IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 01LT01 https://doi.org/10.1088/2632-2153/acbb53

MACHINE LEARNING
Science and Technology

LETTER

Robust simulation-based inference in cosmology with Bayesian neural networks

Pablo Lemos[1,2,*], **Miles Cranmer**[3], **Muntazir Abidi**[4], **ChangHoon Hahn**[3], **Michael Eickenberg**[5], **Elena Massara**[6,7], **David Yallup**[8] and **Shirley Ho**[3,5,9,10]

[1] Department of Physics and Astronomy, University of Sussex, Sussex House, Falmer, Brighton BN1 9RH, United Kingdom
[2] Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom
[3] Department of Astrophysical Science, Princeton University, Peyton Hall, Princeton, NJ 08544, United States of America
[4] Département de Physique Théorique, Université de Genève, 24 quai Ernest Ansermet, 1211 Genève 4, Switzerland
[5] Flatiron Institute Center for Computational Mathematics, 162 5th Ave, 3rd floor, New York, NY 10010, United States of America
[6] Waterloo Centre for Astrophysics, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1, Canada
[7] Department of Physics and Astronomy, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[8] Kavli Institute for Cosmology, Cavendish Laboratory, Madingley Road, Cambridge, CB3 0HA, United Kingdom
[9] Center for Cosmology and Particle Physics, Department of Physics, New York University, NY, NY 10003, United States of America
[10] Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: p.lemos@sussex.ac.uk

Keywords: cosmology, machine learning, likelihood free, implicit likelihood, simulation based, inference, DELFI

OPEN ACCESS

RECEIVED
17 August 2022

REVISED
18 January 2023

ACCEPTED FOR PUBLICATION
10 February 2023

PUBLISHED
22 February 2023

Abstract

Simulation-based inference (SBI) is rapidly establishing itself as a standard machine learning technique for analyzing data in cosmological surveys. Despite continual improvements to the quality of density estimation by learned models, applications of such techniques to real data are entirely reliant on the generalization power of neural networks far outside the training distribution, which is mostly unconstrained. Due to the imperfections in scientist-created simulations, and the large computational expense of generating all possible parameter combinations, SBI methods in cosmology are vulnerable to such generalization issues. Here, we discuss the effects of both issues, and show how using a Bayesian neural network framework for training SBI can mitigate biases, and result in more reliable inference outside the training set. We introduce `cosmoSWAG`, the first application of stochastic weight averaging to cosmology, and apply it to SBI trained for inference on the cosmic microwave background.

1. Introduction

We are entering a new era for cosmology. Machine Learning applications to cosmology allow for the analysis of large datasets, and the exploration of new models and phenomena [1–6] Traditionally, the field has relied on likelihood-based methods, in which we compress our data into summary statistics, for which we can make theoretical predictions and build likelihood functions. However, with the development of practical machine learning tools for high-dimensional data over the last decade, it is now possible to perform cosmological analysis even for intractable likelihoods. Instead of a likelihood, we can use simulations of observables to perform parameter inference, and model comparison. This technique is often called likelihood-free inference, approximate Bayesian computation [7–9], implicit-likelihood inference or simulation-based inference (SBI) [10]. We will adopt the latter term in the remainder of this work. SBI allows us to perform parameter inference and model comparison, even in situations where the likelihood is intractable, such as field-level inference [11].

Multiple SBI methods have been developed in recent years, but particularly relevant to cosmology is density estimation likelihood-free inference ((DELFI), also known as neural posterior estimation) [12–16],

which uses a density estimator to estimate the likelihood[11]. This method has multiple advantages: it uses all available simulations and estimates the full-dimensional posterior distributions, not just marginalised posteriors. However, practical applications of DELFI to cosmology often encounter two issues [17]: The first one is the limited number of available simulations. To circumvent the curse of dimensionality, the original DELFI method proposes using a step of massive data compression, which reduces the dimensionality of the data to the dimensionality of the parameter space. This facilitates the task of density estimation. Proposed data compression methods include massively optimised parameter estimation and data compression (MOPED) [18] and information maximizing neural networks (IMNNs) [19, 20]. These methods, however, rely on either a covariance matrix for the data errors or the ability to generate a large number of simulations to estimate a covariance. When none of these conditions are met, other data compression methods have to be used, which will generally lead to a lossy compression—meaning it does not retain all information about the parameters—and a loss of accuracy. This will be the case if we intend to apply DELFI to an existing suite of simulations, such as the QUIJOTE [21] and CAMELS [22] simulations.

The second issue of practical applications of DELFI, and SBI in general, is difficulty simulating realistic observations. It does not matter how good the performance of our SBI algorithm is if we have failed to generate simulations that model all systematic effects and observational errors. In interesting examples, it is impossible to model everything. Therefore, we try to get as close as possible. But we need to deal with the fact that our simulations are likely to be imperfect. Furthermore, most SBI methods, including DELFI, have no way of informing us whether the observations we are trying to analyse are different from our simulations. How do we then interpret a surprising result coming from an SBI analysis? As a true scientific discovery, or a failure to generate realistic enough simulations? In this work, we present a way to mitigate this effect: We propose using Bayesian neural networks (BNNs) in our SBI analysis. BNNs are well known to provide better generalization to observations that have not been used during training [23–26]. We also expected BNNs to account for some of the epistemic uncertainty introduced in the neural network training. Therefore, in the presence of unknown systematics, BNNs will give us larger errors, instead of biased posteriors. With this goal in mind, in this work, we introduce `cosmoSWAG`, the first application of stochastic weight averaging (SWA) [27, 28] to cosmology[12]. SWAG (SWA Gaussian) was previously used in astronomy [29] to accurately predict planetary instability of five-planet systems, despite only training on three-planet systems. While other methods exist to perform approximate marginalisation over neural network parameters, such as MC dropout [25, 30] or Variational Inference [31, 32], SWAG has been show to perform better over a variety of tasks [27].

The goal of this paper is to study how we can maximize the accuracy of a DELFI analysis, for a fixed suite of simulations, and in the case in which running more simulations is not possible. This is the situation we find ourselves in if we want to perform a DELFI analysis with existing data, in situations where simulations are costly.

2. Simulator

To set up a realistic cosmological analysis, that we can apply DELFI to, we choose to use simulations of the cosmic microwave background (CMB) power spectrum. The main reason to do this is that this is a problem where it is easy, and computationally cheap, to generate a suite of simulations; and that this is a problem where we can actually write down a likelihood and perform a likelihood-based analysis. Therefore, by using this simulator, we can compare our obtained posterior distributions to the ones we should obtain. Cole *et al* [33] already used the CMB to test the performance of an SBI model.

Our approach is therefore the following:

(a) We use `CAMB` [34–36] to generate a suite of 100.000 CMB power spectra[13]. We use $\ell_{\rm max} = 2500$, and use only, the power spectrum of temperature anisotropies.
(b) We then use *Planck* 2018 TT [37] native likelihood used in the code `cobaya` [38], to convert this power spectrum into a binned power spectrum, at 215 multipole bins.
(c) We use the *Planck* TT data and covariance matrix in the same likelihood, as our observed data and error model, respectively. One advantage of this likelihood is that it uses only multipoles $\ell > 30$, and approximates the error model at those scales by a normal distribution.

[11] Different variations of the method estimate the likelihood, and then multiply by the prior, or estimate the posterior directly.
[12] The code is available at https://github.com/Pablo-Lemos/cosmoSWAG.
[13] A previous version of this paper used a suite of 10.000. The effect of varying the number of simulations is explored in appendix E.

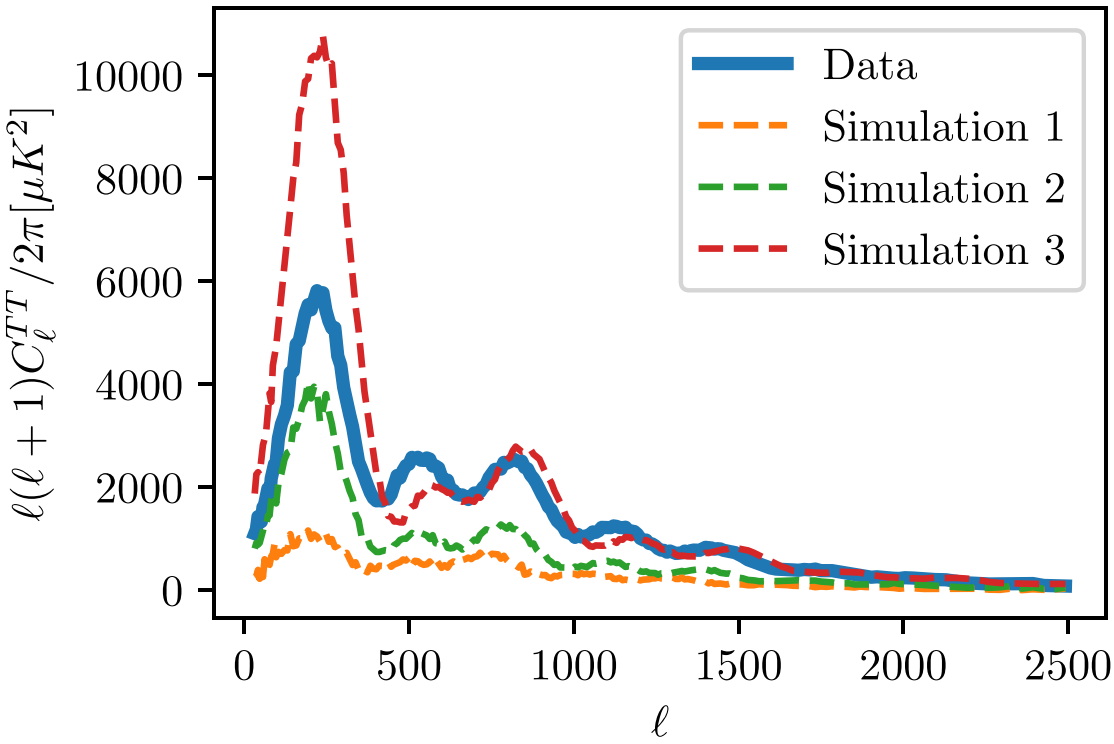

Figure 1. Three example simulations, chosen from three random parameter draws from the prior table A1, and true data as observed by *Planck* in blue.

We show some example simulations, as well as the true observation in figure 1. Our simulations are drawn from a uniform prior, shown in appendix A.

3. Analysis

3.1. DELFI with massive data compression

To perform parameter inference using our CMB simulations, we start by doing the DELFI analysis we would under ideal circumstances, as described in [40, 41]. In this analysis, we start with a step of massive data compression that reduces the dimensionality of the data to the dimensionality of the parameter space. We can ensure this compression is lossless when data is abundant, e.g. in a situation when we can quickly generate large numbers of new simulations, through algorithms such as MOPED and IMNN. However, we want to test how well we can perform using only a fixed set of simulations, to simulate a realistic scenario. In that case, the neural network compression will be inaccurate and hence it may lose information about the parameters. We use a regression network, also known as a neural compressor; which is just a neural network that tries to predict parameter values from data. This compresses the data into the dimensionality of the parameter space, but that compression can be imperfect. We use a neural network with 6 hidden layers, each containing 128 neurons. We use rectified linear unit [42] activation functions, and L_2 regularization of the weights, with a regularization factor 0.1. Our loss is the mean squared error. During training, we add noise to each input according to the noise model described in section 2.

We then use the predictions of this neural network as the compressed data in our DELFI analysis. We use a masked autoregressive flow (MAF) [43] as a density estimator, containing a stack of 5 masked autoencoders [44], each containing two hidden layers with 30 neurons each. We do this using the `pyDELFI` package available at https://github.com/justinalsing/pydelfi.

3.2. DELFI without explicit data compression

Given that we expect the compression to be lossy, it is natural to ask ourselves the question: What about using no explicit data compression, and performing density estimation directly on the data? After all, density estimation techniques such as normalizing flows have been applied successfully to high dimensional data, such as images [45]. Therefore, we try to perform our DELFI analysis directly from the data.

We use mixture density networks (MDN) [46] for density estimation, instead of MAFs. The reason for this is that we want to compare the results of this section, to the results of the following section using `cosmoSWAG`, and at present time `cosmoSWAG` does not support MAFs.

Therefore, we use a neural network with the same structure as the one used in section 3.1, but with a different number of outputs, as described in appendix B.

3.3. DELFI without explicit data compression and with weight marginalisation

Next, we repeat the analysis of DELFI with an MDN, but applying SWA to the neural network. The basic idea is, starting from a pre-trained set of neural network weights, to perform stochastic gradient descent with a constant large learning rate. We average the weights as the model is trained, and use the evolving weight values to approximate a mean and covariance matrix for the neural network weights. While this assumes that the posteriors on the weights are Gaussian, the method provides an estimate of the weight uncertainty, and therefore the uncertainty of the predictions. More moments could be computed to characterize the posterior in more detail and assess the validity of the Gaussian assumption. Furthermore, when estimating the covariance matrix of the neural network weights, we use a tunable 'scale' hyperparameter. The reason for this hyperparameter is that the covariance matrix estimated by SWA will depend on the learning rate. While for an optimal learning rate [47], the scale parameter should be set to 0.5, in practice it is possible to use the scale hyperparameter to rescale the covariance, and therefore the posterior width. In this work, we use the validation set to find the optimal value of the scale hyperparameter, as explained in appendix D. A more detailed description of the method can be found in [27].

BNNs provide two important advantages over traditional neural networks. Robustness to overfitting [48] and generalization properties [28]. Robustness to overfitting means that we are less likely to get biased posteriors. More importantly, the generalization properties mean that our SBI algorithm should perform better when our observed data does not perfectly match the simulations, either because of systematics or observational effects in the data that are not present in the simulations or because the theoretical model we are using to simulate is incorrect. We test this using our simulator in the following section.

3.4. DELFI with massive data compression and with weight marginalisation

Finally, we can repeat the DELFI analysis with explicit massive compression through a regression network, adding marginalisation to the neural network weights. We first compress the data with a regression network, and the apply a MDN on the compressed data. We use weight marginalization on both the compression and the MDN.

4. Results

4.1. Comparison with likelihood-based analysis

The results of applying all four versions of our DELFI analysis are shown in figure 2. We first focus on the left panel, using no explicit compression. We see that the DELFI posteriors do correctly capture the degeneracies of the likelihood, as the ellipses are 'tilted' in the same way as the real ones.

The size of both DELFI posteriors is larger than the likelihood-based one. This is caused by the fact that we are using a limited number of simulations, and due to the added epistemic uncertainty of having to estimate a likelihood from simulations. When we include weight marginalisation with SWA, the size of the contours increases and improves the agreement with the expected result. These results are further confirmed by repeating the analysis on several validation simulations, as shown in appendix C.1: We get slightly under confident results with DELFI, even before weight marginalisation, meaning we can trust the posteriors.

When we instead use data compression with a neural compressor, we see that we obtain slightly smaller contours. Weight marginalisation in this case leads to a small increase in the contours. In both cases, we notice that we no longer fully recover the degeneracy directions, as we did when we did not use a massive data compression step. This is likely due to some loss of information in the regression network. Our validation test (shown in appendix C.1) shows good results for this case, even before weight marginalization.

4.2. Generalization

In this section, we aim to test how our system behaves in the presence of unknown systematics or observational effects in the data, that are not present in the simulations. This issue will, to an extent, always affect SBI analysis when applied to real observations. To test it, we repeat the analysis, changing the observed *Planck* data vector for a synthetic observation. The new data vector is obtained by running `CAMB` at a fiducial cosmology, adding noise from the noise model described in section 2, and then adding extra Gaussian noise

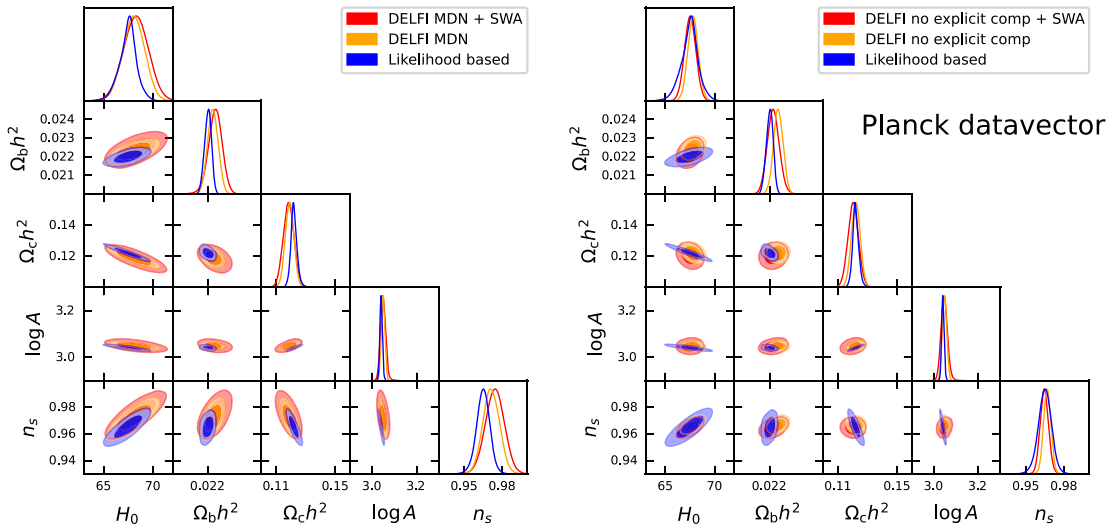

Figure 2. One and two-dimensional marginalised posterior distributions for the different variations of DELFI presented in section 3 using *Planck* data. The left plot shows the results for DELFI without explicit compression, using an MDN, with weight-marginalised results in red, and non weight-marginalised in orange. In blue, we show the likelihood-based analysis. The right plot shows the same for DELFI with compression through a regression network. The contours represent the 68% and 95% confidence levels. This plot was generated using `GetDist` [39].

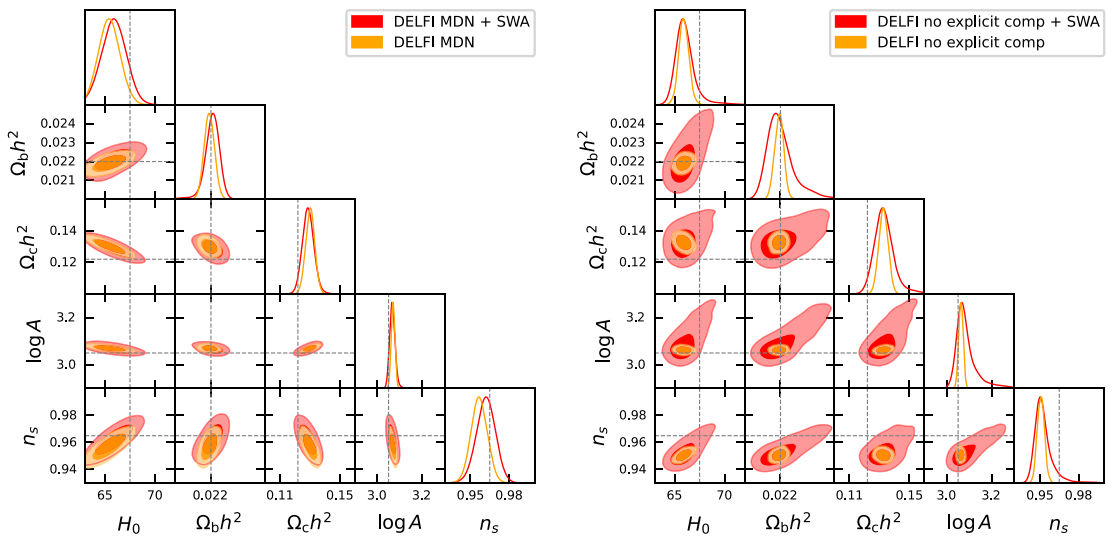

Figure 3. The same as figure 2 using a synthetic data vector with added noise, as described in section 4.2.

at small scales $\ell > 1000$. We choose this multipole range because these are the scales at which Silk damping [49] is the dominant effect. Therefore this artificial systematic could be interpreted as some unknown physics associated with Silk damping. Given that this extra noise has been added to any of the training simulations, our observed data is different from any of the simulations our DELFI algorithm has been trained on.

The results of the analysis are shown in figure 3. Using no explicit compression, we get consistent results, but we can see that when we do not marginalise over the weights with SWA, we get biased posteriors in some parameters. In this case, because we know the true parameters, we can calculate the excess probability (EP) of the true parameters, as described in appendix C. In this case, we get $EP = 0.05$ without weight marginalisation, and $EP = 0.18$ when marginalising over weights, showing that weight marginalisation does improve our results. In appendix C.2, we repeat this for all simulations in the validation set and find that indeed the non weight-marginalised case is overconfident, whereas with weight marginalisation we can get good constraints by adjusting the scale hyperparameter.

When we use a regression network for compression, the results without weight marginalisation show very clear and dramatic biases, with an EP of $EP \sim 2 \times 10^{-3}$. This shows that neural compression can lead to dangerous biases when the observed data is different from the simulations. This is very much in line with the fact that simple neural networks generally do not handle covariate shift very well, since they may include computations that involve combining irrelevant variables in such a way that a distribution shift can lead to drastic changes in outcomes. Weight marginalising greatly improves the reliability of these results, at the expense of increasing the error bars $EP \sim 0.13$. This is expected, given the better generalization properties of BNNs. Therefore, unless we are fully confident that our simulations contain all the observational effects that affect the data, we strongly recommend using weight marginalisation, to avoid biased results. Appendix C.2 repeats this analysis for several validation simulations and again shows biased contours when using compression if we do not marginalise over weights. Thus, we see how in both cases, the generalization properties of BNNs mean that SWA greatly increases the reliability of our SBI analysis when simulations do not perfectly match the data.

5. Conclusions

In this work, we have shown how to address some difficulties encountered in DELFI analyses. We have shown that, in the case of limited simulations, we get larger posterior distributions, and therefore lose constraining power, whether we use massive data compression or not. We also show how using DELFI without explicit compression leads to comparable posteriors. In either case, marginalisation of the neural network parameters prevents overfitting, and increases the reliability of the posteriors, at the expense of slightly less confident posteriors. We show how to do this using `cosmoSWAG`, the first application of SWA to cosmology. Finally, we show that weight marginalisation is even more important in the case of simulations that do not perfectly capture the physics of the data. In that case, DELFI without weight marginalisation can lead to strongly biased results. Therefore, in the likely scenario of imperfect simulations, we recommend adding weight marginalisation to your SBI analysis to increase the reliability of the posteriors.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/Pablo-Lemos/cosmoSWAG.

Acknowledgment

We organize the referees at the ML4Astro Machine Learning for Astrophysics Workshop at the Thirty-ninth International Conference on Machine Learning (ICML 2022), for comments and feedback in previous versions of this work. PL acknowledges support by the UK STFC Grant ST/T000473/1.

Appendix A. Prior

Table A1 shows the prior distributions for the cosmological parameters used to generate our suite of simulations. In this table, H_0 is the Hubble parameter in km s^{-1} Mpc^{-1}, Ω_b and Ω_c are the energy density of baryons and cold dark matter respectively, h is the reduced Hubble parameter ($h = H_0$ [km s^{-1} Mpc^{-1}] / 100), and A_s and n_s are the amplitude and tilt of the primordial power spectrum. This choice of parameter space is the one typically adopted by CMB analyses [37].

Note that our simulations assume a flat ΛCDM cosmology, and fix the optical depth to reionization to $\tau_{\rm re} = 0.06$, and the *Planck* calibration parameter to $A_{\rm Planck} = 1$.

Table A1. The prior distribution used to generate simulations.

PARAMETER	PRIOR
H_0	$\mathcal{U}(50,90)$
$\Omega_b h^2$	$\mathcal{U}(0.01,0.05)$
$\Omega_c h^2$	$\mathcal{U}(0.01,0.5)$
$\log(10^{10}A_s)$	$\mathcal{U}(1.5,3.5)$
n_s	$\mathcal{U}(0.8,1)$

Appendix B. MDN

In this section, we describe the MDN, used for compression-free DELFI introduced in section 3. Our MDN is simply a neural network, taking as inputs the data, and outputting $n_{\rm out}$ outputs, where:

$$n_{\rm out} = \left[n_\theta + n_\theta \cdot \frac{(n_\theta + 1)}{2} + 1\right] + n_{\rm comp}, \tag{B.1}$$

with n_θ the number of parameters in the parameter space (in this case 5), and $n_{\rm comp}$ is the number of components in our MDN (which we set to 3). In this equation, the first term inside square brackets represents the means of the Gaussian distributions μ, the second term are the non-zero elements of the lower triangular matrix obtained from a Cholesky decomposition of the covariance matrix Σ, and the last one is the weight of that component of the MDN α. Therefore, this neural network directly gives us an estimate of the posterior distribution as:

$$P(\theta|D) = \sum_{i=1}^{n_{\rm comp}} \alpha_i(D) \cdot N(\theta|\mu_i(D), \Sigma_i(D)), \tag{B.2}$$

where θ and D and the parameters and data respectively.

Appendix C. Validation

When we know the true parameter values, as is the case in the analysis using a synthetic data vector of section 4.2, we can calculate the EP of the true parameter values. We do this by estimating the probability of a large number of samples from our posterior and calculating the percentage of those samples with a probability smaller than the probability of the true parameters. Therefore, a small EP means that the true parameters are very unlikely, and our SBI analysis is very likely to be biased.

To check if our SBI analysis is biased, we need to repeat this EP calculation for numerous validation simulations, and check if the distribution of EP is uniform [50, 51]. Equivalently, we can calculate the coverage probability, as the cumulative distribution function of the expected probabilities, and then compare it with the expected coverage probability.

C.1. Validation of the default analysis

We first apply this validation test to the analysis of section 4.1. The results are shown in figure C1. As discussed in the main text, the no compression case gets good results before weight marginalisation, and in fact, marginalisation leads to very under confident posteriors even when we use a small scale hyperparameter. This is because the weight marginalisation case uses the average of the weights over the SWA training. On the other hand, the compression case gets good posteriors, even when before we use marginalisation.

C.2. Validation of the generalization analysis

We next apply this validation test to the analysis of section 4.2. For that, we add the extra noise at $\ell > 1000$ for all the simulations in the validation set. The results are shown in figure C2. In this case, adding weight marginalisation allows us to get posteriors of the correct size, with and without data compression.

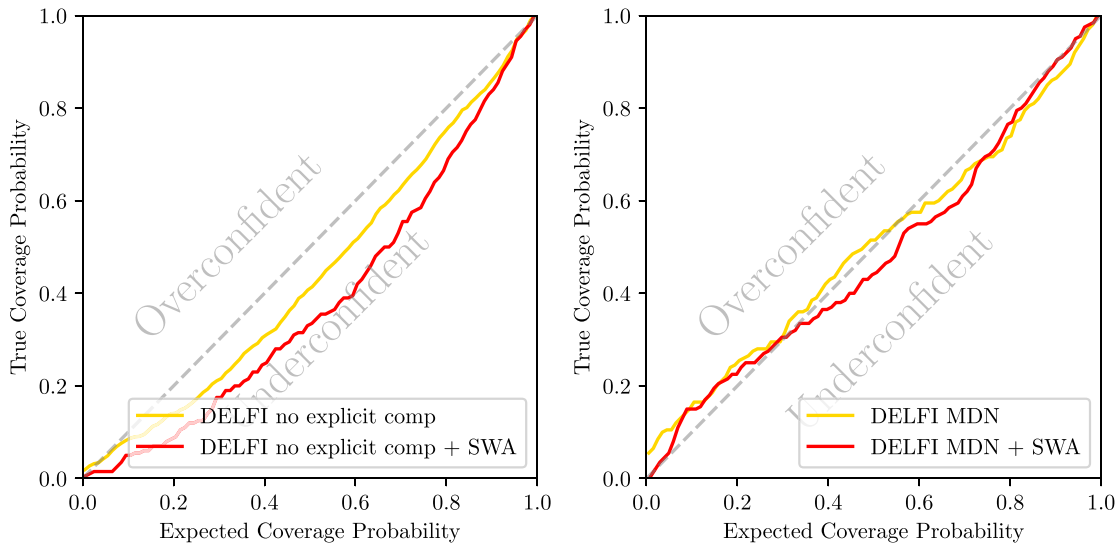

Figure C1. Validation of the default analysis shown in figure 2.

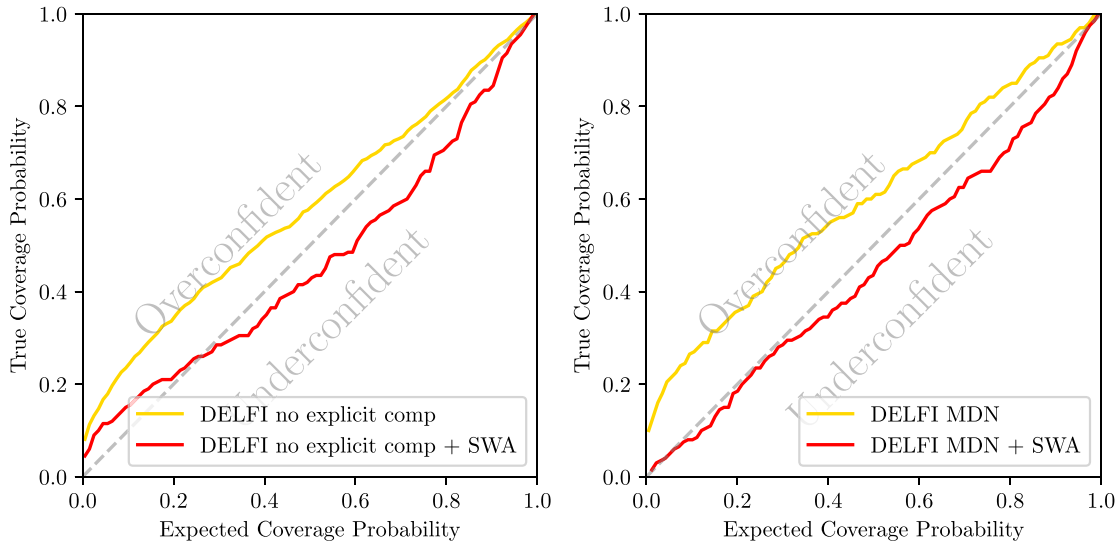

Figure C2. Validation of the generalization analysis shown in figure 3.

Appendix D. Tuning the scale hyperparameter

As described in the main text, the SWA algorithm allows us to rescale the covariance matrix by a scale hyperparameter, to correct for the fact that the covariance matrix can depend on the learning rate used [27]. In this work, we adjust the scale hyperparameter using the validation test described in appendix C. More specifically, we adjust the scale so the line in our coverage probability plots gets as close as possible to the diagonal, erring on the side of under confident posteriors, to avoid biased results. This is illustrated by figure D1, which shows this calibration performed for the DELFI with no compression analysis applied to noisy data vectors of section 4.2. In the figure, we see that a scale of 0.1 leads to overconfident posteriors, and even a scale of 0.5 is too overconfident. When we raise the scale to 1, we find that the line is predominantly under the diagonal, therefore we set the hyperparameter to that value. The advantage of tuning this hyperparameter is that it does not require retraining the network, and therefore different values can be tested fast.

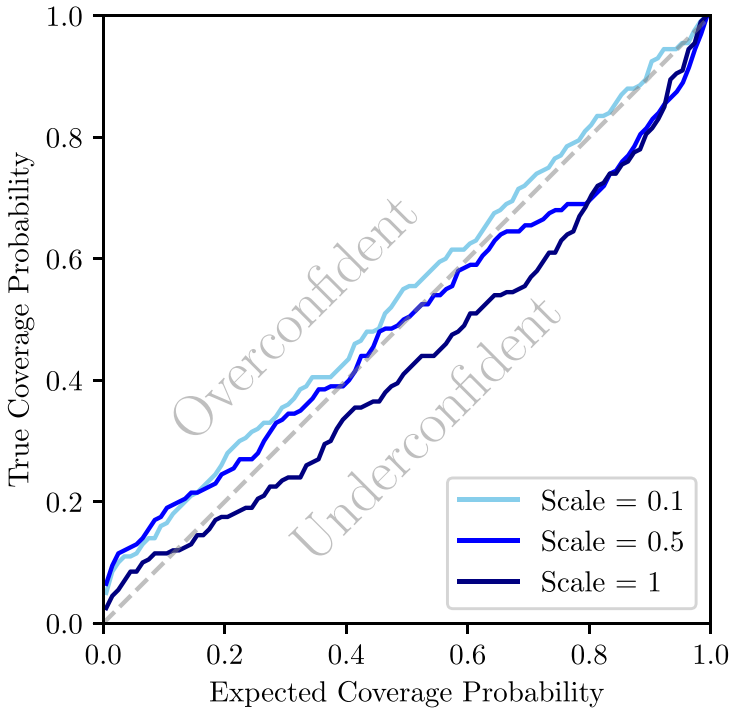

Figure D1. Illustration of how the SWA scale hyperparameter can be calibrated using the validation test described in appendix C. A larger scale factor leads to more under confident posteriors, therefore it can be modified to get as close as possible to the diagonal.

In this work, we used approximate criteria consisting of looking at the coverage probability plots. In future work, we will explore more exact tests and algorithmic tuning of the scale hyperparameter.

Appendix E. Dependence on the number of simulations

In this section, we explore the dependence of our results on the number of available simulations on our posterior inference. The number of simulations is a key parameter in the analysis, as it affects the accuracy of the posterior inference. Simulations are computationally expensive to run in a lot of contexts, and therefore it is important to understand how the number of simulations affects the results.

We repeat the analysis of section 3.2 for different numbers of simulations, and show the results in figure E1. We see that the size of the inferred posterior gets smaller as the number of simulations increases. By the team we reach 50.000 simulations, the posterior size converges, and does not significantly change as more simulations are added. Therefore, we can conclude that 100.000 is a large enough number for our analysis.

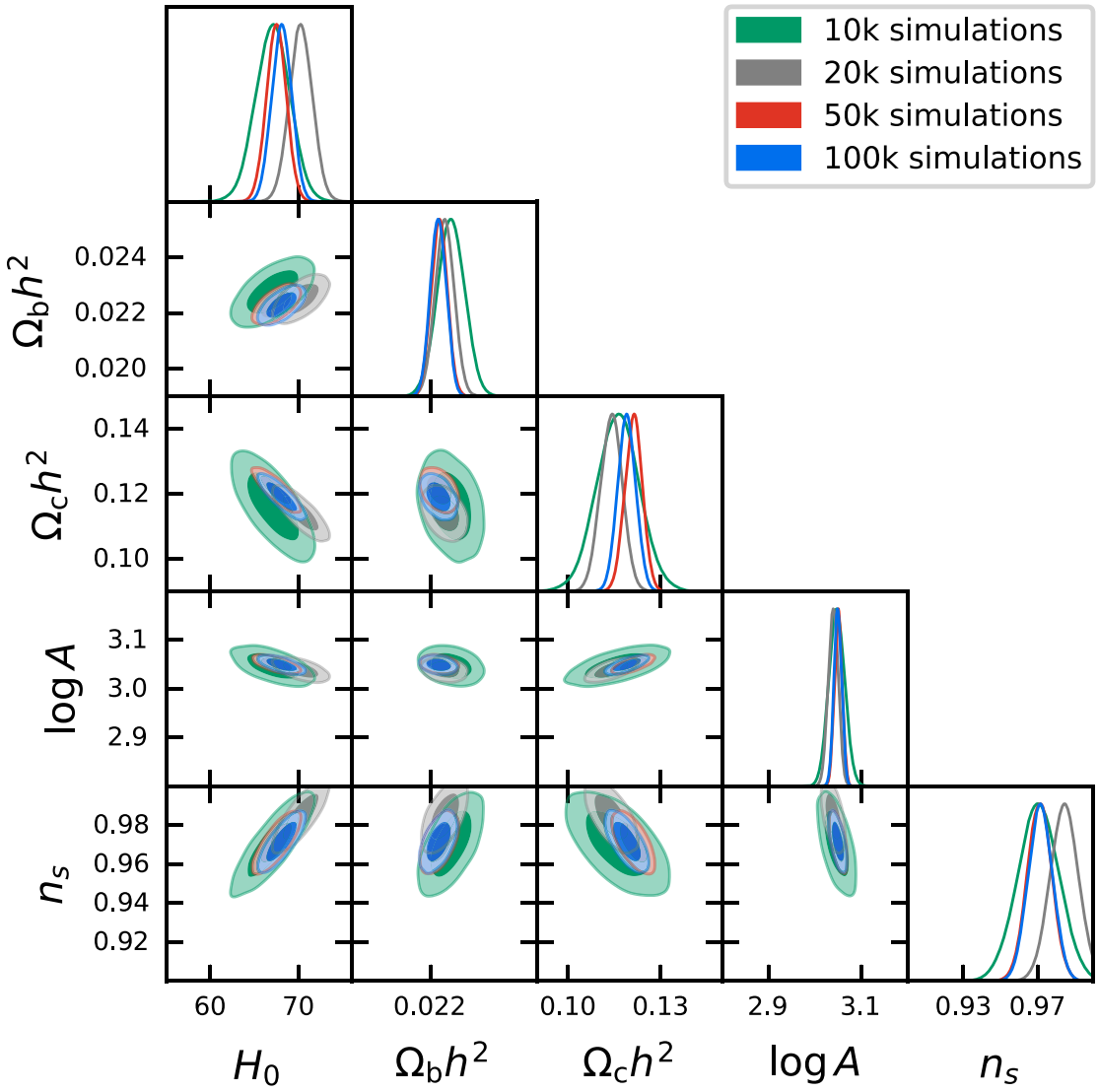

Figure E1. Dependence on the number of simulations. The different contours show the results from the DELFI + MDN analysis when we vary the number of available simulations.

ORCID iD

Pablo Lemos https://orcid.org/0000-0002-4728-8473

References

[1] Kangal E, Salti M and Aydogdu O 2019 *Phys. Dark Universe* **26** 100369
[2] Ntampaka M *et al* 2019 arXiv:1902.10159
[3] Escamilla-Rivera C, Quintero M A C and Capozziello S 2020 *J. Cosmol. Astropart. Phys.* **2020** 008
[4] Tilaver H, Salti M, Aydogdu O and Kangal E E 2021 *Comput. Phys. Commun.* **261** 107809
[5] Salti M, Kangal E E and Aydogdu O 2021 *Astron. Comput.* **37** 100504
[6] Dvorkin C *et al* 2022 arXiv:2203.08056
[7] Csilléry K, Blum M G, Gaggiotti O E and François O 2010 *Trends Ecol. Evol.* **25** 410–18
[8] Beaumont M A 2010 *Annu. Rev. Ecol. Evol. Syst.* **41** 379–406
[9] Sunnåker M, Busetto A G, Numminen E, Corander J, Foll M and Dessimoz C 2013 *PLoS Comput. Biol.* **9** e1002803
[10] Thomas O, Dutta R, Corander J, Kaski S and Gutmann M U 2016 arXiv:1611.10242
[11] Leclercq F and Heavens A 2021 *Mon. Not. R. Astron. Soc.* **506** L85–L90
[12] Bonassi F V, You L and West M 2011 *Stat. Appl. Genet. Mol. Biol.* **10** 1
[13] Fan Y, Nott D J and Sisson S A 2013 *Stat* **2** 34–48
[14] Papamakarios G and Murray I 2016 *Advances in Neural Information Processing Systems* vol 29
[15] Lueckmann J M, Goncalves P J, Bassetto G, Öcal K, Nonnenmacher M and Macke J H 2017 *Advances in Neural Information Processing Systems* vol 30
[16] Lemos P, Jeffrey N, Whiteway L, Lahav O, Noam Libeskind I and Hoffman Y 2021 *Phys. Rev.* D **103** 023009
[17] Cranmer K, Brehmer J and Louppe G 2020 *Proc. Natl Acad. Sci.* 117 30055–62

[18] Heavens A F, Sellentin E, de Mijolla D and Vianello A 2017 *Mon. Not. R. Astron. Soc.* **472** 4244–50
[19] Charnock T, Lavaux G and Wandelt B D 2018 *Phys. Rev.* D **97** 083004
[20] Makinen T L, Charnock T, Alsing J and Wandelt B D 2021 *J. Cosmol. Astropart. Phys.* **2021** 049
[21] Villaescusa-Navarro F *et al* 2020 *Astrophys. J. Suppl. Ser.* **250** 2
[22] Villaescusa-Navarro F *et al* 2021 *Astrophys. J.* **915** 71
[23] Kononenko I 1989 *Biol. Cybern.* **61** 361–70
[24] MacKay D J 1995 *Nucl. Instrum. Methods Phys. Res.* A **354** 73–80
[25] Gal Y and Ghahramani Z 2016 Dropout as a bayesian approximation: representing model uncertainty in deep learning *International Conference on Machine Learning* (PMLR) pp 1050–9
[26] Yallup D, Handley W, Hobson M, Lasenby A and Lemos P 2022 arXiv:2205.11151
[27] Maddox W J, Izmailov P, Garipov T, Vetrov D P and Wilson A G 2019 *Advances in Neural Information Processing Systems* vol 32
[28] Wilson A G and Izmailov P 2020 *Advances in Neural Information Processing Systems* vol 33 pp 4697–708
[29] Cranmer M, Tamayo D, Rein H, Battaglia P, Hadden S, Armitage P J, Ho S and Spergel D N 2021 *Proc. Natl Acad. Sci.* **118** 1091–6490
[30] Gal Y, Hron J and Kendall A 2017 *Advances in Neural Information Processing Systems* vol 3
[31] Graves A 2011 *Advances in Neural Information Processing Systems* vol 24
[32] Kingma D P and Welling M 2013 arXiv:1312.6114
[33] Cole A, Miller B K, Witte S J, Cai M X, Grootes M W, Nattino F and Weniger C 2021 arXiv:2111.08030
[34] Lewis A, Challinor A and Lasenby A 2000 *Astrophys. J.* **538** 473
[35] Lewis A and Bridle S 2002 *Phys. Rev.* D **66** 103511
[36] Howlett C, Lewis A, Hall A and Challinor A 2012 *J. Cosmol. Astropart. Phys.* **2012** 027
[37] Aghanim N *et al* 2020 *Astron. Astrophys.* **641** A6
[38] Torrado J and Lewis A 2021 *J. Cosmol. Astropart. Phys.* **2021** 057
[39] Lewis A 2019 arXiv:1910.13970
[40] Alsing J, Wandelt B and Feeney S 2018 *Mon. Not. R. Astron. Soc.* **477** 2874–85
[41] Alsing J, Charnock T, Feeney S and Wandelt B 2019 *Mon. Not. R. Astron. Soc.* **488** 4440–58
[42] Agarap A F 2018 arXiv:1803.08375
[43] Papamakarios G, Pavlakou T and Murray I 2017 *Advances in Neural Information Processing Systems* vol 30
[44] Germain M, Gregor K, Murray I and Larochelle H 2015 Made: masked autoencoder for distribution estimation *Int. Conf. on Machine Learning* (PMLR) pp 881–9
[45] Helminger L, Djelouah A, Gross M and Schroers C 2020 arXiv:2008.10486
[46] Bishop C M 1994 *Mixture Density Networks* NCRG/94/004 Aston University, Birmingham (available at: https://publications.aston.ac.uk/id/eprint/373/1/NCRG_94_004.pdf)
[47] Mandt S, Hoffman M D and Blei D M 2017 arXiv:1704.04289
[48] Hernández-Lobato J M and Adams R 2015 Probabilistic backpropagation for scalable learning of bayesian neural networks *Int. Conf. on Machine Learning* (PMLR) pp 1861–9
[49] Hu W, Sugiyama N and Silk J 1997 *Nature* **386** 37–43
[50] Levasseur L P, Hezaveh Y D and Wechsler R H 2017 *Astrophys. J. Lett.* **850** L7
[51] Hermans J, Delaunoy A, Rozet F, Wehenkel A and Louppe G 2021 arXiv:2110.06581