Improved Uncertainty Quantification in Physics-Informed Neural Networks Using Error Bounds and Solution Bundles

Pablo Flores[1] **Olga Graf**[2] **Pavlos Protopapas**[3] **Karim Pichara**[1]

[1]Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Santiago, Chile
[2]Department of Computer Science, University of Tübingen, Germany
[3]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA

Abstract

Physics-Informed Neural Networks (PINNs) have been widely used to obtain solutions to various physical phenomena modeled as Differential Equations. As PINNs are not naturally equipped with mechanisms for Uncertainty Quantification, some work has been done to quantify the different uncertainties that arise when dealing with PINNs. In this paper, we use a two-step procedure to train Bayesian Neural Networks that provide uncertainties over the solutions to differential equation systems provided by PINNs. We use available error bounds over PINNs to formulate a heteroscedastic variance that improves the uncertainty estimation. Furthermore, we solve forward problems and utilize the obtained uncertainties when doing parameter estimation in inverse problems in cosmology.

1 INTRODUCTION

Physics-Informed Neural Networks (PINNs), first proposed by Lagaris et al. [1997], solve differential equations (DEs) by embedding the physics of the problem into the network, eliminating the need for extra data. PINNs offer advantages over traditional solvers: they are continuous, differentiable, and parallelizable, allowing them to bypass the need for previous time steps. However, challenges remain in computing solution errors or error bounds, with ongoing research [Liu et al., 2022, 2023, De Ryck and Mishra, 2022b]. Specific equations like Navier-Stokes and Elasticity are also under study [De Ryck and Mishra, 2022a, De Ryck et al., 2022, Guo and Haghighat, 2022].

Since their introduction, PINNs have rapidly gained attention, being applied to heat transfer [Cai et al., 2021b], wave equations [Rasht-Behesht et al., 2022], and fluid mechanics [Jin et al., 2021, Cai et al., 2021a, Mao et al., 2020]. Efforts to address failure modes and optimization issues are ongoing [Nabian et al., 2021, Steger et al., 2022, Krishnapriyan et al., 2021, Daw et al., 2022], with interest in Bayesian PINNs for uncertainty quantification (UQ) [Yang et al., 2020, Linka et al., 2022, Graf et al., 2021, Psaros et al., 2023].

The use of NNs for solving ODEs and PDEs was pioneered by Lagaris et al. [1997] and later advanced by Raissi et al. [2019], who introduced PINNs for forward and inverse problems. Forward problems solve for the solution given the equation and boundary conditions, while inverse problems estimate unknown parameters. Inverse problems were implemented by Raissi et al. [2019] by treating these parameters as trainable variables.

Although effective, this approach lacks a robust mechanism for uncertainty quantification, which Bayesian Neural Networks (BNN) address by introducing distributions over network weights. However, simply applying Bayesian methods does not fully leverage the available information about solution accuracy. This is where Solution Bundles [Flamant et al., 2020] come into play. Solution Bundles enable statistical analysis over multiple possible solutions, providing a more comprehensive view of uncertainty in both the solution and the equation parameters.

In cosmology, DEs aim to explain the universe's expansion. Testing new models against observations typically involves statistical analysis to determine parameter bounds. By combining Solution Bundles with BNNs, we enhance uncertainty quantification, offering a more reliable method for parameter estimation. Our contributions include:

- Introducing error-bound-based heteroscedastic variance for better uncertainty quality.
- Solving forward problems for cosmological equations while quantifying uncertainties.
- Applying Solution Bundles to solve inverse problems for parameter estimation.

2 BACKGROUND

2.1 PROBLEM FORMULATION

We adopt a slightly different formulation from Psaros et al. [2022]. The DEs we will work with can be defined as follows:

$$\mathcal{F}_\lambda[u(\mathbf{x})] = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{1}$$

$$\mathcal{B}_\lambda[u(\mathbf{x})] = b(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{2}$$

where $\mathbf{x}$ is the space-time coordinate, Ω is a bounded domain with boundary Γ, $f(\mathbf{x})$ is the source term, u is the solution of the system, $\mathcal{F}_\lambda$ is a differential operator, $\mathcal{B}_\lambda$ and $b(\mathbf{x})$ are the boundary conditions (BCs) operator and term, respectively, and λ denotes the parameters of the system.

In this paper, we will focus on problems where the operators $\mathcal{F}_\lambda$, $\mathcal{B}_\lambda$ and terms $f(\mathbf{x})$, $b(\mathbf{x})$ are known. If λ is assumed to be known, the goal is to find the solution u, referred to as the *forward problem*. Conversely, if u is known and the aim is to estimate λ, then this is known as the *inverse problem*.

2.2 PHYSICS-INFORMED NEURAL NETWORKS

A *Physics-Informed Neural Network* uses a neural network, $u_\theta(\mathbf{x})$, to approximate the true solution $u(\mathbf{x})$ of a differential equations system. As discussed by Psaros et al. [2023], PINNs can be trained by minimizing a fitting dataset's Mean Squared Error (MSE). A fitting dataset $\mathcal{D} = \{\mathcal{D}_f, \mathcal{D}_b\}$ is composed of noisy observations of f, $\mathcal{D}_f = \{\mathbf{x}_i, f_i\}_{i=0}^{N_f}$, and noisy BCs data $\mathcal{D}_b = \{\mathbf{x}_i, b_i\}_{i=0}^{N_b}$

$$\mathcal{L}(\theta) = \frac{w_f}{N_f}\sum_{i=0}^{N_f}\|f_\theta(\mathbf{x}_i) - f_i\|_2^2 + \frac{w_b}{N_b}\sum_{i=0}^{N_b}\|b_\theta(\mathbf{x}_i) - b_i\|_2^2 \tag{3}$$

Where w_f and w_b are weighting constants. Psaros et al. [2023] call this setup the *Forward Deterministic PDE Problem*, they also describe *Mixed Deterministic PDE Problem* where the objective is to obtain solutions for u and λ, and the *Mixed Stochastic PDE Problem* which deals with stochastic PDEs.

In this work, we adopt a different optimization problem to solve the forward deterministic problem and address the inverse problem separately rather than solving both forward and inverse problems simultaneously.

2.3 SOLVING FORWARD PROBLEMS

We begin by defining the *residual* of a differential equation as

$$r_\theta(\mathbf{x}) = \mathcal{F}_\lambda[u_\theta(\mathbf{x})] - f(\mathbf{x}) \tag{4}$$

The computation of $\mathcal{F}_\lambda[u_\theta(\mathbf{x})]$ is easy to implement thanks to automatic differentiation provided by Deep Learning frameworks such as PyTorch [Paszke et al., 2019]. This formulation allows PINNs to be trained as a self-supervised network. Since any solution u^* to the differential equation satisfies $\mathcal{F}_\lambda[u^*(\mathbf{x})] - f(\mathbf{x}) = 0$, we train the network to minimize the square of the residual

$$\min_\theta \frac{1}{N_r}\sum_i^{N_r} r_\theta^2(\mathbf{x}_i). \tag{5}$$

Usually, $\mathbf{x}_i$ are sampled from Ω with a uniform distribution or by taking an equally-spaced subset. It is important to note that we do not deal with noisy data, unlike the work by Raissi et al. [2019].

2.4 ENFORCING BOUNDARY CONDITIONS

While adding a term for BCs in the loss (see Eq. (3)) when dealing with data is a good way to incorporate such knowledge, there is no guarantee that the conditions will be satisfied. When BCs are known rather than observed in the data, we can use the transformation introduced by Lagaris et al. [1997] that enforces the BCs to be always satisfied. This is achieved by writing the approximate solution as a sum of two terms:

$$\tilde{u}_\theta(\mathbf{x}) := A(\mathbf{x}) + F(\mathbf{x}, u_\theta(\mathbf{x})) \tag{6}$$

where A does not depend on the network parameters θ and it satisfies the BCs. Since we need $\tilde{u}_\theta$ to satisfy BCs, F is constructed so it does not contribute to them. This transformation is also used by Graf et al. [2021], Chen et al. [2020].

One-dimensional Initial Value Problem Given an initial condition $u_0 = u(t_0)$, we consider a transformation

$$\tilde{u}_\theta(t) := u_0 + (1 - e^{-(t-t_0)})u_\theta(t) \tag{7}$$

In Appendix A of the Supplementary Material we show the enforcing of two-dimensional Dirichlet BCs. Similar transformations can be defined for Neumann and mixed BCs.

2.5 SOLUTION BUNDLES

Solution Bundles [Flamant et al., 2020] extend PINNs by allowing the network to take equation parameters $\lambda \in \Lambda$ as inputs. This modification allows the network to approximate a variety of solutions to a parameterized differential equation without the need to retrain for each value of λ.

In Section 2.1 we considered a unique value for λ and one BC term $b(\mathbf{x})$. When working with Solution Bundles instead, we have subsets for the equation parameters and BCs.

$$\lambda \in \Lambda \subset \mathbb{R}^p$$
$$b(\mathbf{x}) \in B(\mathbf{x}) \subset \mathbb{R}^n$$

Λ, $B(\mathbf{x})$ are such subsets that will be used to train the NN. p and n are the dimensionality of the equation parameters and the system's state variable, respectively. The transformation described in Section 2.4 can also be used for Solution Bundles. Eq. (6) turns into:

$$\tilde{u}_\theta(\mathbf{x}, \lambda) := A(\mathbf{x}, \lambda) + F(\mathbf{x}, u_\theta(\mathbf{x}, \lambda)) \tag{8}$$

2.5.1 Training Solution Bundles

Flamant et al. [2020] proposed a weighting function for the residual loss (Eq. (5)) to influence how the approximation error is distributed across the training region. However, in this work, for simplicity, we stick to the unweighted residual loss

$$\tilde{r}_\theta := \mathcal{F}_\lambda[u_\theta(\mathbf{x}, \lambda)] - f(\mathbf{x}) \tag{9}$$

$$\min_\theta \sum_i^N \sum_j^M \tilde{r}_\theta^2(\mathbf{x}_i, \lambda_j). \tag{10}$$

where we have redefined r_θ from Eq. (4) for the Solution Bundle case as $\tilde{r}_\theta$.

2.5.2 Solution Bundles for Solving Inverse Problems

So far we have explained how to solve forward problems, that is, finding a solution to a differential equations system. As we described, for the case of PINNs, the solving step is an optimization problem.

On the other hand, inverse problems aim to find a differential system that best describes some collected data. For this, we assume the differential system can explain the phenomena observed and we seek to estimate the system's parameters for a given dataset.

An effective approach to addressing the parameter estimation problem involves statistical analysis, specifically through the application of Bayesian methods. This approach necessitates multiple computations of the system's solution, corresponding to each parameter value. Traditional numerical methods require the discretization and integration process to be performed for each of these solutions. In contrast, Solution Bundles eliminate the need for retraining for each parameter value, thereby expediting the computational process. A comprehensive explanation of the probabilistic setup for Bayesian parameter estimation is provided in Section 5.

2.6 ERROR BOUNDS FOR PINNS

Good quality uncertainties should correlate with the true error of a solution. Since the true error is not accessible, we use error bounds in Section 4.2 to improve the uncertainty in the Bayesian NNs.

In [Liu et al., 2022, 2023], the authors present algorithms for computing error bounds on PINNs. These bounds apply to linear ODEs, systems of linear ODEs, non-linear ODEs in the form ϵv^k[1], as well as certain types of PDEs. These algorithms are independent of the NN architecture and depend solely on the structure of the equation as defined in Eq. (1) and the residuals of the DE.

For the Solution Bundle setup, the network error is denoted as $\eta(\mathbf{x}, \lambda) := u(\mathbf{x}, \lambda) - \tilde{u}_\theta(\mathbf{x}, \lambda)$, and the error bound is represented by a scalar function $\mathbb{B}$ such that

$$\|\eta(\mathbf{x}, \lambda)\| \leq \mathbb{B}(\mathbf{x}, \lambda) \tag{11}$$

In this work we use the error bounds developed for first-order linear ODEs with nonconstant coefficient. In Appendix B of the Supplementary Material provide the algorithm for its computation and for first-order linear ODEs with constant coefficient. Appendix B.3 and Algorithm 1 describe how obtain tight bounds.

2.7 BAYESIAN NEURAL NETWORKS

To quantify uncertainty, we adopt a Bayesian perspective on the neural networks. We do this by viewing the neural network as a probabilistic model $p(y|x, \theta)$ and placing a prior distribution over its parameters $p(\theta)$. Using Bayes' theorem, we can get the posterior distribution of θ:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \tag{12}$$

$p(\mathcal{D}|\theta)$ is the likelihood distribution over a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$.

The posterior distribution allows us to make predictions about unseen data by taking expectations. Consider a new data point $\hat{x}$, we can obtain the probability of the output y being $\hat{y}$ as:

$$p(\hat{y}|\hat{x}, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\hat{y}|\hat{x}, \theta)] = \int_\Theta p(\hat{y}|\hat{x}, \theta)p(\theta|\mathcal{D})d\theta \tag{13}$$

Usually, the integral in Eq. (13) is analytically intractable, and we have to resort to Monte Carlo (MC) approximations that can be computed as:

$$p(\hat{y}|\hat{x}, \mathcal{D}) \approx \sum_{i=1}^M p(\hat{y}|\hat{x}, \theta_i), \text{ where } \theta_i \sim p(\theta|\mathcal{D}) \tag{14}$$

3 SHORTCOMINGS OF RESIDUAL-BASED UQ METHODS IN PINNS

A direct application of Bayesian Neural Networks to PINNs is straightforward by placing the likelihood over the resid-

[1] Here, v is a variable and $|\epsilon| \ll 1$.

uals r of the PINN solution given a coordinate point $\mathbf{x}$ i.e. $p(r|\mathbf{x}, \theta)$. For more details see Section 4.3.

However, evaluating a PINN solution based only on the residual loss Eq. (5) can be misleading. A low residual at $\mathbf{x}$ does not guarantee a low solution error at $\mathbf{x}$. In Control Theory, for instance, an Integral Controller drives steady-state error to zero, yet errors may still exist:

$$u(t) = k_i \int_0^t e(\tau) d\tau \tag{15}$$

where u is the control input, $e(\tau) := r(\tau) - y(\tau)$ the error, r the reference, y the system output, and k_i the integral gain [Åström and Murray, 2008]. For a DE $\frac{\mathrm{d}u}{\mathrm{d}t} = f(t)$ with IC $u(t_0) = u_0$, the approximation error $\hat{u}(\tau) - u(\tau)$ can be expressed as:

$$\hat{u}(\tau) - u(\tau) = \int_{t_0}^{\tau} R(t) dt \tag{16}$$

where $R(t) := \frac{\mathrm{d}\hat{u}}{\mathrm{d}t} - f(t)$, given the approximation satisfies the ICs. Proof of Eq. (16) is in Proposition F.1 of the Supplementary Material. Eq. (16) suggests that the residual at some point $\mathbf{x}$ is not enough to characterize the solution error at the same point $\mathbf{x}$.

These challenges, along with other *Failure Modes* of PINNs, remain an active area of research [Krishnapriyan et al., 2021, Wang et al., 2022, Penwarden et al., 2023]. Notably, Wang et al. [2022] identify an implicit bias in the PINN framework that can severely violate the temporal causality of dynamical systems. As a result, residual minimization at a given time t_i may occur even when predictions at earlier times are inaccurate.

In Fig. 1 we see how a PINN solution has near maximum and minimum error for the same residual value. This illustrates the decoupling of residuals from solution errors, which can be attributed to a violation of temporal causality. Furthermore, Figs. 13 to 16 show how there is no clear correlation between residuals and solution errors. These figures were obtained by training a deterministic PINN on a cosmological model, the construction details are provided in Appendix I.

Our experiments demonstrate that the baseline method produces uncalibrated predictive distributions and, in some cases, fails to approximate the true solutions. To address these limitations, we propose a two-step approach that incorporates error bounds into the solutions, leading to predictive distributions with improved calibration.

4 TWO-STEP BAYESIAN PINNS

We use a two-step approach to obtain uncertainties in the solutions of equations. In the first step, we train a PINN

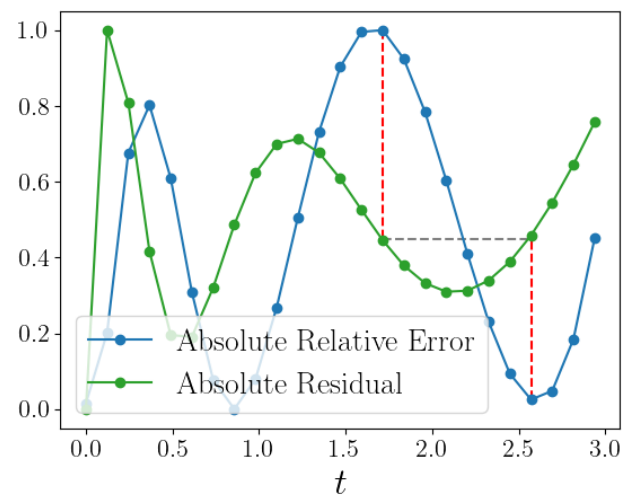

Figure 1: Normalized Absolute Residual vs Normalized Absolute Relative Error of a Deterministic NN Solution for CPL Model.

as a Solution Bundle that we refer to as the *deterministic network*. We denote this network as $u_{\theta_{\text{det}}} : \Omega, \Lambda \to \mathbb{R}$ and θ^*_{det} as the parameters resulting after training.

Note that as shown in Eq. (10), this training step is carried out without the use of any data, but rather sampling from the network domain ($\Omega \times \Lambda$) and minimizing the network's residuals. Fig. 2 shows a diagram of the entire training process.

4.1 BAYESIAN NEURAL NETWORK TRAINING

In the second step, we use the outputs of $u_{\theta^*_{\text{det}}}$ as targets to train a Bayesian Neural Network.

We construct a dataset by taking the space-time coordinates and equation parameters as independent variables, and the outputs of the deterministic net as dependent variables:

$$\mathcal{D} = \{(\mathbf{x}_i, \lambda_i, u_{\theta^*_{\text{det}}}(\mathbf{x}_i, \lambda_i)) \mid \mathbf{x}_i \in \Omega, \lambda_i \in \Lambda, u_{\theta^*_{\text{det}}}(\mathbf{x}_i, \lambda_i) \in \mathbb{R}\}_{i=1}^{N'}$$

We define a BNN u_θ by assigning distributions to the network parameters and dataset as it is done in [Graf et al., 2021]. For all three methods described in Section 2.7 we use a Gaussian prior:

$$\theta \sim \mathcal{N}(0, \sigma_{\text{prior}})$$

To formulate the likelihood, we assume a Gaussian additive noise on the *observations*[2]

$$u_{\theta^*_{\text{det}}}(\mathbf{x}, \lambda) = u_\theta(\mathbf{x}, \lambda) + \eta(\mathbf{x}, \lambda), \quad \text{where } \eta(\mathbf{x}, \lambda) \sim \mathcal{N}(0, \sigma_{\text{Like}}(\mathbf{x}, \lambda)) \tag{17}$$

[2]Here we call $u_{\theta^*_{\text{det}}}(\mathbf{x}, \lambda)$ *observations* even though they are not actual experimental observations.

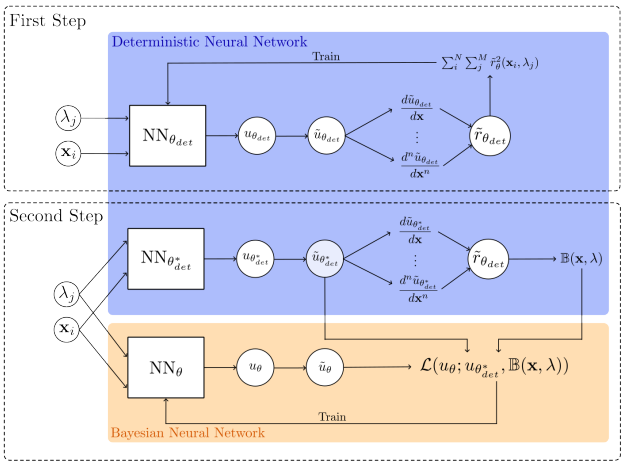

Figure 2: Diagram of Both Training Steps. In the Second Step θ^*_{det} are The Resulting Network Parameters From the First Step and $\mathcal{L}$ is The Likelihood Function.

This setup is often used in machine learning, but we employ it with a different interpretation; see Section 4.2. In this paper, we model the standard deviation $\sigma_{\text{Like}}(\mathbf{x}, \lambda)$ as known, but it can also be modeled as unknown. Given Eq. (17), the resulting likelihood is:

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N'} \mathcal{N}(u_\theta(\mathbf{x}, \lambda), \sigma_{\text{Like}}(\mathbf{x}, \lambda)) \tag{18}$$

We assume Gaussian noise in Eq. (17) due to its simplicity, tractability (see Section 4.1.1), and suitability as a starting point for evaluating our methodology. To assess this assumption, we analyze the distributional behavior of $u_{\theta_{\text{det}}}$, with details provided in Appendix I. Figs. 4 to 12 illustrate the solution distributions obtained by a PINN for various cosmological models. Although these distributions deviate from a perfect Gaussian, they are not significantly distant, suggesting that a Gaussian approximation is a reasonable initial choice.

4.1.1 Posterior Distribution Approximation Methods

To apply the MC approximation in Eq. (13), we must generate samples from the posterior distribution $p(\theta|\mathcal{D})$. Given the complexity of neural networks and the need for computational efficiency, we compare three methods: Neural Linear Models (NLMs), Bayes By Backpropagation (BBB), and Hamiltonian Monte Carlo (HMC). NLMs offer a lightweight approach by approximating the posterior in a linearized feature space, making them suitable for speed-critical applications. BBB uses variational inference to achieve greater accuracy while maintaining tractability for larger networks. HMC is the most advanced method, producing highly accurate samples at the expense of increased computational complexity. Comparing these techniques allows for an evaluation of the trade-offs between accuracy and computational feasibility.

Neural Linear Models (NLMs) NLM represent Bayesian linear regression using a neural network for the feature basis, with only the final layer parameters treated as stochastic. Using Gaussian prior and likelihood allows for tractable inference, leading to the posterior and predictive distributions given by:

$$\begin{aligned} p(u \mid \mathbf{x}, \lambda, \mathcal{D}) &= \mathcal{N}(\mu_{\text{NLM}}(\mathbf{x}, \lambda), \sigma_{\text{NLM}}(\mathbf{x}, \lambda)) \\ \mu_{\text{NLM}}(\mathbf{x}, \lambda) &= \Phi_\theta(\mathbf{x}, \lambda)\mu_{\text{post}} \\ \sigma_{\text{NLM}}(\mathbf{x}, \lambda) &= \sigma^2_{\text{Like}}(\mathbf{x}, \lambda) + \Phi_\theta(\mathbf{x}, \lambda)\Sigma_{\text{post}}\Phi_\theta^T(\mathbf{x}, \lambda) \end{aligned} \tag{19}$$

where μ_{post} and Σ_{post} are posterior parameters, and $\Phi_\theta(\mathbf{x}, \lambda)$ is the learned feature map. The details of μ_{post} and Σ_{post} computation can be found in Appendix C of the Supplementary Material.

Bayes By Backpropagation (BBB) BBB approximates the posterior of network parameters using variational inference (VI) and minimizes the KL divergence between the true posterior and a variational distribution $q(\theta|\rho)$. This results in an approximation $q(\theta|\rho)$ that is used to generate Monte Carlo samples for $p(\theta|\mathcal{D})$. We assume independent parameters under a mean-field approximation[Bishop, 2006], defining $q(\theta|\rho) = \prod_{i=1}^{N} q(\theta_i|\rho_i)$, using a Gaussian distribution for the variational posterior [Blundell et al., 2015].

Hamiltonian Monte Carlo (HMC) HMC employs a Metropolis-Hastings algorithm [Metropolis et al., 2004, Hastings, 1970] to draw samples from $p(\theta|\mathcal{D})$ by simulating particle movement through Hamiltonian dynamics. To enhance sampling efficiency, we utilize the No-U-Turn Sampler [Hoffman and Gelman].

4.2 IMPROVING PREDICTIVE UNCERTAINTY WITH ERROR BOUNDS

When learning from observations $\hat{y}$, the standard approach assumes these observations have an error ϵ:

$$\hat{y} = y_w(\mathbf{x}) + \epsilon \tag{20}$$

where $y_w(\mathbf{x})$ is a deterministic function parameterized by w. This formulation attributes all errors to the observations. However, we can introduce an additional error term η for model error:

$$\hat{y} = y_w(\mathbf{x}) + \eta + \epsilon \tag{21}$$

Separating model error η from observational error ϵ is typically challenging due to their additive nature. If we know there is no observational error, we can use Eq. (21) similarly

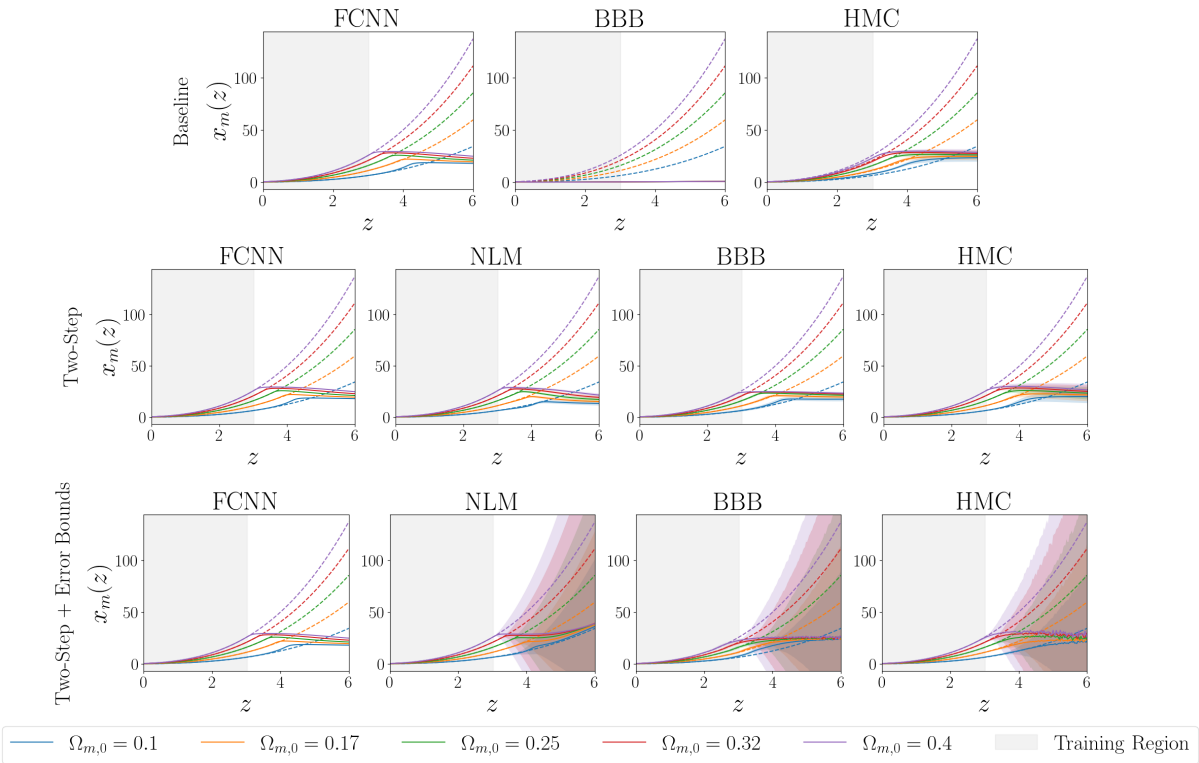

Figure 3: Examples of ΛCDM Bayesian Solutions Obtained Using the Bundle Solver. Analytic Solutions are Presented in Dotted Lines.

to Eq. (20). This is the case in Section 4.1, where the dataset has *observations* $u_{\theta^*_{\text{det}}}(\mathbf{x}, \lambda)$ free of noise:

$$u_{\theta^*_{\text{det}}}(\mathbf{x}, \lambda) = u_\theta(\mathbf{x}, \lambda) + \cancel{\epsilon} + \eta \tag{22}$$

However, the approximate solution $u_{\theta^*_{\text{det}}}$ may still have prediction errors. We model this error with a Gaussian distribution with standard deviation $\sigma_{\text{Like}}(\mathbf{x}, \lambda)$, which we treat as known in two ways:

1. **Homoscedastic:**

$$\sigma_{\text{Like}}(\mathbf{x}, \lambda) = \begin{cases} 0 & \mathbf{x} \in \Gamma, \\ \text{const} & \mathbf{x} \notin \Gamma \end{cases}$$

   We set σ_{Like} to be zero on the boundaries because we enforce BC always to be met.

2. **Error Bounds Based Heteroscedastic:**

$$\sigma_{\text{Like}}(\mathbf{x}, \lambda) = \mathbb{B}(\mathbf{x}, \lambda)$$

   This choice ensures we are taking into account the error made by $u_{\theta^*_{\text{det}}}$ in the predictive uncertainty.

4.3 BASELINE

In the literature, the standard approach to formulating the *forward* problem with Bayesian PINNs relies on noisy observations of $u(\mathbf{x})$, $f(\mathbf{x})$ and $b(\mathbf{x})$ [Psaros et al., 2022, Zou et al., 2023, Yang et al., 2020]. However, since our setup does not incorporate observations, we adopt as a baseline a Bayesian neural network with the likelihood defined over the residuals:

$$p(\mathcal{D}_r|\theta) = \prod_{i=1}^{N^r} \mathcal{N}(r_\theta(\mathbf{x}, \lambda), \sigma_r) \tag{23}$$

The training procedure follows the approach described in Section 4.1, with the exception that we could not implement NLM in this setting. This limitation arises because the residuals depend on the solution, thereby breaking the analytical tractability required for NLM.

5 PROBABILISTIC FORMULATION OF INVERSE PROBLEMS

The Bayesian framework can be used to estimate the parameters λ of a DE system. For a given set of observations $\mathcal{O} = \{(\mathbf{x}_i, \mu_i, \sigma_i) \mid \mathbf{x}_i \in \Omega, \mu_i \in \mathbb{R}, \sigma_i \in \mathbb{R}^+\}_{i=1}^{O}$, we seek to find the posterior distribution $p(\lambda|\mathcal{O})$ of the parameters. Here μ_i and σ_i are the mean and standard deviation of the observations at some point $\mathbf{x}_i$, respectively.

Assume we have a probability distribution over the solutions $p(u_\lambda(\mathbf{x})|\mathbf{x}, \lambda)$, then using Bayes' Theorem, the posterior

Table 1: Cosmology Equations Variables and Parameters. Parameters Marked With * are Inputs in the Bundle Solution, The Remainder are Needed to Compute the Hubble Parameter.

Equation	Type	Variables	Parameters
ΛCDM	Linear Equation	x_m	$\Omega^*_{m,0}, H_0$
CPL	Linear Equation	x_{DE}	$w^*_0, w^*_1, \Omega_{m,0}, H_0$
Quintessence	Non-linear System	x, y	$\lambda^*, \Omega^*_{m,0}, H_0$
HS	Non-linear System	x, y, v, Ω, r	$b^*, \Omega^*_{m,0}, H_0$

can be computed as $p(\lambda|\mathcal{O}) = p(O|\lambda)/p(\lambda)$. The likelihood $p(\mathcal{O}|\lambda)$ is obtained by marginalizing over the solutions as:

$$p(\mathbf{x}_i, \mu_i, \sigma_i|\lambda) = \int_{\mathcal{U}} p(\mathbf{x}_i, \mu_i, \sigma_i|u_\lambda(\mathbf{x}_i)) \cdot p(u_\lambda(\mathbf{x}_i)|\mathbf{x}_i, \lambda) du \tag{24}$$

$$\approx \frac{1}{M} \sum_{j=1}^{M} p(\mu_i, \sigma_i|u_\lambda^{(j)}(\mathbf{x}_i)) \tag{25}$$

$$p(\mathcal{O}|\lambda) = \prod_{i=1}^{O} p(\mathbf{x}_i, \mu_i, \sigma_i|\lambda) \tag{26}$$

where $u_\lambda^{(j)} \sim p(u_\lambda(\mathbf{x}_i)|\mathbf{x}_i, \lambda)$.

The distribution induced by FCNNs can be interpreted as a Delta distribution, where the function learned by the network outputs a specific value $u_{\theta_{\text{det}}}(\mathbf{x})$ for a given input $\mathbf{x}$. In contrast, BNNs naturally provide a distribution over solutions through their posterior predictive, thus Eq. (24) results in a marginalization over the posterior predictive distribution $p(u(\mathbf{x}_i, \lambda)|\mathbf{x}_i, \lambda, \mathcal{D})$.

We use a uniform distribution to aim for an uninformative prior for λ. In the same way Chantada et al. [2022] did, to define the likelihood $p(\mathcal{O}|\lambda)$, we assume the observations are normally distributed around the true solution, i.e., $\mu_i \sim \mathcal{N}(u(\mathbf{x}_i), \sigma_i)$.

Having defined the prior and likelihood, we can apply a sampling algorithm to approximate $p(\lambda|\mathcal{O})$. We use the `emcee` Python package [Foreman-Mackey et al., 2013] which implements the samplers introduced by Goodman and Weare [2010].

6 EXPERIMENTS ON COSMOLOGY MODELS

This section specifies the cosmological equations we used to test our methodology. These equations were solved with PINNs by Chantada et al. [2022]. For brevity, we describe the ΛCDM and Parametric Dark Energy models. The details of Quintessence and $f(R)$ gravity (or HS for the name of the authors Hu and Sawicki) are provided in Appendix G. However, Table 1 lists the variables and parameters of each cosmological model.

ΛCDM The equation and initial conditions are

$$\frac{dx_m}{dz} = \frac{3x_m}{1+z}$$
$$x_m(z=0) = \Omega_{m,0}$$

where $\Omega_{m,0}$ is a parameter. Having a solution to this equation, the Hubble parameter can be obtained as

$$H(z) = H_0\sqrt{x_m(z) + 1 - \Omega_{m,0}} \tag{27}$$

here H_0 is also a parameter.

Parametric Dark Energy We refer to this model as CPL for the names of the authors Chevallier, Polarski and Linder [Linder, 2003, Chevallier and Polarski, 2001]. The equation and initial conditions are

$$\frac{dx_{DE}}{dz} = \frac{3x_{DE}}{1+z}\left(1 + w_0 + \frac{w_1 z}{1+z}\right)$$
$$x_{DE}(z=0) = 1 - \Omega_{m,0}$$

where w_0, w_1 and $\Omega_{m,0}$ are parameters. Having a solution to this equation, the Hubble parameter can be obtained as

$$H(z) = H_0\sqrt{\Omega_{m,0}(1+z)^3 + x_{DE}(z)} \tag{28}$$

here H_0 is also a parameter.

For brevity, we provide the error bounds computation details in Appendix B of the Supplementary Material.

Forward Problems We trained $u_{\theta^*_{\text{det}}}$ using methods from [Chantada et al., 2022], then built a dataset $\mathcal{D}$ to train the BNNs. We compared homoscedastic and error-bounds-based heteroscedastic variance for ΛCDM and CPL equations, while providing homoscedastic variance results for Quintessence and HS. The networks were trained using regular PINNs and Solution Bundles, referred to as *Forward* and *Bundle*, respectively, with implementation details in Appendix H of the Supplementary Material. We provide the complete implementation in a code repository.[3]

We evaluated the NN solutions by calculating the Median Relative Error (MRE) against analytical solutions for ΛCDM and CPL, and numerical solutions for Quintessence and HS, using a Runge-Kutta method. We also computed the median residual, and miscalibration area (MA) [Chung et al., 2021] to assess uncertainty quality. Results are summarized in Table 2, with additional metrics in Tables 13 to 15.

For the CPL model, we utilized a reparameterization from [Chantada et al., 2022] that separates equation parameters from Bundle Network training, although this method is incompatible with Neural Linear Models (NLM).

[3] `https://github.com/ptflores1/improved-pinn-uq`

Table 2: Evaluation Metrics of All Bundle Networks and Equations in the Testing Region. Here we Use 2S for Two-step and EB for Error Bounds.

Equation	Method	Median RE	Median Residual	Miscal. Area
ΛCDM	FCNN	**0.001**	0.0	-
	BBB	0.978	0.147	0.491
	HMC	0.084	0.198	0.4
	NLM + 2S	0.018	8.157	0.183
	BBB + 2S	0.035	0.963	0.143
	HMC + 2S	0.004	0.197	0.123
	NLM + 2S + EB	0.002	10.783	0.063
	BBB + 2S + EB	0.047	2.476	**0.05**
	HMC + 2S + EB	0.003	0.272	0.098
CPL	FCNN	**0.0**	0.001	-
	BBB	0.128	0.014	0.199
	HMC	**0.0**	0.01	0.317
	BBB + 2S	0.063	0.006	0.15
	HMC + 2S	0.004	0.005	0.255
	BBB + 2S + EB	0.033	0.003	0.177
	HMC + 2S + EB	0.011	0.004	**0.145**
Quint.	FCNN	0.007	0.0	-
	BBB	0.094	0.012	0.151
	HMC	**0.002**	0.005	0.147
	NLM + 2S	0.096	0.175	0.136
	BBB + 2S	0.12	0.011	0.119
	HMC + 2S	0.016	0.002	**0.048**
HS	FCNN	**0.001**	0.0	-
	BBB	0.393	0.083	0.449
	HMC	0.226	0.01	0.455
	NLM + 2S	0.259	1.666	**0.315**
	BBB + 2S	0.296	0.13	0.396
	HMC + 2S	0.286	0.345	0.486

Inverse Problems For the inverse problem, we used 30 measurements of the Hubble parameter H from the Cosmic Chronometers (CC) method [Simon et al., 2005, Stern et al., 2010, Moresco et al., 2012, Cong et al., 2014, Moresco, 2015, Moresco et al., 2016]. Each measurement includes a tuple $(z_i, H^{obs}, \sigma_{H^{obs}})$, indicating redshift, observed mean Hubble parameters, and their standard deviation. The CC dataset is available in Table 9 of the Supplementary Material, with Eq. (41) used as the likelihood of observations.

We estimated equation parameters using the Solution Bundles from the forward step and performed inference with the `emcee` package [Foreman-Mackey et al., 2013], running 32 chains for 10,000 steps, resulting in 320,000 samples per parameter. Tables 11 and 12 show the results and their concordance with values found in the literature, respectively.

7 DISCUSSION

The FCNN consistently achieves the lowest median RE and mean residual across all equations, outperforming Bayesian methods in terms of accuracy. This highlights a tradeoff between equipping PINNs with uncertainty quantification and maintaining accuracy. In this work, our primary focus is on uncertainty rather than pure accuracy, as a well-calibrated model with higher error is generally more desirable than an uncalibrated model with low error. Properly calibrated uncertainties enable a meaningful assessment of prediction reliability.

From Table 2, we observe that the baseline Bayesian methods—BBB and HMC with a residual likelihood—exhibit significantly higher errors compared to the deterministic network, particularly for the CPL and HS equations. This effect is especially pronounced for BBB, which fails to approximate the ΛCDM solution. Additionally, both methods exhibit high miscalibration areas, indicating that their uncertainty estimates are poorly calibrated.

Introducing our two-step (2S) method substantially improves calibration, as evidenced by the reduction in miscalibration areas across all cases. For instance, applying 2S to BBB in the ΛCDM equation reduces the miscalibration area from 0.491 to 0.143.

Further incorporating error bounds into the two-step method enhances uncertainty calibration even further. Notably, BBB + 2S + EB achieves the lowest miscalibration area (0.05) for the ΛCDM equation, while HMC + 2S + EB provides the best calibration in CPL (0.145). However, certain cases exhibit excessively large mean residuals, such as in CPL, where BBB + 2S + EB results in a residual of 9.58×10^8. This suggests that while error bounds improve calibration, they may introduce numerical instability in the metrics due to some samples' extremely large errors.

For the Quintessence equation, HMC + 2S achieves the best calibration (0.048) while maintaining a low median RE (0.016), demonstrating an effective balance between accuracy and uncertainty quantification. In contrast, for the HS equation, while the two-step method improves calibration, overall performance remains suboptimal, with high median RE and residuals persisting across all Bayesian approaches.

In Tables 3 and 4, we present results separately for the training and OOD regions. In the training region, the two-step approach consistently improves both accuracy and miscalibration, and the addition of error bounds further enhances calibration across most equations. For example, in the ΛCDM case, combining two steps with error bounds yields the lowest miscalibration area among all methods, indicating effective uncertainty correction when the model operates within its training distribution.

In the OOD region, results are more mixed. For ΛCDM, the combination of two steps and error bounds significantly reduces miscalibration. However, in the CPL model, applying two steps increases miscalibration for BBB but reduces it for HMC; adding error bounds reverses this behavior—improving BBB but slightly worsening HMC. As shown in Fig. 26, this is likely because FCNN closely tracks the ground truth in the OOD region, leading to narrow error bounds, while BBB and HMC tend to drift from the true function, which error bounds fail to capture accurately. For the more complex equations, Quintessence and HS, the benefits of the two-step method are less consistent, and no clear advantage is observed over the baseline approaches.

Table 3: Evaluation Metrics of All Bundle Networks and Equations in the Training Region. Here we Use 2S for Two-step and EB for Error Bounds.

Equation	Method	Median RE	Median Residual	Miscal. Area
ΛCDM	FCNN	**0.0**	0.0	-
	BBB	0.961	0.12	0.491
	HMC	0.041	0.07	0.358
	NLM + 2S	0.003	4.212	0.402
	BBB + 2S	0.008	0.238	0.269
	HMC + 2S	0.001	0.038	0.307
	NLM + 2S + EB	0.001	4.211	**0.016**
	BBB + 2S + EB	0.01	0.228	0.248
	HMC + 2S + EB	0.001	0.014	0.251
CPL	FCNN	**0.0**	0.001	-
	BBB	0.052	0.016	0.161
	HMC	**0.0**	0.009	0.482
	BBB + 2S	0.024	0.009	**0.085**
	HMC + 2S	0.001	0.008	0.436
	BBB + 2S + EB	0.007	0.005	0.14
	HMC + 2S + EB	0.001	0.008	0.159
Quint.	FCNN	0.003	0.0	-
	BBB	0.059	0.005	**0.09**
	HMC	**0.001**	0.003	0.279
	NLM + 2S	0.063	0.097	0.326
	BBB + 2S	0.047	0.002	0.165
	HMC + 2S	0.006	0.0	0.179
HS	FCNN	**0.0**	0.0	-
	BBB	0.373	0.162	0.426
	HMC	0.257	0.121	0.43
	NLM + 2S	**0.244**	0.201	**0.252**
	BBB + 2S	0.284	0.094	0.387
	HMC + 2S	0.279	0.111	0.427

Table 4: Evaluation Metrics of All Bundle Networks and Equations in the OOD Region. Here we Use 2S for Two-step and EB for Error Bounds.

Equation	Method	Median RE	Median Residual	Miscal. Area
ΛCDM	FCNN	0.374	0.027	-
	BBB	0.986	0.168	0.495
	HMC	0.316	11.979	0.444
	NLM + 2S	0.449	10.569	0.43
	BBB + 2S	0.363	12.672	0.495
	HMC + 2S	0.31	12.054	0.424
	NLM + 2S + EB	**0.287**	14.778	**0.113**
	BBB + 2S + EB	0.375	10.695	0.155
	HMC + 2S + EB	0.329	11.463	0.168
CPL	FCNN	**0.033**	0.0	-
	BBB	0.373	0.013	0.357
	HMC	0.082	0.012	0.152
	BBB + 2S	0.59	0.001	0.495
	HMC + 2S	0.275	0.001	**0.072**
	BBB + 2S + EB	0.958	0.001	0.216
	HMC + 2S + EB	0.648	0.001	0.194
Quint.	FCNN	0.022	0.016	-
	BBB	0.125	0.089	0.349
	HMC	**0.006**	0.011	**0.021**
	NLM + 2S	0.138	0.335	0.334
	BBB + 2S	0.221	0.13	0.377
	HMC + 2S	0.038	0.03	0.096
HS	FCNN	0.374	0.027	-
	BBB	0.374	0.161	0.427
	HMC	0.257	0.121	0.43
	NLM + 2S	6.999	0.592	**0.412**
	BBB + 2S	2.124	1.616	0.452
	HMC + 2S	0.885	0.984	0.487

In Table 11 we can see how the Bayesian methods provide tighter intervals than deterministic nets. This is due to observations being more likely under the distributions generated by Bayesian methods. These uncertainties can be propagated through the approximated solution to obtain solution distributions that account for the uncertainties in the equation parameters.

Table 12 shows the level of agreement of our results with the literature. Most of our results are in an agreement of 1σ or 2σ. ΛCDM is an exception, where we observe higher disagreement due to the tight bounds obtained and discrepancy in the means.

Overall, our results demonstrate that leveraging error bounds within a two-step framework significantly enhances uncertainty quantification, particularly in calibration. However, the observed tradeoff between calibration and numerical stability suggests the need for further refinement, especially for highly complex equations such as HS.

Among Bayesian approaches, HMC generally achieves superior accuracy and calibration, but its computational cost is substantial. BBB, on the other hand, performs well when combined with our two-step procedure, offering a more computationally efficient and flexible alternative. There is considerable potential for improving BBB by adopting more expressive variational posterior distributions. Finally, NLM also demonstrates strong performance while being the least expensive method, though its applicability is limited by the underlying distributional assumptions.

References

Yashar Akrami, Renata Kallosh, Andrei Linde, and Valeri Vardanyan. The landscape, the swampland and the era of precision cosmology. *Fortschritte der Physik*, 67 (1-2):1800075, January 2019. ISSN 0015-8208, 1521-3978. doi: 10.1002/prop.201800075. URL http://arxiv.org/abs/1808.09440. arXiv:1808.09440 [astro-ph, physics:gr-qc, physics:hep-ph, physics:hep-th, physics:math-ph].

C. Armendariz-Picon, V. Mukhanov, and Paul J. Steinhardt. Dynamical Solution to the Problem of a Small Cosmological Constant and Late-Time Cosmic Acceleration. *Physical Review Letters*, 85(21):4438–4441, November 2000. doi: 10.1103/PhysRevLett.85.4438. URL https://link.aps.org/doi/10.1103/PhysRevLett.85.4438. Publisher: American Physical Society.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming, Oc-

tober 2018. URL http://arxiv.org/abs/1810.09538. arXiv:1810.09538 [cs, stat].

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, July 2006. ISBN 978-0-387-31073-2.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *32nd International Conference on Machine Learning, ICML 2015*, 2:1613–1622, 2015. arXiv: 1505.05424 ISBN: 9781510810587.

William E Boyce and Richard C DiPrima. *Elementary differential equations and boundary value problems*. John Wiley & Sons, Chichester, England, 9 edition, October 2008.

H. A. Buchdahl. Non-Linear Lagrangians and Cosmological Theory. *Monthly Notices of the Royal Astronomical Society*, 150(1):1–8, September 1970. ISSN 0035-8711. doi: 10.1093/mnras/150.1.1. URL https://doi.org/10.1093/mnras/150.1.1.

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mechanica Sinica*, 37(12):1727–1738, December 2021a. ISSN 1614-3116. doi: 10.1007/s10409-021-01148-1. URL https://doi.org/10.1007/s10409-021-01148-1.

Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, 143(6), April 2021b. ISSN 0022-1481. doi: 10.1115/1.4050542. URL https://doi.org/10.1115/1.4050542.

R. R. Caldwell, Rahul Dave, and Paul J. Steinhardt. Cosmological Imprint of an Energy Component with General Equation of State. *Physical Review Letters*, 80(8):1582–1585, February 1998. doi: 10.1103/PhysRevLett.80.1582. URL https://link.aps.org/doi/10.1103/PhysRevLett.80.1582. Publisher: American Physical Society.

Augusto T. Chantada, Susana J. Landau, Pavlos Protopapas, Claudia G. Scóccola, and Cecilia Garraffo. Cosmological informed neural networks to solve the background dynamics of the Universe, May 2022. URL http://arxiv.org/abs/2205.02945. arXiv:2205.02945 [astro-ph, physics:gr-qc, physics:hep-ph].

Feiyu Chen, David Sondak, Pavlos Protopapas, Marios Mattheakis, Shuheng Liu, Devansh Agarwal, and Marco Di Giovanni. NeuroDiffEq: A Python package for solving differential equations with neural networks. *Journal of Open Source Software*, 5(46):1931, February 2020. ISSN 2475-9066. doi: 10.21105/joss.01931. URL https://joss.theoj.org/papers/10.21105/joss.01931.

Michel Chevallier and David Polarski. Accelerating universes with scaling dark matter. *International Journal of Modern Physics D*, 10(02):213–223, April 2001. ISSN 0218-2718. doi: 10.1142/S0218271801000822. URL https://www.worldscientific.com/doi/abs/10.1142/S0218271801000822. Publisher: World Scientific Publishing Co.

Youngseog Chung, Ian Char, Han Guo, Jeff Schneider, and Willie Neiswanger. Uncertainty Toolbox: an Open-Source Library for Assessing, Visualizing, and Improving Uncertainty Quantification, September 2021. URL http://arxiv.org/abs/2109.10254. arXiv:2109.10254 [cs, stat].

Timothy Clifton, Pedro G. Ferreira, Antonio Padilla, and Constantinos Skordis. Modified gravity and cosmology. *Physics Reports*, 513(1):1–189, March 2012. ISSN 0370-1573. doi: 10.1016/j.physrep.2012.01.001. URL https://www.sciencedirect.com/science/article/pii/S0370157312000105.

Zhang Cong, Zhang Han, Yuan Shuo, Liu Siqi, Zhang Tong-Jie, and Sun Yan-Chun. Four new observational H(z) data from luminous red galaxies in the Sloan Digital Sky Survey data release seven. *Research in Astronomy and Astrophysics*, 14(10):1221, October 2014. ISSN 1674-4527. doi: 10.1088/1674-4527/14/10/002. URL https://dx.doi.org/10.1088/1674-4527/14/10/002.

Edmund J. Copeland, Andrew R. Liddle, and David Wands. Exponential potentials and cosmological scaling solutions. *Physical Review D*, 57(8):4686–4690, April 1998. doi: 10.1103/PhysRevD.57.4686. URL https://link.aps.org/doi/10.1103/PhysRevD.57.4686. Publisher: American Physical Society.

Rocco D'Agostino and Rafael C. Nunes. Measurements of H_0 in modified gravity theories: The role of lensed quasars in the late-time Universe. *Physical Review D*, 101(10):103505, May 2020. ISSN 2470-0010, 2470-0029. doi: 10.1103/PhysRevD.101.103505. URL http://arxiv.org/abs/2002.06381. arXiv:2002.06381 [astro-ph, physics:gr-qc, physics:hep-ph].

Arka Daw, Jie Bu, Sifan Wang, Paris Perdikaris, and Anuj Karpatne. Mitigating Propagation Failures in PINNs using Evolutionary Sampling, October 2022. URL http://arxiv.org/abs/2207.02338. arXiv:2207.02338 [cs].

Tim De Ryck and Siddhartha Mishra. Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs. *Advances in Computational Mathematics*, 48(6):79, November 2022a. ISSN 1572-9044. doi: 10.1007/s10444-022-09985-9. URL https://doi.org/10.1007/s10444-022-09985-9.

Tim De Ryck and Siddhartha Mishra. Generic bounds on the approximation error for physics-informed (and) operator learning. May 2022b. URL https://openreview.net/forum?id=bF4eYy3LTR9.

Tim De Ryck, Ameya D. Jagtap, and Siddhartha Mishra. Error estimates for physics informed neural networks approximating the Navier-Stokes equations, March 2022. URL http://arxiv.org/abs/2203.09346. arXiv:2203.09346 [cs, math].

Cedric Flamant, Pavlos Protopapas, and David Sondak. Solving Differential Equations Using Neural Network Solution Bundles, June 2020. URL https://arxiv.org/abs/2006.14372v1.

Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. emcee: The MCMC Hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, February 2013. ISSN 1538-3873. doi: 10.1086/670067. URL https://iopscience.iop.org/article/10.1086/670067/meta. Publisher: IOP Publishing.

Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1): 65–80, January 2010. ISSN 2157-5452. doi: 10.2140/camcos.2010.5.65. URL https://msp.org/camcos/2010/5-1/p04.xhtml. Publisher: Mathematical Sciences Publishers.

Olga Graf, Pablo Flores, Pavlos Protopapas, and Karim Pichara. Uncertainty Quantification in Neural Differential Equations, November 2021. URL http://arxiv.org/abs/2111.04207. arXiv:2111.04207 [cs].

Mengwu Guo and Ehsan Haghighat. Energy-Based Error Bound of Physics-Informed Neural Network Solutions in Elasticity. *Journal of Engineering Mechanics*, 148(8):04022038, August 2022. ISSN 1943-7889. doi: 10.1061/(ASCE)EM.1943-7889.0002121. URL https://ascelibrary.org/doi/10.1061/%28ASCE%29EM.1943-7889.0002121. Publisher: American Society of Civil Engineers.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444. doi: 10.2307/2334940. URL https://www.jstor.org/stable/2334940. Publisher: [Oxford University Press, Biometrika Trust].

Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.

Wayne Hu and Ignacy Sawicki. Models of $f(R)$ cosmic acceleration that evade solar system tests. *Physical Review D*, 76(6):064004, September 2007. doi: 10.1103/PhysRevD.76.064004. URL https://link.aps.org/doi/10.1103/PhysRevD.76.064004. Publisher: American Physical Society.

Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426:109951, February 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.109951. URL https://www.sciencedirect.com/science/article/pii/S0021999120307257.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks, November 2021. URL http://arxiv.org/abs/2109.01050. arXiv:2109.01050 [cs].

I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, May 1997. doi: 10.1109/72.712178. URL http://arxiv.org/abs/physics/9705023. arXiv: physics/9705023v1.

Eric V. Linder. Exploring the Expansion History of the Universe. *Physical Review Letters*, 90(9): 091301, March 2003. doi: 10.1103/PhysRevLett.90.091301. URL https://link.aps.org/doi/10.1103/PhysRevLett.90.091301. Publisher: American Physical Society.

Kevin Linka, Amelie Schafer, Xuhui Meng, Zongren Zou, George Em Karniadakis, and Ellen Kuhl. Bayesian Physics-Informed Neural Networks for real-world nonlinear dynamical systems, May 2022. URL http://arxiv.org/abs/2205.08304. arXiv:2205.08304 [nlin].

Shuheng Liu, Xiyue Huang, and Pavlos Protopapas. Evaluating Error Bound for Physics-Informed Neural Networks on Linear Dynamical Systems, July 2022. URL http://arxiv.org/abs/2207.01114. arXiv:2207.01114 [cs, math].

Shuheng Liu, Xiyue Huang, and Pavlos Protopapas. *Residual-based error bound for physics-informed neural networks*. June 2023.

Zhiping Mao, Ameya D. Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, March 2020. ISSN 0045-7825. doi: 10.1016/j.cma.2019.112789. URL https://www.sciencedirect.com/science/article/pii/S0045782519306814.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, December 2004. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

M. Moresco, A. Cimatti, R. Jimenez, L. Pozzetti, G. Zamorani, M. Bolzonella, J. Dunlop, F. Lamareille, M. Mignoli, H. Pearce, P. Rosati, D. Stern, L. Verde, E. Zucca, C. M. Carollo, T. Contini, J.-P. Kneib, O. Le Fèvre, S. J. Lilly, V. Mainieri, A. Renzini, M. Scodeggio, I. Balestra, R. Gobat, R. McLure, S. Bardelli, A. Bongiorno, K. Caputi, O. Cucciati, S. de la Torre, L. de Ravel, P. Franzetti, B. Garilli, A. Iovino, P. Kampczyk, C. Knobel, K. Kovač, J.-F. Le Borgne, V. Le Brun, C. Maier, R. Pelló, Y. Peng, E. Perez-Montero, V. Presotto, J. D. Silverman, M. Tanaka, L. A. M. Tasca, L. Tresse, D. Vergani, O. Almaini, L. Barnes, R. Bordoloi, E. Bradshaw, A. Cappi, R. Chuter, M. Cirasuolo, G. Coppa, C. Diener, S. Foucaud, W. Hartley, M. Kamionkowski, A. M. Koekemoer, C. López-Sanjuan, H. J. McCracken, P. Nair, P. Oesch, A. Stanford, and N. Welikala. Improved constraints on the expansion rate of the Universe up to z 1.1 from the spectroscopic evolution of cosmic chronometers. *Journal of Cosmology and Astroparticle Physics*, 2012 (08):006, August 2012. ISSN 1475-7516. doi: 10.1088/1475-7516/2012/08/006. URL https://dx.doi.org/10.1088/1475-7516/2012/08/006.

Michele Moresco. Raising the bar: new constraints on the Hubble parameter with cosmic chronometers at z 2. *Monthly Notices of the Royal Astronomical Society: Letters*, 450(1):L16–L20, June 2015. ISSN 1745-3925. doi: 10.1093/mnrasl/slv037. URL https://doi.org/10.1093/mnrasl/slv037.

Michele Moresco, Lucia Pozzetti, Andrea Cimatti, Raul Jimenez, Claudia Maraston, Licia Verde, Daniel Thomas, Annalisa Citro, Rita Tojeiro, and David Wilkinson. A 6% measurement of the Hubble parameter at z 0.45: direct evidence of the epoch of cosmic re-acceleration. *Journal of Cosmology and Astroparticle Physics*, 2016 (05):014, May 2016. ISSN 1475-7516. doi: 10.1088/1475-7516/2016/05/014. URL https://dx.doi.org/10.1088/1475-7516/2016/05/014.

V. Motta, Miguel A. García-Aspeitia, A. Hernández-Almada, J. Magaña, and Tomás Verdugo. Taxonomy of Dark Energy Models, April 2021. URL http://arxiv.org/abs/2104.04642. arXiv:2104.04642 [astro-ph, physics:gr-qc].

Mohammad Amin Nabian, Rini Jasmine Gladstone, and Hadi Meidani. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021. ISSN 1467-8667. doi: 10.1111/mice.12685. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12685. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12685.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs, stat].

Michael Penwarden, Ameya D. Jagtap, Shandian Zhe, George Em Karniadakis, and Robert M. Kirby. A unified scalable framework for causal sweeping strategies for Physics-Informed Neural Networks (PINNs) and their temporal decompositions. *Journal of Computational Physics*, 493:112464, November 2023. ISSN 00219991. doi: 10.1016/j.jcp.2023.112464. URL http://arxiv.org/abs/2302.14227. arXiv:2302.14227 [physics].

Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons. 2022. doi: 10.48550/arxiv.2201.07766. arXiv: 2201.07766.

Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, March 2023. ISSN 0021-9991. doi: 10.1016/j.jcp.2022.111902. URL https://www.sciencedirect.com/science/article/pii/S0021999122009652.

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. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Majid Rasht-Behesht, Christian Huber, Khemraj Shukla, and George Em Karniadakis. Physics-Informed

Neural Networks (PINNs) for Wave Propagation and Full Waveform Inversions. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023120, 2022. ISSN 2169-9356. doi: 10.1029/2021JB023120. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2021JB023120. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021JB023120.

Joan Simon, Licia Verde, and Raul Jimenez. Constraints on the redshift dependence of the dark energy potential. *Physical Review D*, 71(12): 123001, June 2005. doi: 10.1103/PhysRevD.71.123001. URL https://link.aps.org/doi/10.1103/PhysRevD.71.123001. Publisher: American Physical Society.

Sophie Steger, Franz M. Rohrhofer, and Bernhard C. Geiger. How PINNs cheat: Predicting chaotic motion of a double pendulum. October 2022. URL https://openreview.net/forum?id=shUbBca03f.

Daniel Stern, Raul Jimenez, Licia Verde, Marc Kamionkowski, and S. Adam Stanford. Cosmic chronometers: constraining the equation of state of dark energy. I: H(z) measurements. *Journal of Cosmology and Astroparticle Physics*, 2010(02):008, February 2010. ISSN 1475-7516. doi: 10.1088/1475-7516/2010/02/008. URL https://dx.doi.org/10.1088/1475-7516/2010/02/008.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks, March 2022. URL http://arxiv.org/abs/2203.07404. arXiv:2203.07404 [cs].

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data. March 2020. doi: 10.1016/j.jcp.2020.109913. URL http://arxiv.org/abs/2003.06097. arXiv:2003.06097.

Ivaylo Zlatev, Limin Wang, and Paul J. Steinhardt. Quintessence, Cosmic Coincidence, and the Cosmological Constant. *Physical Review Letters*, 82(5): 896–899, February 1999. doi: 10.1103/PhysRevLett.82.896. URL https://link.aps.org/doi/10.1103/PhysRevLett.82.896. Publisher: American Physical Society.

Zongren Zou, Xuhui Meng, and George Em Karniadakis. Uncertainty quantification for noisy inputs-outputs in physics-informed neural networks and neural operators, November 2023. URL http://arxiv.org/abs/2311.11262. arXiv:2311.11262 [physics].

Karl J. Åström and Richard M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton Univ. Press, Princeton, NJ, 2008. ISBN 978-0-691-13576-2.

Supplementary Material

Pablo Flores[1] **Olga Graf**[2] **Pavlos Protopapas**[3] **Karim Pichara**[1]

[1]Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Santiago, Chile
[2]Department of Computer Science, University of Tübingen, Germany
[3]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA

A ENFORCING IC/BC

A.1 TWO-DIMENSIONAL DIRICHLET BOUNDARY VALUE PROBLEM

In the Dirichlet problem, also known as *first boundary value problem*, the values of the solution function at the boundaries Γ are known [Boyce and DiPrima, 2008]. In this case the re-parameterization is

$$\tilde{u}_\theta(x,y) := A(x,y) + \tilde{x}\big(1-\tilde{x}\big)\tilde{y}\big(1-\tilde{y}\big)u_\theta(x,y)$$

where

$$\begin{aligned}
A(x,y) &= \big(1-\tilde{x}\big)u(x_0,y) + \tilde{x}u(x_1,y) \\
&\quad + \big(1-\tilde{y}\big)\Big(u(x,y_0) - \big(1-\tilde{x}\big)u(x_0,y_0) + \tilde{x}u(x_1,y_0)\Big) \\
&\quad + \tilde{y}\Big(u(x,y_1) - \big(1-\tilde{x}\big)u(x_0,y_1) + \tilde{x}u(x_1,y_1)\Big) \\
\tilde{x} &= \frac{x-x_0}{x_1-x_0} \\
\tilde{y} &= \frac{y-y_0}{y_1-y_0} \\
(x_0,y),&(x_1,y),(x,y_0),(x,y_1) \in \Gamma
\end{aligned}$$

B ERROR BOUNDS COMPUTATION

B.1 FIRST ORDER LINEAR ODE WITH CONSTANT COEFFICIENT

A first-order linear ODE with constant coefficient has the general form

$$u'(t) + (\lambda + i\omega)u(t) = f(t) \tag{29}$$

As shown by Liu et al. [2022], the error of a PINN solution u_θ to Eq. (29) can be bounded as

$$|u_\theta(t) - u(t)| \leq \varepsilon e^{-\lambda t}\int_{t_0}^{t} e^{\lambda\tau}d\tau \tag{30}$$

when the initial conditions are satisfied i.e $u_\theta(t_0) = u(t_0)$. Here, ε is an upper bound on the residuals

$$|u'(t) + (\lambda + i\omega)u(t) - f(t)| \leq \varepsilon \quad \forall t \in I \tag{31}$$

where I can be any of the forms $(t_0,t), (t_0,t], (t_0,\infty)$.

Algorithm 1 Tight Error Bounds

Input: Domain $I = [T_{min}, T_{max}]$, residual network $r(t)$, an expression for $e^{P(t)}$ we will call EP(t), an expression for $\int_{T_{min}}^{t} e^{P(\tau)} d\tau$ we will call IntEP(t), number of partitions N, number of points in each partition K.
Output: A set of times and error bounds at those times $\{t_i, b_i\}_{i=1}^{N}$.

```
{t_i}_{i=0}^N ← Linspace(T_min, T_max, N + 1)
Initialize b_0 := 0
for i ← 1 ... N do
    I_i ← Linspace(t_{i-1}, t_i, K)
    ε_i ← max_{τ∈I_i} |r(τ)|
    b_i ← b_{i-1} + ε_i (IntEP(t_i) − IntEP(t_{i-1})) / EP(t_i)        ▷ Implementation of Eq. (37)
end for
```

Note: Linspace(a, b, n) returns an array of n points equally spaced in the range $[a, b]$.

B.2 FIRST ORDER LINEAR ODE WITH NONCONSTANT COEFFICIENT

A first-order linear ODE with nonconstant coefficient has the general form

$$u'(t) + (p(t) + iq(t))u(t) = f(t) \tag{32}$$

As shown by Liu et al. [2022], when the initial conditions are satisfied, the error of a PINN solution u_θ to Eq. (32) can be bounded as

$$|u_\theta(t) - u(t)| \leq \varepsilon e^{-P(t)} \int_{t_0}^{t} e^{P(\tau)} d\tau \tag{33}$$

Where $P(t) = \int_{t_0}^{t} p(\tau) d\tau$, and ε is an upper bound on the residuals

$$|u'(t) + (\lambda + i\omega)u(t) - f(t)| \leq \varepsilon \quad \forall t \in I \tag{34}$$

where I can be any of the forms $(t_0, t), (t_0, t], (t_0, \infty)$.

B.3 TIGHT BOUNDS COMPUTATION

In Section 2.6, we described how to obtain a bound using the global maximum residual ε in I. However, we can compute a tighter error bound by partitioning the domain $I = I_1 \uplus I_2 \uplus \cdots \uplus I_n$.

The tight bound for the first-order ODE with nonconstant coefficients turns out to be

$$|u_\theta(t) - u(t)| \leq e^{-P(t)} \int_{t_0}^{t} |r_\theta(\tau)| e^{P(\tau)} d\tau \tag{35}$$

To use the partitions I_k, we define the maximum local residual

$$\varepsilon_k := \max_{\tau \in I_k} |r_\theta(\tau)| \tag{36}$$

and we compute the bounds as

$$|u_\theta(t) - u(t)| \leq \sum_{i=1}^{n} \varepsilon_i e^{-P(t)} \int_{\tau=t_{i-1}}^{\tau=t_i} e^{P(\tau)} d\tau \tag{37}$$

where $t_k = \max I_k$ and $t_n = t$. Algorithm 1 in the Supplementary Material shows the implementation of tight bounds computation.

C NLM DETAILS

We now provide details on the NLM derivation presented in Section 4.1.1.

$$\Sigma_{\text{post}} = \left(\Phi_\theta^T \Sigma \Phi_\theta + \sigma_{\text{prior}}^{-2} I\right)^{-1} \tag{38}$$

$$\mu_{\text{post}} = \Sigma_{\text{post}} \left(\Sigma^{-1} \Phi_\theta\right)^T u_{\theta_{\text{det}}} \tag{39}$$

we have writen Φ_θ^T and Σ instead of $\Phi_\theta^T(\mathbf{x}_\mathcal{D}, \lambda_\mathcal{D})$ and $\Sigma(\mathbf{x}_\mathcal{D}, \lambda_\mathcal{D})$ respectively, to simplify notation. In Eq. (38) and Eq. (39),

$$\mathbf{x}_\mathcal{D} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$$
$$\lambda_\mathcal{D} = [\lambda_1, ..., \lambda_N]^T$$
$$\Sigma(\mathbf{x}_\mathcal{D}, \lambda_\mathcal{D}) = \text{diag}([\sigma^2_{\text{Like}}(\mathbf{x}_1, \lambda_1), ..., \sigma^2_{\text{Like}}(\mathbf{x}_M, \lambda_M)])$$

D COMPUTING PREDICTIVE UNCERTAINTY

The computation of predictive uncertainty is straightforward as it is the standard deviation of the posterior predictive distribution. We can obtain it by applying the law of total variance

$$\begin{aligned}\text{Var}(u|\mathbf{x}, \lambda, \mathcal{D}) &= \mathbb{E}_{\theta|\mathcal{D}}\left[\text{Var}(u|\mathbf{x}, \lambda, \theta)\right] + \text{Var}_{\theta|\mathcal{D}}\left[\mathbb{E}(u|\mathbf{x}, \lambda, \theta)\right] \\ &= \mathbb{E}_{\theta|\mathcal{D}}\left[\sigma^2_{\text{Like}}(\mathbf{x}, \lambda)\right] + \text{Var}_{\theta|\mathcal{D}}[u_\theta(\mathbf{x}, \lambda)] \\ &\approx \sigma^2_{\text{Like}}(\mathbf{x}, \lambda) + \frac{1}{M}\sum_{i=1}^{M}(u_{\theta_i}(\mathbf{x}, \lambda) - \overline{u(\mathbf{x}, \lambda)})^2\end{aligned} \tag{40}$$

where $\theta_i \sim p(\theta|\mathcal{D})$ and $\overline{u(\mathbf{x}, \lambda)} = \frac{1}{M}\sum_{i=1}^{M} u_{\theta_i}(\mathbf{x}, \lambda)$ is the sample mean of the network outputs.

The latter approximation applies to BBB and NUTS; for BBB, the samples are taken from the variational posterior, and for NUTS, we use the samples of the true posterior the method yields. We have the analytical expression for NLM as shown in Eq. (38).

E STATISTICAL ANALYSIS

It may also be the case that the observed values are a function f of the solution. In that scenario, the dataset would become $\mathcal{O}^f = \{(\mathbf{x}_i, \mu_i^f, \sigma_i^f) \mid \mathbf{x}_i \in \Omega, \mu_i^f \in \mathbb{R}, \sigma_i^f \in \mathbb{R}^+\}_{i=1}^{O^f}$, where μ_i^f and σ_i^f are the mean and standard deviation of the observed function, respectively. Here f is a function of the true solution and possibly other parameters λ^f, which can also be included in the set of parameters to be estimated. In a similar way to Eq. (24), we can compute the likelihood:

$$p(\mathbf{x}_i, \mu_i, \sigma_i|\lambda, \lambda^f) = \int_\mathcal{U} p(\mathbf{x}_i, \mu_i^f, \sigma_i^f|f(u_\lambda(\mathbf{x}_i), \lambda^f)) \cdot p(u_\lambda(\mathbf{x}_i)|\mathbf{x}_i, \lambda)du \tag{41}$$

F PROOFS

Proposition F.1. *Given a differential equation of the form* $\frac{\mathrm{d}u}{\mathrm{d}t} = f(t)$ *for some function* f *and an initial condition* $u(t_0) = u_0$*. The error of an approximation* $\hat{u}$ *of* u *is:*

$$\hat{u}(\tau) - u(\tau) = \int_{t_0}^{\tau} R(t)dt \tag{42}$$

if the approximation satisfies the ICs, where $R(t) := \frac{\mathrm{d}\hat{u}}{\mathrm{d}t} - f(t)$*.*

Proof. Let us take some approximate solution to the DE

$$\hat{u}(t) = u(t) + e_u(t)$$

where $\hat{u}(t)$ is the approximation, $u(t)$ is the true solution and $e_u(t)$ is an error term. We can do the same for the first derivative of the solution. Using Newton's notation:

$$\hat{\dot{u}}(t) = \dot{u}(t) + e_{\dot{u}}(t) \tag{43}$$

Using the definition of $R(t)$

$$R(t) = \dot{u}(t) + e_{\dot{u}}(t) - f(t)$$

Since $\dot{u}(t)$ is the derivative of the true solution, $\dot{u}(t) - f(t) = 0$, thus $R(t) = e_{\dot{u}}(t)$. Substituting that in Eq. (43), we get

$$R(t) = \hat{\dot{u}}(t) - \dot{u}(t)$$

Integrating both sides with respect to t:

$$\int_{t_0}^{\tau} R(t)dt = \int_{t_0}^{\tau} \hat{\dot{u}}(t)dt - \int_{t_0}^{\tau} \dot{u}(t)dt$$
$$\int_{t_0}^{\tau} R(t)dt = \hat{u}(\tau) - \hat{u}(t_0) + u(t_0) - u(\tau)$$
$$\int_{t_0}^{\tau} R(t)dt + e_u(t_0) = \hat{u}(\tau) - u(\tau)$$

If $u(t_0) = \hat{u}(t_0)$, which is the case for our PINNs because we enforce ICs, then

$$\hat{u}(\tau) - u(\tau) = \int_{t_0}^{\tau} R(t)dt$$

□

G COSMOLOGY MODELS

G.1 ΛCDM ERROR BOUNDS COMPUTATION

Error Bounds Computation Since this model is a first-order linear system with non-constant coefficients, it is possible to compute its error bounds. To compute the error bounds with Algorithm 1 we need an expression for $e^{P(z)}$ and $\int_0^z e^{P(s)}ds$. In the ΛCDM model, we have $p(z) = 3/(1+z)$, thus

$$P(z) = \int_0^z -\frac{3}{(1+s)}ds = -3\ln(1+s) \tag{44}$$

then

$$e^{P(s)} = e^{-3\ln(1+s)} \tag{45}$$

$$\int_0^z e^{P(s)}ds = \int_0^z e^{-3\ln(1+s)}ds = \frac{1}{2} - \frac{1}{2(1+z)^2} \tag{46}$$

which are the inputs needed for Algorithm 1.

G.2 PARAMETRIC DARK ENERGY ERROR BOUNDS COMPUTATION

Once again, we need to find an expression for the inputs to Algorithm 1. CPL's equation is also a first-order linear with a non-constant coefficient equation, so the procedure is the same as for ΛCDM. First, we have

$$p(z) = \frac{-3}{1+z}(1 + w_0 + \frac{w_1 z}{1+z}) \tag{47}$$

$$P(z) = \int_0^z \frac{-3}{1+s}(1 + w_0 + \frac{w_1 s}{1+s})ds \tag{48}$$

$$= -3((w_0 + w_1 + 1)\ln(z+1) + \frac{w_1}{z+1} - w_1) \tag{49}$$

we can then simplify the expression $e^{P(z)}$ as

$$e^{P(z)} = e^{-3((w_0+w_1+1)\ln(z+1)+\frac{w_1}{z+1}-w_1)} \tag{50}$$

$$= -3((w_0+w_1+1)\ln(z+1) + \frac{w_1}{z+1} - w_1) \tag{51}$$

$$= (z+1)^{-3(w_0+w_1+1)} e^{\frac{3w_1 z}{z+1}} \tag{52}$$

Finally, the integral turns out to be

$$\int_0^z e^{P(s)} ds = e^{3w_1}(s+1)^{1-3(w_0+w_1+1)} E_{2-3(w_0+w_1+1)}\left(\frac{3w_1}{s+1}\right)\Bigg|_{s=0}^{s=z} \tag{53}$$

Where $E_n(x)$ is the exponential integral defined as $E_n(x) = \int_1^\infty \frac{e^{-xt}}{t^n} dt$. We used computational tools to find the antiderivative of $e^{P(s)}$.

G.3 QUINTESSENCE

The Quintessence [Caldwell et al., 1998, Armendariz-Picon et al., 2000, Copeland et al., 1998, Zlatev et al., 1999] system of equations is

$$\frac{dx}{dN} = -3x + \frac{\sqrt{6}}{2}\lambda y^2 + \frac{3}{2}x(1+x^2-y^2)$$
$$\frac{dy}{dN} = \frac{\sqrt{6}}{2}xy\lambda + \frac{3}{2}y(1+x^2-y^2)$$

with the following initial conditions

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

where $\lambda, \Omega_{m,0}^\Lambda$ are parameters. Given a solution to the system, the Hubble parameter is computed as

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

Here H_0^Λ is also a parameter.

G.4 $f(R)$ GRAVITY

The $f(R)$ gravity [Clifton et al., 2012, Buchdahl, 1970] model, which we refer to as HS for the authors Hu and Sawicki [2007] consists of a system of 5 equations.

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

where

$$\Gamma(r) = \frac{(r+b)\left[(r+b)^2 - 2b\right]}{4br}$$

The initial conditions are

$$\begin{aligned}
x_0 &= 0 \\
y_0 &= \frac{\Omega^{\Lambda}_{m,0}(1+z_0)^3 + 2(1-\Omega^{\Lambda}_{m,0})}{2\left[\Omega^{\Lambda}_{m,0}(1+z_0)^3 + (1-\Omega^{\Lambda}_{m,0})\right]} \\
v_0 &= \frac{\Omega^{\Lambda}_{m,0}(1+z_0)^3 + 4(1-\Omega^{\Lambda}_{m,0})}{2\left[\Omega^{\Lambda}_{m,0}(1+z_0)^3 + (1-\Omega^{\Lambda}_{m,0})\right]} \\
\Omega_0 &= \frac{\Omega^{\Lambda}_{m,0}(1+z_0)^3}{\Omega^{\Lambda}_{m,0}(1+z_0)^3 + (1-\Omega^{\Lambda}_{m,0})} \\
r_0 &= \frac{\Omega^{\Lambda}_{m,0}(1+z_0)^3 + 4(1-\Omega^{\Lambda}_{m,0})}{1-\Omega^{\Lambda}_{m,0}}
\end{aligned}$$

where $b, \Omega^{\Lambda}_{m,0}$ are the system parameters. Given a solution to the system, the Hubble parameter is computed as

$$H(z) = H^{\Lambda}_0 \sqrt{\frac{r}{2v}(1-\Omega^{\Lambda}_{m,0})}$$

Here H^{Λ}_0 is also a parameter.

H IMPLEMENTATION DETAILS

As described in Section 6, we followed the implementation from [Chantada et al., 2022] to train the FCNNs in the first step. Architecture and hyperparameter details are shown in Tables 5 to 8. NLM shares FCNN details since it builds on FCNN.

We used Pytorch [Paszke et al., 2019] for FCNNs, Pyro [Bingham et al., 2018] for BNNs, and Neurodiffeq [Chen et al., 2020] for aiding in PINNs training.

Table 5: Implementation Details For FCNN.

Equation	Input Dim.	Output Dim.	Hidden Units	Activation	Iterations	Samples per Dim. in a Batch	Learning Rate
ΛCDM	1	1	(32, 32)	Tanh	100,000	64	0.001
CPL	1	2	(32, 32)	Tanh	100,000	128	0.001
Quintessence	1	1	(32, 32)	Tanh	100,000	32	0.001
HS	1	1	(32, 32)	Tanh	600,000	32	0.001

Table 6: Implementation Details For NLM.

Equation	Samples per Dim.	Likelihood Std.
ΛCDM	100	0.1
Quintessence	32	0.005
HS	32	0.005

Table 7: Implementation Details For BBB.

Equation	Input Dim.	Output Dim.	Hidden Units	Activation	Iterations	Samples per Dim.	Prior Std.	Learning Rate	Likelihood Std.
ΛCDM	1	1	(32, 32)	Tanh	20,000	64	1	0.001	0.1
CPL	1	2	(32, 32)	Tanh	20,000	128	1	0.001	0.01
Quintessence	1	1	(32, 32)	Tanh	20,000	32	1	0.001	0.005
HS	1	1	(32, 32)	Tanh	20,000	32	1	0.001	0.005

Table 8: Implementation Details For HMC.

Equation	Input Dim.	Output Dim.	Hidden Units	Activation	Posterior Samples	Tune Samples	Samples per Dim.	Prior Std.	Likelihood Std.
ΛCDM	1	1	(32, 32)	Tanh	10,000	1,000	32	1	0.1
CPL	1	2	(32, 32)	Tanh	10,000	1,000	128	1	0.01
Quintessence	1	1	(32, 32)	Tanh	10,000	1,000	32	1	0.005
HS	1	1	(32, 32)	Tanh	10,000	1,000	32	1	0.005

Table 9: Measurements Of The Hubble Parameter H Using The Cosmic Chronometers Technique

z	$H(z) \pm \sigma_H \left[\frac{\text{km/s}}{\text{Mpc}}\right]$	Ref.
0.09	69 ± 12	[Simon et al., 2005]
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	[Stern et al., 2010]
0.88	90 ± 40	
0.1791	75 ± 4	[Moresco et al., 2012]
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	[Cong et al., 2014]
0.12	68.6 ± 26.2	
0.2	72.9 ± 29.6	
0.28	88.8 ± 36.6	
1.363	160 ± 33.6	[Moresco, 2015]
1.965	186.5 ± 50.4	
0.3802	83 ± 13.5	[Moresco et al., 2016]
0.4004	77 ± 10.2	
0.4247	87.1 ± 11.2	
0.4497	92.8 ± 12.9	
0.4783	80.9 ± 9	

Table 10: Implementation Details of Appendix I.

Equation	Number of NN (N)	Iterations (I)	Time Steps (T)
ΛCDM	1000	1000	50
CPL	1000	1000	50
Quintessence	1500	1000	50
HS	200	5000	50

I EXPLORATION OF SOLUTIONS, RESIDUALS AND ERRORS

To analyze the behavior of residuals and errors in PINNs, we conducted a systematic data collection process. We trained N deterministic PINNs for I iterations to solve each of the cosmological models under study. Throughout training, we recorded solutions and residuals every 10 iterations, evaluating them on a fixed, equidistant set of T time steps. Detailed specifications for each cosmological model are provided in Table 10.

Distribution of Solutions After collecting the data, we generated histograms of the solutions throughout training, as shown in Figs. 4 to 12. For reference, we overlaid Gaussian distributions with the sample mean and variance in orange. The distribution of solutions varies across cosmological models, with ΛCDM and CPL being the closest to Gaussian, while HS exhibits noticeable bimodality.

Residuals and Errors Relationship Understanding the relationship between solution errors and residuals is crucial for assessing whether residuals can serve as a reliable source of calibrated uncertainties. Ideally, a perfect correlation between residuals and errors would indicate that residuals effectively capture uncertainty.

In Figs. 13 to 16, we plot the absolute, normalized values of relative errors against their corresponding residuals for all cosmological models. The results indicate no clear correlation; in fact, the trend often shows higher errors associated with lower residuals. This behavior aligns with the implicit bias described by Wang et al. [2022], where PINNs exhibit lower convergence rates near initial conditions.

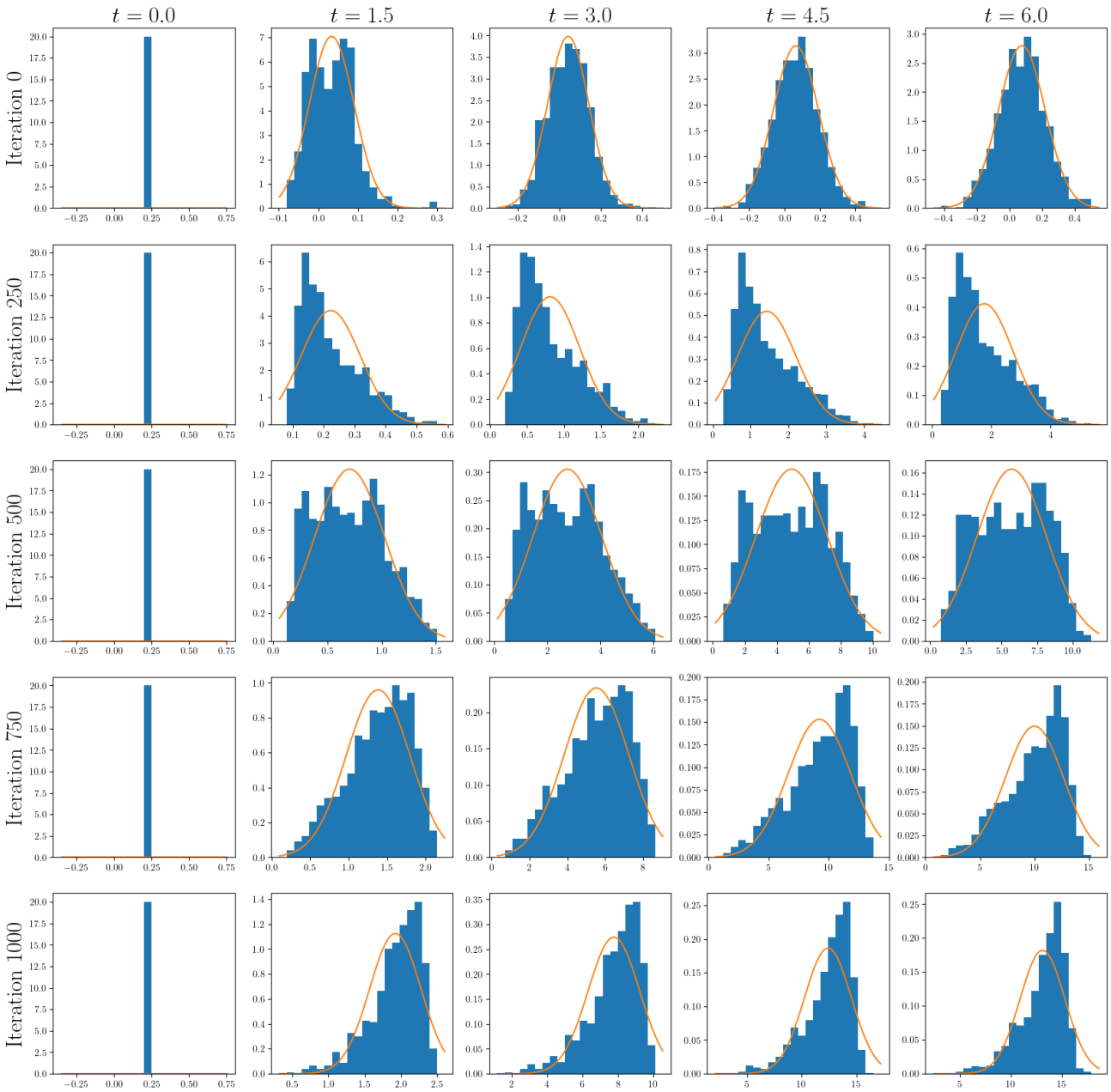

Figure 4: Distribution of $x_m(z)$ from ΛCDM. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

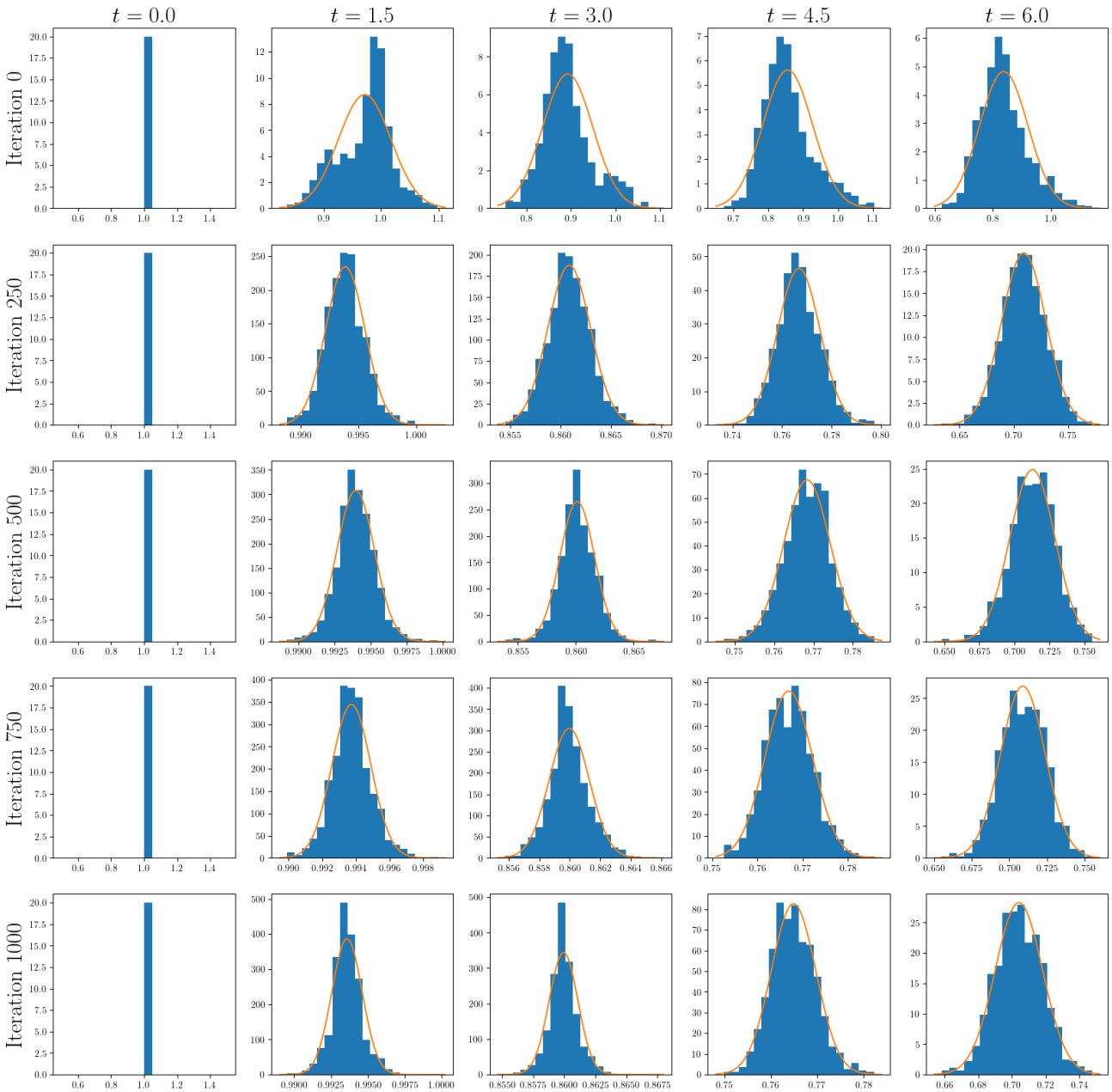

Figure 5: Distribution of $x_{\mathrm{DE}}(z)$ from CPL. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

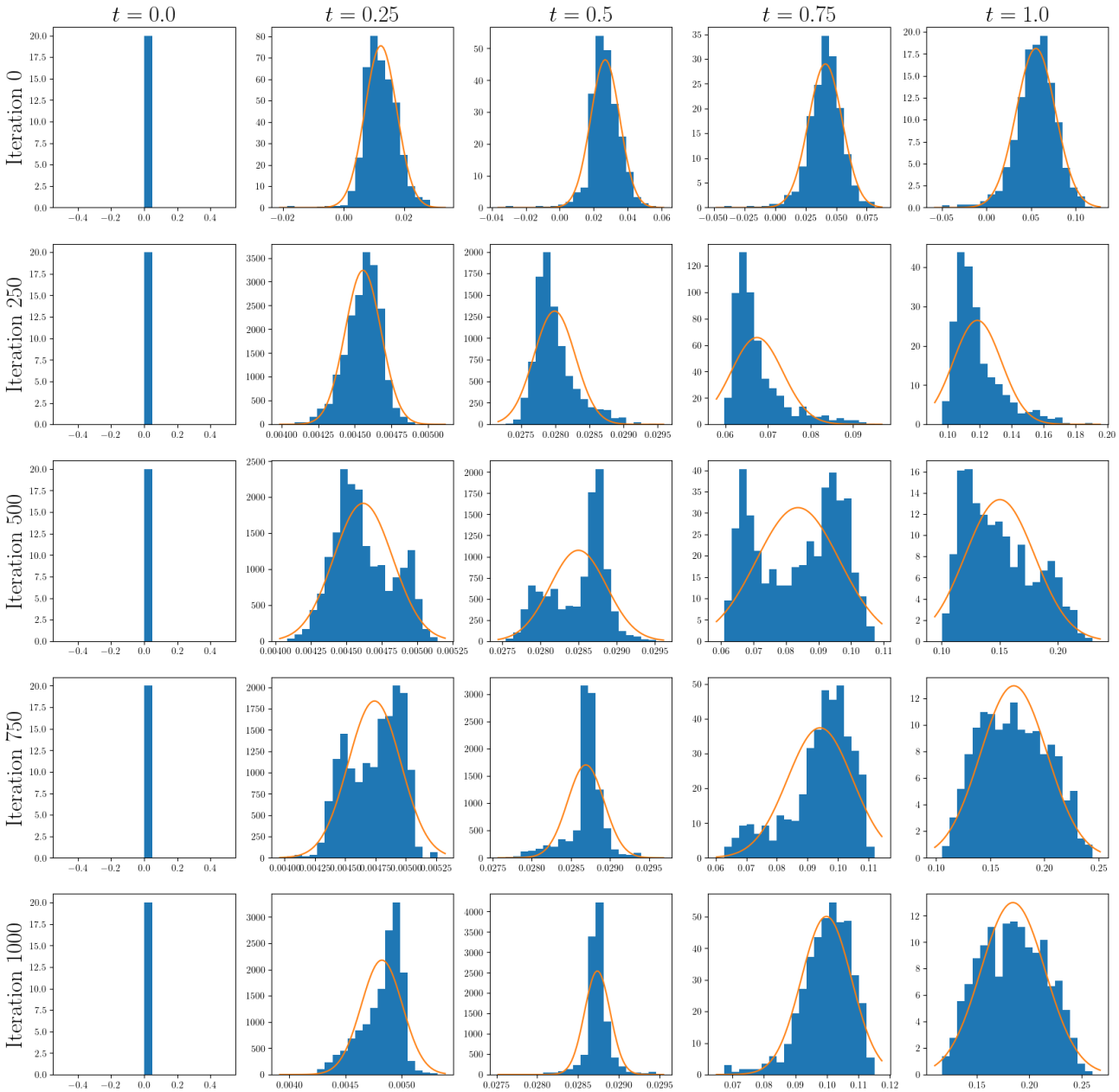

Figure 6: Distribution of $x(N)$ from Quintessence. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

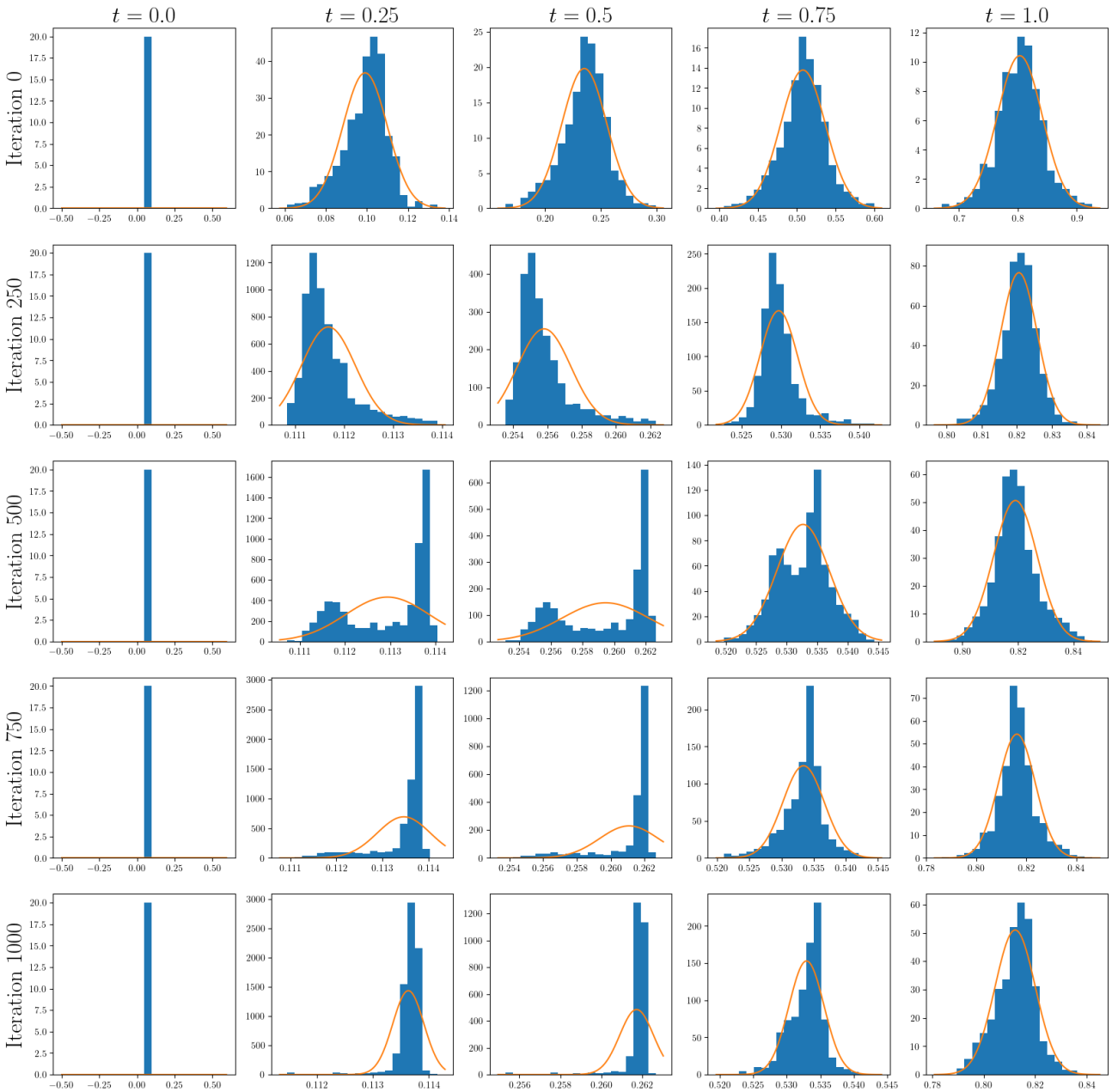

Figure 7: Distribution of $y(N)$ from Quintessence. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

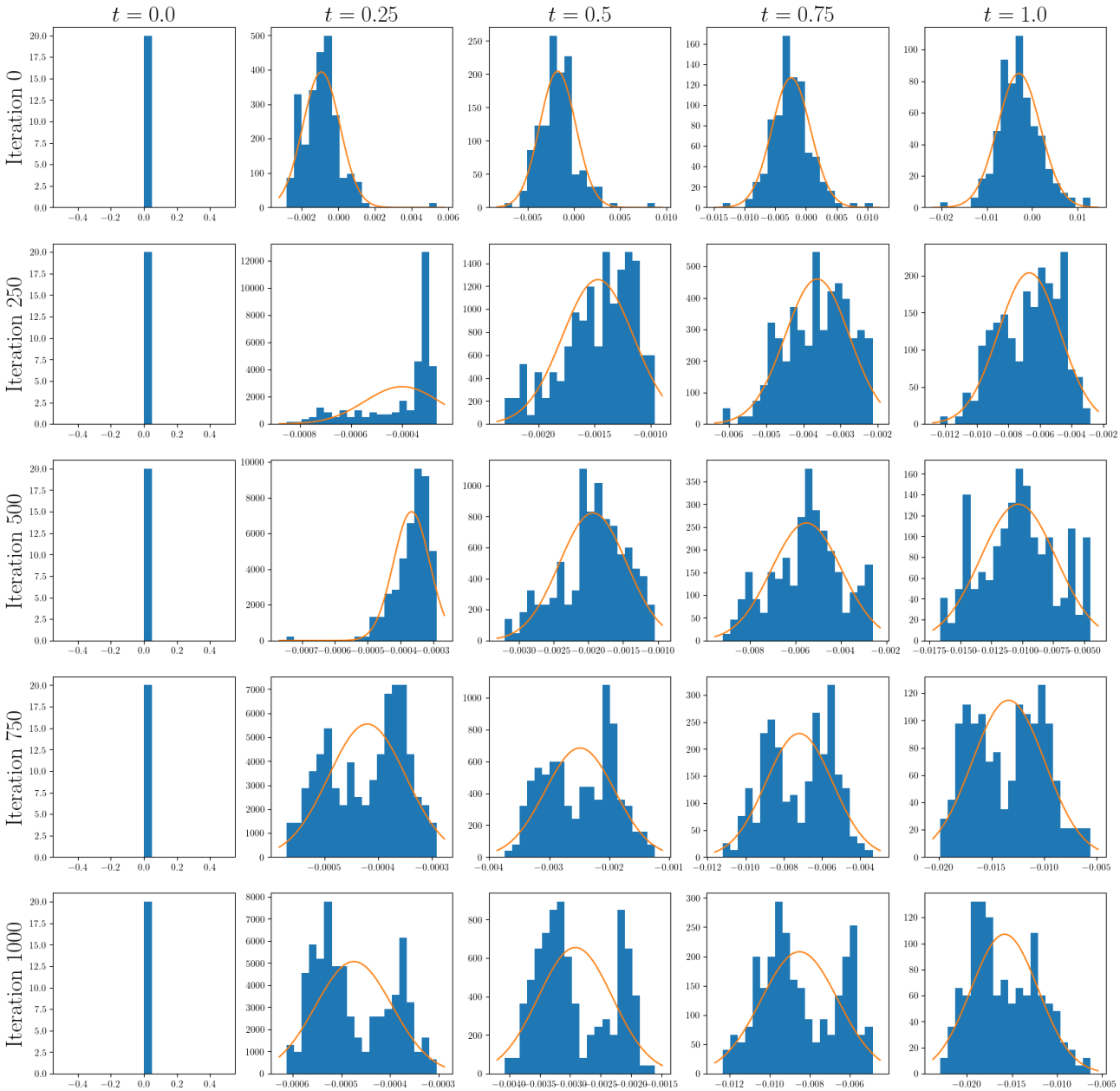

Figure 8: Distribution of $x(z)$ from HS. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

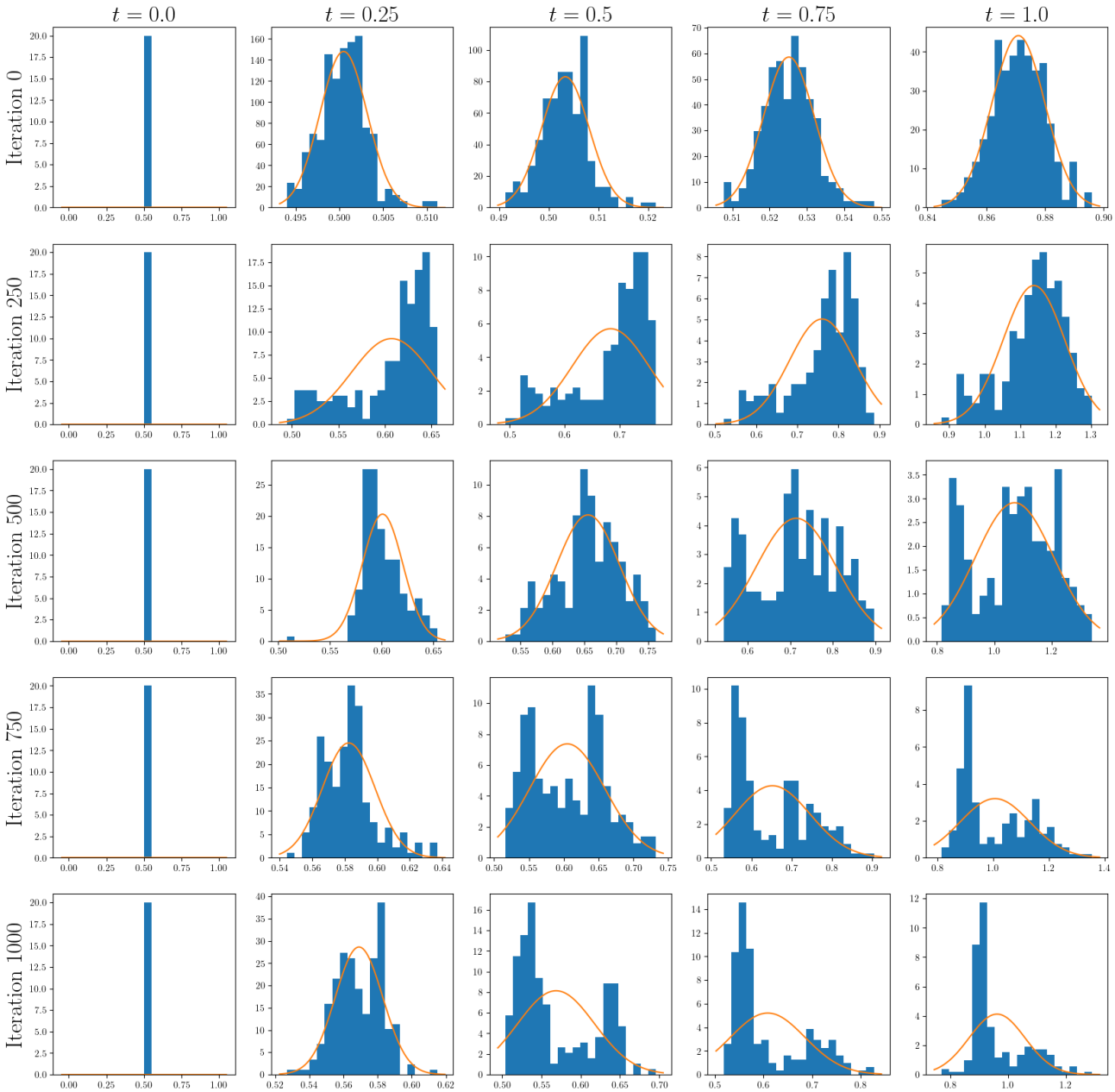

Figure 9: Distribution of $y(z)$ from HS. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

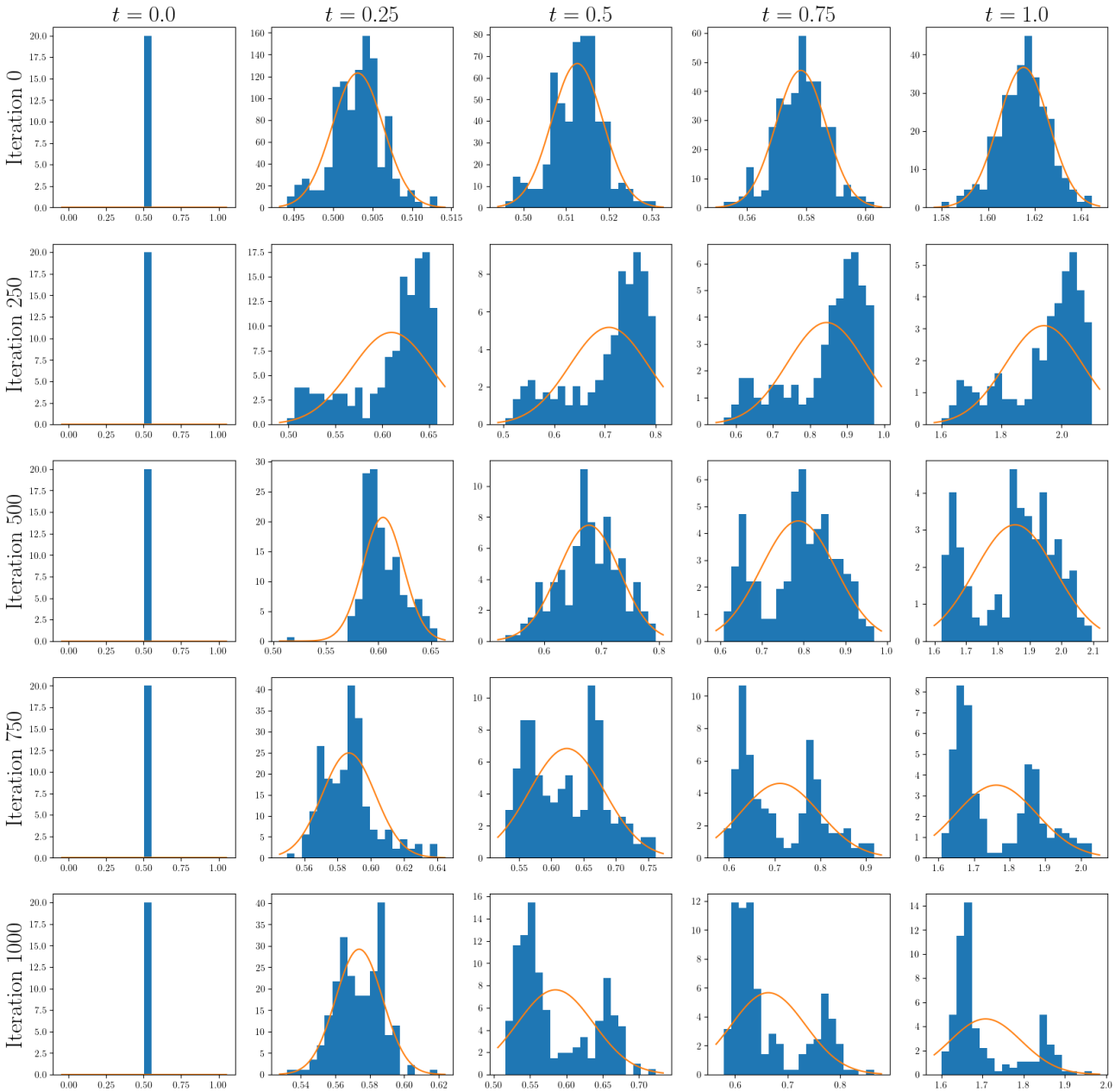

Figure 10: Distribution of $v(z)$ from HS. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

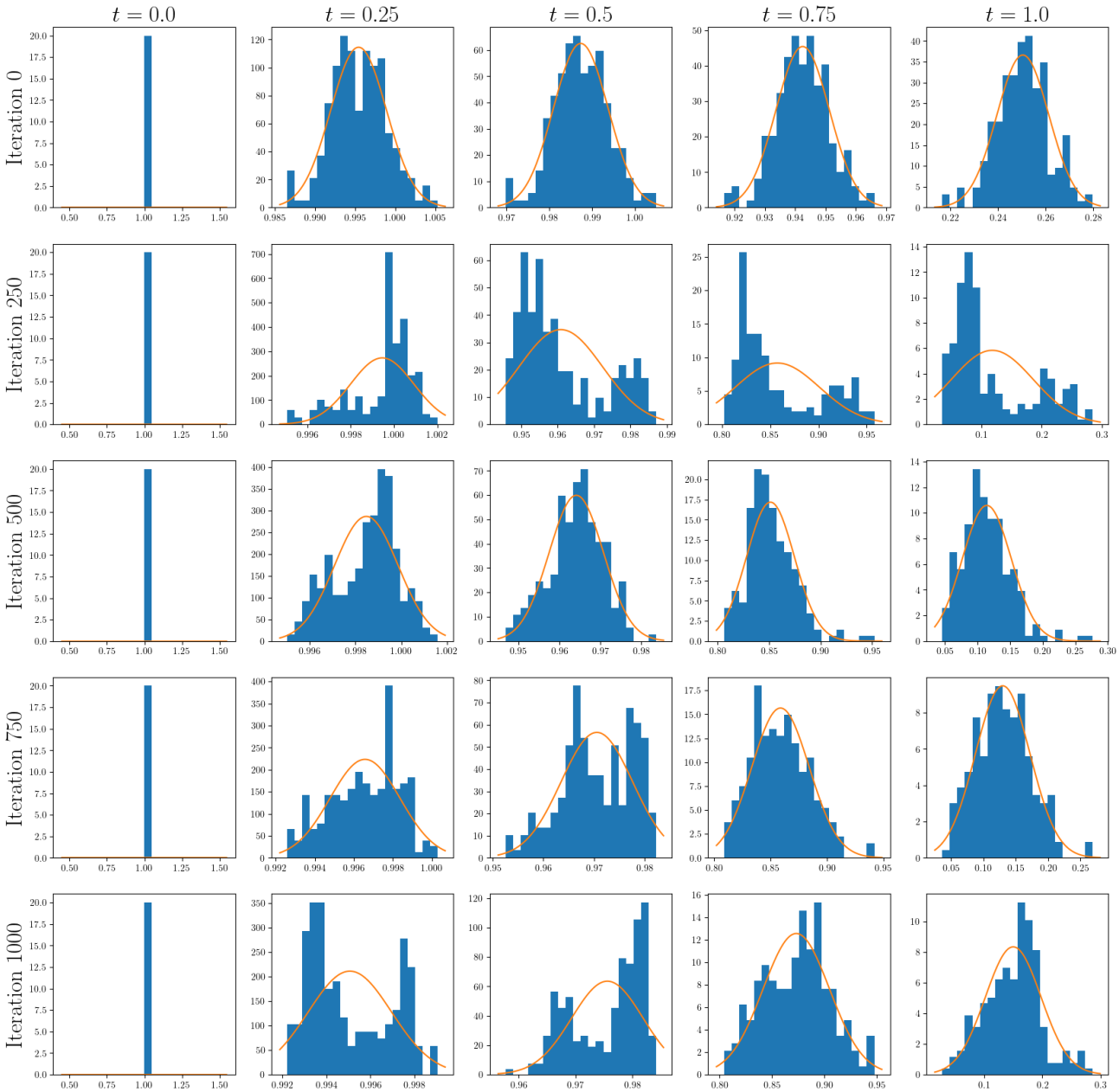

Figure 11: Distribution of $\Omega(z)$ from HS. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

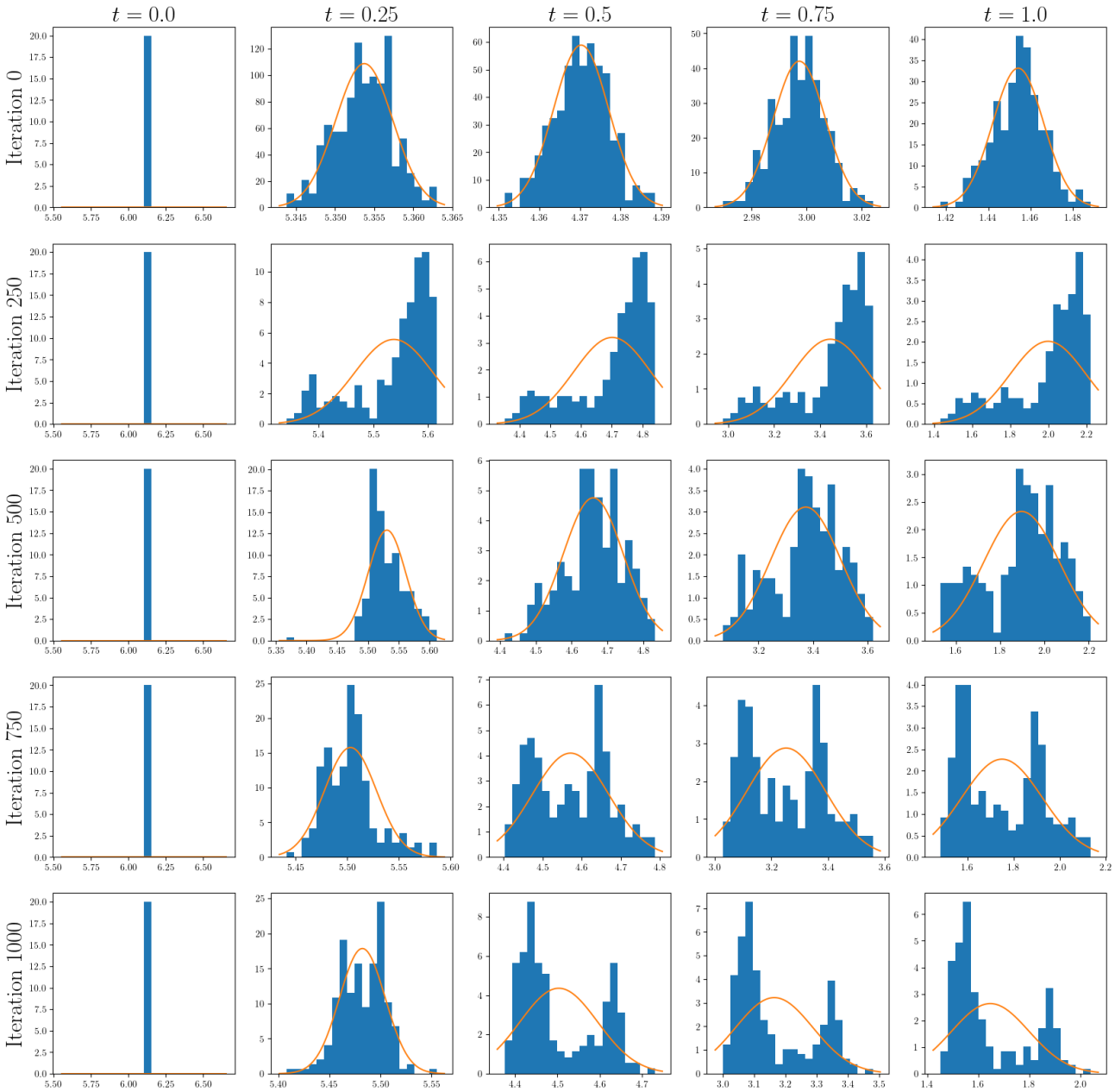

Figure 12: Distribution of $r(z)$ from HS. Samples were collected as described in I. Orange lines show a Gaussian distribution with sample mean and variance.

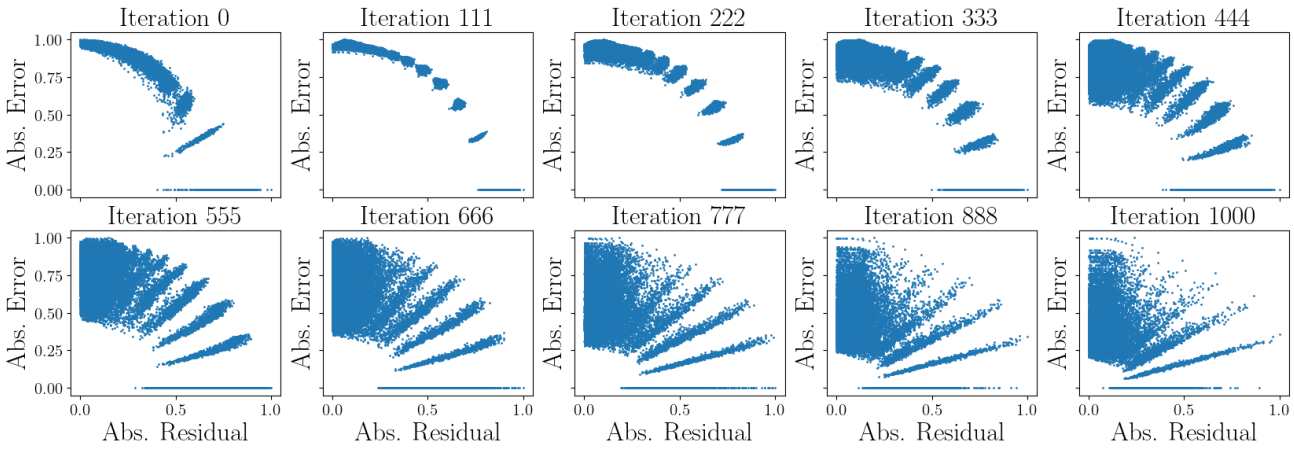

Figure 13: Relationship Between Solution Errors and Residuals from ΛCDM. Samples were collected as described in I.

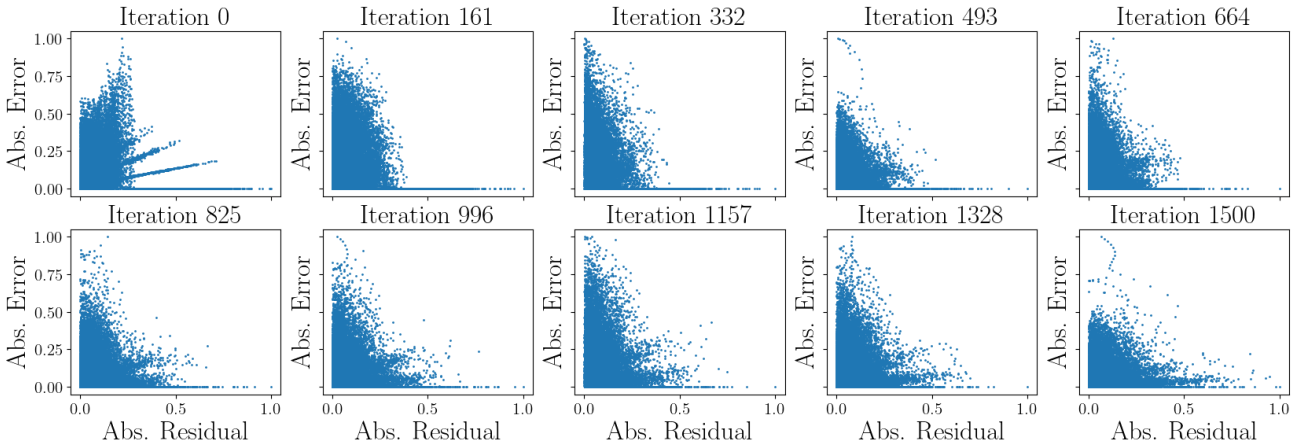

Figure 14: Relationship Between Solution Errors and Residuals from CPL. Samples were collected as described in I.

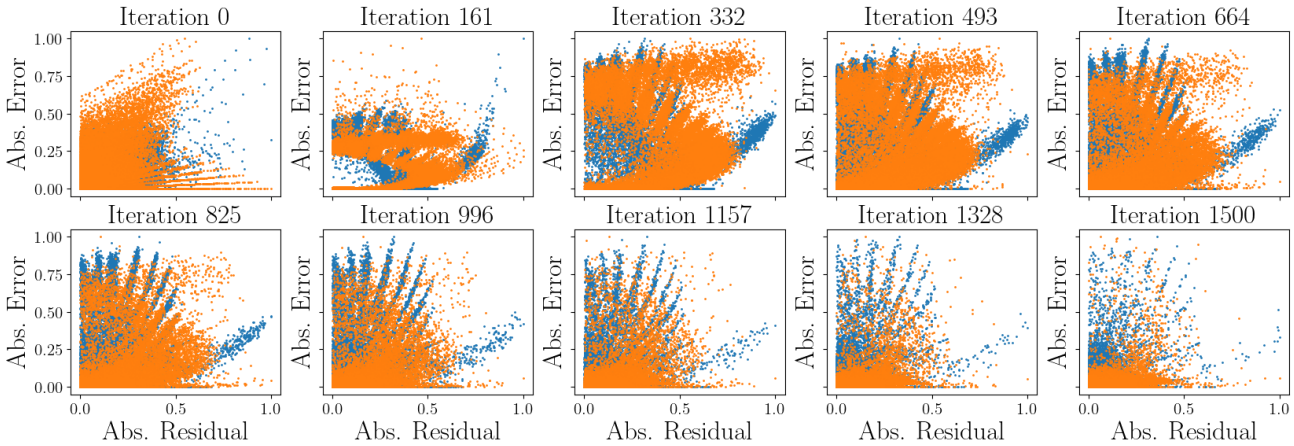

Figure 15: Relationship Between Solution Errors and Residuals from Quintessence. Samples were collected as described in I. Colors differentiate variables in the system.

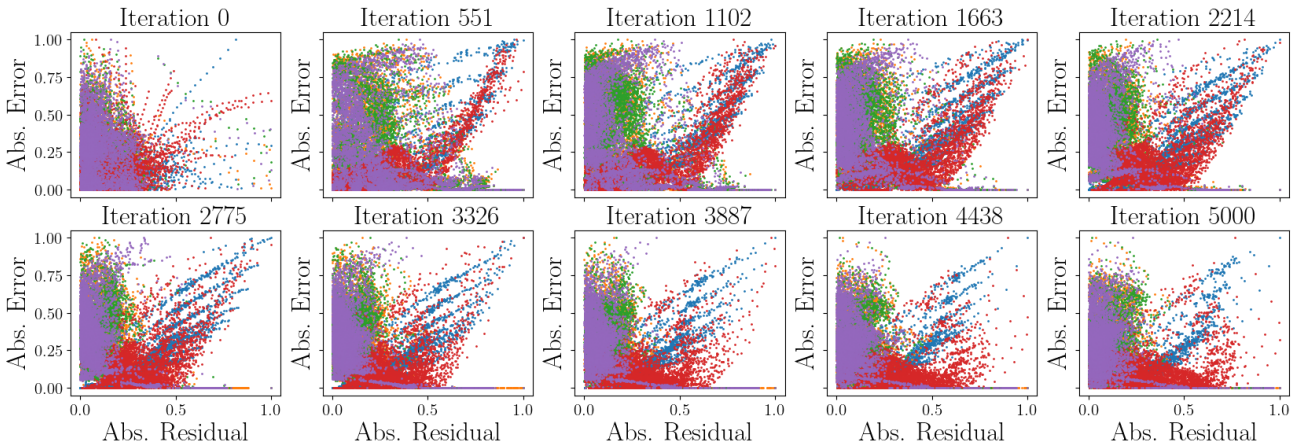

Figure 16: Relationship Between Solution Errors and Residuals from HS. Samples were collected as described in I. Colors differentiate variables in the system.

J ADITIONAL RESULTS

Table 11: Parameters Mean And Standard Deviation Estimation Of All Cosmology Models.

Equation	Method	w_0	w_1	λ	b	$\Omega_{m,0}$	H_0
ΛCDM	FCNN	-	-	-	-	0.31 ± 0.05	68.5 ± 2.63
	BBB	-	-	-	-	0.1 ± 0.01	79.95 ± 0.75
	HMC	-	-	-	-	0.34 ± 0.01	68.23 ± 0.4
	NLM + 2S	-	-	-	-	0.33 ± 0.01	67.5 ± 0.51
	BBB + 2S	-	-	-	-	0.33 ± 0.01	67.42 ± 0.68
	HMC + 2S	-	-	-	-	0.32 ± 0.01	68.05 ± 0.4
	NLM + 2S + EB	-	-	-	-	0.32 ± 0.01	68.14 ± 0.36
	BBB + 2S + EB	-	-	-	-	0.31 ± 0.01	68.42 ± 0.43
	HMC + 2S + EB	-	-	-	-	0.32 ± 0.01	68.16 ± 0.37
CPL	FCNN	-1.03 ± 0.3	-2.49 ± 2.55	-	-	0.34 ± 0.12	65.76 ± 7.61
	BBB	-1.0 ± 0.25	-2.53 ± 3.06	-	-	0.32 ± 0.13	66.08 ± 6.47
	HMC	-1.05 ± 0.3	-2.46 ± 2.88	-	-	0.32 ± 0.14	65.8 ± 7.06
	NLM + 2S	-1.06 ± 0.32	-2.91 ± 2.76	-	-	0.3 ± 0.16	64.33 ± 7.58
	BBB + 2S	-1.01 ± 0.3	-2.82 ± 2.48	-	-	0.31 ± 0.16	65.01 ± 6.63
	HMC + 2S	-1.07 ± 0.33	-3.02 ± 2.45	-	-	0.34 ± 0.15	66.74 ± 7.27
	NLM + 2S + EB	-0.91 ± 0.29	-2.61 ± 2.63	-	-	0.44 ± 0.14	63.53 ± 6.46
	BBB + 2S + EB	-1.06 ± 0.25	-2.24 ± 2.85	-	-	0.32 ± 0.11	66.41 ± 6.7
	HMC + 2S + EB	-1.12 ± 0.28	-2.57 ± 3.18	-	-	0.32 ± 0.14	67.65 ± 6.62
Quint.	FCNN	-	-	1.24 ± 1.0	-	0.27 ± 0.07	67.36 ± 7.14
	BBB	-	-	0.88 ± 1.16	-	0.29 ± 0.07	66.33 ± 5.28
	HMC	-	-	0.92 ± 1.15	-	0.3 ± 0.06	67.96 ± 4.43
	NLM + 2S	-	-	1.36 ± 0.83	-	0.32 ± 0.1	64.5 ± 4.95
	BBB + 2S	-	-	1.13 ± 1.19	-	0.3 ± 0.05	68.05 ± 5.46
	HMC + 2S	-	-	0.8 ± 1.14	-	0.3 ± 0.06	65.7 ± 5.62
HS	FCNN	-	-	-	2.53 ± 1.87	0.28 ± 0.07	69.03 ± 7.94
	BBB	-	-	-	1.82 ± 1.95	0.3 ± 0.06	66.75 ± 6.75
	HMC	-	-	-	2.06 ± 1.96	0.27 ± 0.08	66.73 ± 6.5
	NLM + 2S	-	-	-	2.34 ± 1.84	0.28 ± 0.12	57.44 ± 8.81
	BBB + 2S	-	-	-	1.98 ± 1.87	0.28 ± 0.07	72.78 ± 10.01
	HMC + 2S	-	-	-	2.29 ± 1.67	0.28 ± 0.09	67.82 ± 10.38

Table 12: Smallest Number Of Sigmas Within Which The Results From [D'Agostino and Nunes, 2020, Motta et al., 2021, Akrami et al., 2019] Fall.

Equation	Method	w_0	w_1	λ	b	$\Omega_{m,0}$	H_0
ΛCDM	FCNN	-	-	-	-	1σ	1σ
	BBB	-	-	-	-	19σ	15σ
	HMC	-	-	-	-	5σ	3σ
	NLM	-	-	-	-	4σ	4σ
	BBB	-	-	-	-	3σ	3σ
	HMC	-	-	-	-	4σ	3σ
	NLM + EB	-	-	-	-	4σ	3σ
	BBB + EB	-	-	-	-	3σ	2σ
	HMC + EB	-	-	-	-	4σ	3σ
CPL	FCNN	2σ	1σ	-	-	1σ	1σ
	BBB	3σ	1σ	-	-	1σ	2σ
	HMC	2σ	1σ	-	-	1σ	2σ
	NLM	2σ	1σ	-	-	1σ	2σ
	BBB	2σ	2σ	-	-	1σ	2σ
	HMC	2σ	2σ	-	-	1σ	1σ
	NLM + EB	3σ	1σ	-	-	2σ	2σ
	BBB + EB	2σ	1σ	-	-	1σ	1σ
	HMC + EB	2σ	1σ	-	-	1σ	1σ
Quint.	FCNN	-	-	2σ	-	1σ	-
	BBB	-	-	1σ	-	1σ	-
	HMC	-	-	1σ	-	1σ	-
	NLM	-	-	2σ	-	1σ	-
	BBB	-	-	1σ	-	1σ	-
	HMC	-	-	1σ	-	1σ	-
HS	FCNN	-	-	-	2σ	1σ	1σ
	BBB	-	-	-	1σ	1σ	1σ
	HMC	-	-	-	1σ	1σ	1σ
	NLM	-	-	-	2σ	1σ	2σ
	BBB	-	-	-	1σ	1σ	1σ
	HMC	-	-	-	2σ	1σ	1σ

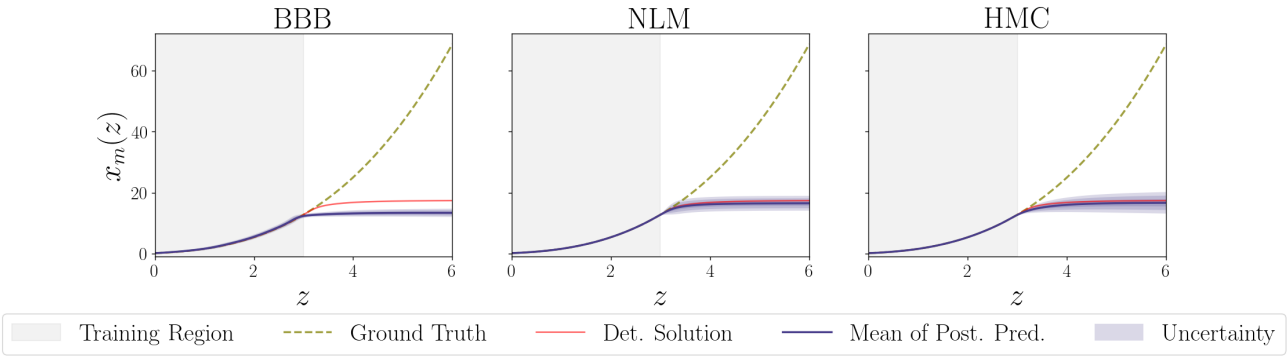

Figure 17: ΛCDM Bayesian Solutions. The Analytic Solution Is Presented In Dotted Lines.

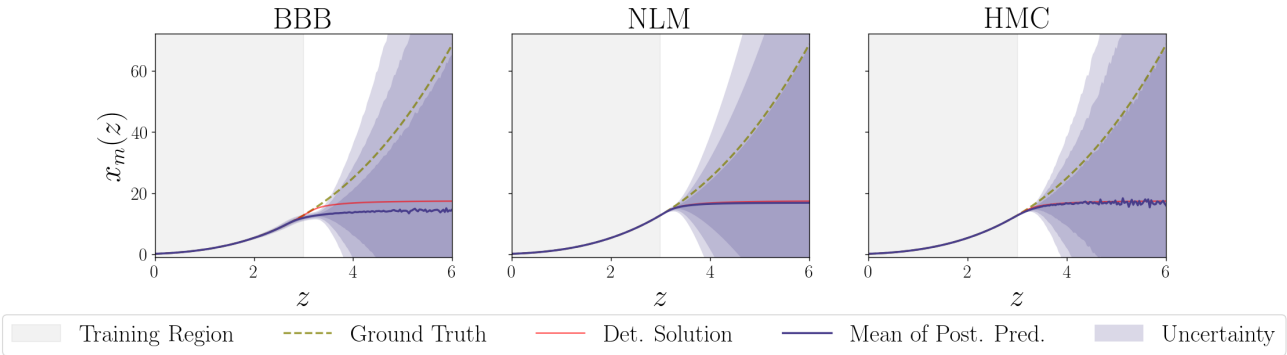

Figure 18: ΛCDM Bayesian Solutions With Error Bounds. The Analytic Solution Is Presented In Dotted Lines.

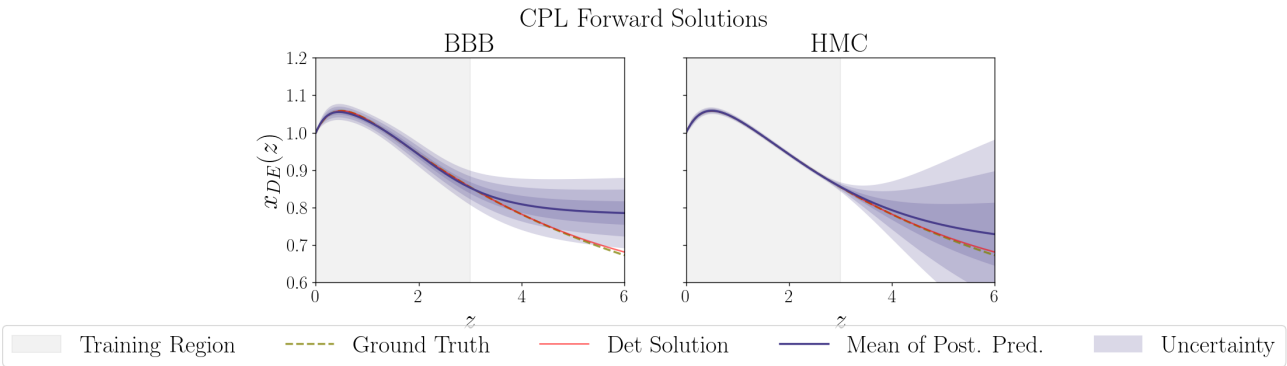

Figure 19: CPL Bayesian Solutions. The Analytic Solution Is Presented In Dotted Lines.

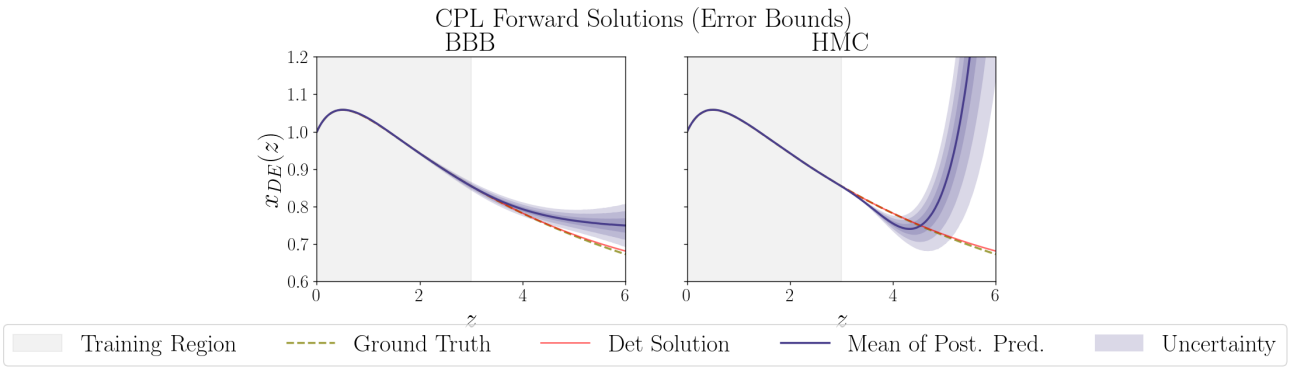

Figure 20: CPL Bayesian Solutions With Error Bounds. The Analytic Solution Is Presented In Dotted Lines.

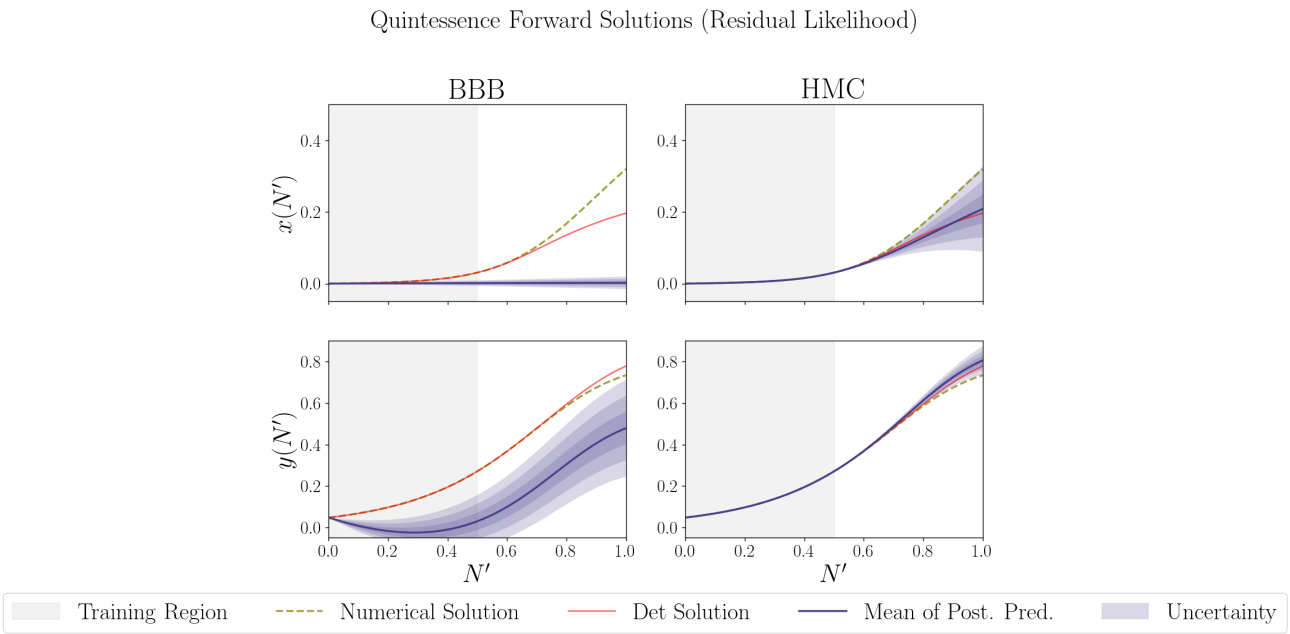

Figure 21: Quintessence Bayesian Solutions Residual Likelihood. The Numerical Solution Is Presented In Dotted Lines.

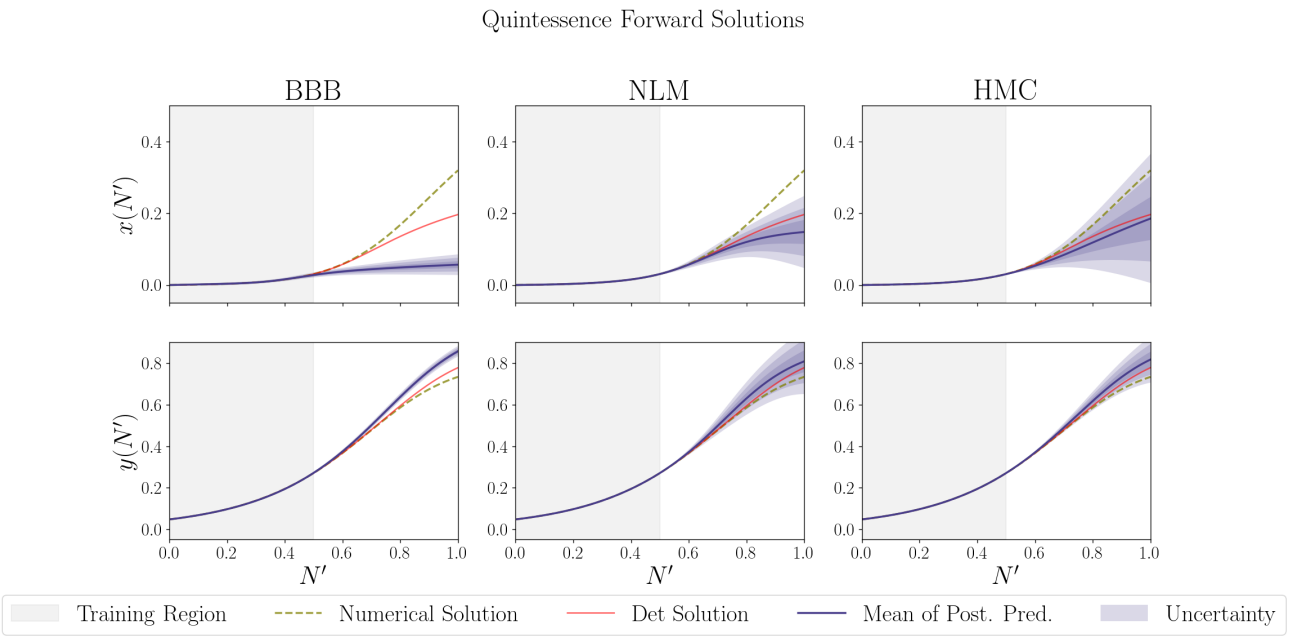

Figure 22: Quintessence Bayesian Solutions. The Numerical Solution Is Presented In Dotted Lines.

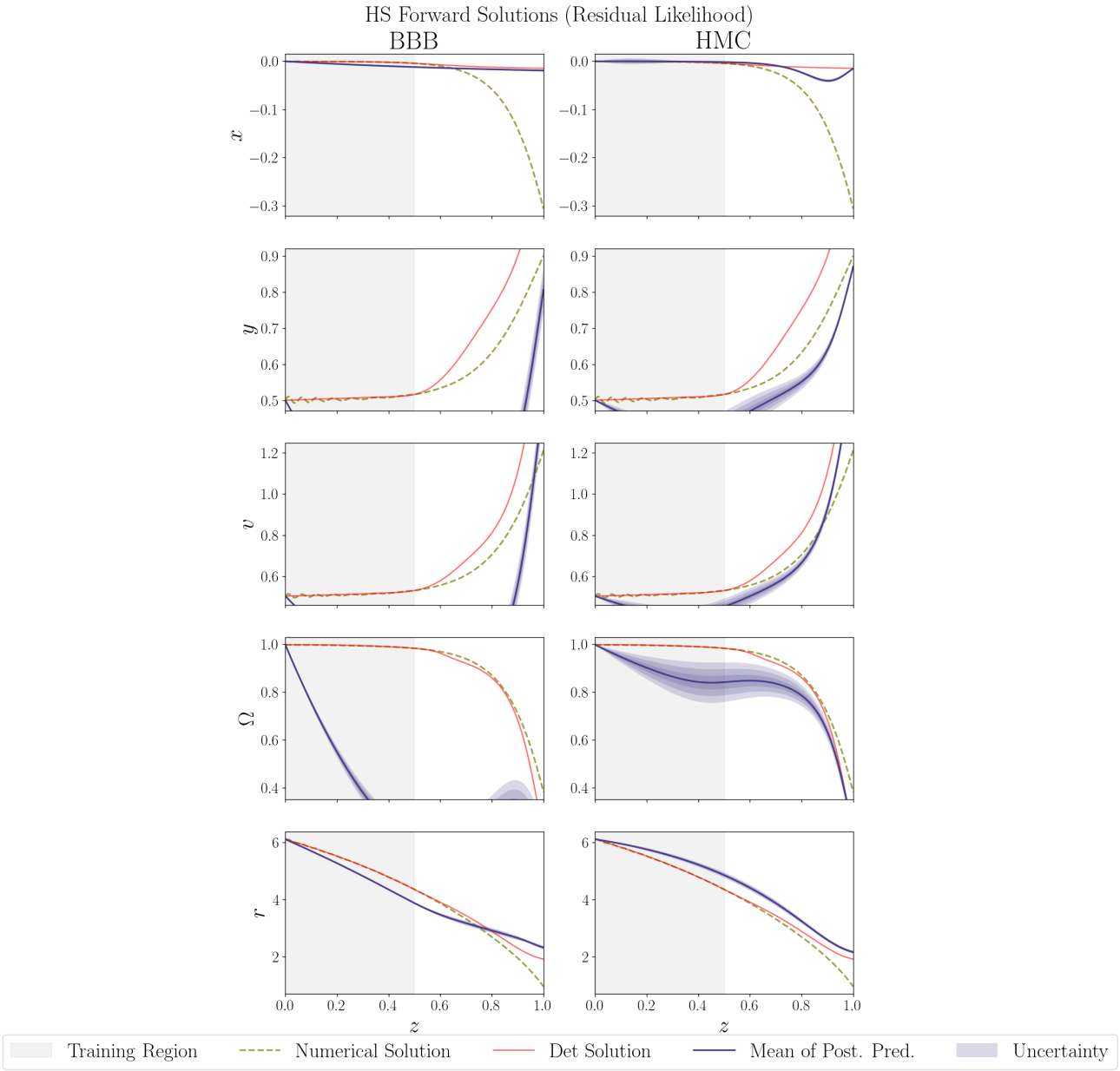

Figure 23: HS Bayesian solutions Residual Likelihood. The Numerical Solution Is Presented In Dotted Lines.

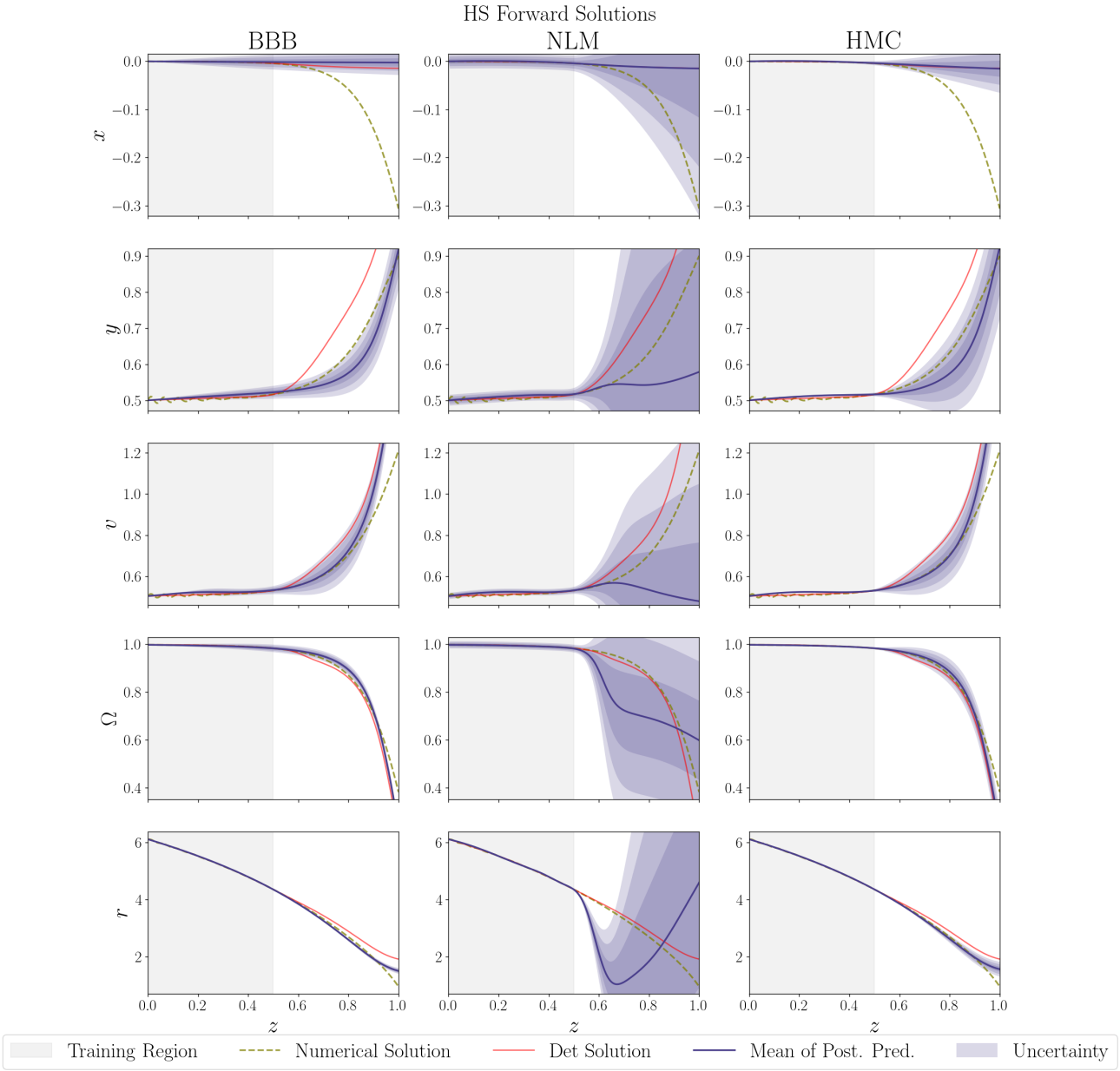

Figure 24: HS Bayesian solutions. The Numerical Solution Is Presented In Dotted Lines.

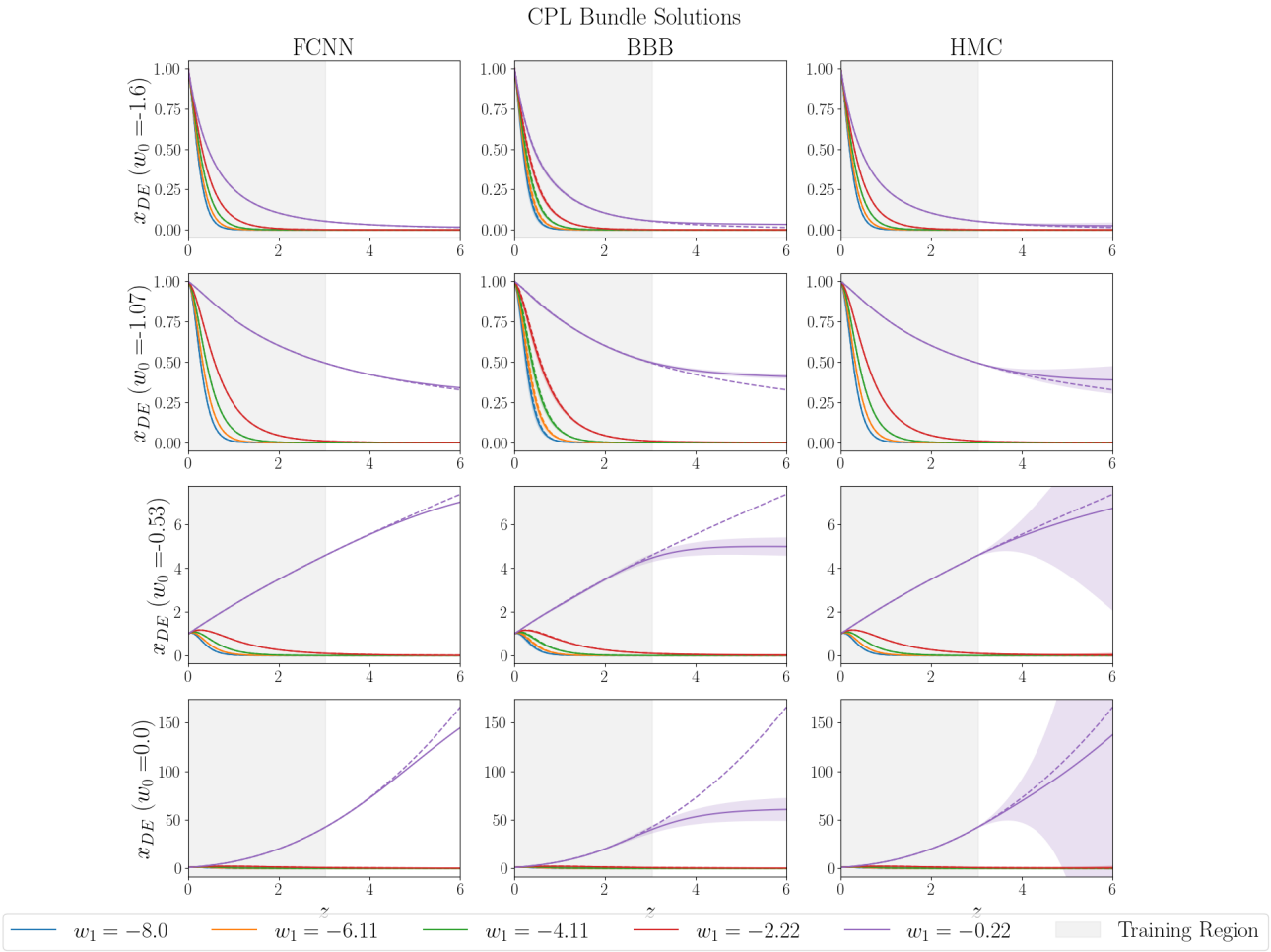

Figure 25: Examples Of CPL Bayesian Solutions Obtained Using The Bundle Solver. Analytic Solutions Are Presented In Dotted Lines.

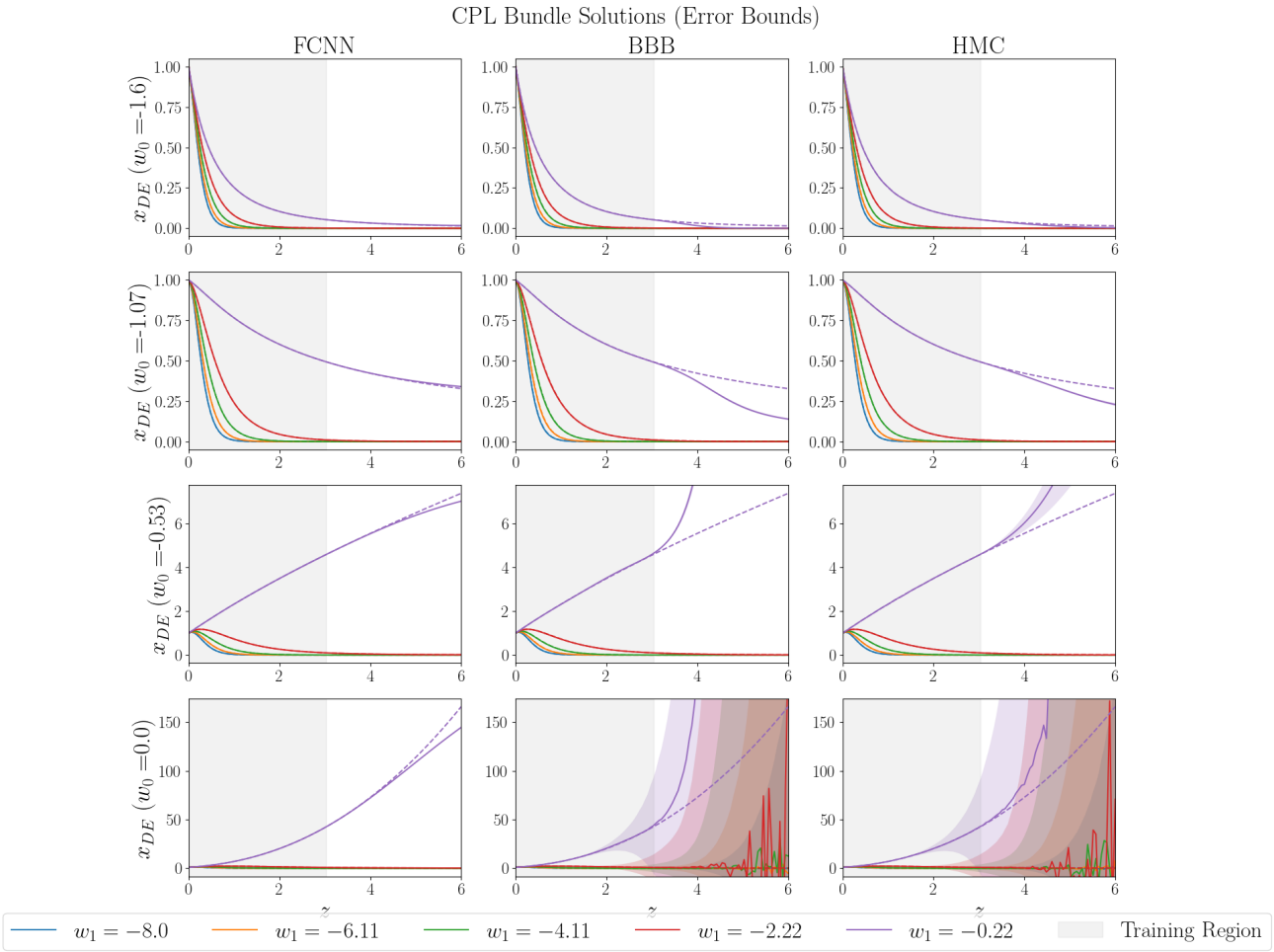

Figure 26: Examples Of CPL Bayesian Solutions Obtained Using The Bundle Solver With Error Bounds. Analytic Solutions Are Presented In Dotted Lines.

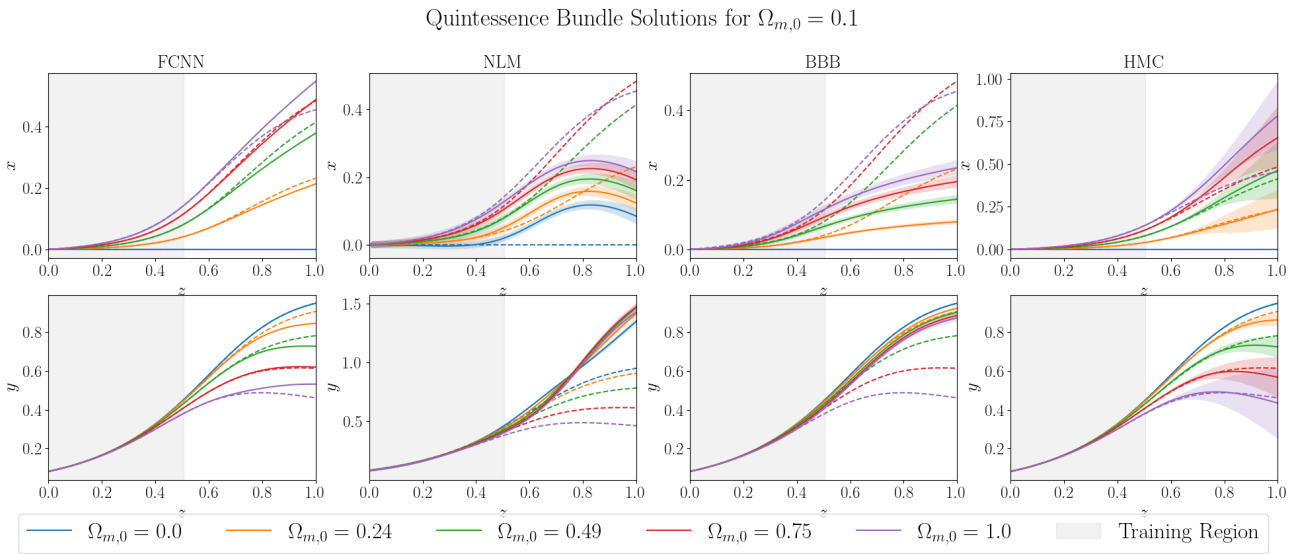

Figure 27: Examples of Quintessence Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.1$. Numerical Solutions Are Presented In Dotted Lines.

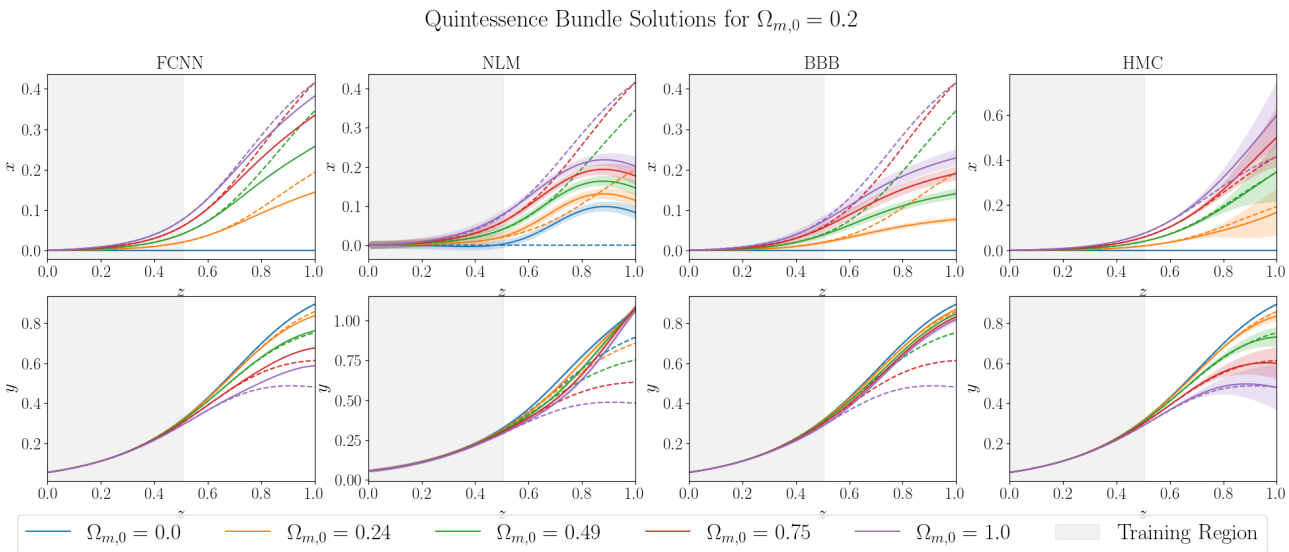

Figure 28: Examples of Quintessence Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.2$. Numerical Solutions Are Presented In Dotted Lines.

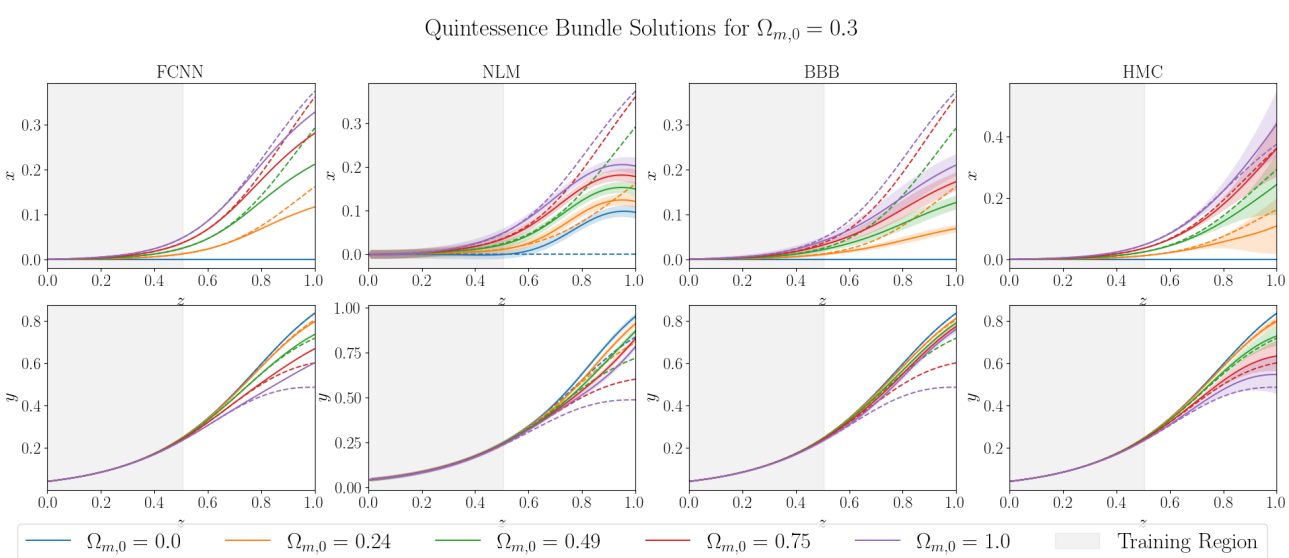

Figure 29: Examples of Quintessence Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.3$. Numerical Solutions Are Presented In Dotted Lines.

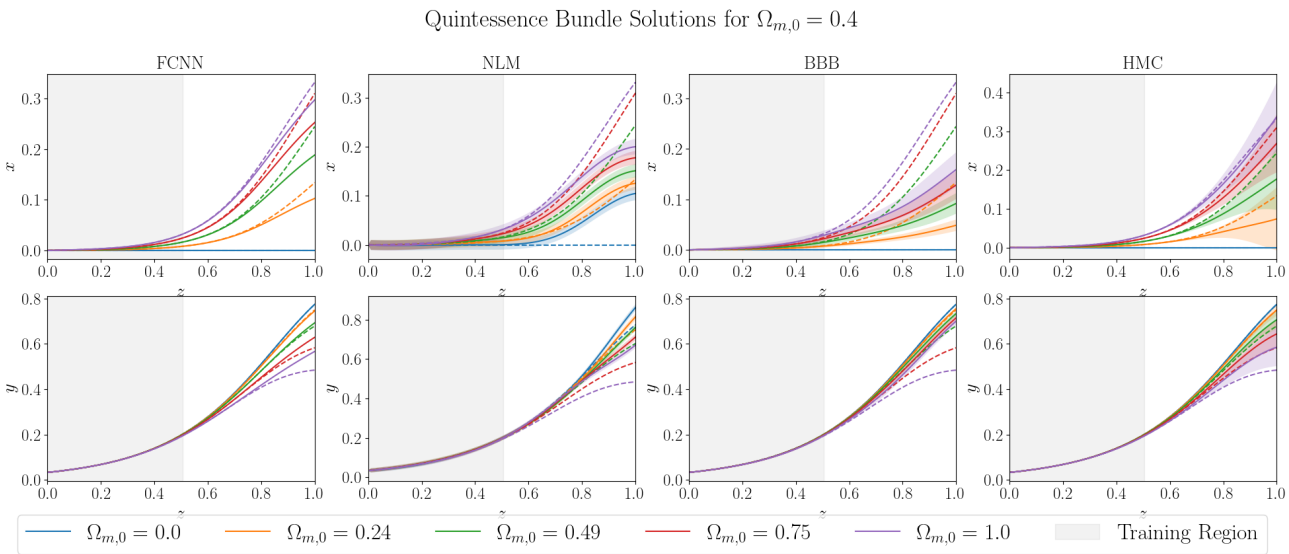

Figure 30: Examples of Quintessence Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.4$. Numerical Solutions Are Presented In Dotted Lines.

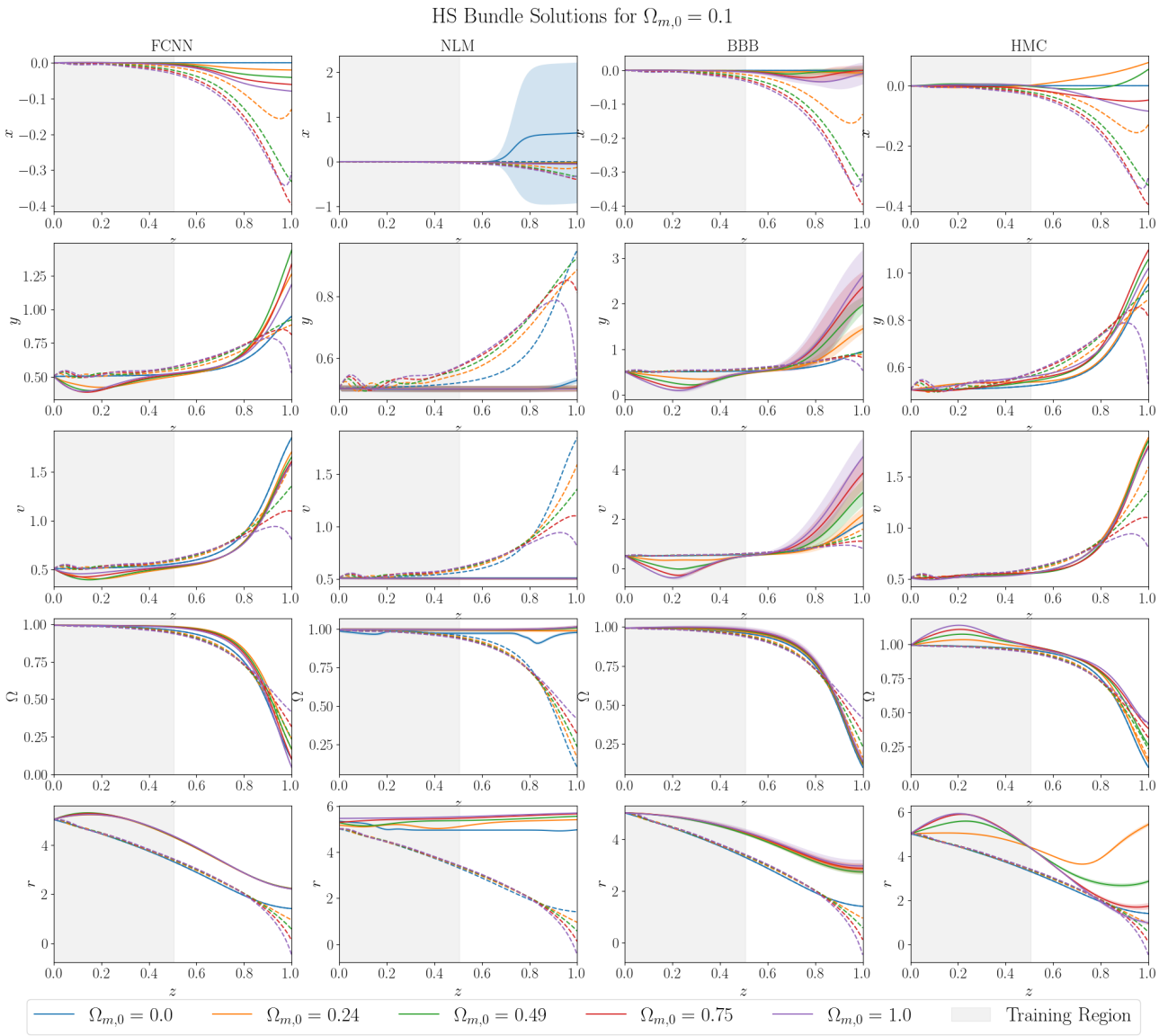

Figure 31: Examples Of HS Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.1$. Numerical Solutions Are Presented In Dotted Lines.

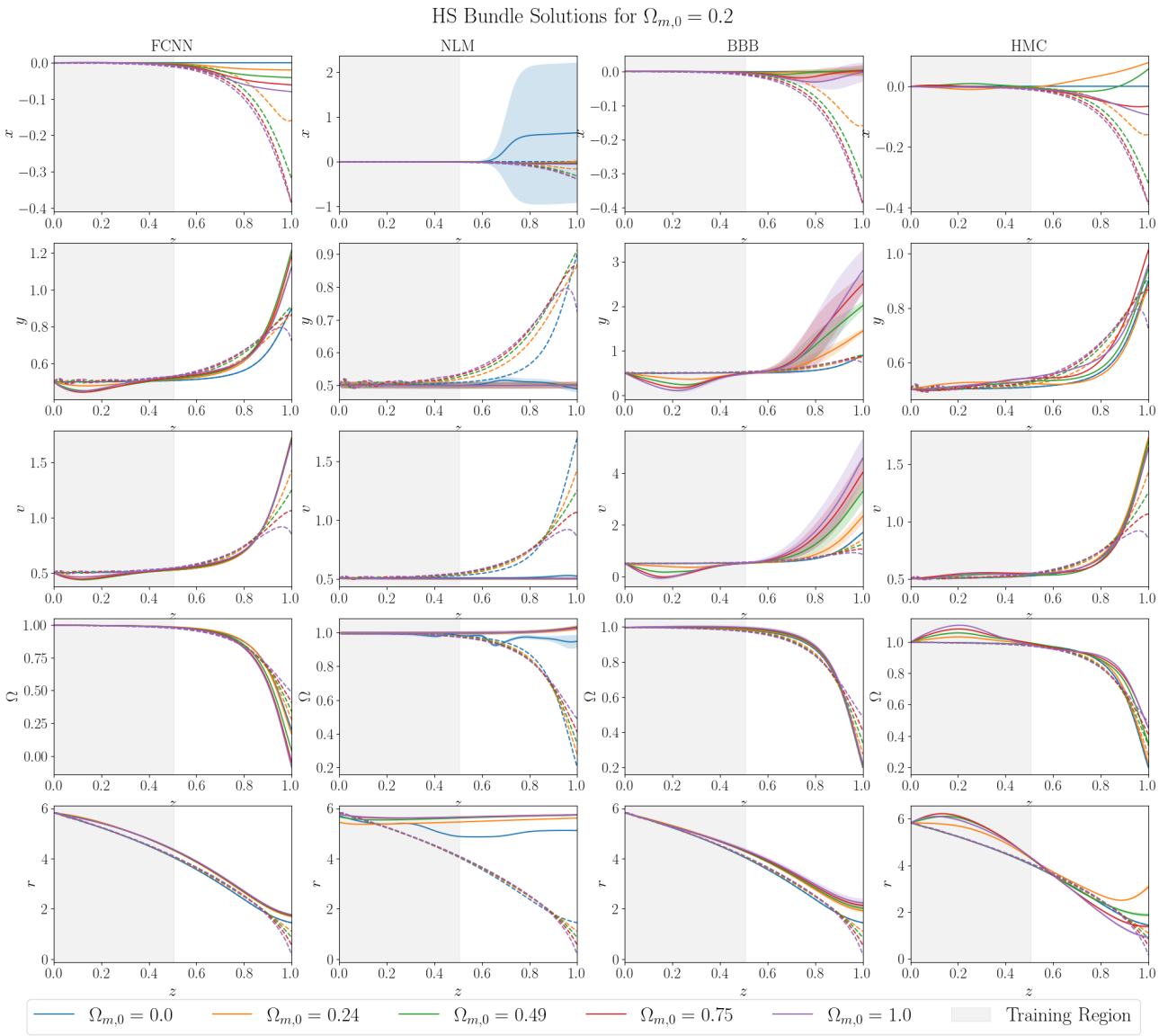

Figure 32: Examples Of HS Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.2$. Numerical Solutions Are Presented In Dotted Lines.

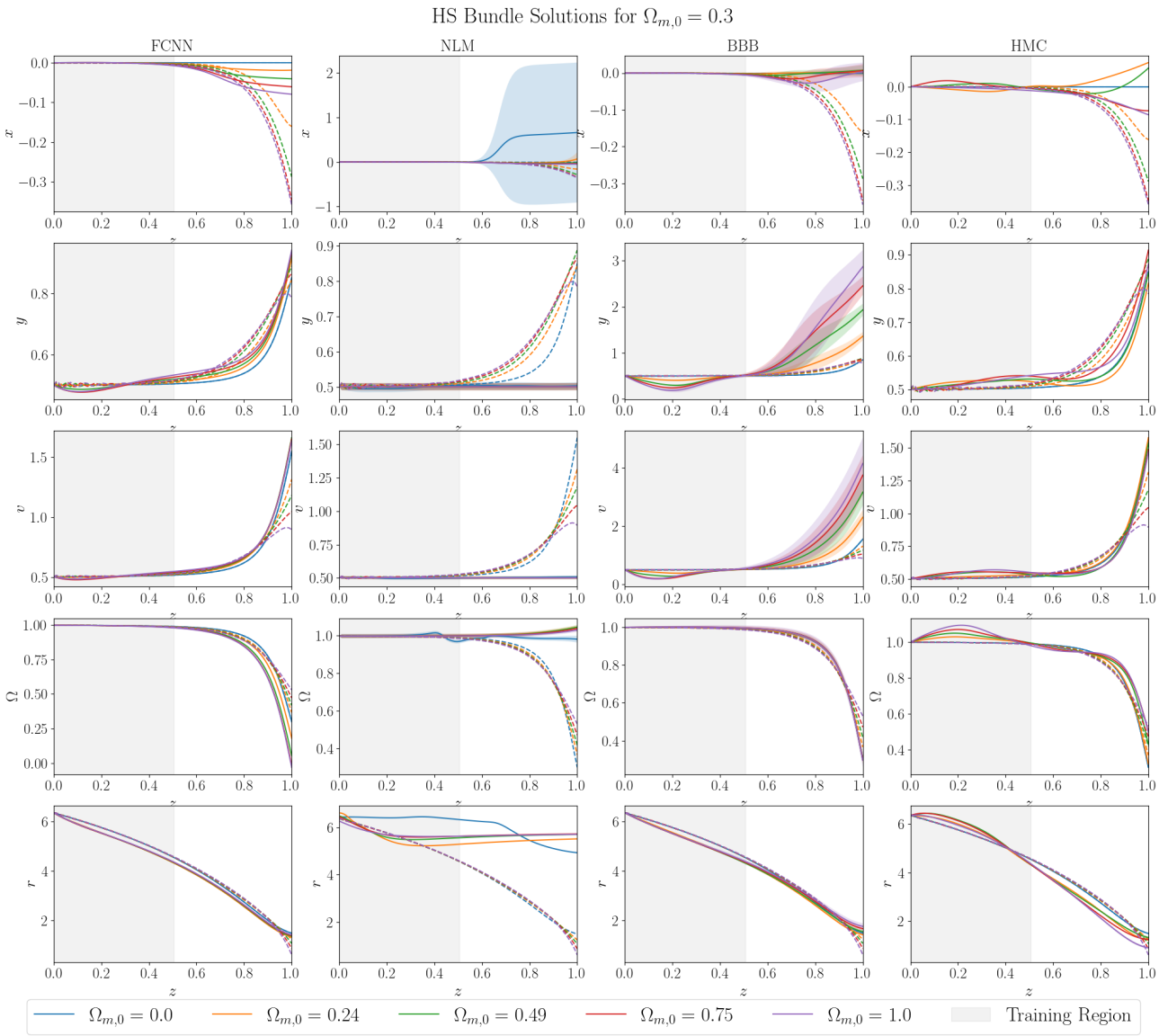

Figure 33: Examples Of HS Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.3$. Numerical Solutions Are Presented In Dotted Lines.

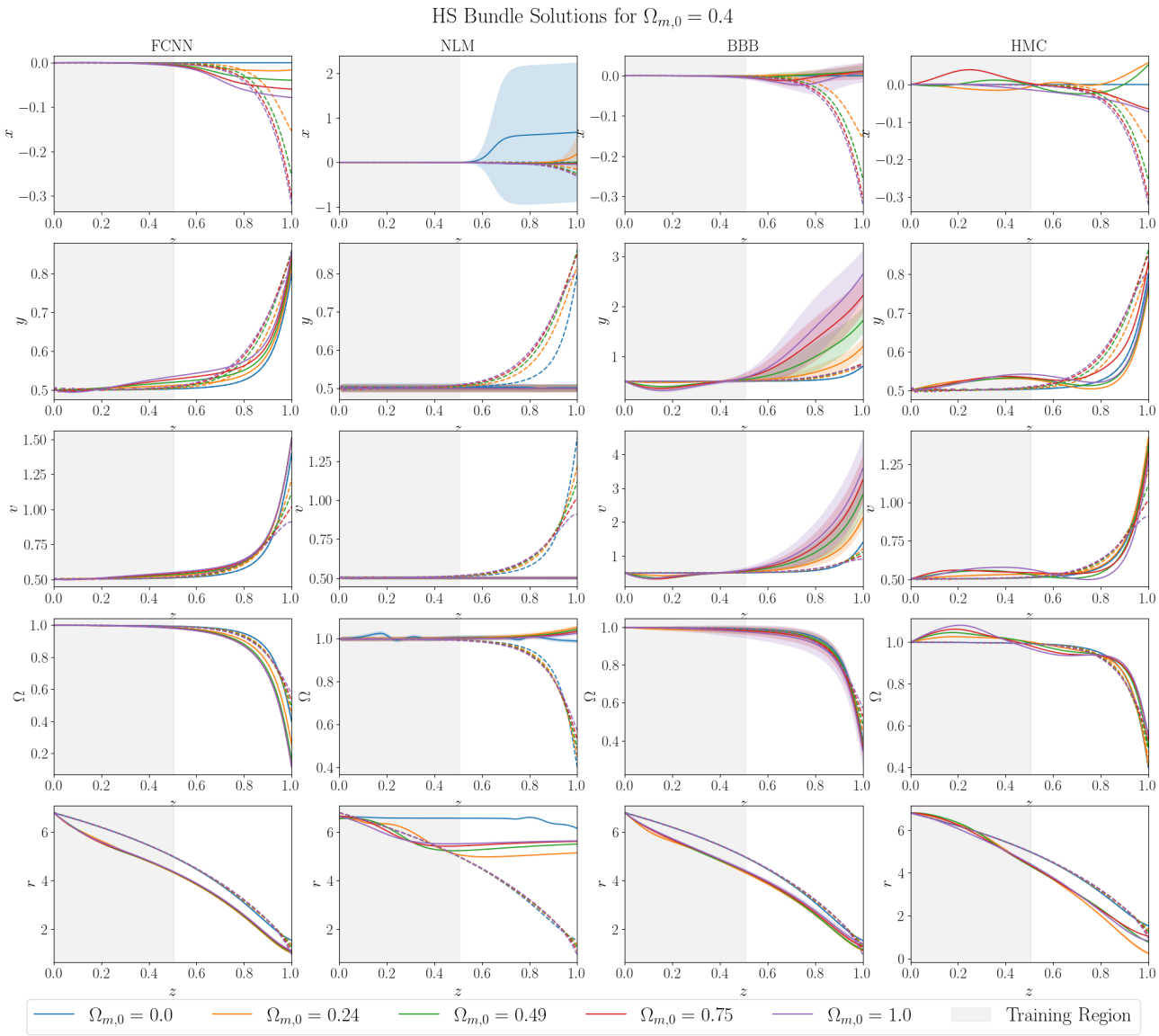

Figure 34: Examples Of HS Bayesian Solutions Obtained Using The Bundle Solver For The Parameter Value $\Omega_{m,0} = 0.4$. Numerical Solutions Are Presented In Dotted Lines.

Table 13: Error Quantiles Of The NN Solutions Relative To The Analytical Solution For ΛCDM And CPL, And The Numerical Solution For Quintessence And HS.

Equation	Method	Forward										Bundle									
		Q10	Q20	Q30	Q40	Q50	Q60	Q70	Q80	Q90	Q100	Q10	Q20	Q30	Q40	Q50	Q60	Q70	Q80	Q90	Q100
ΛCDM	FCNN	0.0	0.0	0.0	0.0	0.0	0.162	0.394	0.558	0.669	0.746	0.0	0.0	0.0	0.001	0.001	0.017	0.248	0.48	0.647	0.819
	NLM	0.0	0.0	0.001	0.001	0.007	0.184	0.418	0.577	0.685	0.759	0.001	0.002	0.003	0.006	0.018	0.058	0.32	0.554	0.701	0.843
	BBB	0.0	0.0	0.0	0.001	0.004	0.162	0.397	0.56	0.671	0.748	0.0	0.0	0.001	0.001	0.002	0.017	0.236	0.47	0.639	0.814
	HMC	0.001	0.001	0.002	0.004	0.019	0.228	0.443	0.593	0.696	0.767	0.001	0.001	0.002	0.003	0.007	0.051	0.212	0.415	0.597	0.799
	NLM + EB	0.0	0.0	0.0	0.0	0.0	0.17	0.406	0.569	0.679	0.754	0.0	0.0	0.001	0.001	0.002	0.017	0.184	0.381	0.537	0.717
	BBB + EB	0.0	0.0	0.0	0.0	0.0	0.156	0.391	0.559	0.666	0.75	0.001	0.002	0.004	0.005	0.007	0.025	0.254	0.49	0.657	0.835
	HMC + EB	0.0	0.0	0.0	0.0	0.004	0.202	0.432	0.58	0.69	0.765	0.0	0.0	0.001	0.001	0.002	0.033	0.152	0.353	0.541	0.78
CPL	FCNN	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.003	0.006	0.013	0.0	0.0	0.0	0.001	0.001	0.017	0.248	0.48	0.647	0.819
	BBB	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.003	0.006	0.011	0.0	0.001	0.002	0.006	0.073	0.302	0.684	2.283	98.738	9.53e+08
	HMC	0.0	0.0	0.0	0.0	0.001	0.008	0.022	0.041	0.066	0.095	0.001	0.002	0.004	0.007	0.033	0.13	0.363	1.711	100.003	8.31e+08
	BBB + EB	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.004	0.009	0.002	0.004	0.007	0.014	0.034	0.239	0.743	2.333	21.77	9.01e+08
	HMC + EB	0.0	0.0	0.0	0.0	0.0	0.01	0.038	0.078	0.124	0.173	0.002	0.004	0.006	0.01	0.021	0.305	0.999	28.567	6.67e+05	3.88e+26
Quint.	FCNN	0.0	0.0	0.0	0.0	0.001	0.006	0.039	0.103	0.171	0.223	0.001	0.001	0.002	0.004	0.007	0.012	0.021	0.039	0.077	0.26
	NLM	0.003	0.006	0.009	0.016	0.034	0.058	0.094	0.16	0.237	0.288	0.014	0.027	0.042	0.065	0.096	0.145	0.211	0.325	0.708	164.163
	BBB	0.001	0.002	0.003	0.005	0.012	0.027	0.067	0.129	0.197	0.918	0.001	0.003	0.005	0.008	0.013	0.02	0.034	0.057	0.104	24.699
	HMC	0.012	0.02	0.026	0.043	0.065	0.135	0.218	0.29	0.348	1.31	0.001	0.003	0.005	0.008	0.015	0.026	0.044	0.071	0.119	18.676
HS	FCNN	0.0	0.0	0.0	0.0	0.0	0.162	0.394	0.558	0.669	0.746	0.0	0.0	0.0	0.001	0.001	0.017	0.248	0.48	0.647	0.819
	NLM	0.003	0.007	0.01	0.015	0.026	0.905	8.203	23.657	40.486	147.148	0.078	0.135	0.182	0.213	0.247	0.635	3.403	16.619	88.025	1.33e+09
	BBB	0.002	0.003	0.003	0.005	0.008	0.022	0.031	0.042	0.119	0.831	0.039	0.089	0.143	0.23	0.429	0.715	1.434	3.463	9.211	7.52e+05
	HMC	0.121	0.305	0.521	0.674	0.742	1.308	2.71	7.066	16.963	59.125	0.067	0.123	0.169	0.217	0.294	0.368	0.629	2.291	8.942	9.06e+05

Table 14: Metrics of the NN Forward Solutions.

Equation	Method	MRE	Mean Residual	Miscal. Area	RMS Cal.	MA Cal.	Sharpness	NLL	CRPS	Check	Interval	Acc. MAE	Acc. RMSE	Acc. MDAE	Acc. MARPD
ΛCDM	FCNN	**0.216**	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.955	0.154	0.491	0.566	0.486	**0.101**	27843.2	20.019	10.01	208.426	20.073	28.172	12.83	187.161
	HMC	0.361	3.127	0.491	0.566	0.486	0.516	319.717	11.214	5.612	113.07	11.453	19.039	2.687	49.016
	NLM + 2S	0.226	6.544	0.224	0.258	0.222	0.546	247.862	9.743	4.876	98.293	9.939	17.917	0.023	32.164
	BBB + 2S	0.29	3.807	0.185	0.222	0.183	0.3	1100.191	11.292	5.648	116.159	11.41	19.662	0.459	41.833
	HMC + 2S	0.23	3.842	0.203	0.235	0.201	0.634	135.058	9.739	4.875	97.661	9.977	17.878	0.075	32.568
	NLM + 2S + EB	0.221	6.623	0.148	0.186	0.147	17.23	**−0.472**	**5.948**	**3.004**	**25.502**	9.792	17.737	**0.005**	**31.357**
	BBB + 2S + EB	0.266	3.69	0.114	**0.137**	0.113	17.265	2.216	6.958	3.513	30.458	11.066	19.178	0.855	38.671
	HMC + 2S + EB	0.222	3.841	**0.111**	0.138	**0.11**	17.251	-0.208	5.953	3.006	25.627	**9.788**	**17.707**	0.055	31.387
CPL	FCNN	0.002	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.008	0.014	0.359	0.394	0.355	0.023	-3.196	0.006	0.003	0.036	0.006	0.011	0.001	0.766
	HMC	0.009	0.014	0.268	0.288	0.266	0.016	-4.458	0.004	0.002	0.02	0.007	0.012	**0.0**	0.905
	BBB + 2S	0.036	0.026	**0.103**	**0.114**	**0.102**	0.019	-2.284	0.021	0.01	0.123	0.026	0.043	0.004	3.444
	HMC + 2S	0.016	0.019	0.267	0.3	0.265	0.034	-3.87	0.008	0.004	0.041	0.012	0.02	0.001	1.592
	BBB + 2S + EB	0.021	0.015	0.115	0.137	0.114	**0.007**	-2.376	0.013	0.006	0.093	0.015	0.027	0.001	1.993
	HMC + 2S + EB	0.158	0.257	0.389	0.443	0.385	0.07	5.18	0.093	0.047	0.7	0.11	0.292	**0.0**	10.464
Quintessence	FCNN	**0.044**	**0.054**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.838	0.109	0.482	0.553	0.477	0.025	163.756	0.127	0.064	1.113	0.14	0.171	0.135	nan
	HMC	0.06	0.07	0.176	0.204	0.174	0.01	-4.034	**0.013**	**0.006**	0.08	**0.015**	**0.03**	**0.0**	nan
	NLM + 2S	0.073	0.257	0.19	0.23	0.188	0.015	-3.317	0.017	0.008	0.109	0.021	0.041	**0.0**	**9.054**
	BBB + 2S	0.201	0.147	0.128	0.181	0.126	**0.004**	40.068	0.036	0.018	0.352	0.038	0.068	0.002	nan
	HMC + 2S	0.072	0.084	**0.124**	**0.142**	**0.123**	0.016	**−4.948**	0.014	0.007	**0.068**	0.019	0.036	**0.0**	nan
HS	FCNN	**0.216**	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	1.224	0.215	0.487	0.56	0.482	0.013	2094.883	0.324	0.162	3.284	0.33	0.378	0.347	nan
	HMC	0.268	0.128	0.44	0.512	0.436	0.013	723.293	0.138	0.069	1.326	0.144	0.169	0.148	nan
	NLM + 2S	0.353	0.324	**0.152**	**0.184**	**0.151**	0.409	**−1.279**	0.149	0.075	0.862	0.197	0.341	0.012	**35.335**
	BBB + 2S	0.158	0.326	0.219	0.253	0.216	**0.012**	59.506	0.028	0.014	0.239	0.031	0.068	0.007	nan
	HMC + 2S	0.262	0.212	0.265	0.304	0.263	0.021	90.284	**0.024**	**0.012**	**0.182**	**0.028**	**0.066**	**0.006**	nan

Table 15: Metrics of the NN Bundle Solutions.

Equation	Method	MRE	Mean Residual	Miscal. Area	RMS Cal.	MA Cal.	Sharpness	NLL	CRPS	Check	Interval	Acc. MAE	Acc. RMSE	Acc. MDAE	Acc. MARPD
ΛCDM	FCNN	0.178	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.94	0.199	0.491	0.566	0.486	**0.076**	67151.127	24.692	12.347	257.601	24.73	36.873	14.224	181.346
	HMC	0.196	5.169	0.4	0.455	0.396	0.822	139.806	10.309	5.161	102.634	10.61	22.142	0.752	25.336
	NLM + 2S	0.206	8.278	0.183	0.211	0.182	0.45	604.068	12.215	6.111	124.943	12.369	25.205	0.032	29.855
	BBB + 2S	0.204	5.386	0.143	0.173	0.141	0.487	935.006	11.829	5.918	121.119	11.965	24.362	0.092	28.573
	HMC + 2S	0.171	5.446	0.123	0.148	0.122	1.4	39.06	10.346	5.182	100.22	10.809	22.873	**0.011**	23.62
	NLM + 2S + EB	**0.145**	9.714	0.063	0.074	0.063	25.004	**−0.532**	**5.74**	**2.898**	**26.943**	**9.357**	**20.003**	0.015	**19.212**
	BBB + 2S + EB	0.198	5.17	**0.05**	**0.058**	**0.05**	25.008	1.647	7.004	3.536	31.722	11.421	23.284	0.294	26.712
	HMC + 2S + EB	0.168	5.104	0.098	0.122	0.097	25.033	-0.243	6.36	3.212	28.814	10.597	22.323	0.029	23.026
CPL	FCNN	**0.062**	**0.001**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.683	1410.627	0.2	0.222	0.198	5777.129	-2.261	3545.896	1777.486	33236.555	3770.007	76032.127	0.005	28.948
	HMC	0.362	1722.466	0.317	0.341	0.314	1.05e+05	**−6.292**	**1324.918**	**668.959**	**7053.056**	**1969.836**	**41625.811**	**0.0**	**13.055**
	BBB + 2S	2.751	818.487	0.15	**0.164**	0.149	2121.18	8.885	5054.062	2529.02	51204.587	5148.735	96419.681	0.004	37.724
	HMC + 2S	15.533	2213.885	0.255	0.272	0.252	2.92e+05	-5.211	2891.45	1459.777	18215.13	2540.452	45826.617	**0.0**	31.464
	BBB + 2S + EB	2.09e+05	9.58e+08	0.177	0.212	0.175	2.85e+09	26693.418	2.68e+08	1.35e+08	2.10e+09	3.09e+08	1.23e+10	0.003	61.241
	HMC + 2S + EB	2.35e+05	7.49e+05	**0.145**	0.165	**0.143**	2.89e+06	90.949	2.74e+05	1.38e+05	2.06e+06	3.20e+05	1.03e+07	0.001	47.367
Quintessence	FCNN	0.024	0.027	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.128	0.079	0.148	0.193	0.147	**0.004**	7.61e+09	0.017	0.009	0.159	0.019	0.041	0.002	nan
	HMC	**0.009**	**0.019**	0.147	0.165	0.146	0.007	7.61e+09	**0.002**	**0.001**	**0.009**	**0.002**	**0.007**	**0.0**	nan
	NLM + 2S	0.39	0.33	0.136	0.158	0.135	0.006	**51.905**	0.039	0.019	0.375	0.04	0.099	0.002	**24.652**
	BBB + 2S	0.159	0.12	0.119	0.165	0.118	**0.004**	7.61e+09	0.027	0.014	0.265	0.029	0.064	0.002	nan
	HMC + 2S	0.041	0.037	**0.048**	**0.054**	**0.047**	0.014	8.50e+09	0.005	0.003	0.026	0.007	0.02	**0.0**	nan
HS	FCNN	0.178	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-
	BBB	0.663	0.136	0.449	0.518	0.445	0.015	5.90e+12	0.264	0.132	2.649	0.27	0.343	0.227	nan
	HMC	0.408	0.152	0.455	0.52	0.45	0.041	5.90e+12	**0.126**	**0.064**	**1.003**	0.145	**0.193**	0.125	nan
	NLM + 2S	3740.057	2.27e+06	**0.315**	**0.363**	**0.312**	0.027	**20707.105**	0.38	0.19	3.926	0.384	0.551	0.264	**41.024**
	BBB + 2S	0.417	0.361	0.396	0.454	0.392	0.049	5.90e+12	0.2	0.101	1.824	0.216	0.349	0.113	nan
	HMC + 2S	2.739	3.456	0.486	0.561	0.482	**0.003**	5.98e+12	0.136	0.068	1.401	**0.137**	0.201	**0.096**	nan