Cosmology-informed neural networks to solve the background dynamics of the Universe

Augusto T. Chantada[1,*] Susana J. Landau[1,2] Pavlos Protopapas[3] Claudia G. Scóccola[4,5] and Cecilia Garraffo[6,7]

[1]*Departamento de Física, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires,*
Avenida Intendente Cantilo S/N, 1428, Ciudad Autónoma de Buenos Aires, Argentina
[2]*IFIBA—CONICET—UBA, 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 & Smithsonian, 60 Garden Street,*
Cambridge, Massachusetts 02138, USA
[7]*Institute for Applied Computational Science, Harvard University,*
33 Oxford Street, Cambridge, Massachusetts 02138, USA

The field of machine learning has drawn increasing interest from various other fields due to the success of its methods at solving a plethora of different problems. An application of these has been to train artificial neural networks to solve differential equations without the need of a numerical solver. This particular application offers an alternative to conventional numerical methods, with advantages such as lower memory required to store solutions, parallelization, and, in some cases, a lower overall computational cost than its numerical counterparts. In this work, we train artificial neural networks to represent a bundle of solutions of the differential equations that govern the background dynamics of the Universe for four different models. The models we have chosen are ΛCDM, the Chevallier-Polarski-Linder parametric dark energy model, a quintessence model with an exponential potential, and the Hu-Sawicki $f(R)$ model. We use the solutions that the networks provide to perform statistical analyses to estimate the values of each model's parameters with observational data; namely, estimates of the Hubble parameter from cosmic chronometers, type Ia supernovae data from the Pantheon compilation, and measurements from baryon acousstic oscillations. The results we obtain for all models match similar estimations done in the literature using numerical solvers. In addition, we estimate the error of the solutions that the trained networks provide by comparing them with the analytical solution when there is one, or to a high-precision numerical solution when there is not. Through these estimations we find that the error of the solutions is at most $\sim$1% in the region of the parameter space that concerns the 95% confidence regions that we find using the data, for all models and all statistical analyses performed in this work. Some of these results are made possible by improvements to the method of solving differential equations with artificial neural networks conceived in this work.

I. INTRODUCTION

To improve our understanding of the Universe, new models are constantly being proposed to better describe its different aspects. All models need to be continuously tested against observational data to determine which models best describe nature. Besides, often the theoretical predictions are calculated from a system of differential equations that need to be solved first to test the model's predictions against the data. The usual approach to solving these equations is to use numerical methods, which usually provide the solution in the form of an array of points matching a discretization of the domain. In addition, any statistical analysis which aims to constrain the parameters of a model requires exploring the parameter space of such a model. For this, the numerical solver needs to integrate each time the parameters of the differential system are changed by any amount, which, in sufficiently complex

---

*augustochantada01@gmail.com

systems, can create a bottleneck in the computation time required to make the analysis statistically robust. The past few years have seen an increased interest in the use of artificial neural networks (ANNs) as new tools for analyzing both data and physical models. For example, the so called physics-informed neural networks are able to interpolate the physical behavior of a given data set, provided the underlying physical laws are assumed [1]. Some approaches in the field of cosmology include using ANNs to reconstruct cosmological data, like in Ref. [2] where this was done for both cosmic expansion and large-scale structure data, and creating artificial data extending the range of current data sets as in Ref. [3]. Additionally, ANNs have been used as a replacement for numerical methods to serve as solutions to physical models. For example, in Ref. [4] ANNs were trained through supervised learning on the outputs of a numerical method to constrain the Lambda cold dark matter (ΛCDM) model with cosmic microwave background (CMB) and large-scale structure data. Also, ANNs trained through unsupervised learning were used to compute the tunneling profiles of cosmological phase transitions in Ref. [5] with a modified version of the method introduced in Ref. [6]. Contrary to the numerical method, the solutions provided by ANNs are continuous and fully differentiable [7].[1] Furthermore, typically the storage for ANNs is tiny, and especially compared to numerical methods. In this way, scientists and engineers can share the solutions used in their research with a wider community, easing the process of reproducibility.

An extension of the unsupervised method for solving differential equations with ANNs was proposed in Ref. [8], which introduces the possibility of training ANNs whose outcome is a bundle of solutions corresponding to a continuous range for the values of the parameters of the differential system, which can include initial and boundary conditions. The main advantage of this method can be summarized as follows: the integration required to obtain the solutions is performed once, during training, after which the solutions can be used indefinitely to perform parameter inference, requiring in some cases (mainly those where the differential system is not easy to integrate) less computational time than with traditional numerical solvers. In this work, we adapt the bundle ANN method introduced in Ref. [8], to obtain solutions of physical models in a cosmological context and perform statistical analyses to obtain constraints on the model's parameters. For this we implement the NeuroDiffEq [9] Python library [10] to solve the background dynamics of four cosmological models. Also, to adapt the method to the cosmological background equations, some improvements to the method are developed.

The physical models considered in this paper are part of an active area of research in cosmology, which constitutes

modeling the late-time accelerated expansion of the Universe as first evidenced by measurements of type Ia supernovae [11,12]. Currently, the favored model to explain this behavior is the ΛCDM model, where a positive cosmological constant Λ is introduced to the Einstein-Hilbert action of general relativity to reproduce the late-time acceleration in the expansion of the Universe. Although this model represents the simplest solution, it has some serious issues, like the lack of a theoretical explanation for the observed value of Λ. Therefore, the physical mechanism that drives the accelerated expansion is still unknown. Another problem is the so-called Hubble tension. This tension concerns the apparent difference in the current value of the Hubble parameter H_0 between the estimate obtained with model-independent observations and those inferred from indirect measurements which therefore also require the assumption of a cosmological model. The most prominent example of this is the 5σ tension between the value reported by direct measurements of the SH0ES Collaboration [13] and the one inferred from CMB data by the *Planck* Collaboration [14]. While this tension could be attributed to ΛCDM not being the correct cosmological model, there is still the possibility that the tension arises from systematics in the direct measurements. For recent reviews on this topic, we refer the reader to Refs. [15,16].

A myriad of alternative models have been considered to explain the current accelerated expansion of the Universe. Such models can fall into one of three categories. A first alternative is to add a fluid to the energy-momentum tensor with an equation of state that can eventually change with time. These kinds of models are referred to in the literature as parametric dark energy models. The second one is to, instead, include a new field that is minimally coupled to gravity or matter in the energy-momentum tensor. The simplest examples of such models are those called quintessence models, where the new field is a self-interacting scalar field. Last, another choice is to consider alternative theories to general relativity to describe the gravitational interaction, such as for example $f(R)$ theories, where a modification to the Einstein-Hilbert action in the form of a function of the Ricci scalar R is introduced. In this work, we focus on one model from each of these categories in addition to ΛCDM and compute the expansion rate of the Universe using ANNs instead of numerical solvers. We consider (i) the Chevallier-Polarski-Linder (CPL) parametric dark energy model [17,18], (ii) a quintessence model with an exponential potential [19,20], and (iii) the Hu-Sawicki $f(R)$ model [21]. For both the ΛCDM and CPL models, the solutions to the differential equations can be found analytically, while the differential equations of quintessence and a cosmological model in $f(R)$ gravity do not have analytical solutions. Then, to conclude, we use the trained ANNs together with observational data from cosmic chronometers (CC) [22–27], type Ia supernovae

---

[1]A discussion concerning this affirmation can be found in Appendix C.

(SNIa) [28] and baryon acoustic oscillations (BAO) [29–37] to do Bayesian inference on the parameters of all of these models.

We start by explaining the details of each model in Sec. II, underlining the similarities between them. In addition, we describe the variables that we use in each model to have dimensionless quantities as the dependent variables of the differential equations. Next, in Sec. III, we explain in further detail the ANN method and how it can be applied to an inference pipeline. We also describe the improvements made to the bundle method to implement it in the cosmological scenario. In Sec. IV, we explain how we apply the ANN method to the cosmological models, and showcase the accuracy of our cosmology-informed neural networks in the parameter space of each model. Then, in Sec. V, we give a description of the different data sets we use, the physical quantities that are derived from them, and their corresponding likelihoods, which are used to test the models' predictions in the statistical analysis. Later on, in Sec. VI, we present the inferred constraints for each model. In Sec. VII we analyze how the neural network method compares to a commonly used numerical method regarding computational efficiency. Finally, we share our conclusions and give a summary of our work in Sec. VIII.

The codes used to train the ANNs in this work can be found in the following repository: https://github.com/at-chantada/cosmo-nets.

II. THEORETICAL MODELS

In this section, we explain the theoretical background of the models tested in this work. While each of the models has distinct characteristics, they still share some common underlying assumptions.

First, let us recall the cosmological principle, according to which the geometry of the spacial part of space-time is homogeneous and isotropic. Such space-time can be described by the Friedmann-Lemaître-Robertson-Walker (FLRW) metric. In reduced circumference polar coordinates, this metric sets the line element as[2]

$$ds^2 = -dt^2 + a(t)^2 \left[\frac{dr^2}{1 - kr^2} + r^2(d\theta^2 + \sin^2\theta d\phi^2) \right]. \quad (1)$$

In this work, we use $k = 0$, representing a spatially flat universe. We recall the definition of the "Hubble parameter," $H = \dot{a}/a$, where the dot denotes the derivative with respect to the cosmological time and $a(t)$ is the scale factor introduced in Eq. (1).

Second, it is usually assumed that the Universe is comprised of a perfect fluid and therefore the energy-momentum tensor is $T^\mu{}_\nu = \text{diag}(-\rho, p, p, p)$, where ρ

---

[2]Throughout this paper we use $c = 1$.

denotes the energy density and p is the pressure of the fluid. Moreover, due to the cosmological principle, both ρ and p are independent of the position. Then, the continuity equation (which in turn follows from the conservation of the energy-momentum tensor) reads

$$\dot{\rho} + 3H(\rho + p) = 0. \quad (2)$$

Every model considered in this paper includes nonrelativistic matter that is assumed to be mostly dust. From this, and Eq. (2), an analytical solution for the nonrelativistic matter density ρ_m as a function of the redshift, $z = \frac{1}{a} - 1$, can be found,

$$\rho_m(z) = \rho_{m,0}(1 + z)^3, \quad (3)$$

where $\rho_{m,0}$ denotes the matter density at the present time. Last, because our analysis is done for values of redshift $z \in [0, 10]$, the effects of radiation can be ignored.

A. ΛCDM

In this model, the background cosmological evolution considering an energy-momentum tensor with only nonrelativistic matter, is described by the Friedmann equations as follows:

$$H^2 = \frac{\kappa \rho_m + \Lambda}{3}, \quad (4a)$$

$$2\dot{H} + 3H^2 = \Lambda, \quad (4b)$$

where Λ is the cosmological constant, $\kappa = 8\pi G$ and G is Newton's gravitational constant. Next, H_0 is defined as the value of $H(z)$ at $z = 0$, and $\Omega_{m,0} = \kappa \rho_{m,0}/3H_0^2$. Therefore, using Eq. (3) in Eq. (4a) we get

$$H(z) = H_0 \sqrt{\Omega_{m,0}(1 + z)^3 + \frac{\Lambda}{3H_0^2}}, \quad (5)$$

which can be evaluated at $z = 0$ to obtain $\Lambda = 3H_0^2(1 - \Omega_{m,0})$. With this result, we can express $H(z)$ as

$$H(z) = H_0 \sqrt{\Omega_{m,0}(1 + z)^3 + 1 - \Omega_{m,0}}. \quad (6)$$

Given the fact that the only differential equation that needs to be solved in this model is Eq. (2) for ρ_m with $p_m = 0$, we define the variable $x = \kappa \rho_m/3H_0^2$ to get the following differential system:

$$\frac{dx}{dz} = \frac{3x}{1+z}, \qquad x(z)|_{z=0} = \frac{\kappa\rho_{m,0}}{3H_0^2} = \Omega_{m,0}. \quad (7)$$

The Hubble parameter obtained from the solution of Eq. (7) is

$$H(z) = H_0\sqrt{x(z) + 1 - \Omega_{m,0}}. \quad (8)$$

B. Parametric dark energy

As described in the Introduction, there are both theoretical and observational reasons to consider alternative cosmological models. A simple alternative to ΛCDM, introduced phenomenologically, is to consider a new component of the energy-momentum tensor, whose equation of state is that of a fluid and a function of redshift. For this model, the equations that describe the expansion of the Universe replace Eqs. (4a) and (4b) to become

$$H^2 = \frac{\kappa}{3}(\rho_m + \rho_{DE}), \quad (9a)$$

$$2\dot{H} + 3H^2 = -\kappa p_{DE}, \quad (9b)$$

where ρ_{DE} and p_{DE} represent the density and pressure of the new fluid component, respectively. This new fluid is often referred to as "dark energy" and it is characterized by the following equation of state:

$$p_{DE} = \omega(z)\rho_{DE}, \quad (10)$$

where $\omega(z)$ is the model's parametrization of the equation of state. In this work, we use the CPL parametric dark energy model introduced in Refs. [17,18], with the equation of state

$$\omega(z) = \omega_0 + \frac{\omega_1 z}{1+z}, \quad (11)$$

where ω_0 and ω_1 are free parameters. Applying the CPL parametrization in Eq. (10) and using Eq. (2), the differential equation of the model is

$$\frac{d\rho_{DE}}{dz} = \frac{3\rho_{DE}}{1+z}\left(1 + \omega_0 + \frac{\omega_1 z}{1+z}\right). \quad (12)$$

This differential equation has the analytical solution

$$\rho_{DE}(z) = \rho_{DE,0}(1+z)^{3(1+\omega_0+\omega_1)}\exp\left(-\frac{3\omega_1 z}{1+z}\right), \quad (13)$$

where $\rho_{DE,0}$ is the value of the dark energy density at $z = 0$. Using Eqs. (3) and (13) in Eq. (9a), along with the definition of H_0 and $\Omega_{m,0}$, the expression for the Hubble parameter is

$$H(z) = H_0\sqrt{\Omega_{m,0}(1+z)^3 + (1-\Omega_{m,0})(1+z)^{3(1+\omega_0+\omega_1)}\exp\left(-\frac{3\omega_1 z}{1+z}\right)}. \quad (14)$$

Doing an analogous change of variable as the one done in ΛCDM, we define $x = \kappa\rho_{DE}/3H_0^2$, and highlighting the fact that $\kappa\rho_{DE,0}/3H_0^2 = 1 - \Omega_{m,0}$, the differential equation and initial condition of the problem can be expressed as

$$\frac{dx}{dz} = \frac{3x}{1+z}\left(1 + \omega_0 + \frac{\omega_1 z}{1+z}\right),$$

$$x(z)|_{z=0} = \frac{\kappa\rho_{DE,0}}{3H_0^2} = 1 - \Omega_{m,0}. \quad (15)$$

Considering this change of variable, the expression of the Hubble parameter is

$$H(z) = H_0\sqrt{\Omega_{m,0}(1+z)^3 + x(z)}. \quad (16)$$

C. Quintessence

An alternative proposal to explain the present accelerated expansion of the Universe, is to consider a scalar field ϕ minimally coupled to gravity with an appropriate potential $V(\phi)$. These kinds of cosmological models assume general relativity as the underlying gravitational theory and are usually referred to in the literature as quintessence models [38–40]. Let us recall that the energy density and pressure of a homogeneous scalar field in a flat FLRW background are given by $\rho_\phi = \dot{\phi}^2/2 + V(\phi)$ and $p_\phi = \dot{\phi}^2/2 - V(\phi)$. Besides, the expansion of the Universe can be described by the following Friedmann equations:

$$H^2 = \frac{\kappa}{3}(\rho_m + \dot{\phi}^2/2 + V(\phi)), \quad (17a)$$

$$2\dot{H} + 3H^2 = -\kappa(\dot{\phi}^2/2 - V(\phi)). \quad (17b)$$

In addition, applying Eq. (2) to ρ_ϕ and p_ϕ, the equation of motion of the scalar field can be obtained:

$$\ddot{\phi} + 3H\dot{\phi} + \frac{dV}{d\phi} = 0. \quad (18)$$

The quintessence model considered in this work corresponds to a model characterized by an exponential potential $V(\phi) = V_0 \exp\left(-\lambda\sqrt{\kappa}\phi\right)$ as proposed in Refs. [19,20], where λ is a free parameter, and V_0 is set by the initial conditions. Equations (17) and (18), along with Eq. (2) applied to the matter density, provide the necessary equations to solve the dynamics of the problem given the initial conditions.

In order to solve Eqs. (17a) and (17b), we use the change of variables introduced in Ref. [41] such that

$$x = \frac{\sqrt{\kappa}\dot{\phi}}{\sqrt{6}H}, \qquad y = \frac{\sqrt{\kappa}V(\phi)}{\sqrt{3}H}. \tag{19}$$

Using these new variables along with Eqs. (17) and (18), plus defining $N = -\ln(1 + z)$, the differential equations to solve for x and y are

$$\begin{cases} \frac{dx}{dN} = -3x + \frac{\sqrt{6}}{2}\lambda y^2 + \frac{3}{2}x(1 + x^2 - y^2), \\ \frac{dy}{dN} = -\frac{\sqrt{6}}{2}xy\lambda + \frac{3}{2}y(1 + x^2 - y^2). \end{cases} \tag{20}$$

To set the initial conditions of x and y, we assume that $H(z)$ should match the corresponding ΛCDM solution at $z = z_0$. In this work, we choose $z_0 = 10$ and set N_0 accordingly. In this way, we can write

$$x_0 = 0, \qquad y_0 = \sqrt{\frac{1 - \Omega_{m,0}^\Lambda}{\Omega_{m,0}^\Lambda(1 + z_0)^3 + 1 - \Omega_{m,0}^\Lambda}}, \tag{21}$$

where the superscript Λ refers to the quantities defined in the ΛCDM model. Therefore, $\Omega_{m,0}^\Lambda$ is the matter density parameter in the ΛCDM model which should be distinguished from $\Omega_{m,0}$ the matter density parameter defined in the quintessence model. Both quantities are related as follows:

$$\Omega_{m,0}H_0^2 = \Omega_{m,0}^\Lambda(H_0^\Lambda)^2. \tag{22}$$

Once the solutions for x and y for the system of equations (20) with initial conditions described by Eq. (21) are obtained, the Hubble parameter is computed from the following expression:

$$H = H_0^\Lambda \sqrt{\frac{\Omega_{m,0}^\Lambda(1 + z)^3}{1 - x^2 - y^2}}. \tag{23}$$

Here the Hubble factor that describes the expansion of the Universe in the quintessence model is expressed in terms of the Hubble constant and matter density of the ΛCDM model. However, we will report the results of the parameter inference (see Sec. VI) in terms of the parameters defined in the quintessence model (H_0 and $\Omega_{m,0}$).

D. $f(R)$ gravity

Another possibility to explain the physical mechanism responsible for the accelerated expansion of the Universe is to assume an alternative theory to general relativity to describe the gravitational interaction [42]. In this paper, we focus on a particular class of theories [referred to in the literature as $f(R)$ theories], namely, those in which the action is written as [43]

$$S = \frac{1}{2\kappa}\int d^4x\sqrt{-g}f(R) + S_m. \tag{24}$$

Here R is the Ricci scalar and $f(R)$ is an arbitrary function of the latter. Applying the modified Einstein equations derived from Eq. (24), along with the assumptions listed at the start of this section, results in the modified Friedmann equations

$$H^2 = \frac{1}{3f_R}\left[\kappa\rho_m + \frac{Rf_R - f}{2} - 3H\dot{R}f_{RR}\right], \tag{25a}$$

$$\begin{aligned} 2\dot{H} + 3H^2 = \frac{-1}{f_R}\Bigg[&-\frac{Rf_R - f}{2} + f_{RRR}\dot{R}^2 \\ &+ (\ddot{R} + 2H\dot{R})f_{RR}\Bigg], \end{aligned} \tag{25b}$$

where the notation $f_R = df/dR$, $f_{RR} = d^2f/dR^2$, etc., is used.

It has been shown that only $f(R)$ models that show a "chameleon-like" effect are able to explain observational bounds from Solar System and equivalence principle tests [44,45]. For this reason, we focus on one of the few $f(R)$ models that have not been ruled out by the aforementioned local gravity tests, namely, the Hu-Sawicki model [21] where[3]

$$f(R) = R - 2\Lambda\left[1 + \frac{1}{\frac{R}{\Lambda b} + 1}\right]. \tag{26}$$

As Eq. (26) shows, when $b \to 0$, the usual general relativity with the cosmological constant is recovered. Therefore, b quantifies the deviation from general relativity.

To finally set the parameters of the problem, we express the cosmological constant in terms of the parameters in ΛCDM as $\Lambda = 3(H_0^\Lambda)^2(1 - \Omega_{m,0}^\Lambda)$. In this way, the parameters of the problem are $(b, \Omega_{m,0}^\Lambda, H_0^\Lambda)$. However, as explained for the quintessence model, results for the parameter inference will be shown in terms of the parameters $\Omega_{m,0}$ and H_0 defined in the $f(R)$ models and related to the ΛCDM parameters through Eq. (22).

---

[3]In the Hu-Sawicki model, usually $f(R) = R - \frac{m^2 c_1 (R/m^2)^n}{c_2(R/m^2)^n + 1}$. In this paper, we focus on the case $n = 1$ and we write $f(R)$ as a function of $b = 2/c_1$ while c_2 and m^2 are fixed such that in the high-curvature regime, the ΛCDM behavior is reproduced.

To solve the modified Friedmann equations, we use the variables introduced in Ref. [46]

$$x = \frac{\dot{R} f_{RR}}{H f_R}, \qquad y = \frac{f}{6H^2 f_R},$$

$$v = \frac{R}{6H^2}, \qquad \Omega = \frac{\kappa \rho_m}{3H^2 f_R}. \qquad (27)$$

Using these variables, Eq. (25a) gives the relation

$$1 = \Omega + v - x - y. \qquad (28)$$

The differential system that follows from the derivatives of the variables in Eq. (27) with respect to z, has the problem of including a function $\Gamma = f_R / R f_{RR}$. For some simple $f(R)$ models, it is possible to express Γ in terms of v and y, but in the case of the Hu-Sawicki model this is not the case. A way to solve this problem is to add an extra variable $r = R/\Lambda$ to Eq. (27) to be able to write Γ in terms of it. With this new addition, the differential system is

$$\begin{cases} \frac{dx}{dz} = \frac{1}{1+z}(-\Omega - 2v + x + 4y + xv + x^2), \\ \frac{dy}{dz} = \frac{-1}{1+z}(vx\Gamma - xy + 4y - 2yv), \\ \frac{dv}{dz} = \frac{-v}{1+z}(x\Gamma + 4 - 2v), \\ \frac{d\Omega}{dz} = \frac{\Omega}{1+z}(-1 + 2v + x), \\ \frac{dr}{dz} = -\frac{r\Gamma x}{1+z}, \end{cases} \qquad (29)$$

where the Γ corresponding to the $f(R)$ in Eq. (26) can be expressed as follows:

$$\Gamma(r) = \frac{(r+b)[(r+b)^2 - 2b]}{4br}. \qquad (30)$$

Details of the calculation of the differential equation for r can be found in Appendix B. As we did for the quintessence model, we choose the initial conditions so that the behavior of the model matches that of the ΛCDM model at $z = z_0$. Furthermore, the relation between $H(z)$ and the Ricci scalar is given by $R = 6(2H^2 + \dot{H})$, and therefore the initial conditions of Eq. (29) can be written as

$$x_0 = 0, \qquad (31a)$$

$$y_0 = \frac{\Omega_{m,0}^\Lambda (1+z_0)^3 + 2(1 - \Omega_{m,0}^\Lambda)}{2[\Omega_{m,0}^\Lambda (1+z_0)^3 + 1 - \Omega_{m,0}^\Lambda]}, \qquad (31b)$$

$$v_0 = \frac{\Omega_{m,0}^\Lambda (1+z_0)^3 + 4(1 - \Omega_{m,0}^\Lambda)}{2[\Omega_{m,0}^\Lambda (1+z_0)^3 + 1 - \Omega_{m,0}^\Lambda]}, \qquad (31c)$$

$$\Omega_0 = \frac{\Omega_{m,0}^\Lambda (1+z_0)^3}{\Omega_{m,0}^\Lambda (1+z_0)^3 + 1 - \Omega_{m,0}^\Lambda}, \qquad (31d)$$

$$r_0 = \frac{\Omega_{m,0}^\Lambda (1+z_0)^3 + 4(1 - \Omega_{m,0}^\Lambda)}{1 - \Omega_{m,0}^\Lambda}. \qquad (31e)$$

After solving the differential equations, one can obtain the solution for H using the following relation:

$$H = H_0^\Lambda \sqrt{\frac{r}{2v}(1 - \Omega_{m,0}^\Lambda)}. \qquad (32)$$

The final change of variables that we make is due to the nature of r. This variable reaches high values and grows rapidly with z. Such behavior presents a challenge to the ANN (see discussion in Sec. III A). Therefore, we propose to use $r' = \ln r$, so now the variable that we solve for does not present such a challenge to the ANN.

It is necessary to mention that this last variable change restricts the values that R can take to just $R > 0$. However, doing an exploration of the parameter space $(z, b, \Omega_{m,0}^\Lambda) \in [0, 10] \times [0, 5] \times [0.1, 0.4]$ using a numerical solver on Eq. (29) without the variable change, we find that $R > 0$ in that region, so this is not a concern.

III. METHODS

As mentioned in the Introduction, we employ ANNs to solve differential equations. In particular, we use the NeuroDiffEq [9] Python library [10], which provides the tools to solve differential equations using ANNs. We explain some aspects of the method that are specific to this application and refer the reader to the original paper, documentation and library for more details. We explain the neural network approach for ordinary differential equations (ODEs) or systems of ODEs, which are the relevant ones in this paper, but note that the library also has the capability of solving partial differential equations (PDEs) and systems of PDEs.

A. Neural networks that solve differential equations

The problem of solving the equations using ANNs is framed as an optimization problem, where the task is to find the set of internal parameters of the network (see Appendix C for details) that minimize a cost (or loss) function L that quantifies how well the output of the ANN satisfies the differential equation. The ANN solution also has to satisfy the initial and/or boundary conditions of the system. We use the simplest form for the loss function:

$$L(\boldsymbol{u}_\mathcal{N}, t) = \sum_i^M \mathcal{R}_i(\boldsymbol{u}_\mathcal{N}, t)^2, \qquad (33)$$

where $\boldsymbol{u}_\mathcal{N} = \boldsymbol{u}_\mathcal{N}(t) = (u_\mathcal{N}^1(t), u_\mathcal{N}^2(t), \ldots, u_\mathcal{N}^M(t))$ is the vector comprised of the outputs of the ANN for each of the M dependent variables of the problem at t (the independent variable) and the superscript constitutes a label for each

dependent variable, while the subscript $\mathcal{N}$ denotes that $u_{\mathcal{N}}$ is the output of the ANN. Besides, $\mathcal{R}_i$ represents the residual of the ith equation in the differential system with M equations. While the true solution satisfies exactly the differential equation, if we evaluate the ith differential equation using the ANN solution ($\boldsymbol{u}_{\mathcal{N}}$), we obtain something different from zero, which we call the residual of the ith equation. In particular, we employ an architecture in which we have one ANN for each dependent variable of the problem. This way, each $u_{\mathcal{N}}^j(t)$ is the only output of a single ANN with a single input t. We remark that the analytical or numerical solutions are not used in the training of the ANNs [which can be inferred from Eq. (33)], stressing the unsupervised nature of the method used in this work.

The criteria for determining when a residual is acceptable is that its scale must be several orders of magnitude lower than the scale of the variables of the problem. Therefore, rescaled variables of order one are preferred for this method. For example, in the case of the Hu-Sawicki $f(R)$ model, one of the variables of the problem reaches high values and therefore we implement a change of variables so that this variable is of order one (see discussion at the end of Sec. II D). Besides, the relation between the ANN solution errors and the residuals has not been analyzed yet for the cosmological context. However, a recent work [47] studied this issue for some types of differential equations, and its application to cosmology is left for future work.

While Eq. (33) is a good representation for how well the ANNs solve the equations of a given differential system, the information for the initial/boundary conditions is missing. To impose them, a reparametrization $\tilde{\boldsymbol{u}}(\boldsymbol{u}_{\mathcal{N}}, t)$ of $\boldsymbol{u}_{\mathcal{N}}$ can be defined. For example, in the case of a system of first-order ODEs with initial conditions as $\boldsymbol{u}(t)|_{t=t_0} = \boldsymbol{u}_0$, the reparametrization must satisfy $\tilde{\boldsymbol{u}}(\boldsymbol{u}_{\mathcal{N}}, t_0) = \boldsymbol{u}_0$. An example of a good reparametrization choice for this case introduced in Ref. [48] is

$$\tilde{\boldsymbol{u}}(t) = \boldsymbol{u}_0 + (1 - e^{-(t-t_0)})\boldsymbol{u}_{\mathcal{N}}(t). \qquad (34)$$

This way, to solve ODEs, the loss function to optimize for is

$$L(\tilde{\boldsymbol{u}}, t) = \sum_i^M \mathcal{R}_i(\tilde{\boldsymbol{u}}, t)^2. \qquad (35)$$

The loss displayed in Eq. (35) takes as input all of the reparametrized outputs of the ANNs, and outputs a scalar that serves as a measurement for the accuracy of the ANNs at solving a given ODE system at t.

B. Bundle solution

An extension of the regular ANN method described before can be made, as suggested in Ref. [8], for the trained neural networks to represent a set (or bundle) of solutions of a given differential system for a continuous range of its S parameters $\boldsymbol{\theta} = (\theta^1, \theta^2, \ldots, \theta^S)$. To accomplish this, $\boldsymbol{u}_{\mathcal{N}}$ must now have S extra inputs for the parameters of the system. Therefore, if t_0 is not part of the S parameters, now Eq. (34) turns into

$$\tilde{\boldsymbol{u}}(t, \boldsymbol{\theta}) = \boldsymbol{u}_0(\boldsymbol{\theta}) + (1 - e^{-(t-t_0)})\boldsymbol{u}_{\mathcal{N}}(t, \boldsymbol{\theta}), \qquad (36)$$

where, because the parameters entering in the initial conditions are also parameters of the system, $\boldsymbol{u}_0 = \boldsymbol{u}_0(\boldsymbol{\theta})$. In addition, Eq. (35) turns into

$$L(\tilde{\boldsymbol{u}}, t, \boldsymbol{\theta}) = \sum_i^M \mathcal{R}_i(\tilde{\boldsymbol{u}}, t, \boldsymbol{\theta})^2, \qquad (37)$$

representing the loss function for the bundle solution at time t, for the differential system defined by the parameters $\boldsymbol{\theta}$. For a concrete example, the details of the implementation of this method to the cosmological background equation of the ΛCDM model can be found in Appendix A 1.

A natural next step, is to integrate this method into a pipeline whose goal is to perform a statistical analysis to test the theoretical models using observational data.

Let us say that we have a data set $\mathcal{D}$ and a model $\mathcal{M}(\boldsymbol{p})$, where $\boldsymbol{p}$ is a vector comprised of all the parameters and observables of the model. Our objective is to draw conclusions on the possible values of $\boldsymbol{p}$ using $\mathcal{D}$. In general, to achieve this outcome, one has to solve a differential system characterized by its differential equations, with initial and/or boundary conditions. We can describe the equations of the differential system by its M residuals $\mathcal{R}_i(\boldsymbol{u}, t, \boldsymbol{\theta})$, where $\boldsymbol{u} = (u^1, u^2, \ldots, u^M)$ are the dependent variables of the system. Here, it is important to clarify that $\boldsymbol{\theta}$ is the vector that contains only the parameters of the differential equations and those that are necessary to write the initial conditions, such as $\Omega_{m,0}$ in Eq. (7). This implies that $\boldsymbol{\theta}$ is contained within $\boldsymbol{p}$, and there can be parameters of the model that do not change the differential system itself but are included in the observable quantities that are considered in the statistical analysis. For example, H_0^{Λ} is included in the expression of the observable $H(z)$ in all models analyzed in this paper [see Eqs. (6), (16), (23), and (32)] but not in their respective differential equations or initial conditions. Now that the problem is defined, the next step is to set up a loss and reparametrization as was described at the start of this subsection, with the goal of training ANNs to solve the given bundle problem. Once the training is finished, $\tilde{\boldsymbol{u}}(t, \boldsymbol{\theta})$ becomes the bundle solution $\boldsymbol{u}(t, \boldsymbol{\theta})$. Then, the latter can be used together with the data in a statistical analysis to infer bounds on the parameters $\boldsymbol{p}$. The method described before is illustrated in Fig. 1 for the case where the differential system is a single ODE.
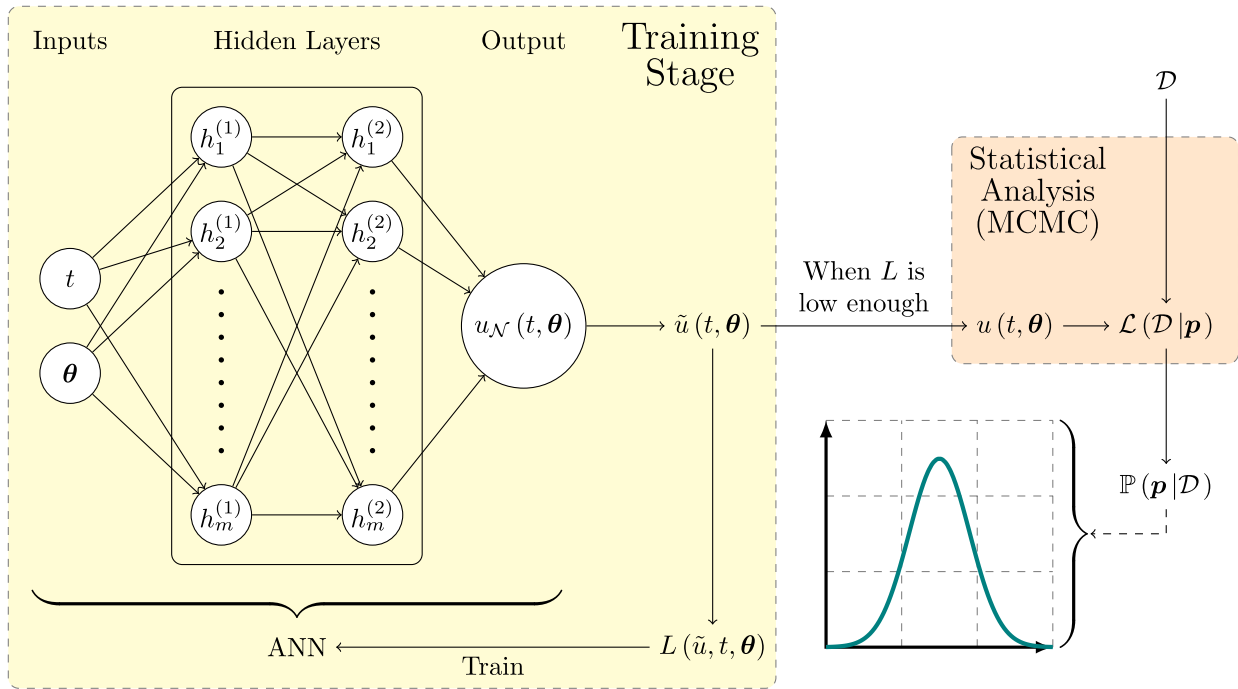
FIG. 1. Illustration of an example implementation of ANN-based bundle solutions to a differential system of a model $\mathcal{M}(\boldsymbol{p})$ in a statistical analysis with a data set $\mathcal{D}$ to obtain samples from the posterior $\mathbb{P}(\boldsymbol{p}|\mathcal{D})$.

C. Case-specific extensions

The regular and bundle methods described so far can be used to tackle any differential system. When applying the bundle method to the cosmological background scenario, we improve the method in Ref. [8] by proposing two new reparametrizations of the ANN solutions. These improvements of the ANN bundle method are applied to the CPL, quintessence, and Hu-Sawicki $f(R)$ cosmological models and are an important achievement of this work.

1. Integrating factor case

To describe the first reparametrization, let us consider a homogeneous linear first-order ODE. We can take advantage of the fact that for the general case

$$\frac{du}{dt} = uf(t), \tag{38}$$

where f is an arbitrary function of the independent variable t, the general solution to the problem with an initial condition u_0 at t_0 is

$$u = u_0 \exp\left[\int_{t_0}^{t} f(t')dt'\right]. \tag{39}$$

Now, we take this one step further into the bundle problem

$$\frac{du}{dt} = uf(t, \boldsymbol{\theta}), \tag{40}$$

with the initial condition $u_0(\boldsymbol{\theta})$. If we consider the special case where the function $f(t, \boldsymbol{\theta})$ can be expressed as $f(t, \boldsymbol{\theta}) = \sum_i^Q h_i(\boldsymbol{\theta})g_i(t),$[4] the general solution to Eq. (40) is

$$u = u_0(\boldsymbol{\theta}) \exp\left[\sum_i^Q h_i(\boldsymbol{\theta}) \int_{t_0}^{t} g_i(t')dt'\right]. \tag{41}$$

We can now take this knowledge and come up with an ANN architecture and reparametrization $\tilde{u}$ that takes advantage of it. Choosing an ANN with Q outputs i and one input t we propose

$$\tilde{u}(t, \boldsymbol{\theta}) = u_0(\boldsymbol{\theta}) \exp\left\{\sum_i^Q h_i(\boldsymbol{\theta})[u_{\mathcal{N},i}(t) - u_{\mathcal{N},i}(t_0)]\right\}, \tag{42}$$

where $u_{\mathcal{N},i}$ is the ith output of the neural network. The main advantage of this reparametrization is that it provides a way of getting a bundle solution where the neural network does not need to have the parameters of the differential system as an input. Also, it is important to note that there is only one ANN involved in this specific implementation, which corresponds to the single dependent variable of Eq. (40).

---

[4]While there can be multiple ways of expressing a given function $f(t, \boldsymbol{\theta})$ as the required sum, one should strive to choose the one that corresponds to the smallest Q possible to avoid unnecessary complexity.

The Q different outputs of this ANN are necessary for the viability of this specific implementation. This reparametrization is implemented for the bundle method in the CPL model. The details are shown in Appendix A 2.

2. Perturbative reparametrization

To make the job of minimizing the loss L in Eq. (35) easier for the ANNs, we propose to modify the reparametrization in Eq. (34) by providing it with an approximate solution $\hat{u}(t)$ to the problem, which also satisfies $\hat{u}(t = t_0) = u_0$. This way the reparametrization becomes

$$\tilde{u}(t) = \hat{u}(t) + (1 - e^{-(t-t_0)})u_{\mathcal{N}}(t). \tag{43}$$

This new reparametrization can be thought of as a perturbative solution to the problem, where the job of the second term of Eq. (43) is to correct the approximate solution.

This approach can be specialized further for the bundle case, by choosing the approximate solution as the true solution for a fixed value of one or more of the S parameters. Here, the approximate solution plays the role of a boundary condition in the bundle solution. In this way, if $\hat{u}(t, \theta^2, ..., \theta^S) = u(t, \theta^1 = \theta_0^1, \theta^2, ..., \theta^S)$ denotes the true solution when the parameter $\theta^1 = \theta_0^1$ we can recast Eq. (36) as

$$\tilde{u}(t, \boldsymbol{\theta}) = \hat{u}(t, \theta^2, ..., \theta^S)$$
$$+ (1 - e^{-(t-t_0)})(1 - e^{-(\theta^1 - \theta_0^1)})u_{\mathcal{N}}(t, \boldsymbol{\theta}), \tag{44}$$

where this can be naturally extended when more than one parameter needs to be fixed to use a true solution as the basis for the reparametrization.

Example: We show a simple example to illustrate the method with the following first-order ODE:

$$\frac{dx}{dt} = x(\alpha + \beta \sin x), \qquad x(t)|_{t=t_0} = x_0. \tag{45}$$

For this ODE, we know that the solution when $\beta = 0$ is $x = x_0 \exp[\alpha(t - t_0)]$, which we can call $\hat{x}(t, x_0, \alpha)$. Therefore, we can use $\hat{x}(t, x_0, \alpha)$ as a boundary condition in the bundle reparametrization when $\beta = 0$. So the perturbative reparametrization would be

$$\tilde{x}(t, x_0, \alpha, \beta) = \hat{x}(t, x_0, \alpha) + (1 - e^{-(t-t_0)})(1 - e^{-\beta})$$
$$\times x_{\mathcal{N}}(t, x_0, \alpha, \beta), \tag{46}$$

where $x_{\mathcal{N}}$ is the output of the ANN. Then, the loss function that the ANN must minimize to solve the problem is

$$L(\tilde{x}, t, x_0, \alpha, \beta) = \left[\frac{d\tilde{x}}{dt} - \tilde{x}(\alpha + \beta \sin \tilde{x})\right]^2. \tag{47}$$

The perturbative reparametrization is implemented for the quintessence and $f(R)$ Hu-Sawicki models. For the latter,

this allows us to tackle the problem introduced by a singularity in the system of equations (see discussion in Sec. IV B).

IV. IMPLEMENTATION SPECIFICS

In this section, we describe how we implement the methods described in Sec. III in the quintessence and Hu-Sawicki $f(R)$ models.[5] Some of the details of each implementation are omitted in this section for the sake of readability. We elaborate on these details in Appendix D. To close out this section, we share an estimation of the errors of our cosmology-informed neural networks that we obtain for each model in their respective parameter space.

A. Quintessence implementation

The perturbative reparametrization described in the last section can be implemented to successfully solve the background equations of the quintessence model. We show in Appendix B 2 that when $\lambda = 0$ the solution of Eq. (20) is equivalent to the ΛCDM solution, which implies that

$$x(N) = 0, \qquad y(N) = \sqrt{\frac{1 - \Omega_{m,0}^\Lambda}{\Omega_{m,0}^\Lambda e^{-3N} + 1 - \Omega_{m,0}^\Lambda}}. \tag{48}$$

Therefore, defining $\hat{x}(N, \Omega_{m,0}^\Lambda) = 0$ and setting $\hat{y}(N, \Omega_{m,0}^\Lambda)$ equal to the $y(N)$ in Eq. (48), we can apply the method described in Sec. III C 2, and set a boundary condition at $\lambda = 0$. Thus, the reparametrizations of the outputs of the ANNs that solve the differential system in Eq. (20) are

$$\tilde{x}(N, \lambda, \Omega_{m,0}^\Lambda) = (1 - e^{-(N-N_0)})(1 - e^{-\lambda})$$
$$\times x_{\mathcal{N}}(N, \lambda, \Omega_{m,0}^\Lambda), \tag{49a}$$

$$\tilde{y}(N, \lambda, \Omega_{m,0}^\Lambda) = \hat{y}(N, \Omega_{m,0}^\Lambda) + (1 - e^{-(N-N_0)})(1 - e^{-\lambda})$$
$$\times y_{\mathcal{N}}(N, \lambda, \Omega_{m,0}^\Lambda). \tag{49b}$$

In this model, there are two individual ANNs, one for each dependent variable. Both of them have N, λ and $\Omega_{m,0}^\Lambda$ as inputs, and the outputs are denoted as $x_{\mathcal{N}}(N, \lambda, \Omega_{m,0}^\Lambda)$ for the network assigned to x, and $y_{\mathcal{N}}(N, \lambda, \Omega_{m,0}^\Lambda)$ for the one that corresponds to y.

B. $f(R)$ gravity implementation

In a similar vein to quintessence, the solution to the $f(R)$ problem matches ΛCDM when $b = 0$.[6] The reparametrization of the variables is then

---

[5]The specifics of the implementations for the ΛCDM and CPL models are detailed in Appendix A.

[6]In the actual implementation we use $b = 10^{-13}$, because at $b = 0$ there is a singularity.

$$\tilde{\boldsymbol{u}}(z, b, \Omega_{m,0}^{\Lambda}) = \hat{\boldsymbol{u}}(z, \Omega_{m,0}^{\Lambda}) + (1 - e^{-(z-z_0)})(1 - e^{-b})$$
$$\times \boldsymbol{u}_{\mathcal{N}}(z, b, \Omega_{m,0}^{\Lambda}), \tag{50}$$

where $\boldsymbol{u}_{\mathcal{N}} = (x_{\mathcal{N}}, y_{\mathcal{N}}, v_{\mathcal{N}}, \Omega_{\mathcal{N}}, r_{\mathcal{N}})^7$ is the vector comprised of each of the individual outputs of the five ANNs that correspond to each dependent variable. Also, $\hat{\boldsymbol{u}} = (\hat{x}, \hat{y}, \hat{v}, \hat{\Omega}, \hat{r})$ is the vector with the exact solutions of each variable when $b = 0$.[8]

If we observe Eqs. (29) and (30), we see that there is a singularity in the differential equations at $b = 0$. Here, the perturbative reparametrization allows us to get values of b that are arbitrarily close to zero without the need for more computational time, which is a clear benefit of the ANN method. This is because, as the value of b tends to zero, the first term in Eq. (50) becomes dominant, and the ANN only provides a minimal correction. In contrast, the numerical approach can demand more operations in those situations, which in turn implies more computational time. This problem became so important that a perturbative analytic method was developed in Ref. [49] to obtain an approximation of the Hu-Sawicki and Starobinsky models to be used to perform parameter inference. For example, in Refs. [50–52] the output of the numerical solvers in the region close to $b = 0$ is replaced by the solution provided by the aforementioned method.

For this model, in addition to the modification of the reparametrization, we add three additional terms to the loss L that involve relationships between the dependent variables of Eq. (29). The first one is Eq. (28), while the other two are the following:

$$\Omega = \frac{2y\Omega_{m,0}^{\Lambda}(1+z)^3(r+b)}{r(1-\Omega_{m,0}^{\Lambda})(r+b-2)}, \tag{51a}$$

$$\Omega = \frac{2v\Omega_{m,0}^{\Lambda}(1+z)^3(r+b)^2}{r(1-\Omega_{m,0}^{\Lambda})[(r+b)^2 - 2b]}. \tag{51b}$$

Both of these equations are related to the conservation of mass through the use of Eq. (3) (the detailed calculation can be found in Appendix B 3). To impose these equations, the extra terms added to the loss are the squares of the relative difference of each equation's right-hand side and left-hand side. By adding these three extra terms we get the following expression:

$$L_{\mathcal{C}}(\tilde{\boldsymbol{u}}, z, \boldsymbol{\theta}) = (\tilde{\Omega} + \tilde{v} - \tilde{x} - \tilde{y} - 1)^2$$
$$+ \left\{ \frac{2\tilde{y}\Omega_{m,0}^{\Lambda}(1+z)^3(\tilde{r}+b)}{\tilde{r}\,\tilde{\Omega}\,(1-\Omega_{m,0}^{\Lambda})(\tilde{r}+b-2)} - 1 \right\}^2$$
$$+ \left\{ \frac{2\tilde{v}\Omega_{m,0}^{\Lambda}(1+z)^3(\tilde{r}+b)^2}{\tilde{r}\,\tilde{\Omega}\,(1-\Omega_{m,0}^{\Lambda})[(\tilde{r}+b)^2 - 2b]} - 1 \right\}^2. \tag{52}$$

Denoting the part of the loss that only regards the residuals as $L_{\mathcal{R}}$, the final loss for this model reads

$$L = L_{\mathcal{R}} + L_{\mathcal{C}}. \tag{53}$$

While Eq. (51) should be approximately obeyed by the ANN-based solutions without the extra term $L_{\mathcal{C}}$ in the loss, we find that explicitly imposing these relations through the extra term yields more accurate solutions.

C. Accuracy of solutions

The loss function is the quantity that reflects how well the ANNs are at solving a given differential system. The ANN method employed in this work lacks the capability of estimating the uncertainty of its solutions.[9] Therefore, we choose to quantify the accuracy of the solutions provided by our trained ANNs as the absolute value of the relative difference between the ANN solution and (i) the analytical solution in the case of the ΛCDM and CPL models or (ii) the numerical solution in the case of quintessence and $f(R)$ models. We stress that this is the only time that the analytical or numerical solutions of the cosmological models are used in this work, i.e., these solutions are never used to train the ANNs or in the inference pipeline. It is important to note that this analysis is mainly done to give a more appropriate quantification of the errors of the solutions over the parameter space of interest of each model, which can not be inferred from the value of the loss function. Nevertheless, it is still necessary to see if the errors shown here are small enough to not affect in any meaningful way the results of an inference pipeline such as the one we illustrated in Fig. 1. In the case of our solutions, this situation is tested later on in Sec. VI by comparing the results of the inference we perform using our cosmology-informed neural networks against ones found in the literature that use either similar or the same data to test the same models, but using numerical methods.

Let us begin this analysis with the accuracy of the bundle solutions of Eq. (7), which correspond to the ΛCDM model. Figure 2 shows the percentage error in the range of z and $\Omega_{m,0}$ that the ANN was trained for. It follows that the maximum error in the whole training range is $\sim 0.16\%$

---

[7]We remind the reader that we use $r'_{\mathcal{N}} = \ln r_{\mathcal{N}}$ in order that the variables are of order 1. See Sec. III.

[8]These solutions are the same as those that were used for the initial conditions in Eq. (31), but with the difference that z is not fixed at z_0.

[9]A recent work [53] showed how to estimate uncertainties in the general framework. The application to the cosmological context is left for future work.
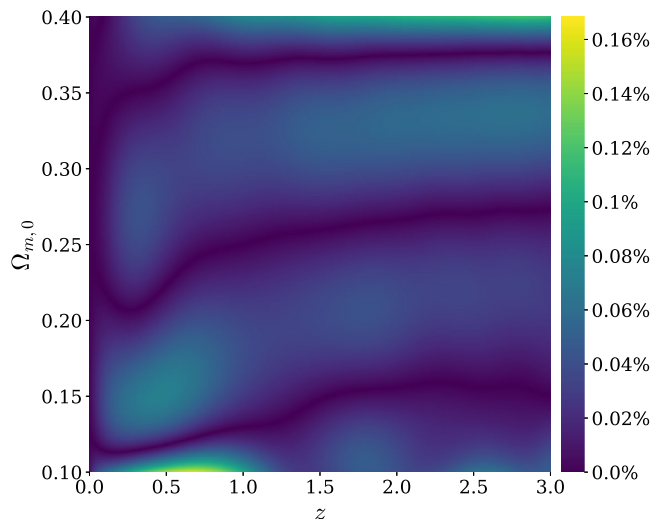
FIG. 2. Percentage error of the neural network–based bundle solution to ΛCDM in its training range, through comparison to the analytical solution.

and that the ANN solution is worse at the edges of its training range for the only parameter in the bundle ($\Omega_{m,0}$).

Now we focus on the CPL parametric dark energy model. Figure 3 shows the percentage error for four different values of ω_0 in a wide range of ω_1 and the range of z for which the solution was trained for. It follows from Eq. (13) (analytical solution) and the reparametrization of the ANN solution [Eq. (A5)] that the percentage error is independent of the initial condition. Besides, the maximum error is ∼0.01% which compared to the results obtained for
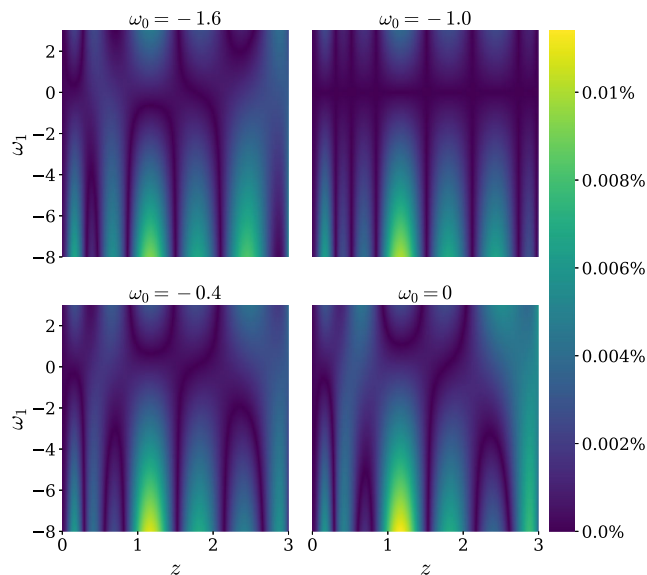

FIG. 3. Percentage error of the neural network–based bundle solution to the CPL parametric dark energy model in its training range of z, a range of ω_1 covering the range used in the statistical analysis, and four different values of ω_0. The percentage error is calculated through comparison to the analytical solution.
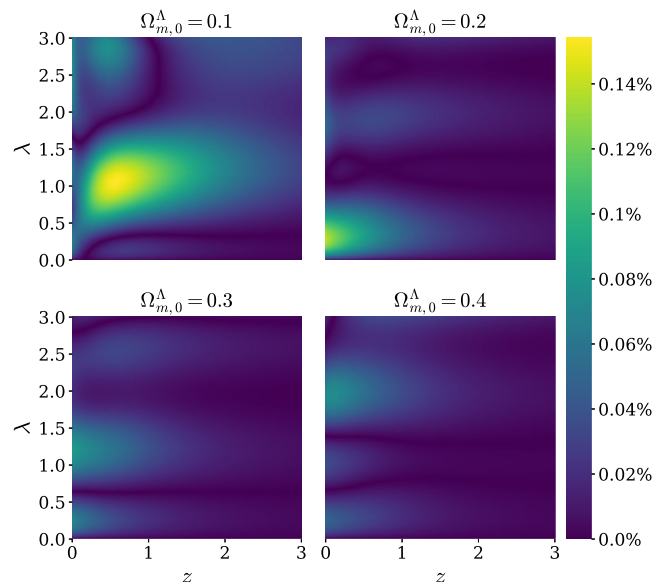

FIG. 4. Percentage error of the neural network–based bundle solutions of $H(z)/H_0^\Lambda$ in the quintessence model, through comparison to numerical solutions. The range of the comparison goes through the training range of the parameters of the bundle.

the ΛCDM model, leads us to conclude that we have obtained a more accurate solution to a more general version of the problem. In particular, the plot corresponding to $\omega_0 = 0$ in Fig. 3 contains the solution to Eq. (7) at $\omega_1 = 0$. This improvement in accuracy is related to the method described in Sec. III C 1.

Figure 4 shows the percentage error for the quintessence model for four different values of $\Omega_{m,0}^\Lambda$, and going through the whole training range of λ. In this case, since there is more than one dependent variable, the percentage error is calculated for $H(z)/H_0^\Lambda$. For the numerical solution, we use an adaptive explicit Runge-Kutta method of order 5(4), where the local estimated error e for any dependent variable of the system u^i can be bounded such that $|e| < a_{tol} + r_{tol}|u^i|$.[10] For the plot in Fig. 4 we consider $r_{tol} = 10^{-10}$ and $a_{tol} = 10^{-13}$.

Although the ANNs are trained for $z \in [0, 10]$, the plot is limited to the redshift range that is relevant for the data we use. Besides, in the region that is not shown in the plot, the value of the percentage error is lower or at most equal to the one in Fig. 4 and it gets lower as z gets closer to the initial condition z_0.

Finally, Fig. 5 shows the percentage error for $H(z)/H_0^\Lambda$ in the $f(R)$ Hu-Sawicki model; here the values of the error tolerance for the numerical solver are set as follows: $r_{tol} = 10^{-11}$ and $a_{tol} = 10^{-16}$. We also restrict the range of b in the plot to the region that is more relevant for the cosmological inference. However, it is important to report that there is a small region with an error ∼5%, which is located at

---

[10]This is in fact a slight oversimplification. For a more detailed description we refer the reader to Ref. [54].
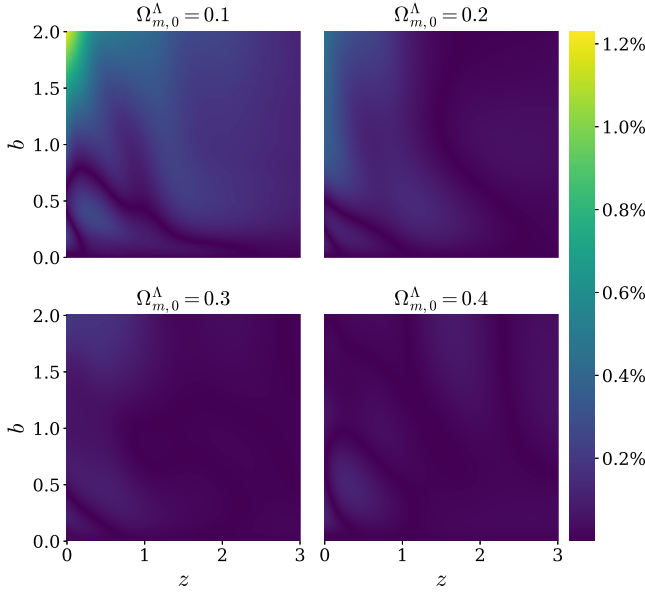
FIG. 5. Percentage error of the neural network–based bundle solution of $H(z)/H_0^\Lambda$ in the Hu-Sawicki model, through comparison to numerical solutions. The range of the comparison goes through a section of the training range of the parameters of the bundle.

$(b, \Omega_{m,0}^\Lambda) \in [2, 5] \times 0.1$ far away from the obtained 2σ region of the parameter space obtained later on in Sec. VI with both data sets (see Fig. 7).

V. OBSERVATIONAL DATA

In this section, we describe the different data sets that we use to do inference on the parameters of the models presented in Sec. II, along with their associated likelihood functions.

A. Cosmic chronometers

The method of cosmic chronometers allows to obtain measurements of $H(z)$ at different values of z through the measurement of the difference in age Δt between passively evolving galaxies whose light is observed at different redshifts, which differ by Δz [55].[11] Using the relation between the scale factor, a, and the cosmological redshift, z, we can obtain the equation

$$H = \frac{\dot{a}}{a} = \frac{-1}{1+z}\frac{dz}{dt}, \qquad (54)$$

from which it is possible to determine $H(z)$, if the following approximation holds: $\Delta z/\Delta t \simeq dz/dt$. A key advantage of this method is that it relies on the measurement of relative ages dt, which has smaller systematic

---

[11]We denote passively evolving galaxies as those galaxies with no current star formation or interaction with other galaxies.

errors than the determination of absolute ages. Besides, the measurement of dt depends only on atomic physics and therefore is model independent. We consider 30 measurements of $H(z)$ obtained with this technique [22–27] within a range of redshift $z \in [0.07, 1.965]$ which we summarize in Table IX of Appendix E. The likelihood function for this data set can be written as

$$\mathcal{L}_{\mathrm{CC}} \propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{30}\left[\frac{H^{\mathrm{obs}}(z_i) - H^{\mathrm{th}}(z_i, \boldsymbol{p})}{\sigma_{H^{\mathrm{obs}}(z_i)}}\right]^2\right\}, \quad (55)$$

where $H^{\mathrm{obs}}(z_i)$ refers to the observational measurements of $H(z)$ obtained with the CC method and $\sigma_{H^{\mathrm{obs}}(z_i)}$ refers to the corresponding error (see Table IX). The model's prediction is $H^{\mathrm{th}}(z_i, \boldsymbol{p})$ obtained with the ANN method where $\boldsymbol{p}$ is a vector that includes the free parameters of the model. All the models share $\Omega_{m,0}^\Lambda$ and H_0^Λ while the following extra parameters are added: ω_0 and ω_1 for the CPL model, λ for the quintessence model and b for the $f(R)$ Hu-Sawicki model; see Sec. II for details.

B. Type Ia supernovae

Type Ia supernovae are powerful and extremely luminous stellar explosions. Due to the homogeneity of their light curves and spectra, they are considered as standard candles, which allows to determine cosmological distances from their luminosities. In this work, we consider the 1048 measurements of supernovae luminosities comprised in the Pantheon compilation in the redshift range $z \in [0.01, 2.3]$ [28]. The observed quantity is the distance modulus μ, which can be expressed by a modified version of the formula proposed by Tripp in Ref. [56]:

$$\mu = m_b - M + \alpha x_1 - \beta c + \Delta_M + \Delta_B, \qquad (56)$$

where the color c, the light-curve shape parameter x_1 and the log of the overall flux normalization m_b are provided by the compilation through light-curve fitting using SALT2 as presented in Ref. [57] and implemented in SNANA [58]. In addition, α is the coefficient of the relation between luminosity and stretch, β is the coefficient of the relation between luminosity and color, and M is the absolute B-band magnitude of a fiducial SNIa with $x_1 = 0$ and $c = 0$. The distance correction, Δ_B, is based on predicted biases from simulations and Δ_M is a distance correction based on the host galaxy mass of the SNIa, which is determined by

$$\Delta_M = \gamma[1 + e^{(-(m-m_{\mathrm{step}})/\tau)}], \qquad (57)$$

where γ is a relative offset in luminosity, m is the mass of the host galaxy, and m_{step} is the mass step for the split that separates the SNIa into ones that have a host galaxy with high mass and low mass. Last, τ is an exponential transition term in the Fermi function that describes the relative

probability of masses being on one side or the other of the split. In this way, α, β, γ and M are called nuisance parameters and they are usually estimated via a statistical analysis where a theoretical model for the dynamics of the Universe is assumed. However, as described before, the nature of these parameters is astrophysical and therefore their values should not depend on the cosmological model.

In this work, we use the mean values estimated by the Pantheon compilation [28] ($\alpha = 0.0154 \pm 0.06$, $\beta = 3.02 \pm 0.06$, $\gamma = 0.053 \pm 0.009$) where a ΛCDM model is assumed. These values have been verified by two independent analyses with alternative cosmological models: the one in Ref. [50] where the $f(R)$ Hu-Sawicki model was assumed and the one in Ref. [59] performed in the context of the modified gravity theory. Moreover, M will be considered as an additional free parameter for all models in the statistical analysis (meaning that it is included in p). To perform the statistical analysis, we compare the value of μ obtained from the data using Eq. (56) against the theoretical value provided by

$$\mu(z) = 25 + 5 \log_{10}(d_L(z)), \qquad (58)$$

where the luminosity distance $d_L(z)$ is related to $H(z)$ through

$$d_L(z) = (1 + z) \int_0^z \frac{dz'}{H(z')}. \qquad (59)$$

The likelihood function concerning this data set can be expressed as [28]

$$\mathcal{L}_{\text{SNIa}} \propto \exp\left\{-\frac{1}{2}\Delta\boldsymbol{\mu}^T \cdot \boldsymbol{C}^{-1} \cdot \Delta\boldsymbol{\mu}\right\}, \qquad (60)$$

where the vector $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}^{\text{obs}} - \boldsymbol{\mu}^{\text{th}}(\boldsymbol{p})$ contains the difference between the observed value and the theoretical prediction of the distance modulus [Eqs. (58) and (59)] for each of the 1048 measurements of the Pantheon compilation. Furthermore, the covariance matrix $\boldsymbol{C}$ is defined as follows:

$$\boldsymbol{C} = \boldsymbol{D}_{\text{stat}} + \boldsymbol{C}_{\text{sys}}, \qquad (61)$$

where the statistical matrix $\boldsymbol{D}_{\text{stat}}$ has only a diagonal component which includes measurement errors such as the photometric error of the SNIa distance, the uncertainty from the distance bias correction, the uncertainty from the peculiar velocity, the redshift measurement uncertainty (added in quadrature), and the uncertainty from stochastic gravitational lensing. Moreover, $\boldsymbol{C}_{\text{sys}}$ is the systematic covariance matrix.

C. Baryon acoustic oscillations

Since the pioneering work in Refs. [60–62] studying the connection between matter fluctuations and the anisotropies in the CMB, the physical mechanisms operating in the early Universe have now been well established. Before the recombination of hydrogen, baryons and photons interact through Thomson scattering, while baryons and dark matter interact gravitationally. The result is a photon-baryon fluid that suffers acoustic oscillations, thanks to

TABLE I. Constraints on the parameters of the ΛCDM model. The table shows the 68% and 95% confidence intervals for each free parameter. The reduced χ_ν^2 is also shown for each analysis.

Data		$\Omega_{m,0}$	$H_0\left[\frac{\text{km/s}}{\text{Mpc}}\right]$	M	χ_ν^2
CC + SNIa	68% C.L.	[0.281, 0.322]	[67.251, 70.959]	[−19.434, −19.326]	
	95% C.L.	[0.26, 0.342]	[65.331, 72.622]	[−19.484, −19.271]	
	Best fit	0.301	68.994	−19.38	0.969
CC + SNIa + BAO	68% C.L.	[0.284, 0.306]	[67.861, 69.277]	[−19.422, −19.347]	
	95% C.L.	[0.275, 0.317]	[67.175, 69.944]	[−19.444, −19.347]	
	Best fit	0.296	68.545	−19.396	0.97

TABLE II. Constraints on the parameters of the CPL model. The table shows the 68% and 95% confidence intervals for each free parameter. The reduced χ_ν^2 is also shown for each analysis.

Data		ω_0	ω_1	$\Omega_{m,0}$	$H_0\left[\frac{\text{km/s}}{\text{Mpc}}\right]$	M	χ_ν^2
CC + SNIa	68% C.L.	[−1.172, −0.822]	[−2.089, 1.401]	[0.296, 0.437]	[66.36, 70.249]	[−19.458, −19.342]	
	95% C.L.	[−1.363, −0.648]	[−5.104, 1.757]	[0.111, 0.47]	[64.563, 72.216]	[−19.514, −19.287]	
	Best fit	−1.051	−0.554	0.343	68.724	−19.39	0.97
CC + SNIa +BAO	68% C.L.	[−1.05, −0.889]	[−0.295, 0.494]	[0.279, 0.307]	[66.219, 69.182]	[−19.46, −19.37]	
	95% C.L.	[−1.12, −0.803]	[−0.827, 0.734]	[0.266, 0.32]	[64.594, 70.444]	[−19.509, −19.334]	
	Best fit	−0.974	0.088	0.291	67.441	−19.426	0.971

TABLE III.   Constraints on the parameters of a quintessence model with an exponential potential. The table shows the 68% and 95% confidence intervals for each free parameter. The reduced χ_ν^2 is also shown for each analysis.

Data		λ	$\Omega_{m,0}$	$H_0 \left[\frac{km/s}{Mpc}\right]$	M	χ_ν^2
CC + SNIa	68% C.L.	[0, 0.584]	[0.27, 0.317]	[67.097, 70.753]	[−19.432, −19.324]	
	95% C.L.	[0, 0.941]	[0.245, 0.338]	[65.382, 72.629]	[−19.486, −19.273]	
	Best fit	0.008	0.301	68.995	−19.38	0.97
CC + SNIa + BAO	68% C.L.	[0.159, 0.744]	[0.284, 0.306]	[66.837, 68.856]	[−19.442, −19.383]	
	95% C.L.	[0, 0.893]	[0.275, 0.317]	[65.638, 69.677]	[−19.474, −19.354]	
	Best fit	0.532	0.295	67.733	−19.415	0.97

the pull of gravity towards matter overdensities, and the push of photons due to the radiation pressure. When the temperature of the Universe becomes sufficiently low due to cosmic expansion, hydrogen atoms are formed, the density of free electrons dramatically drops, and Thomson scattering becomes inefficient. The subsequent decoupling between matter and radiation freezes the oscillations, leaving an imprint both in the CMB and in the distribution of galaxies thanks to the BAO mentioned above. The BAO offers a characteristic scale, namely, the sound horizon r_d at recombination, which is the largest comoving distance a sound wave could have traveled before recombination, and it provides a very useful standard rod. Thus, we can use observations of galaxy clustering as a cosmological probe. The quantities related to the BAO data relevant for this paper are the Hubble parameter $H(z)$, the Hubble distance

$$D_H(z) = \frac{1}{H(z)}, \tag{62}$$

the angular diameter distance

$$D_A(z) = \frac{1}{1+z} \int_0^z \frac{dz'}{H(z')}, \tag{63}$$

the comoving angular diameter distance

$$D_M(z) = (1+z)D_A(z), \tag{64}$$

and the combination of D_M and D_H, introduced in Ref. [63]

$$D_V(z) = \left[D_A(z)^2 \frac{z}{H(z)}\right]^{1/3}. \tag{65}$$

To do inference on the BAO data that we use, we also need to calculate the sound horizon r_d. To do this, we follow the method employed in Ref. [64] which uses the fitting formula for the drag epoch redshift z_d introduced in Ref. [65]. It is important to remark that the BAO data sets we use, which are listed in Table X, are independent. Thus, we define the likelihood function as follows:

$$\mathcal{L}_{BAO} \propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{20}\left[\frac{A_i^{obs}(z_i) - A_i^{th}(z_i, \boldsymbol{p})}{\sigma_{A_i^{obs}(z_i)}}\right]^2\right\}, \tag{66}$$

where the A_i denotes the specific quantity that is measured (such as Hr_d, D_H/r_d, D_A/r_d, and others; see the first column of Table X), $A_i^{obs}(z_i)$ refers to the observed value and $\sigma_{A_i^{obs}(z_i)}$ refers to the observational error.[12] Moreover, $A_i^{th}(z_i, \boldsymbol{p})$ indicates the model's prediction for the quantity A_i at z_i.
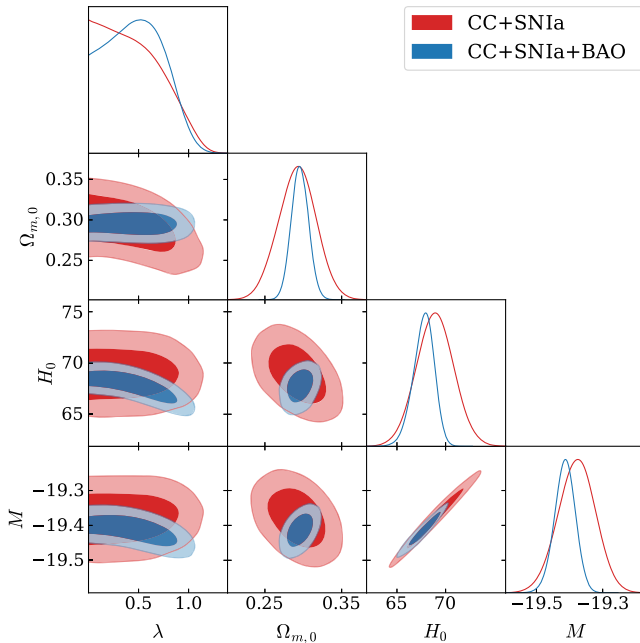

FIG. 6.   Results of the statistical analysis for the quintessence model with an exponential potential. The darker and brighter regions correspond to the 68% and 95% confidence regions, respectively. The plots on the diagonal show the posterior probability density for each of the free parameters of the model.

---

[12]For measurements with both statistical and systematic errors, we use the sum of both in quadrature.

TABLE IV. Our results from the statistical analysis for the parameters of the quintessence model with an exponential potential, as well as similar analyses done with traditional numerical solvers. The table shows the 95% confidence intervals for each free parameter.

Data		λ	$\Omega_{m,0}$	$H_0[\frac{\text{km/s}}{\text{Mpc}}]$
SNIa + BAO + CMB + H_0	Akrami *et al.*	[0, 0.8]	N/A	N/A
CC + SNIa + BAO + H_0	Tosone *et al.*	[0, 0.709]	[0.257, 0.307]	[69.2, 73.9]
CC + SNIa + BAO	This work	[0, 0.941]	[0.275, 0.317]	[65.638, 69.677]

TABLE V. Constraints on the parameters of a cosmological model in Hu-Sawicki $f(R)$ gravity with $n = 1$. The table shows the 68% and 95% confidence intervals for each free parameter. The reduced χ_ν^2 is also shown for each analysis.

Data		b	$\Omega_{m,0}$	$H_0[\frac{\text{km/s}}{\text{Mpc}}]$	M	χ_ν^2
CC + SNIa	68% C.L.	[0, 0.649]	[0.24, 0.302]	[67.236, 70.834]	[−19.425, −19.32]	
	95% C.L.	[0, 1.313]	[0.207, 0.324]	[65.469, 72.547]	[−19.474, −19.266]	
	Best fit	0	0.301	68.994	−19.38	0.97
CC + SNIa + BAO	68% C.L.	[0.031, 0.411]	[0.28, 0.302]	[65.897, 68.289]	[−19.469, −19.399]	
	95% C.L.	[0, 0.708]	[0.27, 0.314]	[64.611, 69.235]	[−19.505, −19.369]	
	Best fit	0.218	0.293	67.349	−19.426	0.97

VI. RESULTS

In this section, we present the results of the statistical analyses performed with the data sets described in Sec. V. We perform the statistical analyses using the likelihood functions through a Markov chain Monte Carlo (MCMC) method. For this, we set the same flat priors for the parameters that all four models share: $H_0^\Lambda \in [50, 80]\ \text{km s}^{-1}\ \text{Mpc}^{-1}$, $\Omega_{m,0}^\Lambda \in [0.1, 0.4]$ [for the CPL model with just CC and SNIa the prior is too restrictive for its parameter space as can be confirmed with the results of the statistical analysis (see Table II),[13] so we use $\Omega_{m,0}^\Lambda \in [0, 0.6]$], and the absolute magnitude $M \in [-22, -18]$ for all models. We recall that in both the ΛCDM and CPL models, $\Omega_{m,0}^\Lambda = \Omega_{m,0}$ and $H_0^\Lambda = H_0$ which is not necessarily the case in both the quintessence and $f(R)$ models. Therefore, for these models we carry out a postprocessing of the chains obtained by MCMC, where for each chain we calculate $H_0 = H(z = 0)$ and use Eq. (22) to obtain $\Omega_{m,0}$. For all models we report the values of H_0 and $\Omega_{m,0}$. To implement MCMC, we use the emcee [66] Python library, which implements the samplers introduced in Ref. [67]. Also, to plot the contours displayed in this paper, we use the GetDist [68] Python library. For each theoretical model considered in this paper we carry out a first analysis with only CC and SNIa data and a second analysis where the BAO data are added to the previous ones. In both cases the joint likelihood used is the product of all the individual likelihoods involved.

Starting with ΛCDM, we show the results of the statistical analysis for this model in Table I and we observe that the constraints on the parameters are tighter when the BAO data are included. Furthermore, the obtained confidence intervals of the analysis with CC and SNIa are in agreement within 1σ[14] with the ones obtained in Ref. [51], where the same data sets were considered.

Regarding the CPL model, the flat priors that we use for the parameters that are not shared with the other models are $\omega_0 \in [-0.4, -1.6]$ and $\omega_1 \in [-8, 3]$. Results of the statistical analysis for this model are shown in Table II. As in the case of the ΛCDM model, we note that the constraints tighten when the BAO data are added, as this effect is more pronounced for the parameter ω_1. Furthermore, the obtained confidence intervals are in agreement within 1σ with the ones obtained in Ref. [69], where only the CC data set were considered.

As regards the statistical analysis for the quintessence model considered in this work, we set a flat prior for λ, namely $\lambda \in [0, 3]$. Results are shown in Table III and Fig. 6. We note three changes in the results of the statistical analysis when the BAO data set is added: (i) the constraints are tighter, (ii) the correlation between $\Omega_{m,0}$ and H_0 changes sign, and (iii) the model parameter λ shows a small correlation with H_0, while its correlation with $\Omega_{m,0}$ noticeably decreases. The comparison of our results with the ones obtained by other authors for the same model are summarized in Table IV. In this table, we show the

---

[13]The first MCMC runs show that the obtained bounds on $\Omega_{m,0}$ match the priors, and therefore it is necessary to enlarge them.

[14]This means that the 1σ confidence intervals obtained with one statistical analysis in a given work overlap with the 1σ confidence intervals calculated in another.
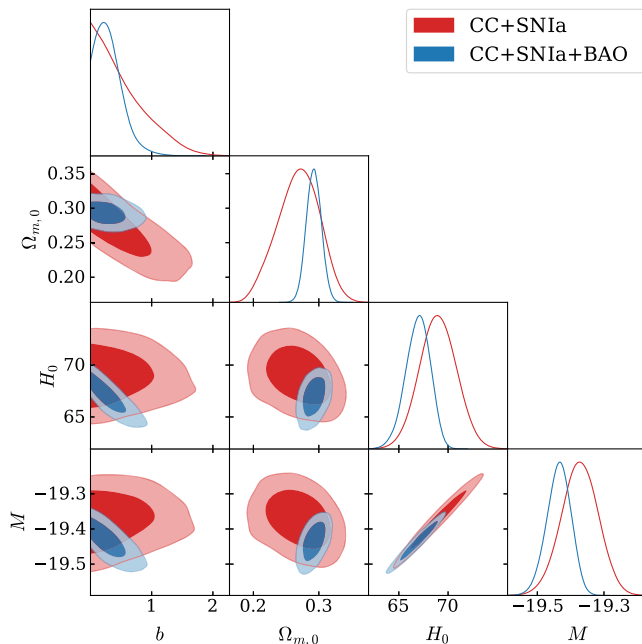
FIG. 7. Results of the statistical analysis for the $f(R)$ Hu-Sawicki model. The darker and brighter regions correspond to the 68% and 95% confidence regions respectively. The plots on the diagonal show the posterior probability density for each of the free parameters of the model.

95% confidence intervals for each parameter instead of the 68% ones because only the former were available in Ref. [70].

The obtained 95% confidence interval for λ is in agreement with the ones obtained in Ref. [70]. Besides, there is also agreement with the 95% intervals of λ and $\Omega_{m,0}$ reported in Ref. [71], and a marginal agreement with the value of H_0. However, we note that the analysis performed in Ref. [71] included the current value of H_0 reported in Ref. [72], which affects the obtained results.

Concerning the statistical analysis of the $f(R)$ Hu-Sawicki model, we set a flat prior for b, namely $b \in [0, 5]$ and show the results in Table V and Fig. 7. Similar to the analysis performed for the quintessence model, we note the following effects when the BAO data set is added to the

statistical analysis: (i) the constraints are tighter, (ii) the correlation between $\Omega_{m,0}$ and H_0 changes sign, and (iii) the parameter b shows an important correlation with H_0, while its correlation with $\Omega_{m,0}$ noticeably decreases.

Let us now compare our results with similar ones published in the literature, which are summarized in Table VI (a similar comparison was shown in Fig. 3 of Ref. [50]). We start with the ones performed in Refs. [50,51] where the same data sets of CC and SNIa were considered. First, we note that the estimates in Ref. [51] for the b parameter were smaller than the ones obtained in this work, while an excellent agreement is found with the results of Ref. [50]. The reason for this is the following: due to the computational instabilities of Eq. (29) in the region $b \to 0$, in Ref. [51] the expression for $H(z)$ was computed from a series expansion proposed in Ref. [49]. This approach allowed the authors to explore only a small region of the parameter space while there are no computational instabilities in the ANN method, and thus a wider region can be explored. On the other hand, in Ref. [50], the series expansion was only used for $b < 0.15$ and a numerical integration was performed for other values of b allowing for a complete exploration of the parameter space. We stress that the excellent agreement between the results of this work and the ones in Ref. [50] where the very same data sets were considered, which shows that the errors of the ANN solutions are not significant enough to affect the results of the statistical analysis. Furthermore, we also note that our estimates for the b parameter are in agreement within 1σ with the ones of another analysis performed in Ref. [51], where data from six systems of strongly lensed quasars analyzed by the H0LiCOW Collaboration [73] were added to the previous ones (CC + SNIa). On the other hand, another recent work [52] also tested the $f(R)$ Hu-Sawicki model considering besides the CC and SNIa data, redshift-space distortions (RSD), BAO, and CMB data. We note that the resulting values of the b parameter were more restricted than in this work, but this can be explained since the former analysis considered more data sets than ours.

Finally, Figs. 8 and 9 show the behavior of $H(z)$ and the SNIa apparent magnitude m_b [related to $d_L(z)$] respectively for each of the cosmological models considered in this

TABLE VI. Our results from the statistical analysis for the parameters of a cosmological model in Hu-Sawicki $f(R)$ gravity with $n = 1$, as well as similar analyses done with traditional numerical solvers. The table shows the 68% confidence intervals for each free parameter.

| Data		b	$\Omega_{m,0}$	$H_0 \left[\frac{\text{km/s}}{\text{Mpc}}\right]$
CC + SNIa	Leizerovich *et al.*	[0, 0.587]	[0.241, 0.303]	[67.296, 70.701]
	D'Agostino *et al.*	[0.07, 0.49]	[0.261, 0.314]	[67.5, 71.5]
	This work	[0, 0.649]	[0.24, 0.302]	[67.236, 70.834]
CC + SNIa + H0LiCOW	D'Agostino *et al.*	[0.03, 0.29]	[0.244, 0.29]	[71, 73.8]
CC + SNIa + BAO	Leizerovich *et al.*	[0.024, 0.382]	[0.281, 0.304]	[65.874, 68.313]
	This work	[0.031, 0.411]	[0.28, 0.302]	[65.897, 68.289]
CC + SNIa + BAO + RSD + CMB	Farrugia *et al.*	[0, 0.00005]	[0.295, 0.307]	[68.043, 69.007]
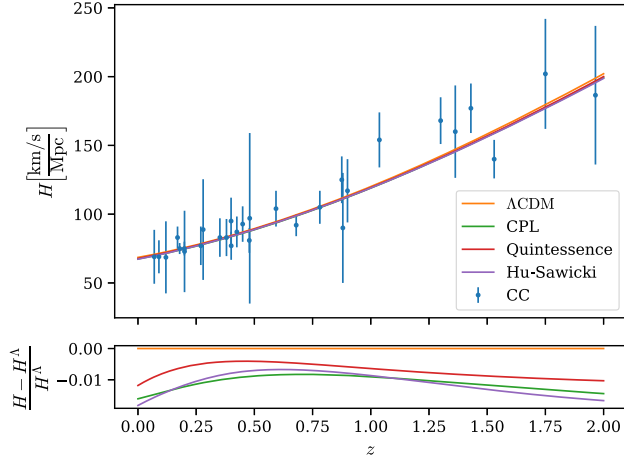
FIG. 8. Top: $H(z)$ as a function of z for all the models studied (using the parameters corresponding to the best fit obtained from the combination CC + SNIa + BAO) together with CC data. Bottom: relative difference between the values of H in the top panel for each model with respect to ΛCDM.
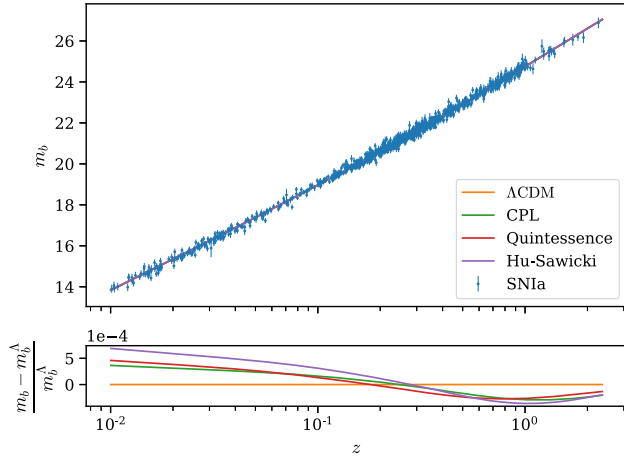

FIG. 9. Top: apparent magnitude of a supernova, m_b, as a function of z for all the models studied (using the parameters corresponding to the best fit obtained from the combination CC + SNIa + BAO) together with SNIa data. The relation between m_b and the Hubble parameter can be seen by looking at Eqs. (56), (58), and (59). Bottom: relative difference between the values of m_b in the top panel for each model with respect to ΛCDM.

paper. For this, we set the values of the free parameters at the best-fit values obtained by cosmological inference with CC + SNIa + BAO. The relative difference of each model with respect to ΛCDM is also shown in Figs. 8 and 9, and it allows us to conclude that the behavior of all these models concerning the CC and SNIa data is very similar.

VII. COMPUTATIONAL EFFICIENCY

To discuss the topic of computational efficiency we perform an analysis on the amount of floating point operations (FLOPs) that it takes for a specific numerical

TABLE VII. MCMC running times in minutes for different hardware and models using the CC + SNIa data set. For the CPU case, we use a ninth-generation Intel i7 (8 cores), while in the GPU case, we use a single Nvidia A100.

Method (hardware)	$f(R)$	Quintessence
Numerical method (CPU)	328 min	120 min
ANN method (CPU)	162 min	375 min
ANN method (GPU)	88 min	240 min

method of common use [Runge-Kutta 5(4) with interpolation [74,75]] to solve the differential equations of a given initial value problem, as well as the equivalent for the ANN approach. In this section we only summarize the results of such an analysis, while the detailed calculation and discussion can be found in Appendix F.

Our analysis shows that if the number of points of the domain where the solution is evaluated is large or the problem is easy to integrate, as in the case of the quintessence model, the numerical solver is likely to outperform the ANNs in terms of computational cost, whereas in the case where the opposite is true (i.e., few points to evaluate or a problem that is difficult to integrate), ANNs are most likely preferred. This is the case for the $f(R)$ Hu-Sawicki model, where the problem is rather more difficult to solve than in the quintessence model. This is confirmed in Table VII where the computational times for the MCMC runs performed for the numerical and ANN methods with different hardware are detailed. The main factor that makes the ANN method preferable in the context of problems that are hard to solve is the fact that once the ANNs are trained, the solutions can be used indefinitely. This is due to the process of solving the model's differential equations being performed only during the training of the network, whereas in the numerical method, the integration is done each time one needs to know an output of the model for a given parameter combination. This feature can be very important when using the obtained solution to perform statistical analyses like with MCMC. In the case of the $f(R)$ Hu-Sawicki model this is further amplified by the fact that, as we have discussed in Sec. IV B, the ANNs require no additional computing time when there is a singularity in the differential system and an analytical solution at the singularity is available, which is not the case with traditional numerical solvers. Besides, we stress that we have left the optimization of the ANN method for future work, while the numerical method has been already tested and optimized. In fact, the use of parallelization can make the cost of both training and parameter inference much more efficient in terms of actual time spent executing the computations, thus making the scenario discussed before more favorable to ANNs.

VIII. CONCLUSIONS

In this work, we trained artificial neural networks through an unsupervised method to serve as bundle

solutions to the differential equations of the background dynamics of the Universe for four different cosmological models. We then performed a statistical analysis using these solutions with observational data to draw conclusions on the possible values of each model's parameters. The novelty in this paper relies on the unsupervised nature of the training of the ANNs. This is in stark contrast to applications of neural networks in cosmology such as in Ref. [4], where the nature of training was supervised due to the use of solutions provided by numerical methods as training data.

We found our results from the statistical analysis to be compatible with the ones in the literature that used traditional numerical solvers. In addition, we provided estimations of the errors of our ANN-based solutions that show all errors to be either below or around $\sim 1\%$ for the 95% confidence level region of the parameter space of all models for all the statistical analyses performed in this work.

Among the main achievements of this paper we stress the development of improvements to the ANN bundle method. One of these improvements was instrumental to making the method viable in the cosmological context by providing it with the capability to solve a special case of stiff[15] problems without additional computational cost. We used this implementation in one of the models we solved, namely the differential equations of a cosmology in the $f(R)$ Hu-Sawicki gravity model, which has been shown to require significant computation time with numerical solvers for a specific region of its parameter space where the equations become stiff due to the presence of a singularity.

We showed in this paper that unsupervised ANN bundle solutions of differential equations can be effectively used to constrain physical models, and pose a viable alternative to traditional numerical methods. We also shared our codes to facilitate the implementation for future users in the following repository: https://github.com/at-chantada/cosmo-nets.

We also performed an analysis of the computational efficiency of the ANN-based method against a common use numerical solver. We showed (see Sec. VII) that once the training of the ANNs is concluded, the time required to perform parameter inference with MCMC is less than in the case of using traditional numerical methods when the differential system is not easy to integrate. In addition, the parallelization capabilities of ANNs can be exploited to make the use of these solutions even more computationally efficient. Nevertheless, we stress that the optimization of the ANN method to reduce computing times was not the main focus of this paper, and thus is left for future work.

While we showed an effective use of the ANN method, there are still many aspects of the method that can be improved which constitutes future work. Some of these

future improvements include uncertainty quantification of the solutions provided by the method, model selection, a general tuning of the architecture (e.g., finding optimal activation functions, numbers of hidden layers and units, etc.), and improvements in the design of the training so as to achieve lower values for the loss, while also lowering the computational cost of training, namely the amount of iterations and batch sizes needed to obtain an optimal solution.

In summary, we used ANNs as the tool of choice to solve the differential equations of cosmological models, and used the solutions they provided to perform MCMC to constrain the models' parameters using data, without the use of numerical solvers in any step of the process.

Finally, our results show that ANNs provide a promising alternative to traditional numerical solvers for use in an inference process.

APPENDIX A: IMPLEMENTATION OF THE ANN METHOD IN SPECIFIC MODELS

1. Implementation in the ΛCDM model

To solve the background equation of the ΛCDM model [Eq. (7)], using the bundle solution, one starts by assigning an ANN to each dependent variable. As described before, the inputs of the ANN are the independent variable, and the parameters of the bundle. Therefore, in this case, we have a single ANN whose output we label $x_{\mathcal{N}}(z, \Omega_{m,0})$. Then, to enforce the initial condition of the system, the reparametrization in Eq. (36) is implemented as follows:

$$\tilde{x}(z, \Omega_{m,0}) = \Omega_{m,0} + (1 - e^{-z})x_{\mathcal{N}}(z, \Omega_{m,0}). \quad \text{(A1)}$$

We recall that $x = \kappa \rho_m(z)/3H_0^2$, so $\tilde{x}$ in Eq. (A1) satisfies the initial condition in Eq. (7). Next, the loss is constructed from the residual, $\mathcal{R}$, which is defined by the differential equation of the system,

$$\mathcal{R}(\tilde{x}, z, \Omega_{m,0}) = \frac{d\tilde{x}}{dz} - \frac{3\tilde{x}}{1+z}, \quad \text{(A2)}$$
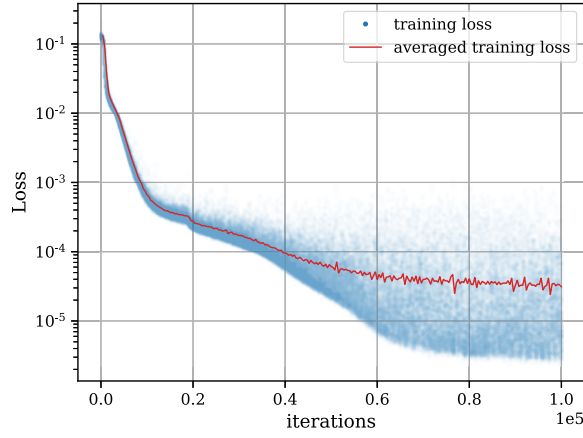

FIG. 10. Example of a plot of the total loss per iteration of training. In blue are the values of the total loss at each iteration, and in red the average value of the total loss per 400 iterations. This particular example comes from training an ANN to solve ΛCDM [Eq. (7)], defining the reparametrization in Eq. (A1) and the loss in Eq. (A3).

which can be used in Eq. (37) to get the loss of the problem:

$$L(\tilde{x}, z, \Omega_{m,0}) = \left(\frac{d\tilde{x}}{dz} - \frac{3\tilde{x}}{1+z} \right)^2. \qquad (A3)$$

As the loss is defined as a function of $\tilde{x}$, this allows that the process of training leads to a solution that meets the initial condition.

The training proceeds as follows. For a fixed value of the network's internal parameters [here we refer to the weights

and biases detailed in Eq. (C3)], $\tilde{x}(z, \Omega_{m,0})$ is obtained and the respective loss [Eq. (A3)] is evaluated. Next, the training proceeds as an optimization problem performed with the gradient descent method where the network's internal parameters vary in order to minimize the loss function. Figure 10 shows one of the main indicators in the training of the ANN for the ΛCDM model, namely, the total loss per iteration. Further details on how to train the ANNs were described in Ref. [6], and in Ref. [10] on NEURODDIFFEQ's specific implementation.

The main takeaway from Fig. 10, is that the value of the total loss decreases rapidly at the start of the training, and as the training continues this behavior starts to slow down. This is often the expected behavior of the loss during training. Therefore, the usual approach is to stop training when the relative change in the average value of the total loss per 400 iterations is of order 10^{-4}.

2. Implementation in the CPL model

To solve the background equation of the CPL dark energy model, we can take advantage of the reparametrization introduced in Sec. III C 1. We can identify $f(t, \boldsymbol{\theta})$ in Eq. (40) as $f(z, \omega_0, \omega_1)$ in the Friedmann equation of the CPL model [Eq. (15)]:

$$f(z, \omega_0, \omega_1) = \left[3(1+\omega_0)\frac{1}{1+z} + 3\omega_1\frac{z}{(1+z)^2} \right]. \qquad (A4)$$

Following Eq. (42), the reparametrization $\tilde{x}$ for the outputs of the ANN in this model is

$$\tilde{x}(z, \omega_0, \omega_1, \Omega_{m,0}) = (1 - \Omega_{m,0}) \exp\left\{ 3(1+\omega_0)[x_{\mathcal{N},1}(z) - x_{\mathcal{N},1}(z_0)] + 3\omega_1[x_{\mathcal{N},2}(z) - x_{\mathcal{N},2}(z_0)] \right\}, \qquad (A5)$$

where $x_{\mathcal{N},1}$ and $x_{\mathcal{N},2}$ represent the two outputs of the ANN and the respective loss function can be defined as follows:

$$L(\tilde{x}, z, \omega_0, \omega_1) = \left[\frac{d\tilde{x}}{dz} - \frac{3\tilde{x}}{1+z}\left(1 + \omega_0 + \frac{\omega_1 z}{1+z} \right) \right]^2. \qquad (A6)$$

APPENDIX B: MODELS' CALCULATIONS

In this section we show some models' calculations that were avoided in the main text for the sake of readability.

1. Dynamics of r

This subsection's goal is to show the calculation of the differential equation for r that results in Eq. (29), starting from the definition

$$\frac{dr}{dz} = \frac{dt}{dz}\frac{dr}{dt} = \frac{1}{\Lambda}\frac{dt}{dz}\frac{dR}{dt} = \frac{-\dot{R}}{\Lambda H(1+z)}. \qquad (B1)$$

Now using the definition of x in Eq. (27) as well as $\Gamma = Rf_R/f_{RR}$, we get

$$\frac{dr}{dz} = \frac{-1}{1+z}\left(\frac{\dot{R}}{\Lambda H} \right)$$

$$= \frac{-1}{1+z}\left(\overbrace{\frac{R}{\Lambda}}^{=r} \overbrace{\frac{\dot{R}f_{RR}}{Hf_R}}^{=x} \overbrace{\frac{f_R}{Rf_{RR}}}^{=\Gamma} \right) = -\frac{r\Gamma x}{1+z}. \qquad (B2)$$

2. $\lambda = 0$

Here we show that when $\lambda = 0$ the behavior of the quintessence model shown in Sec. II C, is the very same as

that of the ΛCDM model. Using $\lambda = 0$, which results in $dV/d\phi = 0$ we can now solve for $\dot{\phi}(z)$ in Eq. (18) and get

$$\dot{\phi}(z) = \dot{\phi}_0 \left(\frac{1+z}{1+z_0} \right)^3. \tag{B3}$$

Given the fact that the initial condition $x_0 = 0$ results in $\dot{\phi}_0 = 0$, using Eq. (B3) and the definition of x in Eq. (19), we can conclude that $x(N) = 0 \ \forall \ z$ when $\lambda = 0$. Now, using the initial condition for y_0 we can obtain $V_0 = (3/\kappa)(H_0^\Lambda)^2(1 - \Omega_{m,0}^\Lambda)$, and using this result with Eq. (17a) one gets $H(z) = H^\Lambda(z)$. Last, by inserting the expression of $H^\Lambda(z)$ [Eq. (6)] into Eq. (23) we get

$$y(N) = \sqrt{\frac{1 - \Omega_{m,0}^\Lambda}{\Omega_{m,0}^\Lambda e^{-3N} + 1 - \Omega_{m,0}^\Lambda}}. \tag{B4}$$

3. Extra loss terms in $f(R)$ implementation

To close out the models' extra calculations, here we show how to get Eq. (51). Starting from the definition of Ω in Eq. (27), and using Eq. (3), we can obtain

$$\Omega = \frac{\kappa \rho_{m,0}(1+z)^3}{3H^2 f_R}. \tag{B5}$$

a. First equation

To obtain Eq. (51a), we start by multiplying and dividing Eq. (B5) by $2f$, and using the definition of y in Eq. (27), the following expression is obtained:

$$\Omega = 2y\kappa\rho_{m,0}(1+z)^3 f^{-1}. \tag{B6}$$

To develop this expression further, we first express Eq. (26) in a more compact form using $r = R/\Lambda$:

$$f(r) = \Lambda \left\{ r - 2\left[1 - \frac{1}{\frac{r}{b}+1} \right] \right\} = \Lambda \left\{ r - 2\left[\frac{r}{r+b} \right] \right\}$$

$$= 3(H_0^\Lambda)^2(1 - \Omega_{m,0}^\Lambda)r\left(\frac{r+b-2}{r+b} \right), \tag{B7}$$

where in the last step $\Lambda = 3(H_0^\Lambda)^2(1 - \Omega_{m,0}^\Lambda)$ was used. Using this last result with the expression for Ω in Eq. (B6), while remembering the definition $\Omega_{m,0}^\Lambda = \kappa\rho_{m,0}/3(H_0^\Lambda)^2$, we obtain

$$\Omega = \frac{2y\Omega_{m,0}^\Lambda(1+z)^3(r+b)}{r(1 - \Omega_{m,0}^\Lambda)(r+b-2)}. \tag{B8}$$

b. Second equation

In a similar fashion to the first equation, we start by multiplying and dividing Eq. (B5) by $R/2$, and make use of the definition of v in Eq. (27), which yields

$$\Omega = \frac{2v\kappa\rho_{m,0}(1+z)^3}{Rf_R}. \tag{B9}$$

We now need an expression for f_R. Performing the derivative with respect to R of f, and using Eq. (26) and $r = R/\Lambda$,

$$f_R = 1 - \frac{2b}{(\frac{R}{\Lambda}+b)^2} = \frac{(r+b)^2 - 2b}{(r+b)^2}. \tag{B10}$$

Replacing this expression for f_R in Eq. (B9) while multiplying and dividing by Λ, we obtain

$$\Omega = \frac{2v\kappa\rho_{m,0}(1+z)^3(r+b)^2}{\Lambda r[(r+b)^2 - 2b]}. \tag{B11}$$

Finally, using $\Lambda = 3(H_0^\Lambda)^2(1 - \Omega_{m,0}^\Lambda)$ along with $\Omega_{m,0}^\Lambda = \kappa\rho_{m,0}/3(H_0^\Lambda)^2$ in Eq. (B11),

$$\Omega = \frac{2v\Omega_{m,0}^\Lambda(1+z)^3(r+b)^2}{r(1 - \Omega_{m,0}^\Lambda)[(r+b)^2 - 2b]}. \tag{B12}$$

APPENDIX C: DETAILS OF THE ANN METHOD

In this appendix, we list a series of details that were omitted in the main explanation of the ANN method. In Fig. 1 we show a schematic of an example ANN. To outline how to calculate the outputs of the ANN, we start by translating what each element of that graph represents mathematically. Each circle represents a unit, and each one of these, except for the ones in the input layer, has an associated bias and activation function. For example, we can denote the bias of the jth unit on the second hidden layer as $b_j^{(2)}$, and the activation function as $g_j^{(2)}$. Then, each line that connects a pair of units has an associated weight. In a similar vein to the previous example, we can denote the weight of the line that connects the ith unit in the input layer to the jth unit in the first hidden layer as $w_{ij}^{(1)}$. Let us consider an ANN architecture composed by an input layer with n units, one hidden layer with m units each and an output layer with k units; this is an ANN of size (n, m, k), and the outputs $u_{\mathcal{N},l}$ with inputs x_i[16] can be expressed as

---

[16]We recall that the inputs of the ANN are the independent variable and the parameters of the differential system ($\boldsymbol{\theta}$).

TABLE VIII. Training details of the models. We state the amount of training iterations employed, the training range for the parameters of each model, as well as their corresponding batch sizes used during training. In addition, we specify the architecture of the ANN that represents the dependent variables and the number of internal parameters of the network. In the last column the time that the training required is listed along with the hardware used to perform the training. In the case of a CPU, we used an Intel i7-6500U, while in the case of a GPU, we used a single Nvidia A100.

Model	Iterations	Training range	Batch size	ANN architecture (in, hid, hid, out)	Number of internal parameters of the ANNs	Training time (Hardware)
ΛCDM	100000	$z \in [0, 3]$	64	$x : (2, 32, 32, 1)$	1185	20 minutes (CPU)
		$\Omega_{m,0} \in [0.1, 0.4]$	64			
CPL	50000	$z \in [0, 3]$	32	$x : (1, 32, 32, 2)$	1186	10 minutes (CPU)
Quintessence	200000	$z \in [0, 10]$	32	$x : (3, 32, 32, 1)$	1217	17 hours (CPU)
		$\lambda \in [0, 3]$	32	$y : (3, 32, 32, 1)$	1217	
		$\Omega_{m,0}^{\Lambda} \in [0.1, 0.4]$	32			
Hu-Sawicki	600000	$z \in [0, 10]$	128	$x : (3, 32, 32, 1)$	1217	2 days (GPU)
				$y : (3, 32, 32, 1)$	1217	
		$b \in [0, 5]$	128	$v : (3, 32, 32, 1)$	1217	
		$\Omega_{m,0}^{\Lambda} \in [0.1, 0.4]$	64	$\Omega : (3, 32, 32, 1)$	1217	
				$r' : (3, 32, 32, 1)$	1217	

$$u_{\mathcal{N},l} = \sum_{j}^{m} w_{jl}^{\text{out}} g_{j}^{(1)} \left(\sum_{i}^{n} w_{ij}^{(1)} x_i + b_j^{(1)} \right) + b_l^{\text{out}}. \quad \text{(C1)}$$

We can dissect this last equation further by noticing that the outputs of the hidden layer are essentially

$$h_j^{(1)} = g_j^{(1)} \left(\sum_{i}^{n} w_{ij}^{(1)} x_i + b_j^{(1)} \right). \quad \text{(C2)}$$

Therefore, if we wanted to calculate the outputs of an ANN of size (n, m, m, k) (this is an ANN that includes two hidden layers with m units each), it would be

$$u_{\mathcal{N},l} = \sum_{j}^{m} w_{jl}^{\text{out}} g_{j}^{(2)} \left(\sum_{i}^{m} w_{ij}^{(2)} h_i^{(1)} + b_j^{(2)} \right) + b_l^{\text{out}}, \quad \text{(C3)}$$

where $h_i^{(1)}$ is the same as in Eq. (C2).

It follows from the last equation that for a choice of continuous activation functions g_i, the continuity of $u_{\mathcal{N},l}$ is assured since the composition of continuous functions is also continuous. Besides, the last equation also shows that $u_{\mathcal{N},l}$ is differentiable with respect to the independent variable and the parameters of the differential system (x_i).

On the other hand, Table VIII shows details that involve specifics of the training like the number of iterations for each model, the number of training points for each input considered in each iteration which is called the batch size, the training range of each parameter, the specific values of the ANN architecture used for the training of each cosmological model, the number of internal parameters of the networks (here we refer to the weights and biases

described above) and the training time for each cosmological model.

For every network in this work we use the Adam [76] optimizer with its default values and all of them use tanh as their only activation function. Regarding the sampling of the points from the independent variable and the parameter spaces, we use a uniform distribution for each one, where the limits correspond to their respective training range.

APPENDIX D: DETAILS OF THE IMPLEMENTATION OF THE ANN METHOD TO COSMOLOGICAL MODELS

In this appendix we describe minor changes to the reparametrization and loss function of some of the models analyzed in this paper, and variable changes to some inputs of the networks so that they do not reach outside of the interval [0, 1] are also discussed in this appendix.

1. ΛCDM

This being the simplest model, we make only one modification to the default loss in Eq. (37) [or more explicitly Eq. (A3)], this being the addition of a weighting function as proposed in Ref. [8]. With this consideration we set the final loss used to solve Eq. (7), for a bundle in $\Omega_{m,0}$ to be

$$L(\tilde{x}_m, z, \Omega_{m,0}) = \mathcal{R}(\tilde{x}_m, z, \Omega_{m,0})^2 e^{-2(z-z_0)}, \quad \text{(D1)}$$

where $\tilde{x}_m$ is the reparametrization of the variable, that uses the default bundle reparametrization in Eq. (36).

2. Parametric dark energy

In spite of the fact that the network trained using Eq. (A5) only has one input, the training is not strictly speaking only done in z. For each iteration of training of this ANN, the loss is calculated for two combinations of the pair (ω_0, ω_1), namely, $(-1, -0.6)$ and $(-0.8, 0)$.

3. Quintessence

Due to the increasing complexity of the problem, and the fact that the inputs of the ANNs corresponding to N and λ are not bound between 0 and 1, we make two changes to Eq. (20). Instead of using N and λ, we define $\lambda' = \lambda/3$ and $N' = N/|N_0| + 1$ (making $N'_0 = 0$). This way, both of these inputs to the ANNs are bound as $N' \times \lambda' \in [0, 1] \times [0, 1]$. Making these changes, the differential system to solve is now

$$\begin{cases} \frac{1}{|N_0|}\frac{dx}{dN'} = -3x + 3\frac{\sqrt{6}}{2}\lambda'y^2 + \frac{3}{2}x(1 + x^2 - y^2), \\ \frac{1}{|N_0|}\frac{dy}{dN'} = -3\frac{\sqrt{6}}{2}xy\lambda' + \frac{3}{2}y(1 + x^2 - y^2). \end{cases} \quad \text{(D2)}$$

Similarly to ΛCDM, we also add an exponential term to this model's loss. But, due to Eq. (49) having a boundary condition on top of an initial one, we choose to modify the weighting function to account for this. With this modification, the loss to solve Eq. (D2) is

$$L(\tilde{x}, \tilde{y}, N', \lambda', \Omega^\Lambda_{m,0}) = \sum_i^2 \mathcal{R}_i(\tilde{x}, \tilde{y}, N', \lambda', \Omega^\Lambda_{m,0})^2 \\ \times e^{-2\lambda'N'}. \quad \text{(D3)}$$

4. $f(R)$

For this model, z is not really a suitable variable to use in conjunction with a reparametrization as in Eq. (34), due to the fact that z_0 is the maximum value of z in the training range. Because of this, and for the same reasons as listed for quintessence, we define $z' = 1 - z/z_0$ and $b' = b/5$. With these changes, the final differential system is now

$$\begin{cases} \frac{dx}{dz'} = \frac{-z_0}{z_0(1-z')+1}(-\Omega - 2v + x + 4y + xv + x^2), \\ \frac{dy}{dz'} = \frac{z_0}{z_0(1-z')+1}(vx\Gamma(r') - xy + 4y - 2yv), \\ \frac{dv}{dz'} = \frac{z_0v}{z_0(1-z')+1}(x\Gamma(r') + 4 - 2v) \\ \frac{d\Omega}{dz'} = \frac{-z_0\Omega}{z_0(1-z')+1}(-1 + 2v + x), \\ \frac{dr'}{dz'} = \frac{z_0\Gamma(r')x}{z_0(1-z')+1}, \end{cases} \quad \text{(D4)}$$

where Γ is now

$$\Gamma(r') = \frac{(e^{r'} + 5b')[(e^{r'} + 5b')^2 - 10b']}{20b'e^{r'}}. \quad \text{(D5)}$$

In addition to this, we also perform a change to the loss analogous to the one in quintessence making the part of final loss $L = L_{\mathcal{R}} + L_{\mathcal{C}}$ to solve Eq. (D4) corresponding to just the residuals $L_{\mathcal{R}}$ as

$$L_{\mathcal{R}}(\tilde{\boldsymbol{u}}, z', b', \Omega^\Lambda_{m,0}) = \sum_i^5 \mathcal{R}_i(\tilde{\boldsymbol{u}}, z', b', \Omega^\Lambda_{m,0})^2 e^{-2b'z'}. \quad \text{(D6)}$$

The final detail to note, is the addition of a hyperparameter α in the reparametrization expressed by Eq. (50), that helps tuning how good the solution provided by the ΛCDM model approximates to the true solution of the Hu-Sawicki model. This change gives the following general expression for the reparametrization used in Eq. (D6):

$$\tilde{\boldsymbol{u}}(z', b', \Omega^\Lambda_{m,0}) = \hat{\boldsymbol{u}}(z', \Omega^\Lambda_{m,0}) + (1 - e^{-z'})(1 - e^{-\alpha b'}) \\ \times \boldsymbol{u}_{\mathcal{N}}(z', b', \Omega^\Lambda_{m,0}). \quad \text{(D7)}$$

It is evident that $\alpha \sim 0$ corresponds to a belief that the Lambda CDM solution is close to the real one for all values of the parameters, whereas $\alpha \gg 1$ denotes that the real solution is far from the one provided by the Lambda CDM model. We choose to use $\alpha = 1/6$.[17]

APPENDIX E: DATA SETS

In this appendix we show the values and associated errors of the CC and BAO data that we use to obtain the results of the statistical analysis. We list in Table IX the measurements of the Hubble parameter H from cosmic chronometers, and in Table X the measurements of different physical quantities that use the scale provided by baryon acoustic oscillations.

APPENDIX F: COMPUTATIONAL EFFICIENCY

In this appendix, we explore how the ANN method stands against its numerical counterpart, and in which circumstances one might find either method more appealing. As mentioned in the Introduction, neural networks can provide an advantage over numerical methods when used to test a model whose predictions are obtained by solving a system of differential equations. To quantify this advantage we calculate an estimation of the amount of FLOPs that it takes for a numerical method to solve the differential equations of a given initial value problem, as well as the equivalent for the ANN approach. In addition, we remark on some other appealing features of the ANN method.

1. Floating point operations estimation

The general form of an initial value problem is as follows:

---

[17]This value was found by trial and error with the goal of obtaining the lowest value for the loss L.

TABLE IX. Measurements of the Hubble parameter H using the Cosmic Chronometers technique described in Sec. V A.

z	$H(z) \pm \sigma_H [\frac{\mathrm{km/s}}{\mathrm{Mpc}}]$	References
0.09	69 ± 12	[22]
0.17	83 ± 8	
0.27	77 ± 14	
0.4	95 ± 17	
0.9	117 ± 23	
1.3	168 ± 17	
1.43	177 ± 18	
1.53	140 ± 14	
1.75	202 ± 40	
0.48	97 ± 62	[23]
0.88	90 ± 40	
0.1791	75 ± 4	[24]
0.1993	75 ± 5	
0.3519	83 ± 14	
0.5929	104 ± 13	
0.6797	92 ± 8	
0.7812	105 ± 12	
0.8754	125 ± 17	
1.037	154 ± 20	
0.07	69 ± 19.6	[25]
0.12	68.6 ± 26.2	
0.2	72.9 ± 29.6	
0.28	88.8 ± 36.6	
1.363	160 ± 33.6	[26]
1.965	186.5 ± 50.4	
0.3802	83 ± 13.5	[27]
0.4004	77 ± 10.2	
0.4247	87.1 ± 11.2	
0.4497	92.8 ± 12.9	
0.4783	80.9 ± 9	

TABLE X. Measurements of different quantities reliant on the scale provided by Baryon Acoustic Oscillations that we used in this paper. For the values with two errors, the first corresponds to the statistical uncertainty, while the second represents the systematic error. In those cases we used the sum of both errors in quadrature.

Observable	z_{eff}	Value $\pm$ error	References
D_V/r_d	0.15	4473 ± 0159	[29]
	0.44	11.548 ± 0.559	[30]
	0.6	14.946 ± 0.680	
	0.73	16.931 ± 0.579	
	1.52	26.005 ± 0.995	[31]
D_A/r_d	0.81	10.75 ± 0.43	[32]
D_M/r_d	0.38	$10.272 \pm 0.135 \pm 0.074$	[33]
	0.51	$13.378 \pm 0.156 \pm 0.095$	
	0.61	$15.449 \pm 0.189 \pm 0.108$	
	0.698	17.65 ± 0.3	[34]
	1.48	30.31 ± 0.79	[35]
	2.3	37.77 ± 2.13	[36]
	2.4	36.6 ± 1.2	[37]
D_H/r_d	0.698	19.77 ± 0.47	[34]
	1.48	13.23 ± 0.47	[35]
	2.3	9.07 ± 0.31	[36]
	2.4	8.94 ± 0.22	[37]
$Hr_d[\mathrm{km/s}]$	0.38	$12044.07 \pm 251.226 \pm 133.002$	
	0.51	$13374.09 \pm 251.226 \pm 147.78$	[33]
	0.61	$14378.994 \pm 266.004 \pm 162.558$	

$$\frac{d\boldsymbol{u}}{dt} = f(t, \boldsymbol{u}), \qquad \boldsymbol{u}(t)|_{t=t_0} = \boldsymbol{u}_0, \qquad \text{(F1)}$$

where $f(t, \boldsymbol{u})$ is an arbitrary function, and the vector $\boldsymbol{u} = (u^1, u^2, \ldots, u^M)$ contains the M dependent variables of the system. With the general problem defined, we begin our analysis by analyzing the FLOPs of numerical methods, and we note that the computational cost can significantly vary depending on the specific method used. Because all the problems in this work are initial value problems, we choose to analyze the FLOPs of SciPy's [77] implementation[18] of the adaptive explicit Runge-Kutta method RK5(4) [74,75]. In this implementation the number of function evaluations of $f(t, \boldsymbol{u})$ in Eq. (F1) is independent from the amount of points that the algorithm is asked to output. Besides, due to the adaptive nature of the integrator, the number of function evaluations is highly dependent on both the differential system and the error tolerance for the solver.

---

[18]The analysis for this implementation was done for SciPy 1.8.0, which was the latest release at the time of submission.

Taking these facts into account, we can define a rough expression for the number of FLOPs, $\mathcal{F}_{\mathrm{RK5(4)}}$, that it takes for the integrator to output the values of the M dependent variables at N_{out} points for a given value of the parameters of the differential system ($\boldsymbol{\theta}$), as follows [54]:

$$\mathcal{F}_{\mathrm{RK5(4)}} \simeq 9N_{\mathrm{out}}(M+1) + N_{\mathrm{eval}}(\mathcal{F}_{\mathrm{eval}} + 20M), \quad \text{(F2)}$$

where N_{eval} is the number of evaluations of $f(t, \boldsymbol{u})$ and $\mathcal{F}_{\mathrm{eval}}$ is the amount of FLOPs that it takes to do such an evaluation. The first term in Eq. (F2) corresponds to the interpolation cost, and the second to the Runge-Kutta steps. We see from this expression, that the cost associated with this numerical method scales linearly with a slope of order 10 on the number of points. On the other hand, the term corresponding to the integration does not depend on the number of points. Besides, it is important to note that N_{eval} is the only factor in Eq. (F2) that is going to be different each time the integration is done for a different value of $\boldsymbol{\theta}$. This becomes especially relevant in cases like the one we mentioned in Sec. IV B where because of a singularity in the parameter space of the differential system, there is a region of that space, close to the singularity, where the numerical solver requires more function evaluations to obtain a solution.

For the ANN method we first only count the number of FLOPs that it takes for the trained network to give the

output we want, and later we include the cost of training the network, which is where most of the computational cost is.

To estimate the amount of FLOPs that it takes to calculate the k outputs of such an ANN given n inputs (called a forward pass), which we denote $\mathcal{F}_{\text{forw}}$, we focus on the output of an ANN of size (n, m, m, k)[19] [see Eq. (C2)]. If we ignore the FLOPs necessary to calculate the activation function, all of the operations inside the ANN are just additions and multiplications. We count one FLOP per multiplication or addition. First there is the calculation of the m outputs of the first hidden layer. Looking at Eq. (C2), we see that this requires n additions and n multiplications, which makes for a total of $2nm$ FLOPs to calculate all the outputs of the first hidden layer. Following the same procedure for each layer until reaching the output layer we obtain

$$\mathcal{F}_{\text{forw}} \simeq 2mn + 2m^2 + 2mk = 2m(n + m + k). \quad \text{(F3)}$$

Regarding the cost of the neural network method specifically, we recall from Sec. III B that the inputs of the networks used correspond to the independent variable and parameters of the differential system. In addition, there is the process of passing the outputs of the ANN through the reparametrization, which entails some additional cost that depends on the reparametrization itself. Therefore, the final amount of FLOPs, $\mathcal{F}_{\text{ANN}}$, that it takes for an ANN to output the values of a dependent variable at N_{out} different points can be estimated as follows:

$$\mathcal{F}_{\text{ANN}} \simeq N_{\text{out}}(\mathcal{F}_{\text{forw}} + \mathcal{F}_{\text{rep}}), \quad \text{(F4)}$$

where $\mathcal{F}_{\text{rep}}$ is the amount FLOPs that it takes to calculate the reparametrization. Besides, because we associate each dependent variable with an individual ANN, as mentioned in Sec. III, the formula in Eq. (F4) is valid for each dependent variable. Therefore if for example one needs only two of the M dependent variables, the cost is just $2\mathcal{F}_{\text{ANN}}$, and there is no need to calculate the $M - 2$ remaining. To close on this part of the analysis, the training cost must be addressed. The training of the ANNs constitutes an iterative process where in each iteration the loss L must be computed at N_{train} training points. To calculate the value of L at each of these points, the outputs of the M ANNs need to be computed, which constitutes M forward passes. Next, the outputs of the ANNs must be reparametrized. Once this has been achieved, the reparametrized outputs are used in Eq. (37) to calculate the value of L.[20]

---

[19]In Fig. 1 we show an example of an ANN with a certain amount of inputs, hidden layers (each with m units), and a single output. A bit more generally, we can think of an ANN with an input layer with n units, two hidden layers with m units each, and an output layer with k units, which we can denote as (n, m, m, k).
[20]We ignore the contribution from the FLOPs necessary to calculate this last step.

Then, the internal parameters of the ANNs are optimized to minimize the value of L, called a backward pass. The cost of this last procedure can be estimated to be at most 3 times the cost of a forward pass [78]. Therefore, the amount of FLOPs, $\mathcal{F}_{\text{train}}$, required for the training made up of N_{iter} iterations can be estimated to be

$$\mathcal{F}_{\text{train}} \simeq N_{\text{iter}} N_{\text{train}} M(4\mathcal{F}_{\text{forw}} + \mathcal{F}_{\text{rep}}). \quad \text{(F5)}$$

It is evident from comparing Eqs. (F4) and (F5) that in the majority of cases the training is where most of the computational cost is going to be. In this case we see that the cost scales with three main factors: (i) the sizes of the ANNs, (ii) the batch size, and (iii) the amount of iterations. All of these can be related to the accuracy of the results from the method. For the case of the batch size and amount of iterations, because there is no risk of overfitting, a higher number of these two quantities tends to correlate with more accurate solutions. In the case of the network size, it is not necessarily always the case that a bigger network, meaning more hidden layers and/or more units per hidden layer, is the route to achieve better results. Taking this into account, we can say that computational cost associated with a complex problem and/or a need for low-error solutions mostly translates into more computation resources in the training stage of the ANN method.

2. Parallelization

While the preceding calculations show a high amount of FLOPs in the ANN method, we have to take into account the viable hardware options to start translating this into time spent computing. This is where the highly parallelizable nature of ANNs comes into play. The usual choice of hardware to accomplish this task is GPUs. During training, these make it possible to calculate the outputs of the forward and backward passes for the N_{train} points in parallel, although this is limited by the memory of the GPU. This also applies for a trained ANN, where the N_{out} points can also be computed in parallel. Both of these uses result in a significant speed up when compared to using CPUs when N_{out} (or in training N_{train}) is large enough. An example of this last case can be seen in Ref. [4], where the process of MCMC with trained ANNs was done on a GPU in an inference pipeline.

3. Storage

An additional capability to consider is the portability of the solutions. In the case of numerical methods, the option to store the solutions often comes in the form of storing a mesh of solutions in the parameter space to do interpolation afterwards. Such a method has several disadvantages such as the storage size of the mesh growing exponentially with the amount of parameters, and the interpolation adding a new source of error that can only be mitigated through making a denser mesh. Therefore, in many cases it is often

more efficient to run the numerical solver each time a solution is needed, instead of storing the solutions. Conversely, neural networks are lightweight and, because of the very nature of the method being used in this work, the only things one needs are the trained ANNs and their associated reparametrizations to the dependent variables. To give a precise example of the storage needed for this method, each individual ANN used in this work takes ~ 10 kB to store.

4. When to use ANN

Combining what we described throughout this appendix, we can now make a more educated selection of the use cases where one might find either numerical methods or ANN-based methods more appealing for solving an initial value problem.

If for a moment we ignore the training, we can take Eqs. (F2) and (F4) and get

$$\frac{\mathcal{F}_{\text{RK5(4)}}}{M\mathcal{F}_{\text{ANN}}} \simeq \frac{9N_{\text{out}}(M+1) + N_{\text{eval}}(\mathcal{F}_{\text{eval}} + 20M)}{MN_{\text{out}}(\mathcal{F}_{\text{forw}} + \mathcal{F}_{\text{rep}})}, \quad \text{(F6)}$$

where we multiplied the ANN cost by M to account for the FLOPs necessary to calculate all of the dependent variables. From Eq. (F6) we can evaluate two different scenarios. First is the case where $9N_{\text{out}}(M+1) \gg N_{\text{eval}}(\mathcal{F}_{\text{eval}} + 20M)$, which represents a situation where either the amount of points is rather large, or the problem at hand is easy to integrate. It is clear that the Runge-Kutta method is the less costly implementation for this case, as we can see if we approximate Eq. (F6) as

$$\frac{\mathcal{F}_{\text{RK5(4)}}}{M\mathcal{F}_{\text{ANN}}} \simeq 9(\mathcal{F}_{\text{forw}} + \mathcal{F}_{\text{rep}})^{-1}. \quad \text{(F7)}$$

On the other hand, given the case where the relation is $9N_{\text{out}}(M+1) \ll N_{\text{eval}}(\mathcal{F}_{\text{eval}} + 20M)$, we are left with

$$\frac{\mathcal{F}_{\text{RK5(4)}}}{M\mathcal{F}_{\text{ANN}}} \simeq \frac{N_{\text{eval}}(\mathcal{F}_{\text{eval}} + 20M)}{MN_{\text{out}}(\mathcal{F}_{\text{forw}} + \mathcal{F}_{\text{rep}})}. \quad \text{(F8)}$$

This approximation is applicable to situations where the amount of points is fairly low or the problem is difficult to solve. Therefore, in these situations we can use Eq. (F8) as an approximate rule of thumb to determine which method is likely to be the most efficient.

Now it is important to recall the training. From what we discussed previously about Eq. (F5), we can see that even in the situations where Eq. (F8) suggests the use of ANNs, the additional cost of training the networks can still make it the less efficient option. Although this may seem discouraging, we have to be reminded of two important things mentioned before: the fact that the training is done just once, and parallelization. First, because the training is performed just once, we can see that the cost from this process can be easily overlooked if the solutions are used several times. We can see that this can be the case in a scenario that involves using MCMC. For example, if in such a scenario the number of chains is N_{chains}, the total cost from the use of the ANN solutions is $N_{\text{chains}}M\mathcal{F}_{\text{ANN}} + \mathcal{F}_{\text{train}}$. Therefore, the larger N_{chains}, the less significant the cost of training the ANN. Besides, the use of parallelization can make the cost of both training and using the ANNs much more efficient in terms of actual time spent executing the computations, thus making the scenario discussed before more favorable to ANNs when the hardware options are available.

To conclude this appendix, it is important to note that the analysis done here for the computational capabilities of the ANN method is valid for any kind of differential system. The only caveat to have in mind for the most general case would be that now the n inputs of the network represent the parameters as well as the multiple independent variables, instead of just a single independent variable in the initial value case.

[1] 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).

[2] 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.

[3] 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.

[4] 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).

[5] M. L. Piscopo, M. Spannowsky, and P. Waite, Solving differential equations with neural networks: Applications to the calculation of cosmological phase transitions, Phys. Rev. D **100**, 016002 (2019).

[6] I. Lagaris, A. Likas, and D. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks **9**, 987 (1998).

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, New York, 2016).

[8] C. Flamant, P. Protopapas, and D. Sondak, Solving differential equations using neural network solution bundles, arXiv:2006.14372.

[9] https://github.com/NeuroDiffGym/neurodiffeq.

[10] F. Chen, D. Sondak, P. Protopapas, M. Mattheakis, S. Liu, D. Agarwal, and M. D. Giovanni, NeuroDiffEq: A Python package for solving differential equations with neural networks, J. Open Source Softwaare **5**, 1931 (2020).

[11] A. G. Riess, A. V. Filippenko, P. Challis, A. Clocchiatti, A. Diercks, P. M. Garnavich, R. L. Gilliland, C. J. Hogan, S. Jha, R. P. Kirshner *et al.*, Observational evidence from supernovae for an accelerating universe and a cosmological constant, Astrophys. J. **116**, 1009 (1998).

[12] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom *et al.*, Measurements of Ω and Λ from 42 high-redshift supernovae, Astrophys. J. **517**, 565 (1999).

[13] A. G. Riess, W. Yuan, L. M. Macri, D. Scolnic, D. Brout, S. Casertano, D. O. Jones, Y. Murakami, G. S. Anand, L. Breuval, T. G. Brink *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).

[14] 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).

[15] P. Shah, P. Lemos, and O. Lahav, A buyer's guide to the Hubble constant, Astron. Astrophys. Rev. **29**, 9 (2021).

[16] E. Abdalla, G. F. Abellán, A. Aboubrahim, A. Agnello, Özgür Akarsu, Y. Akrami, G. Alestas, D. Aloni, L. Amendola, L. A. Anchordoqui *et al.*, Cosmology intertwined: A review of the particle physics astrophysics, and cosmology associated with the cosmological tensions and anomalies, J. High Energy Astrophys. **34**, 49 (2022).

[17] M. Chevallier and D. Polarski, Accelerating universes with scaling dark matter, Int. J. Mod. Phys. D **10**, 213 (2001).

[18] E. V. Linder, Exploring the Expansion History of the Universe, Phys. Rev. Lett. **90**, 091301 (2003).

[19] C. Wetterich, Cosmology and the fate of dilatation symmetry, Nucl. Phys. **B302**, 668 (1988).

[20] B. Ratra and P. J. E. Peebles, Cosmological consequences of a rolling scalar field, Phys. Rev. D **37**, 3406 (1988).

[21] W. Hu and I. Sawicki, Models of $f(R)$ cosmic acceleration that evade solar system tests, Phys. Rev. D **76**, 064004 (2007).

[22] J. Simon, L. Verde, and R. Jimenez, Constraints on the redshift dependence of the dark energy potential, Phys. Rev. D **71**, 123001 (2005).

[23] 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.

[24] 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.

[25] 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).

[26] 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).

[27] 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.

[28] D. M. Scolnic, D. O. Jones, A. Rest, Y. C. Pan, R. Chornock, R. J. Foley, M. E. Huber, R. Kessler, G. 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).

[29] A. J. Ross, L. Samushia, C. Howlett, W. J. Percival, A. Burden, and M. 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).

[30] E. A. Kazin, J. Koda, C. Blake, N. Padmanabhan, S. Brough, M. Colless, C. Contreras, W. Couch, S. Croom, D. J. Croton *et al.*, The WiggleZ dark energy survey: Improved distance measurements to $z = 1$ with reconstruction of the baryonic acoustic feature, Mon. Not. R. Astron. Soc. **441**, 3524 (2014).

[31] M. Ata, F. Baumgarten, J. Bautista, F. Beutler, D. Bizyaev, M. R. Blanton, J. A. Blazek, A. S. Bolton, J. Brinkmann, J. 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 (2017).

[32] T. M. C. Abbott, F. B. Abdalla, A. Alarcon, S. Allam, F. Andrade-Oliveira, J. Annis, S. Avila, M. Banerji, N. Banik K. Bechtol *et al.*, Dark energy survey year 1 results: Measurement of the baryon acoustic oscillation scale in the distribution of galaxies to redshift 1, Mon. Not. R. Astron. Soc. **483**, 4866 (2018).

[33] S. Alam, M. Ata, S. Bailey, F. Beutler, D. Bizyaev, J. A. Blazek, A. S. Bolton, J. R. Brownstein, A. Burden, C.-H. 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).

[34] J. E. Bautista, R. Paviot, M. Vargas Magaña, S. de la Torre, S. Fromenteau, H. Gil-Marín, A. J. Ross, E. Burtin, K. S. Dawson, J. Hou *et al.*, The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: Measurement of the BAO and growth rate of structure of the luminous red galaxy sample from the anisotropic correlation function between redshifts 0.6 and 1, Mon. Not. R. Astron. Soc. **500**, 736 (2020).

[35] R. Neveux, E. Burtin, A. de Mattia, A. Smith, A. J. Ross, J. Hou, J. Bautista, J. Brinkmann, C.-H. Chuang,

K. S. Dawson *et al.*, The completed SDSS-IV extended Baryon Oscillation Spectroscopic Survey: BAO and RSD measurements from the anisotropic power spectrum of the quasar sample between redshift 0.8 and 2.2, Mon. Not. R. Astron. Soc. **499**, 210 (2020).

[36] Julian E. Bautista, Nicolás G. Busca, Julien Guy, James Rich, Michael Blomqvist, Hélion du Mas des Bourboux, Matthew M. Pieri, Andreu Font-Ribera, Stephen Bailey, Timothée Delubac *et al.*, Measurement of baryon acoustic oscillation correlations at z with SDSS DR12 Lyα-forests, Astron. Astrophys. **603**, A12 (2017).

[37] Hélion du Mas des Bourboux, Jean-Marc Le Goff, Michael Blomqvist, Nicolás G. Busca, Julien Guy, James Rich, Christophe Yèche, Julian E. Bautista, Étienne Burtin, Kyle S. Dawson *et al.*, Baryon acoustic oscillations from the complete SDSS-III Lyα-quasar cross-correlation function at $z = 2.4$, Astron. Astrophys. **608**, A130 (2017).

[38] R. R. Caldwell, R. Dave, and P. J. Steinhardt, Cosmological Imprint of an Energy Component with General Equation of State, Phys. Rev. Lett. **80**, 1582 (1998).

[39] I. Zlatev, L. Wang, and P. J. Steinhardt, Quintessence, Cosmic Coincidence, and the Cosmological Constant, Phys. Rev. Lett. **82**, 896 (1999).

[40] C. Armendariz-Picon, V. Mukhanov, and P. J. Steinhardt, Dynamical Solution to the Problem of a Small Cosmological Constant and Late-Time Cosmic Acceleration, Phys. Rev. Lett. **85**, 4438 (2000).

[41] E. J. Copeland, A. R. Liddle, and D. Wands, Exponential potentials and cosmological scaling solutions, Phys. Rev. D **57**, 4686 (1998).

[42] T. Clifton, P. G. Ferreira, A. Padilla, and C. Skordis, Modified gravity and cosmology, Phys. Rep. **513**, 1 (2012).

[43] H. A. Buchdahl, Non-linear Lagrangians and cosmological theory, Mon. Not. R. Astron. Soc. **150**, 1 (1970).

[44] P. Brax, C. van de Bruck, A.-C. Davis, and D. J. Shaw, $f(R)$ gravity and chameleon theories, Phys. Rev. D **78**, 104021 (2008).

[45] L. Hui, A. Nicolis, and C. W. Stubbs, Equivalence principle implications of modified gravity models, Phys. Rev. D **80**, 104002 (2009).

[46] S. Kandhai and P. K. S. Dunsby, Cosmological dynamics of viable $f(R)$ theories of gravity, arXiv:1511.00101.

[47] S. Liu, X. Huang, and P. Protopapas, Evaluating error bound for physics-informed neural networks on linear dynamical systems, arXiv:2207.01114.

[48] M. Mattheakis, D. Sondak, A. S. Dogra, and P. Protopapas, Hamiltonian neural networks for solving equations of motion, Phys. Rev. E **105**, 065305 (2022).

[49] 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).

[50] 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).

[51] R. D'Agostino and R. C. Nunes, Measurements of H_0 in modified gravity theories: The role of lensed quasars in the late-time universe, Phys. Rev. D **101**, 103505 (2020).

[52] C. R. Farrugia, J. Sultana, and J. Mifsud, Spatial curvature in $f(R)$ gravity, Phys. Rev. D **104**, 123503 (2021).

[53] O. Graf, P. Flores, P. Protopapas, and K. Pichara, Uncertainty quantification in neural differential equations, arXiv:2111.04207.

[54] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevD.107.063523 for a description of the Runge-Kutta 5(4) algorithm and an estimation of its computational cost measured in floating point operations.

[55] R. Jimenez and A. Loeb, Constraining cosmological parameters based on relative galaxy ages, Astrophys. J. **573**, 37 (2002).

[56] R. Tripp, A two-parameter luminosity correction for Type IA supernovae, Astron. Astrophys. **331**, 815 (1998).

[57] M. Betoule, R. Kessler, J. Guy, J. Mosher, D. Hardin, R. Biswas, P. Astier, P. El-Hage, M. Konig, S. Kuhlmann *et al.*, Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples, Astron. Astrophys. **568**, A22 (2014).

[58] R. Kessler, J. P. Bernstein, D. Cinabro, B. Dilday, J. A. Frieman, S. Jha, S. Kuhlmann, G. Miknaitis, M. Sako, M. Taylor, and J. Vanderplas, SNANA: A public software package for supernova analysis, Publ. Astron. Soc. Pac. **121**, 1028 (2009).

[59] 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.

[60] J. Silk, Cosmic black-body radiation and galaxy formation, Astrophys. J. **151**, 459 (1968).

[61] P. J. E. Peebles and J. T. Yu, Primeval adiabatic perturbation in an expanding universe, Astrophys. J. **162**, 815 (1970).

[62] R. A. Sunyaev and Y. B. Zeldovich, Small-scale fluctuations of relic radiation, Astrophys. Space Sci. **7**, 3 (1970).

[63] D. J. Eisenstein, I. Zehavi, D. W. Hogg, R. Scoccimarro, M. R. Blanton, R. C. Nichol, R. Scranton, H.-J. Seo, M. Tegmark, Z. Zheng *et al.*, Detection of the baryon acoustic peak in the large-scale correlation function of SDSS luminous red galaxies, Astrophys. J. **633**, 560 (2005).

[64] R. Lazkoz, M. Ortiz-Baños, and V. Salzano, $f(R)$ gravity modifications: From the action to the data, Eur. Phys. J. C **78**, 213 (2018).

[65] D. J. Eisenstein and W. Hu, Baryonic features in the matter transfer function, Astrophys. J. **496**, 605 (1998).

[66] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, emcee: The MCMC Hammer, Publ. Astron. Soc. Pac. **125**, 306 (2013).

[67] J. Goodman and J. Weare, Ensemble samplers with affine invariance, Commun. Appl. Math. Comput. Sci. **5**, 65 (2010).

[68] A. Lewis, GetDist: A Python package for analysing Monte Carlo samples, arXiv:1910.13970.

[69] V. Motta, M. A. García-Aspeitia, A. Hernández-Almada, J. Magaña, and T. Verdugo, Taxonomy of dark energy models, Universe **7**, 163 (2021).

[70] Y. Akrami, R. Kallosh, A. Linde, and V. Vardanyan, The landscape, the swampland and the era of precision cosmology, Fortschr. Phys. **67**, 1800075 (2019).

[71] F. Tosone, B. S. Haridasu, V. V. Luković, and N. Vittorio, Constraints on field flows of quintessence dark energy, Phys. Rev. D **99**, 043503 (2019).

[72] A. G. Riess, S. Casertano, W. Yuan, L. Macri, J. Anderson, J. W. MacKenty, J. B. Bowers, K. I. Clubb, A. V. Filippenko,

D. O. Jones, and B. E. Tucker, New parallaxes of galactic cepheids from spatially scanning the Hubble space telescope: Implications for the Hubble constant, Astrophys. J. **855**, 136 (2018).

[73] K. C. Wong, S. H. Suyu, G. C.-F. Chen, C. E. Rusu, M. Millon, D. Sluse, V. Bonvin, C. D. Fassnacht, S. Taubenberger, M. W. Auger *et al.*, H0LiCOW—XIII. A 2.4 per cent measurement of H_0 from lensed quasars: 5.3σ tension between early- and late-universe probes, Mon. Not. R. Astron. Soc. **498**, 1420 (2019).

[74] J. Dormand and P. Prince, A family of embedded Runge-Kutta formulae, J. Comput. Appl. Math. **6**, 19 (1980).

[75] L. F. Shampine, Some practical Runge-Kutta formulas, Math. Comput. **46**, 135 (1986).

[76] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[77] 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).

[78] M. F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Netw. **6**, 525 (1993).