Deep learning and genetic algorithms for cosmological Bayesian inference speed-up

Isidro Gómez-Vargas[1,2,*] and J. Alberto Vázquez[1,†]

[1]*Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México, 62210, Cuernavaca, Morelos, México*

[2]*Department of Astronomy, University of Geneva, Versoix, 1290, Switzerland*

(Received 7 May 2024; accepted 12 September 2024; published 15 October 2024)

In this paper, we present a novel approach to accelerate the Bayesian inference process, focusing specifically on the nested sampling algorithms. Bayesian inference plays a crucial role in cosmological parameter estimation, providing a robust framework for extracting theoretical insights from observational data. However, its computational demands can be substantial, primarily due to the need for numerous likelihood function evaluations. Our method utilizes the power of deep learning, employing feed forward neural networks to approximate the likelihood function dynamically during the Bayesian inference process. Unlike traditional approaches, our method trains neural networks on-the-fly using the current set of live points as training data, without the need for pretraining. This flexibility enables adaptation to various theoretical models and datasets. We perform the hyperparameter optimization using genetic algorithms to suggest initial neural network architectures for learning each likelihood function. Once sufficient accuracy is achieved, the neural network replaces the original likelihood function. The implementation integrates with nested sampling algorithms and has been thoroughly evaluated using both simple cosmological dark energy models and diverse observational datasets. Additionally, we explore the potential of genetic algorithms for generating initial live points within nested sampling inference, opening up new avenues for enhancing the efficiency and effectiveness of Bayesian inference methods.

DOI: 10.1103/PhysRevD.110.083518

I. INTRODUCTION

Bayesian inference is a powerful tool in several scientific fields where it is essential to constrain mathematical models using experimental data. It allows parameter estimation and model comparison. In particular, it is the data analysis technique per excellence in observational cosmology, as it provides a robust method to obtain valuable statistical information from a theoretical model given a set of observational data. However, a significant disadvantage of Bayesian inference lies in its high computational cost; it requires a considerable number of likelihood function evaluations to generate sufficient samples from the posterior distribution. For example, a small Bayesian inference task could involve thousands of samples and require thousands, or even millions, of likelihood evaluations.

Given the crucial importance of parameter estimation in the context of astronomical surveys, within the fields of cosmology and astrophysics, numerous valuable efforts have been made to address the computational challenge of mitigating the complexity of the likelihood function calculation to speed up Bayesian inference. Some strategies provide an approximation of Bayesian inference by avoiding the computation of the full likelihood function, as suggested by [1–3]. On the other hand, some efforts try to speed up the inference with different statistical techniques [4–8]. Alternatively, other works [9–12] introduced the concept of generating synthetic likelihood distributions. Furthermore, there is an emerging trend of exploiting machine learning tools to accelerate the Bayesian inference process [10,13–16].

The use of artificial neural networks (ANNs) to approximate the likelihood function can greatly improve the efficiency of Bayesian inference [14,16–21]. However, it is necessary to have a careful consideration of the trade-off between accuracy and speed, along with quality monitoring of the resulting posterior samples. In addition, neural networks present several drawbacks that must be taken into account to effectively aid in the performance of Bayesian inference:

(1) ANNs excel at interpolation, but not at extrapolation. Like all machine learning algorithms, ANNs generate models based on datasets, allowing them to learn data structures and predict unseen data within the bounds of the training region. In the Bayesian inference domain, new samples try to find better likelihood values, which could correspond to points

*Contact author: igomez@icf.unam.mx

†Contact author: javazquez@icf.unam.mx

outside the ranges of the random sample used for the ANN training.

(2) The performance of ANNs depends on their hyperparameters. This is perhaps one of the most challenging issues facing neural networks. If the hyperparameters are not chosen carefully, the neural network models can be under- or overfitted.

(3) The selection of hyperparameters depends on the data. There is no unique architecture for an ANN. Each dataset requires certain hyperparameter configurations to have an efficient training of the neural network.

(4) Training an ANN requires computational resources. It is a well-known fact that training a neural network can be computationally demanding, which seems contradictory when the goal is to reduce the computational time in a Bayesian inference process.

We will come back to these issues in Sec. IV by presenting how each of them is addressed by the method we propose.

Previous works using neural networks in cosmological parameter estimation save an amazing amount of computational time training neural networks before the Bayesian inference process [14,16,22]; however, the pretraining time in these cases is expensive and the trained neural networks are only useful for a specific configuration of backgrounds, models, and datasets. For this reason, our work is inspired by BAMBI [18,19], and pyBambi [23], where their neural networks are trained in real-time to learn the likelihood function, which is subsequently replaced within a nested sampling process. The strength of this approach lies in its ability to train the neural network in real-time and accelerate the Bayesian inference process without restricting a cosmological or theoretical model and specific datasets. In our method, we explore features beyond those of our predecessors, such as parallelism, PyTorch implementation [24], and hyperparameter tuning. In addition, we exclusively used live points for training to reduce the dispersion of the training dataset and to obtain results with higher accuracy. A criterion was also chosen to initiate our method that serves as a regulator of the trade-off between accuracy and speed. We also implemented an on-the-fly performance evaluation to accept or reject the neural network predictions. In addition, we have conducted a preliminary investigation on the use of genetic algorithms to generate the initial sample of live points on the nested sampling process.

The structure of the paper is as follows: Sec. II offers an overview of Bayesian inference and nested sampling. Section III provides a concise exposition of the machine learning fundamentals employed in this study. The concept and development of our machine learning strategies are detailed in Sec. IV. Sections V and VI present our results, applied respectively to testing toy models and estimating cosmological parameters. In Sec. VII, we discuss our research findings and present our final reflections. Furthermore, the Appendix features preliminary results about the incorporation of genetic algorithms as initiators of the live points in a nested sampling execution.

II. STATISTICAL BACKGROUND

In this section, we describe an overview of Bayesian inference and neural networks. In particular, we focus on the nested sampling algorithm and feed forward neural networks.

A. Bayesian inference

Considering the Bayes' theorem as follows:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \tag{1}$$

where $P(\theta)$ denotes the prior distribution over parameters θ, encapsulating any prior knowledge about them before observing the data. $P(D|\theta)$ represents the likelihood function, expressing the conditional probability of observing the data given the model. Finally, the Bayesian evidence $P(D)$ serves as a normalization constant through likelihood marginalization

$$P(D) = \int_{\theta}^{N} P(D|\theta)P(\theta)d\theta, \tag{2}$$

where N is the number of dimensions of the parameter space for θ.

It can be assumed that the measurement error ϵ is independent of θ and has a probability density function (PDF) P_{ϵ}. In this case, the predicted value and the measurement error share the same distribution, therefore the likelihood function can be expressed as

$$P(D|\theta) = P_{\epsilon}(D - f(x;\theta)), \tag{3}$$

and if the error $\epsilon \sim N(0, C)$ has a normal distribution centered in zero and a covariance matrix C, then we have the following:

$$P(D|\theta) = \frac{1}{(2\pi)^{N/2}|C|^{1/2}} e^{-0.5(D-f(x;\theta))^T C^{-1}(D-f(x;\theta))}. \tag{4}$$

B. Nested sampling

Nested sampling (NS) belongs to a category of inference methods that estimate the Bayesian evidence along with its uncertainty by sampling the posterior probability density function. It was proposed by John Skilling in 2004 [25,26]. The evidence, or marginalization of the likelihood function, is a key quantity in model comparison, through the Bayes factor in the Jeffreys' scale. It is a more rigorous technique [26,27] than other widely used methods such as the

information criteria approximations [28,29]. NS works by computing the Bayesian evidence while assuming that the parameter space (prior volume or prior mass) shrinks by a certain factor. There are successful nested sampling implementations [30–32] and several applications in cosmology [33–37], astrophysics [38–40], gravitational waves analysis [41–43], biology [44,45], and in other scientific fields [46–48].

To understand the method proposed in this work, we briefly describe some considerations about the NS algorithm. For more details, we recommend Refs. [26,31,32]. First of all, the Bayesian evidence can be written as follows:

$$Z = \int \mathcal{L}(\theta)\pi(\theta)d\theta, \tag{5}$$

where θ represents the free parameters, $\pi(\theta)$ is the prior density, and $\mathcal{L}$ is the likelihood function.

The basic idea of NS is to simplify the integration of Bayesian evidence by mapping the parameter space in a unit hypercube. The fraction of the prior contained within an isolikelihood contour $\mathcal{L}_c$ in the unit hypercube is called prior volume (or prior mass)

$$X(\mathcal{L}) = \int_{\mathcal{L}(\theta)>\mathcal{L}_c} \pi(\theta)d\theta. \tag{6}$$

The Bayesian evidence can be reduced as a one-dimensional integral of the likelihood as a function of the prior volume X

$$Z = \int_0^1 \mathcal{L}(X)dX. \tag{7}$$

NS starts with a specific number n_{live} of random points, termed live points, distributed within the prior volume defined by the constrained prior. These samples are ordered based on their likelihood values. During each iteration the worst point $\mathcal{L}_{\text{worst}}$, with the lowest likelihood value, is removed. A new sample is then generated within a contour bounded by $\mathcal{L}_{\text{worst}}$ and with a likelihood, $\mathcal{L}(\theta) > \mathcal{L}_{\text{worst}}$. Equation (7) can be simplified as a Riemann sum

$$Z \approx \sum_{i=1}^{N} L_i\omega_i, \tag{8}$$

where ω_i is the difference between the prior volume of two consecutive points: $\omega_i = X_{i-1} - X_i$. Throughout the process, NS retains the population of n_{live} live points and ultimately consolidates the final set of live points within a region of high probability. Depending on the sampling approach employed from the constrained prior, various nested sampling algorithms exist. For instance, MultiNest [30] utilizes rejection sampling within ellipsoids, whereas Polychord [31] generates points using slice sampling.

Several stopping criteria exist for terminating a nested sampling run; in this study, we adopt the remaining evidence criterion, which is roughly outlined as follows:

$$\Delta Z_i \approx \mathcal{L}_{\max} X_i, \tag{9}$$

hence defining the logarithmic ratio between the current estimated evidence and the remaining evidence as

$$\Delta \ln Z_i \equiv \ln(Z_i + \Delta Z_i) - \ln Z_i, \tag{10}$$

referred to as `dlogz` hereafter in this paper. Stopping at a value `dlogz` implies sampling until only a fraction of the evidence remains unaccounted for.

III. MACHINE LEARNING BACKGROUND

Machine learning is the field of Artificial Intelligence concerning to the mathematical modeling of datasets. Its methods identify inherent properties of datasets by minimizing a target function until it reaches a satisfactory value. Over the past few years, artificial neural networks (ANNs) have emerged as the most successful type of machine learning models, giving rise to the field of deep learning. On the other hand, genetic algorithms are a special class of evolutionary algorithms, called metaheuristics, facilitating function optimization without derivatives.

This section offers a succinct overview of artificial neural networks and genetic algorithms.

A. Artificial neural networks

An artificial neural network (ANN) is a computational model inspired by biological synapses, aiming to replicate their behavior. It consists of interconnected layers of nodes, or neurons, serving as basic processing units. A fundamental type of ANN is the feed forward neural network, comprising input, hidden, and output layers. In such networks, connections between neurons, known as weights, are parameters of the model. Deep learning, a subset of machine learning, focuses exclusively on neural networks.

The intrinsic parameters of a neural network, known as hyperparameters, are set before training, and include parameters such as the number of layers and neurons, epochs, and activation functions. Parameters of gradient descent and backpropagation algorithms [49], like batch size and learning rate, may also be hyperparameters. While some hyperparameters are predetermined, others are adjusted through tuning strategies.

ANNs are valued for their capacity to model large and complex datasets. The universal approximation theorem asserts that an ANN with a single hidden layer and nonlinear activation functions can model any nonlinear function [50], enhancing its utility for datasets with complex

relationships. Even though an exhaustive review of ANNs is beyond the scope of this paper, great references exist in the literature [51,52]. For a basic introduction to their algorithms in the cosmological context, we recommend reading [53].

B. Genetic algorithms

Genetic algorithms are optimization techniques inspired by genetic population principles, treating each potential solution to an optimization problem as an individual. Initially, a genetic algorithm generates a population comprising multiple individuals within the search space. Across iterations or generations, the population evolves through operations like offspring, crossover, and mutation, progressively approaching the optimal solution of a target function. Genetic algorithms excel in addressing large-scale nonlinear and nonconvex optimization problems in challenging search scenarios [54,55].

To apply genetic algorithms to a specific problem, one must select the objective function to optimize, delineate the search space, and specify the genetic parameters such as crossover, mutation, and elitism. Probability values for crossover and mutation operators are assigned, and a selection operator determines which individuals advance to the subsequent generation. Elitism, represented by a positive integer value, dictates the number of individuals guaranteed passage to the next generation. Overall, genetic algorithms initialize a population and iteratively modify individuals through the operators and the objective function, progressively approaching the optimal solution of the target function.

While this paper does not delve deeply into the mathematical principles underlying genetic algorithms, interested readers are directed to the following Refs. [56,57], particularly for parameter estimation in cosmology [58].

IV. MACHINE LEARNING STRATEGIES

In this section, we outline our proposed method, which integrates machine learning techniques to implement neural networks and genetic algorithms within a nested sampling framework. Below we describe some deep learning techniques utilized in our training:, elucidating their application:

(i) *Data scaling*. Since all samples within the parameter space are already scaled between 0 and 1 during nested sampling, no additional scaling is required for training the neural networks.

(ii) *Early stopping*. It is a regularization technique that monitors the performance of a model on a validation set during training and stops the training process when the performance on the validation set starts to degrade, indicating overfitting. It helps to prevent overfitting and choose the best weight configuration along the epochs of the training. By stopping the training process early, the generalization performance of the model can be improved, particularly when the training data is limited or noisy. We implement early stopping with a patience of 100 epochs to guarantee a minimum number of training epochs, given the smaller size of the dataset. However, our primary focus is on preserving the best-performing weights at the end of the training process.

(iii) *Dynamic learning rate*. There are popular strategies for dynamic learning rates. However, our dynamic learning rate is only adjusted during the nested sampling run and not during the training of a specific neural network. For each new training of the neural network, the learning rate decreases by half. However, during each individual ANN training session, the learning rate remains constant within the adaptive gradient descent algorithm called `Adam` [59].

(iv) *Hyperparameter tunning*. We have implemented the option of using genetic algorithms to find the architecture of the first trained neural network. For this purpose, we use the library `nnogada` [60]. For simplicity in this work, we use genetic algorithms over 3 generations with a population size of 5 to explore combinations of batch size (4 or 8), number of layers (2 or 3), learning rate (0.0005 or 0.001), and number of neurons per layer (50 or 100). In a nested sampling execution, where we can train the neural networks multiple times, we use these small configurations. This approach yields better results compared to not tuning hyperparameters and is more effective than using a hyperparameter grid [60].

We implemented our method inside of the code SimpleMC [61,62],[1] which uses the library `dynesty` [63] for nested sampling algorithms. In all our neural network training, we use the mean squared error (MSE) as the loss function. If early stopping, with a patience of 100 epochs, does not stop the training, we select the configuration of weights that achieved the lowest MSE value.

A. `neuralike` method

Neural networks are widely acclaimed for their formidable capabilities in handling extensive datasets. However, several studies have shown their effectiveness in modeling small datasets as well; even demonstrating that neural models can accommodate a total number of weights exceeding the number of sample data points [64]. In addition, recent research has focused on novel approaches by using neural networks with smaller datasets [65–67]. While it is true that models with a large number of parameters can be prone to overfitting, this risk can be

[1]The modified version of SimpleMC that includes our `neuralike` method is available at https://github.com/igomezv/simplemc_tests.

mitigated through the use of regularization techniques such as dropout and early stopping. In our approach, these techniques, combined with genetic algorithms for optimizing the network's architecture and hyperparameters, ensure that our models generalize well even when the number of parameters exceeds the number of data points.

In nested sampling, as discussed in the previous section, there is a set of live points that maintain a constant number of elements. At a certain point in its execution, a new sample is extracted within a prior isolikelihood or mass surface. Our goal is for the neural network to predict the likelihood of points within this prior volume. To do this, we train the neural network with only the current set of live points. These points, which typically are around hundreds or thousands, are sufficient to effectively train a neural network and have several advantages:

(i) The relatively small dataset size implies that the neural network training process is not computationally intensive.

(ii) By excluding points outside the current prior volume, we can potentially avoid inaccurate predictions in regions where points would be rejected based on the original likelihood. The points that the neural network learns efficiently are those within the prior volume, because they have a higher probability of acceptance according to the original likelihood.

(iii) The quantity of elements within the training set remains constant. Whether the neural network starts its training at the beginning of sampling or at a later stage, the element count does not vary. As a result, the majority of neural network hyperparameters could stay consistent across different datasets.

The likelihood function in cosmological parameter estimation can be quite complex, often involving various types of observational data and intricate numerical operations, such as integrals, derivatives, or approximation methods for solving differential equations. To address this complexity, the idea is to replace the analytical likelihood function with a trained neural network. This substitution reduces the problem to a simple matrix multiplication, where the optimal weights, obtained during ANN training, are stored in a binary file. Consequently, the evaluation of the likelihood becomes significantly faster. This acceleration is particularly advantageous in a Bayesian inference process, where the likelihood function may need to be evaluated thousands or even millions of times, making the reduction in computational time highly beneficial.

Algorithm 1 provides an overview of our proposed methodology within a nested sampling execution. Concerning the neural network implementation, our primary focus is on the segment within the for loop. Once a predetermined number of samples have been reached, or

ALGORITHM 1. Nested sampling with `neuralike`. `dlogz_start` and `nsamples_start` are the two ways to start `neuralike`, with a `dlogz` value (recommended) or given a specific number of generated samples. The `logl_tolerance` parameter represents the neural network prediction tolerance required to be considered valid. `saved_logl` denotes the log-likelihoods of the current live points, and `valid_loss` determines the criterion for accepting or rejecting a neural network training. Any loss function values higher than `valid_loss` will be rejected. The variable *logL* represents the analytical log-likelihood function, while $\mathcal{L}$ can either be *logL* or *ANNmodel*, depending on the successful neural model.

```
using_neuralike = False
if livegenetic == True (optional) then
   Define Pmut and Pcross
   Generate a population P with Nind individuals
   Evolve population through Ngen generations
else
   Generate Nlive live points
for i in range (iteration) do
   if(dlogz < dlogz_start) OR (nsamples >= nsamples_start) then
      if i % N == 0 AND using_neuralike == False then
         Use nlive points as training dataset
         Optional: Use genetic algorithms with nnogada to choose the best architecture
         Use the best architecture to model the likelihood
         if loss function < valid_loss then
            using_neuralike = True
            L = ANNmodel
         else
            continue with NS
      if min(saved_logl) - logl_tolerance < neuralike < max(saved_logl) + logl_tolerance then
         continue else
            like=logL;
            using_neuralike = False
end
```

when the flag `dlogz_start` is activated, the ANN leverages the current live points for its training. The benefit of utilizing only the set of live points is twofold: firstly, it facilitates swift training, and secondly, it ensures that the ANN learns likelihood values strictly within the prior volume. This area is precisely where new samples should be located.

It is important to note that the nested sampling process, including the selection of priors, typically uniform or Gaussian distributions, remains consistent with standard practices. Once the criteria for initiating ANN training are met, the live points are used to train the ANN. If the ANN's performance metrics meet the required threshold, the analytical likelihood function is replaced by the ANN to save computational time. While this substitution does not alter the fundamental nested sampling process, it can significantly enhance efficiency by reducing computational overhead.

B. Using genetic algorithms

We proposed genetic algorithms, like in our `nnogada` library [60], as an optional method to find the hyperparameter of the neural network as part of the workflow of *neuralike*, as it can be noticed in the Algorithm 1. In large parameter estimation processes, it is useful, despite the time required, to find the best neural network architecture.

On the other hand, we explored the first insight about the generation of the initial live points of a nested sampling process with genetic algorithms. It is analyzed in the Appendix. Although we have incorporated the use of genetic algorithms in our code, the primary focus of this paper is on our `neuralike` method (Sec. IV A). As such, further analysis of genetic algorithms in this context will be the subject of future research.

V. TOY MODELS

As a first step in testing our method, we use some toy models as log-likelihood functions. These toy models only generate samplers within the Bayesian inference, without parameter estimation. However, it is useful to check the ability of the neural networks to learn, given a set of live points, the shape of these functions in runtime, and their respective values for the Bayesian evidence. We use the following toy models, with the mentioned hyperparameters

(i) A gaussian, $f(x,y) = -\frac{1}{2}(x^2 + \frac{y^2}{2} - xy)$. Learning rate 5×10^{-3}, 100 epochs, batch size as 1.
(ii) Eggbox function, $f(x,y) = (2 + \cos(\frac{x}{2.0})\cos(\frac{y}{2.0}))^{5.0}$. Learning rate 1×10^{-4}, 100 epochs, batch size as 1.
(iii) Himmelblau's function, $f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$. Learning rate 1×10^{-4}, 100 epochs, batch size as 1.

We have used some toy models as log-likelihood functions: Gaussian, egg-box, and Himmelblau. In Table I, you can see the results of the Bayesian evidence calculation with and without our method for the three toy models, while in Fig. 1, you can see the samples of the three functions, which at first glance are very similar. Based on these results, we can notice that for all these models, the speed of sampling using neural networks is slower than in the case of nested sampling alone; this is because the analytical functions are being evaluated directly without sampling from an unknown posterior distribution; nevertheless, these examples are very useful to verify the accuracy in calculating Bayesian evidence and sampling from the distribution. We can observe that both the log-Bayesian evidence and the graphs of the nested sampling process without and with neural networks are consistent; however, as Table I shows, for more complex functions, we need a lower value of `dlogz_start`, which means that we need to start learning the neural network at a later stage of nested sampling. Therefore, a lower `dlogz_start` parameter is needed to be more accurate but slower, and it is precisely this parameter that regulates the speed-accuracy trade-off.

VI. COSMOLOGICAL PARAMETER ESTIMATION

Assuming the geometric unit system where $\hbar = c = 8\pi G = 1$, the Friedmann equation that describes the late-time dynamical evolution for a flat-ΛCDM model can be written as

$$H(z)^2 = H_0^2[\Omega_{m,0}(1+z)^3 + (1-\Omega_{m,0})], \qquad (11)$$

TABLE I. Comparing Bayesian evidence for toy models with nested sampling alone and using `neuralike`. The column `dlogz_start` indicates the dlogz value marking the start of neural network training; higher values suggest earlier integration of neural networks into Bayesian sampling. *Valid loss* represents the threshold value of the loss function required for accepting a neural network as valid. The last two columns display the total number of samples generated through the nested sampling process and the subset produced by the trained neural networks.

Model	$\log Z$	$\log Z$ `neuralike`	`dlogz_start`	Valid loss	Samples	ANN samples
Gaussian	-2.13 ± 0.05	-2.16 ± 0.05	50	0.05	6774	6773
Eggbox	-235.83 ± 0.11	-235.82 ± 0.11	10	0.05	11794	3688
Himmelblau	-5.59 ± 0.09	-5.64 ± 0.09	5	0.05	10253	4528

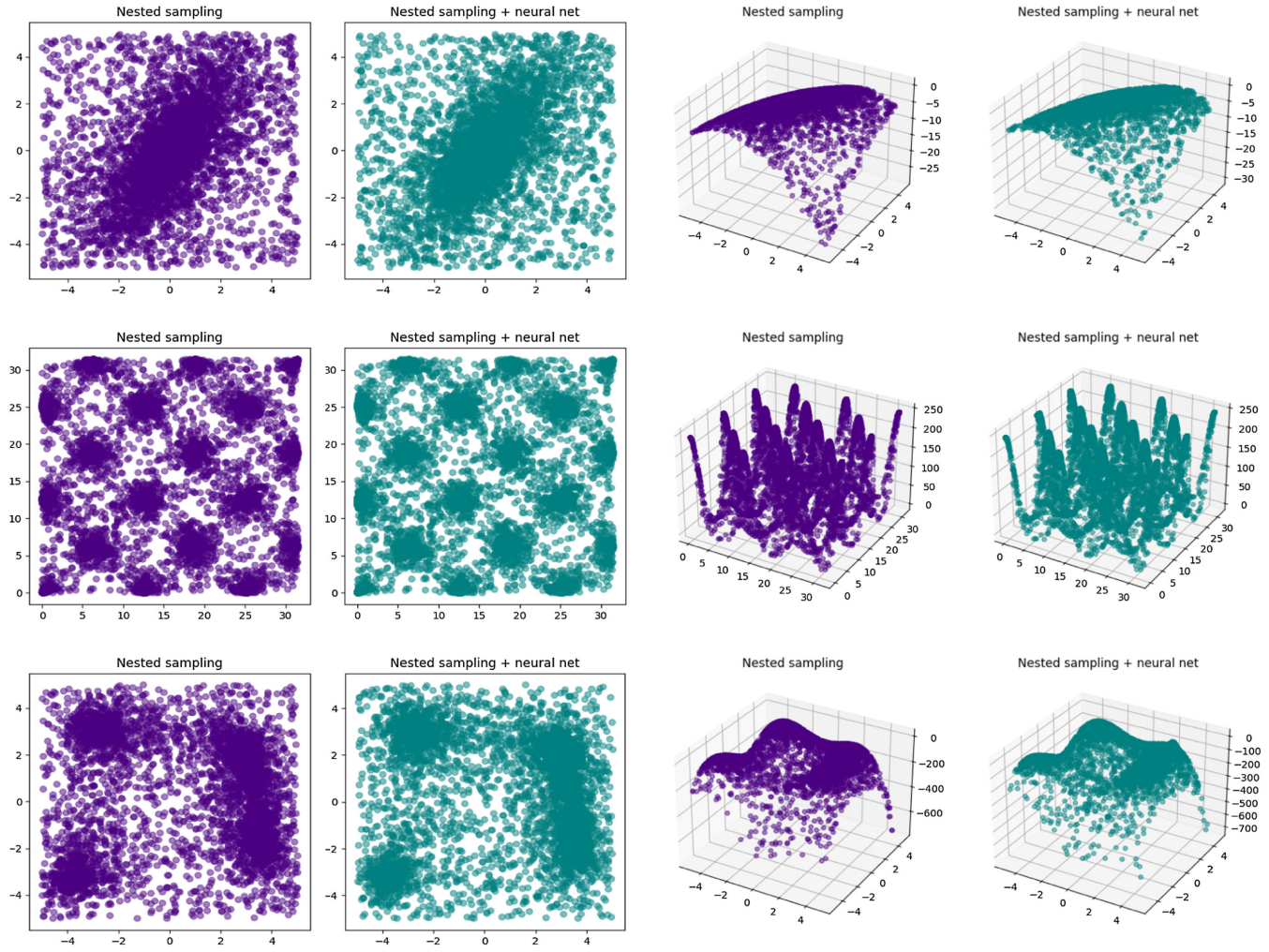

FIG. 1. Comparison of neural likelihoods versus original likelihoods using toy models. Using 1000 live points.

where H is the Hubble parameter and Ω_m is the matter density parameter; subscript 0 attached to any quantity denotes its present-day ($z = 0$) value. In this case, the EoS for the dark energy is $w(z) = -1$.

A step further to the standard model is to consider the dark energy being dynamic, where the evolution of its EoS is usually parameterized. A commonly used form of $w(z)$ is to take into account the next contribution of a Taylor expansion in terms of the scale factor $w(a) = w_0 + (1-a)w_a$ or in terms of redshift $w(z) = w_0 + \frac{z}{1+z}w_a$ (CPL model [68,69]). The parameters w_0 and w_a are real numbers such that at the present epoch $w|_{z=0} = w_0$ and $dw/dz|_{z=0} = -w_a$; we recover ΛCDM when $w_0 = -1$ and $w_a = 0$. Hence the Friedmann equation for the CPL parametrization turns out to be

$$H(z)^2 = H_0^2\Big[\Omega_{m,0}(1+z)^3 + (1-\Omega_{m,0})(1+z)^{3(1+w_0+w_a)}e^{-\frac{3w_a z}{1+z}}\Big]. \quad (12)$$

TABLE II. Exploring Bayesian Inference with Nested Sampling and `neuralike`. The definitions of the columns are consistent with those in Table I. Additionally, the *% saved time* quantifies the speed-up achieved using our method.

	`dlogz_start`	log Z	log Z `neuralike`	Samples	ANN samples	% saved time
Case 1	10	-536.39 ± 0.13	-536.62 ± 0.13	16001	9570	18.8
Case 1	5	-536.39 ± 0.13	-536.8 ± 0.13	16297	7873	5.7
Case 2	20	-536.38 ± 0.07	-536.39 ± 0.07	65216	46147	28.4
Case 3	5	-550.59 ± 0.09	-550.82 ± 0.09	95148	31923	19.6

TABLE III. Wasserstein distances [86] between nested sampling posterior samples without and with `neuralike`, for each free parameter. The closer the value of this distance is to zero, the more similar are the distributions compared. This distance is implemented in `scipy` and takes into account the 1D posterior samples and their respective weights. Overall, parameters Ω_m, $\Omega_b h^2$, and h exhibit relatively small distances across all cases. However, in Case 1a, higher values of w_0 and w_a distances are observed due to a higher `dlogz_start` value and fewer data points used. On the other hand, Case 3 demonstrates smaller (better) distances, attributed to the utilization of more data and a lower `dlogz_start` value.

	Ω_m	$\Omega_b h^2$	h	w_0	w_a	Ω_k	σ_8	Σm_ν
Case 1a (`dlogz_start=10`)	0.00480	0.00004	0.00428	0.01241	0.11283	⋯	⋯	⋯
Case 1b (`dlogz_start=5`)	0.00091	0.00004	0.00518	0.00997	0.05761	⋯	⋯	⋯
Case 2 (`dlogz_start=20`)	0.00095	0.00002	0.00111	0.00810	0.06279	⋯	⋯	⋯
Case 3 (`dlogz_start=5`)	0.00055	0.00001	0.00054	0.00335	0.01396	0.00018	0.00971	0.01753

In this work, we use cosmological datasets from type-Ia Supernovae (SN), cosmic chronometers, growth rate measurements, baryon acoustic oscillations (BAO), and a point with Planck information. Following, we briefly describe them:

(i) *Type-Ia supernovae*. We use the Pantheon SNeIa compilation, a dataset of 1048 type Ia supernovae, with a covariance matrix of systematic errors $C_{\rm sys} \in \mathbb{R}^{1048\times 1048}$ [70].

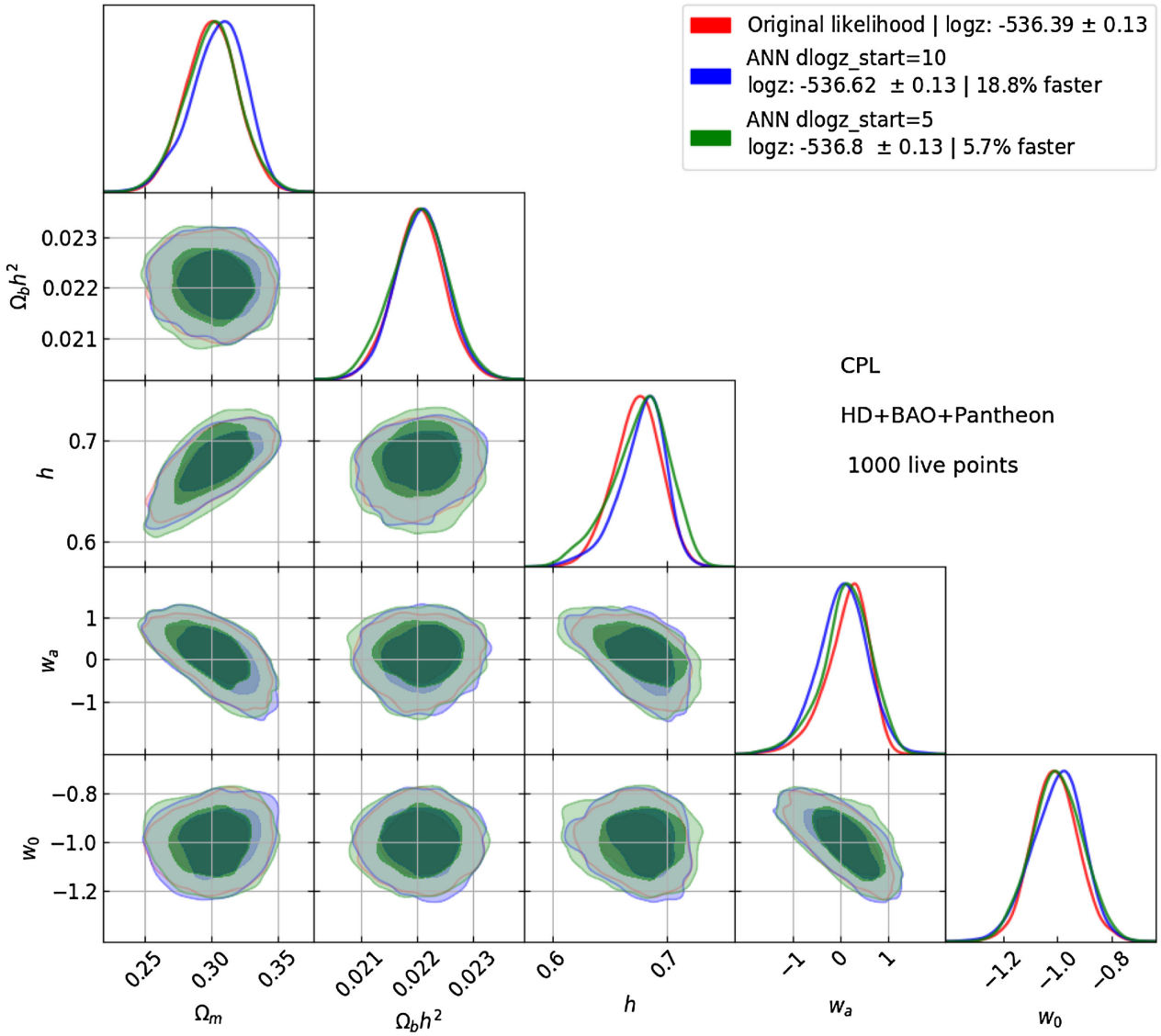

FIG. 2. Case 1. Posterior plots for CPL Pantheon + HD + BAO with the proposed methods in this work.

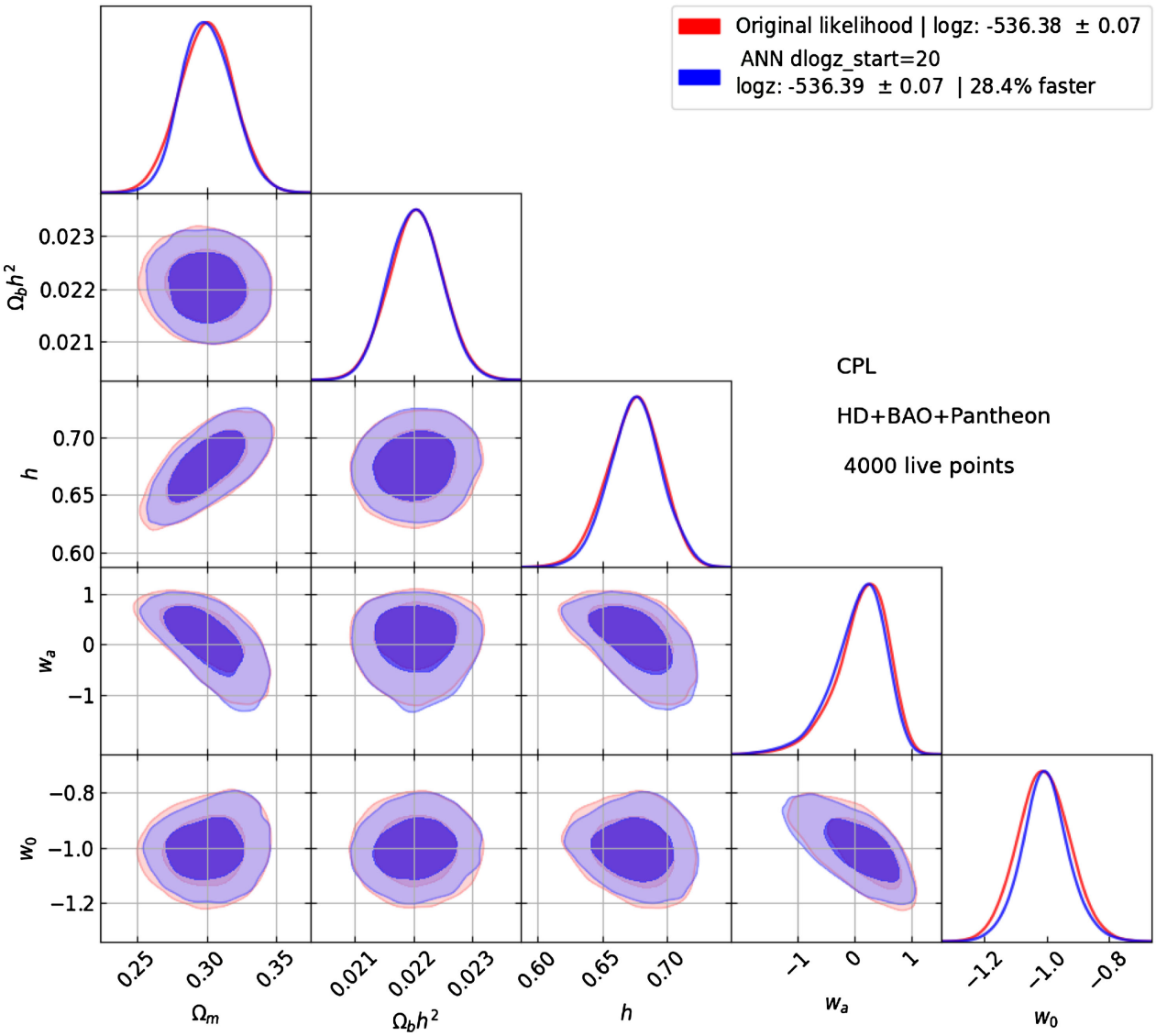

FIG. 3. Case 2. Posterior plots for CPL using Pantheon + HD + BAO with the proposed methods in this work. We use 4000 live points.

(ii) *Cosmic chronometers*. Cosmic chronometers, also known as Hubble distance (HD) measurements, are galaxies that evolve slowly and allow direct measurements of the Hubble parameter $H(z)$. We use a compilation with 31 data points collected over several years within redshifts between 0.09 and 1.965. [71–78].

(iii) *BAO*. We employ data from baryon acoustic oscillation measurements (BAO) with redshifts $z < 2.36$. They are from SDSS Main Galaxy Sample (MGS) [79], Six-Degree Field Galaxy Survey (6dFGS) [80], SDSS DR12 Galaxy Consensus [81], BOSS DR14 quasars (eBOSS) [82], Ly-α DR14 cross-correlation [83] and Ly-α DR14 autocorrelation [84].

(iv) *Growth rate measurements*. We used an extended version of the Gold-2017 compilation available in [85], which includes 22 independent measurements of $f\sigma_8(z)$ with their statistical errors obtained from redshift space distortion measurements across various surveys.

(v) *Planck-15 information*. We also consider a compressed version of Planck-15 information, where the cosmic microwave background (CMB) is treated as a BAO experiment located at redshift $z = 1090$, measuring the angular scale of the sound horizon. For more details, see the Ref. [62].

We executed three cases of parameter estimation to verify the performance of our method. We start with one thousand live points and a model with five free parameters; then, we increase the live points and free parameters, to test our method with higher dimensionality and with higher computational power demand (larger number

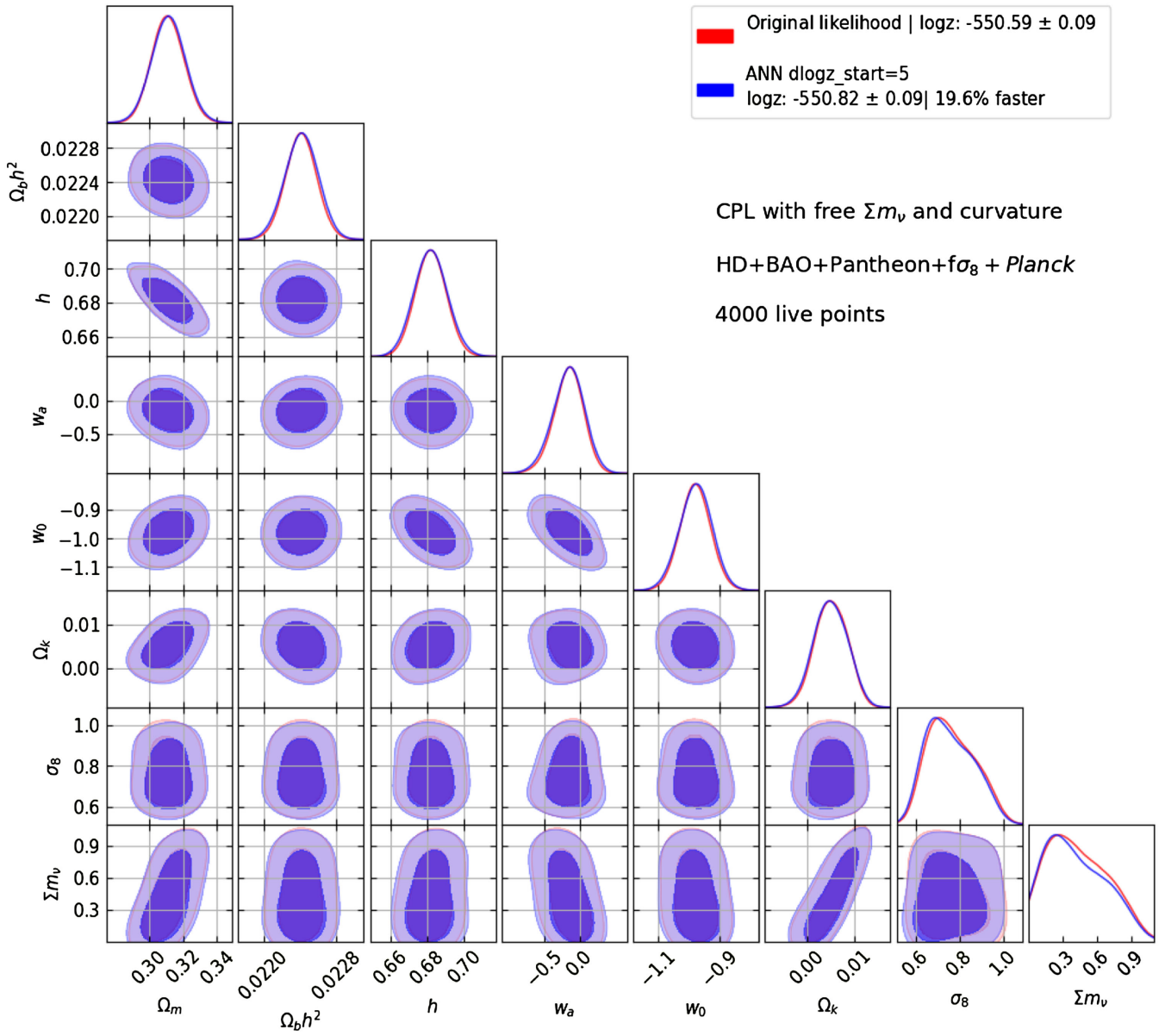

FIG. 4. Case 3. 2D posterior plots for CPL with curvature using Pantheon + HD + BAO + $f\sigma_8$ + Planck with the proposed methods in this work. Using 4000 points and considering 8 free parameters. In this case, because of the complexity, there were three neural networks trained before to substitute the likelihood function, however, the Bayesian inference process using our method was 19.6% faster.

of live points). The results are compared with a nested sampling run with the same datasets and the same configuration (live points, stopping criterion, etc.) but without ANN; this comparison aims to test the accuracy and speedup achieved by our `neuralike` method. For this comparison, we report the parameter estimation and Bayesian evidence obtained with and without our method and, in addition, we calculate the Wasserstein distances [86] between the samples of the posterior nested sampling without and with `neuralike` for each free parameter considering their respective sampling weights.

In the results, a baseline neural network architecture was employed, configured with the following hyperparameters: 3 hidden layers, a batch size of 32, a learning rate of 0.001 (utilizing the Adam gradient descent algorithm for optimization), 500 epochs, and an early stopping patience of 200 epochs. In scenarios where multiple neural networks were required, the learning rate was reduced following the previously mentioned approach. As for evaluating the accuracy of the neural networks, we adopted a `valid_loss` threshold of 0.05 for their training, and a `logl_tolerance` of 0.05 for their predictions.

A. Case 1

First, we perform the Bayesian inference for the CPL model using SNeIa from Pantheon compilation, with cosmic chronometers and BAO data. In this case, we consider only five free parameters: Ω_m, $\Omega_b h^2$, h, w_a, and w_0. We use 1000 live points. Figure 3 shows our results and we can notice that when `dlogz_start=10` the saved time is around 19% and when `dlogz_start=5` it is around only 6%. According to Table II both cases are in agreement with the $logZ$ value for nested sampling alone. If we check Table III we can notice that, in general, the samples of the `dlogz_start=5` are more similar to the nested sampling posterior distributions; it also can be appreciated in the posterior plots shown in the Fig. 2. Although the case of `dlogz_start=5` saves less time than the case of `dlogz_start=10`, it gains in accuracy.

B. Case 2

Secondly, we consider the same model, free parameters, and datasets as in Case 1. The difference in this second case is to analyze the behavior of our method with a larger number of live points. It has three new considerations: (a) the training set for the neural network would be better because has a larger size, (b) the number of operations in parallel for nested sampling is also larger, and (c) we test whether the hypotheses based on a larger number of live points can obtain a better accuracy for the neural network earlier within the nested sampling process (i.e. in a higher value for `dlogz_start`). Therefore, we increase the number of live points to 4000 and `dlogz_start` $= 20$; the outputs are included in Fig. 3 showing an excellent concordance for the Bayesian evidence values with our method, and speed-up around the 28.4%. Table II contains the results of the Bayesian evidence, and it can be noticed that the uncertainty of this case is in better agreement with nested sampling than the two scenarios of Case 1. In addition, we can analyze Table III and conclude that, effectively, its performance has a similar quality to Case 1 with `dlogz_start` $= 5$; however because it uses a higher `dlogz_start` value, the percentage of saved time is notorious.

C. Case 3

Lastly, we included more data: $f\sigma_8$ measurements and a point with Planck-15 information. To have more free parameters, eight in total, we consider contributions of the neutrino masses Σm_ν, growth rate σ_8, and curvature Ω_k. In this case, we also used 4000 live points. With these new considerations, we aim to test our method in higher dimensions and to involve a more complex likelihood function that demands more computational power with each evaluation. We made several tests, but we include the corresponding to `dlogz_start`$= 5$, in which we obtain excellent results as can be noticed in Table II. Due to the complexity of the likelihood, the full nested sampling process had to train three different neural networks, which allowed the use of erroneous predictions during sampling to be avoided.

We needed a lower value for the `dlogz_start` parameter due to the complexity of the model (given by the new free parameters); however, the saved time around of 19% concerning the nested sampling alone is remarkable and the Wasserstein distance shown in Table III indicates that the posterior distributions between the nested sampling with and without our method are similar, it also can be noticed in the posterior plots of the Fig. 4.

VII. CONCLUSIONS

In this paper, we have introduced a novel method that incorporates a neural network trained on-the-fly to learn the likelihood function within a nested sampling process. The main objective is to avoid the time-consuming analytical likelihood function, thus increasing computational efficiency. We present the `dlogz_start` parameter as a tool to handle the trade-off of accuracy and computational speed. In addition, we incorporate several deep learning techniques to minimize the risk of inaccurate neural network predictions.

To verify the effectiveness of our method, we employed several toy models, demonstrating their ability to replicate a probability distribution with remarkable accuracy in the nested sampling framework. Furthermore, in the cosmological parameter estimation, by performing a comparative analysis using the CPL cosmological model and various datasets, we highlighted the potential of our method to significantly improve the speed of nested sampling processes, without compromising the statistical reliability of the results. We found that, as the number of dimensions increased, our method produced a larger time reduction with a lower `dlogz_start` value.

Despite commencing neural network training relatively late in the nested sampling process, the overall time reduction was notable, as evidenced in Table II, showcasing reductions ranging from 6% to 19%. Potential errors in the neural network predictions were not found to be substantial because the training dataset comprised the live points. As such, the likelihood predictions are not expected to deviate significantly from the actual prior volume, which enhances the credibility and robustness of our method and instills confidence in its application in nested sampling. In addition, our constant monitoring of the ANN prediction accuracy with the actual likelihood value allows us to be more confident in the results obtained, because if the criteria were not met, the analytical function would be used again and, after certain samples, another neural network would be retrained.

We also explore the potential utility of genetic algorithms in finding optimal neural network hyperparameters and in generating initial live points for nested sampling. Concerning the former, in scenarios where models are complex or high-dimensional, searching for

an optimal architecture can be beneficial; however, our `neuralike` method allows this hyperparameter calibration to be optional so that hyperparameters can also be set by hand. Regarding the latter, we provide some insight into the potential advantages of using genetic algorithms to generate live points in the Appendix; however, future studies will address further research on this topic.

In this work, we only used observations from the late universe, as our `neuralike` method is integrated with the SimpleMC code that employs mainly background cosmology. However, our method is easily applicable to the use of other types of observations, such as CMB data, an aspect we are currently working on.

We emphasize the importance of high accuracy in neural network predictions in observational cosmology since accurate parameter estimation is crucial for a robust physical interpretation of the results. In light of the machine learning strategies proposed in this paper, we can have greater confidence in the use of neural networks to accelerate nested sampling processes, without compromising the statistical quality of the results.

The implemented algorithm presented in this work is available in Ref. [87] and the original SimpleMC code in Ref. [61], which contains the datasets used in this paper.

ACKNOWLEDGMENTS

I. G. V. thanks the CONACYT postdoctoral grant, the ICF-UNAM support, and Will Handley for his invaluable advisory about nested sampling. J. A. V. acknowledges the support provided by FOSEC SEP-CONACYT Investigación Básica A1-S-21925, FORDECYT-PRONACES-CONACYT 304001, and UNAM-DGAPA-PAPIIT IN117723. This worked was performed thanks to the help of the computational unit of the ICF-UNAM and the clusters Chalcatzingo and Teopanzolco.

APPENDIX: GENETIC ALGORITHMS AS INITIAL LIVE POINTS

Previously, we mentioned that neural networks are good at interpolating, but not at extrapolating. Within the Bayesian inference process, we sample an indeterminate posterior probability distribution, whose shape remains unknown. Despite having some idea of the range of new samples in parameter space, we cannot definitively state that the highest likelihood point is already among the live points; this uncertainty may lead to inaccurate predictions for points close to the maximum likelihood point. In Ref. [88], the authors propose the use of an optimizer to identify the optimal posterior probability sample, albeit at the expense of probabilism. The application of genetic algorithms to generate initial live points could be beneficial in cases where the Bayesian inference process must stop. In such circumstances, the partially generated posterior sampling aided by genetic algorithms will be more aligned with the maximum than a sampling generated without them. This alignment could facilitate a partial posterior sampling analysis. Although further investigation of this foray into genetic algorithms is needed, we have observed that when a small number of live points are used, and the initial live points are produced by a genetic algorithm, the stopping criterion is reached more quickly.

As the first insight into the genetic algorithms to generate the initial live points, we show some results about potential advantages in which genetic algorithms could help a nested sampling execution. In Table IV, we can see some examples in which the use of GA to generate the first live points can reduce computational time without sacrificing the statistical results. However, it is worth noticing that we are using a low number of live points because this is the case in which we observed this advantage, when a higher number of live points is used, in general, NS alone is faster because have points in a sparse region of the search space and the use of GA cluster the points around the optimums losing exploration capacity. Nonetheless, there are possible scenarios in which there could be a low number of live points and in these cases, the incursion of GA to generate the initial sampling points could apport an advantage. This is part of a further study of the exploration in detail of this combination between GA with NS.

TABLE IV. Nested sampling for the eggbox toy model and ΛCDM using 100 live points. In the NS + GA cases, we generate the first live points through genetic algorithms with a probability of mutation equal to 0.5 and a probability of crossover of 0.8.

Model	Eggbox	Eggbox	ΛCDM	ΛCDM
Sampler	NS	NS + GA	NS	NS + GA
$\log Z$	-236.16 ± 0.34	-235.07 ± 0.37	-532.87 ± 0.34	-532.70 ± 0.33
Ω_m	⋯	⋯	0.31 ± 0.011	0.31 ± 0.011
$\Omega_b h^2$	⋯	⋯	0.02 ± 0.0005	0.02 ± 0.0005
h	⋯	⋯	0.683 ± 0.009	0.683 ± 0.009
% saved time	⋯	38	⋯	23

[1] Joël Akeret, Alexandre Refregier, Adam Amara, Sebastian Seehars, and Caspar Hasner, Approximate Bayesian computation for forward modeling in cosmology, J. Cosmol. Astropart. Phys. 08 (2015) 043.

[2] Elise Jennings and Maeve Madigan, astroABC: An approximate Bayesian computation sequential Monte Carlo sampler for cosmological parameter estimation, Astron. Comput. **19**, 16 (2017).

[3] E. E. O. Ishida, S. D. P. Vitenti, M. Penna-Lima, J. Cisewski, R. S. de Souza, A. M. M. Trindade, E. Cameron, and V. C. Busti, COSMOABC: Likelihood-free inference via population Monte Carlo approximate Bayesian computation, Astron. Comput. **13**, 1 (2015).

[4] Aleksandr Petrosyan and Will Handley, Supernest: Accelerated nested sampling applied to astrophysics and cosmology, Phys. Sci. Forum **5**, 51 (2023).

[5] Joanna Dunkley, Martin Bucher, Martin Bucher, Martin Bucher, Pedro G. Ferreira, Pedro G. Ferreira, Kavilan Moodley, Kavilan Moodley, Kavilan Moodley, and Constantinos Skordis, Fast and reliable Markov chain Monte Carlo technique for cosmological parameter estimation, Mon. Not. R. Astron. Soc. **356**, 925 (2005).

[6] Thejs Brinckmann and Julien Lesgourgues, MontePython 3: Boosted MCMC sampler and other features, Phys. Dark Universe **24**, 100260 (2019).

[7] Robert L. Schuhmann, Benjamin Joachimi, and Hiranya V. Peiris, Gaussianization for fast and accurate inference from cosmological data, Mon. Not. R. Astron. Soc. **459**, 1916 (2016).

[8] Antony Lewis, Efficient sampling of fast and slow cosmological parameters, Phys. Rev. D **87**, 103529 (2013).

[9] Masanori Sato, Kiyotomo Ichiki, and Tsutomu T. Takeuchi, Copula cosmology: Constructing a likelihood function, Phys. Rev. D **83**, 023501 (2011).

[10] William A. Fendt and Benjamin D. Wandelt, Pico: Parameters for the impatient cosmologist, Astrophys. J. **654**, 2 (2007).

[11] Marcos Pellejero-Ibanez, Raul E. Angulo, Giovanni Aricó, Matteo Zennaro, Sergio Contreras, and Jens Stücker, Cosmological parameter estimation via iterative emulation of likelihoods, Mon. Not. R. Astron. Soc. **499**, 5257 (2020).

[12] Justin Alsing, Tom Charnock, Stephen Feeney, and Benjamin Wandelt, Fast likelihood-free cosmology with neural density estimators and active learning, Mon. Not. R. Astron. Soc. **488**, 4440 (2019).

[13] Adam Moss, Accelerated Bayesian inference using deep learning, Mon. Not. R. Astron. Soc. **496**, 328 (2020).

[14] Hector J. Hortua, Riccardo Volpi, Dimitri Marinelli, and Luigi Malago, Accelerating MCMC algorithms through Bayesian deep networks, arXiv:2011.14276.

[15] Isidro Gómez-Vargas, Ricardo Medel Esquivel, Ricardo García-Salcedo, and J. Alberto Vázquez, Neural network within a Bayesian inference framework, J. Phys. Conf. Ser. **1723**, 012022 (2021).

[16] Alessio Spurio Mancini, Davide Piras, Justin Alsing, Benjamin Joachimi, and Michael P Hobson, Cosmopower: Emulating cosmological power spectra for accelerated Bayesian inference from next-generation surveys, Mon. Not. R. Astron. Soc. **511**, 1771 (2022).

[17] T. Auld, Michael Bridges, M. P. Hobson, and S. F. Gull, Fast cosmological parameter estimation using neural networks, Mon. Not. R. Astron. Soc. **376**, L11 (2007).

[18] Philip Graff, Farhan Feroz, Michael P. Hobson, and Anthony Lasenby, BAMBI: Blind accelerated multimodal Bayesian inference, Mon. Not. R. Astron. Soc. **421**, 169 (2012).

[19] Philip Graff, Farhan Feroz, Michael P. Hobson, and Anthony Lasenby, Skynet: An efficient and robust neural network training tool for machine learning in astronomy, Mon. Not. R. Astron. Soc. **441**, 1741 (2014).

[20] Héctor J. Hortúa, Riccardo Volpi, Dimitri Marinelli, and Luigi Malagò, Parameter estimation for the cosmic microwave background with Bayesian neural networks, Phys. Rev. D **102**, 103509 (2020).

[21] Andreas Nygaard, Emil Brinch Holm, Steen Hannestad, and Thomas Tram, Connect: A neural network based framework for emulating cosmological observables and cosmological parameter inference, J. Cosmol. Astropart. Phys. 05 (2023) 025.

[22] Augusto T. Chantada, Susana J. Landau, Pavlos Protopapas, Claudia G. Scóccola, and Cecilia Garraffo, Nn bundle method applied to cosmology: An improvement in computational times, Phys. Rev. D **109**, 123514 (2024).

[23] Will Handley, pyBambi, https://pybambi.readthedocs.io/en/latest/#, 2018 (Online: Accessed 9-January-2024).

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga *et al.*, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).

[25] John Skilling, Nested sampling, AIP Conf. Proc. **735**, 395 (2004).

[26] John Skilling *et al.*, Nested sampling for general Bayesian computation, Bayesian Anal. **1**, 833 (2006).

[27] Adrian E. Raftery, Approximate Bayes factors and accounting for model uncertainty in generalised linear models, Biometrika **83**, 251 (1996).

[28] Andrew R. Liddle, Information criteria for astrophysical model selection, Mon. Not. R. Astron. Soc. **377**, L74 (2007).

[29] Andrew R. Liddle, Pia Mukherjee, and David Parkinson, Model selection in cosmology, Astron. Geophys. **47**, 4.30 (2006).

[30] F. Feroz, M. P. Hobson, and M. Bridges, MultiNest: An efficient and robust Bayesian inference tool for cosmology and particle physics, Mon. Not. R. Astron. Soc. **398**, 1601 (2009).

[31] W. J. Handley, M. P. Hobson, and A. N. Lasenby, Polychord: Nested sampling for cosmology, Mon. Not. R. Astron. Soc. **450**, L61 (2015).

[32] Josh Speagle and Kyle Barbary, `dynesty`: Dynamic nested sampling package, Astrophysics Source Code Library, 2018.

[33] Roberto Trotta, Farhan Feroz, Mike Hobson, and Roberto Ruiz de Austri, Recent advances in Bayesian inference in cosmology and astroparticle physics thanks to the MultiNest

algorithm, in *Astrostatistical Challenges for the New Astronomy* (Springer, New York, 2013), pp. 107–119.
[34] David Parkinson, Pia Mukherjee, and Andrew Liddle, Cosmonest: Cosmological nested sampling, ascl, pages ascl–1110, 2011.
[35] Pia Mukherjee, David Parkinson, and Andrew R. Liddle, A nested sampling algorithm for cosmological model selection, Astrophys. J. Lett. **638**, L51 (2006).
[36] Benjamin Audren, Julien Lesgourgues, Karim Benabed, and Simon Prunet, Conservative constraints on early cosmology with Monte Python, J. Cosmol. Astropart. Phys. 02 (2013) 001.
[37] Y. Akrami, F. Arroja, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak *et al.*, Planck 2018 results-X. Constraints on inflation, Astron. Astrophys. **641**, A10 (2020).
[38] I. Bernst, P. Schilke, T. Moeller, D. Panoglou, V. Ossenkopf, M. Roellig, J. Stutzki, and D. Muders, Magix: A generic tool for fitting models to astrophysical data, In *Astronomical Data Analysis Software and Systems XX* (Astronomical Society of the Pacific, San Francisco, 2011), Vol. 442, p. 505.
[39] J. Buchner, A. Georgakakis, K. Nandra, L. Hsu, C. Rangel, M. Brightman, A. Merloni, M. Salvato, J. Donley, and D. Kocevski, X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue, Astron. Astrophys. **564**, A125 (2014).
[40] Enrico Corsaro and Joris De Ridder, Diamonds: A new Bayesian nested sampling tool-application to peak bagging of solar-like oscillations, Astron. Astrophys. **571**, A71 (2014).
[41] Farhan Feroz, Jonathan R. Gair, Michael P. Hobson, and Edward K. Porter, Use of the MultiNest algorithm for gravitational wave data analysis, Classical Quantum Gravity **26**, 215003 (2009).
[42] Matthew Pitkin, Colin Gill, John Veitch, Erin Macdonald, and Graham Woan, A new code for parameter estimation in searches for gravitational waves from known pulsars, J. Phys. Conf. Ser. **363**, 012041 (2012).
[43] Walter Del Pozzo, John Veitch, and Alberto Vecchio, Testing general relativity using Bayesian model selection: Applications to observations of gravitational waves from compact binary systems, Phys. Rev. D **83**, 082002 (2011).
[44] Nick Pullen and Richard J. Morris, Bayesian model comparison and parameter inference in systems biology using nested sampling, PLoS One **9**, e88419 (2014).
[45] Stuart Aitken and Ozgur E. Akman, Nested sampling for parameter inference in systems biology: Application to an exemplar circadian model, BMC Syst. Biol. **7**, 72 (2013).
[46] Lívia B. Pártay, Albert P. Bartók, and Gábor Csányi, Nested sampling for materials: The case of hard spheres, Phys. Rev. E **89**, 022302 (2014).
[47] Robert John Nicholas Baldock, *Classical Statistical Mechanics with Nested Sampling* (Springer, New York, 2017).
[48] Béla Szekeres, Livia B. Partay, and Edit Mátyus, Direct computation of the quantum partition function by path-integral nested sampling, J. Chem. Theory Comput. **14**, 4353 (2018).
[49] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, Learning representations by back-propagating errors, Nature (London) **323**, 533 (1986).
[50] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Netw. **3**, 551 (1990).
[51] Michael A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, San Francisco, CA, USA, 2015), Vol. 25.
[52] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, *Deep Learning* (MIT Press, Cambridge, 2016), Vol. 1.
[53] Juan de Dios Rojas Olvera, Isidro Gómez-Vargas, and Jose Alberto Vázquez, Observational cosmology with artificial neural networks, Universe **8**, 120 (2022).
[54] Kerry Gallagher and Malcolm Sambridge, Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems, Comput. Geosci. **20**, 1229 (1994).
[55] S. N. Sivanandam and S. N. Deepa, *Genetic Algorithms* (Springer, New York, 2008).
[56] Colin R Reeves, Genetic algorithms for the operations researcher, INFORMS J. Comput. **9**, 231 (1997).
[57] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar, A review on genetic algorithm: Past present, and future, Multimedia Tools Appl. **80**, 8091 (2021).
[58] Ricardo Medel-Esquivel, Isidro Gómez-Vargas, Alejandro A. Morales Sánchez, Ricardo García-Salcedo, and José Alberto Vázquez, Cosmological parameter estimation with genetic algorithms, Universe **10**, 11 (2024).
[59] Diederik P. Kingma, Adam: A method for stochastic optimization, arXiv:1412.6980.
[60] Isidro Gómez-Vargas, Joshua Briones Andrade, and J. Alberto Vázquez, Neural networks optimized by genetic algorithms in cosmology, Phys. Rev. D **107**, 043509 (2023).
[61] J. A. Vazquez, I. Gomez-Vargas, and A. Slosar, Updated version of a simple MCMC code for cosmological parameter estimation where only expansion history matters, https://github.com/ja-vazquez/SimpleMC (2024).
[62] Éric Aubourg, Stephen Bailey, Julian E. Bautista, Florian Beutler, Vaishali Bhardwaj, Dmitry Bizyaev, Michael Blanton, Michael Blomqvist, Adam S. Bolton, Jo Bovy *et al.*, Cosmological implications of baryon acoustic oscillation measurements, Phys. Rev. D **92**, 123516 (2015).
[63] Joshua S. Speagle, `dynesty`: A dynamic nested sampling package for estimating Bayesian posteriors and evidences, Mon. Not. R. Astron. Soc. **493**, 3132 (2020).
[64] Salvatore Ingrassia and Isabella Morlini, Neural network modeling for small datasets, Technometrics **47**, 297 (2005).
[65] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler, Deep learning for emotion recognition on small datasets using transfer learning, in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (Association for Computing Machinery, New York, 2015), pp. 443–449.
[66] Antonello Pasini, Artificial neural networks for small dataset analysis, J. Thorac. Dis. **7**, 953 (2015).
[67] Isidro Gómez-Vargas, Ricardo Medel-Esquivel, Ricardo García-Salcedo, and J. Alberto Vázquez, Neural network reconstructions for the Hubble parameter, growth rate and distance modulus, Eur. Phys. J. C **83**, 304 (2023).

[68] Michel Chevallier and David Polarski, Accelerating universes with scaling dark matter, Int. J. Mod. Phys. D **10**, 213 (2001).
[69] Eric V. Linder, Exploring the expansion history of the universe, Phys. Rev. Lett. **90**, 091301 (2003).
[70] Daniel Moshe Scolnic, D. O. Jones, A. Rest, Y. C. Pan, R. Chornock, R. J. Foley, M. E. Huber, R. Kessler, Gautham Narayan, A. G. Riess *et al.*, The complete light-curve sample of spectroscopically confirmed SNe Ia from Pan-STARRS1 and cosmological constraints from the Combined Pantheon Sample, Astrophys. J. **859**, 101 (2018).
[71] Raul Jimenez, Licia Verde, Tommaso Treu, and Daniel Stern, Constraints on the equation of state of dark energy and the Hubble constant from stellar ages and the cosmic microwave background, Astrophys. J. **593**, 622 (2003).
[72] Joan Simon, Licia Verde, and Raul Jimenez, Constraints on the redshift dependence of the dark energy potential, Phys. Rev. D **71**, 123001 (2005).
[73] Daniel Stern, Raul Jimenez, Licia Verde, Marc Kamionkowski, and S. Adam Stanford, Cosmic chronometers: Constraining the equation of state of dark energy. I: $h(z)$ measurements, J. Cosmol. Astropart. Phys. 02 (2010) 008.
[74] Michele Moresco, Licia Verde, Lucia Pozzetti, Raul Jimenez, and Andrea Cimatti, New constraints on cosmological parameters and neutrino properties using the expansion rate of the universe to $z \sim 1.75$, J. Cosmol. Astropart. Phys. 07 (2012) 053.
[75] Cong Zhang, Han Zhang, Shuo Yuan, Siqi Liu, Tong-Jie Zhang, and Yan-Chun Sun, Four new observational $h(z)$ data from luminous red galaxies in the Sloan digital sky survey data release seven, Res. Astron. Astrophys. **14**, 1221 (2014).
[76] Michele Moresco, Raising the bar: New constraints on the Hubble parameter with cosmic chronometers at $z \sim 2$, Mon. Not. R. Astron. Soc. **450**, L16 (2015).
[77] Michele Moresco, Lucia Pozzetti, Andrea Cimatti, Raul Jimenez, Claudia Maraston, Licia Verde, Daniel Thomas, Annalisa Citro, Rita Tojeiro, and David Wilkinson, A 6% measurement of the Hubble parameter at $z \sim 0.45$: Direct evidence of the epoch of cosmic re-acceleration, J. Cosmol. Astropart. Phys. 05 (2016) 014.
[78] A. L. Ratsimbazafy, S. I. Loubser, S. M. Crawford, C. M. Cress, B. A. Bassett, R. C. Nichol, and P. Väisänen, Age-dating luminous red galaxies observed with the Southern African Large Telescope, Mon. Not. R. Astron. Soc. **467**, 3239 (2017).
[79] Ashley J. Ross, Lado Samushia, Cullan Howlett, Will J. Percival, Angela Burden, and Marc Manera, The clustering of the SDSS DR7 main galaxy sample–I. A 4 per cent distance measure at $z = 0.15$, Mon. Not. R. Astron. Soc. **449**, 835 (2015).
[80] Florian Beutler, Chris Blake, Matthew Colless, D. Heath Jones, Lister Staveley-Smith, Lachlan Campbell, Quentin Parker, Will Saunders, and Fred Watson, The 6dF galaxy survey: Baryon acoustic oscillations and the local Hubble constant, Mon. Not. R. Astron. Soc. **416**, 3017 (2011).
[81] Shadab Alam, Metin Ata, Stephen Bailey, Florian Beutler, Dmitry Bizyaev, Jonathan A. Blazek, Adam S. Bolton, Joel R. Brownstein, Angela Burden, Chia-Hsun Chuang *et al.*, The clustering of galaxies in the completed SDSS-III baryon oscillation spectroscopic survey: Cosmological analysis of the DR12 galaxy sample, Mon. Not. R. Astron. Soc. **470**, 2617 (2017).
[82] Metin Ata, Falk Baumgarten, Julian Bautista, Florian Beutler, Dmitry Bizyaev, Michael R. Blanton, Jonathan A. Blazek, Adam S. Bolton, Jonathan Brinkmann, Joel R. Brownstein *et al.*, The clustering of the SDSS-IV extended baryon oscillation spectroscopic survey DR14 quasar sample: First measurement of baryon acoustic oscillations between redshift 0.8 and 2.2, Mon. Not. R. Astron. Soc. **473**, 4773 (2018).
[83] Michael Blomqvist, Hélion Du Mas Des Bourboux, Victoria de Sainte Agathe, James Rich, Christophe Balland, Julian E Bautista, Kyle Dawson, Andreu Font-Ribera, Julien Guy, Jean-Marc Le Goff *et al.*, Baryon acoustic oscillations from the cross-correlation of Lyα absorption and quasars in eBOSS DR14, Astron. Astrophys. **629**, A86 (2019).
[84] Victoria de Sainte Agathe, Christophe Balland, Hélion Du Mas Des Bourboux, Michael Blomqvist, Julien Guy, James Rich, Andreu Font-Ribera, Matthew M Pieri, Julian E Bautista, Kyle Dawson *et al.*, Baryon acoustic oscillations at $z = 2.34$ from the correlations of lyα absorption in eBOSS DR14, Astron. Astrophys. **629**, A85 (2019).
[85] Bryan Sagredo, Savvas Nesseris, and Domenico Sapone, Internal robustness of growth rate data, Phys. Rev. D **98**, 083543 (2018).
[86] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi, On Wasserstein two-sample testing and related families of nonparametric tests, Entropy **19**, 47 (2017).
[87] https://github.com/igomezv/neuralike (2024).
[88] David W. Hogg and Daniel Foreman-Mackey, Data analysis recipes: Using Markov chain Monte Carlo, Astrophys. J. Suppl. Ser. **236**, 11 (2018).