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] **Shirley Ho** [5 3 9 10]

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. 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 (ABC, Csilléry et al., 2010; Beaumont, 2010; Sunnåker et al., 2013), implicit-likelihood inference (ILI) or simulation-based inference (SBI) (Thomas et al., 2016). 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 (Leclercq & Heavens, 2021).

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 Bonassi et al., 2011; Fan et al., 2013; Papamakarios & Murray, 2016; Lueckmann et al., 2017), which uses a density estimator to estimate the likelihood. 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 (Cranmer et al., 2020): 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 MOPED (Heavens et al., 2017) and Information Maximizing Neural Networks (IMMNs, Charnock et al., 2018; Makinen et al., 2021). 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 (Villaescusa-Navarro et al., 2020) and CAMELS (Villaescusa-Navarro et al., 2021) 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

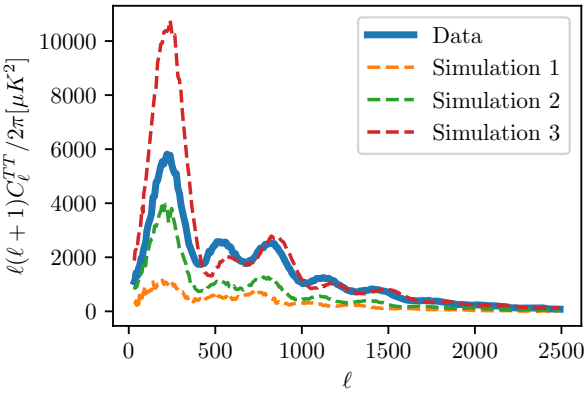

Figure 1. Three example simulations, chosen from three random parameter draws from the prior Tab. 1, and true data as observed by *Planck* in blue.

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 (Kononenko, 1989; MacKay, 1995; Gal & Ghahramani, 2016; Yallup et al., 2022). 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, Maddox et al., 2019; Wilson & Izmailov, 2020) to cosmology.[1] SWAG was previously used in astronomy (Cranmer et al., 2021) to accurately predict planetary instability of five-planet systems, despite only training on three-planet systems.

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., 2021) already used the CMB to test the performance of an SBI model.

Our approach is therefore the following:

1. We use `CAMB` (Lewis et al., 2000; Lewis & Bridle, 2002; Howlett et al., 2012) to generate a suite of 10.000 CMB power spectra. We use $\ell_{\rm max} = 2500$, and use only, the power spectrum of temperature anisotropies.

2. We then use *Planck* 2018 TT (Aghanim et al., 2020) native likelihood used in the code `cobaya` (Torrado & Lewis, 2021), to convert this power spectrum into a binned power spectrum, at 215 multipole bins.

3. 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.

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

3. Analysis

3.1. DELFI with 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 (Alsing et al., 2018; 2019). 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 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 (ReLU, Agarap, 2018) 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 Sec. 2.

[1] The code is available at `https://github.com/Pablo-Lemos/cosmoSWAG`.

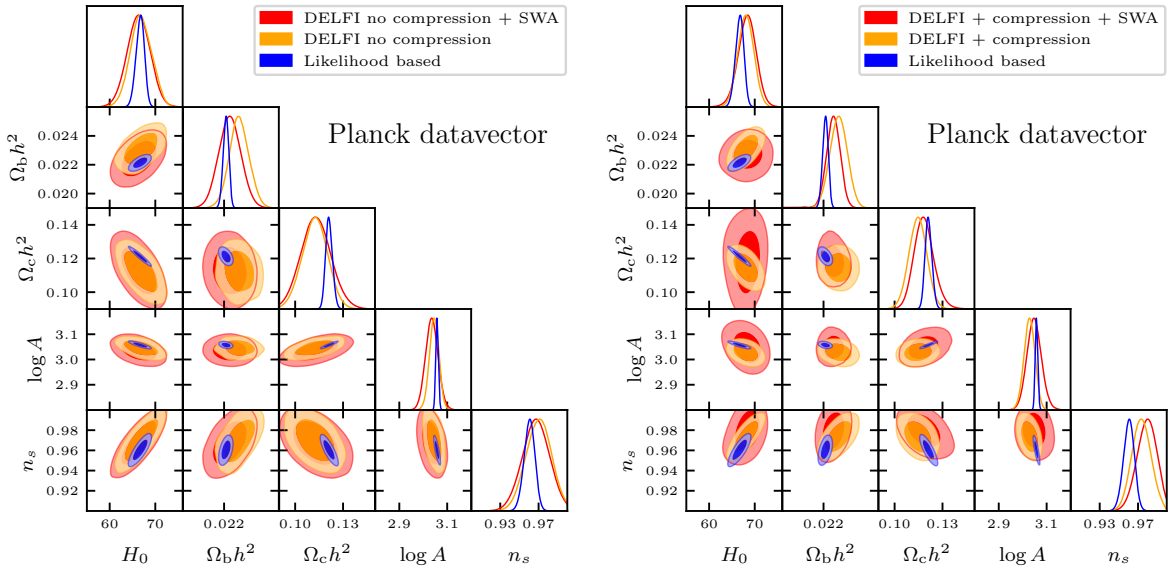

Figure 2. One and two-dimensional marginalised posterior distributions for the different variations of DELFI presented in Sec. 3 using *Planck* data. The left plot shows the results for DELFI without compression, using an MDN, with marginalised results in red, and non-marginalised in orange. In blue, we show the likelihood-based analysis. The right plot shows the same for DELFI with compression. The contours represent the 68 and 95 % confidence levels. This plot was generated using `GetDist` (Lewis, 2019)

We then use the predictions of this neural network as the compressed data in our DELFI analysis. We use a masked autoregressive flow (MAF, Papamakarios et al., 2017) as a density estimator, containing a stack of 5 masked autoencoders (Germain et al., 2015), 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 data compression

Given that we expect the compression to be lossy, it is natural to ask ourselves the question: What about using no compression at all? After all, density estimation techniques such as normalizing flows have been applied successfully to high dimensional data, such as images (Helminger et al., 2020). Therefore, we try to perform our DELFI analysis directly from the data.

We use Mixture Density Networks (MDN, Bishop, 1994) 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. We did compare the results of this section using MDNs and MAFs and found that MAF gets slightly tighter contours, but the differences are not large enough to affect our conclusions.

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

3.3. DELFI without 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 (Mandt et al., 2017), 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.

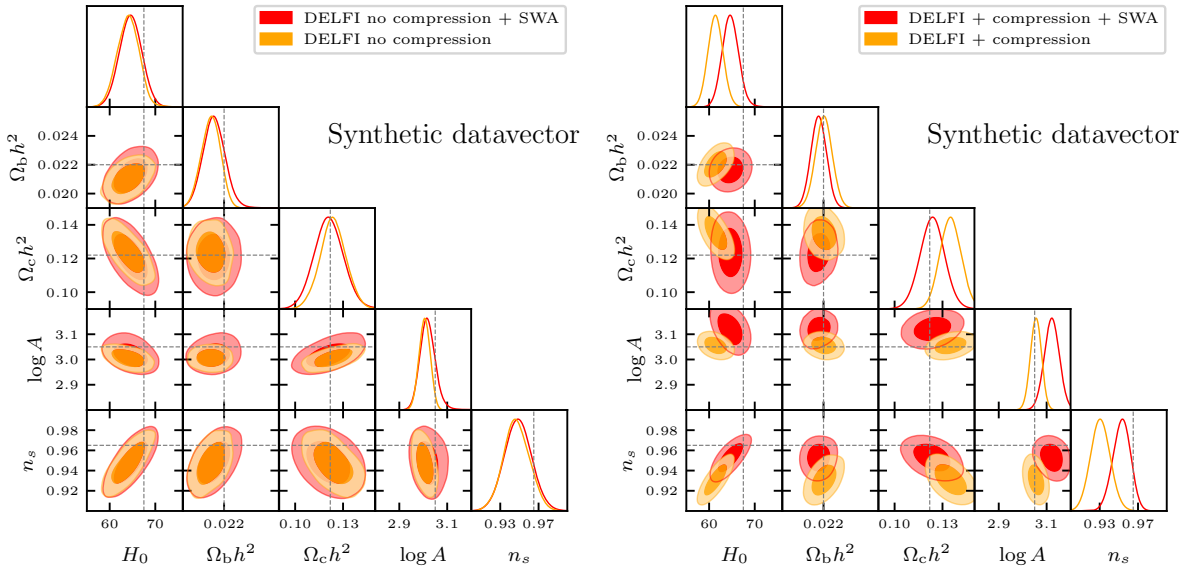

Figure 3. The same as Fig. 2 using a synthetic data vector with added noise, as described in Subsection 4.2.

A more detailed description of the method can be found in (Maddox et al., 2019).

BNNs provide two important advantages over traditional neural networks. Robustness to overfitting (Hernández-Lobato & Adams, 2015) and generalization properties (Wilson & Izmailov, 2020). 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 data compression and with weight marginalisation

Finally, we can repeat the DELFI analysis with massive compression, adding marginalisation. While we could in principle optimize over both compression and density estimation steps, data compression is the most likely to lead to overfitting, and potential biases. Therefore, we use marginalisation with SWA on the data compression only and use a MAF for density estimation. The effect of using marginalisation on both steps will be explored in future work.

4. Results

4.1. Comparison with likelihood-based analysis

The results of applying all four versions of our DELFI analysis are shown in Fig. 2. We first focus on the left panel, using no 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. When we include 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 underconfident results with DELFI, even before marginalisation, meaning we can trust the posteriors.

When we instead use data compression, we see that we obtain slightly smaller contours. Marginalisation in this case leads to a significant increase in the contours. Again, the fact that all our DELFI posteriors are larger than the likelihood-based posterior, is a consequence of our limited number of simulations. In general, when our number of simulations is limited, our SBI analysis will lose some of its potential constraining power. Our validation test (shown in Appendix C.1) shows good results for this case, even before 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 Sec. 2, and then adding extra Gaussian noise at small scales $\ell > 1000$. We choose this multipole range because these are the scales at which Silk damping (Hu et al., 1997) 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 Fig. 3. Using no compression, we get consistent results, but we can see that when we do not marginalise with SWA, we get biased posteriors in some parameters, in this case, especially in $\log A$. 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 $EP = 0.19$ without marginalisation, and $EP = 0.24$ when marginalising, showing that 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-marginalised case is overconfident, whereas with marginalisation we can get good constraints by adjusting the scale hyperparameter.

When we use compression, the results without marginalisation show very clear and dramatic biases, with an excess probability of $EP \sim 3 \cdot 10^{-4}$. 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. Marginalising greatly improves the reliability of these results, at the expense of increasing the error bars $EP \sim 0.088$. 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 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. Therefore, 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 data compression or not. We also show how using DELFI without 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 Stochastic Weight Averaging to cosmology. Finally, we show that 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 marginalisation can lead to strongly biased results. Therefore, in the likely scenario of imperfect simulations, we strongly recommend adding marginalisation to your SBI analysis.

References

Agarap, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A., Barreiro, R., Bartolo, N., Basak, S., et al. Planck 2018 results-vi. cosmological parameters. *Astronomy & Astrophysics*, 641: A6, 2020.

Alsing, J., Wandelt, B., and Feeney, S. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly Notices of the Royal Astronomical Society*, 477(3):2874–2885, 2018.

Alsing, J., Charnock, T., Feeney, S., and Wandelt, B. Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 488(3):4440–4458, 2019.

Beaumont, M. A. Approximate bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406, 2010.

Bishop, C. M. Mixture density networks. 1994.

Bonassi, F. V., You, L., and West, M. Bayesian learning from marginal data in bionetwork models. *Statistical applications in genetics and molecular biology*, 10(1), 2011.

Charnock, T., Lavaux, G., and Wandelt, B. D. Automatic physical inference with information maximizing neural networks. *Physical Review D*, 97(8):083004, 2018.

Cole, A., Miller, B. K., Witte, S. J., Cai, M. X., Grootes, M. W., Nattino, F., and Weniger, C. Fast and credible likelihood-free cosmology with truncated marginal neural ratio estimation. *arXiv preprint arXiv:2111.08030*, 2021.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Cranmer, M., Tamayo, D., Rein, H., Battaglia, P., Hadden, S., Armitage, P. J., Ho, S., and Spergel, D. N. A Bayesian neural network predicts the dissolution of compact planetary systems. *Proceedings of the National Academy of Sciences*, 118(40), October 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2026053118.

Csilléry, K., Blum, M. G., Gaggiotti, O. E., and François, O. Approximate bayesian computation (abc) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

Fan, Y., Nott, D. J., and Sisson, S. A. Approximate bayesian computation via regression density estimation. *Stat*, 2(1): 34–48, 2013.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pp. 881–889. PMLR, 2015.

Heavens, A. F., Sellentin, E., de Mijolla, D., and Vianello, A. Massive data compression for parameter-dependent covariance matrices. *Monthly Notices of the Royal Astronomical Society*, 472(4):4244–4250, 2017.

Helminger, L., Djelouah, A., Gross, M., and Schroers, C. Lossy image compression with normalizing flows. *arXiv preprint arXiv:2008.10486*, 2020.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. Averting a crisis in simulation-based inference. *arXiv preprint arXiv:2110.06581*, 2021.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pp. 1861–1869. PMLR, 2015.

Howlett, C., Lewis, A., Hall, A., and Challinor, A. Cmb power spectrum parameter degeneracies in the era of precision cosmology. *Journal of Cosmology and Astroparticle Physics*, 2012(04):027, 2012.

Hu, W., Sugiyama, N., and Silk, J. The physics of microwave background anisotropies. *Nature*, 386(6620): 37–43, 1997.

Kononenko, I. Bayesian neural networks. *Biological Cybernetics*, 61(5):361–370, 1989.

Leclercq, F. and Heavens, A. On the accuracy and precision of correlation functions and field-level inference in cosmology. *Monthly Notices of the Royal Astronomical Society: Letters*, 506(1):L85–L90, 2021.

Lemos, P., Haan, C., Cranmer, M., and Ho, S. Validation of delfi analyses (in preparation). *In preparation*, 2022.

Levasseur, L. P., Hezaveh, Y. D., and Wechsler, R. H. Uncertainties in parameters estimated with neural networks: Application to strong gravitational lensing. *The Astrophysical Journal Letters*, 850(1):L7, 2017.

Lewis, A. GetDist: a Python package for analysing Monte Carlo samples. 2019. URL https://getdist.readthedocs.io.

Lewis, A. and Bridle, S. Cosmological parameters from cmb and other data: A monte carlo approach. *Physical Review D*, 66(10):103511, 2002.

Lewis, A., Challinor, A., and Lasenby, A. Efficient computation of cosmic microwave background anisotropies in closed friedmann-robertson-walker models. *The Astrophysical Journal*, 538(2):473, 2000.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017.

MacKay, D. J. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1):73–80, 1995.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Makinen, T. L., Charnock, T., Alsing, J., and Wandelt, B. D. Lossless, scalable implicit likelihood inference for cosmological fields. *Journal of Cosmology and Astroparticle Physics*, 2021(11):049, 2021.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Papamakarios, G. and Murray, I. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., and Dessimoz, C. Approximate bayesian computation. *PLoS computational biology*, 9(1):e1002803, 2013.

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*, 2016.

Torrado, J. and Lewis, A. bcobaya: code for bayesian analysis of hierarchical physical models. *Journal of Cosmology and Astroparticle Physics*, 2021(05):057, 2021.

Villaescusa-Navarro, F., Hahn, C., Massara, E., Banerjee, A., Delgado, A. M., Ramanah, D. K., Charnock, T., Giusarma, E., Li, Y., Allys, E., et al. The quijote simulations. *The Astrophysical Journal Supplement Series*, 250 (1):2, 2020.

Villaescusa-Navarro, F., Anglés-Alcázar, D., Genel, S., Spergel, D. N., Somerville, R. S., Dave, R., Pillepich, A., Hernquist, L., Nelson, D., Torrey, P., et al. The camels project: Cosmology and astrophysics with machine-learning simulations. *The Astrophysical Journal*, 915 (1):71, 2021.

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Yallup, D., Handley, W., Hobson, M., Lasenby, A., and Lemos, P. Split personalities in Bayesian Neural Networks: the case for full marginalisation. *arXiv e-prints*, art. arXiv:2205.11151, May 2022.

Table 1. 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	$\mathcal{U}(0.8, 1)$

A. Prior

Table Tab. 1 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 $\mathrm{km\ s^{-1}\ Mpc^{-1}}$, Ω_b and Ω_c are the energy density of baryons and cold dark matter respectively, h is the reduced Hubble parameter ($h = H_0[\mathrm{km\ s^{-1}\ Mpc^{-1}}]/100$), 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 (Aghanim et al., 2020).

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

B. Mixture Density Network

In this section, we describe the Mixture Density Network (MDN), used for compression-free DELFI introduced in Sec. 3. Our MDN is simply a neural network, taking as inputs the data, and outputting n_{out} outputs, where

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

with n_θ the number of parameters in the parameter space (in this case 5), and n_{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 Mixture Density Network α. Therefore, this neural network directly gives us an estimate of the posterior distribution as:

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

where θ and D and the parameters and data respectively.

C. Validation

When we know the true parameter values, as is the case in the analysis using a synthetic data vector of Subsection 4.2, we can calculate the excess probability 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 excess probability 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 excess probability calculation for numerous validation simulations, and check if the distribution of excess probabilities is uniform (Levasseur et al., 2017; Hermans et al., 2021; Lemos et al., 2022). 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.

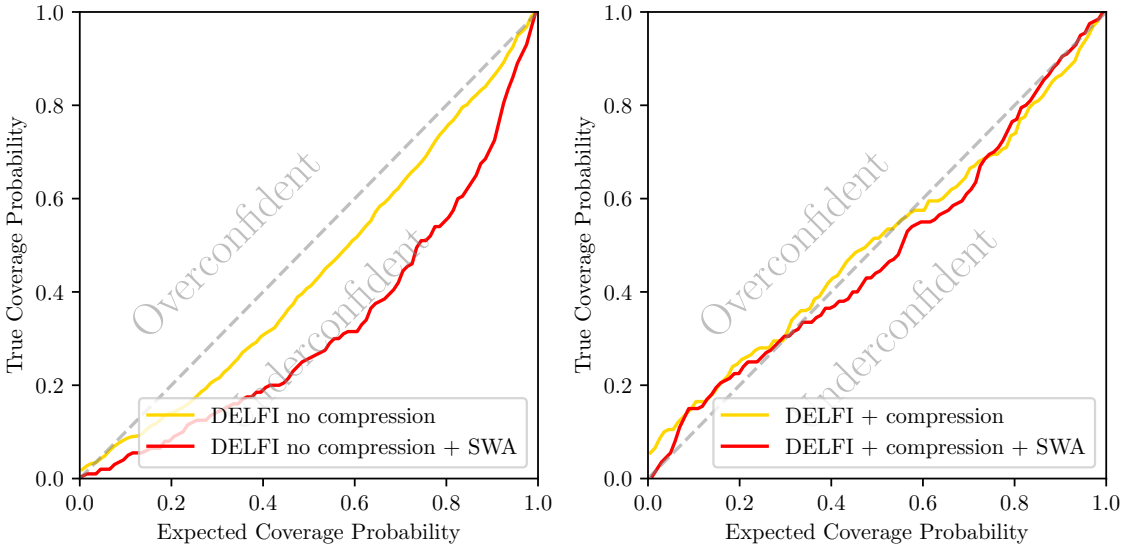

Figure 4. Validation of the default analysis shown in Fig. 2.

C.1. Validation of the default analysis

We first apply this validation test to the analysis of Subsection 4.1. The results are shown in Fig. 4. As discussed in the main text, the no compression case gets good results before marginalisation, and in fact, marginalisation leads to very underconfident posteriors even when we use a small scale hyperparameter. This is because the 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.

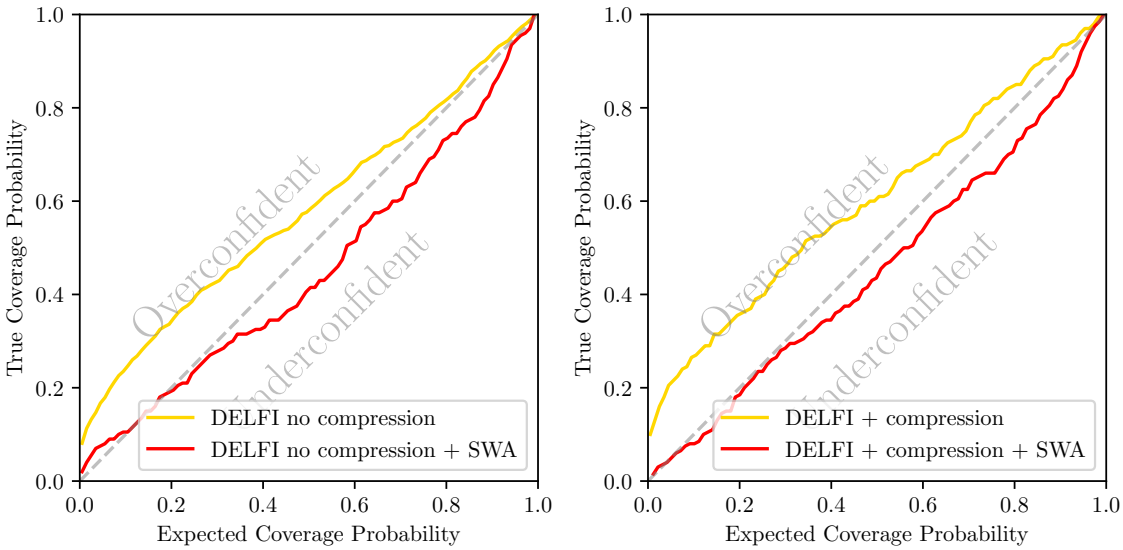

Figure 5. Validation of the generalization analysis shown in Fig. 3.

C.2. Validation of the generalization analysis

We next apply this validation test to the analysis of Subsection 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 Fig. 5. In this case, adding marginalisation allows us to get posteriors of the correct size, with and without data compression.

D. Tuning the Scale Hyperparameter

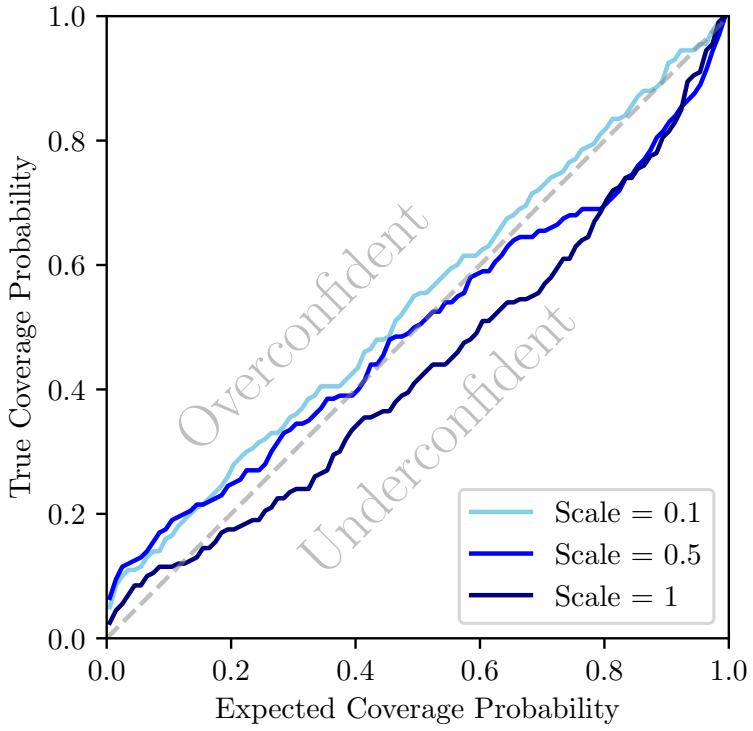

Figure 6. Illustration of how the SWA scale hyperparameter can be calibrated using the validation test described in Appendix C. A larger scale factor leads to more underconfident posteriors, therefore it can be modified to get as close as possible to the diagonal.

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 (Maddox et al., 2019). 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 underconfident posteriors, to avoid biased results. This is illustrated by Fig. 6, which shows this calibration performed for the DELFI with no compression analysis applied to noisy data vectors of Subsection 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.

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.