Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CosmoFit: A Graphical User Interface for Bayesian Cosmological Parameter Estimation with User-Defined H(z) Models

GIOVANY CRUZ HUERFANO

giovanycruzh@gmail.com

Universidad Veracruzana https://orcid.org/0000-0002-2877-6922

Research Article

Keywords: Cosmological parameter estimation, Bayesian inference, graphical user interface, user-defined H(z) models, Cobaya, GetDist, observational cosmology

Posted Date: July 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10437914/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

CosmoFit: A Graphical User Interface for Bayesian Cosmological Parameter Estimation with User-Defined $H(z)$ Models

Giovany Cruz*

We present CosmoFit, a desktop graphical interface for Bayesian parameter inference in background cosmology. The application is designed to simplify cosmological analyses by reducing the need to manually program each step, while maintaining the transparency and reproducibility of the results. Users can define an expansion history $H(z)$, specify fixed and sampled parameters, assign prior ranges and proposal scales, select the available observational likelihoods, and run Markov chain Monte Carlo analyses from a unified graphical interface. CosmoFit uses Cobaya for Bayesian sampling and GetDist for posterior analysis and visualization. The current version supports cosmic-chronometer data and the official Pantheon+, Pantheon+SH0ES, and Union3 supernova likelihoods available through Cobaya. For each run, the application stores the configuration files, metadata, execution logs, chains, and numerical summaries required to inspect and reproduce the analysis. The performance of CosmoFit was evaluated through unit tests, a parameter-recovery test using synthetic data, and an independent comparison with an analysis performed directly in Cobaya. The posterior mean values obtained with both procedures agree to within less than one percent. In addition, the synthetic-data test recovers the fiducial parameters within the expected credible regions. As an illustrative example, we perform a spatially flat wCDM analysis using Pantheon+SH0ES and obtain marginalized constraints and triangle plots directly from the interface. Within its current scope, CosmoFit provides an accessible, transparent, and reproducible tool for research and academic applications involving explicit models of the background cosmological expansion.

I. INTRODUCTION

The increasing precision and diversity of cosmological observations have made statistical inference an essential component of modern cosmology. Measurements obtained from Type Ia supernovae, cosmic chronometers, baryon acoustic oscillations, the cosmic microwave background, and the large-scale distribution of matter are routinely used to constrain cosmological parameters and test competing descriptions of the expansion history of the Universe. In this context, Bayesian inference and Markov chain Monte Carlo methods provide a systematic framework for estimating model parameters, quantifying statistical uncertainties and parameter degeneracies, and evaluating the compatibility between theoretical predictions and observational data [1, 2].

Several mature computational packages have been developed to support cosmological parameter estimation. Codes such as CosmoMC [2], MontePython [3, 4], and Cobaya [5] provide flexible environments for defining cosmological models, observational likelihoods, prior distributions, and sampling strategies. In particular, Cobaya offers a modular framework capable of combining theoretical calculations, external likelihoods, and different sampling algorithms through configuration-based analyses. These tools are extensively used in cosmology because they separate the statistical machinery from the physical implementation of the model and facilitate the construction of reproducible inference pipelines.

Despite these advantages, the direct use of general-purpose inference frameworks usually requires familiarity with programming languages, package management, command-line tools, configuration files, likelihood implementations, and posterior-processing utilities. For experienced users, this flexibility constitutes one of their main strengths. However, it may also represent an important entry barrier for students, educators, and researchers who wish to test a background cosmological model without first implementing a complete computational pipeline. This situation is particularly relevant for models whose primary theoretical prediction is a modified expansion law ($H(z)$), while their observational analysis relies on standard background probes such as Type Ia supernovae or cosmic-chronometer measurements.

A graphical interface can reduce this technical barrier, but a scientifically useful interface must do more than hide command-line instructions. It must preserve the physical and statistical assumptions of the analysis, validate the mathematical model before sampling, record the exact configuration passed to the inference engine, isolate long numerical calculations from the graphical process, and retain the files required to reproduce the results independently. Furthermore, allowing users to introduce mathematical expressions presents an additional software-design challenge. A naive implementation based on the unrestricted evaluation of user-provided text could allow arbitrary code execution

* giocruz@uv.mx

and compromise both security and reproducibility. A reliable application therefore requires a restricted and explicitly validated mathematical-expression system.

In this work, we present CosmoFit, an open-source desktop graphical interface for Bayesian parameter inference in homogeneous and isotropic background cosmology. CosmoFit allows users to define a cosmological expansion history through a mathematical expression for $(H(z))$, declare fixed and sampled parameters, assign prior intervals, select a supported observational dataset, execute a Markov chain Monte Carlo analysis, and inspect the resulting posterior constraints. The numerical inference is performed by Cobaya, whereas GetDist [6] is used to obtain marginalized parameter summaries and one- and two-dimensional posterior distributions. CosmoFit is therefore not intended to replace these established packages. Instead, it provides an accessible, transparent, and reproducible layer for configuring and executing background-level cosmological analyses.

A central feature of CosmoFit is its restricted parser for user-defined expressions. Each expression is parsed and validated before an inference run is created. It can also be evaluated numerically over a selected redshift interval in order to identify undefined domains, singularities, complex values, or non-finite predictions before the sampling stage. This approach provides flexibility in the definition of cosmological models while avoiding unrestricted execution of user-supplied instructions.

The graphical configuration is translated automatically into a valid Cobaya input. The resulting sampling process is launched in an isolated worker process, preventing long calculations or numerical failures from blocking the graphical event loop. Each analysis is stored in a dedicated output directory containing the original user input, a normalized internal configuration, the generated Cobaya configuration, runtime metadata, execution status, logs, posterior chains, and numerical summaries. Consequently, a run initiated from the graphical interface can be inspected, repeated, or reconstructed independently outside the application.

The current release of CosmoFit focuses on late-time background cosmology and assumes spatially flat distance relations. It supports direct measurements of $(H(z))$ from cosmic chronometers and the official Cobaya likelihood components associated with the Pantheon+, Pantheon+SH0ES, and Union3 Type Ia supernova compilations [7–9]. Cosmological perturbations, Boltzmann-code calculations, and structure-growth observables are outside the scope of the initial version. This restricted scope is intentional: the main objective is to provide a reliable environment for testing standard and non-standard late-time expansion histories defined directly through $(H(z))$.

The scientific reliability of the graphical workflow is evaluated through several complementary verification and validation procedures. These include unit and integration tests of the mathematical parser, configuration builder, likelihood layer, execution worker, and output system; synthetic parameter-recovery experiments with known fiducial cosmological parameters; and direct comparisons between analyses generated by CosmoFit and independently constructed Cobaya calculations. These tests are designed to verify that the graphical configuration layer does not modify the underlying statistical problem and that the complete workflow reproduces reference posterior constraints.

The principal contributions of the present work are therefore:

1. A desktop graphical workflow for configuring Bayesian analyses of user-defined cosmological expansion histories;
2. A restricted mathematical-expression parser that avoids the unrestricted execution of user-provided code.
3. Automatic and transparent generation of reproducible Cobaya configurations.
4. Isolated execution of Markov chain Monte Carlo analyses with persistent status and logging information.
5. Integrated posterior inspection and visualization through GetDist.
6. Support for cosmic chronometers and official Cobaya Type Ia supernova likelihoods.
7. Validation through synthetic parameter recovery and comparisons with independent Cobaya calculations.

The source code, documentation, validation material, and reproducible examples are publicly available through the project repository: `https://github.com/GiovanyCruz/cosmofit`

II. COSMOLOGICAL FRAMEWORK AND MODEL IMPLEMENTATION

A. Cosmological Background and Expansion History

CosmoFit is designed to study the background evolution of homogeneous, isotropic, and spatially flat universes. Within this framework, the spacetime geometry is described by the Friedmann–Lemaître–Robertson–Walker metric,

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

Here, $a(t)$ is the scale factor and c is the speed of light. The evolution of the scale factor determines the expansion rate of the Universe, which is characterized by the Hubble parameter,

$$H(t) = \frac{\dot{a}(t)}{a(t)}, \tag{2}$$

where the dot denotes differentiation with respect to cosmic time. The cosmological redshift is related to the scale factor through

$$1 + z = \frac{a_0}{a(t)}, \tag{3}$$

where a_0 is the present value of the scale factor. Throughout this work, the normalization $a_0 = 1$ is adopted. In terms of redshift, a general expansion history can be written as

$$H(z) = H_0 E(z; \boldsymbol{\theta}), \tag{4}$$

where H_0 is the present value of the Hubble parameter, $E(z; \boldsymbol{\theta})$ is the dimensionless expansion function, and $\boldsymbol{\theta}$ denotes the set of cosmological parameters. The main theoretical input required by CosmoFit is an explicit mathematical expression for $H(z)$. The parameters appearing in this expression may be defined either as fixed quantities or as sampled parameters. For each sampled parameter, the user specifies a prior interval, a reference value, and, when required, a proposal scale through the graphical interface. This approach is particularly suitable for models whose background dynamics can be expressed in the form

$$H(z) = H(z; \theta_1, \theta_2, \ldots, \theta_N). \tag{5}$$

Therefore, the application can be used with both standard cosmological models and phenomenological or effective proposals. When a model requires the simultaneous numerical solution of a system of differential equations, its solution must first be reduced to an explicit function of redshift before it can be incorporated into the CosmoFit workflow.

B. User-Defined Expansion Models

The spatially flat ΛCDM model provides a simple example of how an expansion history can be introduced into the application. Neglecting the radiation contribution at late times, the Hubble parameter is given by

$$H_{\Lambda\mathrm{CDM}}(z) = H_0\sqrt{\Omega_m(1+z)^3 + 1 - \Omega_m}, \tag{6}$$

where Ω_m is the present matter-density parameter. In CosmoFit, this expansion history can be entered as

```
H0*sqrt(Om*(1+z)**3 + 1-Om)
```

where `H0` and `Om` are declared as model parameters. CosmoFit is not restricted to the ΛCDM model. The user may introduce any expression that is compatible with the mathematical syntax accepted by the application. This flexibility allows the study of effective dark-energy models[10], modified gravity theories [11, 12], braneworld scenarios [13, 14], and other non-standard cosmologies, provided that their background predictions can be expressed explicitly through $H(z)$.

C. Cosmological Observables

The expansion history can be confronted with different late-time cosmological observables. In the case of Type Ia supernovae, the comparison with observational data is performed through cosmological distances. For a spatially flat universe, the line-of-sight comoving distance is

$$D_C(z) = c \int_0^z \frac{dz'}{H(z')}. \tag{7}$$

In this case, the transverse comoving distance coincides with $D_C(z)$, and the luminosity distance is given by

$$D_L(z) = (1+z) D_C(z). \tag{8}$$

The corresponding theoretical distance modulus is

$$\mu_{\rm th}(z) = 5 \log_{10} \left[\frac{D_L(z)}{\rm Mpc} \right] + 25. \tag{9}$$

These relations connect the expansion history defined by the user with the Type Ia supernova likelihoods supported by the application.

Cosmic chronometers, in contrast, provide direct measurements of the Hubble parameter at different redshifts. For these data, the intermediate calculation of cosmological distances is not required. The theoretical prediction is obtained by directly evaluating

$$H_{\rm th}(z_i) = H(z_i; \boldsymbol{\theta}), \tag{10}$$

at each observed redshift z_i. Thus, the same expression for $H(z)$ can be used both to construct integrated observables, such as the luminosity distance, and to compare the model directly with measurements of the expansion rate.

D. Numerical Domain and Scope of the Implementation

For an expansion history to be physically and numerically consistent, $H(z)$ must remain real, finite, and positive throughout the redshift interval used in the analysis. Parameter combinations that generate singularities, complex values, undefined operations, or non-finite results cannot be used to evaluate the observational likelihood correctly. Before starting the sampling process, CosmoFit checks the syntax of the expression and performs a preliminary numerical evaluation using the reference parameter values supplied by the user. This procedure allows common input errors to be identified before a numerical run is created, including incorrect parentheses, unsupported functions, undefined domains, and non-finite values. The preliminary evaluation does not guarantee that every region explored during sampling will be physically valid. Therefore, whenever a parameter combination produces an inadmissible prediction, the corresponding model evaluation is rejected by assigning a non-finite likelihood value.

The application does not compute scalar, vector, or tensor cosmological perturbations, nor does it use Boltzmann solvers such as CLASS or CAMB. These restrictions clearly define the scope of the first release. CosmoFit is not intended to replace general cosmological inference tools operating at the perturbation level. Its purpose is to provide an accessible and transparent environment for defining, validating, and confronting background expansion histories with late-time observational data.

III. BAYESIAN INFERENCE FRAMEWORK

CosmoFit connects the expansion history defined by the user with observational data through a Bayesian analysis. Once the function $H(z)$, the model parameters, and the datasets have been specified, the objective is to determine which parameter values provide the best description of the observations.

A. Posterior distribution

Let $\boldsymbol{\theta}$ denote the set of parameters of a cosmological model $\mathcal{M}$, and let D represent the observational data. In the Bayesian approach, the information available before the analysis is described by the prior, while the agreement between the model and the data is quantified by the likelihood. These two ingredients are combined through Bayes' theorem,

$$P(\boldsymbol{\theta} \mid D, \mathcal{M}) = \frac{\mathcal{L}(D \mid \boldsymbol{\theta}, \mathcal{M})\pi(\boldsymbol{\theta} \mid \mathcal{M})}{\mathcal{Z}(D \mid \mathcal{M})}, \tag{11}$$

where $\mathcal{L}$ is the likelihood, π is the prior, and $\mathcal{Z}$ is the Bayesian evidence. Since the current version of CosmoFit is intended for parameter estimation within a fixed model, the evidence acts only as a normalization constant. Therefore,

$$P(\boldsymbol{\theta} \mid D, \mathcal{M}) \propto \mathcal{L}(D \mid \boldsymbol{\theta}, \mathcal{M})\pi(\boldsymbol{\theta} \mid \mathcal{M}). \tag{12}$$

The posterior distribution therefore identifies the combinations of parameters that are compatible with the chosen priors and, at the same time, provide an adequate description of the observational data.

B. Likelihoods and priors

The likelihood measures the agreement between the predictions of the model and the observations. For Gaussian data with covariance matrix $\mathbf{C}$, this agreement can be expressed as

$$\chi^2(\boldsymbol{\theta}) = [\boldsymbol{d} - \boldsymbol{t}(\boldsymbol{\theta})]^{\mathrm{T}}\, \mathbf{C}^{-1}\, [\boldsymbol{d} - \boldsymbol{t}(\boldsymbol{\theta})]\,, \qquad \ln \mathcal{L} = -\frac{1}{2}\chi^2 + \text{constant}, \tag{13}$$

where $\boldsymbol{d}$ represents the observational data and $\boldsymbol{t}(\boldsymbol{\theta})$ is the corresponding theoretical prediction. When statistically independent datasets are used, their contributions to the total log-likelihood are added.

In CosmoFit, the cosmic-chronometer likelihood is evaluated directly within the application, while the official supernova likelihoods are handled through their corresponding Cobaya components. The construction of these likelihoods is described in Sec. V.

Each model parameter can be defined as either fixed or sampled. For sampled parameters, the user specifies a uniform prior, a reference value, and a proposal scale. The choice of prior ranges is important: an interval that is too narrow may artificially truncate the posterior, while an excessively broad interval may reduce sampling efficiency or include regions with no physical relevance.

C. MCMC sampling

For most cosmological models, the posterior distribution cannot be obtained analytically. Markov chain Monte Carlo methods are therefore used to sample it numerically [15, 16].

CosmoFit does not implement its own sampler. The configuration selected through the interface is converted into a valid Cobaya input, and the MCMC calculation is performed using its established implementation [5]. In this way, the interface simplifies the preparation of the analysis without modifying the statistical procedure used by Cobaya.

For each proposed parameter set $\boldsymbol{\theta}$, the theoretical component evaluates

$$H(z; \boldsymbol{\theta}), \tag{14}$$

and computes the quantities required by the selected likelihood. Parameter combinations that produce undefined, complex, or non-finite values are rejected automatically.

The calculation is executed in a process separated from the graphical interface. This allows the application to remain responsive during sampling and enables the user to inspect the run status, review diagnostic messages, or cancel the execution.

D. Posterior analysis and convergence

Once the sampling process has finished, the chains are analyzed with GetDist [6]. From the results panel, CosmoFit provides marginalized means, standard deviations, credible intervals, one-dimensional distributions, two-dimensional contours, and triangle plots. Before these results are calculated, the user may discard an initial fraction of the stored chain. This removes samples that may still be influenced by the starting point. The ignored fraction, the selected credible level, and the plotting options are stored together with the results of the analysis.

Convergence must be checked before interpreting the posterior constraints. One of the diagnostics used by Cobaya is the Gelman–Rubin statistic [17], which compares the variation within the chains and between them. Convergence is approached as

$$R - 1 \longrightarrow 0. \tag{15}$$

However, this quantity should not be considered in isolation. Since consecutive samples in an MCMC chain are generally correlated, the number of effectively independent samples may be smaller than the total chain length. For this reason, convergence, chain length, and the stability of the marginalized results should be examined together.

E. Reproducibility

For each run, CosmoFit preserves the model expression, parameter definitions, prior intervals, selected likelihoods, the generated Cobaya input, metadata, execution logs, chains, and posterior summaries. Two independent MCMC runs are not expected to generate exactly the same sequence of samples, even when the same configuration is used. In this context, reproducibility means that the same model, data, priors, and sampler settings lead to statistically compatible posterior distributions. For this reason, the validation procedure compares posterior means, standard deviations, credible intervals, convergence information, and the recovery of fiducial values, rather than requiring the chains to agree sample by sample.

IV. DESIGN AND IMPLEMENTATION OF COSMOFIT

CosmoFit was developed as a desktop application for configuring, running, and inspecting Bayesian analyses of background cosmological models. Its purpose is to simplify the preparation of a Cobaya analysis without hiding the physical assumptions, parameter definitions, or numerical configuration on which the results depend.

The application allows the user to define an explicit expression for $H(z)$, declare fixed and sampled parameters, assign prior intervals and reference values, select an observational likelihood, configure the sampler, and inspect the resulting chains through GetDist. Each stage is represented explicitly in the graphical workflow, while the generated configuration files and numerical outputs remain available for independent inspection.

A. Software architecture

The software is organized as a set of connected modules rather than as a single graphical program. Its general workflow can be summarized as

$$\begin{aligned} \text{user input} &\longrightarrow \text{validation} \longrightarrow \text{Cobaya configuration} \\ &\longrightarrow \text{sampling} \longrightarrow \text{GetDist analysis}. \end{aligned} \tag{16}$$

The graphical interface is implemented with PySide6, while the numerical calculations are delegated to established scientific packages, including NumPy, SciPy, Cobaya, and GetDist. This separation keeps the interface focused on model configuration, validation, execution control, and result inspection.

Internally, the information entered through the graphical widgets is converted into a normalized configuration that is independent of the interface itself. This representation contains the model expression, parameter definitions, priors, likelihood selection, sampler options, and output settings. It can be stored, tested, and translated into the final input required by Cobaya.

B. Model definition and validation

The user defines the expansion history by entering an explicit mathematical expression for $H(z)$. For example, the spatially flat ΛCDM model can be written as

```
H0*sqrt(Om*(1+z)**3 + 1-Om)
```

with `H0` and `Om` declared in the parameter table. Each parameter may be fixed at a chosen value or sampled within a specified prior interval. The expression is interpreted through a restricted mathematical parser. The accepted syntax includes the redshift variable z, declared parameter names, numerical constants, arithmetic operators, parentheses, and a limited set of mathematical functions. Symbols or operations outside this supported set are rejected before the analysis is launched. The parser first converts the expression into an abstract syntax tree and then builds the numerical function used during sampling. This makes it possible to check the structure of the model before passing it to the inference engine. The application also evaluates the model at the supplied reference values over a selected redshift interval. This preliminary test can reveal divisions by zero, invalid square roots or logarithms, overflows, complex results, and non-finite values. Although this check cannot guarantee that every point visited by the sampler will be valid, it detects many common configuration errors before numerical execution begins. Invalid parameter combinations encountered during sampling are assigned a non-finite likelihood and rejected.

C. Parameter configuration and Cobaya input

Each parameter is stored in a normalized form. A fixed parameter keeps the same value throughout the run, whereas a sampled parameter is assigned a uniform prior,

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}, \tag{17}$$

together with a reference value and a proposal scale. Before the configuration is accepted, CosmoFit verifies that the prior bounds are properly ordered and that the reference value lies inside the selected interval. The validated definitions are then translated into the parameter block used by Cobaya. The complete translation follows the sequence

$$\text{graphical state} \longrightarrow \text{normalized configuration} \longrightarrow \text{Cobaya input}. \tag{18}$$

The generated input includes the fixed and sampled parameters, prior bounds, reference values, proposal scales, likelihood components, external theoretical function, sampler settings, output prefix, and package paths required by external datasets. Since this file is stored with the run, the calculation can be inspected or repeated independently of the graphical application.

D. Likelihood integration

The current implementation supports an internal cosmic-chronometer likelihood and the official Pantheon+, Pantheon+SH0ES, and Union3 supernova likelihoods available through Cobaya.
For cosmic chronometers, CosmoFit reads a table with the columns

```
z,H,sigma
```

and evaluates the user-defined model at the observed redshifts. The data loader checks the column names, array dimensions, finite entries, and positive uncertainties before the likelihood is constructed.

For the supernova compilations, the corresponding likelihood calculations are delegated to the official Cobaya components. Their external data packages are installed separately and are not redistributed with CosmoFit. This avoids duplicating likelihood implementations that are already maintained within the Cobaya ecosystem. The observational datasets and their statistical treatment are discussed in Sec. V.

E. Execution and posterior analysis

MCMC calculations may require many evaluations of the theoretical model and can remain active for an extended period. To prevent the graphical interface from becoming unresponsive, the analysis is executed in a worker process separate from the main application. The worker receives the generated configuration, runs Cobaya, writes the chains and diagnostic messages, and records whether the calculation finished successfully. This separation also makes it possible to cancel an active run and to examine the reason for a failure after the interface has been closed. Once the sampling process is complete, the chains can be loaded and analyzed with GetDist. The user may calculate marginalized means, standard deviations, and credible intervals, as well as generate one-dimensional, two-dimensional, and triangle plots. Because posterior processing is separated from sampling, the same chains can be analyzed repeatedly with different ignored fractions, confidence levels, parameter selections, or plotting options without running the MCMC calculation again. The numerical summaries are stored in machine-readable form, so the graphical plots are not the only record of the posterior results. They can therefore be used in independent comparisons, tables, or validation scripts.

F. Run outputs and reproducibility

Each analysis is stored in a dedicated directory containing the scientific configuration and the products generated during execution. A typical run includes

```
input.yaml
normalized_config.json
cobaya_input.yaml
metadata.json
status.json
summary.json
logs/
chains/
```

The file `input.yaml` records the scientific choices entered by the user, while `normalized_config.json` contains their validated internal representation. The file `cobaya_input.yaml` stores the configuration passed to Cobaya. Runtime information and execution state are written to `metadata.json` and `status.json`, whereas `summary.json` contains the posterior statistics generated after processing the chains. The directories `logs/` and `chains/` store diagnostic messages and MCMC samples, respectively. This structure allows a completed or failed analysis to be examined without depending on the temporary state of the interface. The model expression, parameter ranges, selected likelihood, sampler configuration, execution status, and posterior products remain associated with the same run. Reproducibility in an MCMC calculation does not require two independent runs to generate identical chains. Rather, analyses performed with the same model, data, priors, and sampler settings should lead to statistically compatible posterior distributions. Fixed random seeds are used in the reference validation workflows whenever they are supported by the underlying calculation.

G. Distribution and requirements

The current version of CosmoFit is distributed as source code through its public repository. Running the application requires Git, Python 3.12, and the project dependencies, including PySide6, Cobaya, GetDist, NumPy, SciPy, and PyYAML. Testing and development additionally use tools such as pytest and Ruff.
A standard installation can be performed with

```
git clone https://github.com/GiovanyCruz/cosmofit.git
cd cosmofit
python3.12 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install -e .
```

After installation, the interface can be launched with

```
cosmofit
```

or with

```
python -m cosmofit.ui
```

The external data packages required by the official supernova likelihoods must be installed through Cobaya. Additional installation, dataset, and troubleshooting instructions are provided in the project documentation.

V. OBSERVATIONAL DATASETS AND LIKELIHOODS

The current version of CosmoFit includes observational probes of the late-time expansion of the Universe. It supports direct measurements of the Hubble parameter from cosmic chronometers and the official Cobaya likelihoods for the Pantheon+, Pantheon+SH0ES, and Union3 Type Ia supernova compilations. These probes are complementary. Cosmic chronometers constrain the expansion rate $H(z)$ directly, whereas Type Ia supernovae constrain the luminosity-distance–redshift relation. Both can therefore be connected to a user-defined background expansion history without requiring the calculation of cosmological perturbations.

A. Cosmic chronometers

The cosmic-chronometer method estimates the Hubble parameter from the differential age evolution of passively evolving galaxies [18–21]. It is based on the relation

$$H(z) = -\frac{1}{1+z}\frac{dz}{dt}, \tag{19}$$

which allows $H(z)$ to be inferred without first integrating the expansion history to obtain a cosmological distance. For a set of measurements $\{z_i, H_{\rm obs}(z_i), \sigma_{H,i}\}$, CosmoFit evaluates the user-defined model at the observed redshifts and computes

$$\chi^2_{\rm CC}(\boldsymbol{\theta}) = \sum_i \frac{\left[H_{\rm obs}(z_i) - H(z_i;\boldsymbol{\theta})\right]^2}{\sigma^2_{H,i}}. \tag{20}$$

This expression corresponds to statistically independent measurements. The normalization of the Gaussian likelihood is constant when the uncertainties do not depend on the cosmological parameters and therefore does not affect the relative posterior probabilities. The graphical workflow accepts tabular files with the columns

```
z,H,sigma
```

where H and σ are expressed in $\mathrm{km\,s^{-1}\,Mpc^{-1}}$. Before the likelihood is created, the loader checks that the required columns are present, the arrays have consistent lengths, all entries are finite, and the uncertainties are positive. This internal likelihood is also used in the synthetic-recovery and independent-comparison tests described in Sec. VI.

B. Type Ia supernovae

Type Ia supernovae are standardizable candles that constrain the relation between luminosity distance and redshift. For the spatially flat cosmologies considered in the present release, the theoretical luminosity distance is

$$D_L(z;\boldsymbol{\theta}) = c(1+z)\int_0^z \frac{dz'}{H(z';\boldsymbol{\theta})}. \tag{21}$$

The corresponding distance modulus follows from

$$\mu_{\rm th}(z;\boldsymbol{\theta}) = 5\log_{10}\left[\frac{D_L(z;\boldsymbol{\theta})}{\rm Mpc}\right] + 25. \tag{22}$$

The precise construction of the supernova likelihood depends on the selected compilation, including its covariance matrix, calibration information, and treatment of nuisance parameters. Rather than reproducing these procedures, CosmoFit provides the background predictions required by the official likelihood components distributed through Cobaya [5]. This choice avoids maintaining independent copies of complex observational likelihoods and makes it possible to compare a run configured through CosmoFit with a direct Cobaya calculation using the same component.

1. Pantheon+

Pantheon+ contains 1701 light curves corresponding to 1550 distinct Type Ia supernovae over an approximate redshift range $0.001 < z < 2.26$. The compilation incorporates improved photometric calibration, redshift information, peculiar-velocity corrections, selection-bias corrections, and systematic covariance modelling [7, 22]. It is accessed through the official component

```
sn.pantheonplus
```

In the absence of an external absolute calibration, this likelihood mainly constrains relative distances and the shape of the expansion history. The overall normalization remains degenerate with the supernova absolute magnitude, so a Pantheon+-only analysis should not normally be interpreted as an independent measurement of H_0.

2. Pantheon+SH0ES

Pantheon+SH0ES combines the Pantheon+ sample with Cepheid-host calibration information from the SH0ES distance ladder [7, 8]. This additional information breaks the relative-distance normalization degeneracy and allows the absolute expansion scale to be constrained within the assumptions of the likelihood. The corresponding component is

```
sn.pantheonplusshoes
```

CosmoFit preserves the distinction between Pantheon+ and Pantheon+SH0ES because the two likelihoods do not contain the same information. The calibration and nuisance-parameter treatment are those implemented by the official Cobaya component.

3. Union3

Union3 contains approximately 2000 cosmologically useful Type Ia supernovae analyzed with the UNITY hierarchical Bayesian framework [9]. It combines observations from several surveys using a common treatment of calibration, selection effects, population distributions, and systematic uncertainties. Within CosmoFit, the likelihood is selected through

```
sn.union3
```

As in the Pantheon+ cases, CosmoFit supplies the expansion and distance predictions, while the observational and statistical treatment remains that of the official likelihood.

C. External packages and model compatibility

The supernova datasets are not redistributed with CosmoFit. Their official data packages must be installed through Cobaya, for example,

```
cobaya-install \
  sn.pantheonplus \
  sn.pantheonplusshoes \
  sn.union3 \
  --packages-path "$COBAYA_PACKAGES_PATH"
```

The selected likelihood and the package path are recorded in the generated run configuration, while the observational files remain subject to their original distribution, citation, and licensing conditions. The theoretical quantity required by the likelihood depends on the selected probe. Cosmic chronometers use $H(z;\boldsymbol{\theta})$ directly, whereas supernova likelihoods require the luminosity distance obtained by integrating $1/H(z)$. Consequently, a model is compatible with the present likelihoods only if it produces a real, positive, and finite expansion rate over the full redshift range of the selected dataset. The preliminary model preview checks the expression at a finite set of redshifts and reference parameter values. During sampling, the likelihood performs the required evaluations over the actual dataset range. Parameter combinations that produce invalid predictions are assigned a non-finite log-likelihood and rejected.

D. Current observational scope

The present release is restricted to background observables. It does not yet include baryon acoustic oscillations, cosmic microwave background likelihoods, redshift-space distortions, weak lensing, structure-growth measurements, or perturbation-level predictions from CLASS or CAMB. This scope is consistent with the current design of the application, whose main theoretical input is an explicit expression for $H(z)$. Additional background probes and perturbation-level interfaces may be incorporated in future versions.

VI. VERIFICATION AND NUMERICAL VALIDATION

The reliability of CosmoFit was assessed at both the software and scientific levels. The validation program combines automated tests, synthetic-data recovery, and comparisons with independently configured Cobaya analyses. Together, these checks cover the complete workflow,

$$\begin{aligned}\text{model definition} &\longrightarrow \text{configuration} \longrightarrow \text{likelihood}\\ &\longrightarrow \text{sampling} \longrightarrow \text{posterior analysis.}\end{aligned} \tag{23}$$

The purpose of these tests is to verify that the graphical layer correctly translates the user configuration into the numerical problem executed by Cobaya, and that the resulting posterior distributions are consistent with independent calculations.

A. Automated tests and end-to-end execution

The repository includes unit and integration tests executed with pytest. They cover the expression parser, parameter validation, normalized configurations, generation of Cobaya inputs, loading of cosmic-chronometer data, likelihood evaluation, communication with the theoretical provider, isolated execution, run artifacts, and posterior processing with GetDist.
The complete test suite can be run with

```
pytest -q
```

One of the integration tests compares the cosmic-chronometer likelihood evaluated through the provider mechanism with a direct evaluation of the same model and dataset. Agreement between both values verifies that the theoretical prediction is passed correctly to the likelihood component.
The repository also includes a smoke test that executes a minimal flat ΛCDM analysis from configuration generation to posterior output:

```
python -m cosmofit.cobaya_engine.smoke_run
```

A successful execution creates the normalized configuration, the generated Cobaya input, metadata, logs, chains, status information, and the posterior summary. The reference smoke run completed with

$$\texttt{exit_code} = 0, \tag{24}$$

confirming that the principal components can operate together in a complete analysis.

B. Reference model and synthetic recovery

The principal validation model is the spatially flat ΛCDM background,

$$H(z) = H_0\sqrt{\Omega_m(1+z)^3 + 1 - \Omega_m}. \tag{25}$$

Within the interface, it is entered as

```
H0*sqrt(Om*(1+z)**3 + 1-Om)
```

A synthetic cosmic-chronometer dataset containing 15 redshift points was generated with the fiducial parameters

$$H_0^{\rm fid} = 70\,{\rm km\,s^{-1}\,Mpc^{-1}}, \qquad \Omega_m^{\rm fid} = 0.30. \tag{26}$$

The dataset is stored in

```
validation/data/
cosmic_chronometers_synthetic_recovery.csv
```

This experiment tests the complete sequence of data loading, model evaluation, likelihood calculation, MCMC sampling, chain processing, and posterior summarization.
For the Hubble constant, the recovered marginalized constraint was

$$H_0 = 71.061649 \pm 1.582522\,{\rm km\,s^{-1}\,Mpc^{-1}}. \tag{27}$$

The 68% and 95% credible intervals were, respectively,

$$\begin{aligned} &69.526532 < H_0 < 72.640599, \\ &67.858860 < H_0 < 74.096396, \end{aligned} \qquad {\rm km\,s^{-1}\,Mpc^{-1}}. \tag{28}$$

The fiducial value $H_0^{\rm fid} = 70\,{\rm km\,s^{-1}\,Mpc^{-1}}$ lies within both intervals. Its difference from the posterior mean is approximately 0.67 posterior standard deviations. The fiducial value $\Omega_m^{\rm fid} = 0.30$ was likewise recovered within the 68% and 95% marginalized credible regions. These results show that the complete graphical workflow can recover the known parameters of the synthetic reference cosmology.

C. Comparison with an independent Cobaya analysis

A more direct validation was carried out by comparing two independently constructed analyses:

1. a run generated and executed through CosmoFit;
2. a direct Cobaya run created without using the CosmoFit configuration builder.

Both calculations used the same flat ΛCDM model, cosmic-chronometer dataset, parameter definitions, priors, and posterior-processing prescription. The first 30% of each stored chain was discarded before calculating the marginalized statistics.
The resulting constraints are shown in Table I.

TABLE I: Comparison between the posterior constraints obtained with CosmoFit and an independently configured direct Cobaya run.

Parameter	CosmoFit mean	CosmoFit σ	Direct Cobaya mean	Direct Cobaya σ
$H_0\,[{\rm km\,s^{-1}\,Mpc^{-1}}]$	72.344391	1.102487	72.346278	1.036889
Ω_m	0.234969	0.015398	0.234712	0.014701

The relative differences between the posterior means were

$$\delta_\mu(H_0) = 0.0026\%, \qquad \delta_\mu(\Omega_m) = 0.1098\%, \tag{29}$$

while the differences between the posterior standard deviations were

$$\delta_\sigma(H_0) = 6.1324\%, \qquad \delta_\sigma(\Omega_m) = 4.6342\%. \tag{30}$$

The posterior means agree well below the one-percent level. The smaller differences in posterior width are compatible with the expected variation between finite MCMC realizations, including the effects of autocorrelation, burn-in selection, and numerical estimation of marginalized distributions.
The comparison can be reproduced with

```
validation/compare_cc_runs.py
```

and its numerical output is stored in

```
validation/results/cc_lcdm_comparison.txt
```

D. Official supernova likelihoods

The validation procedure was also applied to the official Pantheon+, Pantheon+SH0ES, and Union3 likelihood components. For each case, a CosmoFit-generated analysis was compared with an independently written Cobaya configuration using the same model, priors, likelihood, and posterior-processing choices.
The consolidated results and the complete protocol are stored in

```
validation/results/validation_summary.csv
validation/results/validation_summary.md
docs/validation_protocol.md
```

This extends the validation beyond the internal cosmic-chronometer likelihood and checks that the background quantities produced by CosmoFit can be passed consistently to the official external likelihood components.

E. Posterior processing and reproducibility

The same GetDist settings were used when comparing the CosmoFit and direct Cobaya chains. In the reference cosmic-chronometer analysis, an initial fraction

$$f_{\rm burn} = 0.30 \tag{31}$$

was excluded from both chains. A small diagnostic outlier fraction of approximately 1.4×10^{-3} was found during processing and had no appreciable effect on the central posterior constraints. The validation runs preserve the model expression, parameter definitions, priors, likelihood selection, sampler configuration, random seed when applicable, generated inputs, chains, logs, and numerical summaries. Fixed seeds are used whenever supported, although independent MCMC runs are not expected to produce identical chains. Reproducibility is therefore assessed through statistical agreement: analyses performed with the same scientific and numerical configuration should recover compatible posterior means, widths, credible intervals, and fiducial parameters. Overall, the validation program shows that the principal software components operate as intended, that the complete workflow recovers the parameters of a known synthetic cosmology, and that analyses configured through CosmoFit produce posterior constraints consistent with independent Cobaya calculations. These tests do not guarantee that every user-defined model is physically meaningful or numerically stable, but they show that the interface preserves the underlying inference problem for the models and likelihoods considered in this work.

VII. ILLUSTRATIVE wCDM WORKFLOW

To illustrate the complete workflow of CosmoFit, we consider a spatially flat wCDM model constrained with the calibrated Pantheon+SH0ES supernova likelihood. The aim of this example is to show how a user-defined expansion history is entered, validated, sampled, and analyzed within the same graphical environment.

Neglecting radiation at late times, the expansion history is

$$H(z) = H_0 \left[\Omega_m (1+z)^3 + (1-\Omega_m)\,(1+z)^{3(1+w)}\right]^{1/2}, \tag{32}$$

where H_0 is the present Hubble parameter, Ω_m is the present matter-density parameter, and w is the constant dark-energy equation-of-state parameter. The ΛCDM limit is recovered for $w = -1$.

A. Model and parameter configuration

The expansion history is entered directly in the model panel as

```
H0*sqrt(Om*(1+z)**3 + (1-Om)*(1+z)**(3*(1+w)))
```

The symbols `H0`, `Om`, and `w` are then declared in the parameter table. The model was previewed over the interval $0 \le z \le 2.3$, which approximately covers the redshift range relevant to the selected supernova sample. This interval is used only for numerical validation and does not impose a cut on the likelihood data. The validation step checks that the expression uses declared parameters and supported mathematical functions, and that it produces real and finite values for the reference configuration. Figure 1 shows the successfully validated model.

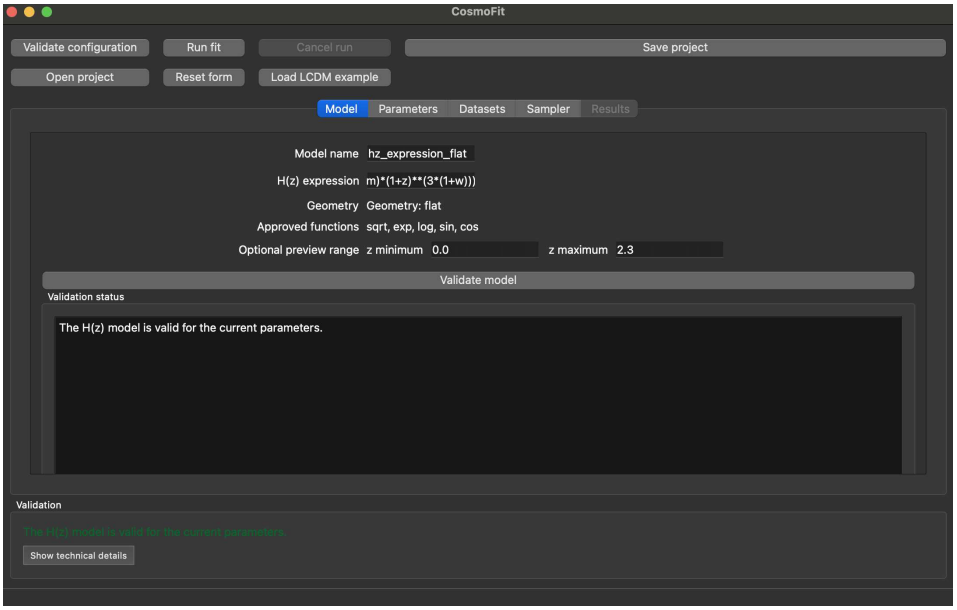


FIG. 1: Definition and validation of the spatially flat wCDM expansion history. The model is evaluated over an optional redshift interval before the remaining analysis settings are configured.

All three parameters were treated as sampled quantities. Their prior ranges, reference values, and initial proposal widths are displayed graphically in Fig. 2.

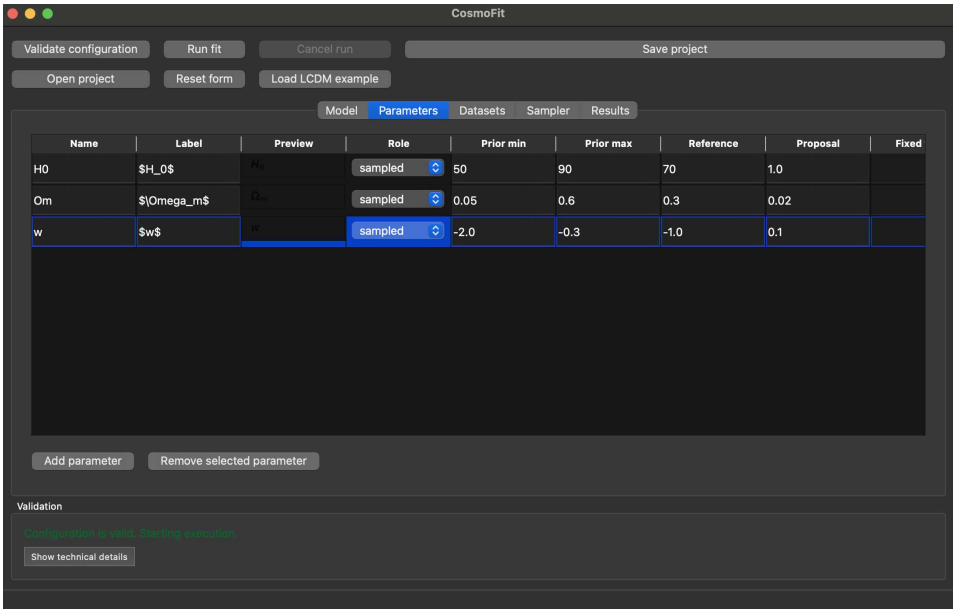


FIG. 2: Parameter and prior configuration for the illustrative analysis. The parameters H_0, Ω_m, and w are sampled with uniform priors, reference values, and initial proposal widths.

The explicit parameter names must match those appearing in the expression for $H(z)$, while the mathematical labels are used only in plots and numerical summaries.

B. Likelihood and sampler settings

The calibrated Pantheon+SH0ES likelihood was selected through the official Cobaya component

```
sn.pantheonplusshoes
```

as shown in Fig. 3. The corresponding external data are located through the local Cobaya packages directory. CosmoFit supplies the expansion history and luminosity-distance predictions, while the calibration and nuisance-parameter treatment remain those of the official likelihood.

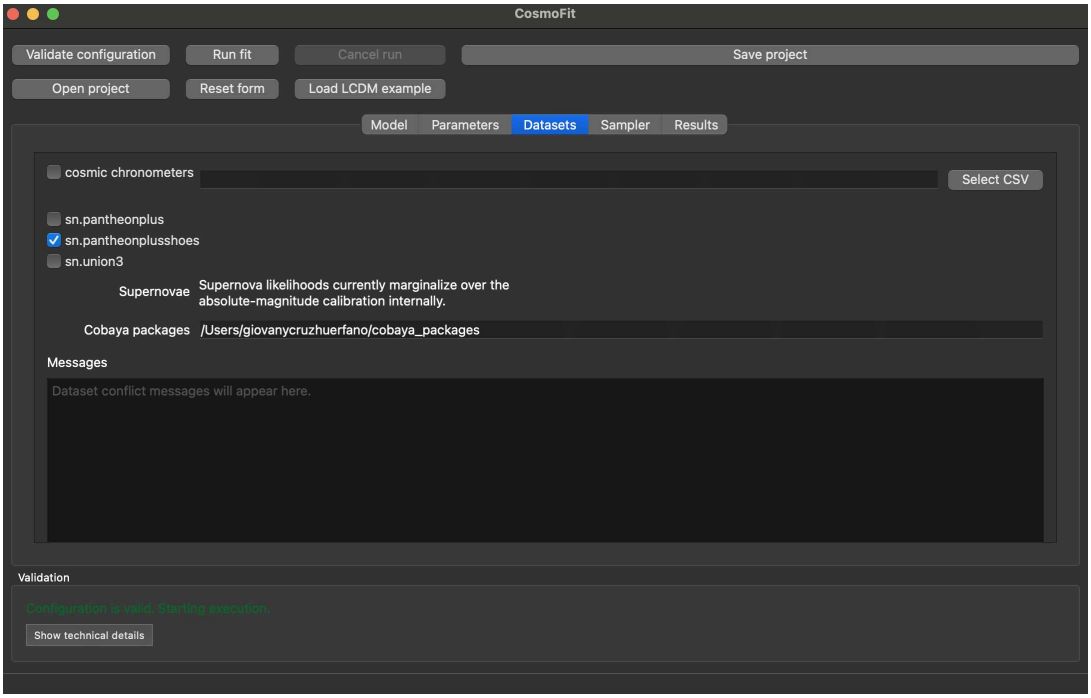


FIG. 3: Selection of the Pantheon+SH0ES likelihood for the illustrative wCDM run. The panel also displays the local Cobaya packages directory used to locate the external data.

The sampling calculation was performed with the MCMC implementation of Cobaya. The main settings were

$$s = 1234, \qquad N_{\max} = 10\,000, \qquad R - 1 < 0.01, \qquad R_{\mathrm{cl}} - 1 < 0.2. \tag{33}$$

Proposal learning was enabled so that the sampler could adapt to correlations among the parameters. The sampler-level burn-in was set to zero because the initial portion of the chain was removed later during posterior processing. The complete configuration is shown in Fig. 4.

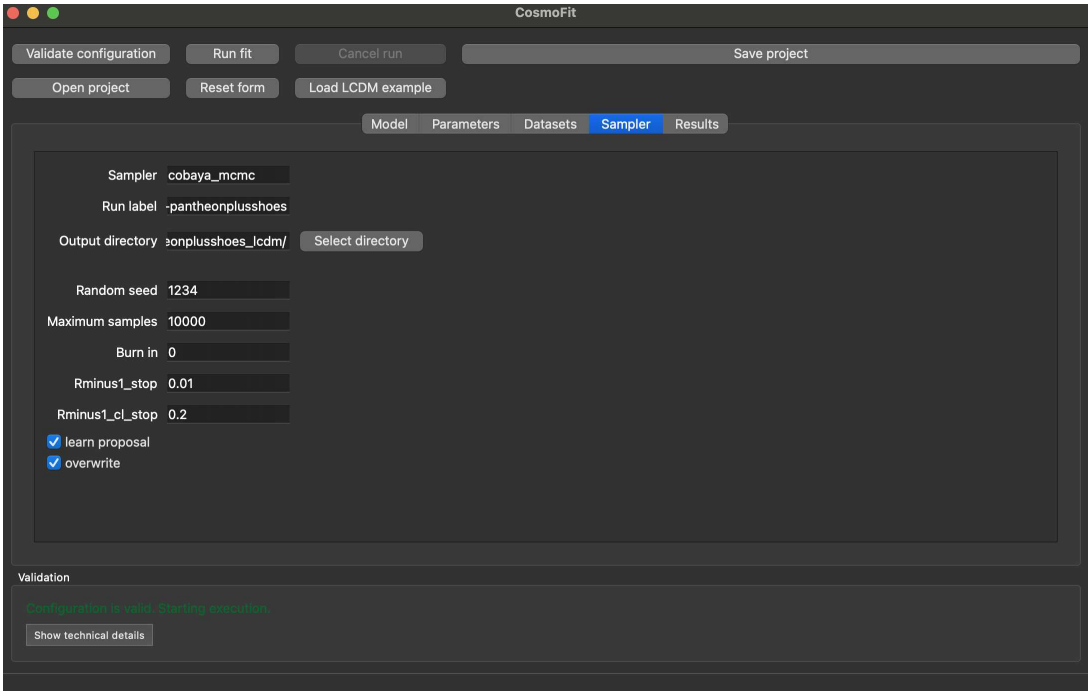


FIG. 4: Sampler configuration for the wCDM analysis. The panel records the random seed, maximum number of samples, convergence thresholds, output location, and proposal-learning option.

C. Execution of the analysis

After validating the complete configuration, the analysis is launched with the `Run fit` action. CosmoFit then creates the run directory, writes the generated Cobaya input, and starts the MCMC calculation in a worker process separated from the graphical interface.

During execution, the results panel reports the state as `Running`, displays the output directory, and shows diagnostic messages produced by the background-theory provider and the Cobaya worker. Posterior-processing actions remain disabled until the calculation has finished.

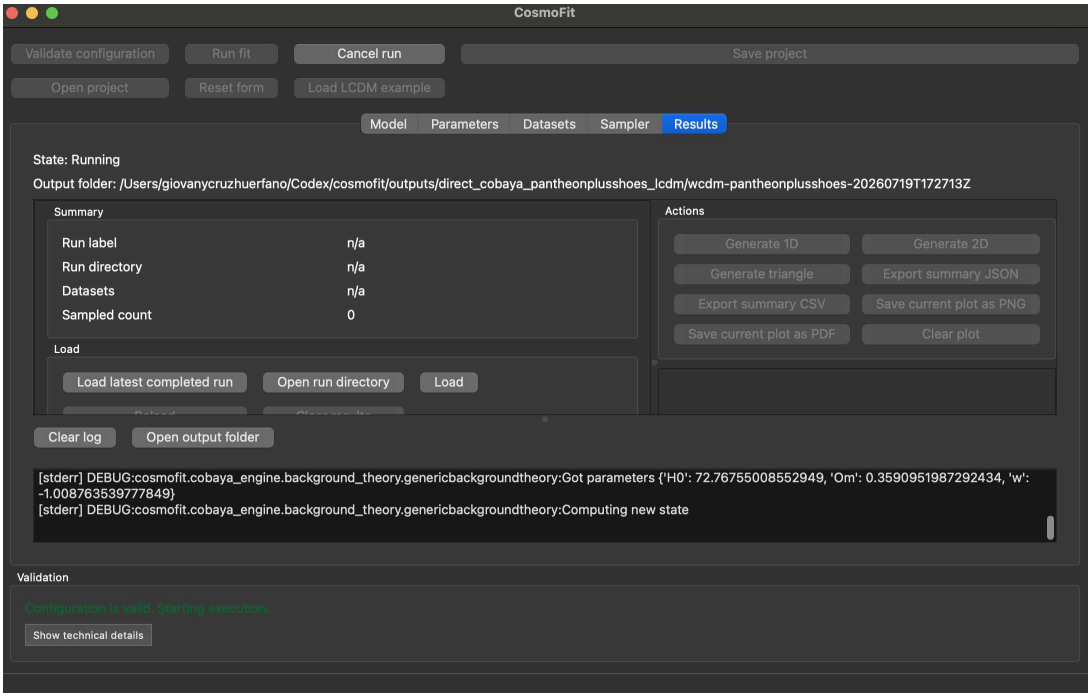


FIG. 5: Execution of the illustrative wCDM analysis after pressing `Run fit`. The interface reports the active run state, output directory, and diagnostic messages while posterior actions remain disabled.

The separate worker process keeps the interface responsive and allows the calculation to be cancelled without closing the application. The run finishes when the convergence conditions are satisfied or the maximum sample limit is reached.

D. Posterior processing

After successful completion, the run state changes to `Ready`. The reference execution ended after 2280 accepted MCMC steps. The chains were then processed with GetDist, discarding the first 30% of the stored samples and using a primary credible level of 68%.

Figure 6 shows the corresponding post-processing configuration. Filled contours were enabled, and the plot title and legend were assigned before generating the graphical products.

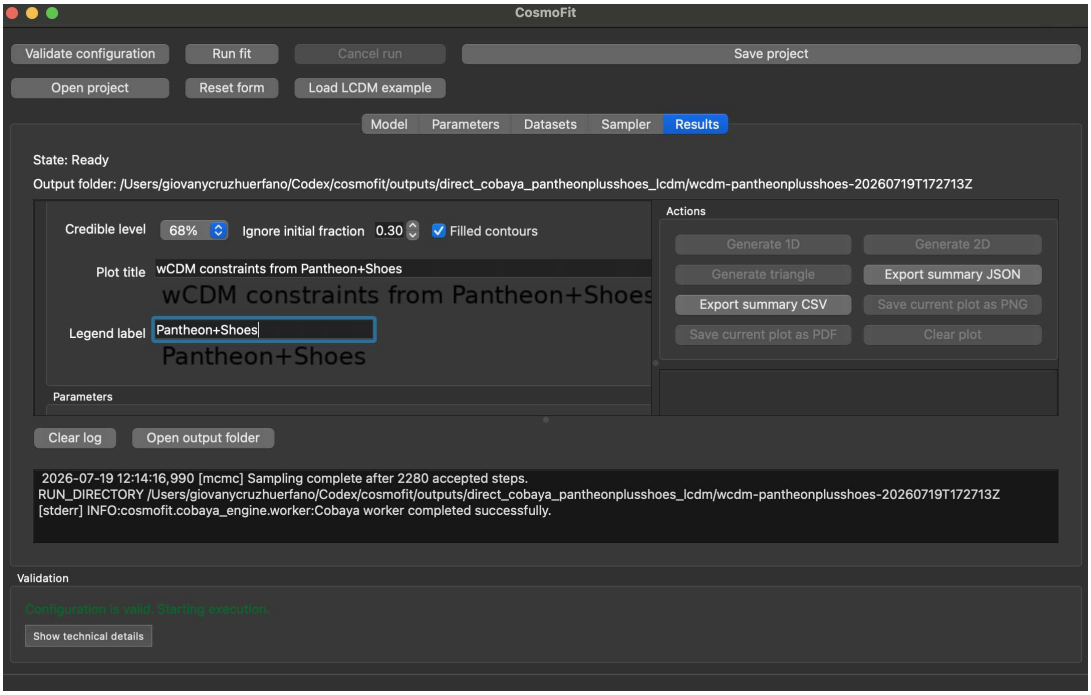


FIG. 6: Posterior-processing configuration after completion of the run. The first 30% of the chain is ignored, the primary credible level is set to 68%, and filled contours are enabled.

The user may select one parameter to obtain a one-dimensional marginalized distribution, two parameters to generate joint contours, or several parameters to produce a triangle plot. In the present example, H_0, Ω_m, and w were selected together.

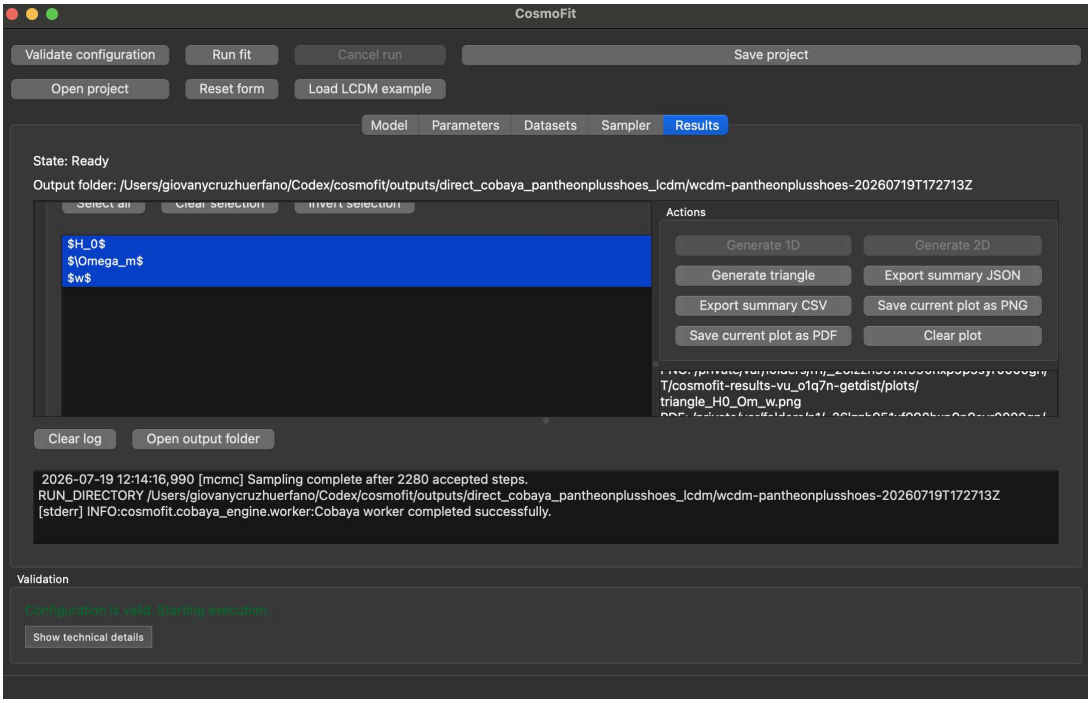


FIG. 7: Selection of H_0, Ω_m, and w for posterior visualization. The interface can generate one-dimensional, two-dimensional, and triangle plots, as well as export numerical summaries.

Because posterior processing is independent of the sampling stage, the same completed chains can be reanalyzed with different ignored fractions, credible levels, parameter selections, or plotting options without repeating the MCMC calculation.

E. Numerical and graphical results

After removing the initial fraction, 1596 sample rows remained in the stored chain. The marginalized numerical results are summarized in Table II.

TABLE II: Posterior summary for the illustrative spatially flat wCDM analysis with Pantheon+SH0ES.

Parameter	Mean	Standard deviation	Median
$H_0\,[\mathrm{km\,s^{-1}\,Mpc^{-1}}]$	73.4973	1.04405	73.4954
Ω_m	0.295457	0.0652577	0.299333
w	−0.927074	0.143111	−0.919027

The same quantities are displayed directly in the results panel, together with the retained sample count, number of loaded chains, ignored fraction, and a preview of the generated triangle plot.

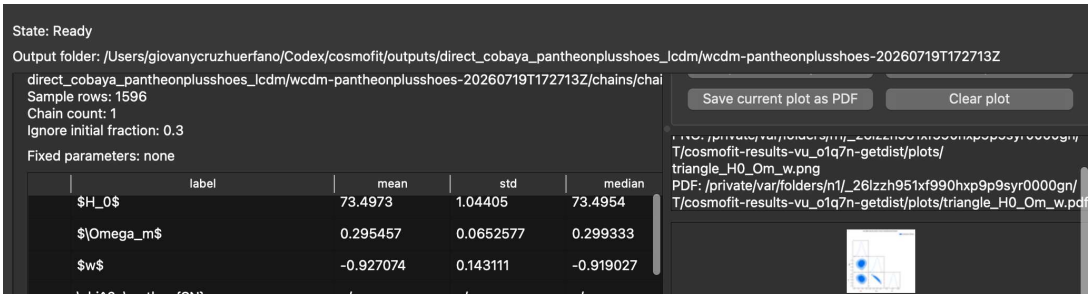


FIG. 8: Numerical posterior summary displayed by CosmoFit. The panel reports the retained sample count and the marginalized mean, standard deviation, and median of H_0, Ω_m, and w.

The publication-quality triangle plot is shown in Fig. 9. The diagonal panels contain the one-dimensional marginalized distributions, while the lower panels show the joint credible regions.

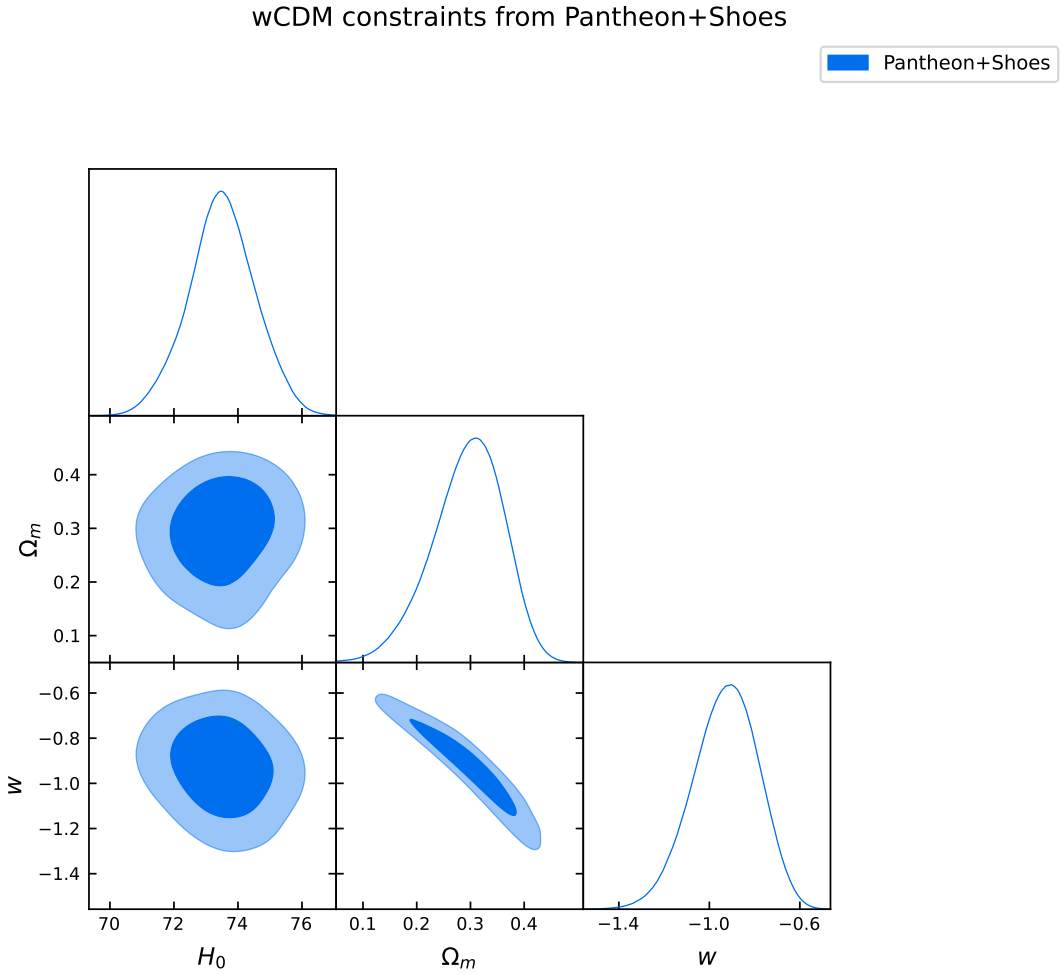


FIG. 9: Triangle plot generated by CosmoFit for the spatially flat wCDM model with Pantheon+SH0ES. The diagonal panels show the marginalized distributions of H_0, Ω_m, and w, and the lower panels show their joint credible regions.

The one-dimensional distributions are approximately unimodal, and the close agreement between their means and medians indicates only mild asymmetry. The most visible degeneracy appears in the Ω_m–w plane, where the elongated contour reflects the partial compensation between both parameters in the luminosity-distance relation.

The marginalized value

$$w = -0.927074 \pm 0.143111 \tag{34}$$

is compatible with the cosmological-constant value $w = -1$, differing from it by approximately 0.51 posterior standard deviations.

This result should be interpreted as an illustration of the complete graphical workflow rather than as a definitive cosmological constraint. The posterior summary was obtained from one stored chain, and the main purpose of the example is to show that CosmoFit can connect model definition, sampling, numerical summaries, and publication-quality posterior plots within a single reproducible analysis.

The numerical summaries can be exported in JSON or CSV format, while the plots can be saved as PNG or PDF. The PDF format is particularly suitable for publication because it preserves vector text, axes, and contour boundaries.

VIII. DISCUSSION

The results presented in the previous sections show that CosmoFit provides a practical graphical environment for defining, running, and analyzing Bayesian studies of background cosmology. Its main contribution is not a new sampling method or a new observational likelihood, but the integration of established tools into a transparent and reproducible workflow. By relying on Cobaya for MCMC sampling and on GetDist for posterior analysis, the application preserves the numerical procedures already used in cosmological inference. The interface simplifies the preparation of the analysis by allowing the user to specify the expansion history, parameters, priors, likelihoods, and sampler settings without manually writing the complete Cobaya configuration. This approach is particularly useful when the background evolution can be written explicitly as $H(z)$. Standard and non-standard expansion histories can be tested by changing the model expression and the corresponding parameter definitions, while the generated configuration remains available for inspection and independent execution. The interface therefore reduces repetitive technical work without hiding the scientific structure of the analysis. The complete procedure can be summarized as

$$\begin{aligned} &\text{theoretical model} \longrightarrow \text{parameters and priors} \longrightarrow \text{likelihood} \\ &\qquad \longrightarrow \text{sampling} \longrightarrow \text{posterior constraints.} \end{aligned} \tag{35}$$

Presenting these stages separately makes the relation between the physical model, the observational data, and the statistical assumptions easier to follow. It also helps identify configuration errors before a numerical run is started. This organization may also be useful in academic settings. Students who are beginning to work with cosmological inference can focus first on the physical meaning of the model, priors, likelihoods, and posterior distributions without being limited by an initial lack of experience with programming or configuration files. At the same time, the generated inputs and outputs remain accessible, so the interface can also serve as a transition toward direct command-line or script-based analyses. Transparency is therefore an essential part of the implementation. For each run, CosmoFit preserves the normalized configuration, the generated Cobaya input, metadata, logs, chains, and numerical summaries. The graphical interface is not the only record of the analysis, and every relevant choice can be inspected after the calculation has finished. The restricted mathematical parser contributes to the same goal. It accepts the redshift variable, declared parameters, supported operators, and a defined set of mathematical functions. This provides enough flexibility for a broad class of background models while keeping the model definition consistent with the internal numerical representation. The validation results indicate that the graphical layer preserves the underlying inference problem. In the cosmic-chronometer comparison, the posterior means obtained through CosmoFit agreed at the sub-percent level with those from an independently configured Cobaya analysis. The synthetic-recovery test also showed that the full workflow can recover known fiducial parameters within the expected credible regions. These checks are important because convenience alone is not sufficient for a scientific interface. The agreement with independent calculations supports the interpretation of CosmoFit as an orchestration layer over Cobaya, rather than as a separate statistical implementation. The illustrative wCDM example with Pantheon+SH0ES demonstrates the complete workflow, from the definition and validation of $H(z)$ to the generation of numerical summaries and triangle plots. It also shows that the same completed chains can be reprocessed with different plotting options and posterior settings without repeating the MCMC calculation. The present release is limited to background cosmology with spatially flat distance relations. It does not yet compute cosmological perturbations or connect directly to Boltzmann solvers such as CLASS or CAMB. In addition, the model must currently be supplied as an explicit function of redshift. Models defined only through coupled differential equations would require an additional numerical background solver. The current implementation also supports a limited set of priors and likelihoods. Future versions could include Gaussian,

log-uniform, and user-defined priors, together with additional background probes such as baryon acoustic oscillations. The modular separation among the graphical interface, expression parser, configuration builder, likelihood layer, execution worker, and posterior tools provides a natural basis for these extensions. New components can be added without changing the overall workflow, provided that the required theoretical quantities can be connected to the internal model representation. Overall, CosmoFit provides an intermediate solution between a fully manual inference pipeline and a closed black-box application. It reduces the initial burden of programming and configuration while preserving the scientific assumptions, numerical inputs, chains, logs, and posterior products required for inspection and reproducibility.

IX. CONCLUSIONS

We have presented CosmoFit, a desktop graphical interface designed to facilitate Bayesian parameter inference for background cosmological models. The application integrates model definition, parameter configuration, observational-likelihood selection, MCMC execution, and posterior analysis within a unified workflow based on Cobaya and GetDist. The interface allows users to define an explicit expansion history $H(z)$, specify fixed and sampled parameters, assign prior intervals and proposal scales, select supported observational datasets, and configure the MCMC sampler without manually constructing the complete underlying configuration files. The generated scientific inputs and outputs nevertheless remain available, preserving the transparency and reproducibility of the analysis. The implementation was verified through unit tests, a synthetic-parameter recovery experiment, and a direct comparison with an independently configured Cobaya analysis. The comparison showed sub-percent agreement between the recovered posterior means, while the synthetic test confirmed that the complete workflow can recover known fiducial parameters within the expected credible regions. These results support the interpretation of CosmoFit as a graphical orchestration layer that preserves the underlying statistical calculation. The illustrative flat wCDM analysis demonstrated the complete use of the application, from the graphical definition of the Hubble function to the generation of marginalized posterior distributions and numerical summaries. The example also showed how the same completed chains can be processed with different GetDist options and exported in graphical and tabular formats. The current release is focused on background cosmology with spatially flat distance relations and models that can be expressed through an explicit function $H(z)$. Within this scope, CosmoFit provides a practical environment for testing standard and non-standard expansion histories using supernova and cosmic-chronometer data. Its modular architecture also provides a basis for the future incorporation of additional priors, likelihoods, background observables, and numerical model components. Beyond its use in research workflows, the interface may also support academic activities by reducing the initial programming barrier associated with cosmological inference. This can allow users to focus first on the connection between the theoretical model, observational data, Bayesian assumptions, and posterior constraints, while retaining access to the generated files for more advanced independent analyses. Overall, CosmoFit offers an intermediate approach between direct manual configuration and a closed black-box application. It simplifies the construction and execution of background-level cosmological analyses while preserving the inputs, logs, chains, and posterior products required for inspection, validation, and reproducibility.

CODE AND DATA AVAILABILITY

The source code of CosmoFit is publicly available at

`https://github.com/GiovanyCruz/cosmofit`

The version associated with this work is CosmoFit v0.1.0. The repository includes the graphical interface, the cosmological background module, the Cobaya execution layer, the GetDist post-processing tools, unit tests, validation scripts, and example configuration files. The synthetic cosmic-chronometer dataset used in the parameter-recovery test is included in the repository under

`validation/data/`
`cosmic_chronometers_synthetic_recovery.csv`

The scripts and numerical outputs used for the independent comparison between CosmoFit and a direct Cobaya implementation are available under

`validation/`
`validation/results/`

The observational supernova likelihoods used in the illustrative analysis are provided through the official Cobaya likelihood modules. Their corresponding data files are obtained and managed according to the installation and data-access procedures defined by Cobaya. Each CosmoFit run produces the configuration files, metadata, execution logs, MCMC chains, status information, and posterior summaries required to inspect and reproduce the analysis. No proprietary data were used in this work.

ACKNOWLEDGMENTS

GC acknowledges SECIHTI support through the program *Estancias Posdoctorales por México 2023(1)*. This work was partially supported by S.N.I.I. (SECIHTI-México).

[1] R. Trotta, Bayes in the sky: Bayesian inference and model selection in cosmology, Contemporary Physics **49**, 71 (2008), arXiv:0803.4089 [astro-ph].
[2] A. Lewis and S. Bridle, Cosmological parameters from cmb and other data: A monte carlo approach, Physical Review D **66**, 103511 (2002), arXiv:astro-ph/0205436.
[3] B. Audren, J. Lesgourgues, K. Benabed, and S. Prunet, Conservative constraints on early cosmology with monte python, Journal of Cosmology and Astroparticle Physics **2013** (02), 001, arXiv:1210.7183 [astro-ph.CO].
[4] T. Brinckmann and J. Lesgourgues, Montepython 3: boosted mcmc sampler and other features, Physics of the Dark Universe **24**, 100260 (2019), arXiv:1804.07261 [astro-ph.CO].
[5] J. Torrado and A. Lewis, Cobaya: Code for bayesian analysis of hierarchical physical models, Journal of Cosmology and Astroparticle Physics **2021** (05), 057, arXiv:2005.05290 [astro-ph.IM].
[6] A. Lewis, Getdist: a python package for analysing monte carlo samples, (2019), arXiv:1910.13970 [astro-ph.IM].
[7] D. Brout, D. Scolnic, B. Popovic, A. G. Riess, A. Carr, J. Zuntz, R. Kessler, T. M. Davis, S. Hinton, D. Jones, *et al.*, The pantheon+ analysis: cosmological constraints, The Astrophysical Journal **938**, 110 (2022).
[8] A. G. Riess, W. Yuan, L. M. Macri, D. Scolnic, D. Brout, S. Casertano, D. O. Jones, Y. Murakami, G. S. Anand, L. Breuval, *et al.*, A comprehensive measurement of the local value of the hubble constant with 1 km s- 1 mpc- 1 uncertainty from the hubble space telescope and the sh0es team, The Astrophysical journal letters **934**, L7 (2022).
[9] D. Rubin, G. Aldering, M. Betoule, A. S. Fruchter, X. Huang, A. G. Kim, C. Lidman, E. V. Linder, S. Perlmutter, P. Ruiz-Lapuente, and N. Suzuki, Union through UNITY: Cosmology with 2000 SNe using a unified bayesian framework, The Astrophysical Journal **986**, 231 (2025), arXiv:2311.12098 [astro-ph.CO].
[10] L. Amendola and S. Tsujikawa, *Dark energy: theory and observations* (Cambridge University Press, 2010).
[11] A. De Felice and S. Tsujikawa, f (r) theories, Living Reviews in Relativity **13**, 1 (2010).
[12] T. Padmanabhan and D. Kothawala, Lanczos–lovelock models of gravity, Physics Reports **531**, 115 (2013).
[13] A. Arroyo, R. Cordero, G. Cruz, and E. Rojas, Lovelock type brane cosmology, Classical and Quantum Gravity **43**, 055008 (2026).
[14] G. Dvali, G. Gabadadze, and M. Porrati, 4d gravity on a brane in 5d minkowski space, Physics Letters B **485**, 208 (2000).
[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, The Journal of Chemical Physics **21**, 1087 (1953).
[16] W. K. Hastings, Monte carlo sampling methods using markov chains and their applications, Biometrika **57**, 97 (1970).
[17] A. Gelman and D. B. Rubin, Inference from iterative simulation using multiple sequences, Statistical Science **7**, 457 (1992).
[18] R. Jimenez and A. Loeb, Constraining cosmological parameters based on relative galaxy ages, The Astrophysical Journal **573**, 37 (2002), arXiv:astro-ph/0106145 [astro-ph].
[19] M. Moresco, A. Cimatti, R. Jimenez, L. Pozzetti, G. Zamorani, M. Bolzonella, J. Dunlop, F. Lamareille, M. Mignoli, H. Pearce, *et al.*, Improved constraints on the expansion rate of the universe up to $z \sim 1.1$ from the spectroscopic evolution of cosmic chronometers, Journal of Cosmology and Astroparticle Physics **2012** (08), 006, arXiv:1201.3609 [astro-ph.CO].
[20] M. Moresco, R. Jimenez, L. Verde, A. Cimatti, L. Pozzetti, C. Maraston, and D. Thomas, Constraining the time evolution of dark energy, curvature and neutrino properties with cosmic chronometers, Journal of Cosmology and Astroparticle Physics **2016** (12), 039.
[21] M. Moresco, R. Jimenez, L. Verde, A. Cimatti, and L. Pozzetti, Setting the stage for cosmic chronometers. ii. impact of stellar population synthesis models systematics and full covariance matrix, The Astrophysical Journal **898**, 82 (2020).
[22] D. Scolnic, D. Brout, A. Carr, A. G. Riess, T. M. Davis, A. Dwomoh, D. O. Jones, N. Ali, P. Charvu, R. Chen, *et al.*, The pantheon+ analysis: the full data set and light-curve release, The Astrophysical Journal **938**, 113 (2022).