

PyBird-JAX: Accelerated inference in large-scale structure with model-independent emulation of one-loop galaxy power spectra

Alexander Reeves^{id},^a Pierre Zhang^{id}^{a,b,c} and Henry Zheng^{id}^{d,e}

^a*Institute for Particle Physics and Astrophysics, ETH Zürich, 8093 Zürich, Switzerland*

^b*Institut für Theoretische Physik, ETH Zürich, 8093 Zürich, Switzerland*

^c*Dipartimento di Fisica “Aldo Pontremoli”, Università degli Studi di Milano, 20133 Milan, Italy*

^d*Stanford Institute for Theoretical Physics, Physics Department, Stanford University, Stanford, CA 94306, U.S.A.*

^e*Kavli Institute for Particle Astrophysics and Cosmology, SLAC and Stanford University, Menlo Park, CA 94025, U.S.A.*

E-mail: areeves@phys.ethz.ch, pierre.zhang@unimi.it

ABSTRACT: We present PyBird-JAX, a differentiable, JAX-based implementation of PyBird, using internal neural network emulators to accelerate computationally costly operations for rapid large-scale structure (LSS) analysis. PyBird-JAX computes one-loop EFTofLSS predictions for redshift-space galaxy power spectrum multipoles in 1.2 ms on a CPU and 0.2 ms on a GPU, achieving 3–4 orders of magnitude speed-up over PyBird. The emulators take a compact spline-based representation of the input linear power spectrum $P(k)$ as feature vectors, making the approach applicable to a wide range of cosmological models. We rigorously validate its accuracy against large-volume simulations and on BOSS data, including cosmologies not explicitly represented in the training set. Leveraging automatic differentiation, PyBird-JAX supports Fisher forecasting, Taylor expansion of model predictions, and gradient-based searches. Interfaced with a variety of samplers and Boltzmann solvers, PyBird-JAX provides a high-performance, end-to-end inference pipeline. Combined with a symbolic- $P(k)$ generator, a typical Stage-4 LSS MCMC converges in minutes on a GPU. Our results demonstrate that PyBird-JAX delivers the precision and speed required for upcoming LSS surveys, opening the door to accelerated cosmological inference with minimal accuracy loss and no pretraining. In a companion paper [1], we put PyBird-JAX to use in achieving LSS marginalised constraints free from volume projection effects through non-flat measures.

KEYWORDS: cosmological parameters from LSS, galaxy clustering, Machine learning

ARXIV EPRINT: [2507.20990](https://arxiv.org/abs/2507.20990)

Contents

1	Introduction	1
2	Model-independent emulation of one-loop galaxy power spectra	3
2.1	Model-independent NN-based emulator	4
2.2	Performance	10
3	PyBird-JAX: boosted cosmological inference in LSS	13
3.1	JAX-powered features	14
3.2	AD-powered features	15
4	Cosmological applications	17
4.1	PT challenge simulations	20
4.2	BOSS data	20
4.3	Stage-4 LSS survey forecast	24
5	Conclusions	27
A	Vectorised FFTLog-based IR-resummation	28
B	Emulator accuracy up to $k_{\max} = 0.3 h \text{ Mpc}^{-1}$	30

1 Introduction

Cosmology has entered an era of precision science. The increasing data volume gathered by wide-field instruments demands better accuracy in the modeling of the large-scale structure (LSS). In order to access information residing beyond linear scales, it is necessary to introduce progressively more nuisance parameters to marginalise over our ignorance of nonlinearities and astrophysical processes. At the same time, the multitude and diversity of cosmological probes greatly increase the dimension of the parameter space to explore when performing combined probes analyses. In the context of spectroscopic experiments, full-shape analyses introduce $\sim \mathcal{O}(10 - 40)$ nuisance parameters for the power spectrum or bispectrum modeled at the one-loop level from the Effective Field Theory of Large-Scale Structure (EFTofLSS) [2–5]. A quick estimate for the next observables in this program, i.e., two-loop power spectrum and one-loop trispectrum, yields up to $\sim \mathcal{O}(100)$ nuisance parameters considering both the bias expansion [6, 7] and extra counterterms in redshift space [4]. A clustering analysis with $\sim \mathcal{O}(10)$ tracers/redshift targets from e.g., DESI [8] and Euclid [9] naively requires scanning over $\sim \mathcal{O}(10^3)$ parameters, depending on the observables considered. Current N -point function analyses already marginalise over up to $\sim \mathcal{O}(100)$, depending on the refinements in the predictions and the setup considered (see e.g., refs. [5, 10–15]). Exploring parameter spaces with a large number of dimensions can be prohibitively time and resource consuming with traditional sampling methods. Where possible, analytical

marginalisation can significantly reduce the dimensionality of the problem [10, 13, 16]. More generally, recent years have seen the advent of new computational techniques particularly well suited to tackle this problem. For example, emulators, designed to interpolate smooth, cosmological predictions, can significantly speed up the likelihood call with controlled accuracy loss. Furthermore, differentiable likelihoods enable efficient exploration of high-dimensional parameter space using gradient-informed techniques. Straightforward solutions that capture both of these benefits consist in Taylor-expanding the cosmological observables around a fiducial point [17–19], training neural networks (NNs), Gaussian processes, or other methods, to emulate them (see e.g., refs. [20–32]), or leveraging modern coding languages such as JAX¹ or Julia² (see e.g., refs. [33–39]).

In this paper, we present an updated version of `PyBird`: the Python code for Biased tracers in redshift space [40]. `PyBird` provides nonlinear predictions and likelihoods for cosmological correlators based on the EFTofLSS. As one of the earliest implementations of this kind, extensively tested against multiple N -body simulations with diverse halo-galaxy connection models [13, 19, 41–45], `PyBird` has been benchmarked against other pipelines [46], particularly within the contexts of DESI [19, 47] and Euclid [48]. `PyBird` has been used to put constraints on a wide variety of physics (see e.g., refs. [40, 49–68]) from various LSS datasets [13, 44, 45], including the first results from the full shape of DESI first data release [14, 15]. A number of alternative codes for computing the one-loop power spectrum have also been developed [69–75], further reflecting the community’s active interest in this area. While these codes agree in their core calculations, each offers distinct features and implementation details. We encourage readers to explore these complementary tools.

In this work, we revisit the implementation of `PyBird` in several ways, resulting in a fully differentiable pipeline with significantly boosted performance. While keeping compatibility with most cosmological codes, Boltzmann solvers, numerical samplers, etc., within the rich Python environment, we upgrade `PyBird` to JAX. Our new JAX implementation, that we dub `PyBird-JAX`, allows for just-in-time compilation to dedicated computing hardware, vectorisation, and automatic differentiation (AD). Furthermore, we accelerate the slowest parts of `PyBird-JAX` by emulating their computation with NN-based emulators. We will refer to the latter scenario as `PyBird-Emu`. Leveraging the fact that `PyBird` takes as input the linear matter power spectrum which can be readily decomposed onto a compact set of basis functions, we train NNs only on the decomposition coefficients, rendering our emulator explicitly *cosmology-independent*. This strategy contrasts with existing `PyBird` emulators and similar codes (see, e.g., [76–81]), which require retraining for every cosmological model and usually demand additional accuracy checks against full, non-emulated analyses (see however ref. [32] for an on-the-fly efficient emulation strategy during sampling, though applied on different observables). At present, the emulation is carried on the redshift-space one-loop galaxy power spectrum, however our method is general and can apply to other observables such as higher-loop corrections and higher- N point functions, as well as multi-tracer statistics. Overall, our improvements result in massive acceleration approximately three (four) orders of

¹<https://docs.jax.dev/>.

²<https://julialang.org/>.

magnitude on CPU (GPU) units, making one evaluation with `PyBird-Emu` as fast as 1.2 ms (0.2 ms). Numerical accuracy on predictions are found below 0.1σ relative to a representative Stage-4 spectroscopic survey volume of $V_{\text{eff}} = 50 \text{ Gpc}^3$, for 95% of the distribution of an independent validation set. This makes `PyBird-JAX` sufficiently accurate for galaxy clustering analyses in upcoming surveys over the next decade. `PyBird-JAX` is publicly released in the same repository of `PyBird`.³ `PyBird-JAX` can be activated via a simple backend switch on top of `PyBird` as core implementation are shared, with the difference being the underlying libraries—`JAX` or `Python`— called during execution.

Our paper is organised as follows. In section 2, we present a general, emulator-based, model-independent method for LSS observables, which we implement to accelerate the evaluation of `PyBird` one-loop galaxy power spectra. Section 3 summarises the new features introduced in `PyBird-JAX`, with a focus on AD and enhanced compatibility with various Boltzmann solvers and samplers. In section 4, we benchmark the fully upgraded pipeline on challenging cosmological analyses, including simulations, BOSS data, and a Stage-4 LSS spectroscopic mock survey. We conclude in section 5. Additional technical details on the implementation of IR-resummation in `PyBird` are provided in appendix A, while further accuracy assessments of the emulator are presented in appendix B.

As an important application, our companion paper [1] leverages the differentiability of `PyBird-JAX` to construct robust estimators for parameter inference. We show that, with properly defined expectation values, the posterior means of the parameters of interest can be accurately recovered — without significant bias from volume projection effects — even when marginalizing over a large number of nuisance parameters. By offering practical tools and theoretical insights in parameter inference, this two-paper series enables efficient and robust cosmological analysis from the LSS in the new era of precision cosmology.

2 Model-independent emulation of one-loop galaxy power spectra

In this section, we present our methodology for implementing model-agnostic emulators in `PyBird`. The most expensive steps of the evaluation of a given galaxy power spectrum using `PyBird` are the loop integrals and the IR-resummation as described in ref. [65] (see also ref. [42]). As we show in section 2.2, whilst implementing these routines in `JAX` and using just-in-time compilation already gives a speed-up of a factor of $\mathcal{O}(10)$, we can gain a further speed boost by a factor of $\mathcal{O}(100)$ by emulating these components with `JAX`-based NNs. Schematically, the galaxy power spectrum in redshift space $P_{g,r}$ reads

$$P_{g,r}(k, \mu, z) = \sum_{i=0}^N \sum_{j=0}^N b_i b_j \mu^{r_{ij}} f(z)^{s_{ij}} P^{ij}(k, z), \quad (2.1)$$

where b_i , $i = 1, \dots, N$, are the EFT parameters describing the one-loop power spectrum in redshift space in the EFTofLSS (see section 4 for details), $b_0 \equiv 1$, μ is the cosine of angle of \mathbf{k} with the line-of-sight, and r_{ij}, s_{ij} are some associated integer powers that are non-zero for line-of-sight-dependent terms arising in redshift space. Here $P^{ij} \equiv P^{ij}[P_{\text{lin}}]$ are scale-dependent functions that generically depend on P_{lin} , the input linear matter power spectrum,

³See <https://github.com/pierrexyz/pybird> and <https://pierrexyz.github.io/pybird/>.

which encodes their cosmology dependence. In appendix A, we show how the IR-resummation scheme used in PyBird can also be written in a functional form akin to eq. (2.1). The most expensive pieces correspond to loop integrals of the 22-diagram type (see eq. (2.7)).

We train an emulator for the loop integrals and the IR-resummation and use an input parameter space which represents a decomposition of the linear matter power spectrum as described in the next section. This approach is independent of specific cosmologies and redshifts, and can thus be used for any cosmological exploration and survey specification. This means the emulators will not have to be retrained when exploring a new specific model or survey set up. Notice that, in principle, this strategy could also be used to emulate all parts of the galaxy power spectrum. However, the linear galaxy power spectrum and EFT counterterms are trivially dependent on P_{lin} and hence do not need to be emulated as their evaluation is already rapid. By keeping these pieces computed using the analytic expression, we limit the impact of numerical inaccuracy in the emulation to only the loop and IR-resummation contributions. Since these pieces represent only a fraction of the total signal of $\sim \mathcal{O}(10\%)$, achieving $\sim 1\%$ relative accuracy in the emulation leads to a only a $\sim 0.1\%$ relative error in the total galaxy power spectrum. We now detail our procedure to find an optimised decomposition of the input linear power spectrum for the input to the NN emulators.

2.1 Model-independent NN-based emulator

Optimised decomposition of linear power spectrum. A desired parametrisation for the linear matter power spectrum has the following properties. First, it should be general enough to encompass a variety of shapes and features (arising from the diversity of cosmological considerations). Second, the number of parameters should be reasonably small enough such that the NN can accurately interpolate between across the input parameter space without requiring a prohibitively large number of training samples. To meet these constraints, we choose to interpolate $\log P_{\text{lin}}(k)$ with a linear spline on predefined k -knots in $\log k$ space. To reduce dynamic range, we normalise P_{lin} at its maximum to 1 by dividing by the maximum value of the power spectrum, P_{max} . We optimise (through Monte Carlo sampling) the knot placements on the CosmoRef bank detailed below and in table 3. Explicitly, we minimise the residual differences between normalised linear power spectra and the corresponding recovered spline interpolation weighted by the inverse of a Gaussian covariance to mimic data sensitivity together with a $(k/k_{\text{NL}})^2$ scaling as a proxy of the loop k -dependence and size. We use a total of 80 k -knots spaced between $(k_{\text{min}}, k_{\text{max}}) = (0.0001, 0.7) h \text{ Mpc}^{-1}$ which represents a generous window in k -space over which cosmological data are sensitive. Outside of this range, $\log P_{\text{lin}}(k)$ is extrapolated linearly in $\log k$. Here we take advantage of the smallness of the loop and IR-resummation at low- k and their relatively small sensitivity (at all k 's) to the IR-part of P_{lin} . Furthermore, the UV-part of P_{lin} , whilst changing the values of the loop integrals, will in fact not matter in the full prediction. This follows the fact that the UV-sensitivity of the loop integrals are absorbed by the counterterms, assuming that the actual values of the latter do not matter for the user. Would the actual values of the counterterms matter (as when one wishes to impose informative prior), PyBird includes a

Parameter	Prior (Gaussian \times flat bound)
$\ln(10^{10} A_s)$	$\mathcal{N}(3.04, 0.5) \times [2, 4]$
h	$\mathcal{N}(0.68, 0.10) \times [0.40, 1.10]$
Ω_m	$\mathcal{N}(0.310, 0.05) \times [0.20, 0.40]$
n_s	$\mathcal{N}(0.965, 0.10) \times [0.70, 1.20]$
ω_b	$\mathcal{N}(0.0223, 0.01) \times [0.010, 0.030]$
$\sum m_\nu$ [eV]	$\mathcal{N}(0.10, 0.50) \times [0, 1.5]$
ΔN_{eff}	$\mathcal{N}(0, 1.0) \times [-2, 2]$
Ω_k	$\mathcal{N}(0, 0.10) \times [-0.3, 0.3]$
w_0	$\mathcal{N}(-1.0, 0.50) \times [-2, 0]$
w_a	$\mathcal{N}(0, 1.0) \times [-2, 2]$
z	LogNormal(0.8, 0.14) \times [0.005, 4.0]

Table 1. The CosmoRef bank — Prior ranges used to construct a bank of linear matter power spectra guiding the design of the input coverage for our model-independent emulator. Cosmological parameters are sampled from wide Gaussian distributions centred on values close to the *Planck* Λ CDM preferred values (and nominal standard values for beyond- Λ CDM parameters). The Gaussian widths are set to approximately two to three times the 1σ uncertainties from the BOSS full-shape mock analysis described in section 4.2, with the exception of h , for which we consider a larger width given current debate around its value. For the baryon abundance ω_b , we consider a width of ~ 25 times the BBN prior instead. Gaussians are truncated at roughly 5σ or more relative to BOSS errors. Redshifts z are sampled from the clipped log-normal distribution described in the text. The CosmoRef bank used to design the Gaussian copula distribution over the emulator’s input space comprises 10^6 linear matter power spectrum computed via the Boltzmann solver CLASS, drawn from the above prior within a latin hypercube.

numerical method to control the UV matching explicitly by computing the contributions to the counterterms from the UV limit of the loop integrals.⁴

To test the spline decomposition we use a bank of 10^5 linear matter power spectra in cosmologies drawn from a 11-dimensional cosmological latin hypercube, dubbed the CosmoRef bank. One dimension consists in a truncated log-normal draw for the redshifts which is clipped between $z \in [0.005, 4]$ and the log-normal distribution is chosen to peak at $z = 0.6$. This choice is motivated by the fact that the linear matter power spectrum scales

⁴Explicitly, focusing on the 13-loop in real space for definiteness, the UV limit of the loop kernels K_{13} expanded up to the relevant order for our computation is

$$\lim_{\frac{q}{k} \rightarrow \infty} K_{13}(q, k) = \alpha_0 \left(\frac{k}{q}\right)^0 + \alpha_2 \left(\frac{k}{q}\right)^2 + \dots,$$

where $\alpha_0, \alpha_2, \dots$, are rational numbers. PyBird then adds, for instance, a contribution in the form of the counterterm,

$$\Delta P_{13}(k) = 4\pi \alpha_2 k^2 P_{\text{lin}}(k) \int_0^\infty dq \Delta P_{\text{lin}}(q),$$

where $\Delta P_{\text{lin}}(k) = P_{\text{lin}}^{\text{ref}}(k) - P_{\text{lin}}(k)$, with $P_{\text{lin}}(k) \approx P_{\text{lin}}^{\text{ref}}(k)$ for $k \leq k_{\text{max}} = 0.7 h \text{ Mpc}^{-1}$. Consisting in a 1D integral of the UV limit of the loop kernel times the difference between the log-interpolated P_{lin} with a specified reference $P_{\text{lin}}^{\text{ref}}$, this evaluation is relatively inexpensive.

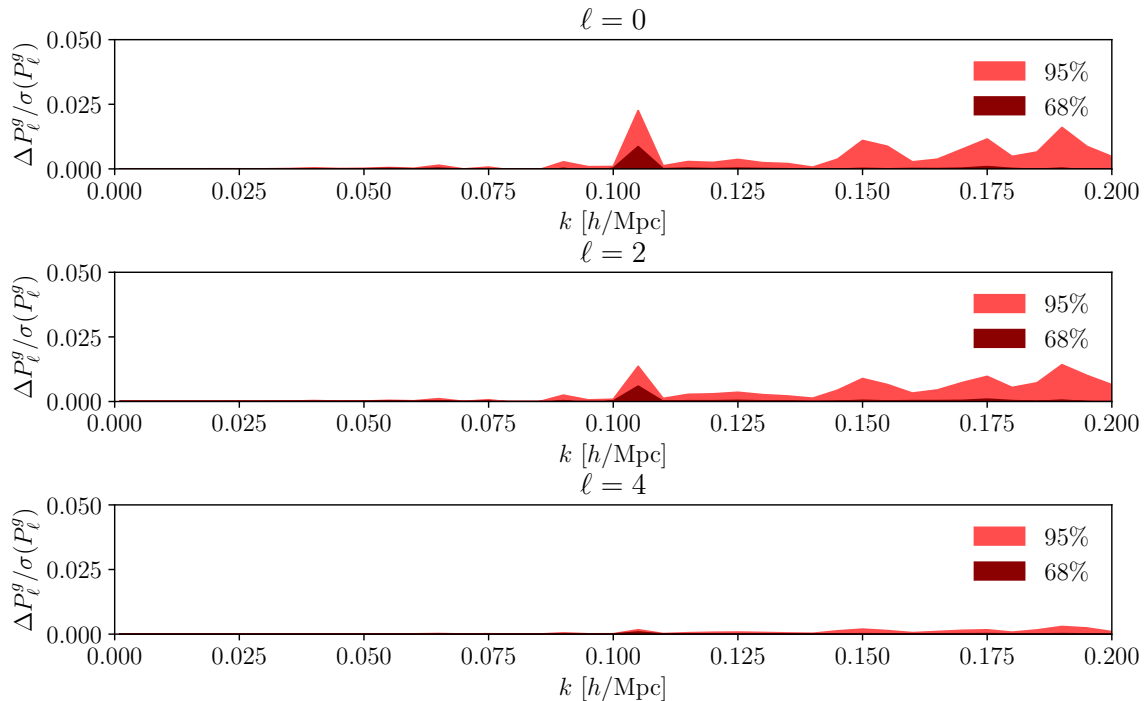


Figure 1. Spline decomposition accuracy — Distribution of errors in the galaxy power spectrum multipoles computed with `PyBird`, comparing results obtained using either the input linear matter power spectrum or its spline-reconstructed counterpart, across the `CosmoRef` testing bank described in table 1. Errors are shown relative to representative uncertainties expected for Stage-4 LSS surveys.

as $P_{\text{lin}}(k, z) \propto D^2(z) \simeq (1 \times z)^{-2}$; a *uniform* draw in z therefore yields a distribution of amplitudes that is very concentrated near zero. The log-normal distribution in z shifts the probability mass away from 0 while retaining a rather flat high- z tail, producing a flatter distribution in amplitude (and in growth rate f following the same argument) more suitable for agnostic training. The remaining dimensions consist of ten cosmological parameters: $\{\omega_b, \Omega_m, h, n_s, \ln(10^{10} A_s), \Omega_k, \sum m_\nu, N_{\text{eff}}, w_0, w_a\}$, corresponding respectively to the physical baryons abundance, the fractional dark matter abundance, the reduced Hubble constant, the spectral tilt, the log of the rescaled amplitude of the primordial power spectrum, the fractional curvature abundance, the total neutrino mass, the relativistic number of species, and the dark energy equation of state parameters. Each cosmological parameter is distributed in the hypercube according to a Gaussian centered on values close to *Planck* preferred ones [82], with standard deviation of about two or three times the size of the 68%CL uncertainties σ obtained on BOSS [13], and clipped at roughly 5σ . The parameter distributions are shown in table 1.

Figure 1 presents the quantiles of the residuals distribution on the galaxy power spectrum multipoles computed with `PyBird` taking either as input the full linear matter power spectrum or its spline over the 80 optimised k -knots. Each residual is weighted by the forecast 1σ uncertainties of a representative Stage-4 LSS survey (as described in section 2.2), and the statistic is evaluated across the entire test bank. With $N_k = 80$ knots and optimised placements, we find that the 95% quantile of the error distribution does not reach greater

than 3% of Stage-4 LSS uncertainties. The error from the spline decomposition is thus virtually negligible given our target precision.

In principle, other decompositions, potentially more suitable to capture the features in P_{lin} with a reduced number of basis functions, could be considered (see e.g., refs. [83–85]). However, these typically entail tradeoffs in flexibility and coverage, making the emulator less general to input cosmologies (and thus more prone to out-of-distribution anomalies) or leading to a loss of interpretability. For instance, the `FFTLog` decomposition used in refs. [83, 84] provides a useful basis in which to compute loop integrals analytically. However, as a complex power-law decomposition, the broadband of the linear matter power spectrum typically requires many basis functions to be described to sufficient accuracy. For reference, `PyBird` uses $N_{\text{FFT}} = 512$, much larger than $N_{\text{knots}} = 80$. In contrast, the `COBRA` method introduced in ref. [85] allows for an optimal decomposition with very few basis functions, $N \sim 10$. However, this is optimised for a given cosmological training bank and would require re-testing when applied to a model outside of the original training bank. Meanwhile, with our simple-minded choice, the spacing between k -knots is on average about $\Delta k \sim 0.01$, which is enough to resolve to good accuracy the features in $P_{\text{lin}}(k)$ from most cosmologies. If one doubts the accuracy of the parametrisation e.g., when exploring fast oscillating features with period below this k -resolution, the emulator can be simply switched off whilst still maintaining the speed boost of the pure `JAX` implementation.

Input emulator parameter space. Our input parameter space thus mainly consists in the log-power band values at the k -knots of the spline previously chosen. Note that the normalisation factor P_{max} does not need to enter the emulator for the loop integrals, as the loop integrals will be simply rescaled by P_{max}^2 . The same argument applies to the correction terms appearing in the IR-resummation scheme reviewed in appendix A. As shown by eq. (A.6), they would also be rescaled by powers of P_{max} . However, for practical reasons related to memory usage, we use another emulator depending further on P_{max} and the growth factor f for the IR-resummation pieces. The inclusion of f allows us to compute the matrix coefficients $Q^{\ell\ell', m, \alpha} \parallel_{N-j}(f)$ in eq. (A.6), such that we save only one IR-correcting term per linear / loop / counter- term, instead of $N_\ell \times M \times N_\alpha \sim 200$. We find that emulating the full IR-resummed loop pieces, as opposed to a separate loop and IR-correction piece, yields the best performance in both speed and accuracy. As mentioned before, we use the full analytic expressions for the linear and counterterm pieces so for these we train only the corresponding IR-correction parts. The next section details our prior choice for the input parameters, selected to ensure wide coverage for different cosmological models.

To provide the emulator with a large yet physically faithful training bank, we begin with a parent, representative ensemble, that we take to be the `CosmoRef` bank. This consists in $N_{\text{parent}} = 10^6$ linear power spectra drawn according to the distribution specified in table 1. The purpose of this bank is to inform the distribution in log-power band space to ensure some level of regularity and smoothness in the resulting representation of the linear power spectra. We then use a Gaussian-copula resampling strategy to generate additional, out-of-sample spectra, extrapolating beyond the restrictions from the inherent (cosmology-dependent) choice of the parent ensemble. Our approach is as follows. Each spectrum and their corresponding

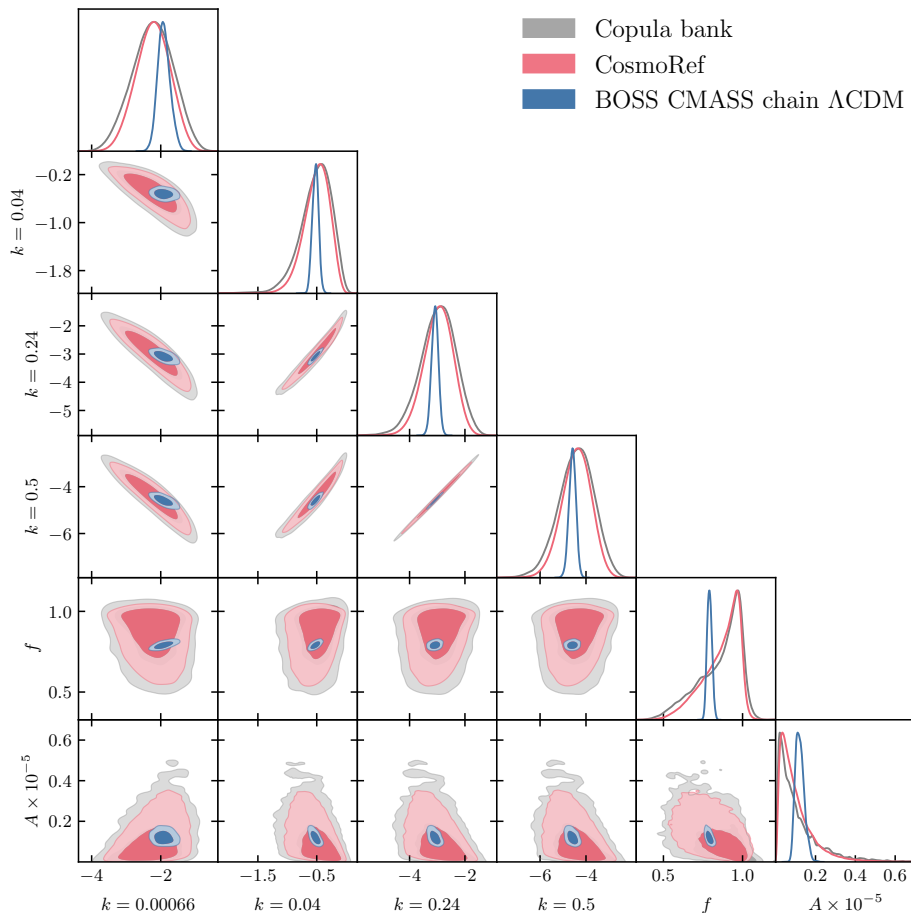


Figure 2. Representative coverage over emulator input parameter space — Inflated Gaussian copula distribution over the emulator training input space, shown for selected knots (all k s in $h \text{Mpc}^{-1}$), maximal power amplitude $A = P_{\text{max}}$ (in $(\text{Mpc } h^{-1})^3$), and growth factor f . The original reference *CosmoRef* bank and samples from BOSS CMASS ΛCDM analysis are also shown for comparison.

growth rate f are first compressed to the 82-component feature vector

$$\mathbf{x} = (a_1, \dots, a_{80}, A, f), \quad P_{\text{max}} \equiv \max_k P_{\text{lin}}(k), \quad f \equiv \frac{d \ln D}{d \ln a}, \quad (2.2)$$

where $a_j = \log P_{\text{lin}}(k_j)/P_{\text{max}}$, $j = 1, \dots, 80$, are the logarithmic band power amplitudes on our previously chosen fixed knots k_j for the input linear power spectrum spline. Because we desire the samples for the training distribution to respect both the empirical band power 1D marginals and the inter-knot correlations of the parent set, we decouple these two pieces with the copula construction. First, every coordinate x_i is “uniformised” by its empirical 1D marginal cumulative distribution function (CDF) F_i ,

$$u_i = F_i(x_i), \quad (2.3)$$

so that the variables $\{u_i\}$ are individually uniform on $[0, 1]$. We then use a copula to encode the correlations within the uniformised parameter space. Assuming a *Gaussian copula* means

imposing a latent normal representation

$$\mathbf{q} = \Phi^{-1}(\mathbf{u}) \sim \mathcal{N}(0, \Sigma), \quad (2.4)$$

where \mathbf{q} denotes the ‘‘Gaussianised’’ latent variables obtained by mapping each uniform variate u_i through the probit Φ^{-1} , i.e., through the inverse CDF of a standard normal. This transformation places the data in the space of a Gaussian copula: the multidimensional dependence is now encoded entirely in the covariance matrix Σ , which we estimate from the 10^6 linear bank realizations. Once Σ is known we can generate new correlated knot amplitudes in two computationally cheap steps: draw a single sample from $\mathcal{N}(0, \Sigma)$ and then apply, in reverse, the sequence Φ followed by each empirical inverse CDF F_i^{-1} . The resulting mock vectors faithfully reproduce both the 1D marginals and the inter-knot correlations of the training data, yet remain more agnostic to any particular cosmological model than the bank samples themselves; they therefore provide a fast, general-purpose source of realistic spline-space samples for emulator training.

To make the training data wider we multiply Σ by a factor $\alpha > 1$,

$$\tilde{\Sigma} = \alpha \Sigma, \quad (2.5)$$

which expands every latent variable standard deviation by $\sqrt{\alpha}$. New latent vectors $\tilde{\mathbf{q}} \sim \mathcal{N}(0, \tilde{\Sigma})$ are mapped back through the inverse transforms $\tilde{\mathbf{u}} = \Phi(\tilde{\mathbf{q}})$ and $\tilde{\mathbf{x}} = F^{-1}(\tilde{\mathbf{u}})$. Because the rescaled latent variable 1D marginals are broader than standard normals, the \tilde{u}_i are no longer strictly uniform; instead they are skewed toward the boundaries 0 and 1. Consequently the inverse-CDF transform drives the features \tilde{x}_i deeper into the empirical tails, allowing the training bank to explore regions that lie beyond the original `CosmoRef` ensemble and to exhibit reduced inter-knot correlations. In practice we find a modest value of $\alpha = 1.2$ to be a good balance between preserving the cosmologically-relevant inter-knot correlations whilst allowing for a widened parameter space coverage. The resulting bank captures the correct correlation structure between the knots to ensure physically viable power spectra (implying some level of regularity and smoothness in k) whilst the Gaussian copula with inflation by α allows for a wider class of models to be represented than in the original `CosmoRef` bank (both in the allowed amplitudes and shapes of linear matter power spectra). The training distribution over the emulator input power space is displayed in figure 2. We note that there could be alternative ways to generalise representative parent ensemble to a model-agnostic distribution, such as training normalising flows to a latent space. We leave such exploration for future work.

Training. We run `PyBird-JAX` for 10^7 samples of the feature vector x defined in eq. (2.2) drawn from the widened Gaussian Copula distribution over the emulator input parameter space to construct the training set. In practice, we emulate only the components of `PyBird` that are numerically expensive, which consist in

- The one-loop terms, either resummed or non-resummed, with one NN per multipole (monopole, quadrupole, hexadecapole), and one set of NNs for the resummed or non-resummed case;
- The IR-correction pieces for resumming the linear contributions, using one NN;
- The IR-correction pieces for resumming the counterterms, using one NN.

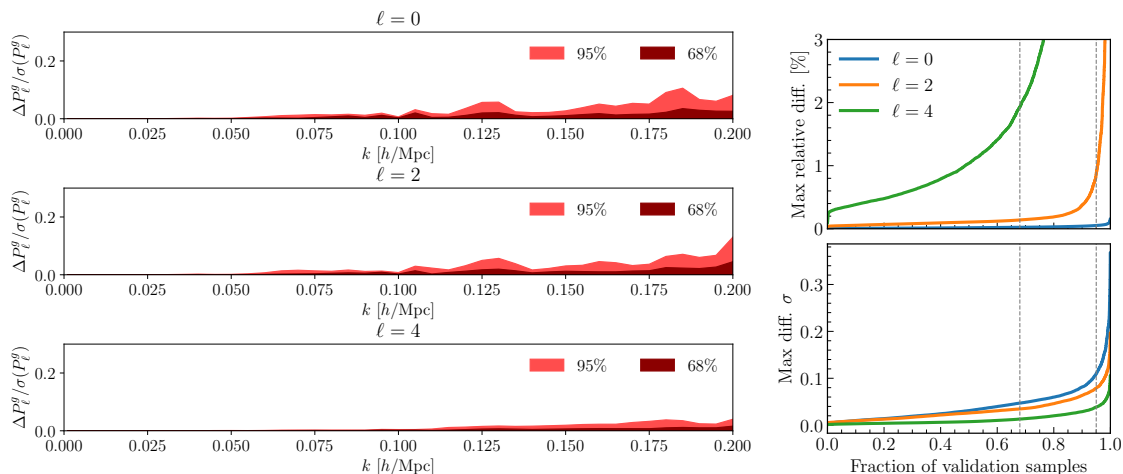


Figure 3. Emulator accuracy — *Left panel:* 68% and 95% quantiles of the differences in the galaxy power spectrum multipoles across scales, computed with PyBird. The comparison is between the NN-based emulator predictions and the full calculations, evaluated over the independent validation set described in section 2.2. Errors are shown relative to representative uncertainties expected for Stage-4 LSS surveys as described in the text. *Right panel:* Cumulative histogram of the maximum absolute differences across the full range of scales, with the two vertical dashed lines indicating the 68% and 95% quantiles of the validation samples.

Before training we compress every output vector with a principal-component analysis (PCA), retaining $N_{\text{PCA}} = 256$ modes for each NN. Before compression the output vector size is $\mathcal{O}(10^4)$ for each of the emulators. This yields two practical advantages: (i) the reduced dimensionality lowers GPU memory requirements, allowing faster training; (ii) the smaller dynamic range of the compressed targets improves numerical stability and accelerates convergence. We adopt the NN architecture from ref. [25],⁵ implemented in TensorFlow [86]. Each model consists of three hidden layers with 512 neurons per layer. The activation function in each layer is given by

$$f(\vec{x}) = \left(\vec{\gamma} + (1 + \exp(-\vec{\beta} \cdot \vec{x}))^{-1} (1 - \vec{\gamma}) \right) \cdot \vec{x}, \quad (2.6)$$

where \vec{x} is the vector of inputs and the hyper-parameters $\vec{\beta}$ and $\vec{\gamma}$ are trained alongside the usual network weights. We use the ADAM optimiser [87] with default parameters, a batch size of 1024, and train each emulator for 1000 epochs (i.e., full passes of the entire training dataset) for each of 5 separate ADAM learning rates (LR) that are decreased in a ladder from $LR = 10^{-3}$ to $LR = 5 \times 10^{-5}$. For this optimisation, four fifths of the training set are used as a pure training set, whilst the remaining fifth is used as an internal validation set.

2.2 Performance

Accuracy. While full end-to-end precision tests of PyBird-Emu on recovered cosmological parameters are presented in section 4, we first assess the emulator’s accuracy using an independent validation set constructed as follows. Note that this validation set is different than the CosmoRef bank that we used to inform the training of our emulator. We adopt

⁵Available at <https://github.com/alessiospurriomancini/cosmopower>.

the same cosmological parameter space used to train COMET [71], an emulator for the galaxy power spectrum in redshift space developed to meet Euclid’s precision requirements. These cosmologies are drawn from the following flat priors: $\omega_b \in [0.02050, 0.02415]$, $\omega_{\text{cdm}} \in [0.085, 0.155]$, $n_s \in [0.92, 1.01]$, $\sigma_{12} \in [0.2, 1.0]$, and $f \in [0.5, 1.05]$, where σ_{12} denotes the amplitude of fluctuations in spheres of radius 12 Mpc. Crucially, evolution parameters such as A_s , w_0 , w_a , and $\Omega_k h^2$ (as opposed to shape parameters such as ω_b , ω_{cdm} , or n_s), as well as the redshift z dependence, affect the linear power spectrum in a manner equivalent to a rescaling of σ_{12} [88]. This means that the resulting test bank covers wide range of cosmological evolution scenarios and therefore provides a stringent test of the emulator. We generate 5×10^3 such linear matter power spectra, which are then passed through both PyBird-JAX and PyBird-Emu for comparison.

Figure 3 shows the distribution of fractional errors on the independent validation set, normalised by representative observational uncertainties for Stage-4 LSS surveys. The survey characteristics assumed are an effective volume of $V_{\text{eff}} = 50 \text{ Gpc}^3$ and a tracer number density of $\bar{n} = 5 \times 10^{-4} (h \text{ Mpc}^{-1})^3$. These uncertainties are estimated under the Gaussian approximation, assuming a fiducial Kaiser power spectrum at the *Planck* best fit cosmology with linear bias $b_1 \sim 2$, evaluated at the effective redshift z_{eff} corresponding to each tested spectrum. Note that $V_{\text{eff}} = 50 \text{ Gpc}^3$ is representative of the total effective volume of a Stage-4 LSS survey, while the volume per redshift bins in practice is usually less than 10 Gpc^3 (see table [2] of our companion paper for DESI-like survey characteristics per redshift bins). The emulators meet the accuracy requirements comfortably across the full parameter space, with 95% of the residuals remaining below 0.1σ for all multipoles. Here we have focused on assessing emulator accuracy within the regime where the one-loop theoretical model for the redshift-space galaxy power spectrum is expected to be reliable, namely up to $k_{\text{max}} = 0.2 h \text{ Mpc}^{-1}$. As data precision increases, the theoretical k -reach is expected to decrease. Consequently, as survey volumes grow, the emulator is expected to remain relevant, as its accuracy improves at larger scales (lower k). Nevertheless, in appendix B, we provide additional results on emulator accuracy across a broader range of scales, extending to $k_{\text{max}} = 0.3 h \text{ Mpc}^{-1}$. There, we observe that the relative error increases, reaching approximately 0.5σ . However, we do expect that the induced bias on cosmological parameters to be smaller, as (i) the induced error on the total χ^2 is a mean square error averaged over all k ’s, and (ii) numerical errors tend to average out in posterior sampling. Indeed, as shown in section 4.1, when applied to large-volume simulations, the resulting bias on inferred cosmological parameters, when fitting the data in wedges up to $k_{\text{max}} = 0.3 h \text{ Mpc}^{-1}$, remains tolerably small in practice.

Speed. Table 2 compares the wall-clock time required to evaluate the three redshift-space multipoles of the galaxy power spectrum with different back-ends and hardware configurations. Switching from PyBird to its JAX rewrite already provides a $5\times$ speed-up on a CPU through just-in-time (JIT) compilation. Replacing the costly loop integrals and IR-resummation with NN-based emulators brings a gain of two order-of-magnitude on the CPU, with another order moving to a Tesla A100 GPU. Furthermore, using JAX’s native vmap functionality, the vectorisation of the code enables another factor of $\sim 3 - 10$ speed up on the GPU hardware we used for benchmarking, depending on the input batch size (tested from 48 to 128).

Architecture	PyBird variant	Time per $3 \times P_k^{(\ell)}$
CPU	PyBird ^a	560 ms
CPU	PyBird-JAX	102 ms
CPU	PyBird-Emu	1.2 ms
GPU	PyBird-JAX	2.0 ms
GPU	PyBird-Emu	0.19 ms
GPU	PyBird-Emu (vectorised, batch = 128)	0.023 ms

^a Note that previous public version of PyBird was using by default $N_{\text{FFT}} = 256$, for which the corresponding timing is 285ms. See also appendix A.

Table 2. Wall-clock time to evaluate the monopole, quadrupole, and hexadecapole ($3 \times P_k^{(\ell)}$) of the redshift-space galaxy power spectrum for different back-ends.

These runtime gains are broadly consistent with back-of-the-envelope counts of floating-point operations (flop). For the full PyBird calculation, the dominant cost stems from the 22-diagram loop integrals, which is computed using the FFTLog as a matrix multiplication [40, 42, 83]

$$P_{22}(k) = \sum_{n=1}^{N_{\text{FFT}}} \sum_{m=1}^{N_{\text{FFT}}} c_n k^{-2\nu_n} \cdot M_{22}(m, n) \cdot k^{-2\nu_m} c_m, \quad (2.7)$$

where ν_n are complex powers resulting from the FFTLog decomposition of P_{lin} , and where $M_{22}(m, n)$ are matrices that are pre-computed.⁶ The correlation function, used in the IR-resummation scheme of PyBird as detailed in appendix A, is computed likewise [40, 42]. The complexity of these operations can be approximated by

$$\mathcal{C}_{\text{loops}} \simeq N_{\text{FFT}}^2 N_k N_{\text{loop}}.$$

With the default values FFTLog points $N_{\text{FFT}} = 512$, length of the internal k -array $N_k = 77$, and number of 22-loop terms $N_{\text{loop}} = 35$, this gives $\mathcal{C}_{\text{loops}} \sim 1 \times 10^9$ flop. In practice, optimised matrix contraction through NumPy einsum leveraging BLAS backend reduces the flop counts by roughly an order-one factor. By contrast, a forward pass through the neural emulator involves

$$\mathcal{C}_{\text{emu}} \simeq N_{\text{multipoles}} [N_{\text{in}} N_{\text{nodes}} + N_{\text{layers}} N_{\text{nodes}}^2 + N_{\text{out}} N_{\text{nodes}}],$$

where $N_{\text{multipoles}} = 3$ is the number of multipoles, $N_{\text{in}} = 82$ is the number of input parameters, $N_{\text{layers}} = 3$ the number of hidden layers, $N_{\text{nodes}} = 512$ the width of each layer, and $N_{\text{out}} = 256$ the number of retained PCA modes. Evaluating this expression yields $\mathcal{C}_{\text{emu}} \sim 2 \times 10^6$ flop. The ~ 500 -fold reduction in arithmetic work largely accounts for the observed timing improvement; the remaining discrepancy arises from Python overheads, memory traffic, and specific architecture designs, which are not captured by a simple flop estimate.

⁶Note that P_{22} here is not P^{ij} in eq. (2.1) with $i = 2 = j$. Instead, many P^{ij} are of the 22-diagram type, and are computed following eq. (2.7) for their respective M_{22} .

Observational effects. The evaluation time reported in table 2 are benchmarked for the IR-resummed one-loop power spectrum multipoles, without accounting for additional observational effects that need to be modelled to accurately fit the data. Following ref. [81], we upgrade the numerical scheme in PyBird for correcting geometrical distortions from the Alcock-Paczynski (AP) effect to a Gauss-Lobatto-Legendre integration rule using 8 fixed points. This is enough to achieve a 10^{-5} relative precision compared to a reference quadrature integration, while making the runtime spent on correct the AP effects virtually negligible. As for the window function convolution (that in the case of periodic boxes reduces to a convolution with a binning matrix), it takes about $X - Y\%$ of the total runtime, depending on PyBird back-ends.

3 PyBird-JAX: boosted cosmological inference in LSS

PyBird-JAX is a differentiable code written in Python-JAX delivering fast evaluation of EFTofLSS predictions for galaxy-clustering observables and cosmological likelihoods of LSS probes. Providing as input a linear matter power spectrum and growth functions computable with a Boltzmann solver, PyBird-JAX outputs the nonlinear ‘one-loop’ correlation functions of galaxies in redshift space, dressed with modeling of observational effects from a given survey. As a standalone tool, it can be used in combination with any Boltzmann code and samplers. Yet, we provide:

- An end-to-end pipeline to analyse N -point statistics of spectroscopic data: BOSS DR12 luminous red galaxies [89, 90], eBOSS DR16 quasars [91, 92], and simulated data of upcoming surveys. We intend to continuously add support for likelihoods of future data as they become available.
- Built-in compatibility with existing Boltzmann solvers or emulators: CLASS⁷ [93], CosmoPower-JAX⁸ [94], DISCO-EB⁹ [35], and Symbolic-Pk¹⁰ [30, 95].
- Built-in compatibility with several contemporary numerical samplers for cosmological inference: ensemble sampling from emcee¹¹ [96] or zeus¹² [97], Hamiltonian Monte Carlo (HMC) (see ref. [98] for a review) based on the No-U-Term algorithm (NUTS) [99] from BlackJAX¹³ [100], Microcanonical Hamiltonian Monte Carlo (MC-HMC) [101] also from BlackJAX, and nested sampling from Nautilus¹⁴ [102].
- An embedding within the widely-used cosmological inference software MontePython¹⁵ [103, 104]. We also plan to upgrade the current PyBird integration [19] to PyBird-JAX

⁷<http://class-code.net/>.

⁸<https://github.com/dpiras/cosmopower-jax>.

⁹<https://github.com/ohahn/DISCO-EB>.

¹⁰https://github.com/DeaglanBartlett/symbolic_pofk.

¹¹<https://emcee.readthedocs.io/>.

¹²<https://zeus-mcmc.readthedocs.io/>.


¹³<https://blackjax-devs.github.io/blackjax/>.

¹⁴<https://github.com/johannesulf/nautilus>.

¹⁵https://github.com/brinckmann/montepython_public.

within `desilike`,¹⁶ the official analysis pipeline of DESI, and to integrate `PyBird-JAX` into `cloe`,¹⁷ the Euclid analysis pipeline (currently under development).

At its core, `PyBird-JAX` is essentially the same code as `PyBird`¹⁸ — the Python code for Biased tracers in redshift space [40], but updated to `JAX`, with further acceleration from the NN-based emulator presented in section 2. For this first release, `PyBird-JAX` has been gauged and tested for the one-loop power spectrum of galaxies in redshift space [3]. We intend to extend `PyBird-JAX` to other cosmological observables, in particular the ones currently supported by `PyBird`: one-loop bispectrum [4, 13] and two-point correlation function in configuration space [42].

A word on the implementation. Due to `JAX` being literally composable transformations of a native Python and NumPy¹⁹ program into a differentiable, vectorised, and GPU-compatible compiled version, `PyBird-JAX` is turned on with a simple backend switch on top of `PyBird`, with the main code being agnostic on whether functions are called from the `JAX` ecosystem or standard Python library. In particular, functions that differ in their call between their Python and `JAX` version are wrapped within single instances called by the main code of `PyBird`, such that the latter is unique for both versions. Users and developers of `PyBird` will enjoy all features brought by `JAX`, that we detail in the following, with no particular adjustment other than installing extra dependencies. As a first introduction to `PyBird` and its accelerating variants, we provide a general demonstration notebook here .

3.1 JAX-powered features

Just-in-time compilation. A key strength of `JAX` lies in its ability to remain fully compatible with the broader Python environment while enabling Just-In-Time (JIT) compilation.²⁰ In typical cosmological inference workflows where `PyBird-JAX` is called repeatedly (see examples in section 4), both `PyBird-JAX` and target likelihoods can be JIT compiled once at the beginning. This ensures that subsequent core computations are executed directly at the machine level, bypassing overheads of interpreting higher-level codes. JIT compilation brings substantial speedups, as demonstrated in section 2.2, particularly when run on a GPU. A typical JIT compilation of `PyBird-JAX` completes in a few seconds, with additional time depending on the specific Boltzmann solver used in conjunction.

Neural network embedding. Although the neural network weights are trained using TensorFlow (as described in section 2), we embed the trained model smoothly into the `JAX` ecosystem using the `flax`²¹ library. This integration allows our emulator-enhanced `PyBird-JAX` to be fully JIT-compiled, with automatic differentiation operating swiftly through the NN architecture. The speed gains enabled by the emulator were presented in section 2.2, while its accuracy in cosmological inference is evaluated in section 4.

¹⁶<https://desilike.readthedocs.io/>.

¹⁷<https://github.com/cloe-org>.

¹⁸<https://github.com/pierrexzyz/pybird>.

¹⁹<https://numpy.org/>.

²⁰<https://docs.jax.dev/en/latest/jit-compilation.html>.

²¹<https://flax.readthedocs.io/>.

Boltzmann solver streaming. While an increasing number of Boltzmann solvers or emulators are becoming JAX-compatible —enabling seamless integration with PyBird-JAX as we provide — benchmark codes such as CLASS [93] and CAMB²² [105, 106] are not. These established solvers continue to offer unmatched precision and a comprehensive set of cosmological models for exploration. However, being written in C and interfaced with Python via Cython,²³ CLASS cannot be directly embedded into the JAX ecosystem.²⁴ To retain a fully JIT-compiled and differentiable pipeline when using such solvers, PyBird-JAX can switch to a finite-difference-based Taylor expansion of the observables, as detailed in section 3.2. Built-in compatibility between PyBird-JAX and Boltzmann solvers are showcased here [↗](#).

Vectorised ensemble sampling. PyBird-JAX, when combined with a Boltzmann solver, computes predictions for LSS likelihoods based on input cosmological parameter sets. Thanks to JAX automatic vectorisation via `vmap`,²⁵ PyBird-JAX can efficiently process batches of input parameters. This leads to significant speedups over sequential evaluation by leveraging optimised matrix algebra from low-level libraries such as BLAS, while also minimising overheads. Vectorisation is particularly advantageous for ensemble sampling, allowing multiple walkers to run in parallel. For PyBird-JAX, we observe favorable time scaling with batch sizes up to ~ 1024 . In practice, using around ~ 128 walkers offers a good trade-off: rapid convergence to the target distribution is achieved with a manageable number of burn-in steps and efficient sampling is obtained thereafter in a few steps. An example of cosmological inference with PyBird-JAX using vectorised ensemble sampling is provided here [↗](#).

3.2 AD-powered features

JAX enables machine-precision differentiation with modest overhead via automatic differentiation (AD). AD exploits the fact that a program is a composition of elementary operations with known derivatives. By recording the computation graph (“*evolution tracing*”) and applying the chain rule, one obtains derivatives of the full composition (see ref. [107]). For scalar objectives of many parameters (e.g., likelihoods), reverse-mode AD (backpropagation) is typically optimal, propagating sensitivities from outputs back to inputs.²⁶ Reverse-mode proceeds in two passes: a forward pass that builds the trace, and a backward pass that accumulates gradients. In brief, AD in PyBird-JAX provides fast, accurate derivatives of EFTofLSS predictions and their likelihoods. The new AD-enabled functionalities are summarised below.

Fisher matrix. With AD, estimating Fisher matrix as the Hessian of the log-likelihood functions of target experiments has never been easier — a one-liner code for a numerically stable computation. Beyond the common use of forecasting parameter constraints, the Fisher matrix can be leveraged to increase efficiency of numerical sampling by starting at positions drawn from a multivariate Gaussian centered on some fiducials (e.g., the maximum

²²<https://camb.info/>.

²³<https://cython.org/>.

²⁴This may be possible in the future with tools such as Enzyme-JAX: <https://github.com/EnzymeAD/Enzyme-JAX>.

²⁵<https://docs.jax.dev/en/latest/automatic-vectorization.html>.

²⁶On modern hardware such as GPUs, the simpler primitives of forward-mode AD can sometimes compensate for the extra evaluations; see, e.g., ref. [108].

a posteriori) with covariance given by the inverse-Fisher matrix, thereby reducing the burn-in phase. For a typical $\mathcal{O}(10)$ -parameter likelihood in galaxy clustering, PyBird-JAX computes the Fisher matrix in seconds (with the timing largely dominated by JIT compilation), an amount of time worth spending to initialise subsequent sampling, reducing significantly the burn-in time. An example is provided here [🔗](#). The Fisher matrix also enters the non-flat integration measure used to define unbiased expectation values as shown in our companion paper [1] (also detailed below).

Taylor expansion of observables. Observables in LSS such as the power spectrum and bispectrum are smooth functions of cosmological parameters. Their cosmological dependence can therefore be Taylor-expanded around a fiducial cosmology, enabling extremely fast emulation of theoretical predictions [17, 18]. With AD, PyBird-JAX can precompute in minutes the required Taylor expansions for the observables entering the target likelihoods, allowing for highly accelerated sampling. When the Boltzmann solver is not itself differentiable, PyBird-JAX switches to finite difference, and the resulting emulator from the Taylor expansion is then effectively differentiable. Accuracy benchmarks are reported in section 4.3, and the corresponding implementation is showcased here [🔗](#).

Gradient-based optimisation. The minimum χ^2 is the standard metric used to assess the quality of a model’s fit to experimental data. Correspondingly, the maximum a posteriori (MAP) estimate, or posterior mode, is commonly reported as the most likely point estimate for model parameters under a given likelihood and prior. Finding the MAP boils down to solving an optimisation problem over a scalar objective function, which is typically non-analytic and must be approached numerically. In high-dimensional or multimodal parameter spaces, gradient information becomes essential for efficient exploration. This is particularly relevant for galaxy-clustering likelihoods, where the number of nuisance parameters scales with the number of observed sky patches — easily reaching $\mathcal{O}(100)$ parameters in current Stage-4 LSS surveys. Additionally, increasingly sophisticated modeling compounds this complexity. With PyBird-JAX, we find that the classic MIGRAD algorithm, as implemented in the widely used Minuit²⁷ minimiser [109], performs reliably in low-dimensional settings ($\lesssim 10$ parameters), especially for near-Gaussian posteriors as typically encountered in cosmology. However, it often fails to locate the MAP as the number of parameters increases — either due to added sky patches or more complex models — particularly when poorly constrained directions lead to nearly flat regions in parameter space, i.e., large parameter degeneracies. In contrast, ADAM²⁸ [87], a first-order gradient-based stochastic optimiser, consistently performs well across all practical cases tested with PyBird-JAX (see section 4). We anticipate that its performance can surpass that of global optimisers such as simulated annealing, as implemented in Procoli [110] and PROSPECT [111], both in terms of robustness and convergence speed. We leave a detailed comparison for future work. Typically, ADAM converges within minutes for models with $\mathcal{O}(10)$ parameters in galaxy clustering applications using PyBird-JAX. Because point estimates like the MAP can be sensitive to numerical noise, we recommend computing them without using the emulator, which introduces small but non-negligible errors at each

²⁷<https://scikit-hep.org/iminuit/>.

²⁸As implemented in <https://optax.readthedocs.io/>.

point in parameter space. These minor inaccuracies can shift the MAP estimate, especially in flat regions of the posterior. In contrast, over many evaluations, such stochastic errors tend to average out, making the emulator well suited for sampling-based inference and accurate estimation of full posterior distributions, that we now detail.

Gradient-based sampling. To explore increasingly large parameter spaces — such as those encountered in galaxy clustering analyses that incorporate multiple tracers across different sky patches and redshift bins, each contributing their own set of nuisance parameters — gradient-based sampling offers a practical alternative to traditional samplers, which often do not scale efficiently with dimensionality. Performance of HMC-NUTS for a $\mathcal{O}(10\text{--}100)$ -parameter case are compared against ensemble sampling in section 4. In practice, the total runtime depends on several factors, including the tuning of hyperparameters, the length of the burn-in period, and critically, the additional cost of computing gradients, which we find to be around $3\times$ the likelihood call. This makes gradient-based sampling a method of choice, as demonstrated in section 4.3 and here [↻](#).

Marginals with non-flat volume measure. Perhaps among the additional features enabled by AD, this one stands out for its novelty. As demonstrated in our companion paper [1], we propose to improve parameter estimation by redefining our expectation values with respect to non-flat volume measures, removing bias in marginal posteriors known as volume projection effects (see e.g., refs. [11, 13, 14, 46, 72, 112]). For a smooth posterior distribution $\mathcal{P}(\boldsymbol{\theta}|y)$ over model parameters $\boldsymbol{\theta}$ obtained from the data y that admits a global maximum, the optimal volume measure we put forward takes a form close to the well-known Jeffreys *prior* [1],

$$\mathcal{M}_{\mathcal{H}}(\boldsymbol{\theta}) = \sqrt{\det \mathcal{H}(\boldsymbol{\theta})} d^N \boldsymbol{\theta}, \quad \mathcal{H}_{\mu\nu}(\boldsymbol{\theta}) = \partial_{\mu} \partial_{\nu} \log \mathcal{P}(\boldsymbol{\theta}|y = m(\boldsymbol{\theta}_{*})). \quad (3.1)$$

Here $\mathcal{H}_{\mu\nu}$ is a matrix taken as the Hessian over the distribution $\log \mathcal{P}(\boldsymbol{\theta}|y = m(\boldsymbol{\theta}_{*}))$, where the data are taken to be the model m evaluated on the mode $\boldsymbol{\theta}_{*}$. In practice, we can add to each sample of the log-posterior obtained under the standard flat measure a log-measure weight to compute marginal statistics. This measure is found to correct for the average mean bias we derive at leading order in the Laplace expansion of the posterior [1],

$$b[\mathcal{F}, \boldsymbol{\theta}_{*}] = (\mathcal{F}_{\alpha\mu}^{-1} \mathcal{F}_{\nu\rho}^{-1} \partial_{\rho} \mathcal{F}_{\mu\nu})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{*}} \quad (3.2)$$

Alternatively, this bias can be computed once and used as post-debiasing correction to shift the posterior, yielding an unbiased representation of marginal distributions and their credible intervals. With the upgrade to PyBird-JAX, both the log-measure weight (3.1) or the bias term (3.2) can now be computed seamlessly and at machine precision, making the construction of unbiased marginal inferences in large-scale structure efficient and robust. Since this paper focuses on the speed and accuracy performance of PyBird-JAX, we do not apply this correction in the applications discussed in section 4. Practical examples of its use are instead provided in our companion paper [1] with implementation showcased here [↻](#).

4 Cosmological applications

To demonstrate the accuracy and flexibility of the cosmology-independent emulator of PyBird-JAX presented in section 2, that we dub PyBird-Emu, we compare posteriors of

Category	Parameter	Prior
Galaxy biases	b_1	$[0, 5]$
	b_2	$\mathcal{N}(0, 5)$
	b_3	$\mathcal{N}(0, 5)$
	b_4	$\mathcal{N}(0, 5)$
Counterterms	c_{ct}	$\mathcal{N}(0, 5)$
	$c_{r,1}$	$\mathcal{N}(0, 5)$
	$c_{r,2}$	$\mathcal{N}(0, 5)$
Stochastic terms	$c_{\epsilon,0}$	$\mathcal{N}(0, 2)$
	$c_{\epsilon,1}$	$\mathcal{N}(0, 5)$
	$c_{\epsilon,2}$	$\mathcal{N}(0, 5)$
NNLO counterterms	$c_{r,4}$	$\mathcal{N}(0, 5)$
	$c_{r,6}$	$\mathcal{N}(0, 5)$

Table 3. Priors for EFT nuisance parameters used in all analyses — All are assigned Gaussian priors $\mathcal{N}(\mu, \sigma)$, with μ denoting the central value and σ the standard deviation, except for b_1 , which is sampled with a flat prior. These priors, inspired by refs. [10, 14, 46], are based on order-of-magnitude estimates for the parameter sizes, motivated by naturalness considerations to ensure that EFTofLSS predictions remain within the perturbative regime where the theory is valid. For the EFT scales associated with the counterterms, we adopt $k_{\text{NL}} = k_{\text{M}} = 0.7 h \text{ Mpc}^{-1}$ and $k_{\text{R}} = 0.25 h \text{ Mpc}^{-1}$. See main text for a detailed summary of the EFT parameters and their role in modeling the galaxy power spectrum in redshift space at one loop in the EFTofLSS.

cosmological parameters from LSS analysis obtained with or without the use of the emulator. In section 4.1, we fit large-volume simulations to assess the accuracy of `PyBird-Emu` in recovering cosmological parameters. In section 4.2, we fit the Sloan Digital Sky Survey Baryon Oscillation Spectroscopic Survey (SDSS BOSS) [90] Luminous Red Galaxies (LRGs) [89] of modest volume compared to ongoing and upcoming surveys such as DESI [8] or Euclid [9], making it ideal to stress test `PyBird-Emu`'s coverage in parameter and model space. Finally in section 4.3, we perform a forecast on a DESI-like survey consisting of 7 individual skies, each fit with its unique EFT nuisance parameters to capture variations in the selection function and redshift, making a total of 84 parameters to explore on top of the cosmological parameters of interest. This provides an ideal realistic setup to test the various sampling strategies presented in section 3. Taken together, this suite of analyses aims to demonstrate the raw precision, coverage in parameter and model space, and efficiency of `PyBird-Emu` in inferring cosmological parameters from the incoming data of Stage-4 spectroscopic surveys in the next decade.

Cosmological inference setup. For all analyses of galaxy-clustering data presented in this section, as well as in our companion paper [1], we work with the following inference setup. We fit a data vector y consisting of the first three even power spectrum multipoles ($\ell = 0, 2, 4$) concatenated together in k -bins falling in the range $[k_{\text{min}}, k_{\text{max}}] = [0.01, 0.20] h \text{ Mpc}^{-1}$, except if stated otherwise. We use a Gaussian likelihood to describe the data y ,

$$-2 \log \mathcal{L}(y|\boldsymbol{\theta}) = (m(\boldsymbol{\theta}) - y)^T C^{-1} (m(\boldsymbol{\theta}) - y), \quad (4.1)$$

where $m(\boldsymbol{\theta})$ is the model (summarised below) evaluated with `PyBird` or its variants considered in this work, with $\boldsymbol{\theta}$ the model parameters consisting in the cosmological parameters of interest and the EFT parameters describing each sky composing the dataset considered. Here C^{-1} is the inverse covariance whose estimation is described below for each dataset considered in our works. To estimate the posterior of $\boldsymbol{\theta}$, we run MCMC chains using various samplers described in section 3. When not stated otherwise, we use the Metropolis-Hasting algorithm implemented in `MontePython` [103, 104], running 8 – 16 chains in parallel, with convergence monitored using the Gelman-Rubin criterion $R < 0.01$. All triangle plots in this paper are produced using `GetDist`²⁹ [113].

Model and priors. The one-loop model for redshift-space galaxy power spectrum that we consider in this work depends on twelve EFT parameters:

- *Galaxy biases*: four parameters $b_i, i = 1, \dots, 4$, entering the perturbation theory loop kernels of galaxies in redshift space, as formulated in the Basis of Descendants [114, 115] (see also refs. [4, 116–119]).
- *Counterterms*: three parameters $c_{\text{ct}}, c_{r,1}, c_{r,2}$, renormalising the UV-divergence of the P_{13} loop diagram [3, 120–122]. These scale relative to P_{lin} as $\frac{k^2}{k_{\text{M}}^2}$, and $\frac{k^2}{k_{\text{R}}^2}\mu^2, \frac{k^2}{k_{\text{R}}^2}\mu^4$, with k_{M}^{-1} and k_{R}^{-1} denoting characteristic EFTofLSS scales controlling the spatial derivatives or redshift-space velocity products [43, 123].
- *Stochastic terms*: three parameters $c_{\epsilon,0}, c_{\epsilon,1}, c_{\epsilon,2}$, renormalising the UV-divergence of the P_{22} loop diagram and scaling (in units of the mean number density \bar{n}^{-1}) as $k^0, \frac{k^2}{k_{\text{M}}^2}$, and $\frac{k^2}{k_{\text{M}}^2}\mu^2$, respectively [3].
- *Next-to-next-leading order redshift-space counterterms*: two parameters $c_{r,4}, c_{r,6}$, enhanced given a large k_{R}^{-1} (compared to $k_{\text{M}}^{-1} \sim k_{\text{NL}}^{-1}$), scaling relatively to P_{lin} as $\frac{k^4}{k_{\text{R}}^4}\mu^4$ and $\frac{k^4}{k_{\text{R}}^4}\mu^6$ [43].

Our priors on the EFT parameters consist in large Gaussian distributions specified in table 3. All EFT parameters — except the galaxy biases b_1, b_2, b_3 — enter the model linearly and are analytically marginalised as detailed in appendix D of our companion paper [1]. Cosmological parameters are scanned with large flat priors if not stated otherwise. We sample the Λ CDM parameters $\{\omega_{\text{b}}, \omega_{\text{cdm}}, \ln(10^{10}A_s), n_s\}$, with eventually extra parameters in the model extensions presented below. We however choose to present posteriors for the fractional total matter abundance Ω_m and the clustering amplitude σ_8 instead of ω_{cdm} and $\ln(10^{10}A_s)$ to meet the conventions in the LSS community. When analysing the BOSS data, we consider one neutrino at minimal mass $m_\nu = 0.06$ eV and two massless following *Planck* prescription [82] (see e.g., refs. [124, 125] for the validity of this approximation). When analysing simulations of synthetic data, we do not vary neutrinos.

²⁹<https://getdist.readthedocs.io>.

4.1 PT challenge simulations

To gauge the accuracy of `PyBird-Emu`, we make use of the PT challenge mocks, a suite of 10 high-resolution N -body simulations painted with a high-fidelity Halo Occupation Distribution (HOD) model, described in ref. [41]. By fitting the average power spectrum obtained from combining measurements across individual boxes, corresponding to a total data volume of $566 (\text{Gpc}/h)^3$, the PT challenge offers a stringent accuracy test for theoretical prediction codes in galaxy clustering. This setup enables direct comparison to the (blinded) truth of the simulations, as originally designed for testing predictions from the EFTofLSS [41].³⁰ In the original challenge presented in ref. [41], a pre-version of `PyBird` was used, showing that the power spectrum multipoles could be fit up to $k_{\text{max}} \simeq 0.14 h/\text{Mpc}$ reliably for the given simulation volume without significant bias in the inferred cosmological parameters. In ref. [43], it was further demonstrated using `PyBird` that higher k_{max} could be reached by constructing a linear combination of the power spectrum multipoles dubbed \mathcal{P} where the terms proportional to powers of μ , arising from the redshift-space distortions, are killed (see also ref. [126]). In short, the redshift-space distortions are controlled by a large renormalisation scale $\sim 1/k_{\text{R}}$ (see also ref. [127]). In comparison, the EFTofLSS prediction for \mathcal{P} is roughly controlled by the nonlinear scale $1/k_{\text{NL}}$ (and the galaxy spatial extension $1/k_{\text{M}}$), which is typically a few times smaller than $1/k_{\text{R}}$. Thanks to an increased convergence, the k -reach in the fit to \mathcal{P} can be pushed beyond the one of the traditional multipoles, typically up to $k_{\text{max}} \simeq 0.30 h/\text{Mpc}$ on the PT challenge as demonstrated in ref. [43]. The fit to \mathcal{P} combination from the PT challenge power spectrum thus makes an highly demanding accuracy test for `PyBird-Emu` covering all modes up to the highest k s reachable in current galaxy clustering survey with the EFTofLSS at one loop.

In figure 4, we show cosmological results obtained by fitting either the power spectrum multipoles P_ℓ (using $k_{\text{max}} = 0.14 h/\text{Mpc}$) or \mathcal{P} (using $k_{\text{max}}^{\mathcal{P}} = 0.30 h/\text{Mpc}$) and two wedges w_1, w_2 (using $k_{\text{max}}^{1/2} = 0.20, 0.10 h/\text{Mpc}$; see ref. [43] for precise definition of these combinations and justifications for the choice of k_{max} s). We see that `PyBird-Emu` perform equally well than `PyBird` in recovering the true cosmological parameters within uncertainties, with their posteriors differing less than 0.06% for all parameters in both cases, with the exception of σ_8 from $\mathcal{P} + w_{1,2}$ that differs at $\sim 0.3\%$. These represent negligible errors with respect to target precision in cosmology in the next decade (of the order $\sim 1\%$). `PyBird-Emu` is thus sufficiently accurate in recovering cosmological parameters from the full-shape power spectrum of ongoing and upcoming galaxy surveys.

4.2 BOSS data

Having validated the raw accuracy of `PyBird-Emu`, we now assess its performance in regards of two other metrics: coverage across cosmological parameter space and flexibility in reliably fitting extended cosmological models.

Data and likelihoods. Our main dataset is built from the BOSS DR12 LRG samples [89]. The power spectrum multipoles, obtained in ref. [13], were measured using the estimators

³⁰Submitted results are available at [here](#).

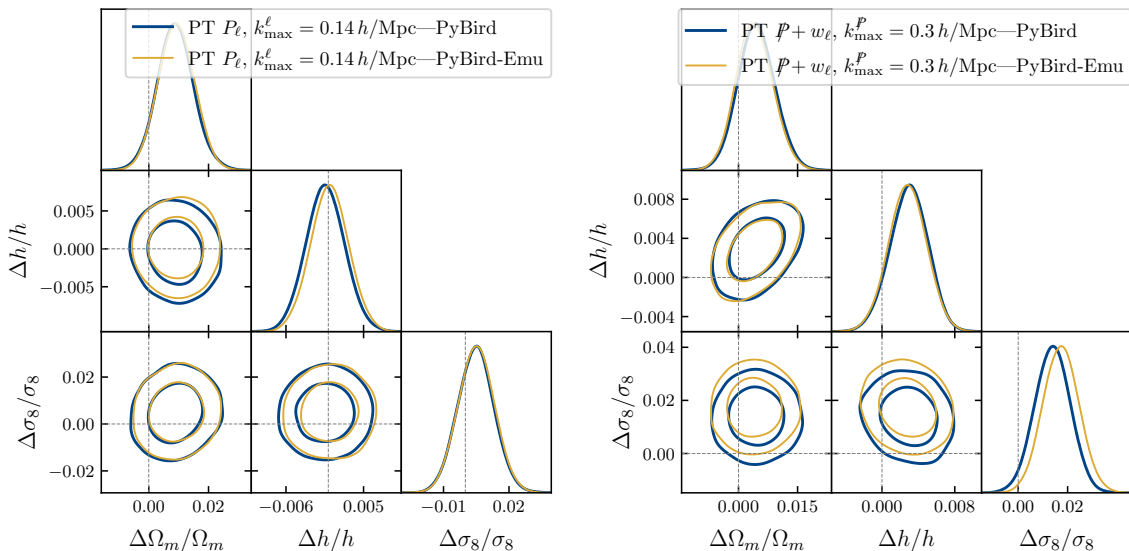


Figure 4. Comparison PyBird vs. PyBird-Emu on PT simulation data — 1D and 2D marginal posterior distributions of inferred Λ CDM parameters from the PT challenge simulations, with fixed ω_b and n_s . For all configurations, all parameters are recovered within $\sim 1\sigma$, with the truth shown in dashed line. The posteriors from PyBird (blue contours) and PyBird-Emu (yellow contours) agree at subpercent level ($< 0.3\%$ of the parameter values on 1D marginals) whether fit in multipoles P_ℓ up to $k_{\max}^\ell = 0.14 h \text{ Mpc}^{-1}$ (left panel) or in wedges $\dot{P} + w_\ell$ with \dot{P} analysed up to $k_{\max}^{\dot{P}} = 0.3 h \text{ Mpc}^{-1}$ (right panel). This validates the raw accuracy of PyBird-Emu in recovering cosmological parameters to high precision.

developed in refs. [128–133] using the FFT-based code `Rustico`³¹ [133], from the BOSS DR12 catalogs (v5) combined CMASS-LOWZ³² described in ref. [89]. The covariance is estimated as the scatter across measurements on the patchy mock suites [134]. We apply the Hartlap factor to correct bias from the finite number of mocks used to estimate the inverse covariance [135]. We correct for the Alcock-Paczynski effects and window functions as described in ref. [10] (with the window, which formalism was developed in refs. [136–138], measured using `fkpwin`³³ [138]), and fiber collisions following ref. [139]. The likelihood, priors, and posterior sampling are described above in the beginning of section 4.

We further make use of various external probes depending on the cosmological exploration. For all our BOSS analyses, we use a prior from Big-Bang Nucleosynthesis (BBN) experiments [140–143], that essentially constrain ω_b . We also sometimes use information from low-redshift supernovae from Pantheon+ [144], with magnitude either uncalibrated or calibrated with the SH_0ES distance ladder [145].

Due to its relatively modest volume (especially when compared to ongoing and upcoming Stage-4 surveys), the BOSS data serves as an ideal testbed for assessing PyBird-Emu’s coverage in both parameter and model space. Notably, BOSS has been widely used to constrain a broad range of cosmological models (see e.g., refs. [40, 44, 49–68]), including

³¹<https://github.com/hectorgil/Rustico>.

³²Publicly available at <https://data.sdss.org/sas/dr12/boos/lss/>.

³³<https://github.com/pierrexzy/fkpwin>.

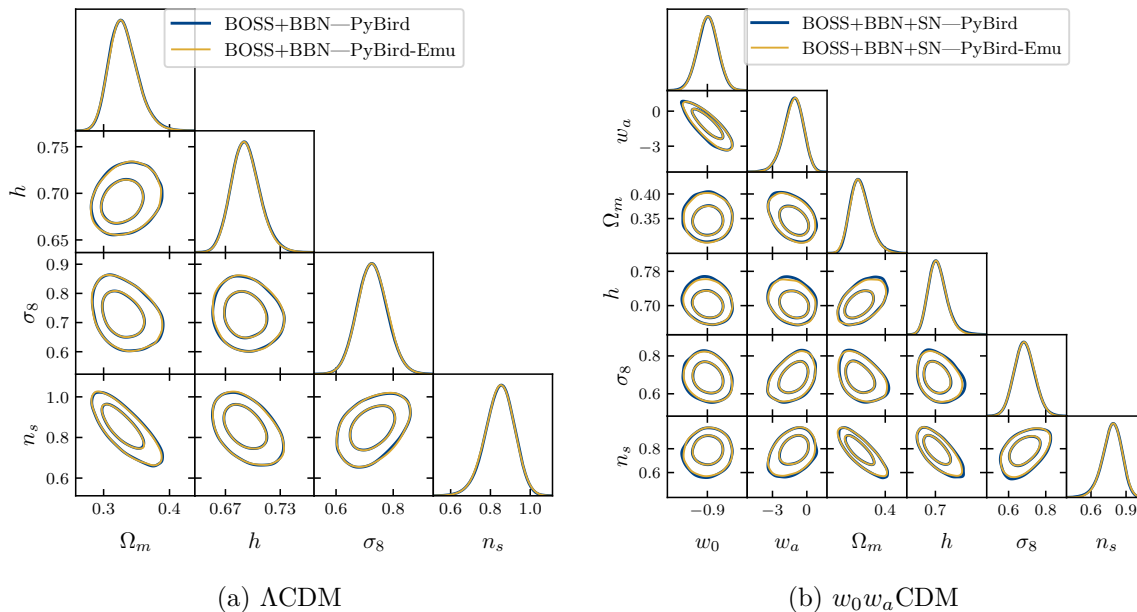


Figure 5. Comparison (I) PyBird vs. PyBird-Emu on BOSS data — 1D and 2D marginal posterior distributions of inferred parameters in Λ CDM (left panel) or w_0w_a CDM (right panel) from BOSS data, with a BBN prior on ω_b (not shown for clarity). The w_0w_a CDM fit also includes low-redshift supernova data from Pantheon+. Given the large parameter uncertainties and non-negligible shifts in the means with respect to the central values used for the emulator validation shown in table 1, these analysis setups stand as stringent parameter coverage tests for PyBird-Emu. The posteriors of PyBird (blue contours) and PyBird-Emu (yellow contours) are practically indistinguishable for both scenarios, validating the ability of PyBird-Emu in recovering cosmological parameters across wide parameter ranges.

models that yield non-trivial departures from Λ CDM. Then, the question we can ask is, do the posterior distributions of cosmological parameters obtained with PyBird-Emu and PyBird remain consistent in extended scenarios where cosmologies, potentially significantly different from Λ CDM, are probed within the ranges allowed by BOSS?

w_0w_a CDM. As a first step beyond Λ CDM, we consider a two-parameter extensions w_0w_a CDM. Since this model is included in the cosmology bank informing the $\log P_{\text{in}}$ prior of the emulator training set, this serves primarily as a sanity check rather than a challenging test. BOSS alone covers only a limited range of redshifts. In order to get meaningful constraints in w_0w_a CDM, we combine the BOSS likelihood with low-redshift data from Pantheon+ supernovae, with their absolute magnitude left uncalibrated. Results of the comparison are shown alongside Λ CDM ones in figures 5(a) and 5(b), where the posteriors from PyBird and PyBird-Emu are virtually indistinguishable, validating the accuracy of PyBird-Emu over a wide range of parameters, given the wide size of the uncertainties.

We now turn to two additional models: early dark energy (EDE) [146] and self-interacting neutrinos (SI ν). These scenarios were not included in the design of the emulator input coverage, and thus it is not guaranteed that the linear power spectra preferred by the data for these models lie within the emulator’s training distribution. These fits therefore test the robustness

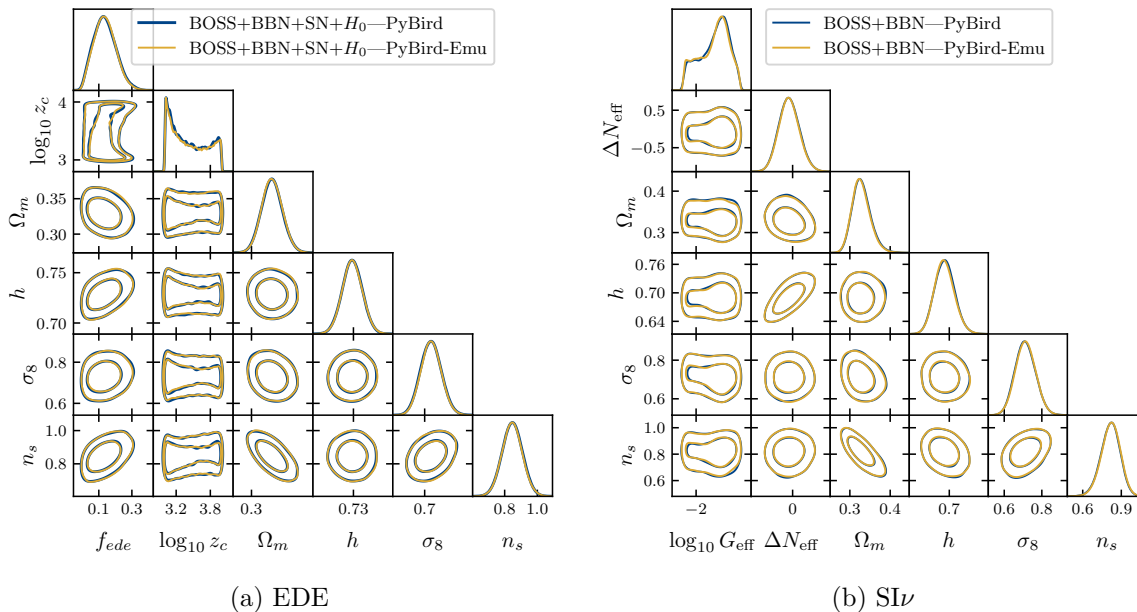


Figure 6. Comparison (II) PyBird vs. PyBird-Emu on BOSS data — 1D and 2D marginal posterior distributions of inferred parameters in EDE (*left panel*) or SI ν (*right panel*) from BOSS data, with a BBN prior on ω_b (not shown for clarity). The EDE fit also includes low-redshift supernova data from Pantheon+ with magnitude calibrated from the SH_0 ES distance ladder, implying a large H_0 compared to its preferred value in Λ CDM (without SH_0 ES). We restrict $\log_{10} G \in [-2.5, -0.3]$ to probe exclusively the preferred strongly-interacting regime by BOSS. These analysis setups thus provide far-away cosmologies to test the accuracy of PyBird-Emu. Crucially, EDE and SI ν were not used when designing the emulator’s input coverage. The posteriors of PyBird (*blue contours*) and PyBird-Emu (*yellow contours*) are practically indistinguishable for all scenarios, validating the flexibility of PyBird-Emu in recovering cosmological parameters across model space.

of our emulator design described in section 2, which aims to ensure broad and general coverage in $\log P_{\text{lin}}$ space, avoiding out-of-distribution failures.

Early dark energy. Designed to solve the Hubble tension (see e.g., ref. [147]), EDE introduces a new component that behaves like dark energy around the time of recombination, that subsequently decays, allowing the universe to follow the standard thermal history. The injection of EDE increases the physical sound horizon seen in the cosmic microwave background (CMB), thereby reducing the comoving distance from us to the CMB, and thus increasing H_0 accordingly, in order to keep the well-measured angular acoustic scale constant (see e.g., ref. [148] for refined details on subleading effects to account when modifying the sound horizon as early-time solution attempts to the Hubble tension). Considering axion-like EDE [147], three extra parameters are introduced with respect to Λ CDM: z_c , the redshift at which EDE is injected, f_{ede} , the fraction of EDE injected at z_c , and the initial axion-like pseudo-scalar field value ϕ_i (fixing $n = 3$). Notably, when fit in combination with SH_0 ES and Pantheon+ [144, 145], a large value of H_0 is preferred in this model compared to the preferred value in Λ CDM by BOSS [67], together with a non-zero f_{ede} at $\sim 2\sigma$. Whether EDE remains a viable solutions to the H_0 tension is currently under debate, especially in

light of new cosmological data (see e.g., refs. [65–67, 149–151]). As far as we are concerned here, EDE provides a non-trivial example of the cosmology-independence of our emulator. For this test, we fit the BOSS data together with Pantheon+ supernovae with magnitude calibrated with the SH_0ES distance ladder, and a prior from BBN experiments. For this analysis, we use `AxiCLASS`³⁴ [152, 153], a modified version of `CLASS`. The comparison between results from `PyBird` and `PyBird-JAX` is shown in figure 6(a), where no visible shift between the posteriors can be seen.

Self-interacting neutrinos. The other extension we consider is a scenario in which one or more of the three left-handed neutrinos display a non-standard self-interactions (see e.g., ref. [154] for a review). In presence of self-interactions, the onset of their free-streaming is delayed with respect to their decoupling with the Standard Model bath (around temperature $T \sim 1$ MeV), thereby suppressing the anisotropic stress sourcing the difference of the metric potentials. Such effect is partially degenerate with the presence of additional relativistic species, parametrised by ΔN_{eff} . Intriguingly, hints of strongly interacting neutrinos ($\log_{10}(G_{\text{eff}}/\text{MeV}^{-2}) \sim -1.5$) have been found in the BOSS data [155, 156], that however vanish once combined with CMB [157]. Whether the so-called strongly-interacting regime remains a viable description of the cosmological data does not matter for our concern here. Instead, we use the opportunity to test our emulator on far-away cosmologies explicitly not represented when building the training set of our emulator. For this test, we fit the BOSS data together with a BBN prior, and use a flat prior on $\log_{10}(G_{\text{eff}}/\text{MeV}^{-2})$ of $[-2.5, -0.3]$ to probe the preferred strongly-interacting mode [155] (as corresponding to far-away cosmologies useful to test our emulator accuracy). We also vary the number of extra relativistic species ΔN_{eff} , and fix the neutrino mass to minimal, $m_\nu = 0.06$ eV. For this analysis, we use `class_interacting_neutrinos`,³⁵ a modified version of `CLASS`. Again, no shift are visible between the posteriors obtained with `PyBird` and `PyBird-Emu` shown in figure 6(b) for this model, thus validating the accuracy of `PyBird-Emu` to recover parameters across a wide range of cosmologies.

4.3 Stage-4 LSS survey forecast

In this section, we present a forecast for a Stage-4 LSS survey, using specifications inspired by DESI Year 6, extrapolated from refs. [8, 158]. Details on the generation of the synthetic mock data are provided in our companion paper [1], with survey characteristics summarised in table 2 therein. The forecast includes 12 EFT parameters per redshift bin across 7 bins, resulting in a total of 84 nuisance parameters. Of these, 9 out of 12 EFT parameters enter the model linearly and are analytically marginalised following appendix D of ref. [1], leaving 21 nuisance parameters to be sampled numerically in addition to the cosmological parameters. We focus on ΛCDM , fixing ω_b and n_s to their fiducial values for simplicity. The likelihood, model, and priors are the same as those described at the beginning of section 4. For this forecast, we use `PyBird-Emu` together with our modified version of `Symbolic-Pk Boltzmann` emulator [30, 95] to enable a fast, fully differentiable pipeline. For reference, a

³⁴<https://github.com/PoulinV/AxiCLASS>.

³⁵https://github.com/PoulinV/class_interacting_neutrinos.

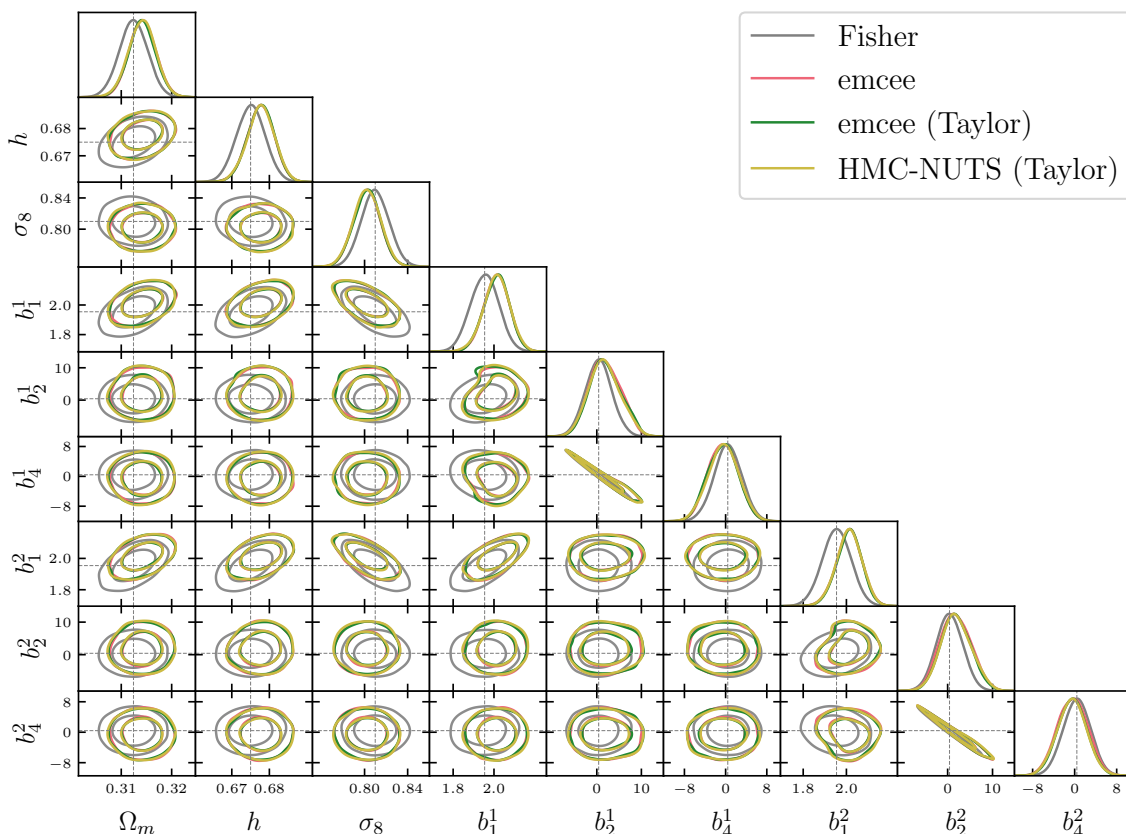


Figure 7. Sampler accuracy with PyBird-JAX for Stage-4 LSS survey forecast — 1D and 2D marginal posterior distributions of inferred Λ CDM parameters (with ω_b and n_s fixed) and some representative EFT parameters from DESI Year 6 mock data, obtained using various sampling strategies: `emcee` with PyBird-JAX (red contours), `emcee` with PyBird-Taylor (green contours), and HMC-NUTS with PyBird-Taylor (yellow contours). All results agree within convergence. The Fisher forecast (grey contours) and the fiducial values (dashed lines) are shown for comparison.

Sampler	ESS/s	Acceptance rate
<code>emcee</code> sequential	1.3	19%
<code>emcee</code> vectorised (batch 48)	3.4	19%
HMC-NUTS	2.2	90%

Table 4. Sampler performance with PyBird-JAX for Stage-4 LSS survey forecast — Effective sample size per seconds (ESS/s) and acceptance rate of MCMC chains from ensemble sampling with `emcee` in sequential or vectorised mode, or gradient-based sampling with HMC-NUTS. These performances are achieved on one GPU core using PyBird-Taylor. For this modest 24-dimensional, near-Gaussian posterior, `emcee` achieves a decent acceptance rate and delivers the best performance when used with vectorised PyBird-JAX. Although HMC-NUTS attains a significantly higher acceptance, performance gain is limited in this low-dimensional setting by the additional computational cost of gradient evaluations, which are approximately three times more expensive per likelihood call.

single likelihood evaluation using `PyBird-Emu` takes approximately 1 ms on a GPU core, with gradient evaluations taking about three times longer.

While previous analyses relied on the Metropolis-Hastings algorithm implemented in `MontePython` [103, 104], here we take the opportunity to showcase alternative samplers that leverage the new features of `PyBird-JAX`. These include the Fisher matrix and the gradient-based sampler `NUTS-HMC` [99], both making use of AD, and the ensemble sampler `emcee` [96], benefiting from vectorisation. We also compare results obtained using the Taylor expansion of the model, with derivatives computed via AD. For details on these `JAX`-based features, see sections 3.1 and 3.2. Figure 7 compares the cosmological posteriors obtained with `PyBird-Emu` across the different sampling strategies. Table 4 presents the effective sample size per second (ESS/s) for each sampler under various analysis configurations.

Taylor expansion of LSS observables. The posteriors obtained using the Taylor expansion of the model — computed to third order in derivatives and referred to as `PyBird-Taylor`— are in excellent agreement with those obtained without the expansion. The same holds true for BOSS data (not shown). Remarkably, `PyBird-Emu` runs in approximately the same time as `PyBird-Taylor`, being only about 0-40% slower depending on the input batch size. The speed advantage diminishes rapidly as the batch size increases, with virtually no gain observed for batches of 10 inputs or more (on the hardware used in this test). In more realistic analysis settings that include observational effects, `PyBird-Taylor` may offer greater speed-up. It is worth noting, however, that computing the derivatives for `PyBird-Taylor` requires a non-negligible pre-computation time, which is about 150 seconds for our DESI mock survey, amounting to roughly 10% of the total sampling walltime. Overall, it represents a valuable alternative for fast and accurate inference, although we emphasise the importance of validating its accuracy in beyond- Λ CDM scenarios.

Sampler comparison. We compare three posterior sampling strategies: sequential and vectorised ensemble sampling using `emcee`, and gradient-based sampling with `HMC-NUTS`. To monitor convergence, we use the integrated autocorrelation time τ , requiring that the number of samples satisfy $N_{\text{samples}} \gtrsim 50\tau$. The effective sample size (ESS) is computed using `GetDist` [113] as

$$\text{ESS} = \frac{N_{\text{samples}}}{2\tau}. \quad (4.2)$$

We then compute the ESS per second (ESS/s) by dividing by the total sampling walltime. Importantly, we exclude the burn-in period for `emcee` and the warm-up phase for `HMC-NUTS`, during which hyperparameters are tuned. Both take approximately an extra timing of about 30% of the sampling one. As shown in table 4, `HMC-NUTS` outperforms sequential `emcee` in our DESI mock analysis by a factor of 1.7. This improvement is largely due to its higher acceptance rate (90% for `HMC-NUTS` vs. 20% for `emcee`), though tempered by the additional computational cost of gradient evaluations (roughly $3\times$ slower per likelihood call). The expected speed-up of about 1.5 is consistent with the observed factor of 1.7. Vectorised `emcee`, using 48 walkers in parallel, achieves the highest ESS/s of 3.4. This gain aligns with the $\sim 3\times$ speed-up in model evaluation enabled by processing input batches via vectorised `PyBird-Emu`.

Based on this specific case study, we conclude that vectorised ensemble sampling is the most efficient strategy for posterior estimation with `PyBird-Emu`, as long as the acceptance rate stays above $\sim 10\%$, assuming that the acceptance rate for `HMC-NUTS` stays at $\sim 90\%$. Other samplers available in `PyBird-JAX` and described in section 3 were excluded from this comparison, as we do not expect them to offer additional insights beyond the representative strategies analysed here. For example, while `MC-HMC` may achieve a higher acceptance rate than `HMC-NUTS`, it is unlikely to yield a significant overall speed-up — particularly given potentially long warm-up time required for hyperparameter tuning. In practice, requiring $N_{\text{samples}} \gtrsim 50 \tau$ for each parameter, the representative sampling walltime for our DESI Year 6 forecast ranges from about 7 to 20 seconds (on our GPU hardware), depending on the sampling strategy. Therefore, in practice, as long as the acceptance rate remains reasonable, any sampler will converge in a decent amount of time for fiducial LSS analyses; pre-computing time can therefore not be neglected when choosing the best sampling strategy.

5 Conclusions

We have presented `PyBird-JAX` and its model-independent, NN-based emulator `PyBird-Emu` as a new, fully differentiable pipeline for LSS analyses. By rewriting the `PyBird EFTofLSS` implementation in `JAX` and emulating the costly loop and IR-resummation terms with NNs, we obtain *order-of-magnitude* speed-ups: one-loop galaxy power-spectrum multipoles can be evaluated in $\mathcal{O}(\text{ms})$ on a CPU and $\mathcal{O}(0.1 \text{ ms})$ on a single GPU, a gain of $10^3\text{--}10^4$ relative to the original code.

Crucially, `PyBird-Emu` is *cosmology-independent*. Instead of learning in the space of cosmological parameters, the emulator is trained on the spline coefficients of the linear matter power spectrum on a pre-optimised fixed grid in ks . To ensure some levels of smoothness while retaining flexibility, we use a Gaussian copula distribution generalising an initial extended cosmological bank of linear power spectrum. This design delivers wide coverage and negligible bias: we have validated that `PyBird-Emu` recovers unbiased posteriors, consistent with brute-force `PyBird`, even for challenging cases like Early Dark Energy or self-interacting neutrinos that were not explicitly represented in the training set. It would be valuable to explore if other decomposition bases and NN architectures are well suited for this problem, eventually achieving higher accuracy and/or flexibility. We stress, however, that `PyBird-Emu` is valid to explore cosmologies for which the current `EFTofLSS` implementation of `PyBird` itself is a valid description: Galilean-invariant single adiabatic clock with exact-time dependence. For models modifying the structure of the `EFTofLSS` kernels nontrivially (see e.g., ref. [159]), `PyBird-Emu` would need to be trained again.

Because `PyBird-JAX` is written entirely in `JAX`, users inherit AD, JIT compilation, and vectorisation *for free*. Machine-precision gradients and Hessians with AD enable HMC, Fisher forecasts, and gradient-based optimisers, while `vmap` allows hundreds of MCMC walkers to be evaluated in parallel. Native GPU support lets the same `Python` code scale effortlessly with emerging hardware. As a first non-trivial application of `PyBird-JAX`, we demonstrate that volume projection effects are mitigated in LSS analyses once defining our expectation values with respect to non-flat measures computed with the power of AD in our companion paper [1].

These features position `PyBird-JAX` as a ready-made engine for Stage-4 LSS surveys such as DESI and *Euclid*, where likelihoods will involve $\mathcal{O}(10^2)$ nuisance parameters and billions of modes. Looking ahead, the same strategy can accelerate higher-order perturbative predictions and complementary observables (e.g., one-loop bispectrum [4, 13], configuration-space correlation functions [42], projected angular functions [39], multi-tracer spectra [160, 161]). In summary, together with joint, parallel efforts such as `effort.jl` [81], `PyBird-JAX` brings the EFTofLSS into the era of high-performance, differentiable computing. It couples the flexibility of `Python` to the throughput of GPUs, turning previously prohibitive analyses into routine tasks and ensuring that the community can fully exploit forthcoming LSS data.

Acknowledgments

We thank Marco Bonici, Arnaud de Mattia, Martin Kärcher, and Alexandre Refregier for discussions. We thank Takahiro Nishimichi for providing the data of the PT Challenge simulations. PZ acknowledges support from Fondazione Cariplo under the grant No. 2023-1205. Some computations were performed on the *Euler* high-performance computing cluster at ETH Zürich, and we acknowledge assistance from the support team.

A Vectorised FFTLog-based IR-resummation

In this appendix, we review the IR-resummation in `PyBird` [40], highlighting a key difference in its implementation with its first release: the vectorisation in one single 1D array of the terms appearing in the nested sums of eq. (A.6). Compared to naive nested `for` loops, this results in a factor $\sim \mathcal{O}(10)$ acceleration for the evaluation of the IR-resummation, which now takes $\sim 1/10$ of the loop evaluation time. Overall, `PyBird` now takes about 500 ms (with ~ 60 ms spent on the IR-resummation) to evaluate one power spectrum as detailed in section 2.2.

Contrary to most alternative codes (with the exception of `Velocileptors` [70]), `PyBird` implement the full Lagrangian IR-resummation first derived in refs. [162, 163] and further developed in refs. [40, 127, 164], and thus does not rely on the wiggle-no-wiggle split of the linear power spectrum. The j -th loop term of the multipoles of the IR-resummed power spectrum at the N -loop order, $P_\ell^j(k)|_N$, can be written as an integral over the unresummed correlation function multipoles $\xi_{\ell'}^j(s)$ modulated by a convolution matrix $Q_{\ell\ell'}||_{N-j}(k, s)$ (see e.g., [164]),

$$P_\ell^j(k)|_N = \sum_{\ell'}^{N_\ell} 4\pi(-i)^{\ell'} \int ds s^2 Q_{\ell\ell'}||_{N-j}(k, s) \xi_{\ell'}^j(s). \quad (\text{A.1})$$

We remind that `PyBird` computes swiftly and stably the correlation function analytically using the FFTLog [40, 42]. Here, $Q_{\ell\ell'}||_{N-j}(k, s)$ represents an exponential damping $F||_{N-j}$ resumming the large, $\mathcal{O}(1)$ -long-wavelength displacements, weighted by an exponential phase factor from the Fourier transform together with Legendre polynomials $\mathcal{L}_\ell(\hat{k} \cdot \hat{n})\mathcal{L}_{\ell'}(\hat{s} \cdot \hat{n})$ from the multipole expansions of the redshift-space power spectrum and correlation function (where \hat{n} is the line-of-sight direction), angle-averaged in both Fourier and configuration space,

$$Q_{\ell\ell'}||_{N-j}(k, s) = \frac{2\ell + 1}{2} \int_{-1}^1 d(\hat{k} \cdot \hat{n}) \int d^2\hat{s} e^{iks(\hat{k} \cdot \hat{s})} F||_{N-j}(k, s, \hat{k} \cdot \hat{s}) \mathcal{L}_\ell(\hat{k} \cdot \hat{n}) \mathcal{L}_{\ell'}(\hat{s} \cdot \hat{n}). \quad (\text{A.2})$$

At one-loop order, the linear and one-loop terms are resummed using respectively $F||_1$ and $F||_0$, defined as (see e.g., [164])

$$F||_0 = \exp \left[-\frac{1}{2} \langle (\mathbf{k} \cdot \Delta\boldsymbol{\psi})^2 \rangle \right], \quad F||_1 = F||_0 \times \left[1 - \frac{1}{2} \langle (\mathbf{k} \cdot \Delta\boldsymbol{\psi})^2 \rangle \right]. \quad (\text{A.3})$$

Here, from the two-point function of the linear displacements $\boldsymbol{\psi}$, we have

$$\langle (\mathbf{k} \cdot \Delta\boldsymbol{\psi})^2 \rangle = k_i k_j (X_0(s) \delta_{ij} + X_2(s) \hat{s}_i \hat{s}_j), \quad (\text{A.4})$$

where the IR-filters are defined as

$$X_0(s) = \frac{1}{3} \int^{\Lambda_{\text{IR}}} \frac{dp}{2\pi^2} P_{\text{lin}}(p) [1 - j_0(ps) - j_2(ps)], \quad X_2(s) = \int^{\Lambda_{\text{IR}}} \frac{dp}{2\pi^2} P_{\text{lin}}(p) j_2(ps). \quad (\text{A.5})$$

To make the evaluation practical, the trick is to Taylor expand the exponential factors in eq. (A.3) to a sufficiently high order M ($M \sim 20$ in `PyBird`) so that the resulting truncated Taylor series are a good numerical approximation of the full exponential [40]. We can then simplify $Q_{\ell\ell'}|_{N-j}(k, s)$ in eq. (A.2) following ref. [127]. The idea is to expand the exponential phase factor in the Legendre decomposition of a plane wave, absorbing the powers of $\hat{k} \cdot \hat{s}$ as derivatives acting on it, such that the angular integrals over $d^2\hat{s}$ can be performed using the orthogonality of Legendre polynomials. The resulting integrals over $d(\hat{k} \cdot \hat{n})$ of two Legendre times powers of $\hat{k} \cdot \hat{n}$ can be performed analytically. At the end of the day, the IR-resummed power spectrum implemented in `PyBird` reads [40]

$$P_\ell(k)|_N = P_\ell(k) + \sum_{j=0}^N \sum_{\ell'}^{N_\ell} \sum_{m=1}^M \sum_{\alpha}^{N_\alpha} 4\pi (-i)^{\ell'} k^{2m} \mathcal{Q}_{||N-j}^{\ell\ell', m, \alpha}(f) \int ds s^2 [X(s)]^m \xi_{\ell'}^j(s) j_\alpha(ks), \quad (\text{A.6})$$

where $[X]^m$ represents either X_0^m , or $X_0^{m-1}X_2$, etc., and $\mathcal{Q}_{||N-j}^{\ell\ell', m, \alpha}(f)$ are simple numerical factors that depends on the growth rate f . The IR-corrections that upgrade the non-resummed power spectrum P_ℓ to the resummed one $P_\ell|_N$ appearing in eq. (A.6) are spherical Bessel transforms (SBTs) of the correlation function multipoles filtered by $[X]^m$, that can be computed swiftly using the `FFTLog` [165]. Note that we also use the `FFTLog` to compute X_0 and X_2 defined in eq. (A.5), making our overall implementation of the IR-resummation stable numerically and fast. To make the algorithm efficient, instead of looping over the sums appearing in eq. (A.6), we perform a unique SBT³⁶ on a multidimensional array of size $(N_\ell, N_{\text{lin}} + N_{\text{loop}}, N_\ell, M, N_\alpha)$, where N_{lin} and N_{loop} represent the total number of EFT parameter-independent linear and loop terms, respectively. This is much faster than performing $N_\ell \times (N_{\text{lin}} + N_{\text{loop}}) \times N_\ell \times M \times N_\alpha$ sequential SBTs, as it takes advantages of `BLAS` routines and avoids slow `Python for` loops. The multidimensional array is then contracted along the relevant axis with $\mathcal{Q}_{||N-j}^{\ell\ell', m, \alpha}(f)$, and finally the IR-correction terms are added to each (unresummed) linear and loop contributions. Note that as being fully vectorised, `PyBird` implementation is straightforwardly ported to `JAX`.

To close this appendix, we take the opportunity to study the sensitivity of inferred cosmological parameters on the IR-cutoff Λ_{IR} entering in eq. (A.5). To do so, we use the

³⁶Implemented at <https://github.com/pierrexzyz/fftlog>.

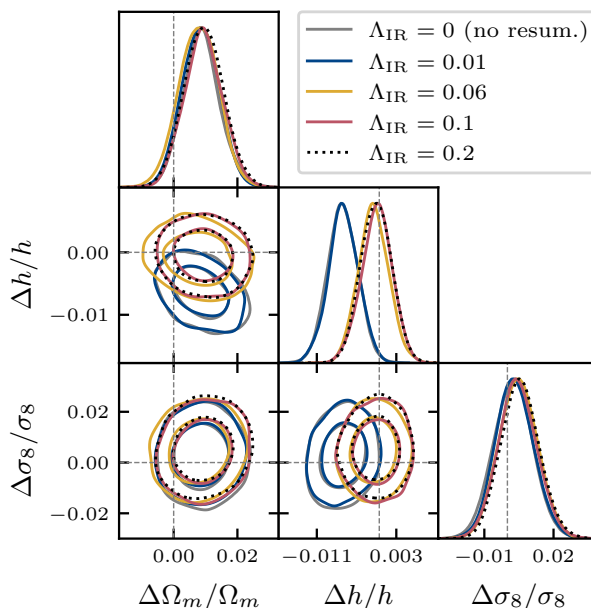


Figure 8. Λ_{IR} -sensitivity on PT challenge data — 1D and 2D marginal posterior distributions of inferred Λ CDM parameters from the PT challenge simulations, with fixed ω_b and n_s , as functions of the resummation cutoff Λ_{IR} . In the limit $\Lambda_{\text{IR}} \rightarrow 0$, we recover the results using unresummed predictions, while for $\Lambda_{\text{IR}} > 0.06 h \text{ Mpc}^{-1}$, all results display good agreement. All λ_{IR} s are in $h \text{ Mpc}^{-1}$.

PT challenge simulations presented in section 4.1, that we fit with various values for Λ_{IR} (implemented as a Gaussian damping rather than a sharp cutoff such that X_0 and X_2 can be computed using the FFTLog) in figure 8. For Λ_{IR} close to 0, we recover the same results obtained using unresummed predictions, as it should. We see that from $\Lambda_{\text{IR}} \sim 0.06 h \text{ Mpc}^{-1}$ and above, the posteriors of the inferred cosmological parameters remains fairly consistent with each others. Based on this comparison, we set $\Lambda_{\text{IR}} = 0.1 h \text{ Mpc}^{-1}$ by default in PyBird.

B Emulator accuracy up to $k_{\text{max}} = 0.3 h \text{ Mpc}^{-1}$

As mentioned in the main text, our emulators within PyBird are trained to operate up to a higher k_{max} than the typical value used in LSS analyses, namely $k_{\text{max}} = 0.2 h \text{ Mpc}^{-1}$ (as adopted in our analyses in section 4). Nonetheless, it is informative to examine emulator performance at higher k values for two main reasons. First, the theoretical power spectrum is typically convolved with a window function matrix (see, e.g., ref. [138]), which has finite support in k . As a result, the convolved power spectrum at a given k_* receives some contributions from modes with $k > k_*$. Second, analyses are sometimes performed in wedges rather than multipoles. In particular, for wedges constructed such that $\mu \sim 0$, the slowly converging series expansion of redshift-space distortions is suppressed. This leaves a more rapidly converging series for that wedge, effectively controlled by the EFTofLSS parameters of the real-space expansion, that allow an extended k -reach (see section 4.1 for further discussions). In figure 9, we present the emulator residual errors up to $k_{\text{max}} = 0.3 h \text{ Mpc}^{-1}$. While the emulator’s accuracy degrades somewhat at higher k , 95% of the validation samples

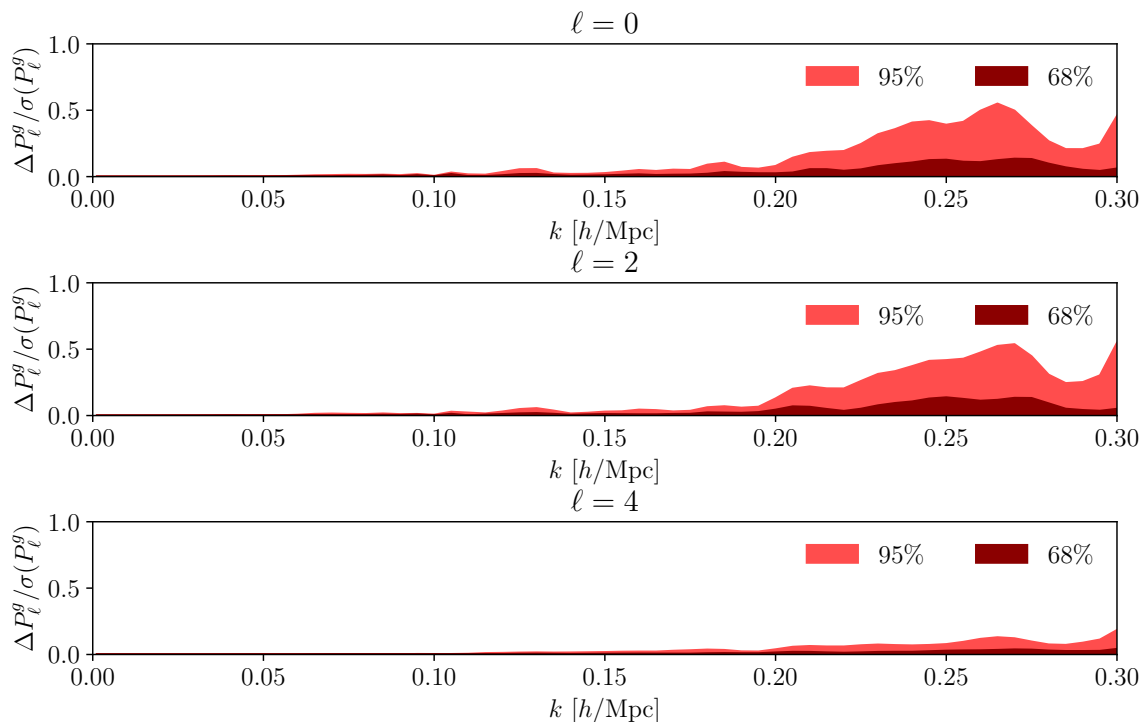


Figure 9. 68% and 95% quantiles of the differences in the galaxy power spectrum multipoles across scales up to $k_{\max} = 0.3 \text{ h Mpc}^{-1}$, computed with PyBird. The comparison is between the NN-based emulator predictions and the full calculations, evaluated over the independent validation set described in section 2.2. Errors are shown relative to representative uncertainties expected for Stage-4 LSS surveys.

remain within 0.5σ relative to the representative Stage-4 LSS uncertainties, demonstrating that the emulator still performs reasonably well in this extended regime. We note that the volume used for these uncertainties is at least 5 times larger than the typical per redshift bin volume for DESI.

References

- [1] A. Reeves, P. Zhang and H. Zheng, *Debiasing inference in large-scale structure with non-flat volume measures*, [arXiv:2507.20991](#) [INSPIRE].
- [2] V. Desjacques, D. Jeong and F. Schmidt, *Large-Scale Galaxy Bias*, *Phys. Rept.* **733** (2018) 1 [[arXiv:1611.09787](#)] [INSPIRE].
- [3] A. Perko, L. Senatore, E. Jennings and R.H. Wechsler, *Biased Tracers in Redshift Space in the EFT of Large-Scale Structure*, [arXiv:1610.09321](#) [INSPIRE].
- [4] G. D’Amico et al., *The one-loop bispectrum of galaxies in redshift space from the Effective Field Theory of Large-Scale Structure*, *JCAP* **07** (2024) 041 [[arXiv:2211.17130](#)] [INSPIRE].
- [5] O.H.E. Philcox et al., *Cosmology with the redshift-space galaxy bispectrum monopole at one-loop order*, *Phys. Rev. D* **106** (2022) 043530 [[arXiv:2206.02800](#)] [INSPIRE].
- [6] F. Schmidt, *An n -th order Lagrangian Forward Model for Large-Scale Structure*, *JCAP* **04** (2021) 033 [[arXiv:2012.09837](#)] [INSPIRE].

- [7] Y. Donath, M. Lewandowski and L. Senatore, *Direct signatures of the formation time of galaxies*, *Phys. Rev. D* **109** (2024) 123510 [[arXiv:2307.11409](#)] [[INSPIRE](#)].
- [8] DESI collaboration, *DESI 2024 II: sample definitions, characteristics, and two-point clustering statistics*, *JCAP* **07** (2025) 017 [[arXiv:2411.12020](#)] [[INSPIRE](#)].
- [9] EUCLID collaboration, *Euclid. I. Overview of the Euclid mission*, *Astron. Astrophys.* **697** (2025) A1 [[arXiv:2405.13491](#)] [[INSPIRE](#)].
- [10] G. D’Amico et al., *The Cosmological Analysis of the SDSS/BOSS data from the Effective Field Theory of Large-Scale Structure*, *JCAP* **05** (2020) 005 [[arXiv:1909.05271](#)] [[INSPIRE](#)].
- [11] M.M. Ivanov, M. Simonović and M. Zaldarriaga, *Cosmological Parameters from the BOSS Galaxy Power Spectrum*, *JCAP* **05** (2020) 042 [[arXiv:1909.05277](#)] [[INSPIRE](#)].
- [12] S.-F. Chen, Z. Vlah and M. White, *A new analysis of galaxy 2-point functions in the BOSS survey, including full-shape information and post-reconstruction BAO*, *JCAP* **02** (2022) 008 [[arXiv:2110.05530](#)] [[INSPIRE](#)].
- [13] G. D’Amico et al., *The BOSS bispectrum analysis at one loop from the Effective Field Theory of Large-Scale Structure*, *JCAP* **05** (2024) 059 [[arXiv:2206.08327](#)] [[INSPIRE](#)].
- [14] DESI collaboration, *DESI 2024 V: Full-Shape galaxy clustering from galaxies and quasars*, *JCAP* **09** (2025) 008 [[arXiv:2411.12021](#)] [[INSPIRE](#)].
- [15] DESI collaboration, *DESI 2024 VII: cosmological constraints from the full-shape modeling of clustering measurements*, *JCAP* **07** (2025) 028 [[arXiv:2411.12022](#)] [[INSPIRE](#)].
- [16] A.N. Taylor and T.D. Kitching, *Analytic Methods for Cosmological Likelihoods*, *Mon. Not. Roy. Astron. Soc.* **408** (2010) 865 [[arXiv:1003.1136](#)] [[INSPIRE](#)].
- [17] M. Cataneo, S. Foreman and L. Senatore, *Efficient exploration of cosmology dependence in the EFT of LSS*, *JCAP* **04** (2017) 026 [[arXiv:1606.03633](#)] [[INSPIRE](#)].
- [18] T. Colas et al., *Efficient Cosmological Analysis of the SDSS/BOSS data from the Effective Field Theory of Large-Scale Structure*, *JCAP* **06** (2020) 001 [[arXiv:1909.07951](#)] [[INSPIRE](#)].
- [19] Y. Lai et al., *A comparison between ShapeFit compression and Full-Modelling method with PyBird for DESI 2024 and beyond*, *JCAP* **01** (2025) 139 [[arXiv:2404.07283](#)] [[INSPIRE](#)].
- [20] T. Auld, M. Bridges, M.P. Hobson and S.F. Gull, *Fast cosmological parameter estimation using neural networks*, *Mon. Not. Roy. Astron. Soc.* **376** (2007) L11 [[astro-ph/0608174](#)] [[INSPIRE](#)].
- [21] J. Albers et al., *CosmicNet. Part I. Physics-driven implementation of neural networks within Einstein-Boltzmann Solvers*, *JCAP* **09** (2019) 028 [[arXiv:1907.05764](#)] [[INSPIRE](#)].
- [22] M. Zennaro et al., *The BACCO simulation project: biased tracers in real space*, *Mon. Not. Roy. Astron. Soc.* **524** (2023) 2407 [[arXiv:2101.12187](#)] [[INSPIRE](#)].
- [23] G. Aricò, R.E. Angulo and M. Zennaro, *Accelerating Large-Scale-Structure data analyses by emulating Boltzmann solvers and Lagrangian Perturbation Theory*, [arXiv:2104.14568](#) [[DOI:10.12688/openreseurope.14310.2](#)] [[INSPIRE](#)].
- [24] A. Mootoivaloo, A.H. Jaffe, A.F. Heavens and F. Leclercq, *Kernel-based emulator for the 3D matter power spectrum from CLASS*, *Astron. Comput.* **38** (2022) 100508 [[arXiv:2105.02256](#)] [[INSPIRE](#)].
- [25] A. Spurio Mancini et al., *CosmoPower: emulating cosmological power spectra for accelerated Bayesian inference from next-generation surveys*, *Mon. Not. Roy. Astron. Soc.* **511** (2022) 1771 [[arXiv:2106.03846](#)] [[INSPIRE](#)].

- [26] A. Nygaard, E.B. Holm, S. Hannestad and T. Tram, *CONNECT: a neural network based framework for emulating cosmological observables and cosmological parameter inference*, *JCAP* **05** (2023) 025 [[arXiv:2205.15726](#)] [[INSPIRE](#)].
- [27] S. Günther et al., *CosmicNet II: emulating extended cosmologies with efficient and accurate neural networks*, *JCAP* **11** (2022) 035 [[arXiv:2207.05707](#)] [[INSPIRE](#)].
- [28] M. Bonici, L. Biggio, C. Carbone and L. Guzzo, *Fast emulation of two-point angular statistics for photometric galaxy surveys*, *Mon. Not. Roy. Astron. Soc.* **531** (2024) 4203 [[arXiv:2206.14208](#)] [[INSPIRE](#)].
- [29] A. Reeves et al., *12 × 2 pt combined probes: pipeline, neutrino mass, and data compression*, *JCAP* **01** (2024) 042 [[arXiv:2309.03258](#)] [[INSPIRE](#)].
- [30] D.J. Bartlett et al., *A precise symbolic emulator of the linear matter power spectrum*, *Astron. Astrophys.* **686** (2024) A209 [[arXiv:2311.15865](#)] [[INSPIRE](#)].
- [31] Z. Zhang, *A general polynomial emulator for cosmology via moment projection*, [arXiv:2507.02179](#) [[DOI:10.1093/mnras/staf2039](#)] [[INSPIRE](#)].
- [32] S. Günther et al., *OLÉ — Online Learning Emulation in cosmology*, *JCAP* **09** (2025) 059 [[arXiv:2503.13183](#)] [[INSPIRE](#)].
- [33] J.-E. Campagne et al., *JAX-COSMO: An End-to-End Differentiable and GPU Accelerated Cosmology Library*, *Open J. Astrophys.* **6** (2023) 1 [[arXiv:2302.05163](#)] [[INSPIRE](#)].
- [34] M. Bonici, F. Bianchini and J. Ruiz-Zapatero, *Capse.jl: efficient and auto-differentiable CMB power spectra emulation*, [arXiv:2307.14339](#) [[DOI:10.21105/astro.2307.14339](#)] [[INSPIRE](#)].
- [35] O. Hahn, F. List and N. Porqueres, *DISCO-DJ I: a differentiable Einstein-Boltzmann solver for cosmology*, *JCAP* **06** (2024) 063 [[arXiv:2311.03291](#)] [[INSPIRE](#)].
- [36] J. Ruiz-Zapatero et al., *LimberJack.jl: auto-differentiable methods for angular power spectra analyses*, [arXiv:2310.08306](#) [[DOI:10.21105/astro.2310.08306](#)] [[INSPIRE](#)].
- [37] L. Balkenhol, C. Trendafilova, K. Benabed and S. Galli, *candl: cosmic microwave background analysis with a differentiable likelihood*, *Astron. Astrophys.* **686** (2024) A10 [[arXiv:2401.13433](#)] [[INSPIRE](#)].
- [38] A. Reeves, A. Nicola and A. Refregier, *Tuning the cosmic instrument: robust cosmology through combined probes*, *JCAP* **06** (2025) 042 [[arXiv:2502.01722](#)] [[INSPIRE](#)].
- [39] L. Reymond, A. Reeves, P. Zhang and A. Refregier, *SwiftC_ℓ: fast differentiable angular power spectra beyond Limber*, [arXiv:2505.22718](#) [[INSPIRE](#)].
- [40] G. D’Amico, L. Senatore and P. Zhang, *Limits on w CDM from the EFTofLSS with the PyBird code*, *JCAP* **01** (2021) 006 [[arXiv:2003.07956](#)] [[INSPIRE](#)].
- [41] T. Nishimichi et al., *Blinded challenge for precision cosmology with large-scale structure: results from effective field theory for the redshift-space galaxy power spectrum*, *Phys. Rev. D* **102** (2020) 123541 [[arXiv:2003.08277](#)] [[INSPIRE](#)].
- [42] P. Zhang et al., *BOSS Correlation Function analysis from the Effective Field Theory of Large-Scale Structure*, *JCAP* **02** (2022) 036 [[arXiv:2110.07539](#)] [[INSPIRE](#)].
- [43] G. D’Amico, L. Senatore, P. Zhang and T. Nishimichi, *Taming redshift-space distortion effects in the EFTofLSS and its application to data*, *JCAP* **01** (2024) 037 [[arXiv:2110.00016](#)] [[INSPIRE](#)].
- [44] T. Simon, P. Zhang and V. Poulin, *Cosmological inference from the EFTofLSS: the eBOSS QSO full-shape analysis*, *JCAP* **07** (2023) 041 [[arXiv:2210.14931](#)] [[INSPIRE](#)].

- [45] R. Zhao et al., *A multitracers analysis for the eBOSS galaxy sample based on the effective field theory of large-scale structure*, *Mon. Not. Roy. Astron. Soc.* **532** (2024) 783 [[arXiv:2308.06206](#)] [[INSPIRE](#)].
- [46] T. Simon, P. Zhang, V. Poulin and T.L. Smith, *Consistency of effective field theory analyses of the BOSS power spectrum*, *Phys. Rev. D* **107** (2023) 123530 [[arXiv:2208.05929](#)] [[INSPIRE](#)].
- [47] M. Maus et al., *A comparison of effective field theory models of redshift space galaxy power spectra for DESI 2024 and future surveys*, *JCAP* **01** (2025) 134 [[arXiv:2404.07272](#)] [[INSPIRE](#)].
- [48] EUCLID collaboration, *Euclid preparation — XLIV. Modelling spectroscopic clustering on mildly nonlinear scales in beyond- Λ CDM models*, *Astron. Astrophys.* **689** (2024) A275 [[arXiv:2311.13529](#)] [[INSPIRE](#)].
- [49] S. Bottaro et al., *Unveiling Dark Forces with Measurements of the Large Scale Structure of the Universe*, *Phys. Rev. Lett.* **132** (2024) 201002 [[arXiv:2309.11496](#)] [[INSPIRE](#)].
- [50] A. Poudou et al., *Self-interacting neutrinos in light of recent CMB and LSS data*, *Phys. Rev. D* **112** (2025) 103535 [[arXiv:2503.10485](#)] [[INSPIRE](#)].
- [51] R. Calderón, T. Simon, A. Shafieloo and D.K. Hazra, *Primordial Features in light of the Effective Field Theory of Large Scale Structure*, [arXiv:2504.06183](#) [[INSPIRE](#)].
- [52] G. D’Amico, M. Lewandowski, L. Senatore and P. Zhang, *Limits on primordial non-Gaussianities from BOSS galaxy-clustering data*, *Phys. Rev. D* **111** (2025) 063514 [[arXiv:2201.11518](#)] [[INSPIRE](#)].
- [53] L. Piga et al., *Constraints on modified gravity from the BOSS galaxy survey*, *JCAP* **04** (2023) 038 [[arXiv:2211.12523](#)] [[INSPIRE](#)].
- [54] S. Spaar and P. Zhang, *Cosmological constraints from combined probes with the three-point statistics of galaxies at one-loop precision*, *Phys. Rev. D* **111** (2025) 023529 [[arXiv:2312.15164](#)] [[INSPIRE](#)].
- [55] P. Taule et al., *Constraints on dark energy and modified gravity from the BOSS Full-Shape and DESI BAO data*, *JCAP* **03** (2025) 036 [[arXiv:2409.08971](#)] [[INSPIRE](#)].
- [56] Z. Lu, T. Simon and P. Zhang, *Preference for evolving dark energy in light of the galaxy bispectrum*, [arXiv:2503.04602](#) [[INSPIRE](#)].
- [57] A. Glanville, C. Howlett and T.M. Davis, *Full-shape galaxy power spectra and the curvature tension*, *Mon. Not. Roy. Astron. Soc.* **517** (2022) 3087 [[arXiv:2205.05892](#)] [[INSPIRE](#)].
- [58] A. Laguè et al., *Constraining ultralight axions with galaxy surveys*, *JCAP* **01** (2022) 049 [[arXiv:2104.07802](#)] [[INSPIRE](#)].
- [59] M. Gonzalez, M.P. Hertzberg and F. Rompineve, *Ultralight Scalar Decay and the Hubble Tension*, *JCAP* **10** (2020) 028 [[arXiv:2006.13959](#)] [[INSPIRE](#)].
- [60] I.J. Allali, M.P. Hertzberg and F. Rompineve, *Dark sector to restore cosmological concordance*, *Phys. Rev. D* **104** (2021) L081303 [[arXiv:2104.12798](#)] [[INSPIRE](#)].
- [61] I.J. Allali, F. Rompineve and M.P. Hertzberg, *Dark sectors with mass thresholds face cosmological datasets*, *Phys. Rev. D* **108** (2023) 023527 [[arXiv:2305.14166](#)] [[INSPIRE](#)].
- [62] T. Simon et al., *Constraining decaying dark matter with BOSS data and the effective field theory of large-scale structures*, *Phys. Rev. D* **106** (2022) 023516 [[arXiv:2203.07440](#)] [[INSPIRE](#)].
- [63] F. Niedermann and M.S. Sloth, *New Early Dark Energy is compatible with current LSS data*, *Phys. Rev. D* **103** (2021) 103537 [[arXiv:2009.00006](#)] [[INSPIRE](#)].

- [64] J.S. Cruz et al., *Profiling cold new early dark energy*, *Phys. Rev. D* **108** (2023) 023518 [[arXiv:2302.07934](#)] [[INSPIRE](#)].
- [65] G. D’Amico, L. Senatore, P. Zhang and H. Zheng, *The Hubble Tension in Light of the Full-Shape Analysis of Large-Scale Structure Data*, *JCAP* **05** (2021) 072 [[arXiv:2006.12420](#)] [[INSPIRE](#)].
- [66] T.L. Smith et al., *Early dark energy is not excluded by current large-scale structure data*, *Phys. Rev. D* **103** (2021) 123542 [[arXiv:2009.10740](#)] [[INSPIRE](#)].
- [67] T. Simon, P. Zhang, V. Poulin and T.L. Smith, *Updated constraints from the effective field theory analysis of the BOSS power spectrum on early dark energy*, *Phys. Rev. D* **107** (2023) 063505 [[arXiv:2208.05930](#)] [[INSPIRE](#)].
- [68] R. Gsponer et al., *Cosmological constraints on early dark energy from the full shape analysis of eBOSS DR16*, *Mon. Not. Roy. Astron. Soc.* **530** (2024) 3075 [[arXiv:2312.01977](#)] [[INSPIRE](#)].
- [69] A. Chudaykin, M.M. Ivanov, O.H.E. Philcox and M. Simonović, *Nonlinear perturbation theory extension of the Boltzmann code CLASS*, *Phys. Rev. D* **102** (2020) 063533 [[arXiv:2004.10607](#)] [[INSPIRE](#)].
- [70] S.-F. Chen, Z. Vlah and M. White, *Consistent Modeling of Velocity Statistics and Redshift-Space Distortions in One-Loop Perturbation Theory*, *JCAP* **07** (2020) 062 [[arXiv:2005.00523](#)] [[INSPIRE](#)].
- [71] A. Eggemeier et al., *COMET: Clustering observables modelled by emulated perturbation theory*, *Mon. Not. Roy. Astron. Soc.* **519** (2022) 2962 [[arXiv:2208.01070](#)] [[INSPIRE](#)].
- [72] P. Carrilho, C. Moretti and A. Pourtsidou, *Cosmology with the EFTofLSS and BOSS: dark energy constraints and a note on priors*, *JCAP* **01** (2023) 028 [[arXiv:2207.14784](#)] [[INSPIRE](#)].
- [73] H.E. Noriega, A. Aviles, S. Fromenteau and M. Vargas-Magaña, *Fast computation of non-linear power spectrum in cosmologies with massive neutrinos*, *JCAP* **11** (2022) 038 [[arXiv:2208.02791](#)] [[INSPIRE](#)].
- [74] A. Pezzotta et al., *Extending evolution mapping to massive neutrinos with COMET*, *Phys. Rev. D* **112** (2025) 023520 [[arXiv:2503.16160](#)] [[INSPIRE](#)].
- [75] D. Linde et al., *CLASS-OneLoop: accurate and unbiased inference from spectroscopic galaxy surveys*, *JCAP* **07** (2024) 068 [[arXiv:2402.09778](#)] [[INSPIRE](#)].
- [76] J. DeRose, S.-F. Chen, M. White and N. Kokron, *Neural network acceleration of large-scale structure theory calculations*, *JCAP* **04** (2022) 056 [[arXiv:2112.05889](#)] [[INSPIRE](#)].
- [77] J. Donald-McCann, K. Koyama and F. Beutler, *matryoshka II: accelerating effective field theory analyses of the galaxy power spectrum*, *Mon. Not. Roy. Astron. Soc.* **518** (2022) 3106 [[arXiv:2202.07557](#)] [[INSPIRE](#)].
- [78] S. Ramirez et al., *Full shape cosmology analysis from BOSS in configuration space using neural network acceleration*, *JCAP* **08** (2024) 049 [[arXiv:2310.17834](#)] [[INSPIRE](#)].
- [79] Y. Lai, C. Howlett and T.M. Davis, *Faster cosmological analysis with power spectrum without simulations*, *Mon. Not. Roy. Astron. Soc.* **530** (2024) 4519 [[arXiv:2306.00388](#)] [[INSPIRE](#)].
- [80] S. Trusov, P. Zarrouk and S. Cole, *Neural network-based model of galaxy power spectrum: fast full-shape galaxy power spectrum analysis*, *Mon. Not. Roy. Astron. Soc.* **538** (2025) 1789 [[arXiv:2403.20093](#)] [[INSPIRE](#)].

- [81] M. Bonici, G. D’Amico, J. Bel and C. Carbone, *Effort.jl: a fast and differentiable emulator for the Effective Field Theory of the Large Scale Structure of the Universe*, *JCAP* **09** (2025) 044 [[arXiv:2501.04639](#)] [[INSPIRE](#)].
- [82] PLANCK collaboration, *Planck 2018 results. VI. Cosmological parameters*, *Astron. Astrophys.* **641** (2020) A6 [*Erratum ibid.* **652** (2021) C4] [[arXiv:1807.06209](#)] [[INSPIRE](#)].
- [83] M. Simonović et al., *Cosmological perturbation theory using the FFTLog: formalism and connection to QFT loop integrals*, *JCAP* **04** (2018) 030 [[arXiv:1708.08130](#)] [[INSPIRE](#)].
- [84] C. Anastasiou, D.P.L. Bragança, L. Senatore and H. Zheng, *Efficiently evaluating loop integrals in the EFTofLSS using QFT integrals with massive propagators*, *JHEP* **01** (2024) 002 [[arXiv:2212.07421](#)] [[INSPIRE](#)].
- [85] T. Bakx, N.E. Chisari and Z. Vlah, *COBRA: Optimal Factorization of Cosmological Observables*, *Phys. Rev. Lett.* **134** (2025) 191002 [[arXiv:2407.04660](#)] [[INSPIRE](#)].
- [86] M. Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, [arXiv:1603.04467](#) [[INSPIRE](#)].
- [87] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, [arXiv:1412.6980](#) [[INSPIRE](#)].
- [88] A.G. Sánchez, *Arguments against using h^{-1} Mpc units in observational cosmology*, *Phys. Rev. D* **102** (2020) 123511 [[arXiv:2002.07829](#)] [[INSPIRE](#)].
- [89] BOSS collaboration, *SDSS-III Baryon Oscillation Spectroscopic Survey Data Release 12: galaxy target selection and large scale structure catalogues*, *Mon. Not. Roy. Astron. Soc.* **455** (2016) 1553 [[arXiv:1509.06529](#)] [[INSPIRE](#)].
- [90] BOSS collaboration, *The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: cosmological analysis of the DR12 galaxy sample*, *Mon. Not. Roy. Astron. Soc.* **470** (2017) 2617 [[arXiv:1607.03155](#)] [[INSPIRE](#)].
- [91] EBOSS collaboration, *The Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: Large-scale structure catalogues for cosmological analysis*, *Mon. Not. Roy. Astron. Soc.* **498** (2020) 2354 [[arXiv:2007.09000](#)] [[INSPIRE](#)].
- [92] EBOSS collaboration, *Completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: Cosmological implications from two decades of spectroscopic surveys at the Apache Point Observatory*, *Phys. Rev. D* **103** (2021) 083533 [[arXiv:2007.08991](#)] [[INSPIRE](#)].
- [93] D. Blas, J. Lesgourgues and T. Tram, *The Cosmic Linear Anisotropy Solving System (CLASS) II: Approximation schemes*, *JCAP* **07** (2011) 034 [[arXiv:1104.2933](#)] [[INSPIRE](#)].
- [94] D. Piras and A. Spurio Mancini, *CosmoPower-JAX: high-dimensional Bayesian inference with differentiable cosmological emulators*, [arXiv:2305.06347](#) [[DOI:10.21105/astro.2305.06347](#)] [[INSPIRE](#)].
- [95] C. Sui et al., *SYREN-NEW: Precise formulae for the linear and nonlinear matter power spectra with massive neutrinos and dynamical dark energy*, *Astron. Astrophys.* **698** (2025) A1 [[arXiv:2410.14623](#)] [[INSPIRE](#)].
- [96] D. Foreman-Mackey, D.W. Hogg, D. Lang and J. Goodman, *emcee: The MCMC Hammer*, *Publ. Astron. Soc. Pac.* **125** (2013) 306 [[arXiv:1202.3665](#)] [[INSPIRE](#)].
- [97] M. Karamanis, F. Beutler and J.A. Peacock, *zeus: a python implementation of ensemble slice sampling for efficient Bayesian parameter inference*, *Mon. Not. Roy. Astron. Soc.* **508** (2021) 3589 [[arXiv:2105.03468](#)] [[INSPIRE](#)].

- [98] M. Betancourt, *A Conceptual Introduction to Hamiltonian Monte Carlo*, [arXiv:1701.02434](#) [INSPIRE].
- [99] M.D. Hoffman and A. Gelman, *The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo*, [arXiv:1111.4246](#) [INSPIRE].
- [100] A. Cabezas et al., *BlackJAX: Composable Bayesian inference in JAX*, [arXiv:2402.10797](#).
- [101] J. Robnik, G.B. De Luca, E. Silverstein and U. Seljak, *Microcanonical Hamiltonian Monte Carlo*, [arXiv:2212.08549](#) [INSPIRE].
- [102] J.U. Lange, *nautilus: boosting Bayesian importance nested sampling with deep learning*, *Mon. Not. Roy. Astron. Soc.* **525** (2023) 3181 [[arXiv:2306.16923](#)] [INSPIRE].
- [103] B. Audren, J. Lesgourgues, K. Benabed and S. Prunet, *Conservative Constraints on Early Cosmology: an illustration of the Monte Python cosmological parameter inference code*, *JCAP* **02** (2013) 001 [[arXiv:1210.7183](#)] [INSPIRE].
- [104] T. Brinckmann and J. Lesgourgues, *MontePython 3: boosted MCMC sampler and other features*, *Phys. Dark Univ.* **24** (2019) 100260 [[arXiv:1804.07261](#)] [INSPIRE].
- [105] A. Lewis, A. Challinor and A. Lasenby, *Efficient computation of CMB anisotropies in closed FRW models*, *Astrophys. J.* **538** (2000) 473 [[astro-ph/9911177](#)] [INSPIRE].
- [106] C. Howlett, A. Lewis, A. Hall and A. Challinor, *CMB power spectrum parameter degeneracies in the era of precision cosmology*, *JCAP* **04** (2012) 027 [[arXiv:1201.3654](#)] [INSPIRE].
- [107] A.G. Baydin, B.A. Pearlmutter, A.A. Radul and J.M. Siskind, *Automatic differentiation in machine learning: a survey*, [arXiv:1502.05767](#) [INSPIRE].
- [108] D. Piras et al., *The future of cosmological likelihood-based inference: accelerated high-dimensional parameter estimation and model comparison*, [arXiv:2405.12965](#) [[DOI:10.33232/001c.123368](#)] [INSPIRE].
- [109] F. James and M. Roos, *Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations*, *Comput. Phys. Commun.* **10** (1975) 343 [INSPIRE].
- [110] T. Karwal et al., *Procoli: Profiles of cosmological likelihoods*, [arXiv:2401.14225](#) [INSPIRE].
- [111] E.B. Holm et al., *PROSPECT: a profile likelihood code for frequentist cosmological parameter inference*, *Mon. Not. Roy. Astron. Soc.* **535** (2024) 3686 [[arXiv:2312.02972](#)] [INSPIRE].
- [112] S. Paradiso et al., *Reducing nuisance prior sensitivity via non-linear reparameterization, with application to EFT analyses of large-scale structure*, [arXiv:2412.03503](#) [[DOI:10.1088/1475-7516/2025/07/005](#)] [INSPIRE].
- [113] A. Lewis, *GetDist: a Python package for analysing Monte Carlo samples*, *JCAP* **08** (2025) 025 [[arXiv:1910.13970](#)] [INSPIRE].
- [114] R. Angulo, M. Fasiello, L. Senatore and Z. Vlah, *On the Statistics of Biased Tracers in the Effective Field Theory of Large Scale Structures*, *JCAP* **09** (2015) 029 [[arXiv:1503.08826](#)] [INSPIRE].
- [115] T. Fujita et al., *Very Massive Tracers and Higher Derivative Biases*, *JCAP* **01** (2020) 009 [[arXiv:1609.00717](#)] [INSPIRE].
- [116] P. McDonald and A. Roy, *Clustering of dark matter tracers: generalizing bias for the coming era of precision LSS*, *JCAP* **08** (2009) 020 [[arXiv:0902.0991](#)] [INSPIRE].
- [117] V. Assassi, D. Baumann, D. Green and M. Zaldarriaga, *Renormalized Halo Bias*, *JCAP* **08** (2014) 056 [[arXiv:1402.5916](#)] [INSPIRE].

- [118] M. Mirbabayi, F. Schmidt and M. Zaldarriaga, *Biased Tracers and Time Evolution*, *JCAP* **07** (2015) 030 [[arXiv:1412.5169](#)] [[INSPIRE](#)].
- [119] A. Eggemeier, R. Scoccimarro and R.E. Smith, *Bias Loop Corrections to the Galaxy Bispectrum*, *Phys. Rev. D* **99** (2019) 123514 [[arXiv:1812.03208](#)] [[INSPIRE](#)].
- [120] J.J.M. Carrasco, M.P. Hertzberg and L. Senatore, *The Effective Field Theory of Cosmological Large Scale Structures*, *JHEP* **09** (2012) 082 [[arXiv:1206.2926](#)] [[INSPIRE](#)].
- [121] E. Pajer and M. Zaldarriaga, *On the Renormalization of the Effective Field Theory of Large Scale Structures*, *JCAP* **08** (2013) 037 [[arXiv:1301.7182](#)] [[INSPIRE](#)].
- [122] L. Senatore and M. Zaldarriaga, *Redshift Space Distortions in the Effective Field Theory of Large Scale Structures*, [arXiv:1409.1225](#) [[INSPIRE](#)].
- [123] L. Senatore, *Bias in the Effective Field Theory of Large Scale Structures*, *JCAP* **11** (2015) 007 [[arXiv:1406.7843](#)] [[INSPIRE](#)].
- [124] M. Archidiacono, S. Hannestad and J. Lesgourgues, *What will it take to measure individual neutrino mass states using cosmology?*, *JCAP* **09** (2020) 021 [[arXiv:2003.03354](#)] [[INSPIRE](#)].
- [125] D. Racco, P. Zhang and H. Zheng, *Neutrino masses from large-scale structures: Future sensitivity and theory dependence*, *Phys. Dark Univ.* **47** (2025) 101803 [[arXiv:2412.04959](#)] [[INSPIRE](#)].
- [126] M.M. Ivanov et al., *Cosmological constraints without nonlinear redshift-space distortions*, *Phys. Rev. D* **105** (2022) 043531 [[arXiv:2110.00006](#)] [[INSPIRE](#)].
- [127] M. Lewandowski et al., *EFT of large scale structures in redshift space*, *Phys. Rev. D* **97** (2018) 063526 [[arXiv:1512.06831](#)] [[INSPIRE](#)].
- [128] H.A. Feldman, N. Kaiser and J.A. Peacock, *Power spectrum analysis of three-dimensional redshift surveys*, *Astrophys. J.* **426** (1994) 23 [[astro-ph/9304022](#)] [[INSPIRE](#)].
- [129] K. Yamamoto et al., *A measurement of the quadrupole power spectrum in the clustering of the 2dF QSO Survey*, *Publ. Astron. Soc. Jap.* **58** (2006) 93 [[astro-ph/0505115](#)] [[INSPIRE](#)].
- [130] BOSS collaboration, *The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: Testing gravity with redshift-space distortions using the power spectrum multipoles*, *Mon. Not. Roy. Astron. Soc.* **443** (2014) 1065 [[arXiv:1312.4611](#)] [[INSPIRE](#)].
- [131] D. Bianchi, H. Gil-Marín, R. Ruggeri and W.J. Percival, *Measuring line-of-sight dependent Fourier-space clustering using FFTs*, *Mon. Not. Roy. Astron. Soc.* **453** (2015) L11 [[arXiv:1505.05341](#)] [[INSPIRE](#)].
- [132] R. Scoccimarro, *Fast Estimators for Redshift-Space Clustering*, *Phys. Rev. D* **92** (2015) 083532 [[arXiv:1506.02729](#)] [[INSPIRE](#)].
- [133] BOSS collaboration, *The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: RSD measurement from the LOS-dependent power spectrum of DR12 BOSS galaxies*, *Mon. Not. Roy. Astron. Soc.* **460** (2016) 4188 [[arXiv:1509.06386](#)] [[INSPIRE](#)].
- [134] F.-S. Kitaura et al., *The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: mock galaxy catalogues for the BOSS Final Data Release*, *Mon. Not. Roy. Astron. Soc.* **456** (2016) 4156 [[arXiv:1509.06400](#)] [[INSPIRE](#)].
- [135] J. Hartlap, P. Simon and P. Schneider, *Why your model parameter confidences might be too optimistic: Unbiased estimation of the inverse covariance matrix*, *Astron. Astrophys.* **464** (2007) 399 [[astro-ph/0608064](#)] [[INSPIRE](#)].

- [136] M.J. Wilson, J.A. Peacock, A.N. Taylor and S. de la Torre, *Rapid modelling of the redshift-space power spectrum multipoles for a masked density field*, *Mon. Not. Roy. Astron. Soc.* **464** (2017) 3121 [[arXiv:1511.07799](#)] [[INSPIRE](#)].
- [137] BOSS collaboration, *The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: Anisotropic galaxy clustering in Fourier-space*, *Mon. Not. Roy. Astron. Soc.* **466** (2017) 2242 [[arXiv:1607.03150](#)] [[INSPIRE](#)].
- [138] F. Beutler, E. Castorina and P. Zhang, *Interpreting measurements of the anisotropic galaxy power spectrum*, *JCAP* **03** (2019) 040 [[arXiv:1810.05051](#)] [[INSPIRE](#)].
- [139] C.H. Hahn et al., *The effect of fibre collisions on the galaxy power spectrum multipoles*, *Mon. Not. Roy. Astron. Soc.* **467** (2017) 1940 [[arXiv:1609.01714](#)] [[INSPIRE](#)].
- [140] N. Schöneberg, J. Lesgourgues and D.C. Hooper, *The BAO+BBN take on the Hubble tension*, *JCAP* **10** (2019) 029 [[arXiv:1907.11594](#)] [[INSPIRE](#)].
- [141] R. Consiglio et al., *PArthENoPE reloaded*, *Comput. Phys. Commun.* **233** (2018) 237 [[arXiv:1712.04378](#)] [[INSPIRE](#)].
- [142] R.J. Cooke, M. Pettini and C.C. Steidel, *One Percent Determination of the Primordial Deuterium Abundance*, *Astrophys. J.* **855** (2018) 102 [[arXiv:1710.11129](#)] [[INSPIRE](#)].
- [143] E. Aver, K.A. Olive and E.D. Skillman, *The effects of He I $\lambda 10830$ on helium abundance determinations*, *JCAP* **07** (2015) 011 [[arXiv:1503.08146](#)] [[INSPIRE](#)].
- [144] D. Brout et al., *The Pantheon+ Analysis: Cosmological Constraints*, *Astrophys. J.* **938** (2022) 110 [[arXiv:2202.04077](#)] [[INSPIRE](#)].
- [145] A.G. Riess et al., *A Comprehensive Measurement of the Local Value of the Hubble Constant with $1 \text{ km s}^{-1} \text{ Mpc}^{-1}$ Uncertainty from the Hubble Space Telescope and the SH0ES Team*, *Astrophys. J. Lett.* **934** (2022) L7 [[arXiv:2112.04510](#)] [[INSPIRE](#)].
- [146] V. Poulin, T.L. Smith, T. Karwal and M. Kamionkowski, *Early Dark Energy Can Resolve The Hubble Tension*, *Phys. Rev. Lett.* **122** (2019) 221301 [[arXiv:1811.04083](#)] [[INSPIRE](#)].
- [147] V. Poulin, T.L. Smith and T. Karwal, *The Ups and Downs of Early Dark Energy solutions to the Hubble tension: A review of models, hints and constraints circa 2023*, *Phys. Dark Univ.* **42** (2023) 101348 [[arXiv:2302.09032](#)] [[INSPIRE](#)].
- [148] V. Poulin, T.L. Smith, R. Calderón and T. Simon, *Implications of the cosmic calibration tension beyond H_0 and the synergy between early- and late-time new physics*, *Phys. Rev. D* **111** (2025) 083552 [[arXiv:2407.18292](#)] [[INSPIRE](#)].
- [149] M.M. Ivanov et al., *Constraining Early Dark Energy with Large-Scale Structure*, *Phys. Rev. D* **102** (2020) 103502 [[arXiv:2006.11235](#)] [[INSPIRE](#)].
- [150] A. Reeves et al., *Restoring cosmological concordance with early dark energy and massive neutrinos?*, *Mon. Not. Roy. Astron. Soc.* **520** (2023) 3688 [[arXiv:2207.01501](#)] [[INSPIRE](#)].
- [151] G. Efstathiou, E. Rosenberg and V. Poulin, *Improved Planck Constraints on Axionlike Early Dark Energy as a Resolution of the Hubble Tension*, *Phys. Rev. Lett.* **132** (2024) 221002 [[arXiv:2311.00524](#)] [[INSPIRE](#)].
- [152] V. Poulin et al., *Cosmological implications of ultralight axionlike fields*, *Phys. Rev. D* **98** (2018) 083525 [[arXiv:1806.10608](#)] [[INSPIRE](#)].
- [153] T.L. Smith, V. Poulin and M.A. Amin, *Oscillating scalar fields and the Hubble tension: a resolution with novel signatures*, *Phys. Rev. D* **101** (2020) 063523 [[arXiv:1908.06995](#)] [[INSPIRE](#)].

- [154] J.M. Berryman et al., *Neutrino self-interactions: A white paper*, *Phys. Dark Univ.* **42** (2023) 101267 [[arXiv:2203.01955](#)] [[INSPIRE](#)].
- [155] D. Camarena, F.-Y. Cyr-Racine and J. Houghteling, *Confronting self-interacting neutrinos with the full shape of the galaxy power spectrum*, *Phys. Rev. D* **108** (2023) 103535 [[arXiv:2309.03941](#)] [[INSPIRE](#)].
- [156] A. He, R. An, M.M. Ivanov and V. Gluscevic, *Self-interacting neutrinos in light of large-scale structure data*, *Phys. Rev. D* **109** (2024) 103527 [[arXiv:2309.03956](#)] [[INSPIRE](#)].
- [157] D. Camarena and F.-Y. Cyr-Racine, *Strong constraints on a simple self-interacting neutrino cosmology*, *Phys. Rev. D* **111** (2025) 023504 [[arXiv:2403.05496](#)] [[INSPIRE](#)].
- [158] DESI collaboration, *DESI DR2 results. II. Measurements of baryon acoustic oscillations and cosmological constraints*, *Phys. Rev. D* **112** (2025) 083515 [[arXiv:2503.14738](#)] [[INSPIRE](#)].
- [159] F. Verdiani, E. Castorina, E. Salvioni and E. Sefusatti, *The Effective Field Theory of Large Scale Structure for Mixed Dark Matter Scenarios*, [arXiv:2507.08792](#) [[INSPIRE](#)].
- [160] T. Mergulhão, H. Rubira, R. Voivodic and L.R. Abramo, *The effective field theory of large-scale structure and multi-tracer*, *JCAP* **04** (2022) 021 [[arXiv:2108.11363](#)] [[INSPIRE](#)].
- [161] T. Mergulhão, H. Rubira and R. Voivodic, *The effective field theory of large-scale structure and multi-tracer II: redshift space and realistic tracers*, *JCAP* **01** (2024) 008 [[arXiv:2306.05474](#)] [[INSPIRE](#)].
- [162] R.A. Porto, L. Senatore and M. Zaldarriaga, *The Lagrangian-space Effective Field Theory of Large Scale Structures*, *JCAP* **05** (2014) 022 [[arXiv:1311.2168](#)] [[INSPIRE](#)].
- [163] L. Senatore and M. Zaldarriaga, *The IR-resummed Effective Field Theory of Large Scale Structures*, *JCAP* **02** (2015) 013 [[arXiv:1404.5954](#)] [[INSPIRE](#)].
- [164] P. Zhang and Y. Cai, *BOSS full-shape analysis from the EFTofLSS with exact time dependence*, *JCAP* **01** (2022) 031 [[arXiv:2111.05739](#)] [[INSPIRE](#)].
- [165] A.J.S. Hamilton, *Uncorrelated modes of the nonlinear power spectrum*, *Mon. Not. Roy. Astron. Soc.* **312** (2000) 257 [[astro-ph/9905191](#)] [[INSPIRE](#)].