Evidence Networks: simple losses for fast, amortized, neural Bayesian model comparison

To cite this article: Niall Jeffrey and Benjamin D Wandelt 2024 *Mach. Learn.: Sci. Technol.* **5** 015008

View the article online for updates and enhancements.

PAPER

Evidence Networks: simple losses for fast, amortized, neural Bayesian model comparison

Niall Jeffrey[1,*] and Benjamin D Wandelt[2,3]

[1] Department of Physics & Astronomy, University College London, Gower St., London, United Kingdom
[2] Institut d'Astrophysique de Paris (IAP), UMR 7095 CNRS, Sorbonne Université, Paris, France
[3] Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, New York, NY, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: n.jeffrey@ucl.ac.uk

Abstract

Evidence Networks can enable Bayesian model comparison when state-of-the-art methods (e.g. nested sampling) fail and even when likelihoods or priors are intractable or unknown. Bayesian model comparison, i.e. the computation of Bayes factors or evidence ratios, can be cast as an optimization problem. Though the Bayesian interpretation of optimal classification is well-known, here we change perspective and present classes of loss functions that result in fast, amortized neural estimators that directly estimate convenient functions of the Bayes factor. This mitigates numerical inaccuracies associated with estimating individual model probabilities. We introduce the leaky parity-odd power (l-POP) transform, leading to the novel 'l-POP-Exponential' loss function. We explore neural density estimation for data probability in different models, showing it to be less accurate and scalable than Evidence Networks. Multiple real-world and synthetic examples illustrate that Evidence Networks are explicitly independent of dimensionality of the parameter space and scale mildly with the complexity of the posterior probability density function. This simple yet powerful approach has broad implications for model inference tasks. As an application of Evidence Networks to real-world data we compute the Bayes factor for two models with gravitational lensing data of the Dark Energy Survey. We briefly discuss applications of our methods to other, related problems of model comparison and evaluation in implicit inference settings.

1. Introduction

1.1. Background

Bayesian model comparison aims to evaluate the relative posterior probability of different underlying models given some observed data x_O. The ratio of probabilities (the *posterior odds*) for two models, M_1 and M_0, can be related to the *model evidence* via:

$$\frac{p(M_1|x_O)}{p(M_0|x_O)} = \frac{p(x_O|M_1)}{p(x_O|M_0)} \frac{p(M_1)}{p(M_0)}, \tag{1}$$

where $p(x_O|M_1)$ is the *evidence* for model M_1 and $p(M_1)$ is the prior probability of the model. The evidence ratio for the two models is known as the Bayes factor, K (see Jaynes 2003 for details).

Under the assumption of equal prior probability for both models, the Bayes factor (i.e. the ratio of model evidences) becomes equal to the ratio of posterior probabilities for models:

$$K = \frac{p(x_O|M_1)}{p(x_O|M_0)} = \frac{p(M_1|x_O)}{p(M_0|x_O)}, \text{ if } p(M_1) = p(M_0). \tag{2}$$

The evaluation of the Bayes factor, given some data and taking into account all sources of uncertainty, is typically the ultimate goal for model comparison. Bayes factors have been used or discussed in the fields of

epidemiology, engineering, particle physics, cosmology, law, neuroscience, and many others (Wakefield 2009, Yuen 2010, Jasa and Xian 2012, Feroz 2013, Fenton *et al* 2016, Handley *et al* 2019, Keysers *et al* 2020, Massimi 2020). Though the simplest interpretation is that of an odds ratio under equal model priors, interpretations like the *Jeffreys' scale* (Jeffrey 1998) are also popular for Bayesian model comparison. In general these interpretations are concerned with the log Bayes factor, thought of as the relative magnitude of the odds.

1.2. Paper summary

In section 1.3, 'Motivation' we introduce the challenges faced in Bayesian model comparison that are solved by Evidence Networks: intractability of computationally challenging integration, unknown likelihood/prior, and lack of model parameterization. In section 1.4, 'Evidence Networks & comparison with existing methods', we consider existing methods in the context of these three motivations.

In section 2, 'Optimization objective', we show how to construct bespoke losses to estimate convenient functions of the model evidence, with focus on the log Bayes factor. Section 2.1 describes the first case of the 'Direct estimate of model posterior probability'. Section 2.2 introduces our primary case of 'Direct estimate of Bayes factors', which includes the description of the novel 'l-POP' transform and associated 'l-POP-Exponential' loss. We include a 'Summary of "symmetric losses"' in section 2.3.

In section 3, we discuss 'Pitfalls & validation techniques' with Evidence Networks. Section 3.1 describes the practical 'Challenge' in validating a trained Evidence Network in the standard case in which the true Bayes factor of the validation data is not known. Section 3.2 presents a 'Blind coverage testing' solution.

Section 4 presents 'Demonstration: Analytic time-series'. This demonstration uses a 'Generative model' (section 4.2) defined such that the Bayes factors can be calculated analytically using a closed-form expression. Section 4.3, 'Evidence Network results', describes the results of the Evidence Networks, demonstrates the blind coverage test, and explores the choice of loss functions. Alternatives are considered in section 4.4, 'Comparison to state-of-the-art: nested sampling with known likelihood', and section 4.5, 'Alternative likelihood-free method: density estimation'.

Section 5 presents 'Demonstration: Dark Energy Survey data', in which Evidence Networks are used to compare two alternative models of galaxy intrinsic alignments using gravitational lensing data.

Section 6, 'Pedagogical Example: Rastrigin Posterior', addresses a conceptual point: there are cases where estimation of the Bayes factor (e.g. with Evidence Networks) for model comparison can be done even in cases where solving the parameter inference problem is intractable, e.g. due to the complexity of the posterior probability distribution.

We conclude in section 7, with section 7.1 presenting a discussion of 'Simple extensions for further applications': absolute evidence (rather than Bayes factor ratios), the connection to frequentist hypothesis testing, and Evidence Networks for posterior predictive tests.

1.3. Motivation

The challenge for Bayesian model comparison arises in the calculation of the model evidence. Assuming the statistical model is known, the model evidence can be expressed as the *marginal likelihood*:

$$p(x_O|M_1) = \int p(x_O|\theta, M_1) \; p(\theta|M_1) \; \mathrm{d}\theta. \tag{3}$$

This integral over the space of permissible model parameters θ is often intractable. There could be three distinct reasons for this:

(i)   The integral is computationally challenging. This often renders the calculation effectively impossible for a high-dimensional parameter space.
(ii) The probability densities $p(x_O|\theta)$ and $p(\theta)$ are unknown.
(iii) There is no known underlying parameterization that generates the data. For example, real-world data can be drawn from different model classes, and there may be no evident underlying parameterization.

We expand upon these three reasons in section 1.4 below. However, for practitioners used to the marginal likelihood presentation of Bayesian evidence (equation (3)), the key points to consider first are:

- *model evidence does not require an integration over model parameters.*
- *model evidence does not require knowledge of the unnormalized posterior distribution.*

The marginal likelihood is a technique to marginalize the joint distribution $p(x_O, \theta|M_1)$ to recover $p(x_O|M_1)$, however a knowledge of $p(x_O, \theta|M_1)$ is not actually necessary to evaluate the model evidence $p(x_O|M_1)$. Evidence Networks provide an alternative approach that avoids equation (3) entirely.

1.4. Evidence Networks & comparison with existing methods

Evidence Networks are fast amortized neural estimators of Bayes factors for model comparison. In section 2 we describe classes of loss functions that lead to the correct optimization objective for achieving this goal. In section 3 we describe the potential pitfalls and validation techniques that are applicable to real-world applications.

We will now address the three motivations highlighted above, comparing to existing methods and linking to our demonstrations that are presented within this paper.

1.4.1. Motivation (i) - intractable integral

Assuming the integrand of equation (3) is known, the marginal likelihood integral can be intractable in practice. This is despite the large number of computational techniques have been devised to address this problem: straightforward estimation of the evidence integral, reversible jump Markov chain Monte Carlo, the Laplace or Saddle Point Approximation, (annealed) Importance Sampling, Path Sampling, Parallel Tempering, Thermodynamic Integration, and Nested Sampling (for reviews see Kass and Raftery 1995, Han and Carlin 2001, Knuth *et al* 2014).

Consider nested sampling, as it is one of the most popular methods for evidence calculation (Feroz *et al* 2009, Handley *et al* 2015); even this approach becomes onerous for high-dimensional parameter spaces, when $\dim(\theta) \gtrsim 10^2$. We will show that the performance of Evidence Networks does not suffer when there are large numbers of parameters as the evaluation of this integral is side-stepped.

1.4.2. Motivation (ii) - unknown likelihood/prior

Beyond the challenge of evaluating equation (3), the likelihood $p(x_O|\theta, M_1)$ or prior $p(\theta|M_1)$ is not available for many inference tasks. In the implicit inference framework, which includes likelihood-free or simulation-based inference, these unknown distributions can be learned from training data (Papamakario and Murray 2016, Alsing *et al* 2018, Brehmer *et al* 2020, Cranmer *et al* 2020). This parameter inference approach that has enjoyed recent success with scientific applications (Alsing *et al* 2019, Brehmer *et al* 2019, Taylor *et al* 2019, Jeffrey *et al* 2020, Lemos *et al* 2020, Ramanah *et al* 2020). We discuss three alternative options for Bayesian model comparison in this setting.

- One option is to first estimate the unknown distribution and then perform the marginal likelihood integration, equation (3). For example, one might first estimate $p(x|\theta, M_1)$ from simulated data; this is is known as Neural Likelihood Estimation (NLE). This type of approach has been explored in Spurio Mancini *et al* (2022), which leverages a harmonic mean estimator that can be applied, for example, to the learned distributions. However, learning intermediate probability densities in terms of model parameters θ is an additional and unnecessary step, which scales exponentially poorly with the number of model parameters (the curse of dimensionality). Evidence Networks avoid this poor scaling because they do no 'see' the model parameters as the parameters do not appear in the expression for the Bayes factor.

  Typically it also is necessarily to reduce the data dimensionality through compression to make the density estimation tractable. Since this changes the dimensionality of the evidence probability density function, this can significantly alter the Bayesian model evidence values. By comparison, with Evidence Networks we can also use uncompressed data, which preserves the correct Bayes factor estimates.

- Another little-explored option for simulation-based (likelihood-free) inference is direct density estimation of the probability density $p(x|M_1)$, for which the Bayesian factor is just the density ratio evaluated for the observed data: $\frac{p(x_O|M_1)}{p(x_O|M_1)}$.

  This approach is appealing as it also avoids the marginal likelihood integration to directly estimate the Bayes factor for model comparison. However, this approach suffers from the curse of dimensionality for high-dimensional data. This is explored in section 4.5.

- Direct estimation of the model evidence (or model posterior) corresponds to the $\mathcal{M}$-closed limit (i.e. all possible models are accounted for, so that $\sum_i p(M_i) = 1$) as described in Radev *et al* (2020). In this work we explore alternative forms of the loss. We find that for model comparison, rather than taking ratios of individual model posterior estimates, it is better to estimate the Bayes factors directly using a bespoke loss function.

1.4.3. Motivation (iii) - no parameterization

For many real-world model comparison tasks the underlying parameterization is itself unknown. A set of natural images, draws from a generative model, or human-generated data have no obvious model

parameters[4]. Evidence Networks do not need to refer to model parameters to compute Bayes factors and therefore enable Bayesian model comparison even for machine-learned models. The expression for Bayesian evidence (e.g. $p(x_O|M_1)$) makes no reference to parameters. This observation provides a clue that parameters can be bypassed entirely in evaluating Bayes factors.

2. Optimization objective

We will now show how to design classifier training losses that lead to networks that estimate convenient functions of the model evidence (e.g. the log Bayes factor) for a given data set.

We assume training data drawn from $p(x|\mathcal{M})$ for different models $\mathcal{M}$, e.g. for model M_1:

$$x_i \sim p(x|M_1) \tag{4}$$

In the case of simulated data, for the optimization objective to recover the correct model posterior or Bayes factor estimates, we can assume any underlying model parameters θ are drawn from the desired prior distribution, e.g. for model M_1:

$$\begin{aligned} \theta_i &\sim p(\theta|M_1) \\ x_i &\sim p(x|\theta_i, M_1) \quad . \end{aligned} \tag{5}$$

These model parameter values are then not used by the Evidence Network, though their distribution is the implicit prior in the final model evidence or Bayes factor estimation. If we do want to use a different parameter prior $p(\theta)$ distribution this can be taken into account with appropriate θ-dependent weights in the loss function (akin to importance sampling). The training set consists only of data paired with model labels.

The fraction of each model label in the training data acts as an implicit prior of the models. For simplicity, we can assume $p(M_1) = p(M_0)$, which corresponds to each model having equal number of samples in the training data. If a model is over-represented in the training data, or one indeed prefers a given model *a priori*, then a simple re-weighting can be applied to the result.

For a given loss function $\mathcal{V}$, the global optimization objective is given by the functional

$$I[f] = \sum_{m \in \{0,1\}} \int \mathcal{V}(f(x), m) \ p(x, m) \ dx, \tag{6}$$

acting on functions f of the data, using model labels $m \in \{0, 1\}$ for a pair of models M_0 and M_1. Let f^* be the global optimal function, i.e. that which minimizes I; this will be approximated with a neural network. Certain specific loss functions $\mathcal{V}$ yield optimal functions f^* that are estimates of specific useful functions of the model evidence. Overall, we recommend the *l-POP-Exponential Loss* (described in section 2.2.4) to estimate the log Bayes factor.

2.1. Direct estimate of model posterior probability

Though we favor direct estimation of functions of the Bayes factor, we will first review simple optimization objectives that result in neural network estimates of the posterior model probability $p(M_i|x_O)$.

The first of these cases, the Squared Loss, can be interpreted as simple special case of parameter inference (e.g. in the context of Moment Networks: Jeffrey and Wande 2020). The alternative case, the Cross-Entropy Loss, matches the $\mathcal{M}$-closed limit (i.e. all possible models are accounted for such that $\sum_i p(M_i) = 1$) as described in Radev *et al* (2020). In section 4 we will show that, in comparison to direct Bayes factor estimation, this approach of direct model posterior estimation is not ideal for model comparison.

2.1.1. Squared & Polynomial Loss
First let us consider the typical Squared Loss

$$\mathcal{V}(f(x), m) = (f(x) - m)^2. \tag{7}$$

The functional to be optimized from equation (6) is then given by:

$$I[f] = \int (f(x) - 1)^2 \ p(x, M_1) + f(x)^2 \ p(x, M_0) \ dx. \tag{8}$$

---

[4] The weights and biases of neural generative models are parameters, but so numerous that Bayesian Model comparison via marginal likelihood is intractable.

If we minimize this objective using standard methods from variational calculus, i.e. $\delta I = 0$, we recover the solution for the global optimal function:

$$f^*(x) = p(m = 1 | x) = p(M_1 | x) \ \forall \, x, \tag{9}$$

under the assumption that there are two models (so that $p(M_1|x) = 1 - p(M_0|x)$). Therefore, the optimal function, evaluated for the observed data x_O, gives the model posterior for model M_1:

$$f^*(x_O) = p(M_1 | x_O). \tag{10}$$

We can further generalize to a Polynomial Loss with a power $\alpha \in \mathbb{R}$, $\alpha \neq 1$,

$$\mathcal{V}(f(x), m) = m(1 - f(x))^\alpha + (1 - m)f(x)^\alpha, \tag{11}$$

we recover the same result as for the Squared Loss, but with the model posterior probability raised to a power. This results in:

$$f^*(x_O) = p(M_1 | x_O)^{\frac{1}{\alpha - 1}}, \tag{12}$$

where $\alpha = 2$ corresponds to the Squared Loss result, as expected. The choice of α leads to different loss landscapes during training, meaning α is a possible hyper-parameter that can be tuned depending on the use-case.

2.1.2. Cross-entropy loss
If we take the loss function

$$\mathcal{V}(f(x), m) = m \log(f(x)) + (1 - m) \log(1 - f(x)) \ , \tag{13}$$

following the same procedure as before gives the result:

$$f^*(x_O) = p(M_1 | x_O) \ . \tag{14}$$

2.2. Direct estimate of Bayes factors
2.2.1. Exponential Loss
Taking the loss function

$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right)f(x)} \tag{15}$$

gives the global optimization objective:

$$I[f] = \int e^{-\frac{1}{2}f(x)} p(x, M_1) + e^{\frac{1}{2}f(x)} p(x, M_0) \ \mathrm{d}x. \tag{16}$$

Following the same procedure as previously, i.e. minimizing this objective using variational calculus, recovers the optimal function:

$$f^*(x) = \log\left(\frac{p(M_1 | x)}{p(M_0 | x)}\right) \ \forall \, x. \tag{17}$$

Therefore, the optimal function evaluated for our observed data is:

$$f^*(x_O) = \log\left(K \frac{p(M_1)}{p(M_0)}\right) = \log K + \log\left(\frac{p(M_1)}{p(M_0)}\right). \tag{18}$$

If the model priors are equal (having the same number of examples of each model label in the training data for the neural network), then the second term vanishes.

2.2.2. Logistic loss

An alternative form gives the same final result as the Exponential Loss, with

$$\mathcal{V}\left(f(x),m\right) = \log\left(1 + e^{(1-2m)f(x)}\right). \tag{19}$$

Following the same optimization procedure as before once again results in an estimate of the log Bayes factor with an additional model prior ratio term:

$$f^*\left(x_O\right) = \log K + \log\left(\frac{p\left(M_1\right)}{p\left(M_0\right)}\right). \tag{20}$$

2.2.3. α-Exponential and log-α-Exponent

Analogously to the generalization from Squared Loss to Polynomial Loss with the α hyper-parameter, we can introduce a power of α to the losses that directly estimate the Bayes factor or log-Bayes factor.

The 'α-Exponential' Loss is given by:

$$\mathcal{V}\left(f(x),m\right) = \left(1 + e^{(1-2m)f(x)}\right)^{\alpha}. \tag{21}$$

Following the same optimization procedure as before, we recover results the estimate of the rescaled log Bayes factor with an additional model prior ratio term:

$$\begin{aligned} f^*\left(x_O\right) &= \log K^{\frac{1}{1+\alpha}} + \log\left(\frac{p\left(M_1\right)}{p\left(M_0\right)}\right)^{\frac{1}{1+\alpha}}, \\ &= \frac{1}{1+\alpha}\log K + \frac{1}{1+\alpha}\log\left(\frac{p\left(M_1\right)}{p\left(M_0\right)}\right). \end{aligned} \tag{22}$$

Similarly, we can construct the 'α-log-Exponential' Loss:

$$\mathcal{V}\left(f(x),m\right) = f(x)^{\left(\frac{1}{2}-m\right)\alpha}. \tag{23}$$

For this case, we analogously recover the Bayes factor to a power, multiplied by a model prior ratio factor:

$$f^*\left(x_O\right) = K^{\frac{1}{1+\alpha}} \times \left(\frac{p\left(M_1\right)}{p\left(M_0\right)}\right)^{\frac{1}{1+\alpha}}. \tag{24}$$

2.2.4. l-POP-Exponential Loss

Finally, we present the 'l-POP-Exponential Loss', which is our recommended choice of default loss for the Evidence Network. This loss applies a bespoke transformation to the network output followed by the usual Exponential Loss. This is completely equivalent to defining a final layer activation function; for this discussion we include the transformation within the loss to aid interpretability.

We first define the *parity-odd power* transformation, as:

$$\tilde{\mathcal{J}}_{\alpha}\left(x\right) := x\left|x\right|^{\alpha-1} \ \ \forall\, x \in \mathbb{R} \text{ where } \alpha \in \mathbb{R} \text{ and } \alpha \geqslant 1. \tag{25}$$

Using the sign function (sgn), another way of representing the parity-odd power transformation is:

$$\tilde{\mathcal{J}}_{\alpha}\left(x\right) = \operatorname{sgn}\left(x\right)\,\left|x\right|^{\alpha}. \tag{26}$$

This function is differentiable everywhere. Note, however, that the intermediate sgn or absolute value functions are not differentiable at $x = 0$, so care should be taken in code implementation.

Though the parity-odd power transform $\tilde{\mathcal{J}}$ works well, its value and gradient remains close to zero, an effect which increases with increasing $\alpha > 1$. We therefore use the *leaky parity-odd power*, or l-POP, transform, which we define as:

$$\mathcal{J}_{\alpha}\left(x\right) := x + x\left|x\right|^{\alpha-1}. \tag{27}$$

IOP Publishing ·

Mach. Learn.: Sci. Technol. **5** (2024) 015008 · N Jeffrey and B D Wandelt

Table 1. A set of simple loss functions that can be used to estimate useful functions of the model posterior probability (in particular the Bayes factor K for Bayesian model comparison).

Loss name	$\mathcal{V}(f(x), m)$ using model label $m \in \{0, 1\}$	$f^*(x_0)$ if $p(M_1) = p(M_0)$
Polynomial	$m(1 - f(x))^\alpha + (1 - m)f(x)^\alpha$	$p(M_1 \mid x_O)^{\frac{1}{\alpha - 1}}$
Cross Entropy	$-m\log(f(x)) - (1 - m)\log(1 - f(x))$	$p(M_1 \mid x_O)$
Exponential	$e^{(\frac{1}{2} - m)f(x)}$	$\log K$
Logistic	$\log(1 + e^{(1 - 2m)f(x)})$	$\log K$
α-Exponential	$(1 + e^{(1 - 2m)f(x)})^{\alpha - 1}$	$\frac{1}{\alpha}\log K$
α-log-Exponent	$f(x)^{(\frac{1}{2} - m)\alpha}$	$K^{\frac{1}{\alpha}}$
l-POP-Exponential	$e^{(\frac{1}{2} - m)\mathcal{J}_\alpha(f(x))}$	$\mathcal{J}_\alpha^{-1}(\log K)$

We could choose to generalize this further, using $\beta x + x\,|x|^{\alpha - 1}$, though we effectively choose $\beta = 1$ for all that follows. The resulting parity-odd power Exponential Loss, or 'l-POP Exponential Loss' is given by:

$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right)\mathcal{J}_\alpha(f(x))}. \tag{28}$$

As discussed above, this is just equivalent to using $\mathcal{J}_\alpha$ as a novel choice of activation function in the network output layer.

Following the same optimization procedure as before, we recover a rescaled estimate of the log Bayes factor plus an additional model prior ratio term:

$$
\begin{aligned}
f^*(x_O) &= \mathcal{J}_\alpha^{-1}\left(\log K + \log\frac{p(M_1)}{p(M_0)}\right) \\
&= \mathrm{sgn}\left(\log K + \log p(M_1) - \log p(M_0)\right)\left|\log K + \log p(M_1) - \log p(M_0)\right|^{\frac{1}{\alpha}}.
\end{aligned} \tag{29}
$$

For equal model priors, $p(M_1) = p(M_0)$, we expect that

$$\log K = \mathcal{J}_\alpha\left(f(^*x_O)\right), \tag{30}$$

for the optimal trained network f^*.

As with many of the losses that we have previously described, the l-POP-Exponential Loss results in a (transformed) estimate of the log Bayes factor, which is ideal for Bayesian model comparison (e.g. the Jeffreys Scale).

We recommend a default value of $\alpha = 2$. This provides the reweighting of errors for high Bayes factor values, ensuring accuracy across many orders of magnitude, while being sufficiently simple that the network trains consistently well.

The l-POP transformation $\mathcal{J}_\alpha$ to act on $f(x)$ is not the only available choice that could be used to construct useful losses. Any strictly monotonic function $\mathcal{J} : \mathbb{R} \to \mathbb{R}$ would fulfil the condition that $\mathcal{J}(f(^*x_O)) = \log K$. Nevertheless, the choice is not arbitrary. For example, we have considered the sinh function in the place of $\mathcal{J}_\alpha$, but find the exponential scaling for high K leads to unreliable performance. Since we have empirically found our l-POP transform $\mathcal{J}_\alpha$ to give accurate results for several test cases presented in this paper we recommend it as the default choice.

2.3. Summary of 'symmetric losses'

In table 1, we summarize both the losses presented in the previous section and the corresponding optimal network outputs (the latter shown in terms of model posteriors or Bayes factors).

This list is not at all comprehensive; we have not attempted to study the family of all loss functions that can be used to construct an Evidence Network. Nor are all of these loss functions novel to this paper. The Squared Loss (a special case of the Polynomial), Cross-Entropy, Exponential, and Logistic have long been known in terms of Bayes risk for classification (Masnadi-Shiraz and Vasconcelos 2008). The '$\alpha-$'type losses are new variations on existing losses and the POP or l-POP Exponential losses are novel to this paper. The main aim, however, is to use these losses in such a way that we can perform quantitative Bayesian model comparison via an Evidence Network.

Furthermore, we have included only 'symmetric losses', i.e. those that are symmetric with respect to relabelling M_0 and M_1. There are many alternative loss functions without this property; such functions might be useful for certain applications (e.g. when it is known *a priori* that the data will prefer one model far more than another).

For a network to calculate $p(M_1|x)$, such an 'asymmetric' loss might have the form:

$$\mathcal{V}(f(x), m) = m\mathcal{A}(f(x)) + (1-m)\mathcal{B}(f(x)), \tag{31}$$

for a choice of functions $\mathcal{A}$ and $\mathcal{B}$ where $\mathcal{A}(f(x)) \neq \mathcal{B}(1-f(x))$. Some of these losses have been explored for computing likelihood-ratios using classifiers in Rizvi *et al* (2023).

3. Pitfalls & validation techniques

3.1. Challenge
Even if *theoretically* there exists an optimal function f^* that minimizes the global optimization objective equation (6), in practice three factors (typical of neural methods) prevent us from finding it exactly. First, we must approximate the space of functions with a set of neural network architectures. Second, we have access to only a finite number of samples from $p(x)$ and hence we only have a (Monte Carlo) approximation to the optimization objective, via:

$$I[f] \approx \frac{1}{N} \sum_{j=1}^{N} \mathcal{V}(f(x_j), m_j). \tag{32}$$

Third, we have no means of guaranteeing that a local optimum discovered during training is in fact a global optimum. It is, therefore, inevitable that there will be some error associated with our estimate of the Bayes factor.

In the next section 4, we will demonstrate that the Evidence Network does indeed recover robust estimates of the Bayes factor. For such synthetic tests, we can fully validate our approach by choosing a model for which the true Bayes factors can be calculated analytically (i.e. in closed-form).

In actual applications, the likelihood and or prior are intractable or implicitly defined only through labeled training data (e.g. consisting only of data realizations and their associated model labels from simulations or a generative model). In this setting a closed-form solution will neither be available nor necessary since the true Bayes factors or model posteriors are not required to train an Evidence Network. Despite this, it is still necessary to validate the trained network. We will now describe a method for validating the Bayes factor computation by the Evidence Network that is generally applicable and requires only labeled training data of the same type that is needed to train the Evidence Network itself.

3.2. Blind coverage testing
The solution to this unique validation problem is blind coverage testing. We can compare the estimate of $p(M_1|x)$ to the fraction of validation data with the model label M_1.

Let us assume that the trained neural network is optimal, such that the network recovers the function f^*. For example, if the network is trained using the l-POP-Exponential Loss (equation (28)), for equal model priors we expect that:

$$\mathcal{J}_\alpha(f^*(x_O)) = \log K. \tag{33}$$

For simplicity, consider the case of only two models M_1 and M_0 (this can be easily generalized for any number of models). We can then rearrange the previous expression to give the model posterior for M_1:

$$p(M_1|x) = \frac{\exp(\mathcal{J}_\alpha(f^*(x_O)))}{1 + \exp(\mathcal{J}_\alpha(f^*(x_O)))}. \tag{34}$$

Such an expression can be constructed for any of the losses in table 1 that estimate some function of K.

Fix P and let $\epsilon > 0$. Consider validation data $x^v \sim p(x)$ for which the posterior probability $p(M_1|x^v)$ lies in $[P, P+\epsilon]$. As $\epsilon \to 0$, the expected fraction of such data having label M_1 converges to P. By comparing the true fraction of model labels in some small probability range with that expected from the estimated K, we can validate the trained network without knowing the true model posterior probabilities.

Put another way, consider a set of N validation data samples that all have the same posterior probability $p(M_1|x)$. In this set, if the number of data samples that are drawn from model M_1 is n_1, then the expected fraction of data from model M_1 is given by $\langle \frac{n_1}{N} \rangle = p(M_1|x)$, following the usual definition of posterior probability. In practice, no two samples in the validation data have exactly equal posterior probabilities, so we use small bins of probability to calculate this fraction (corresponding to a small ϵ). To quantify the statistical significance of the scatter in the probability bins, we can use a binomial distribution to model the count statistics (see section 5 for an example).

This procedure resembles the re-calibration of class probabilities in classification tasks (Zadrozny and Elkan [2002](), Niculescu-Mizil and Carua [2005]()). In this work, it is used as a blind validation of our Bayes factor estimates for Bayesian model comparison.

4. Demonstration: analytic time-series

4.1. Overview

We construct two generative models to sample time series data x. These are defined such that the model evidences and associated Bayes factors can be calculated analytically using a closed-form expression. In section [4.2](), this is compared to the estimated log Bayes factors from our simple Evidence Network.

We compare the Evidence Network to likely alternative methods. Even if the true likelihood/prior were known, with the use of the most advanced nested sampling methods (Handley *et al* [2015]()) the problem can still be too arduous for high parameter-space dimensionality $\dim(\theta) \gtrsim 10^2$; we demonstrate this in section [4.4](). Assuming that we only have data samples and model labels, we would have no knowledge of the distributions $p(x|\theta)$ or $p(\theta)$ to evaluate the marginal likelihood (equation ([3]())). As previously discussed, one could try to estimate those probability densities using implicit inference, but with high data dimensionality $\dim(x) \gtrsim 10^2$ or high parameter-space dimensionality $\dim(\theta) \gtrsim 10^2$, density estimation becomes intractable or prone to significant error; we show this in section [4.5]().

4.2. Generative model

We construct two nested models that generate time series data with heteroschedastic random noise. Both models are linear Gaussian models with a linear operation that is non-trivial with respect to time.

The prior for the model parameters θ is a simple normal distribution for each parameter θ_i:

$$p(\theta_i) = \mathcal{N}(\theta; 0, 1). \tag{35}$$

Per data (vector) realization x, each element x_j is given by:

$$x_j = \sum_i A_{ji}\theta_j + n_j = \mu_j^x + n_j, \tag{36}$$

where the heteroscedastic noise is drawn from a multivariate normal distribution, such that:

$$p(n) = \mathcal{N}(n; 0, \Sigma). \tag{37}$$

The noise covariance Σ is a diagonal matrix with diagonal elements Σ_{kk} given by:

$$\sqrt{\Sigma_{kk}} = \sqrt{\frac{N}{100}\left(\left(k + \frac{5}{2}\right)\frac{8}{5}\right)^2}. \tag{38}$$

For the first model, M_1, the linear operation is given by:

$$A_{ji} = \cos\left(\left(i - \frac{1}{2}\right)t_j\right) \tag{39}$$

for $i > 0$. For $i = 0$, we have

$$A_{j0} = 2t_j. \tag{40}$$

The $i = 0$ term corresponds to a linear growth over time. For the second model, M_0, this first term is removed, so there is no linear growth term over time. This is equivalent to setting $\theta_0 = 0$.

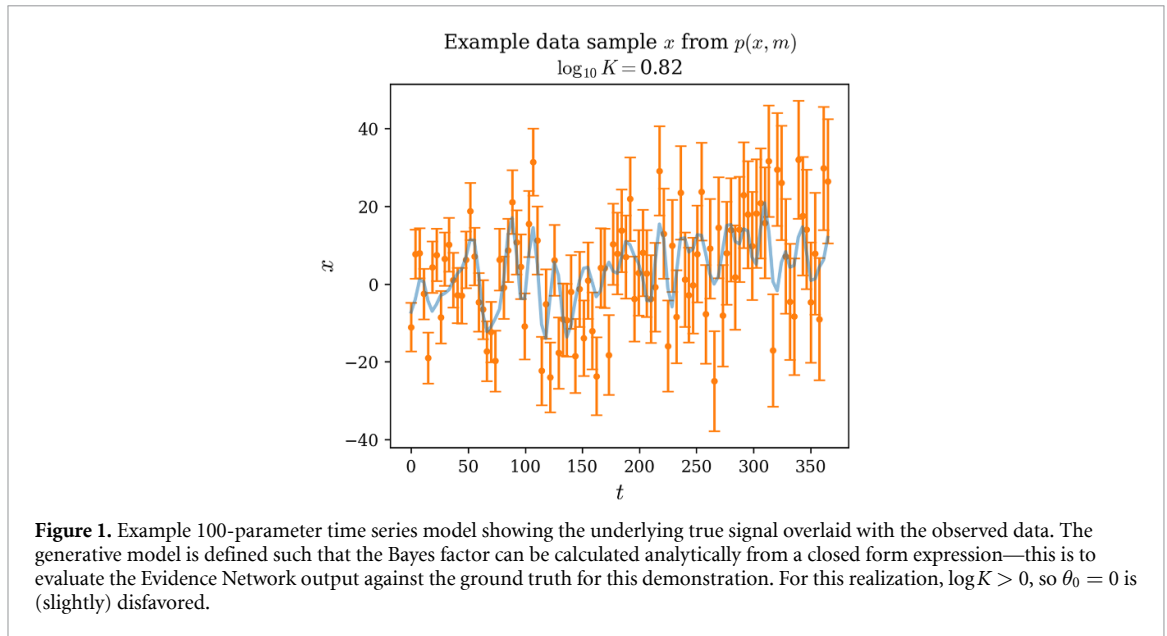An example data sample is shown in figure [1]().

In both of these cases we can evaluate the marginal evidence integral (equation ([3]())) analytically to recover a closed-form solution for the Bayesian model evidence, which is equivalent to evaluating a normal distribution:

$$p(x|\mathcal{M}) = Z(x|\mathcal{M}) = \mathcal{N}(x; \mu_Z, C_Z), \tag{41}$$

where

$$\mu_Z = A \cdot \mu^x, \tag{42}$$

Figure 1. Example 100-parameter time series model showing the underlying true signal overlaid with the observed data. The generative model is defined such that the Bayes factor can be calculated analytically from a closed form expression—this is to evaluate the Evidence Network output against the ground truth for this demonstration. For this realization, $\log K > 0$, so $\theta_0 = 0$ is (slightly) disfavored.

and

$$C_Z = \Sigma + AA^T. \tag{43}$$

This can be simply evaluated for each model to calculate the ratio to give the Bayes factor K:

$$K = \frac{Z(x|M_1)}{Z(x|M_0)}. \tag{44}$$

4.3. Evidence Network results

Using data drawn from the generative model described in the previous Subsection, we train the Evidence Network using the l-POP-Exponential Loss with $\alpha = 2$ to recover a simple transformation of $\log K$ (see table 1). This choice of loss function is our default recommendation:

$$\text{l-POP-Exponential}\, Loss : e^{\mathcal{J}_{\alpha=2}(f(x))} = e^{\left(\frac{1}{2}-m\right)(f(x)+f(x)|f(x)|)} \implies f^*(x_O)\left(1+|f^*(x_O)|\right) = \log K. \tag{45}$$

Using the model labelling described above, a Bayes factor greater than one corresponds to a preference for model M_1 (i.e. $\theta_0 = 0$ is not preferred in this case).

We find that taking the average output of an ensemble of networks, each with independent random weight initialization, significantly improves the accuracy of our log Bayes factor estimation. For these results, each network has the identical architecture.

We have chosen a simple network with little tuning that uses 6 dense layers (see appendix for details). The network was purposefully chosen to be simple to demonstrate that the network does not need to be tuned for the Bayes factor estimates to be reliable.

Though we have chosen a simple dense architecture, Evidence Networks can use any type of network architecture, provided they use the correct optimization objective and are validated using our proposed coverage test. Depending on the application, they could use geometric architectures (e.g. deep sets, graph networks; Battagli *et al* 2018), recurrent neural networks (e.g. long short-term memory—LSTM; Hochreiter and Schmidhuber 1997), or any appropriate architecture.

Our main result uses the generative model described above with $\dim(\theta) = 10^2$ and $\dim(x) = 10^2$. As the data samples for this generative model are symmetric under a sign flip, we make use of this standard data augmentation during training.

The left panel of figure 2 shows our main result with the ensemble Evidence Network. We recover estimates of the Bayes factor over many orders of magnitude. When compared with the analytic (true) Bayes factors, we calculate the root-mean-square error (RMSE) and find a $\log K$ error ≈ 0.02.

This is an excellent estimation of the Bayes factor for such a high dimensional problem. The high dimensionality of the data and parameters render this task either intractable or error-prone for alternative methods as we demonstrate in the following subsections.

Figure 2. *Left panel*: Estimated log Bayes factor K using an ensemble Evidence Network (with four networks) compared to analytic calculation using a closed-form expression for K. These data are time series with 100 data elements draw from a generative model with 100 model parameters. This result uses our default network architecture with 10^6 training samples. As we show in section 4, all standard methods to compute the Bayes factor either fail or incur significant error for this high-dimensional example. *Right panel*: Evidence Network model posterior probabilities derived from the estimated Bayes factors (equation (34)) compared with the relative model fraction in the validation data. This allows validating the Evidence Network output when ground-truth model evidence is unavailable.

Figure 3. Three choices of loss to estimate the log Bayes factor K. Each panel shows results from the same network architecture; we use only a single network, rather than an ensemble as in figure 2, as this is sufficient to show the different systematic errors with increasing K. The *left panel* uses the Cross Entropy Loss to estimate individual model posteriors from which the Bayes factor is estimated. The *centre panel* uses the Exponential Loss to estimate $\log K$. The *right panel* uses the l-POP-Exponential Loss ($\alpha = 2$) to estimate $\log K$.
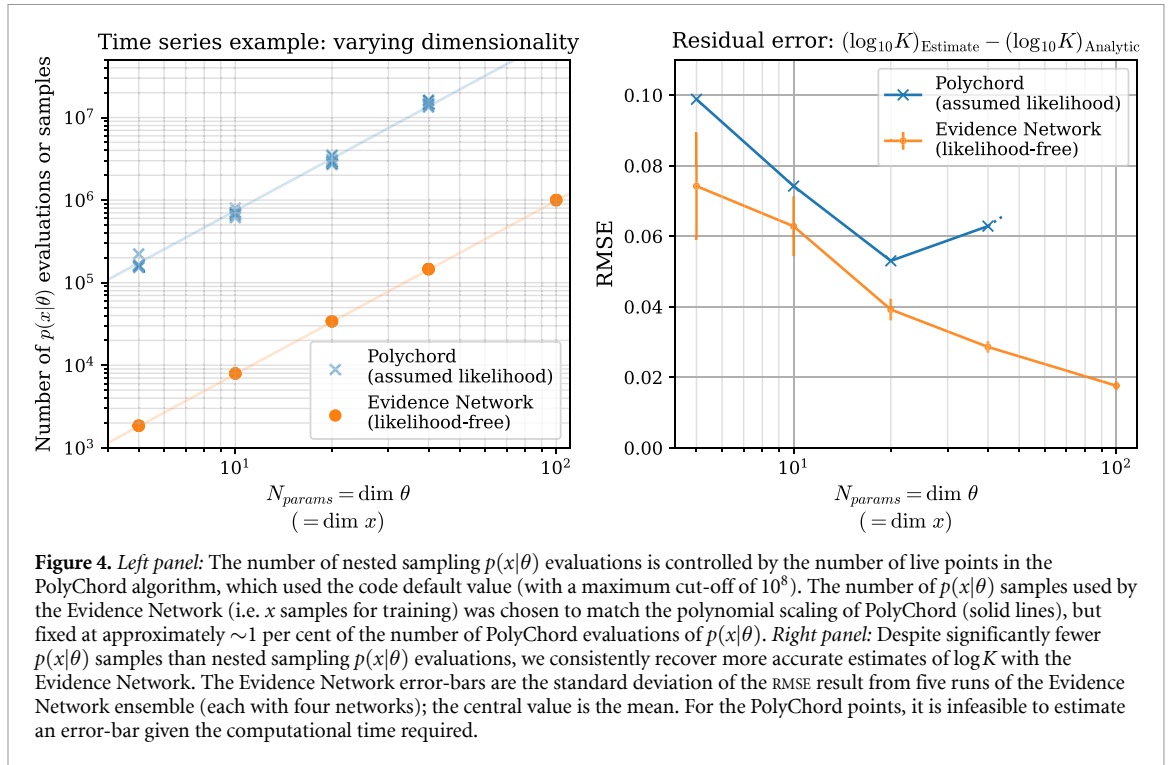
For most applications, the true Bayes factor value will typically not be available for validation; our training data consists only of data realizations with their associated model label. We can, however, leverage our blind coverage test to validate the model posterior probabilities are recovered. This is shown in the right panel of figure 2 for the same trained network and data.

The plotted error-bars in the right panel of figure 2 are the expected standard error for binomial distributed samples: $\sigma_{\mathrm{err}} = \sqrt{p(M_1|x_O)(1 - p(M_1|x_O))/n_p}$, where n_p are the number of data samples in a given p probability bin. We find the measured standard deviation of the residual error (between the expected and the predicted probabilities) does indeed match σ_{err}, further validating the method.

For a general problem where the true Bayes factors are not known, the result of such a blind coverage test can be used to construct error bounds for the Bayes factors estimates.

Figure 3 shows different choice of loss functions. The *left panel* does not use a loss function to estimate the Bayes factor directly, but uses the Cross Entropy Loss to individually estimate $p(M_1|x_O)$ and $p(M_0|x_O)$, whose ratio gives the Bayes factor K. This approach leads to systematic inaccuracies for moderately high values of K. These inaccuracies are ameliorated with the choice of the Exponential Loss (*center panel*), from which the Evidence Network directly estimates $\log K$. Using the l-POP-Exponential Loss (*right panel*) the systematic error at high K is no longer apparent.

Figure 4. *Left panel:* The number of nested sampling $p(x|\theta)$ evaluations is controlled by the number of live points in the PolyChord algorithm, which used the code default value (with a maximum cut-off of 10^8). The number of $p(x|\theta)$ samples used by the Evidence Network (i.e. x samples for training) was chosen to match the polynomial scaling of PolyChord (solid lines), but fixed at approximately ~ 1 per cent of the number of PolyChord evaluations of $p(x|\theta)$. *Right panel:* Despite significantly fewer $p(x|\theta)$ samples than nested sampling $p(x|\theta)$ evaluations, we consistently recover more accurate estimates of $\log K$ with the Evidence Network. The Evidence Network error-bars are the standard deviation of the RMSE result from five runs of the Evidence Network ensemble (each with four networks); the central value is the mean. For the PolyChord points, it is infeasible to estimate an error-bar given the computational time required.

4.4. Comparison to state-of-the-art: nested sampling with known likelihood

In this subsection we compare the Evidence Network with nested sampling; we assume the correct likelihood for nested sampling, while restricting the Evidence Network to the likelihood-free problem, and find that the Evidence Network is still able to outperform nested sampling for high-dimensional parameter spaces.

Let us assume we know the likelihoods, $p(x|\theta, M_1)$ and $p(x|\theta, M_0)$, and priors, $p(\theta|M_1$ and $p(\theta|M_0)$. We assume that we are able to either evaluate a given likelihood, $\mathcal{L}_1(\theta) = p(x_O|\theta, M_1)$, or sample data realizations from it, $x_i \sim p(x|\theta, M_1)$. For a Gaussian likelihood, these two operations are of equal computational complexity.

We use the PolyChord (Handley *et al* 2015) code to perform the nested sampling; this uses an advanced algorithm that combines both cluster detection for multi-modal distributions and slice sampling using affine 'whitening' transformations. We use the default number of LIVE POINTS$= 25 \times \dim(\theta)$, which then leads to a certain number of $p(x|\theta)$ evaluations. We use a maximum cut-off of 10^8 likelihood evaluations for reasons of computation time.

We choose the number of $p(x|\theta)$ samples for the Evidence Network to match the same polynomial scaling with $\dim(\theta)$ as PolyChord evaluation, but with only approximately 1 per cent of that of PolyChord. Despite significantly fewer $p(x|\theta)$ samples than nested sampling $p(x|\theta)$ evaluations (figure 4, *left panel*), we consistently recover more accurate estimates of $\log K$ with the Evidence Network (figure 4, *right panel*).
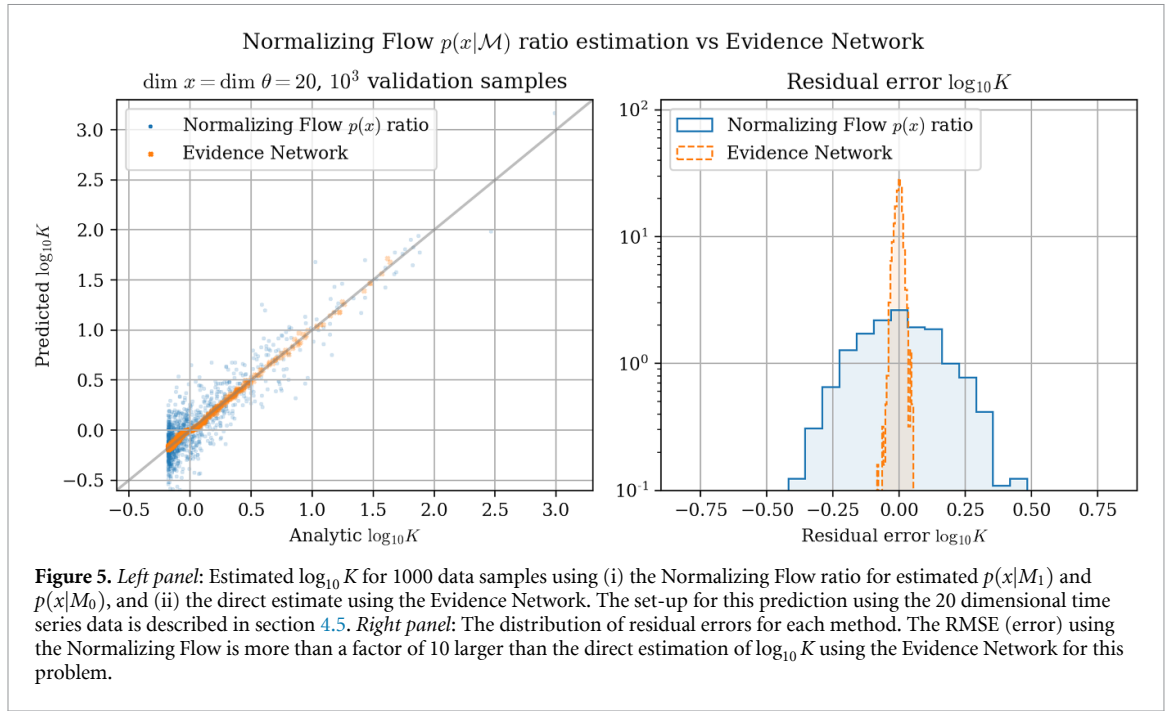
This is an indication that it may be more efficient to train an Evidence Network to estimate Bayes Factors *even when the likelihood is known*. For many closed-form likelihoods, it is a relatively simple code change to replace the $p(x|\theta)$ evaluation with a $p(x|\theta)$ sample of x.

4.5. Alternative likelihood-free method: density estimation

Let us now consider alternative approaches that are available for likelihood-free (e.g. simulation-based) inference problems.

As previously discussed in section 1.4, we could first learn the form of $p(x|\theta)$ from the training data; this is known as NLE. We would still need to perform the high-dimensional marginal evidence integration and using nested sampling (or a similar method). This step is unnecessary: the intermediate density estimation step can only induce potential errors and we nevertheless return to the difficult procedure discussed in section 4.4.

An alternative density estimation approach would be to learn the form of $p(x|\mathcal{M})$ from the training data. So far, this method has not been widely considered for scientific model comparison, even though it would also avoid marginal evidence integration. Therefore, we compare Evidence Networks with this density estimation approach.

Figure 5. *Left panel*: Estimated $\log_{10} K$ for 1000 samples using (i) the Normalizing Flow ratio for estimated $p(x|M_1)$ and $p(x|M_0)$, and (ii) the direct estimate using the Evidence Network. The set-up for this prediction using the 20 dimensional time series data is described in section 4.5. *Right panel*: The distribution of residual errors for each method. The RMSE (error) using the Normalizing Flow is more than a factor of 10 larger than the direct estimation of $\log_{10} K$ using the Evidence Network for this problem.

We use an ensemble of neural density estimators to estimate $p(x|M_1)$ and $p(x|M_0)$ and use an ensemble of neural networks (matching the set-up used so far) to form the Evidence Network to estimate $\log K$. In both cases we use 10^6 samples of training data x. Density estimation suffers particularly badly from increasing dimensionality. We will, therefore, reduce from a dimensionality of $\dim(x) = 100$ (as in figure 2) to a more manageable $\dim(x) = \dim(\theta) = 20$ for this comparison.

We use Normalizing Flows for neural density estimation; we use Neural Spline Flows (Durkan *et al* 2019), which are a popular and flexible neural density estimation method. As an implementation, we use the `PZFlow` package (Crenshaw *et al* 2021) with the default settings.

Figure 5 shows the comparison between the two methods. The *left panel* shows the analytic $\log K$ value against the estimated value using both the Evidence Network and the Normalizing flow ratio (using $p(x|M_1)/p(x|M_0)$). The relatively large residual scatter for the Normalizing Flow can also be seen in the *right panel*, which shows the histogram of the residual errors.

The RMSE (error) in the $\log K$ using the Normalizing Flow is more than a factor of 10 larger than the Evidence Network. The reason for this is twofold: (i) density estimation suffers exponentially from increasing dimensionality (which is even a problem for this restricted 20 dimensional data), and (ii) the errors in each $p(x|\mathcal{M})$ combine in the ratio that gives $\log K$.

5. Demonstration: Dark Energy Survey (DES) data

The DES is a major, ongoing galaxy survey that, among other science goals, measures the shapes of galaxies in order to use the gravitational lensing effect to make cosmological inferences. With its wide-field camera mounted on the Blanco 4-meter telescope in Chile, DES captures high-resolution images of the southern sky, allowing for the measurement of lensing distortions using the shapes of millions of galaxies. By analyzing these distortions statistically, DES provides information on the distribution of matter, both visible and dark, on large scales. However, to interpret these observations accurately, it is crucial to account for intrinsic alignment of the galaxies, as this can introduce systematic biases. For a summary of weak gravitational lensing with the DES Year 3 data see Jeffrey *et al* (2021), Amon *et al* (2022), Secco *et al* (2022). In this section, we demonstrate the use of Evidence Networks to diagnose or detect intrinsic alignment effects. The aim of this analysis is to demonstrate the ease of Bayesian model comparison with Evidence Networks for real-world data.

Using DES as a practical example, we construct a simple Bayesian model comparison problem. One model, M_1, can be summarized as: galaxies shapes are intrinsically aligned according to the simple non-linear alignment (NLA) model with an unknown amplitude A_{IA} with prior probability $p(A_{IA}) = \mathcal{U}(-3, 3)$. In the other model, M_0, galaxies have no alignment, i.e. $A_{IA} = 0$.

The data for our test are the auto- and cross-spectra (angular power spectra) of the weak gravitational lensing signal in tomographic bins at varying distances (characterized by the redshift of galaxies in each bin).
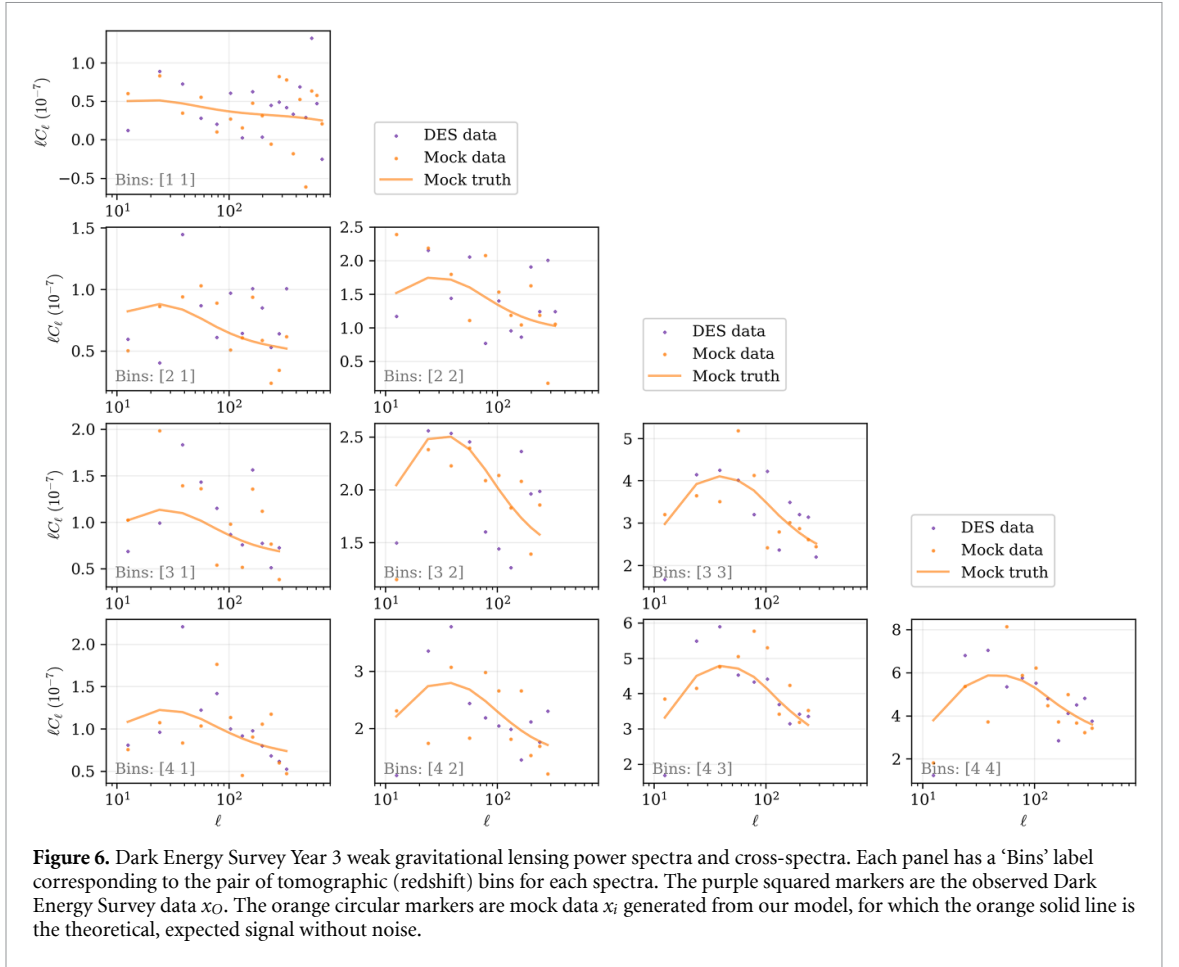
Figure 6. Dark Energy Survey Year 3 weak gravitational lensing power spectra and cross-spectra. Each panel has a 'Bins' label corresponding to the pair of tomographic (redshift) bins for each spectra. The purple squared markers are the observed Dark Energy Survey data x_O. The orange circular markers are mock data x_i generated from our model, for which the orange solid line is the theoretical, expected signal without noise.

We use the same data, with same data cuts of small angular scales, as described in Doux *et al* (2022). The data model is the Λ-cold-dark-matter (ΛCDM) model, with a Gaussian likelihood and priors for cosmological parameters matching (Doux *et al* 2022) and all others unknown systematic parameters fixed to their fiducial values for simplicity.

To generate the training data under the assumed Gaussian likelihood, we simply draw data realizations from $p(x|\theta, \mathcal{M})$, in which the model parameters θ are themselves random draws from the prior $p(\theta|\mathcal{M})$. In practice, this is a simple change in the likelihood and prior code to sample from each probability distribution instead of evaluating each probability distribution.
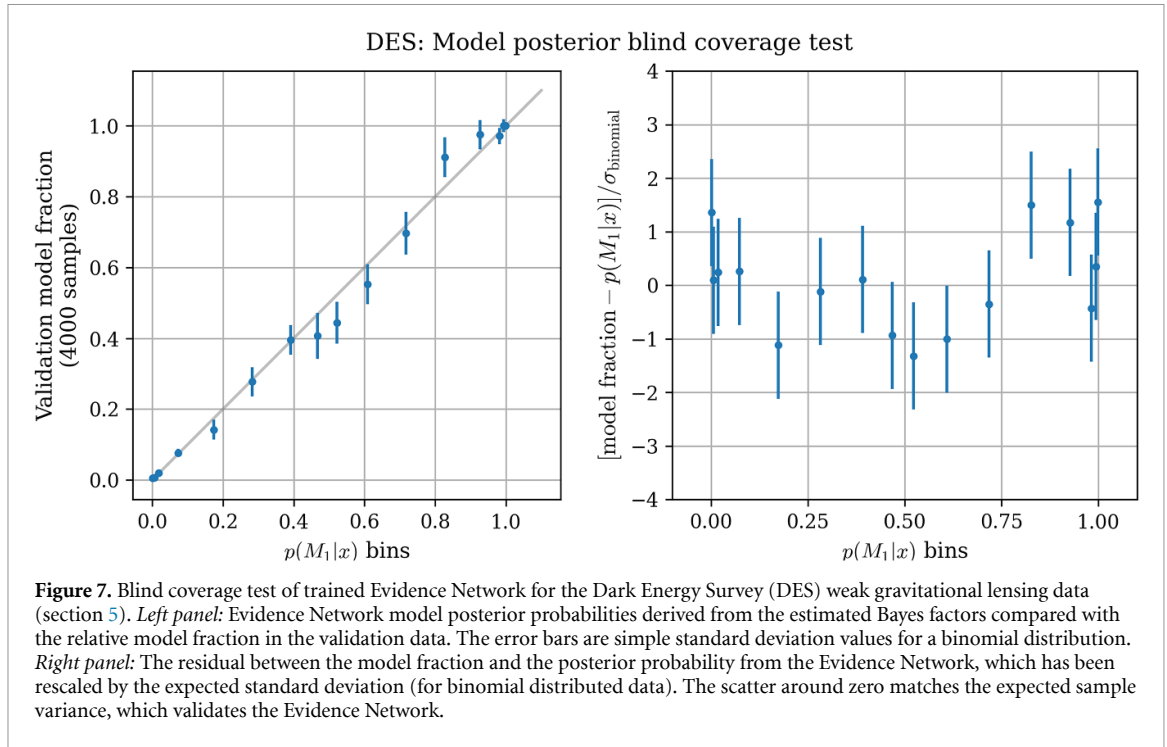
Figure 6 shows the DES data in the form of the angular power spectra, C_ℓ. In addition to the observed data x_O, we show mock generated data (including noise) along with the mock truth (the theoretical expectation). We generate 2×10^4 data samples x_i from each model using a Gaussian likelihood assumption. We train an Evidence Network to estimate $\log K$, using an ensemble of 13 networks with the same architecture as in section 4, but with one layer removed, a longer training time, and a higher initial learning rate (10^{-3}).

We find $\alpha = 1$ gives reliable performance for this example in terms of consistent training. The default choice of $\alpha = 2$ led to over-fitting for some networks of the ensemble, but this is simple to diagnose and solve with a different (typically lower) choice of α. Poor coverage test results can also be used to decide to use a non-standard value of $\alpha \neq 2$. These decisions are described in appendix.

For this problem we do not have the analytic Bayes factor calculation available for each element in our validation data (as we did in section 4), so we take advantage of our blind coverage test (described in section 3). Figure 7 shows this coverage test. The *right panel* shows the residuals between the model fraction and the expected $p(M_1|x)$ derived from the Evidence Network. This has been rescaled by σ_{binomial}, the expected standard deviation for the binomial distribution. The scatter (around zero) of the rescaled residuals is what we would expect given σ_{binomial}.

Evaluating our trained Evidence Network on the observed DES weak gravitational lensing (power spectrum) data, we find:

$$\log_{10} K(x_O) = -0.8 \ (\pm 0.3) \ .$$

Figure 7. Blind coverage test of trained Evidence Network for the Dark Energy Survey (DES) weak gravitational lensing data (section 5). *Left panel:* Evidence Network model posterior probabilities derived from the estimated Bayes factors compared with the relative model fraction in the validation data. The error bars are simple standard deviation values for a binomial distribution. *Right panel:* The residual between the model fraction and the posterior probability from the Evidence Network, which has been rescaled by the expected standard deviation (for binomial distributed data). The scatter around zero matches the expected sample variance, which validates the Evidence Network.

This gives a very mild preference for the model without intrinsic alignments of galaxies with odds of approximately $6/1$. The quoted error is a simple jacknife resampling estimate using the 13 individual estimates from the network ensemble.

This result is not meant as a novel scientific result (not least because the systematic error modelling has been simplified), but it serves to show the simple steps involved in such a Bayesian model comparison with Evidence Networks.

6. Pedagogical example: Rastrigin posterior

We have discussed the advantages of Evidence Networks in cases where the likelihood and/or the prior are not explicitly known but only implicit in the form of simulated data samples with model labels. In this section, we aim to address a separate conceptual point: the possibility of straightforward model comparison in cases where computing the posterior for inference is intractably hard, even when prior and likelihood are explicitly specified up to parameter-independent normalization constants. To illustrate this concept, we present a synthetic data case study involving a highly multimodal posterior density. We compare two models: one model where the components of the parameter vector θ_i, $i = 1, \ldots, n$ have a Rastrigin prior with well-separated modes,

$$p(\theta_i | M_1) \propto e^{-\frac{\theta_i^2}{4} + 10(\cos(2\pi\theta_i) - 1)}, \quad \text{(Rastrigin)} \tag{46}$$
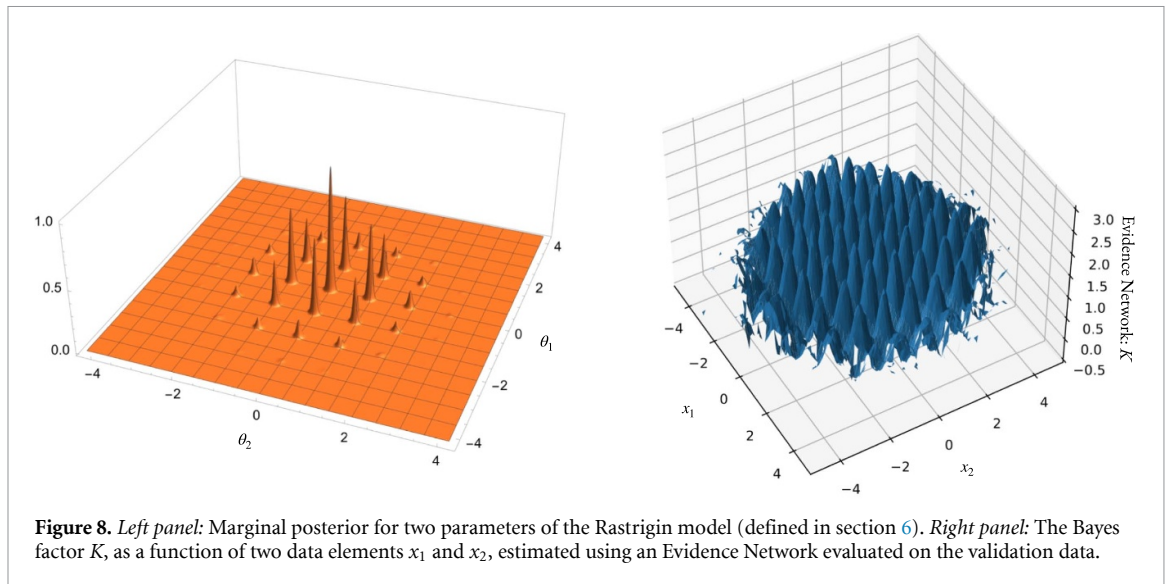
and a second model that omits the oscillatory term in the exponent, leading to the Gaussian prior

$$p(\theta_i | M_0) \propto e^{-\frac{\theta_i^2}{4}}. \quad \text{(Gaussian)} \tag{47}$$

We choose the Rastrigin function as it is commonly used to illustrate the challenges of multimodality for optimization and inference problems, e.g. as in Handley *et al* (2015).

Consider the data model $x = \theta + n$ where n is normal, $n \sim N(0, 1)$. Taking the data $x = 0$ for illustration results in the highly multimodal posterior shown in figure 8 for the $n = 2$ dimensional case. Classical approaches to model comparison such as annealed importance sampling, thermodynamic integration, or nested sampling are methods to integrate the sampled posterior. The Rastrigin model produces a highly multimodal posterior whose number of well-separated modes increases exponentially with the number of parameters n, causing classical methods such as nested sampling to require an exponentially large number of samples in order to succeed. Similarly, approximating $p(x|M_i)$ with a neural density estimator such as a normalizing flow is generally challenging for large n since the absolute normalization depends on the global properties of the distribution.

Figure 8. *Left panel:* Marginal posterior for two parameters of the Rastrigin model (defined in section 6). *Right panel:* The Bayes factor K, as a function of two data elements x_1 and x_2, estimated using an Evidence Network evaluated on the validation data.

This problem is most severe when the data are noisy, since the posterior will be similar to the prior, and hence contain a large number of modes for each parameter dimension. We will show that this exponential complexity in the posterior can in fact be entirely avoided in our approach, since the evidence or Bayes factor make no reference to the parameters.

In fact, focusing on the Bayes factor, the computation becomes trivial in the noisy limit. The easiest way to see this conceptually is that for sufficiently large noise, the probability densities of the data $p(x|M_0)$ and $p(x|M_1)$ become identical, and the Bayes factor will approach unity because the models are indistinguishable. For large noise, the prior is the same as the posterior and the Bayes factor is approximately 1 across all data realizations generated under the two models.

The Evidence Network trivially does the right thing. It simply 'sees' indistinguishable data simulations from both models and will learn to predict 1 regardless of input.

We show a numerical example for a non-trivial case with moderate noise variance set to 1/16 in figure 8. Even in this case the Bayes factor oscillates around 1 with an amplitude of $O(1)$. The Evidence Network learns the local relative probability for any particular data set to be drawn from one model or the other, correctly estimating the Bayes factor for the range of data that is represented in the training set.

The lesson from this example is that there are cases where estimation on the Bayes factor for model comparison (e.g. with Evidence Networks) can be done even in cases where solving the parameter inference problem is intractable (thus prohibiting nested sampling or annealing). Note that we do not claim that Evidence Networks can always compute the Bayes factor trivially[5]. Our claim is that Evidence Networks can provide accurate estimates of the Bayes factor in regimes where other methods (and notably nested sampling) fail due to the high complexity of the posterior.

7. Conclusion and extensions

Computing the the Bayes factor (evidence ratio) for Bayesian model comparison presents a significant challenge in computational statistics. The leading approaches, such as nested sampling, can require the evaluation of intractable integrals over the space of permissible model parameters. This problem becomes increasingly difficult as the dimensionality of the parameter space or the complexity of the posterior increases. We have introduced Evidence Networks as a solution to this problem. We cast the computation as an optimization problem, solved by neural networks trained on data realizations from the model with a designer loss. Couched in this way, the computation makes no reference to the parameters of the underlying model, does not require exploring the posterior density, and can use uncompressed data. The result of the

---

[5] For example, in the large n limit of our current example, the Bayes factor would become a very rapidly varying function of the data and would require large amounts of training data and a highly expressive neural network to converge.

optimization is an amortized neural estimator of the Bayes factor that can be applied to any data set in a single forward pass through the neural network.

In addition to decoupling the model comparison problem from the unnormalized posterior or the dimensionality of the parameter space, our approach is also suited for implicit (likelihood-free or simulation-based) inference, where the explicit forms of either the likelihood and prior distributions are unknown or intractable. All that is required to train our estimator are examples of data from the models that are to be compared. These would generally be computer simulations from the different models, they could be the result of neural generators or be labeled real-world data sets.

In accordance with expectation, we observe in numerical experiments that the estimator performance is robust to the dimension of parameter space. For the training set sizes and example problems we have studied the fractional accuracy of the computed Bayes factor is of order 2 per cent with accurate predictions over many orders of magnitude. Given the subjective nature of interpreting the strength of evidence for model selection, this is sufficient for many applications of model comparison.

We will now discuss a number of ways to use Evidence Networks straightforwardly for applications that go beyond the Bayes factor or model posterior probabilities.

7.1. Simple extensions for further applications

7.1.1. Absolute evidence

We have focused our discussion on model comparison based on (functions of) the evidence ratios. We compute these based on (functions of) the posterior model probabilities, e.g. the log Bayes factor, since these are what our approach returns natively.

Should the absolute evidence be required, for model M_1, say, this can be obtained easily: choose model M_0 to be a reference model where the evidence is analytically tractable (such as a linear Gaussian model), giving $p_{\text{analytic}}(x|M_0)$; we can then use our method to compute the Bayes factor K and then solve for the required evidence as $p(x|M_1) = Kp_{\text{analytic}}(x|M_0)$.

7.1.2. Connection to frequentist hypothesis testing

Numerous writers have advocated the Bayesian formalism as a way to generate statistical procedures with good frequentist properties, regardless of the philosophical differences between the two approaches (see for example Carlin and Louis 2010). In the context of hypothesis testing this has taken the form of deriving practical guidelines for frequentist test thresholds (p-values) that are calibrated on the Bayes factor for certain applications (Johnso 2013). To make contact between Bayes factors and significance thresholds more broadly it would be helpful to be able to study the frequentist properties of the Bayes factor. The computational challenges that motivated our present study have so far prevented this as the Bayes factor for each data sample would be necessary. But Evidence Networks are amortized, meaning it is near-instantaneous to compute accurate Bayes factors for simulated data sets. This permits the use of Bayes factors as a test statistic in hypothesis testing for a broad set of problem classes, including those not covered by existing guidelines.

7.1.3. Using Evidence Networks to do posterior predictive tests

While the interpretation of the Bayes factor is clear, the strong dependence on prior choice can be seen as a weakness. A common alternative is posterior predictive testing (PPT, see Gelman *et al* 2013 for overview). PPT partitions the data x into subsets, x_0 and x_1, say, and evaluates the relative probabilities of two models having predicted x_1 conditional on x_0. This typically involves evaluating $p(\theta|x_0)$ (posterior inference), computing the integral

$$p(x_1|M_i, x_0) = \int p(x_1|\theta, x_0)\, p(\theta|x_0)\ \mathrm{d}\theta \tag{48}$$

for M_0 and M_1, and then computing the posterior odds ratio

$$K_{\text{PPT}} = \frac{p(x_1|M_1, x_0)}{p(x_1|M_0, x_0)}. \tag{49}$$

The difficulty is that all the challenges of computing the evidence ratio with conventional methods, which motivated the present study, also apply to K_{PPT}.

But it is straightforward estimate K_{PPT} using our methodology. The simplest way to see this is to write

$$K_{\text{PPT}} = \frac{p(x_1|M_1, x_0)}{p(x_1|M_0, x_0)} = \frac{p(x_0, x_1|M_1)}{p(x_0, x_1|M_0)}\frac{p(x_0|M_0)}{p(x_0|M_1)} = K(x)/K(x_0), \tag{50}$$

which can be computed simply by training two Evidence Networks, one to compute the Bayes factor for the full data x and one for the subset x_0.

In conclusion, we have presented a novel approach for Bayesian model comparison with a wide range of applications; it overcomes the limitations of even state-of-the-art classical computational methods.

Data availability statement

A code demonstrating Evidence Networks, using the time series example, is available on GitHub: https://github.com/NiallJeffrey/EvidenceNetworksDemo.

Acknowledgment

Appendix. Network architecture & recommendations

The below table summarizes the simple default network architecture used in the demonstrations. The network is a dense network, with a few skip connections, using leaky ReLU (Maas *et al* 2013, Xu *et al* 2015) and Batch Normalization (Ioffe and Szegedy 2015). The default training uses the Adam optimizer (Kingma and Ba 2017) with an initial learning rate of 10^{-4} with an exponential decay rate of 0.95. Unless otherwise stated, this architecture is used for the examples in the main body of the paper.

This architecture was chosen to be simple in order to show the general applicability of Evidence Networks (i.e. they do not require architecture tuning to perform well). For the user we still recommend choosing an architecture best suited for the given data/task. In the typical manner, Network architecture can be tuned, or optimized, to give the lowest validation loss.

To choose a non-standard value of α for the l-POP loss function (i.e. $\alpha \neq 2$), the validation loss cannot be used; varying α effectively changes the interpretation of the loss.

A choice to vary α may be because the default choice leads to: (i) unreliable network training, with over-fitting and high-variation within an ensemble, or (ii) poor coverage test results. In case (i) it is straightforward to vary α (we recommend lowering α) to improve over-fitting and consistency within the ensemble. In both cases though, the coverage test is the gold standard test. As demonstrated with DES data in section 5, a coverage test is successful if the rescaled probability residuals, [model fraction $- p(M_1|x)]/\sigma_{\text{binomial}}$, are consistent with having mean zero and standard deviation of one.

```
Default (simple) architecture                       (width1 = input_shape * 1.1 + 20)
Layer (type)                  Output Shape      Connected to
=========================================================================================
input_1 (InputLayer)          [(, 100)]
-----------------------------------------------------------------------------------------
dense (Dense)                 (,width1)         input_1[0][0]
-----------------------------------------------------------------------------------------
leaky_re_lu (LeakyReLU)       (,width1)         dense[0][0]
-----------------------------------------------------------------------------------------
batch_normalization(BatchNorm)(,width1)         leaky_re_lu[0][0]
-----------------------------------------------------------------------------------------
dense_1 (Dense)               (, 16)            batch_normalization[0][0]
-----------------------------------------------------------------------------------------
leaky_re_lu_1 (LeakyReLU)     (, 16)            dense_1[0][0]
-----------------------------------------------------------------------------------------
batch_normalization_1(BatchNorm)(, 16)          leaky_re_lu_1[0][0]
-----------------------------------------------------------------------------------------
dense_2 (Dense)               (, 16)            batch_normalization_1[0][0]
-----------------------------------------------------------------------------------------
leaky_re_lu_2 (LeakyReLU)     (, 16)            dense_2[0][0]
-----------------------------------------------------------------------------------------
batch_normalization_2(BatchNorm)(, 16)          leaky_re_lu_2[0][0]
-----------------------------------------------------------------------------------------
dense_3 (Dense)               (, 16)            batch_normalization_2[0][0]
-----------------------------------------------------------------------------------------
leaky_re_lu_3 (LeakyReLU)     (, 16)            dense_3[0][0]
-----------------------------------------------------------------------------------------
tf_op_layer_AddV2 (TensorFlowOp [(, 16)]        leaky_re_lu_3[0][0]
                                                batch_normalization_1[0][0]
-----------------------------------------------------------------------------------------
batch_normalization_3(BatchNorm)(, 16)          tf_op_layer_AddV2[0][0]
-----------------------------------------------------------------------------------------
dense_4 (Dense)               (, 16)            batch_normalization_3[0][0]
-----------------------------------------------------------------------------------------
leaky_re_lu_4 (LeakyReLU)     (, 16)            dense_4[0][0]
-----------------------------------------------------------------------------------------
dense_5 (Dense)               (, 1)             leaky_re_lu_4[0][0]
=========================================================================================
```

ORCID iD

Niall Jeffrey ● https://orcid.org/0000-0003-2927-1800

References

Alsing J, Charnock T, Feeney S and Wandelt B 2019 Fast likelihood-free cosmology with neural density estimators and active learning *Mon. Not. R. Astron. Soc.* **488** 4440–58

Alsing J, Wandelt B and Feeney S 2018 Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology *Mon. Not. R. Astron. Soc.* **477** 2874–85

Amon A and Gruen D (DES Collaboration) 2022 Dark energy survey year 3 results: cosmology from cosmic shear and robustness to data calibration *Phys. Rev.* D **105** 023514

Battaglia P W *et al* 2018 Relational inductive biases, deep learning, and graph networks (arXiv:1806.01261)

Brehmer J, Louppe G, Pavez J and Cranmer K 2020 Mining gold from implicit models to improve likelihood-free inference *Proc. Natl Acad. Sci.* **117** 5242–9

Brehmer J, Mishra-Sharma S, Hermans J, Louppe G and Cranmer K 2019 Mining for dark matter substructure: inferring subhalo population properties from strong lenses with machine learning *Astrophys. J.* **886** 49

Carlin B and Louis T 2010 *Bayes and Empirical Bayes Methods for Data Analysis* 2nd edn (Chapman & Hall/CRC Texts in Statistical Science)

Cranmer K, Brehmer J and Louppe G 2020 The frontier of simulation-based inference *Proc. Natl Acad. Sci.* **117** 30055-62

Crenshaw J F, Connolly A and Kalmbach B 2021 PZFlow: normalizing flows for cosmology, with applications to forward modeling galaxy photometry *American Astronomical Society Meeting Abstracts* (*American Astronomical Society Meeting Abstracts*) vol 53 p 230.01

Doux C and Collaboration D E S 2022 Dark energy survey year 3 results: cosmological constraints from the analysis of cosmic shear in harmonic space *Mon. Not. R. Astron. Soc.* **515** 1942–72

Durkan C, Bekasov A, Murray I and Papamakarios G 2019 Neural spline flows (arXiv:1906.04032)

Fenton N, Neil M and Berger D 2016 Bayes and the law *Annu. Rev. Stat. Appl.* **3** 51–77

Feroz F 2013 Calculation and applications of bayesian evidence in astrophysics and particle physics phenomenology *2013 IEEE 13th Int. Conf. on Data Mining Workshops* pp 8–15

Feroz F, Hobson M P and Bridges M 2009 MultiNest: an efficient and robust bayesian inference tool for cosmology and particle physics *Mon. Not. R. Astron. Soc.* **398** 1601–14

Gelman A, Carlin J, Stern H, Dunson D, Vehtari A and Rubin D 2013 *Bayesian Data Analysis* 3rd edn (Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis)

Han C and Carlin B P 2001 Markov chain monte carlo methods for computing bayes factors *J. Am. Stat. Assoc.* **96** 1122–32

Handley W J, Hobson M P and Lasenby A N 2015 Polychord: next-generation nested sampling *Mon. Not. R. Astron. Soc.* **453** 4385–99

Handley W J, Lasenby A N, Peiris H V and Hobson M P 2019 Bayesian inflationary reconstructions from planck 2018 data *Phys. Rev.* D **100** 103511

Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80

Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *Proc. Mach. Learn. Res.* **37** 448–56

Jasa T and Xiang N 2012 Nested sampling applied in bayesian room-acoustics decay analysis *J. Acoust. Soc. Am.* **132** 3251–62

Jaynes E T 2003 *Probability Theory: the Logic of Science* (Cambridge University Press)

Jeffrey N, Alsing J and Lanusse F 2020 Likelihood-free inference with neural compression of DES SV weak lensing map statistics *Mon. Notices Royal Astron. Soc.* **501** 954–69

Jeffrey N and Wandelt B D 2020 Solving high-dimensional parameter inference: marginal posterior densities & moment networks (arXiv:2011.05991)

Jeffrey N *et al* (DES Collaboration) 2021 Dark energy survey year 3 results: curved-sky weak lensing mass map reconstruction *Mon. Not. R. Astron. Soc.* **505** 4626–45

Jeffreys H 1998 *The Theory of Probability* (*Oxford Classic Texts in the Physical Sciences*) (OUP Oxford)

Johnson V E 2013 Revised standards for statistical evidence *Proc. Natl Acad. Sci.* **110** 19313–7

Kass R E and Raftery A E 1995 Bayes factors *J. Am. Stat. Assoc.* **90** 773–95

Keysers C, Gazzola V and Wagenmakers E-J 2020 Using bayes factor hypothesis testing in neuroscience to establish evidence of absence *Nat. Neurosci.* **23** 788–99

Kingma D P and Ba J 2017 Adam: a method for stochastic optimization *3rd Int. Conf. for Learning Representations* (*San Diego*) (arXiv:1412.6980)

Knuth K H, Habeck M, Malakar N K, Mubeen A M and Placek B 2014 Bayesian evidence and model selection (arXiv:1411.3013)

Lemos P, Jeffrey N, Whiteway L, Lahav O, Libeskind N I and Hoffman Y 2020 The sum of the masses of the Milky Way and M31: a likelihood-free inference approach (arXiv:2010.08537)

Maas A L *et al* 2013 Rectifier nonlinearities improve neural network acoustic models *Proc. Icml* (*Atlanta, Georgia, USA*)

Masnadi-Shirazi H and Vasconcelos N 2008 On the design of loss functions for classification: theory, robustness to outliers and savageboost *Advances in Neural Information Processing Systems* p 21

Massimi M 2020 A Philosopher's look at the dark energy survey: reflections on the use of the bayes factor in cosmology *The Dark Energy Survey* (World Scientific)

Niculescu-Mizil A and Caruana R 2005 Obtaining calibrated probabilities from boosting *Proc. of the Twenty-1st Conf. on Uncertainty in Artificial Intelligence (UAI'05)* (AUAI Press) pp 413–20

Papamakarios G and Murray I 2016 Fast ε-free inference of simulation models with bayesian conditional density estimation *Advances in Neural Information Processing Systems* pp 1028–36

Radev S T, D'Alessandro M, Mertens U K, Voss A, Köthe U and Bürkner P-C 2020 Amortized Bayesian model comparison with evidential deep learning (arXiv:2004.10629)

Ramanah D K, Wojtak R, Ansari Z, Gall C and Hjorth J 2020 Dynamical mass inference of galaxy clusters with neural flows *Mon. Notices Royal Astron. Soc.* **499** 1365–2966

Rizvi S, Pettee M and Nachman B 2023 Learning likelihood ratios with neural network classifiers (arXiv:2305.10500)

Secco L and Samuroff S (DES Collaboration) 2022 Dark energy survey year 3 results: cosmology from cosmic shear and robustness to modeling uncertainty *Phys. Rev.* D **105** 023515

Spurio Mancini A, Docherty M M, Price M A and McEwen J D 2022 Bayesian model comparison for simulation-based inference (arXiv:2207.04037)

Taylor P L, Kitching T D, Alsing J, Wandelt B D, Feeney S M and McEwen J D 2019 Cosmic shear: inference from forward models *Phys. Rev.* D **100** 023519

Wakefield J 2009 Bayes factors for genome-wide association studies: comparison with p-values *Genet. Epidemiol.* **33** 79–86

Xu B, Wang N, Chen T and Li M 2015 Empirical evaluation of rectified activations in convolutional network (arXiv:1505.00853)

Yuen K-V 2010 Recent developments of bayesian model class selection and applications in civil engineering *Struct. Saf.* **32** 338–46

Zadrozny B and Elkan C 2002 Transforming classifier scores into accurate multiclass probability estimates *Proc. Eighth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '02* (Association for Computing Machinery) pp 694–9