

Faster Bayesian inference with neural network bundles and new results for $f(R)$ models

Augusto T. Chantada^{1,*}, Susana J. Landau², Pavlos Protopapas³, Claudia G. Scóccola^{4,5} and Cecilia Garraffo⁶

¹*Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Física, Avenida Intendente Cantilo S/N, 1428, Ciudad Autónoma de Buenos Aires, Argentina*

²*CONICET—Universidad de Buenos Aires, Instituto de Física de Buenos Aires (IFIBA), Avenida Intendente Cantilo S/N, 1428, Ciudad Autónoma de Buenos Aires, Argentina*

³*John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*

⁴*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Godoy Cruz 2290, 1425, Ciudad Autónoma de Buenos Aires, Argentina*

⁵*Facultad de Ciencias Astronómicas y Geofísicas, Universidad Nacional de La Plata, Observatorio Astronómico, Paseo del Bosque, B1900FWA La Plata, Argentina*

⁶*Center for Astrophysics / Harvard and Smithsonian, 60 Garden Street, Cambridge, Massachusetts 02138, USA*



(Received 28 November 2023; accepted 8 May 2024; published 7 June 2024)

In the last few years, there has been significant progress in the development of machine learning methods tailored to astrophysics and cosmology. We have recently applied one of these, namely, the neural network bundle method, to the cosmological scenario. Moreover, we showed that in some cases the computational times of the Bayesian inference process can be reduced. In this paper, we present an improvement to the neural network bundle method that results in a significant reduction of the computational times of the statistical analysis. The novel aspect consists of the use of the neural network bundle method to calculate the luminosity distance of type Ia supernovae, which is usually computed through an integral with numerical methods. In this work, we have applied this improvement to the Hu-Sawicki and Starobinsky $f(R)$ models. We also performed a statistical analysis with data from type Ia supernovae of the Pantheon+ compilation and cosmic chronometers. Another original aspect of this work is the different treatment we provide for the absolute magnitude of type Ia supernovae during the inference process, which results in different estimates of the distortion parameter than the ones obtained in the literature. We show that the statistical analyses carried out with our new method require lower computational times than the ones performed with both the numerical and the neural network method from our previous work. This reduction in time is more significant in the case of a difficult computational problem such as the ones addressed in this work.

DOI: [10.1103/PhysRevD.109.123514](https://doi.org/10.1103/PhysRevD.109.123514)

I. INTRODUCTION

One of the central challenges in theoretical cosmology lies in identifying the physical mechanism responsible for the current accelerated expansion of the Universe. According to the standard cosmological model Λ CDM, this acceleration can be explained by including a cosmological constant in Einstein equations. Although this model successfully explains a broad set of existing observational data [1], it falls short in accounting for the value of the cosmological constant inferred from observations. Furthermore, there are some inconsistencies within the Λ CDM framework, such as the so-called Hubble tension, a disagreement between the value of the Hubble constant inferred from data of the cosmic

microwave background (CMB) assuming the standard cosmological model [1], and the one obtained from type Ia supernovae and Cepheid data [2], which are model independent.

Motivated by this observational discrepancy, researchers explore alternative cosmological models to account for the current accelerated expansion and resolve the Hubble tension. These alternative frameworks generally fall into two families: (i) dark energy and (ii) modified gravity. In dark energy models, a new component is added to the energy-momentum tensor, either in the form of a fluid with a time-dependent equation of state or a scalar field that is minimally coupled to gravity with a specific potential. On the contrary, modified gravity theories propose alternatives to general relativity to describe the gravitational interaction. One such example are $f(R)$ theories, which introduce modifications to

*augustochantada01@gmail.com

the Einstein-Hilbert action in the form of a function of the Ricci scalar. Typically, the differential systems that describe the cosmological dynamics in these modified gravity models are more complex than those of the standard Λ CDM model. Consequently, the computational times to perform parameter inference in these alternative models are significantly larger than those needed for the Λ CDM model.

Recently, neural networks (NNs) have gained traction as powerful tools for handling both data and models. In cosmology these have been employed to extract additional information from existing data in order to constrain cosmological models [3–6] and to create faster Monte Carlo samplers for Bayesian inference [7,8]. Other applications to cosmology include training NNs as emulators of Einstein-Boltzmann solvers [9,10]. Additionally, unsupervised methods for solving differential equations have been developed [11] and used in applications like physics-informed neural networks [12]. A notable extension of these methods is the NN bundle method [13], which generates a set of solutions of a given differential system. The advantage of this method is that the outcome of the NN method can be regarded as a function of the independent variable and the free parameters of the differential system. Moreover, this method does not require the use of traditional numerical solvers to obtain the solutions.

We have previously demonstrated the efficacy of this method in solving the background dynamics of the Universe for four different cosmological models [14]. We validated the solutions obtained through the NN bundle method by comparing them with solutions from a traditional numerical solver. The main advantage of this method is that the integration of the differential system is performed only once, and thereafter the solutions can be used indefinitely for parameter inference. This reduces the computational time for statistical analysis using the Markov Chain Monte Carlo (MCMC) algorithm, particularly for cosmological models that are computationally intensive to integrate.

In this work, we take a step further by demonstrating that using the NN method to compute the luminosity distance significantly reduces the time required for the inference process. To illustrate this, we consider the $f(R)$ Hu-Sawicki model, which we tackled in our previous work, and additionally we analyze the Starobinsky model. The latter is computationally more intensive to integrate than the former and therefore represents a more challenging test for the NN method. We test the predictions of the background dynamics of these models with recent data from cosmic chronometers [15–22] and type Ia supernovae (SNIa) [23]. The originality of our statistical analysis lies in the choice of the prior of SNIa absolute magnitude, which is different than the ones adopted in the literature for the same models and datasets [24]. This results in different values of the inferred distortion parameter b of the $f(R)$ model.

The structure of this article is as follows. In Sec. II, we briefly describe the basics of both the Hu-Sawicki and Starobinsky $f(R)$ models. In Sec. III, we describe the details

of the NN bundle method when applied to both $f(R)$ models. We also describe the improvements to the inference process that involve the NN bundle method and leads to the optimization of computational times. Afterward, we show a comparison between the solutions obtained with the NN method for the $f(R)$ models and the ones obtained with a traditional numerical solver. Additionally, we describe the appropriate treatment of the prior assumed for the absolute magnitude of SNIa. In Sec. IV, we show the results of the statistical analysis where we compare the predictions of our model with data from cosmic chronometers and type Ia supernovae. We also discuss our results as compared with similar analyses [24,25]. Most importantly, we show the significant time reduction of computational times achieved in this study compared to both our previous work and the numerical method. In Sec. V, we present our conclusions.

II. THEORETICAL MODELS

In this paper, we focus on a particular class of modified gravity theories, namely, $f(R)$ theories, whose action can be written as

$$S = \frac{1}{2\kappa} \int d^4x \sqrt{-g} f(R) + S_m. \quad (1)$$

Here $\kappa = 8\pi G$ (we adopt $c = 1$ throughout this paper), R is the Ricci scalar, g is the metric determinant, $f(R)$ is an arbitrary function of R , and S_m refers to the matter action. From Eq. (1), the modified Einstein equations can be derived.

The assumption of an isotropic and homogeneous universe, which is the usual cosmological scenario, naturally leads us to use the Friedmann-Lemaître-Robertson-Walker metric. By applying the modified Einstein equations to the cosmological setting, we obtain the following modified Friedmann equations, which describe the dynamical evolution of the background:

$$\begin{aligned} H^2 &= \frac{1}{3f_R} \left[\kappa\rho_m + \frac{Rf_R - f}{2} - 3H\dot{R}f_{RR} \right], \\ 2\dot{H} + 3H^2 &= -\frac{1}{f_R} \left[-\frac{Rf_R - f}{2} + f_{RRR}\dot{R}^2 \right. \\ &\quad \left. + (\ddot{R} + 2H\dot{R})f_{RR} \right], \end{aligned} \quad (2)$$

where we use the notation $f = f(R)$, $f_R = df/dR$, $f_{RR} = d^2f/dR^2$, etc., and the dot refers to the derivative with respect to cosmic time. Also, ρ_m is the total matter energy density and H refers to the Hubble parameter.

In this paper, we analyze the Hu-Sawicki and Starobinsky models with $n = 1$. For the Hu-Sawicki model [26],

$$f_{\text{HS}}(R) = R - 2\Lambda \left[1 - \frac{1}{1 + \frac{R}{\Lambda b}} \right], \quad (3)$$

while for the Starobinsky model [27],

$$f_S(R) = R - 2\Lambda \left[1 - \frac{1}{1 + \left(\frac{R}{\Lambda b}\right)^2} \right]. \quad (4)$$

Here Λ is the cosmological constant and b is the distortion parameter that measures the model's departure from Λ CDM. To solve the Friedmann equations, we assume the following change of variables [14,28]:

$$\begin{aligned} x &= \frac{\dot{R}f_{RR}}{Hf_R}, & v &= \frac{R}{6H^2}, & y &= \frac{f}{6H^2f_R}, \\ \Omega &= \frac{\kappa\rho_m}{3H^2f_R}, & r &= \frac{R}{\Lambda}. \end{aligned} \quad (5)$$

In this way, the Friedmann equations can be written as follows:

$$\begin{cases} \frac{dx}{dz} = \frac{1}{1+z} (-\Omega - 2v + x + 4y + xv + x^2) \\ \frac{1}{\Gamma} \frac{dy}{dz} = \frac{-1}{1+z} \left(vx - \frac{xy + 4y - 2yv}{\Gamma} \right) \\ \frac{1}{\Gamma} \frac{dv}{dz} = \frac{-v}{1+z} \left(x + \frac{4-2v}{\Gamma} \right) \\ \frac{d\Omega}{dz} = \frac{\Omega}{1+z} (-1 + 2v + x) \\ \frac{1}{\Gamma} \frac{dr}{dz} = \frac{-rx}{1+z}, \end{cases} \quad (6)$$

where the factor $\Gamma = \frac{f_R}{Rf_{RR}}$. If we assume the Hu-Sawicki model, Γ can be expressed as follows:

$$\Gamma_{\text{HS}}(r) = \frac{(r+b) \left[(r+b)^2 - 2b \right]}{4br}, \quad (7)$$

while for the Starobinsky model,

$$\Gamma_S(r) = \frac{(r^2 + b^2) \left[(r^2 + b^2)^2 - 4b^2r \right]}{4rb^2(3r^2 - b^2)}. \quad (8)$$

To facilitate the implementation of the method, an additional variable change ($r' = \ln r$) is done in Eq. (6) before the training of the NN.

The initial conditions of the system were chosen so that at high redshift the model behaves as the Λ CDM one; in this work for obtaining the solutions with the NN method we use $z_0 = 10$, where z_0 is the redshift at which the initial conditions of Eq. (6) are set (see Sec. III B for a discussion of the initial conditions). For further details on the change of variables and initial conditions, we refer the reader to our previous work [14].

Note that the second, third, and fifth equations in the differential system in Eq. (6) are expressed nontraditionally, that is, the factor Γ appears in the denominator, on the left-hand side of the equation. This way of writing the

differential system was used only for the Starobinsky model and the reason for this will be made clearer in Sec. III.

To compare the theoretical model with the observations, the Hubble parameter is needed. Therefore, after solving the system of equations described in Eq. (6), one can obtain the solution for H using the following relation:

$$H = H_0^\Lambda \sqrt{\frac{r}{2v} (1 - \Omega_{m,0}^\Lambda)}. \quad (9)$$

Here $\Omega_{m,0}^\Lambda$ is the matter density parameter in the Λ CDM model which should be distinguished from $\Omega_{m,0}$, the matter density parameter defined in the $f(R)$ model. Similarly, H_0^Λ denotes the Hubble constant in the Λ CDM model, while H_0 is the same quantity in the $f(R)$ model. Since the matter energy density is an observable quantity, the following relation holds:

$$\Omega_{m,0} H_0^2 = \Omega_{m,0}^\Lambda (H_0^\Lambda)^2. \quad (10)$$

III. METHOD

A. The new NN bundle method

In our previous work [14], we extensively discussed the application of the NN bundle method in the cosmological context. Below, we recall the main aspects of the method. First, we describe the method for the case in which neither the differential system nor the initial condition depends on any parameter. The NN method can be formulated as an optimization problem. The goal is to tune the networks' internal parameters to minimize a specific loss function. We denote the vector of the outputs of the NNs as $\mathbf{u}_{\mathcal{N}}(t)$, where t is the independent variable. The loss function to be minimized is then¹

$$L(\tilde{\mathbf{u}}, t) = \sum_i^M \mathcal{R}_i(\tilde{\mathbf{u}}, t)^2, \quad (11)$$

where \mathcal{R}_i are the residuals of the M differential equations, i.e., the left-hand side of the i th differential equation minus its right-hand side,² and $\tilde{\mathbf{u}}(t)$ is a reparametrization of the outputs $\mathbf{u}_{\mathcal{N}}(t)$ so as to enforce the initial condition \mathbf{u}_0 .

¹Actually, the loss function in Eq. (11) is evaluated at N_{batch} random points of the domain in each iteration of the optimization process so that the total loss at each iteration is $J = \frac{1}{N_{\text{batch}}} \sum_i^{N_{\text{batch}}} L(\tilde{\mathbf{u}}(t_i), t_i)$.

²It follows that, if $\tilde{\mathbf{u}}$ is the exact solution, it satisfies the differential system and therefore the loss function is exactly zero. Since the outcomes of the NN method are not exact solutions, the loss measures how far the solutions are from satisfying the differential system. For details, see Ref. [14].

$$\tilde{\mathbf{u}}(t) = \mathbf{u}_0 + (1 - e^{-(t-t_0)})\mathbf{u}_{\mathcal{N}}(t). \quad (12)$$

Now, we move to the NN bundle method, which is applied when the differential system and/or the initial conditions depend on free parameters. We will call $\boldsymbol{\theta}$ the vector that includes all those parameters. In this case, the loss function to be optimized reads³

$$L(\tilde{\mathbf{u}}, t, \boldsymbol{\theta}) = \sum_i^M \mathcal{R}_i(\tilde{\mathbf{u}}, t, \boldsymbol{\theta})^2, \quad (13)$$

and $\tilde{\mathbf{u}}(t, \boldsymbol{\theta})$ is a reparametrization of the outputs $\mathbf{u}_{\mathcal{N}}(t, \boldsymbol{\theta})$ so as to enforce the initial condition $\mathbf{u}_0(\boldsymbol{\theta})$,

$$\tilde{\mathbf{u}}(t, \boldsymbol{\theta}) = \mathbf{u}_0(\boldsymbol{\theta}) + (1 - e^{-(t-t_0)})\mathbf{u}_{\mathcal{N}}(t, \boldsymbol{\theta}). \quad (14)$$

In this way, when the optimization process finishes, the trained neural network represents the solution of the differential system for any value of the free parameters $\boldsymbol{\theta}$ within the training range. This is different from numerical methods, where the system needs to be integrated for every choice of the free parameters.

With regards to the unsupervised nature of the training, the key to understanding it lies in the residuals, which are calculated from the differential equations of the system. In this way, each residual depends on the independent variable, the bundle parameters, the evaluated reparametrized networks, and nothing else. At each iteration, the residuals are evaluated at random points of the domain (independent variable and bundle parameters). This calculation involves evaluating the reparametrized networks and their derivatives⁴ (with respect to the independent variable) at these points. Then, the value of the loss is calculated, and the internal parameters of the network are tuned in order to lower it.

To implement the method to solve the background equations of the $f(R)$ models, we use the loss described in Eq. (13).⁵ When the solution of a given differential system is known for a fixed value of one or more of the free parameters, it is useful to consider the perturbative reparametrization that was proposed and applied in our previous

³Actually, the loss function in Eq. (13) is evaluated at N_{batch} random points of the domain in each iteration of the optimization process so that the total loss at each iteration is $J = \frac{1}{N_{\text{batch}}} \sum_i^{N_{\text{batch}}} L(\tilde{\mathbf{u}}, t_i, \boldsymbol{\theta}_i)$.

⁴These are not to be confused with derivatives involving the stochastic gradient descent algorithm used to optimize the networks. The derivatives mentioned here are the ones necessary to compute the residuals, i.e., the derivatives that appear in the differential equations.

⁵We also add an extra term to the loss in Eq. (13) to explicitly enforce relationships between the dependent variables that must be satisfied (see Appendix B 3 of our previous work [14] for more information).

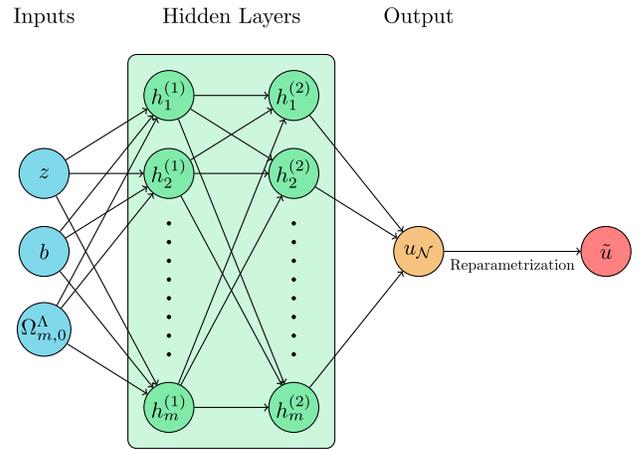


FIG. 1. Graph that represents a neural network like the ones employed in this work. It has three inputs that correspond to the independent variable and the two parameters of the bundle, two hidden layers with the same amount of units each, and a single output that is then reparametrized to satisfy the initial condition.

work [14] to the quintessence and $f(R)$ Hu-Sawicki models. When applied to the $f(R)$ models that are concerned here, this takes the form

$$\tilde{\mathbf{u}}(z, b, \Omega_{m,0}^\Lambda) = \hat{\mathbf{u}}(z, \Omega_{m,0}^\Lambda) + (1 - e^{-(z-z_0)}) \times (1 - e^{-b})\mathbf{u}_{\mathcal{N}}(z, b, \Omega_{m,0}^\Lambda). \quad (15)$$

Here $\hat{\mathbf{u}}(z, \Omega_{m,0}^\Lambda)$ denotes the Λ CDM solution of Eq. (6) (i.e., for $b \rightarrow 0$) which is already known, and the job of the second term is to correct it in order to make the solution valid for the whole bundle. We recall that the bundle parameters are b and $\Omega_{m,0}^\Lambda$.⁶ In Fig. 1 we show a graph that represents the neural network described before.

This reparametrization helps to circumvent the singularity of the differential system at $b = 0$ [see Eqs. (7) and (8)]. In contrast, for numerical methods, an approximate analytical function is usually used when b approaches 0, since the singularity causes an increase in the computational times of the statistical analysis [29].

Regarding the implementation of the method to the Starobinsky model, it is important to highlight our approach in formulating Eq. (6). Traditional numerical solvers typically require differential equations to be arranged with the derivative of the independent variable isolated on one side and the rest on the other side. However, the NN method is not bound by this constraint; the residual can be structured in any way that preserves the same equality that the equation conveys. Thus, we chose to define the residuals based on the formulation presented in

⁶The parameter b enters in Eq. (6) through Γ [Eqs. (7) and (8)], while the initial conditions depend on $\Omega_{m,0}^\Lambda$ (see our previous work [14]).

Eq. (6), inspired by the fact that the singularity in Γ makes the balance between the equations uneven. Consequently, we use the residuals that correspond to the second, third, and fifth equations of Eq. (6). Therefore, these residuals no longer exhibited a divergent behavior close to $b = 0$. This shows a useful flexibility that the NN method has that numerical methods do not. This last technique was needed in addition to the perturbative reparametrization to make the solutions accurate at low values of b for the Starobinsky model. In the case of the Hu-Sawicki model, the differential equations and, in consequence, the residuals and the loss are written in the usual way (see our previous paper for details [14]).

In this work, we present an improvement that significantly reduces the time it takes to perform an MCMC analysis. This improvement is related to the computation of the luminosity distance of type Ia supernovae,

$$d_L(z) = (1+z) \int_0^z \frac{dz'}{H(z')}. \quad (16)$$

Building upon the previous equation, it becomes evident that we must carry out an integration involving an integrand $\frac{1}{H}$, derived from the output of the NN method. In the context of numerical integration methods, the integrand (which in this case is a function of the NN) must be evaluated at N_{int} different values of z . The computational time for this process is directly proportional to N_{int} . Furthermore, it is important to note that the accuracy of the numerical integration method improves with higher values of N_{int} . Therefore, this poses a computational bottleneck, particularly when this integral must be recalculated at each parameter sample in the MCMC process.

To mitigate this computational challenge, one can consider NNs once again to evaluate the integral. Using the fundamental theorem of calculus, we can transform the integral equation into a differential equation. This way, we define

$$\frac{dI}{dz} = \frac{1}{E(z)}, \quad I(z)|_{z=0} = 0, \quad (17)$$

where $E(z) = H(z)/H_0^\Lambda$ and $I(z) = d_L(z)H_0^\Lambda/(1+z)$. Next we describe the specific steps of our new NN method. First, a set of NNs are trained to obtain $E(z, b, \Omega_{m,0}^\Lambda)$ from Eq. (6). The trained NNs are then used in Eq. (17) to define another differential equation which is also solved with the NN bundle method. Therefore, after training, we obtain a new NN that represents $I(z, b, \Omega_{m,0}^\Lambda)$. In this way, the integration that is needed to obtain the luminosity distance is no longer computed during the MCMC process. Instead, it is performed in the NN training that yields $I(z, b, \Omega_{m,0}^\Lambda)$ (i.e., a bundle solution of the integral), resulting in a reduction of the computational times of the inference process. We also note that, in this way, the likelihood is

completely analytical since all of the theoretical predictions of the observable quantities come from NNs that work as functions of the parameters and the independent variable. In short, with our new approach the NN bundle method is applied twice: first to solve the differential system and second to compute the luminosity distance. On the contrary, in our previous work [14], we only used the NN bundle method to solve the differential system, while the luminosity distance was computed with numerical methods.

The concept of using NNs to perform an integration has been explored by other authors [30–33]. Specifically, the idea of using the fundamental theorem of calculus to turn an integral equation into a differential one and after that use a NN-based method to solve it has been explored in Refs. [31–33]. Nevertheless, our work is the first one to extend this concept to the case where the integrand is a solution previously obtained by the NN bundle method. Indeed, the integrand of Eq. (17) is computed from the solution of the differential system described by Eq. (6).

B. Accuracy of solutions

All of the NNs in this work have three inputs (corresponding to z , b , and $\Omega_{m,0}^\Lambda$), two hidden layers of 32 units each, and one output for a total of 1217 trainable parameters per NN. Each NN has a single output so that, when reparametrized as in Eq. (15), it should approximate the associated dependent variable that solves the corresponding differential equation. It should be noted that the specific architecture used in this work was chosen based on its simplicity and effectiveness. Indeed, it has proven to be effective with a relatively low amount of trainable parameters.

The training ranges for the input parameters of the NNs are as follows: for the Hu-Sawicki model, $z \in [0, 10]$, $b \in (0, 3]$, and $\Omega_{m,0}^\Lambda \in [0.05, 0.4]$, while for the Starobinsky model, they are $z \in [0, 10]$, $b \in (0, 4]$, and $\Omega_{m,0}^\Lambda \in [0.1, 0.4]$.

When the training is finished, the NN internal parameters adopt the values that correspond to the lowest value of the total loss. This latter value resulted to be $L_{\text{min}} = 1.14 \times 10^{-5}$ [for Eq. (6)] and $L_{\text{min}} = 5.43 \times 10^{-9}$ [for Eq. (17)] in the Hu-Sawicki model, while they turned out to be $L_{\text{min}} = 2.19 \times 10^{-6}$ [for Eq. (6)] and $L_{\text{min}} = 8.16 \times 10^{-9}$ [for Eq. (17)] in the Starobinsky model.

However, the training of the NNs does not give any information about the error in the quantities that we are interested in (H and d_L). For this reason, we evaluate the precision of the outputs given by the NNs by computing the absolute value of the relative difference between these NN-based solutions and the ones obtained from a numerical solver. In particular, we used the Runge-Kutta methods⁷ found in SciPy's [34] tool to solve initial value problems and

⁷We used RKF45 for the Hu-Sawicki model and Radau IIA family of order 5 for the Starobinsky model. The latter method is meant for stiff problems and it was necessary to use it for the convergence of the Starobinsky model.

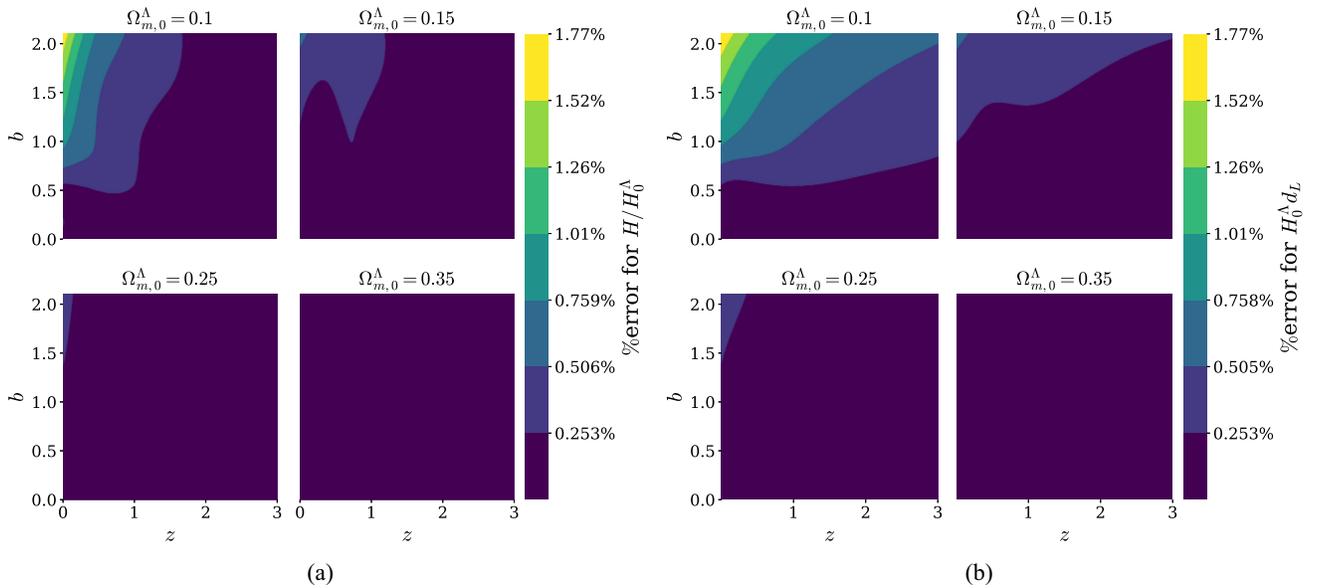


FIG. 2. The percentage error of the NN-based bundle solution of H/H_0^Λ (a) and $H_0^\Lambda d_L$ (b) in the Hu-Sawicki model with $n = 1$, through comparison to numerical solutions. The range of the comparison goes through a section of the training range of the parameters of the bundle.

consider low tolerance for both relative and absolute errors, 10^{-7} and 10^{-12} , respectively. We chose the Runge-Kutta method for being well-established and tested, as well as for its presence in a widely used library.

Now, we focus on two specific quantities, H/H_0^Λ and $H_0^\Lambda d_L$. The results are shown in Figs. 2 and 3, where the range of the parameters covers the region of the parameter space that is within the 95% confidence level,

$(b, \Omega_{m,0}^\Lambda) \in (0, 2.1] \times [0.1, 0.35]$ and $(b, \Omega_{m,0}^\Lambda) \in (0, 3.2] \times [0.2, 0.35]$ for Hu-Sawicki and Starobinsky models, respectively, as we will later show in Sec. III C. The figures also show the dependence with redshift in the region relevant to the datasets we are considering. These plots show that the errors are, at most, $\sim 1.8\%$ and $\sim 0.8\%$ in the region of the parameter space evaluated, for Hu-Sawicki and Starobinsky models, respectively.

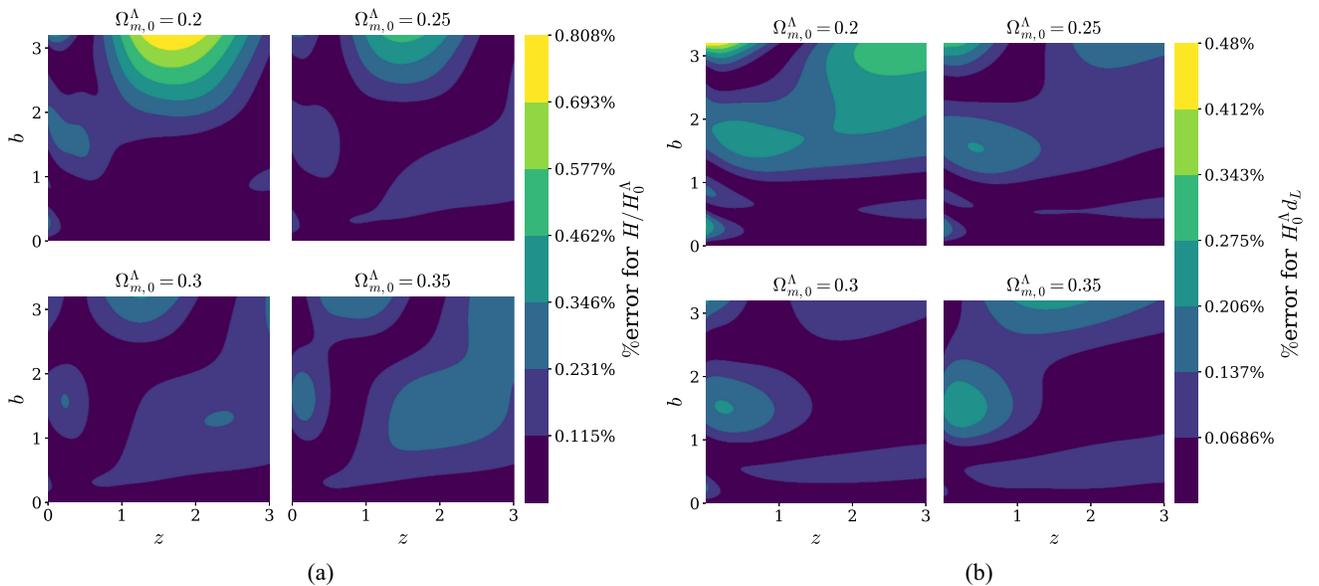


FIG. 3. The percentage error of the NN-based bundle solution of H/H_0^Λ (a) and $H_0^\Lambda d_L$ (b) in the Starobinsky model with $n = 1$, through comparison to numerical solutions. The range of the comparison goes through a section of the training range of the parameters of the bundle.

The choice of the initial condition deserves a detailed explanation. The bottom line is that we used $z_0 = 10$ for the NN method for both models, but in the case of the numerical method, we could only use this value for the Hu-Sawicki model, while for the Starobinsky model we needed to compute the initial condition more carefully, as we explain in what follows.

Specifically, Ref. [24] highlights that using a fixed and low value of z_0 can lead to numerical instabilities in some areas of the parameter space. We verified by ourselves that there are instabilities in the numerical method in specific regions of the parameter space when using the initial condition $z_0 = 10$ in the case of the Starobinsky model. On the other hand, a more thorough method for computing z_0 was developed in Refs. [25,35], where its value depends on both b and $\Omega_{m,0}^\Lambda$. For the Starobinsky model with $n = 1$, this is

$$z_0 = \left[\frac{1 - \Omega_{m,0}^\Lambda - \Omega_{r,0}^\Lambda}{\Omega_{m,0}^\Lambda} \left(b \sqrt{\frac{1}{\epsilon} - 1 - 4} \right) \right]^{1/3} - 1 \quad (18)$$

with $\Omega_{r,0}^\Lambda = 2.97 \times 10^{-4} \Omega_{m,0}^\Lambda$, and $\epsilon \ll 1$.⁸ This alternative method can lead to significantly higher values of z_0 , sometimes on the order of 100. For such high z_0 values, the radiation component of the Universe can no longer be neglected in the Friedmann equations and, consequently, in the initial conditions. Also, a high value of z_0 results in an increase in the computing times of the solution. Nevertheless, using Eq. (18) to determine the initial condition resolved the numerical issues, and therefore we used the initial condition taken from Eq. (18) in all calculations performed with the numerical method for the Starobinsky model. We stress that for the Hu-Sawicki model we always use $z_0 = 10$. This discussion is relevant to understand the comparison between the results of the inference process obtained using the numerical method with those obtained using NNs (as shown later) and for the estimation of the computing times of both methods.

Going back to Fig. 3, which displays the comparison between the NN and the numerical solutions for the Starobinsky model, we notice that the solutions obtained with the NN method using the initial condition at $z_0 = 10$ closely match those obtained using the numerical method, which employs Eq. (18). This agreement is observed within the parameter space region encompassing the 95% confidence level. Hence, one advantage of the NN method is its ability to achieve stability without requiring a more comprehensive initial condition. A comparison between $z_0 = 10$ and $z_0 = z_0(b, \Omega_{m,0}^\Lambda)$ with the numerical method revealed no significant differences, with variations

remaining below $\sim 1.74\%$ for the Hu-Sawicki model and $\sim 0.06\%$ for the Starobinsky model in the same parameter space region as the one shown in Figs. 2 and 3. However, due to stability issues during the computations of the MCMC process, it was required to use the more complex initial condition for the Starobinsky case when using the numerical method.

C. Statistical analysis

1. Datasets

In this section we briefly describe the datasets used in the statistical analysis of this work: cosmic chronometers and SNIa.

We use 32 measurements [15–22] of H that were obtained using the cosmic chronometers technique within a range of redshift $z \in [0.07, 1.965]$. The likelihood function is

$$\mathcal{L}_{\text{CC}} \propto \exp \left\{ -\frac{1}{2} \sum_{i=1}^{32} \left[\frac{H^{\text{obs}}(z_i) - H^{\text{th}}(z_i, b, \Omega_{m,0}^\Lambda)}{\sigma_{H^{\text{obs}}(z_i)}} \right]^2 \right\}, \quad (19)$$

where $H^{\text{obs}}(z_i)$ refers to the observational measurements of $H(z)$ and $\sigma_{H^{\text{obs}}(z_i)}$ refers to the corresponding error, while $H^{\text{th}}(z_i, b, \Omega_{m,0}^\Lambda)$ is the model's prediction that is obtained by evaluating two trained NNs [one corresponding to $v(z_i, b, \Omega_{m,0}^\Lambda)$ and another to $r(z_i, b, \Omega_{m,0}^\Lambda)$ ⁹ in Eq. (9)]. We emphasise that these two NNs are trained together with another three, representing the five variables in Eq. (5), to solve Eq. (6).

It is widely accepted that SNIa can be considered as standard candles due to the homogeneity of their light curves. This makes them ideal for determining distances and also to constrain cosmological parameters. From the supernovae light curves, it is possible to infer the distance moduli μ , which can be expressed as

$$\mu_{\text{obs}} = \tilde{m}_b - M, \quad (20)$$

where \tilde{m}_b refers to the corrected overall flux normalization (see Appendix A) and M is the absolute magnitude. This last equation is a simplified version of the Tripp formula [36], which includes additional corrections described by extra parameters that will not be used in this work. In Appendix A, we describe the complete Tripp formula and the reasons that lead us to use Eq. (20) in this work. On the other hand, the distance moduli are related to the luminosity distance [Eq. (16)] as

$$\mu(z) = 25 + 5 \log_{10}(d_L(z)). \quad (21)$$

⁸The value of ϵ is usually determined by imposing $f[R(z = z_0)] = R - 2\Lambda(1 - \epsilon)$, thus it depends on the specific $f(R)$ being used [25]. We use the values used in Ref. [24].

⁹We recall that the actual dependant variable used is $r' = \ln r$.

In this work, we consider 1590 SNIa, at redshifts $z \in [0.01, 2.26]$ from the Pantheon+ compilation [23].

The likelihood function can be written as

$$\mathcal{L}_{\text{SNIa}} \propto \exp \left\{ -\frac{1}{2} \Delta \boldsymbol{\mu}^T \cdot \mathbf{C}^{-1} \cdot \Delta \boldsymbol{\mu} \right\}, \quad (22)$$

where the vector $\Delta \boldsymbol{\mu} = \boldsymbol{\mu}^{\text{obs}} - \boldsymbol{\mu}^{\text{th}}(b, \Omega_{m,0}^\Lambda)$ contains the difference between the observed value and the theoretical prediction of the distance modulus for each measurement in the compilation, while \mathbf{C} denotes the covariance matrix. The theoretical predictions of $\boldsymbol{\mu}$ are computed using a single NN corresponding to $I(z, b, \Omega_{m,0}^\Lambda)$ in combination with the fact that $d_L = I(1+z)/H_0^\Lambda$ and Eq. (21). This last NN is only trained after the training associated with solving Eq. (6) is done. Then, using the resulting trained NNs, Eq. (17) can be defined to now train the NN that represents $I(z, b, \Omega_{m,0}^\Lambda)$.

2. Methodology for the statistical analysis

The likelihood function used in the MCMC process is the product of two separate likelihood functions: the one in Eq. (19) and the one in Eq. (22). We recall that the free parameters of the posterior distribution are b , $\Omega_{m,0}^\Lambda$, H_0^Λ , and M . We use uniform priors for b , $\Omega_{m,0}^\Lambda$, and H_0^Λ ; $b \in (0, 3]$, $\Omega_{m,0}^\Lambda \in [0.05, 0.4]$, and $H_0^\Lambda \in [50, 100]$, respectively for the Hu-Sawicki model. For the Starobinsky model we use the same prior for H_0^Λ , and the flat priors for the other two are $b \in (0, 4]$ and $\Omega_{m,0}^\Lambda \in [0.1, 0.4]$.

For the treatment of the absolute magnitude of SNIa, various approaches exist in the literature. These can be summarized as (i) using a uniform prior and marginalizing M [24,37], (ii) taking M as a free parameter with a uniform prior [14,38], and (iii) taking M as a free parameter and using a Gaussian prior with mean and standard deviation determined from independent data. We choose the (iii) method as developed in [39,40] where the Cepheid data from the SH0ES Collaboration are used to determine the prior parameters of M . In particular, we used a Gaussian distribution centered on $\mu_M = -19.243$ with $\sigma_M = 0.32$. We stress that the methods described in (i) and (ii) assume equal probability of all values of M , which would be reasonable if no further information about this quantity is available. However, as is well known, Cepheid data from the SH0ES Collaboration can be used to extract information about M [39,40]. Therefore, our treatment is more appropriate than the one reported in Ref. [24]. On the other hand, leaving M as a free parameter or performing the marginalization with a uniform prior can bias the estimation of cosmological parameters.¹⁰ In fact, we did perform the

¹⁰We notice that the marginalization proposed in [37] is performed over the variable $\mathcal{M} = M + 5 \log(\frac{c/H_0}{\text{Mpc}})$, which involves the Hubble constant. Therefore, this marginalization is not recommended to estimate H_0 to address the Hubble tension.

inference process considering the approaches (i) and (ii) described before, and we found a significant change in the estimated values of b . On the other hand, it has been shown in Ref. [40] that considering the Gaussian prior in this way yields very similar results to the case when the Pantheon+ and SH0ES data are considered. In this way, our approach allows us to include the information provided by the SH0ES Collaboration that is relevant for the Hubble tension while providing a prior on M that is independent of the dataset considered for the statistical analysis.

IV. RESULTS

In this section, we present and discuss the results of the statistical analysis using the MCMC algorithm with the NN-based solutions. We then compare the time required for the MCMC analysis using this new method with the one required by the numerical method and the method employed in our previous work [14].

A. Results from the statistical analysis

In Figs. 4 and 5, we show the 68% and 95% confidence level contours of the posterior probability distribution of the parameters, using a numerical method and using the NN-based solutions for both $f(R)$ models analyzed in this work. As it can be seen in the figures, the results are fully compatible with each other, showing that the errors shown

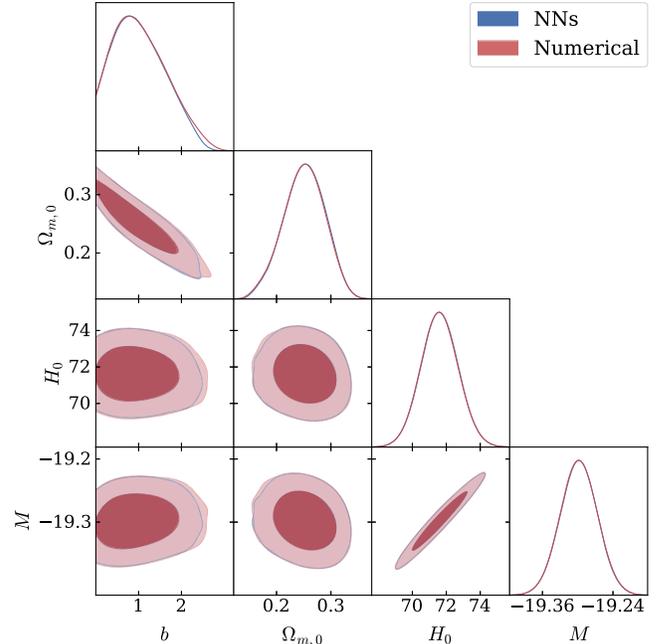


FIG. 4. Results of the statistical analysis for the $f(R)$ Hu-Sawicki model using NNs (in blue) and using a numerical solver (in red). The darker and brighter regions correspond to the 68% and 95% confidence regions, respectively. The plots on the diagonal show the posterior probability density for each of the free parameters of the model.

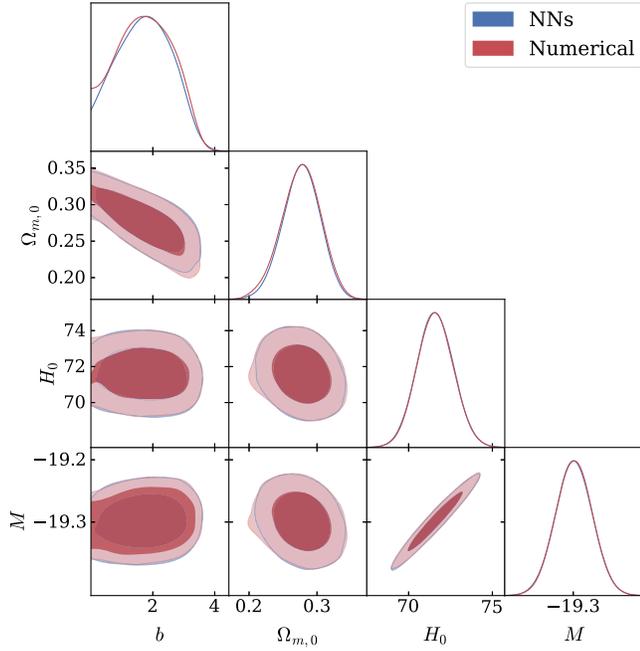


FIG. 5. Results of the statistical analysis for the $f(R)$ Starobinsky model using NNs (in blue) and using a numerical solver (in red). The darker and brighter regions correspond to the 68% and 95% confidence regions, respectively. The plots on the diagonal show the posterior probability density for each of the free parameters of the model.

in Figs. 2 and 3 do not significantly affect the results of the statistical analysis. We also show the resulting 68% and 95% confidence intervals and mean and best fit values for the parameters of the models obtained using NNs in Tables I and II for Hu-Sawicki and Starobinsky models, respectively. We note that the results are shown for $\Omega_{m,0}$ and H_0 which are the parameters of the $f(R)$ model instead of $\Omega_{m,0}^\Lambda$ and H_0^Λ (parameters of the Λ CDM model). These quantities are related by Eq. (10) allowing us to obtain $\Omega_{m,0}$ and H_0 from $\Omega_{m,0}^\Lambda$ and H_0^Λ by a postprocessing. For this, we also need the value of H_0 which is obtained evaluating for each model $H(z=0)$.

The posterior distributions and confidence contours of the cosmological parameters $\Omega_{m,0}$ and H_0 are very similar for both models analyzed in this paper. Therefore, we

offer next the corresponding discussion that applies to both models. We note that the obtained value of $\Omega_{m,0}$ is consistent with the ones inferred considering other cosmological datasets [1,41] and assuming the Λ CDM model. Furthermore, the obtained value of H_0 is consistent with the value estimated by Ref. [2] using data from SNIa and Cepheids from the SH0ES Collaboration. This result is not surprising since we are using the same datasets: SNIa data together with the prior on M that is derived from the SH0ES data (see discussion above). The only difference is that the analysis performed by Ref. [2] is completely model independent, while we are assuming an alternative cosmological model based on modified gravity theories. On the other hand, the obtained confidence interval of H_0 in Ref. [24] [for both Hu-Sawicki and Starobinsky $f(R)$ models], when the SH0ES data are not included in the analysis, is consistent with the one inferred from CMB data [1] and in tension with the one obtained by Ref. [2]. Therefore, we verify here, as discussed elsewhere, that the high values of H_0 are only obtained when the SH0ES data are considered.

Now, we focus on the posterior distribution of the distortion parameter b for the Hu-Sawicki model. We note that the obtained confidence intervals of the b parameter are shifted to larger values with respect to the one obtained in our previous work for the same model and the Pantheon compilation. This is the same behavior as the one obtained with the Starobinsky model in this paper and also in Ref. [24].

Next, we discuss the posterior distribution of the b parameter for the Starobinsky model. We observe significant variations in the confidence intervals for parameter b based on the choice of datasets used for the statistical analysis. Specifically, the confidence interval for b obtained here with the Pantheon+ compilation data differs notably from the values reported in Ref. [25], which used the Pantheon Collaboration data together with other datasets. On the other hand, the obtained confidence interval of b in Ref. [24], where the Pantheon+ compilation data are considered, has also larger values than the ones obtained in Ref. [25], as well as larger values than the ones obtained here. The reason for this last discrepancy can be found in the choice of the prior on M . We recall that a large value of b in the Starobinsky model does not imply a huge departure

TABLE I. Constraints on the parameters of the $f(R)$ Hu-Sawicki cosmological model with $n = 1$. The table shows the 68% and 95% confidence intervals (C.I.) for each free parameter, along with their mean values. The reduced χ^2_ν is also shown for the best fit.

| | b | $\Omega_{m,0}$ | $H_0 \left(\frac{\text{km/s}}{\text{Mpc}} \right)$ | M | χ^2_ν |
|----------|----------------|----------------|---|--------------------|--------------|
| 68% C.I. | [0.255, 1.465] | [0.216, 0.291] | [70.582, 72.662] | [-19.327, -19.267] | |
| 95% C.I. | [0, 2.049] | [0.181, 0.321] | [69.572, 73.682] | [-19.357, -19.239] | |
| Mean | 1.018 | 0.251 | 71.613 | -19.298 | |
| Best fit | 0.296 | 0.306 | 67.883 | -19.416 | 0.876 |

TABLE II. Constraints on the parameters of the $f(R)$ Starobinsky cosmological model with $n = 1$. The table shows the 68% and 95% confidence intervals (C.I.) for each free parameter, along with their mean values. The reduced χ^2_ν is also shown for the best fit.

| | b | $\Omega_{m,0}$ | $H_0 \left(\frac{\text{km/s}}{\text{Mpc}} \right)$ | M | χ^2_ν |
|----------|----------------|----------------|---|--------------------|--------------|
| 68% C.I. | [0.872, 2.706] | [0.252, 0.304] | [70.551, 72.636] | [-19.329, -19.269] | |
| 95% C.I. | [0.135, 3.157] | [0.226, 0.328] | [69.534, 73.63] | [-19.36, -19.24] | |
| Mean | 1.714 | 0.277 | 71.588 | -19.299 | |
| Best fit | 0.753 | 0.312 | 67.842 | -19.417 | 0.876 |

from Λ CDM as is the case for the $f(R)$ Hu-Sawicki model [25].

Our results and the ones of Ref. [24] (for both the Hu-Sawicki and the Starobinsky models), point out that the data from the Pantheon+ compilation are more consistent with departures from the Λ CDM model than previous datasets from the same collaboration. We also noted an absence of degeneration in the H_0 - b plane (in both models) and this behavior is also observed in Ref. [24] where a different choice for the prior on M was made. Therefore, we conclude that the negligible correlation between H_0 and b is not affected by the choice of the prior on M . This also discards the possibility that the larger value of b allowed by the Pantheon+ dataset is rooted in any degeneracy between b and H_0 . Furthermore, we note a degeneration in the Ω_m - b plane in Fig. 5 as was also observed in Ref. [24].

B. Improved computational efficiency

One of the main motivations behind this work is to improve the speedup achieved on inference processes with the NN bundle method in our previous work.

In this section, we show our results regarding the time that it took to complete different MCMC processes, which use the same datasets, depending on the method used to compute the solutions of the given model being tested. These methods include a numerical method (Runge-Kutta), the NN method of our previous work, and the NN method presented here. For the NN method corresponding to our previous paper, we run all MCMC processes on CPU as well as on GPU. On the other hand, we run only on CPU for the method introduced here (see discussion below for more details). For the numerical method,¹¹ we set the error tolerances to $r_{\text{tol}} = 10^{-3}$ and $a_{\text{tol}} = 10^{-6}$, which are both the default values in SciPy's [34] method for solving initial value problems, and also match the order of the error in Figs. 2 and 3 (using the same ground truths as comparison).

¹¹We recall here that we used the following Runge-Kutta methods from the SciPy library: RKF45 for the Hu-Sawicki model and Radau IIA family of order 5 for the Starobinsky model. We verified that using Radau IIA for the Hu-Sawicki model results in an increase of the computational times, while using it for the Starobinsky model implies an important reduction.

We perform these analyses for the Hu-Sawicki and Starobinsky $f(R)$ models.

Table III shows our results for the MCMC algorithm computing times. In each column, we observe that the fastest method is the one presented in this work. Indeed, the new NN method moves the computational cost of both solving Eq. (6) and computing Eq. (16) to the training stage of the NNs. In this way, the computations needed at each step of the MCMC algorithm are just the evaluations of the necessary NNs on only the 1622 data points (cosmic chronometers and SNIa). Also, the fact that these points are not so many means that there is no incentive to leverage the parallelization capabilities of a GPU, due to the CPU being faster. We have corroborated that this is indeed what happens in our case. Nevertheless, this could be subject to change if the amount of data is large enough. This behavior is not the case for the old NN method. In that case, the parallelization that the GPU provides is far more important. The reason for this is the necessity of evaluating the NNs in a large number of points to perform the integration in Eq. (16) (see Sec. III for more details).

One more trend that can be observed from Table III is that the advantage of using either NN method is larger for the Starobinsky model than the Hu-Sawicki model. This is because the Starobinsky model is more complex than the Hu-Sawicki model, which makes it more difficult to solve for the numerical method, and thus the advantage that either NN method provides over the numerical is enhanced.

TABLE III. The computation times of MCMC processes applied to two different $f(R)$ models (Hu-Sawicki and Starobinsky) using the same datasets are compared across various techniques for computing solutions. These techniques include a numerical method (Runge-Kutta), the NN method described in our previous work [14], and the NN method introduced in this study. All CPU computations were performed on an Intel i5-8400 and, for GPU computations, a single NVidia A100 was utilized.

| Method (hardware) | Hu-Sawicki model | Starobinsky model |
|------------------------|------------------|-------------------|
| Numerical method (CPU) | 11 h 51 min | >2 days 17 h |
| Old NN method (CPU) | 17 h 21 min | 2 h 25 min |
| Old NN method (GPU) | 11 h 51 min | 1 h 33 min |
| New NN method (CPU) | 2 h 6 min | 18 min |

This further shows that, regardless of the improvements made here, a major factor on whether the numerical method or either NN method is faster comes down to the difficulty of the problem that is being solved.

On the other hand, let us mention some differences in the settings of the MCMC process of this work with respect to the ones of our previous work that affect the calculation of the respective computational times: (i) in this work, we did not consider the times of postprocessing of the chains as was done in our previous work, (ii) the tolerance of the Runge-Kutta method is different, and (iii) the amount of iterations of the MCMC process also changed. A detailed discussion of all these changes can be found in Appendix B.

There is an important clarification to be made regarding the time that it takes to compute the calculations of the MCMC algorithm using the numerical method for the Starobinsky model in Table III. For that run of the MCMC algorithm we found that the numerical method struggled to compute solutions in certain regions of the parameter space [roughly $(b, \Omega_{m,0}^\Lambda) \in [3, 4] \times [0.1, 0.17]$]. We estimate that this would make the time for the the MCMC algorithm to complete on the order of a month. To mitigate this issue we adjusted the prior on $\Omega_{m,0}^\Lambda$ to be uniform such that $\Omega_{m,0}^\Lambda \in [0.17, 0.4]$, differing from the more natural prior of $\Omega_{m,0}^\Lambda \in [0.1, 0.4]$ we used for the NN method. We saw that this change in the prior did not seem to affect the results given the fact that the posterior distributions did not have a significant number of samples close to the boundary of $\Omega_{m,0}^\Lambda = 0.17$. Nevertheless, because this choice of prior is not a natural one, we argue that the time we obtain is actually a lower bound on one corresponding to a more realistic scenario. We stress that we did not encounter this problem in either of the NN methods.

A relevant discussion to be had is the one corresponding to the time needed for the training of the NNs. The training that corresponds to Eq. (6) was two days, while for Eq. (17) was 5 h 31 min, with both being done on a single NVidia A100 GPU. Even though these numbers seem to negate the advantage suggested by the ones shown in Table III, we echo our arguments from our previous work in Appendix F 4 of Ref. [14]. There we stressed that, because the training of the NNs is done only once, the time taken for training becomes negligible over repeated use of the solutions.

While it is true that, generally, the training of the NNs is done once, it is important to remark on cases where additional training would be needed: (i) a different theoretical model is considered to be tested against the data, (ii) new data that lay outside of the training range of the independent variable are considered, and (iii) the inclusion of new data results in confidence intervals whose range is outside the training range of the NNs. This last example was the case when we tried to use the NNs we trained for the Hu-Sawicki model in our

previous work [14] for the statistical analysis performed in this work. Therefore, we trained new NNs with a wider training range in $\Omega_{m,0}^\Lambda$. These newly trained NNs cover a wide enough range so that we expect that there will be no need to retrain this networks in the future.

V. CONCLUSIONS

In this work, we introduce the application of the NN bundle method to solve integral equations in scenarios where the integrand is computed with solutions of a differential system. We apply this approach to compute the luminosity distance, defined by an integral equation whose integrand is a function of the Hubble parameter which, in turn, is calculated by solving a differential system. By employing this technique, we develop a completely analytical likelihood function for use in an MCMC process aimed at constraining a cosmological model's parameters. This results in a notable reduction in the MCMC algorithm completion time compared to utilizing a numerical method or the NN method employed in our previous work [14].

During our study, we found that, for the Starobinsky $f(R)$ model, certain numerical instabilities reported in Ref. [24] and replicated here are not present when using the NN method. In the numerical approach, these instabilities are typically addressed by modifying the initial conditions in a manner that increases the computational cost of solving the differential equations.

One of the main achievements of this work is the set of trained neural networks of alternative cosmological models that offer an improvement in speed of the MCMC inference process. Once trained, these networks can be used indefinitely with current and future datasets. Furthermore, the idea of computing the luminosity distance with the new NN method can be applied to other theoretical models to reduce the computational times of the inference process. Greater improvements in computational times will be obtained from using the new NN method for models that are computationally intensive to integrate.

On the other hand, this work provides an original analysis of the Hu-Sawicki and Starobinsky $f(R)$ models with recent data from cosmic chronometers and SNIa from the Pantheon+ compilation. The originality of our analysis lies in the appropriate treatment of the absolute magnitude of SNIa, as has been discussed in Sec. III C 2. This choice affects the estimation of the distortion parameter b which measures the departure of the model with respect to Λ CDM (see discussion in Sec. IV).

In summary, we present here a novel way to incorporate the NN bundle method into an inference pipeline, making the likelihood analytical for any model whose observables are described by solutions to differential and integral equations. Consequently, the time required to complete the statistical analysis is significantly reduced.

Finally, further improvements that can be brought into the method to make it an even more appealing option are left for future work. Among them, we can mention:

- (i) to improve the method so that it can provide the error of its own solution (this has been explored in Ref. [42], but the application to the cosmological scenario is missing);
- (ii) optimization of the NN architecture to further improve the speed of the training stage, as well as the inference process; and
- (iii) perform the inference process the Hamiltonian Monte Carlo algorithm [43] instead of the MCMC algorithm (this can be achieved due to the fact that the likelihood now is analytical and, even more important, differentiable).

ACKNOWLEDGMENTS

This collaboration was facilitated by AstroAI at the Center for Astrophysics | Harvard & Smithsonian. The computations performed in this paper, which used GPUs, were run on the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. S. L. is supported by Grant No. PIP 11220200100729CO CONICET and Grant No. 20020170100129BA UBACYT. C. G. S. is supported by Grant No. PIP 11220200102876CO CONICET, and Grant No. G175 from UNLP. C. G. was funded by NASA Contract No. NAS8-03060 to the CXC and by AstroAI.

APPENDIX A: TYPE IA SUPERNOVAE NUISANCE PARAMETERS

The distance modulus can be expressed by a modified version of the formula proposed in Ref. [36],

$$\mu = m_b - M + \alpha x_1 - \beta c + \Delta_M + \Delta_B. \quad (\text{A1})$$

Here α and β are global nuisance parameters that relate stretch (x_1) and color (c), respectively, to luminosity. Additionally, M is the absolute magnitude of a SNIa. Δ_B is a correction term that accounts for selection biases (the procedure is described in detail in Appendix A of Ref. [23]). Finally, Δ_M is the luminosity correction for residual correlations between the standardized brightness of a SNIa and the host galaxy, which depends on the nuisance parameter γ . This way, the corrected overall flux normalization is defined as $\tilde{m}_b = m_b + \alpha x_1 - \beta c + \Delta_M + \Delta_B$. The so-called nuisance parameters (α, β, γ) are usually estimated from the comparison of the observational data with the standard cosmological model (Λ CDM). It has been shown that considering alternative cosmological models does not significantly change the estimation of these parameters [38,44]. Therefore, for the purposes of testing cosmological models, the Pantheon+ compilation also provides a dataset where all nuisance parameters are fixed assuming the Λ CDM model and, in consequence, Eq. (A1) is reduced to Eq. (20).

APPENDIX B: COMPUTING TIME DETAILS

The main goal of our previous work [14] was to test the viability and advantages of the NN bundle method in the cosmological context. As part of that analysis, we reported the computing times of the inference process with a similar procedure to the one in Sec. IV B. Nevertheless, given that the main goal of the work presented here is the advantage that the NNs provide in regard to the computing times, we have refined our methodology for the comparison between the numerical method and the NN method. For this, we have introduced two changes in the settings of the MCMC process.

First, we have increased the value of the error tolerances of the Runge-Kutta method in this paper to $r_{\text{tol}} = 10^{-3}$ and $a_{\text{tol}} = 10^{-6}$ from $r_{\text{tol}} = 10^{-6}$ and $a_{\text{tol}} = 10^{-9}$ of our previous work. This is more fair to the numerical method because it makes its precision comparable to the NN method while making the method faster. It is also important to recall that the tolerances used here are the default ones in SciPy's [34] method for solving initial value problems.

The other change in methodology with respect to our previous work is that we no longer take into account the time needed to do the postprocessing of the chains required to obtain H_0 and $\Omega_{m,0}$ (for more details, see Sec. II). While this computation is unavoidable, it is theoretically possible to optimize the MCMC algorithm so that it is no longer necessary to integrate the equations again to do the postprocessing. Because of this fact, it was unfair for the numerical method to include the postprocessing time because it meant to integrate again for each chain in order to obtain H_0 , while for the NNs it just entailed evaluating them at $z = 0$.

One last note on the difference between the times shown in Table III and the ones shown in our previous work is the difference in the amount of iterations of the MCMC algorithm. Because we changed the data and priors for the Hu-Sawicki model (new cosmic chronometer data, updated Pantheon+ sample, the Gaussian prior for M , and the bounds on the prior of $\Omega_{m,0}^\Lambda$), the amount of iterations needed for the MCMC algorithm changed from our previous work. For example, the amount of iterations needed for the MCMC algorithm to converge for the Hu-Sawicki model in our previous work was 22000, while the one corresponding for this work was 206500. With regard to the Starobinsky model, the iterations needed were 28400. This is why the time it took for the MCMC algorithm to complete on the Hu-Sawicki model on the old NN method in Table III does not match the one in our previous paper, even though the NN method is the same. Also, this difference between the amount of iterations explains why the NN method took more time for the Hu-Sawicki model in Table III than for the Starobinsky model. Otherwise, the process should take the same amount of time, because in both cases the computations are the same (i.e., evaluating three NNs on the same data points in each iteration of the MCMC algorithm).

- [1] N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, S. Basak *et al.* (Planck Collaboration), Planck 2018 results—VI. Cosmological parameters, *Astron. Astrophys.* **641**, A6 (2020).
- [2] A. G. Riess, W. Yuan, L. M. Macri, D. Scolnic, D. Brout, S. Casertano, D. O. Jones, Y. Murakami, G. S. Anand, L. Breuval *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**, L7 (2022).
- [3] G.-J. Wang, X.-J. Ma, and J.-Q. Xia, Machine learning the cosmic curvature in a model-independent way, *Mon. Not. R. Astron. Soc.* **501**, 5714 (2021).
- [4] C. Escamilla-Rivera, M. A. C. Quintero, and S. Capozziello, A deep learning approach to cosmological dark energy models, *J. Cosmol. Astropart. Phys.* **03** (2020) 008.
- [5] K. Dialektopoulos, J. L. Said, J. Mifsud, J. Sultana, and K. Z. Adami, Neural network reconstruction of late-time cosmology and null tests, *J. Cosmol. Astropart. Phys.* **02** (2022) 023.
- [6] G.-J. Wang, X.-J. Ma, S.-Y. Li, and J.-Q. Xia, Reconstructing functions and estimating parameters with artificial neural networks: A test with a hubble parameter and SNe Ia, *Astrophys. J. Suppl. Ser.* **246**, 13 (2020).
- [7] A. Moss, Accelerated Bayesian inference using deep learning, *Mon. Not. R. Astron. Soc.* **496**, 328 (2020).
- [8] C.-H. To, E. Roza, E. Krause, H.-Y. Wu, R. H. Wechsler, and A. N. Salcedo, LINNA: Likelihood inference neural network accelerator, *J. Cosmol. Astropart. Phys.* **01** (2023) 016.
- [9] A. Spurio Mancini, D. Piras, J. Alsing, B. Joachimi, and M. P. Hobson, CosmoPower: Emulating cosmological power spectra for accelerated Bayesian inference from next-generation surveys, *Mon. Not. R. Astron. Soc.* **511**, 1771 (2022).
- [10] A. Nygaard, E. B. Holm, S. Hannestad, and T. Tram, CONNECT: A neural network based framework for emulating cosmological observables and cosmological parameter inference, *J. Cosmol. Astropart. Phys.* **05** (2023) 025.
- [11] I. Lagaris, A. Likas, and D. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Networks* **9**, 987 (1998).
- [12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [13] C. Flamant, P. Protopapas, and D. Sondak, Solving differential equations using neural network solution bundles, [arXiv:2006.14372](https://arxiv.org/abs/2006.14372).
- [14] A. T. Chantada, S. J. Landau, P. Protopapas, C. G. Scóccola, and C. Garraffo, Cosmology-informed neural networks to solve the background dynamics of the Universe, *Phys. Rev. D* **107**, 063523 (2023).
- [15] J. Simon, L. Verde, and R. Jimenez, Constraints on the redshift dependence of the dark energy potential, *Phys. Rev. D* **71**, 123001 (2005).
- [16] D. Stern, R. Jimenez, L. Verde, M. Kamionkowski, and S. A. Stanford, Cosmic chronometers: Constraining the equation of state of dark energy. I: $H(z)$ measurements, *J. Cosmol. Astropart. Phys.* **02** (2010) 008.
- [17] M. Moresco, A. Cimatti, R. Jimenez, L. Pozzetti, G. Zamorani, M. Bolzonella, J. Dunlop, F. Lamareille, M. Mignoli, H. Pearce *et al.*, Improved constraints on the expansion rate of the Universe up to $z \sim 1.1$ from the spectroscopic evolution of cosmic chronometers, *J. Cosmol. Astropart. Phys.* **08** (2012) 006.
- [18] C. Zhang, H. Zhang, S. Yuan, S. Liu, T.-J. Zhang, and Y.-C. 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).
- [19] M. 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).
- [20] M. Moresco, L. Pozzetti, A. Cimatti, R. Jimenez, C. Maraston, L. Verde, D. Thomas, A. Citro, R. Tojeiro, and D. 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.
- [21] 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).
- [22] N. Borghi, M. Moresco, and A. Cimatti, Toward a better understanding of cosmic chronometers: A new measurement of $H(z)$ at $z \sim 0.7$, *Astrophys. J. Lett.* **928**, L4 (2022).
- [23] D. Scolnic, D. Brout, A. Carr, A. G. Riess, T. M. Davis, A. Dwomoh, D. O. Jones, N. Ali, P. Charvu, R. Chen *et al.*, The Pantheon+ analysis: The full data set and light-curve release, *Astrophys. J.* **938**, 113 (2022).
- [24] K. Ravi, A. Chatterjee, B. Jana, and A. Bandyopadhyay, Investigating the accelerated expansion of the Universe through updated constraints on viable $f(R)$ models within the metric formalism, *Mon. Not. R. Astron. Soc.* **527**, 7626 (2023).
- [25] C. R. Farrugia, J. Sultana, and J. Mifsud, Spatial curvature in $f(R)$ gravity, *Phys. Rev. D* **104**, 123503 (2021).
- [26] W. Hu and I. Sawicki, Models of $f(R)$ cosmic acceleration that evade solar system tests, *Phys. Rev. D* **76**, 064004 (2007).
- [27] A. A. Starobinsky, Disappearing cosmological constant in $f(R)$ gravity, *JETP Lett.* **86**, 157 (2007).
- [28] S. Kandhai and P. K. S. Dunsby, Cosmological dynamics of viable $f(R)$ theories of gravity, [arXiv:1511.00101](https://arxiv.org/abs/1511.00101).
- [29] S. Basilakos, S. Nesseris, and L. Perivolaropoulos, Observational constraints on viable $f(R)$ parametrizations with geometrical and dynamical probes, *Phys. Rev. D* **87**, 123529 (2013).
- [30] S. Lloyd, R. A. Irani, and M. Ahmadi, Using neural networks for fast numerical integration and optimization, *IEEE Access* **8**, 84519 (2020).
- [31] D. B. Lindell, J. N. P. Martel, and G. Wetzstein, AutoInt: Automatic integration for fast neural volume rendering, in *2021 IEEE/CVF Conference on Computer Vision and*

- Pattern Recognition (CVPR)* (2021), pp. 14551–14560, [10.1109/CVPR46437.2021.01432](https://doi.org/10.1109/CVPR46437.2021.01432).
- [32] G. Teichert, A. Natarajan, A. Van der Ven, and K. Garikipati, Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions, *Comput. Methods Appl. Mech. Eng.* **353**, 201 (2019).
- [33] D. Maître and R. Santos-Mateos, Multi-variable integration with a neural network, *J. High Energy Phys.* **03** (2023) 221.
- [34] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in python, *Nat. Methods* **17**, 261 (2020).
- [35] S. D. Odintsov, D. Sáez-Chillón Gómez, and G. S. Sharov, Is exponential gravity a viable description for the whole cosmological history?, *Eur. Phys. J. C* **77**, 862 (2017).
- [36] R. Tripp, A two-parameter luminosity correction for type Ia supernovae, *Astron. Astrophys.* **331**, 815 (1998).
- [37] A. Conley, J. Guy, M. Sullivan, N. Regnault, P. Astier, C. Balland, S. Basa, R. G. Carlberg, D. Fouchez, D. Hardin *et al.*, Supernova constraints and systematic uncertainties from the first three years of the supernova legacy survey, *Astrophys. J. Suppl. Ser.* **192**, 1 (2010).
- [38] M. Leizerovich, L. Kraiselburd, S. Landau, and C. G. Scóccola, Testing $f(R)$ gravity models with quasar x-ray and UV fluxes, *Phys. Rev. D* **105**, 103526 (2022).
- [39] D. Camarena and V. Marra, Local determination of the Hubble constant and the deceleration parameter, *Phys. Rev. Res.* **2**, 013028 (2020).
- [40] D. Camarena and V. Marra, The tension in the absolute magnitude of type Ia supernovae, [arXiv:2307.02434](https://arxiv.org/abs/2307.02434).
- [41] T. M. C. Abbott, M. Aguena, A. Alarcon, S. Allam, O. Alves, A. Amon, F. Andrade-Oliveira, J. Annis, S. Avila, D. Bacon *et al.* (DES Collaboration), Dark energy survey year 3 results: Cosmological constraints from galaxy clustering and weak lensing, *Phys. Rev. D* **105**, 023520 (2022).
- [42] O. Graf, P. Flores, P. Protopapas, and K. Pichara, Error-aware B-PINNs: Improving uncertainty quantification in Bayesian physics-informed neural networks, [arXiv:2212.06965](https://arxiv.org/abs/2212.06965).
- [43] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, *Phys. Lett. B* **195**, 216 (1987).
- [44] C. Negrelli, L. Kraiselburd, S. Landau, and C. G. Scóccola, Testing modified gravity theory (MOG) with type Ia supernovae, cosmic chronometers and baryon acoustic oscillations, *J. Cosmol. Astropart. Phys.* **07** (2020) 015.