SimSIMS: Simulation-based Supernova Ia Model Selection with thousands of latent variables

Konstantin Karchev[1] `kkarchev@sissa.it`

Roberto Trotta[1,2,3,4] `rtrotta@sissa.it`

Christoph Weniger[5] `c.weniger@uva.nl`

[1] Theoretical and Scientific Data Science, SISSA, Trieste, Italy
[2] Department of Physics, Imperial College London, United Kingdom
[3] Italian Research Center on High Performance Computing, Big Data and Quantum Computing
[4] National Institute for Nuclear Physics, Trieste, Italy
[5] GRAPPA Institute, University of Amsterdam, The Netherlands

Abstract

We present principled Bayesian model comparison through simulation-based neural classification applied to SN Ia analysis. We validate our approach on realistically simulated SN Ia light curve data, demonstrating its ability to recover posterior model probabilities while marginalizing over > 4000 latent variables. The amortized nature of our technique allows us to explore the dependence of Bayes factors on the true parameters of simulated data, demonstrating Occam's razor for nested models. When applied to a sample of 86 low-redshift SNæ Ia from the Carnegie Supernova Project, our method prefers a model with a single dust law and no magnitude step with host mass, disfavouring different dust laws for low- and high-mass hosts with odds in excess of 100:1.

1 Introduction

Classification problems are a quintessential machine learning task, just as hypothesis testing is at the heart of science. Bayesian model selection improves upon traditional frequentist tests by implementing an automatic quantitative version of Occam's razor (the principle that "simple" models ought to be preferred [1]). Traditionally, calculating Bayesian model evidences has required performing an integral over the model's whole parameter space, which quickly becomes intractable when analysing large data sets with complicated Bayesian hierarchical models (BHMs).

Neural simulation-based inference (SBI)[1] is a relatively recent alternative approach to Bayesian inference that is rapidly gaining popularity in the physical sciences due to its scalability to large data sets and ability to include realistic models. The keystone of SBI is the use of a stochastic simulator able to produce mock data, incorporating arbitrarily complex physical effects difficult to model in likelihood-based pipelines. A neural network (NN) trained on the simulated examples is then used for inference in place of explicit likelihood evaluations. NNs are quick to train via gradient descent, easy to deploy on modern high-performance computing hardware like graphics processing units (GPUs), and allow SBI practitioners to exploit the rapid development in the field of deep learning. Furthermore, amortised inference enables both validation of the approximate posteriors [4–6] as well as the constructions of confidence regions with guaranteed frequentist coverage [7, 8].

Here, we combine the power of SBI with the elegance of Bayesian model selection to perform principled analysis of a BHM with thousands of latent variables. We address the controversial topic of

[1] See [2, 3] for overviews and https://simulation-based-inference.org/ and https://github.com/smsharma/awesome-neural-sbi for references to software and applications.

the possible existence of a "magnitude step" [9, 10] in type Ia supernovæ (SNæ Ia)—standardisable candles that enabled the discovery of the accelerated expansion of the Universe [11–13]—an intrinsic difference in magnitude correlated with the mass of their host galaxies, and its interplay with the host dust properties: see the references in [14] and [15, hereafter TM22], on which we base our modelling. So far, the problem has been plagued by an inability to treat all considered effects self-consistently due to the limitations of likelihood-based analyses, which are lifted by SBI.

2 Simulation-based model selection

Bayesian model selection assigns posterior probabilities $\mathrm{p}(\mathcal{M}_k \,|\, \mathbf{d})$ to models $\mathcal{M}_k \in \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ (instead of to values of their parameters $\boldsymbol{\theta}_k$), conditional on observed data $\mathbf{d}$. The conventional approach is to compute the marginal likelihood (or *evidence*) $\mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k)$, which is the average likelihood $\mathrm{p}(\mathbf{d} \,|\, \boldsymbol{\theta}_k)$ of parameters distributed according to the prior $\mathrm{p}(\boldsymbol{\theta}_k)$:

$$\mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k) = \int \mathrm{p}(\mathbf{d} \,|\, \boldsymbol{\theta}_k)\, \mathrm{p}(\boldsymbol{\theta}_k)\, \mathrm{d}\boldsymbol{\theta}_k \tag{1}$$

(where the presence of $\mathcal{M}_k$'s parameters $\boldsymbol{\theta}_k$ implies conditioning on $\mathcal{M}_k$ in the right-hand side). The prior belief in the model, $\mathrm{p}(\mathcal{M}_k)$, is then updated to its posterior probability in accordance with Bayes' theorem: $\mathrm{p}(\mathcal{M}_k \,|\, \mathbf{d}) \propto \mathrm{p}(\mathbf{d} \,|\, \mathcal{M}_k)\, \mathrm{p}(\mathcal{M}_k)$, normalised over all models considered.

As pointed out by Jeffrey & Wandelt [16], this has two disadvantages: first, it might be unclear what exactly the complete set of model parameters is, in what space they are defined (there are models with varying numbers of parameters: see e.g. trans-dimensional Monte Carlo [17]), and what their likelihood is. For example, in cosmology, so-called *selection effects* arise when the probability of detecting an object and including it in the analysed sample depends on the very parameters of interest. Even when the integral in eq. (1) is well defined, it is usually computationally prohibitive to evaluate for high-dimensional parameter spaces: variants of nested sampling, the *de facto* standard technique for the task, typically only scale up to a few hundreds of parameters [see e.g. 18, 19, for reviews], far from the millions required for contemporary cosmological data sets.[2]

Marginal simulation-based inference circumvents both issues since the simulator abstracts latent parameters from the inference procedure altogether: latent stochastic variables sampled during a forward run are implicitly marginalised. For the purpose of Bayesian parameter estimation, the NN can be trained to approximate either the likelihood, the posterior, or the likelihood-to-evidence ratio. The latter approach, called neural ratio estimation (NRE), recasts the inference task into a classification problem between pairs $\boldsymbol{\theta}, \mathbf{d} \sim \mathrm{p}(\boldsymbol{\theta}, \mathbf{d})$ versus $\boldsymbol{\theta}, \mathbf{d} \sim \mathrm{p}(\boldsymbol{\theta})\, \mathrm{p}(\mathbf{d})$ and uses the classification probability to derive the posterior over model parameters $\boldsymbol{\theta}$. NRE is founded on the well-known principle that, in order to minimise the Bayesian risk of misclassification, a classifier must base its decision on the ratio of the densities of the examples it has been trained on [see e.g. 21], implying that if the classes represent data simulated according to the different models being compared (in proportion to the model priors $\mathrm{p}(\mathcal{M}_k)$), the NN learns their posterior probabilities.

The ratio estimator used in NRE is usually trained to minimise the binary cross-entropy (BCE) loss [see e.g. 4] used for binary classification. In machine learning applications, the case is ubiquitously extended to multiple classes via the multi-class cross-entropy loss, whereby the neural network outputs one real number for each model considered: $\{x_1, \ldots, x_N\}$; these are then normalised via the softmax function: $y_k = \exp(x_k) / \sum_j \exp(x_j)$. Training a sufficiently expressive neural network to maximise the entry corresponding to the true model leads to it outputting (after the normalisation) the posterior probabilities of the models.

Related work. In the field of SN Ia analysis, SBI has focused on marginal parameter inference: of cosmological parameters by using summary statistics derived in likelihood-based fits to light curves [7, 22–26], and of the properties of an individual object from its raw light curve [27].

A number of studies have addressed simulation-based Bayesian model selection in general. Jeffrey & Wandelt [16] focused on loss functions for two-way model comparison with an emphasis on recovering accurate extreme Bayes factors. Radev et al. [28] proposed estimating a Dirichlet distribution over an arbitrary number of models using a NN and variational optimisation, while Elsemüller et al. [29] advocated in favour of a cross-entropy loss, which we use in this work.

[2] With the exception perhaps of proximal nested sampling, which scales to millions-dimensional models [20].

3 Application to SN Ia analysis: magnitude step and dust laws

The data we analyse are light curves (collections of calibrated flux measurements in different passbands and at different times) of the 86 low-redshift SNæ Ia from the Carnegie Supernova Project (CSP) [30] previously investigated by TM22 with a likelihood-based BHM, which we re-implement as a forward simulator. We fix the principal components of the SN Ia spectral time series to those inferred by Mandel et al. [31] and sample the remaining model parameters (including intrinsic light curve variations and dust optical depth) from their respective priors (the NN implicitly marginalises them for the purposes of model comparison) and implement 6 models whose posterior probabilities we wish to evaluate:

- the possible existence of an intrinsic magnitude step between SNæ in low- and high-mass hosts, such that the magnitude scatter follows $\mathcal{N}\left(\Delta M, \sigma_0^2\right)$ for SNæ in high-mass galaxies ($\log_{10}(M_*/M_\odot) > 10.5$), and $\mathcal{N}\left(0, \sigma_0^2\right)$ otherwise[3]; we place a uniform hyperprior $\Delta M \sim \mathcal{U}(-0.2, 0.2)$ and a broad hyperprior on σ_0 as in Mandel et al. [31]); we label with “`dM`” (“`M0`”) the model with (without) a magnitude step, so that `dM` $\rightarrow$ `M0` when $\Delta M \rightarrow 0$;
- the population of host dust parameters R_V^s (which in all cases are restricted to the range $[0.5, 6]$ as in TM22), describing the wavelength dependency of dust absorption in the Fitzpatrick law [32]:
  - a “global” dust model, the simplest among them, has $R_V^s = \mu_R$, i.e. all SNæ are subject to the same dust law (albeit with individual optical depth described by A_V^s);
  - a “local” dust model, assuming a hierarchical relationship $R_V^s \sim \mathcal{N}\left(\mu_R, \sigma_R^2\right)$, i.e. a single population of dust; “local” $\rightarrow$ “global” when $\sigma_R \rightarrow 0$;
  - a “split” dust model, with two independent distributions of R_V^s for high- and low-mass hosts: $R_V^s \sim \mathcal{N}\left(\mu_R^{\text{low}}, (\sigma_R^{\text{low}})^2\right)$ or $\mathcal{N}\left(\mu_R^{\text{high}}, (\sigma_R^{\text{high}})^2\right)$, such that “split” $\rightarrow$ “local” when $(\cdot)^{\text{low}} \rightarrow (\cdot)^{\text{high}}$.

  We use the same priors on global dust parameters as in TM22, a fixed mass split location at $\log_{10}(M_*/M_\odot) = 10.5$ (resulting in a 49/37 split), and stellar masses M_* as included in SNANA [33], ignoring stellar mass uncertainty (see [34]).

We fix the SN cosmological redshifts (after peculiar velocity corrections [35]) and the cosmological model to that used in TM22: a flat ΛCDM with $\Omega_{\text{m}0} = 0.28$ and $H_0 = 73.24\,\text{km/s/Mpc}$ (and SN Ia absolute magnitude $M_0 = -19.5$). Overall, our models have 47 parameters *per SN* (42 of them describing the residual correlated light curve variability) for a total of more than 4000 for the analysed data set with 86 SNæ Ia. For comparison, current state-of-the-art compilations of about 2000 SNæ Ia [36] would require $\sim 10^5$ latent variables.

The neural network we use is implemented in `pytorch` [37] and is based solely on fully connected layers. Full details about the architecture and training are given in appendix A. Before showing results on the real data set, we validate the performance of the trained classifier using a set $\{\mathbf{d}_i\}$ simulated from models $\{\mathcal{M}_i\}$ in proportion to the prior model probabilities (in this case, in equal amounts).

Calibration. We first plot the “reliability diagram”[4] [38], which shows the fraction of examples that were simulated from a given model versus the posterior probability of that model given the simulated data. To produce fig. 1 we bin the validation examples according to the network output for model k (i.e. the posterior probability $\mathrm{p}(\mathcal{M}_k \mid \mathbf{d}_i)$) and within each bin calculate the fraction of examples that were actually simulated from model k. The nearly diagonal lines we observe indicate good calibration.

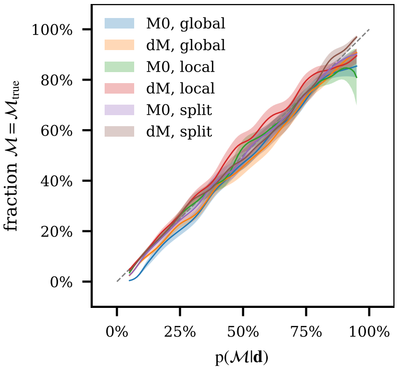

Figure 1: Reliability diagram for the trained classifier network.

[3] Our ΔM has opposite sign to TM22: here, $\Delta M < 0$ corresponds to brighter SNæ Ia in more massive hosts.

[4] Jeffrey & Wandelt [16] use the same diagnostic, calling it “blind coverage testing”.

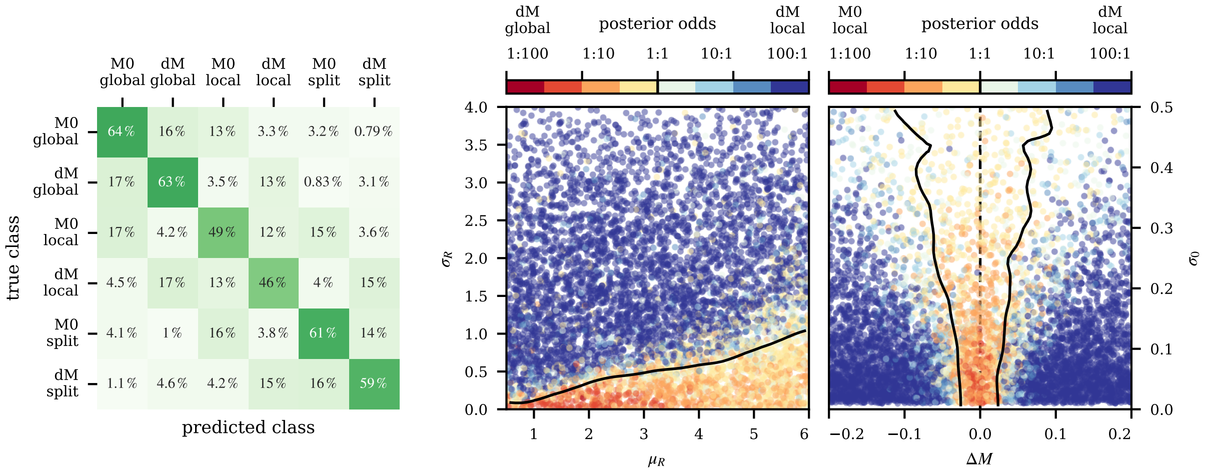

Figure 2: Evaluation of the trained classifier network on the validation set of simulations. *Left:* Each row shows the posterior over models (as labelled above), averaged over a collection of data simulated with the model indicated on the left. *Right:* $\log_{10}$ Bayes factors (evidence ratios) for different simulated datasets as a function of the input parameters. For the μ_R–σ_R plot the compared models are “local” and “global” ($\sigma_R = 0$), marginalising over ΔM, while for the ΔM–σ_0 plot, the models are “dM” and “M0” ($\Delta M = 0$), assuming a non-split R_V^s distribution (“local”). The solid black lines indicate parameters leading *on average* to equal posterior odds.

Refinedness (in the sense of [38]). We show in fig. 2 (left) the average posterior probabilities for data simulated with a given model. A refined classifier would assign the most probability to the “correct” model, leading to a pronounced diagonal; but unlike in usual machine learning applications, Bayesian model comparison assigns non-zero posterior probability to all models (i.e. non-zero off-diagonal entries). The prominence of the diagonal, then, depends both on how powerful the data itself is in distinguishing the models as well as on the parameters’ priors [1].

Owing to amortisation, we are able to explore Bayes factors (ratios of evidences) across a range of ground-truth parameters of simulated data, which is computationally unfeasible with traditional methods. Figure 2 (right), which compares nested models (“local” → “global” in μ_R–σ_R space and dM → M0 in ΔM–σ_0 space), clearly demonstrates Occam’s razor: data resulting from parameters sufficiently close to the location of the nested model ($\sigma_R = 0$ or $\Delta M = 0$) favour the simpler model (yellow/red regions). We also observe that, naturally, a step in magnitudes is harder to detect when their scatter (σ_0) is larger. A scatter in R_V^s (i.e. $\sigma_R > 0$) is also harder to detect when μ_R is large because, in that region, the effect on data is smaller due to the non-linear nature of the dust law.

Results from the CSP data set are presented in fig. 3 in terms of posterior model probabilities and Bayes factors with respect to the most probable model: a global dust law and no magnitude step. Our results follow Occam’s razor, with no clear preference for a mass step and a mildly disfavoured (by a factor ≈2) spread of R_V^s. A split in the dust laws for low- and high-mass hosts is clearly disfavoured, regardless of the magnitude step, with a Bayes factor of ≈100, contrary to the conclusions of both Thorp & Mandel [15] and Brout & Scolnic [14].

In fig. 3, we also present posteriors (derived via NRE trained on the same simulations used for the model comparison network), which support the conclusions of model comparison. In agreement with TM22, we find a magnitude step of $\Delta M \approx -0.05$, and approximately 2σ away from 0, with the results only mildly affected by the dust model. We find a larger value $\sigma_0 \approx 0.2$ (cf. ≈0.1 in TM22) since this quantity in our analysis absorbs all residual variability present in the data, including peculiar velocity uncertainties, which we do not model explicitly. All of the global dust parameter posteriors are in good agreement with TM22, and we obtain similar posteriors when treating low- and high-mass hosts separately as when we assume a single dust distribution (after marginalising over ΔM in all cases). This justifies the “split” dust model being strongly disfavoured, due to its larger prior volume.

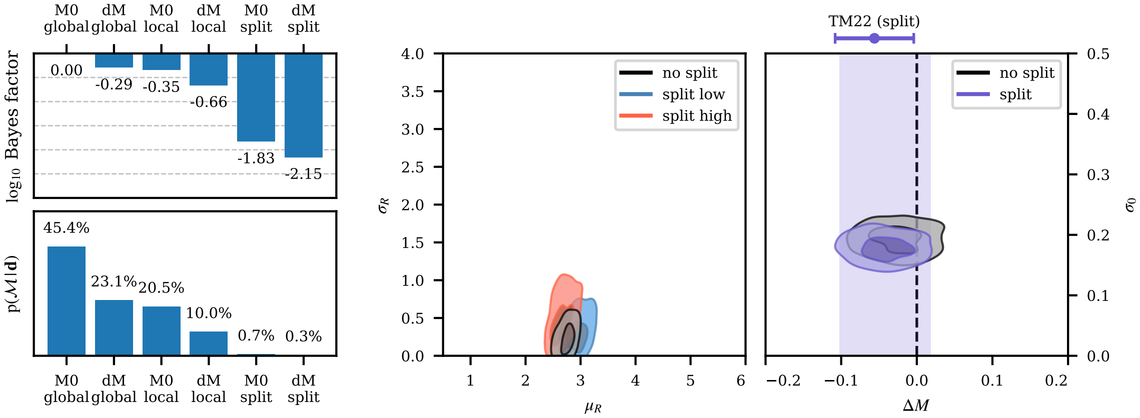

Figure 3: Posteriors from CSP data. *Left:* Models' posterior probabilities (bottom) and (top) $\log_{10}$ Bayes factor with respect to the highest-ranked model (`M0`, global: no step, global dust law). *Right:* Approximate marginal posteriors (1σ and 2σ) from NRE. The μ_R–σ_R plot compares the posteriors for ($\cdot^{\text{low}}$, $\cdot^{\text{high}}$) from the "split" model, with the result for a single dust-law distribution ("no split"). The ΔM–σ_0 plot compares posteriors from the same two models. The shaded strip is the 1-D 2σ marginal ΔM posterior from the "split" model, in agreement with TM22 (2σ error bar above).

4 Conclusions

Enabled by neural SBI, we have performed Bayesian model comparison on an unsolved problem in cosmology that requires realistic modelling of SN Ia light curves and marginalising over thousands of latent variables. A demonstration of Occam's razor, our results from low-redshift SN Ia data favour a global dust law and no magnitude step (with $45\,\%$ posterior probability up from $16.6\,\%$ *a priori*). The existence of a magnitude step or a distribution of R_V^s remain plausible (with posterior odds of $\approx 1:2$), while a split in global dust populations across $\log_{10} M_*/M_\odot = 10.5$ is disfavoured with odds in excess of 100:1. We emphasise, however, that Bayesian model comparison is always dependent on the prior volumes considered. The scalability of our approach allows it to be applied to much larger data sets than demonstrated here, both present and future, with even more sophisticated Bayesian models (e.g. marginalising out the location of the mass split, accounting for stellar mass uncertainty), and more realistic simulators (self-consistently estimating redshifts and peculiar velocities, including selection effects and non-Ia contamination), ushering in the era of principled simulation-based fully Bayesian SN Ia cosmology.

Acknowledgments and Disclosure of Funding

RT acknowledges co-funding from Next Generation EU, in the context of the National Recovery and Resilience Plan, Investment PE1 – Project FAIR "Future Artificial Intelligence Research". This resource was co-financed by the Next Generation EU [DM 1555 del 11.10.22]. RT is partially supported by the Fondazione ICSC, Spoke 3 "Astrophysics and Cosmos Observations", Piano Nazionale di Ripresa e Resilienza Project ID CN00000013 "Italian Research Center on High-Performance Computing, Big Data and Quantum Computing" funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di "campioni nazionali di R&S (M4C2-19)" – Next Generation EU (NGEU). CW acknowledges funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864035).

References

[1] Trotta R., 2008, Contemporary Physics, 49, 71

[2] Cranmer K., Brehmer J., Louppe G., 2020, in Proceedings of the National Academy of Sciences. National Academy of Sciences, pp 30055–30062, doi:10.1073/pnas.1912789117, `https://www.pnas.org/content/117/48/30055`

[3] Lueckmann J.-M., Boelts J., Greenberg D., Goncalves P., Macke J., 2021, in Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. PMLR, pp 343–351, https://proceedings.mlr.press/v130/lueckmann21a.html

[4] Hermans J., Delaunoy A., Rozet F., Wehenkel A., Begy V., Louppe G., 2022, Transactions on Machine Learning Research

[5] Delaunoy A., Hermans J., Rozet F., Wehenkel A., Louppe G., 2022, Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation (arxiv:2208.13624)

[6] Miller B. K., Cole A., Forré P., Louppe G., Weniger C., 2021, in 35th Conference on Neural Information Processing Systems. (arXiv:2107.01214), doi:10.5281/zenodo.5043706

[7] Karchev K., Trotta R., Weniger C., 2023, Monthly Notices of the Royal Astronomical Society, 520, 2209.06733

[8] Crisostomi M., Dey K., Barausse E., Trotta R., 2023, Physical Review D, 108, 044029

[9] Kelly P. L., Hicken M., Burke D. L., Mandel K. S., Kirshner R. P., 2010, The Astrophysical Journal, 715, 743

[10] Sullivan M., et al., 2010, Monthly Notices of the Royal Astronomical Society, 406, 782

[11] Perlmutter S., et al., 1997, The Astrophysical Journal, 483, 565

[12] Perlmutter S., et al., 1999, The Astrophysical Journal, 517, 565

[13] Riess A. G., et al., 1998, The Astronomical Journal, 116, 1009

[14] Brout D., Scolnic D., 2021, The Astrophysical Journal, 909, 26

[15] Thorp S., Mandel K. S., 2022, Monthly Notices of the Royal Astronomical Society, 517, 2360

[16] Jeffrey N., Wandelt B. D., 2023, Evidence Networks: simple losses for fast, amortized, neural Bayesian model comparison (arxiv:2305.11241), doi:10.48550/arXiv.2305.11241, http://arxiv.org/abs/2305.11241

[17] Green P. J., 2003, in Hjort N. L., Richardson S., eds, Oxford Statistical ScienceNo. 27, Highly Structured Stochastic Systems, 1st edn, Oxford University Press, pp 179–206

[18] Ashton G., et al., 2022, Nature Reviews Methods Primers, 2, 39

[19] Buchner J., 2023, Physical Sciences Forum, 5, 46

[20] Cai X., McEwen J. D., Pereyra M., 2022, Proximal nested sampling for high-dimensional Bayesian model selection (arxiv:2106.03646), doi:10.48550/arXiv.2106.03646, http://arxiv.org/abs/2106.03646

[21] Devroye L., Györfi L., Lugosi G., 1996, A Probabilistic Theory of Pattern Recognition, corrected edition edn. Springer, New York

[22] Weyant A., Schafer C., Wood-Vasey W. M., 2013, The Astrophysical Journal, 764, 116

[23] Alsing J., Wandelt B., 2019, Monthly Notices of the Royal Astronomical Society, 488, 5093

[24] Wang G.-J., Cheng C., Ma Y.-Z., Xia J.-Q., 2022, The Astrophysical Journal Supplement Series, 262, 24

[25] Wang G.-J., Cheng C., Ma Y.-Z., Xia J.-Q., Abebe A., Beesham A., 2023, CoLFI: Cosmological Likelihood-free Inference with Neural Density Estimators (arxiv:2306.11102), doi:10.48550/arXiv.2306.11102, http://arxiv.org/abs/2306.11102

[26] Chen J.-F., Wang Y.-C., Zhang T., Zhang T.-J., 2023, Physical Review D, 107, 063517

[27] Villar V. A., 2022, Amortized Bayesian Inference for Supernovae in the Era of the Vera Rubin Observatory Using Normalizing Flows (arxiv:2211.04480), doi:10.48550/arXiv.2211.04480, http://arxiv.org/abs/2211.04480

[28] Radev S. T., D'Alessandro M., Mertens U. K., Voss A., Köthe U., Bürkner P.-C., 2021, Amortized Bayesian model comparison with evidential deep learning (arxiv:2004.10629), doi:10.48550/arXiv.2004.10629, http://arxiv.org/abs/2004.10629

[29] Elsemüller L., Schnuerch M., Bürkner P.-C., Radev S. T., 2023, A Deep Learning Method for Comparing Bayesian Hierarchical Models (arxiv:2301.11873), doi:10.48550/arXiv.2301.11873, http://arxiv.org/abs/2301.11873

[30] Krisciunas K., et al., 2017, The Astronomical Journal, 154, 211

[31] Mandel K. S., Thorp S., Narayan G., Friedman A. S., Avelino A., 2022, Monthly Notices of the Royal Astronomical Society, 510, 3939

[32] Fitzpatrick E. L., 1999, Publications of the Astronomical Society of the Pacific, 111, 63

[33] Kessler R., et al., 2009, Publications of the Astronomical Society of the Pacific, 121, 1028

[34] Shariff H., Dhawan S., Jiao X., Leibundgut B., Trotta R., van Dyk D. A., 2016, Monthly Notices of the Royal Astronomical Society, 463, 4311

[35] Carrick J., Turnbull S. J., Lavaux G., Hudson M. J., 2015, Monthly Notices of the Royal Astronomical Society, 450, 317

[36] Scolnic D., et al., 2022, The Astrophysical Journal, 938, 113

[37] Paszke A., et al., 2019, in Wallach H., Larochelle H., Beygelzimer A., d\textquotesingle Alché-Buc F., Fox E., Garnett R., eds, , Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp 8024–8035

[38] DeGroot M. H., Fienberg S. E., 1983, Journal of the Royal Statistical Society. Series D (The Statistician), 32, 12

[39] Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. R., 2012, Improving neural networks by preventing co-adaptation of feature detectors (arxiv:1207.0580), doi:10.48550/arXiv.1207.0580, http://arxiv.org/abs/1207.0580

[40] Smith L. N., Topin N., 2018, Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates (arxiv:1708.07120), doi:10.48550/arXiv.1708.07120, http://arxiv.org/abs/1708.07120

A Neural network architecture and training

We use a neural network that consists entirely of fully connected linear layers followed by online whitening (which shifts and rescales its inputs to have null mean and unit standard deviation) and rectified linear unit (ReLU) non-linearities. Since the number of observations for each supernova varies from one object to another, we use a bespoke linear layer $\mathbb{R}^{N^s_{\text{obs}}} \to \mathbb{R}^{256}$ to embed each SN in a common-dimensional space. The embedding is processed by two more layers (shared among all SNæ), resulting in a SN featurisation in $\mathbb{R}^{32}$. The resulting $N_{\text{SN}} = 86$ feature vectors are flattened to form a $86 \times 32 = 2752$-dimensional representation of the whole data set, which is fed through three additional layers leading finally to the 6 predicted class probabilities (unnormalised logits input into a `CrossEntropyLoss`). We implement the network (detailed in table 1) in `pytorch` [37], using a $50\,\%$ dropout [39] after the flattening layer. We train on a single Nvidia A-100 GPU using 96 000 examples from each of the 6 models (set to fit in the GPU memory) and a `OneCycle` learning rate schedule [40]. Generating the training data and training until convergence (for about 100 000 steps) took about $1\,\mathrm{h}$ each.

Table 1: Architecture of the neural network we use.

input	shape: $(\sum_{s=1}^{N_{\text{SN}}} N^s_{\text{obs}},)$	WhitenOnline()	
loop $s \in 1, \ldots, N_{\text{SN}}$	$\text{Linear}(N^s_{\text{obs}}, 256)$	WhitenOnline()	ReLU()
Stack	shape: $N_{\text{SN}} \times (256,) \to (N_{\text{SN}}, 256)$		
SNæ as batch dim.	$\text{Linear}(256, 256)$	WhitenOnline()	ReLU()
	$\text{Linear}(256, 256)$	WhitenOnline()	ReLU()
	$\text{Linear}(256, 32)$		
Flatten	shape: $(N_{\text{SN}}, 32) \to (N_{\text{SN}} \times 32,)$		$\text{Dropout}(0.5)$
	$\text{Linear}(256, 256)$	WhitenOnline()	ReLU()
	$\text{Linear}(256, 256)$	WhitenOnline()	ReLU()
	$\text{Linear}(256, 256)$		
	$\text{Linear}(256, 6)$	classifier output	(→ CrossEntropyLoss)