of the entire set of comparisons is only at the level of five percent over all $k$ (for $z = 0$). There are, however, a few cosmologies for which the comparison is significantly poorer. The fact that there is no $k$-region where the comparison is nearly perfect is explained by the fact that CosmicEmu emulates the full nonlinear power spectrum directly while EuclidEmulator2 emulates the NLC only. It is thus not surprising that there is some generalisation error also on large scales for CosmicEmu, while EuclidEmulator2 is accurate in this regime by construction.

8.2.6 Comparison to the BACCO-emulator

The quantity emulated by the BACCO-emulator (Angulo et al. 2020) is also the NLC. For this very reason the comparison between EuclidEmulator2 and the BACCO-emulator (version 1.1.1) is conducted at the level of the NLC rather than at the fully nonlinear power spectrum level. The result of this comparison is shown in Fig. 16. Clearly, the agreement between these two state-of-the art emulators is extremely good over wide ranges of spatial scales and redshifts. First we discuss the comparison between the two emulators at the Euclid Redshift Cosmology for different redshifts. Notice that the BACCO-emulator allows prediction of the NLC only up to $z = 1.5$. For this reason, the comparison at $z = 2$ included in the previous comparisons to HALOFIT, HMCode and CosmicEmu is omitted here. It is found that the agreement at the tested redshifts is mostly at the per cent level, where a suppression of power in the BACCO-emulator is observed relative to EuclidEmulator2 at small scales. This is explained by the fact that the BACCO-emulator is based on simulation with a resolution parameter of $\ell^{-1} = 3\, h\, \text{Mpc}^{-1}$ while the

---

**Figure 12.** Comparison of EuclidEmulator1 and EuclidEmulator2 to the NLC computed from an ensemble average of fifty Gaussian random field simulations. It can be seen that the cosmic variance is responsible for the oscillatory peaky pattern in Fig. 11, as the vertical, black, dashed lines correspond to the two highlighted peaks in that figure.

**Figure 13.** The comparison between HALOFIT (Bird et al. 2012) and EuclidEmulator2 is consistent with what is found in Euclid Collaboration: Knabenbans et al. (2019). While there are disagreements on intermediate and nonlinear scales, the measured errors stay within the bounds of 5 to 10% as reported in Bird et al. (2012).
Figure 14. Comparison between EuclidEmulator2 and HMCode over a set of 84 cosmologies at \( z = 0 \). The agreement on large scales is nearly perfect while the errors stay within the few percent level over all \( k \) as reported in Mead et al. (2016).

Figure 15. Comparison between EuclidEmulator2 and CosmicEmu over a set of 84 comparison cosmologies at \( z = 0 \). The mean and even the standard deviation are well below the 10 to 15% level reported in Lawrence et al. (2017).

Figure 16. Comparison between EuclidEmulator2 and the BACCO emulator over several redshifts. The relative error is mostly smaller than 3% corresponding to the expected accuracy of the BACCO emulator reported in Angulo et al. (2020).

EuclidEmulator2 NLC were resolution corrected as explained in section 5.

The two emulators were also compared at 47 different cosmologies at redshift \( z = 0 \). The overall agreement is also mostly at the 3\% level over the entire \( k \)-range, where it is reported in Angulo et al. (2020) that the BACCO-emulator is expected to predict the NLC with an accuracy of 3\%. The high-frequency oscillatory pattern at intermediate \( k \)-scales may be explained by a somewhat poor sampling of the BAOs in the BACCO simulations.

9 EXPLORATION OF DEGENERACIES IN THE NONLINEAR MATTER POWER SPECTRUM

EuclidEmulator2 is expected to be applied to parameter forecasts because it is able to very efficiently produce highly accurate predictions of the NLC and hence of the fully nonlinear power spectrum. The Euclid mission aims at measuring the absolute neutrino mass scale (Laureijs et al. 2011) by analysing the effects of neutrinos on cosmic structure formation. Massive neutrinos suppress power particularly at small scales (see e.g. Viel et al. 2010; Bird et al. 2012; Hannestad et al. 2012 and others). While this is true also for the linear power spectrum, the effect is largest in the nonlinear power at scales around \( k \sim 1 \, h \, \text{Mpc}^{-1} \). The reaction of the linear and nonlinear power spectra to varying the total neutrino mass is shown in Fig. 17. In this figure, the base line is given by the Euclid Reference Cosmology with \( \sum m_\nu = 0.058 \, \text{eV} \).

While of course a proper Bayesian inference is required to forecast the neutrino mass (as is done e.g. in Audren et al. 2013), we shall use EuclidEmulator2 in order to investigate the uniqueness of the neu-
The EuclidEmulator2

Figure 17. Linear and nonlinear power spectra variation due to varying the cosmological parameters (neutrino mass in the left panel, \(w_a\) in the right panel). The power spectra are normalised to the Euclid Reference Cosmology power spectrum. The NLCs for the nonlinear power spectra have been predicted with EuclidEmulator2 in all cases. These plots show clearly that the power spectra differ the most at scales around \(k \sim 1 \ h \ Mpc^{-1}\).

Table 5. Two cosmologies with nonlinear power spectra that agree to better than 1\% over all scales \(0.01 \ h \ Mpc^{-1} \leq k \leq 10 \ h \ Mpc^{-1}\) at \(z = 0\).

| \(\Omega_b\) | \(\Omega_m\) | \(\sum m_\nu\) | \(n_s\) | \(h\) | \(w_0\) | \(w_a\) | \(A_s\) |
|---|---|---|---|---|---|---|---|
| 0.049 | 0.3194 | 0.015 eV | 0.96 | 0.67 | -1.0 | 0.0 | 2.1 \times 10^{-9} |
| 0.049 | 0.009 | 0.00 eV | 0.97 | 0.67 | -1.0 | 0.0 | 2.01 \times 10^{-9} |

10 CONCLUSION

For this work we have modified PKDGRAV3 in such a way that DM is not only evolved fully nonlinearly due to self-interaction but also is subject to an additional gravity source due to massive neutrinos, radiation, DE and the metric field perturbations. The latter four species themselves are, however, only evolved linearly. To this end, PKDGRAV3 has been interfaced with COCEPT and CLASS. While in older simulations, as those used in Euclid Collaboration: Knabenhans et al. (2019), the traditional back-scaling approach has been used for the construction of initial conditions of the N-body simulations, now we employ a novel approach taking advantage of the fully correct linear evolution of particles carried out in Einstein-Boltzmann codes (here CLASS). As a result, PKDGRAV3 recovers linear theory accurately at all redshifts even in the presence of massive neutrinos.

Moreover, we work with transfer functions in the N-body gauge instead of the more standard synchronous gauge. In this way, results computed with a purely Newtonian N-body code such as PKDGRAV3 can be interpreted within the framework of general relativity without the need of including general relativistic corrections at the N-body code level.
be applied in a post processing step. The result of applying this correction to a power spectrum measured in a $\ell^{-1} = 3 \, h \, \text{Mpc}^{-1}$-simulation is a power spectrum that approximates very closely that obtained from an equivalent simulation with $\ell^{-1} = 8 \, h \, \text{Mpc}^{-1}$ up to $k \sim 10 \, h \, \text{Mpc}^{-1}$.

We have then produced a set of 127 P+F simulations of $1 \, Gpc^3/h^3$ with 3000$^3$ particles, corresponding to a resolution of $\ell^{-1} = 3 \, h \, \text{Mpc}^{-1}$. This corresponds to a computational cost of roughly 650,000 node hours which we have invested using the Piz Daint supercomputer located at the Swiss National Scientific Supercomputing Centre (CSCS). At redshift $z = 0$ this implies that the simulations are converged at the 2% level at $k = 10 \, h \, \text{Mpc}^{-1}$ (and at 1% up to $k \sim 5 \, h \, \text{Mpc}^{-1}$). At redshift $z = 2.76$ we achieve 1% convergence up to $k \sim 2 \, h \, \text{Mpc}^{-1}$ and 2% up to $k \sim 3.5 \, h \, \text{Mpc}^{-1}$. By applying the resolution correction factor, the convergence is subsequently improved to $\sim 0.5\%$ at $k \sim 10 \, h \, \text{Mpc}^{-1}$ and $z = 0$ and to $\sim 1\%$ at $k \sim 10 \, h \, \text{Mpc}^{-1}$ and $z = 2$ at the cost of introducing an additional source of uncertainty (see Fig. 4). We leave the decision about whether or not the resolution correction should be applied to the user of EuclidEmulator2 by not including it in the training data.

The key goal of this publication was to construct an emulator which is able to quickly and accurately predict the NLC for $w_0w_a\Lambda$CDM+$\Sigma m_\nu$ cosmologies up to scales of $k \sim 10 \, h \, \text{Mpc}^{-1}$. The emulator takes inputs from within the parameter box defined in Table 2. In order to investigate the behaviour of such an emulator and its dependencies on various quantities such as training set size or number of principal components taken into account, we created a mock emulator based on HALOFIT data. We project that we can achieve a generalisation error of sub-1% inside the axis-aligned hyperellipsoid inscribed in the parameter box if we exclude a problematic region in the $(w_0, w_a)$-plane in which the first order principal component weight shows exponential behaviour. We exclude this region from the training set by ignoring all cosmologies with $w_a > 0.5$, reducing the training set to 108 training examples.

Finally, we construct the actual emulator EuclidEmulator2 based on the PKDGRAV3 simulation data containing 108 training cosmologies. We train the emulator using the MATLAB package UQLab within only 9 seconds. The projected error of below 1% up to scales of $k \sim 10 \, h \, \text{Mpc}^{-1}$ is confirmed with a small validation set. We stress that for smaller scales EuclidEmulator2 does not allow the computation of the NLC. At these scales one has to fall back on suitable alternative methods as e.g. HALOFIT. Further, EuclidEmulator2 is compared to multiple other fast predictors such as its predecessor EuclidEmulator1 (Fig. 11), HALOFIT (Fig. 13), HMCode (Fig. 14) and CosmicEmu (Fig. 15). In all comparisons the error bounds are reported on in the corresponding publications have been respected. We have also performed a Sobol’ sensitivity analysis (Fig. 8) which clearly revealed that $w_a$ is a parameter that adds considerable complexity to the underlying model while $\Sigma m_\nu$ is quasi negligible, at least for the relatively narrow range in $\Sigma m_\nu$ we have considered.

In first benchmark tests using UQLab we have measured that EuclidEmulator2 can be evaluated in $\sim 0.3$ seconds on a usual laptop. This compares well to the $\sim 0.4$ seconds per evaluation of EuclidEmulator1 using the python wrapper e2py. We reiterate here that this implies that the computation of the linear power spectrum by CAMB or CLASS is now the bottleneck in the computation of the fully nonlinear power spectrum.

We have applied EuclidEmulator2 to investigate degeneracies of the nonlinear matter power spectrum between the total neutrino mass and the other seven cosmological parameters. We have shown that tomographic surveys exploiting both linear and nonlinear scales are critical as it is possible to find different cosmologies with nonlinear

![Figure 18. Fit of a nonlinear power spectrum of $w_0w_a\Lambda$CDM+$\Sigma m_\nu$ cosmology with only massless neutrinos. The purple line in the upper panel is a fit of the reference data to better than 1% at all scales of interest. The different colours correspond to different redshifts. It is evident, that even though the “fit” cosmology approximates the reference very well at $z = 0$, the degeneracies between the neutrino mass and the other parameters are broken at different redshifts, highlighting the importance of tomographic surveys.](image_url)
matter power spectra agreeing better than 1% at $z = 0$ (in our case we have tested a massive neutrino and a massless neutrino cosmology).

Further efforts should be taken in multiple directions. While the power spectrum (and thus the NLC) clearly belong to the most used summary statistics of cosmic large-scale structure, higher-order statistics are becoming more and more used and thus emulators for their prediction are desirable. An example for such a predictor was recently published (Takahashi et al. 2019). A different, more holistic approach is taken in He et al. (2019) where the displacement field is emulated directly, such that any statistic can be derived from the predicted density field. For simulations of resolution as high as the ones used in this work, it is however questionable to what extent such an approach is practical. Further, in order to assess more deeply the accuracy of the NLC predictions at small scales, it is not only necessary to estimate the generalisation error and the convergence of the underlying simulations depending on box size and resolution but also to investigate how well different codes agree with each other at the scales under consideration. While such a study has been performed in Schneider et al. (2016) (and augmented by another code in Garrison et al. 2019) we advocate for new efforts in this direction, as with the new updates to PKDGRAV3 and developments in other codes the situation may have changed significantly.

Disregarding such uncertainties in the underlying N-body code, at this point we shall summarise the error contributions to EuclidEmulator and their dependence on spatial scales and redshift. At low redshifts, the emulation-only generalisation error is virtually zero by construction on large scales ($k < 0.01 h \text{ Mpc}^{-1}$) such that in this regime the dominant error contribution in the emulator comes from cosmic variance. Based on the results of Angulo & Pontzen (2016), the cosmic variance is expected to be sub-percent. At small scales ($k \geq 1 h \text{ Mpc}^{-1}$), cosmic variance is expected to be irrelevant. In this regime, the dominant error contribution (neglecting additional physics such as baryons) is due to emulation itself. The level of the dominant error at high $k$ is estimated to be at the $\sim 0.7\%$ level according to Fig. 9. This error is estimated from only a very small sample of validation simulations, however, the error level is also consistent with the estimate in Fig. 6 and hence we regard this error estimate to be representative at small scales. Estimating the overall error level in the intermediate $k$ range is tricky because several effects contribute errors at a similar level: on the one hand it is evident in Fig. 9 that an accurate prediction of the NLC around the BAOs is challenging (the observed accuracy is also at the level of $\sim 0.6\%$). At the same time, residual cosmic variance (after pairing-and-fixing) is nonlinearly amplified at these scales. We estimate the error in the intermediate range ($0.01 h \text{ Mpc}^{-1} < k < 1 h \text{ Mpc}^{-1}$) to be at the level of 1%. The comparison as shown in Fig. 11 suggests an error at the $2\%$ level. This may, however, be overly conservative because the cosmic variance in EuclidEmulator is phase-shifted with respect to EuclidEmulator, leading to an enhancement of errors within this comparison.

The redshift evolution does not greatly change the error contributions discussed above. However, the overall error at intermediate scales is reduced at higher redshifts compared to the low-redshift case. At small scales the resolution effects become the dominant source of error as is visible in Fig. 4. As resolution is currently not corrected in a cosmology-dependent manner, the error is expected to be at the level of 1% at small scales ($k \geq 1 h \text{ Mpc}^{-1}$) and higher redshift ($z \sim 3$).

Last but not least we have seen how the large number of dimensions of the parameter space is really starting to become a major challenge regarding the number of simulations required to arrive at the targeted generalisation error. As more and more dimensions can be expected to be added in the next couple of years, it may be of interest to also compare different emulation strategies to each other in order to potentially identify the strategy that generalises best based on only very few examples per dimension.

EuclidEmulator2 is the successor of EuclidEmulator1 and will again be published on https://github.com/miknab/EuclidEmulator2.

**GLOSSARY**

**Codes:**

- **CAMB** Code for anisotropies in the microwave background, 5, 22
- **CLASS** Cosmological linear anisotropy solving system, 3, 5, 6, 14–17, 21, 22
- **CONCEPT** Cosmological $N$-body code in Python, 5, 6, 21
- **CosmicEmu** Cosmic emulator based on the Mira-Titan cosmological simulation suite (successor of FrankenEmu based on the Coyote simulation suite), 2, 16, 19, 20, 22
- **EuclidEmulator1** Version 1 of EuclidEmulator. This is a code to predict the nonlinear corrections to DM power spectra for $w_0\omega$CDM cosmologies, 2, 5–7, 13, 15–19, 22, 23
- **EuclidEmulator2** Version 2 of EuclidEmulator. This is a code to predict the nonlinear corrections to DM power spectra for $w_0\omega$CDM+$\sum m_{\nu}$, 2–6, 8–11, 13–23, 25
- **HALOFIT** Analytical code to produce nonlinear power spectra, 2, 3, 10, 11, 13, 14, 16–20, 22, 25
- **HMCode** Fast predictor for the nonlinear power spectrum based on the halomodel approach, 2, 16, 19, 20, 22
- **NGenHalofit** Code to produce non-linear power spectra using a semi-analytical approach for large and a smoothing-spline-fit model for small scales 2
- **PKDGRAV3** Parallel k-D tree gravity code (version 3); Cosmological N-body tree code 3, 5–7, 16, 17, 21–23
- **UQLab** Matlab-based uncertainty quantification framework, 11, 22
- **e2py** Python wrapper for EuclidEmulator, 22

**Acronyms:**

- **1LPT** First order Lagrangian perturbation theory, 5, 7
- **2LPT** Second order Lagrangian perturbation theory 7
- **BAO** Baryon acoustic oscillations, 17–20, 23
- **CDM** Cold dark matter, 3, 5, 6, 10, 18, 20–22
- **CMB** Cosmic microwave background, 2, 10
- **DE** Dark energy, 2–6, 10, 12, 16, 19, 21
- **DM** Dark matter, 2–6, 21
- **ED** Experimental design, 8, 15
- **EFS2** Euclid Flagship v2.0 Simulation, 7, 17, 21, 26
- **EqS** Equation of state, 4, 10, 12, 16
- **FMM** Fast multi-pole method, 5, 6
- **GR** General theory of relativity, 4, 5
- **GRF** Gaussian random field, 7, 26

**MNRAS 000, 000–000 (2019)**
REFERENCES
Ahmed S. N., Anthony A. E., Beier E. W., et al. 2004, Physical Review Letters, 92, 181301
Akebono R., Armas L., Bachelet E., et al. 2019, arXiv pre-print (arXiv:1902.05569)
Angulo R. E., Pontzen A., 2016, Monthly Notices of the Royal Astronomical Society: Letters, 462, L1
Angulo R. E., Zennaro M., Contreras S., Aricó G., Pellejero-Ibañez M., Stucker J., 2020, arXiv pre-print (arXiv:2004.06245)
Audren B., Lesgourgues J., Bird S., et al. 2013, Journal of Cosmology and Astroparticle Physics, 026, 23
Banerjee A., Powell D., Abel T., Villaescusa-Navarro F., 2018, Journal of Cosmology and Astroparticle Physics, 2018
Becker-Szendy R., Bratton C. B., Casper D., et al. 1992, Physical Review D, 46, 3720
Betoule M., Kessler R., Guy J., et al. 2014, Astronomy and Astrophysics, 568, 32
Bird S., Viel M., Haehnelt M. G., 2012, Monthly Notices of the Royal Astronomical Society, 420, 2551
Bird S., Ali-Haimoud Y., Feng Y., Liu J., 2018, MNRAS, 481, 1486
Blatman G., 2009, PhD thesis, Université de Clermont-Ferrand
Blatman G., Sudret B., 2009, in 19ème Congrès Français de Mécanique. AFM, Maison de la Mécanique, 39/41 rue Louis Blanc - 92400 Courbevoie, Marseille, pp 01–06, http://hdl.handle.net/2042/37252
Blatman G., Sudret B., 2010, Probabilistic Engineering Mechanics, 25, 183
Blatman G., Sudret B., 2011, Journal of Computational Physics, 230, 2345
Blatman G., Sudret B., 2013, in Deodatis G., Ellingwood B., Frangopol D., eds, Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures - Proceedings of the 11th International Conference on Structural Safety and Reliability, ICOSAR 2013. CRC Press (Taylor & Francis Group), London, pp 3245–3252, doi:10.1201/b16387-469
Chisari N. E., Zaldarriaga M., 2011, Physical Review D, 83, 123505
Copeland D., Taylor A., Hall A., 2020, MNRAS, 493, 1640
Coulton W. R., Liu J., Madhavacheril M. S., Böhm V., Spergel D. N., 2019, Journal of Cosmology and Astroparticle Physics, 2019, 043
Crombècq K., Laermans E., Dhane T., 2011, European Journal of Operational Research, 214, 683
DESI Collaboration: Aghamousa A., Aguilar J., Ahlen S., et al. 2016, arXiv pre-print (arXiv:1611.00036)
DUNE Collaboration: Acciarri R., Acero M., Adamowski M., et al. 2015, arXiv pre-print (arXiv:1512.06148)
Dakin J., Hannestad S., Tram T., 2019a, Journal of Cosmology and Astroparticle Physics, 06, 1
Dakin J., Hannestad S., Tram T., et al. 2019b, Journal of Cosmology and Astroparticle Physics, 08, 1
Damblin B., Couplet M., Iooss B., 2013, Journal of Simulation, 7, 276
Dawson K. S., Schlegel D. J., Ahn C. P., et al. 2013, Astronomical Journal, 145, 41
DeRose J., Wechsler R. H., Tinker J. L., et al. 2019, The Astrophysical Journal, 875, 69
Efron B., Hastie T., Johnstone I., et al. 2004, The Annals of Statistics, 32, 407
Euclid Collaboration: Knaebelhans M., Stadel J. G., Marelli S., et al. 2019, Monthly Notices of the Royal Astronomical Society, 484, 5509
Fang W., Hu W., Lewis A., 2008, Physical Review D, 78, 4
Fidler C., Tram T., Rampf C., et al. 2016, Journal of Cosmology and Astroparticle Physics, 2016, 031
Fidler C., Tram T., Rampf C., et al. 2017, Journal of Cosmology and Astroparticle Physics, 2017, 043
Fraenkel F., 2008, Journal of Physics: Conference Series, 120, 052028
Fukuda Y., Hayakawa T., Ichihara E., et al. 1998a, Physical Review Letters, 81, 1158
Fukuda Y., Hayakawa T., Ichihara E., et al. 1998b, Physical Review Letters, 81, 1562
Garg V. V., Stogner R. H., 2017, Journal of the American Statistical Association, 112, 673
Garrison L. H., Eisenstein D. J., Pinto P. A., 2019, Monthly Notices of the Royal Astronomical Society, 485, 3370
Giblin B., Cataneo M., Moews B., et al. 2019, Monthly Notices of the Royal Astronomical Society, 490, 4828
Gratton A., Wilkinson M. I., 2019, Monthly Notices of the Royal Astronomical Society, 485, 4878
Hannestad S., Haugbølle T., C.S., 2012, Journal of Cosmology and Astroparticle Physics, 2012, 1
Hannestad S., Upadhye A., Wong Y., 2020, arXiv pre-print (arXiv:2006.04995)
He S., Li Y., Feng Y., et al. 2019, Proceedings of the National Academy of Sciences of the United States of America, 116, 13825
Heitmann K., Higdon D., White M., et al. 2009, The Astrophysical Journal, 705, 156
Heitmann K., White M., Wagner C., et al. 2010, The Astrophysical Journal, 715, 104
Heitmann K., Lawrence E., Kwan J., et al. 2014, The Astrophysical Journal, 780, 111
Heitmann K., Bingham D., Lawrence E., et al. 2016, The Astrophysical Journal, 820, 108
Hildebrandt H., Viola M., Heymans C., et al. 2017, Monthly Notices of the Royal Astronomical Society, 465, 1454
Hu W., Sawicki I., 2007, Physical Review D - Particles, Fields, Gravitation
and Cosmology, 76, 104043.
Ichiki K., Takada M., Takahashi T., 2009, Phys. Rev. D, 79
LSST Science Collaboration: Abell P. A., Allison J., Anderson S. F., et al. 2009, arXiv pre-print (arXiv:0912.0201)
Laureijs R., Amiaux J., Arduini S., et al. 2011, arXiv pre-print (arXiv:1110.3193)
Lawrence E., Heitmann K., White M., et al. 2010, The Astrophysical Journal, 713, 1322
Lawrence E., Heitmann K., Kwan J., et al. 2017, The Astrophysical Journal, 847, 12
Liefvendahl M., Stocki R., 2006, Journal of Statistical Planning and Inference, 136, 3231
Marelli S., Sudret B., 2014, in Vulnerability, Uncertainty, and Risk (Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool, United Kingdom. American Society of Civil Engineers, pp 2554–2563, doi:10.1061/9780784413609.257
Marelli S., Sudret B., 2017, Technical report, UQLab user manual - Polynomial Chaos Expansion. Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Zurich, doi:10.13140/RG.2.1.3778.7366
Marelli S., Sudret B., 2018, Structural Safety, 75, 67
Marelli S., Lamas C., Sudret B., 2017, Technical report, UQLab user manual - Sensitivity analysis. Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich, Zurich, doi:10.13140/RG.2.1.1419.4404
Massara E., Villaescusa-Navarro F., Viel M., 2014, Journal of Cosmology and Astroparticle Physics, 12, 053
McClintock T., Rozeo E., Becker M. R., et al. 2019, The Astrophysical Journal, 872, 53
McKay M. D., Beckmann R. J., Conover W. J., 1979, Technometrics, 21, 239
Mead A. J., Heymans C., Lombriser L., et al. 2016, Monthly Notices of the Royal Astronomical Society, 459, 1468
Nishimichi T., Takada M., Takahashi R., et al. 2019, The Astrophysical Journal, 874, 129
Particle Data Group: Tanabashi M., Hagiwara K., Hikasa K., et al. 2018, Chinese Physics C, 40, 1
Particle Data Group: Tanabashi M., Hagiwara K., Hikasa K., et al. 2016, Chinese Physics C, 486, 1448
Potter D., Stadel J. G., 2016, PKDGRAV3: Parallel gravity code, http://ascl.net/1609.016
Potter D., Stadel J. G., Teissier R., 2017, Computational Astrophysics and Cosmology, 4, 2
Rogers K. K., Peiris H. V., Pontzen A., et al. 2019, Journal of Cosmology and Astroparticle Physics, 02, 1
Schneider A., Teissier R., Potter D., et al. 2016, Journal of Cosmology and Astroparticle Physics, 04, 047
Schneider A., Stora N., Refregier A., et al. 2019a, arXiv pre-print (arXiv:1910.11537)
Schneider A., Refregier A., Grandis S., et al. 2019b, arXiv pre-print (arXiv:1911.08494)
Schwetz T., Tórtola M., Valle J. W. F., et al. 2008, New Journal of Physics, 10, 10
Sheikhholeslami R., Razavi S., 2017, Environmental Modelling and Software, 93, 109
Smith R. E., Angulo R. E., 2019, Monthly Notices of the Royal Astronomical Society, 486, 1448
Stadel J. G., 2001, PhD thesis, University of Washington, https://search.proquest.com/docview/251727174
Swiler L. P., Sloype R., Giunta A., 2006, in 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Newport, Rhode Island, pp 1–24, doi:10.2514/6.2006-1827
Takahashi R., Sato M., Nishimichi T., et al. 2012, The Astrophysical Journal, 761, 152
Takahashi R., Nishimichi T., Namikawa T., et al. 2019, arXiv pre-print (arXiv:1911.07886)
Tang B., 1993, Journal of the American Statistical Association, 88, 1392
Torre E., Marelli S., Embrechts P., et al. 2019, Journal of Computational Physics, 388, 601
Tram T., Brandbyge J., Dakin J., et al. 2019, Journal of Cosmology and Astroparticle Physics, 03, 1
Valcin D., Villaescusa-Navarro F., Verde L., et al. 2019, Journal of Cosmology and Astroparticle Physics, 2019, 057
Valle J. W. F., 2005, AIP Conference Proceedings, 805, 128
Viel M., Haehnelt M. G., Springel V., 2010, Journal of Cosmology and Astroparticle Physics, 06, 15
Wang G. G., 2003, Journal of Mechanical Design, 125, 210
Weinheimer C., 2002, Progress in Particle and Nuclear Physics, 48, 141
Winther H. A., Casas S., Baldi M., et al. 2019, Physical Review D, 100, 1
Wolf J., 2010, in Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. pp 442–444, doi:10.1016/j.nima.2010.03.030
Xiu D., Karniadakis G. E., 2002, SIAM Journal on Scientific Computing, 24, 619
Yang M., Liu Z., Li W., 2017, Journal of Statistical Computation and Simulation, 87, 3398
Zhai Z., Tinker J. L., Becker M. R., et al. 2019, The Astrophysical Journal, 874, 12

APPENDIX A: SIMULATION TABLE
Here we summarize all simulations that we have produced specifically for this paper. For each simulation, its unique ID as well as its specifications are listed. The specifications consist of the box size (L), the number of particles per side length (N), whether it is a P+P run (PF yes/no), what order of Lagrangian perturbation theory (LPT) was used to construct the initial conditions, the number of runs, the run time in node hours and on what machine the simulation was executed. Simulations T001 to T127 are the runs that form the actual training set of EuclidEmulator2 while HRV001–HRV003 were used for the end-to-end test reported in section 8.2.1. The runs VCT1–VT5 were used for the volume convergence test and RCT1–RCT5 for the resolution convergence test (see Fig. 2 and Fig. 3).
We used the RES3 and RES8 simulations in order to estimate the variance of the cosmology dependence in the resolution correction factor (see Fig. 4). The PF simulation was used in the comparison to the simulations GRF1–GRF50 in order to investigate the cosmic sampling variance in P+P simulations (see Fig. 12). The PV runs were used to estimate the output variance on both boost factor and power spectrum level when one of the parameters $\alpha_s$, $\omega_w$ or $\sum m_v$ is varied based on which the parameter box of EuclidEmulator2 was chosen. The total run time for all simulations sums up to over 700 000 node hours.

APPENDIX B: ERROR MAPS OF THE HALOFIT-BASED MOCK EMULATOR
In this appendix we plot error maps for a two exemplary coordinate planes of the 8D parameter box: the $(\sum m_v, h)$- and the $(\omega_0, \omega_a)$-plane. The errors are defined as follows:

$$\epsilon = \max_{k \in [0.01, 1.0]} \frac{|B_{\text{emu}}(k, z = 0) - B_{\text{sim}}(k, z = 0)|}{B_{\text{sim}}(k, z = 0)}$$

B1

The emulator for this investigation was trained with HALOFIT based on exactly the same 108 cosmologies that were used for the actual, simulation-based EuclidEmulator2. We stress that the errors all are measured at $z = 0$. The colour bars are ranging from 0% to 8% for both plot panels. The hyperellipsoid inscribed in the parameter box is shown. Notice that the vast majority of cosmologies inside
Table A1. Simulations used for this publication. The table is organized as follows: The training simulations are in the first row followed by the validation simulations in the second and third row. In rows 4 to 13 we list all simulations used in the volume convergence tests (VCT) and the resolution convergence tests (RCT), respectively. Then, in rows 14 and 15 we list the simulations used to investigate the RCF dependence on cosmology. In rows 16 and 17 the simulations used to compare the P+F approach to the traditional ensemble averaging of GRF simulations are listed. Ultimately, we mention the simulations used to estimate the parameter ranges in the rows 18-21.

| Simulation identifier | \(L\) [\(h^{-1}\) Mpc] | \(N\) | PF | LPT | number of runs | total runtime [node hours] | machine |
|-----------------------|--------------------------|-----|----|-----|----------------|-----------------------------|---------|
| T001-T127             | 1000                     | 3000| yes| 1LPT| 254            | \(~500\,000\)                | Piz Daint (*) |
| EFS2                  | 3600                     | 16000| no| 1LPT| 1              | \(~1\,000\,000\)             | Piz Daint (●) |
| HRV001-HRV003         | 1000                     | 3000| yes| 1LPT| 6              | \(~37\,000\)                 | zBox4+ |
| VCT1                  | 512                      | 170 | yes| 1LPT| 2              | 10                          | zBox4+ |
| VCT2                  | 1024                     | 342 | yes| 1LPT| 2              | 24                          | zBox4+ |
| VCT3                  | 2048                     | 682 | yes| 1LPT| 2              | 118                         | zBox4+ |
| VCT4                  | 4096                     | 1356| yes| 1LPT| 2              | 435                         | zBox4+ |
| VCT5                  | 8192                     | 2730| yes| 1LPT| 2              | 2780                        | zBox4+ |
| RCT1                  | 512                      | 512 | no | 1LPT| 1              | 37                          | zBox4+ |
| RCT2                  | 512                      | 1024| no | 1LPT| 1              | 212                         | zBox4+ |
| RCT3                  | 512                      | 1536| no | 1LPT| 1              | 919                         | zBox4+ |
| RCT4                  | 512                      | 2048| no | 1LPT| 1              | 1987                        | zBox4+ |
| RCT5                  | 512                      | 4046| no | 1LPT| 1              | 10,353                      | zBox4+ |
| RES3_1-RES3_20        | 128                      | 384 | yes| 1LPT| 40             | \(~320\)                     | zBox4+ |
| RES8_1-RES8_20        | 128                      | 1024| yes| 1LPT| 40             | \(~4800\)                   | zBox4+ |
| PF                    | 1024                     | 980 | yes| 1LPT| 2              | 69                          | zBox4+ |
| GRF1-GRF50            | 1024                     | 980 | no | 1LPT| 50             | \(~3400\)                   | zBox4+ |
| PV\(_{\alpha}\)       | 640                      | 1024| no | 2LPT| 6              | \(~1100\)                   | zBox4+ |
| PV\(_{\mu}\)         | 640                      | 1024| no | 2LPT| 6              | \(~1100\)                   | zBox4+ |
| PV\(_{\mu}\)         | 640                      | 1024| no | 2LPT| 6              | \(~670\)                    | zBox4+ |
| PV\(_{\text{center}}\) | 640                     | 1024| no | 2LPT| 1              | 77                          | zBox4+ |

* with GPUs

**Affiliations**

1 Institute for Computational Science, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland
2 Department of Physics and Astronomy, University of Aarhus, Ny Munkegade 120, DK-8000 Aarhus C, Denmark
3 Chair of Risk, Safety and Uncertainty Quantification, Dept. of Civil Engineering, ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland
4 Institute for Particle Physics and Astrophysics, Dept. of Physics, ETH Zurich, Wolfgang-Pauli-Strasse 27, 8093 Zurich, Switzerland
5 INAF-Osservatorio Astronomico di Brera, Via Brera 28, I-20122 Milano, Italy
6 INAF-Osservatorio di Astrofisica e Scienza dello Spazio di Bologna, Via Piero Gobetti 93/3, I-40129 Bologna, Italy
7 SISSA, International School for Advanced Studies, Via Bonomea 265, I-34136 Trieste TS, Italy
8 INFN, Sezione di Trieste, Via Valerio 2, I-34127 Trieste TS, Italy
9 INAF-Osservatorio Astronomico di Trieste, Via G. B. Tiepolo 11, I-34131 Trieste, Italy
10 Universidad de la Laguna, E-38206, San Cristóbal de La Laguna, Tenerife, Spain
11 Instituto de Astrofísica de Canarias, Calle Vía Láctea s/n, 38204, San Cristóbal de La Laguna, Tenerife, Spain
12 Istituto Nazionale di Astrofisica (INAF) - Osservatorio di Astrofisica e Scienza dello Spazio (OAS), Via Gobetti 93/3, I-40127 Bologna, Italy
13 Dipartimento di Fisica e Astronomia, Università di Bologna, Via Gobetti 93/2, I-40129 Bologna, Italy
14 INFN-Sezione di Bologna, Viale Berti Pichat 6/2, I-40127 Bologna, Italy
15 INAF-IASF Bologna, Via Piero Gobetti 101, I-40129 Bologna, Italy
16 Universität-Erwin-Meitner-München, Fakultät Physik, Ludwig-Maximilians-Universität München, Scheinerstrasse 1, 81679 München, Germany
17 Max Planck Institute for Extraterrestrial Physics, Giessenbachstr. 1, D-85748 Garching, Germany
18 IFPU, Institute for Fundamental Physics of the Universe, via Beirut 2, 34151 Trieste, Italy
19 Department of Astronomy, University of Geneva, ch. d’Écogia 16, CH-1290 Versoix, Switzerland
20 INFN-Sezione di Roma Tre, Via della Vasca Navale 84, I-00146, Roma, Italy
21 Department of Mathematics and Physics, Roma Tre University, Via della Vasca Navale 84, I-00146 Rome, Italy
22 INAF-Osservatorio Astronomico di Roma, Via Frascati 33, I-00078 Monteporzio Catone, Italy
23 INAF-Osservatorio Astronomico di Capodimonte, Via Moiariello
Figure B1. Error map of the $(\Sigma m_V, h)$-plane (top) and the $(w_0, \omega_m)$-plane (bottom). These two error maps represent also those of the remaining 26 parameter planes. Most error maps feature only very low errors like the top panel (all errors in the top panel are $\lesssim 1\%$) in this figure.
