Amortized Bayesian Model Comparison With Evidential Deep Learning

Stefan T. Radev, Marco D'Alessandro, Ulf K. Mertens, Andreas Voss, Ullrich Köthe, and Paul-Christian Bürkner

Abstract— **Comparing competing mathematical models of complex processes is a shared goal among many branches of science. The Bayesian probabilistic framework offers a principled way to perform model comparison and extract useful metrics for guiding decisions. However, many interesting models are intractable with standard Bayesian methods, as they lack a closed-form likelihood function or the likelihood is computationally too expensive to evaluate. In this work, we propose a novel method for performing Bayesian model comparison using specialized deep learning architectures. Our method is purely simulation-based and circumvents the step of explicitly fitting all alternative models under consideration to each observed dataset. Moreover, it requires no hand-crafted summary statistics of the data and is designed to amortize the cost of simulation over multiple models, datasets, and dataset sizes. This makes the method especially effective in scenarios where model fit needs to be assessed for a large number of datasets, so that case-based inference is practically infeasible. Finally, we propose a novel way to measure epistemic uncertainty in model comparison problems. We demonstrate the utility of our method on toy examples and simulated data from nontrivial models from cognitive science and single-cell neuroscience. We show that our method achieves excellent results in terms of accuracy, calibration, and efficiency across the examples considered in this work. We argue that our framework can enhance and enrich model-based analysis and inference in many fields dealing with computational models of natural processes. We further argue that the proposed measure of epistemic uncertainty provides a unique proxy to quantify absolute evidence even in a framework which assumes that the true data-generating model is within a finite set of candidate models.**

Index Terms— **Bayesian inference, computational and artificial intelligence, machine learning, neural networks, statistical learning.**

Manuscript received October 3, 2020; revised March 2, 2021; accepted October 18, 2021. This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) "Statistical Modeling in Psychology" under Grant GRK 2277, in part by the DFG under Germany's Excellence Strategy - EXC 2075 - 390740016, and in part by the Technology Industries of Finland Centennial Foundation (Artificial Intelligence for Research and Development) under Grant 70007503. *(Corresponding author: Stefan T. Radev.)*

Stefan T. Radev, Ulf K. Mertens, and Andreas Voss are with the Department of Quantitative Research Methods, Heidelberg University, 69117 Heidelberg, Germany (e-mail: stefan.radev93@gmail.com).

Marco D'Alessandro is with the Department of Psychology and Cognitive Science, University of Trento, 38122 Trento, Italy.

Ullrich Köthe is with the Visual Learning Laboratory, IWR, Heidelberg University, 69117 Heidelberg, Germany.

Paul-Christian Bürkner is with the Department of Computer Science, Aalto University, 02150 Espoo, Finland.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3124052.

Digital Object Identifier 10.1109/TNNLS.2021.3124052

I. Introduction

Researchers from various scientific fields face the problem of selecting the most plausible theory for an empirical phenomenon among multiple alternative theories. These theories are often formally stated as mathematical models which describe how observable quantities arise from unobservable (latent) parameters. Focusing on the level of mathematical models, the problem of theory selection then becomes one of *model selection*.

For instance, neuroscientists might be interested in comparing different models of spiking patterns given *in vivo* recordings of neural activity [22]. Epidemiologists, on the other hand, might consider different models for predicting the spread and dynamics of an unfolding infectious disease [54]. Crucially, the preference for one model over alternative models in these examples can have important consequences for research projects or social policies.

Accounting for complex natural phenomena often requires specifying complex models which entail some degree of randomness. Inherent stochasticity, incomplete description, or epistemic ignorance all call for some form of uncertainty awareness. To make matters worse, empirical data on which models are fit are necessarily finite and can only be acquired with finite precision. Finally, the plausibility of many nontrivial models throughout various branches of science can be assessed only approximately, through expensive simulation-based methods [7], [8], [22], [34], [43], [50].

Ideally, a method for approximate model comparison should meet the following desiderata.

1) *Theoretical Guarantee*: Model probability estimates should be, at least in theory, calibrated to the true model probabilities induced by an empirical problem.
2) *Accurate Approximation*: Model probability estimates should be accurate even for finite or small sample sizes.
3) *Occam's Razor*: Preference for simpler models should be expressed by the model probability estimates.
4) *Scalability*: The method should be applicable to complex models with implicit likelihood within reasonable time limits.
5) *Efficiency*: The method should enable fully amortized inference over arbitrarily many models, datasets, and different dataset sizes.
6) *Maximum Data Utilization*: The method should capitalize on all information contained in the data and avoid information loss through insufficient summary statistics of the data.

In this article, we address these desiderata with a novel method for Bayesian model comparison based on evidential deep neural networks. Our method works in a purely

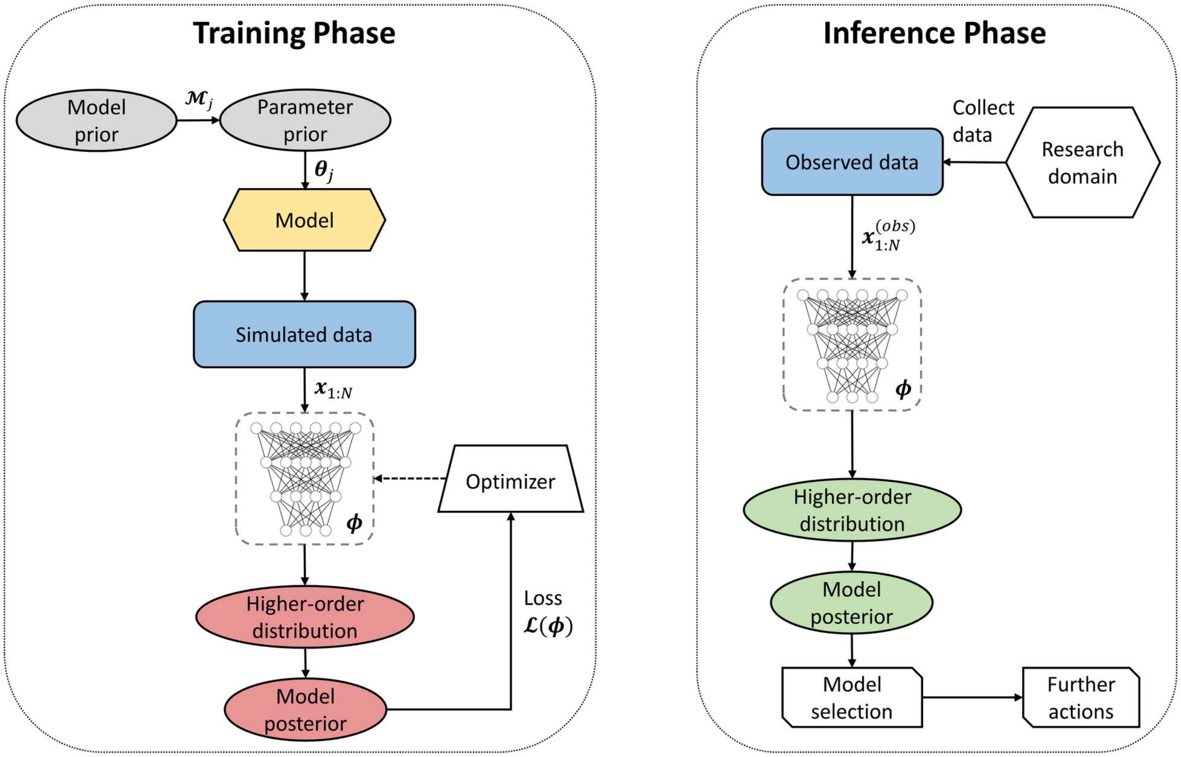


Fig. 1. (Left) Simulation-based training phase of our evidential method. (Right) Inference phase with real data and a pretrained evidential network.

simulation-based manner and circumvents the step of separately fitting all alternative models to each dataset. To this end, for any particular model comparison problem, we propose to train a *specialized expert network* which encodes global information about the generative scope of each model family. In this way, Bayesian model comparison amortizes over multiple models, datasets, and dataset sizes, which makes our method applicable in scenarios where case-based inference is way too costly to perform with standard methods (cf. Fig. 1).

In addition, we propose to avoid hand-crafted summary statistics (a feature on which standard methods for simulation-based inference heavily rely) using novel deep learning architectures which are aligned to the probabilistic structure of the raw data (e.g., permutation invariant networks [2], recurrent networks [13]).

Finally, we explore a novel way to measure epistemic uncertainty in model comparison problems, following the pioneering work of [45] on image classification. We argue that this measure of epistemic uncertainty provides a unique proxy to quantify absolute evidence even in an $\mathcal{M}$-closed framework, which assumes that the true data-generating model is within the candidate set [58].

II. Background

A. Bayesian Inference

A consistent mathematical framework for describing uncertainty and quantifying model plausibility is offered by the Bayesian view on probability theory [23]. In a Bayesian setting, we start with a collection of J competing generative models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_J\}$. Each $\mathcal{M}_j$ is associated with a generative mechanism g_j, typically realized as a Monte Carlo simulation program, and a corresponding parameter space Θ_j. Ideally, each g_j represents a theoretically plausible (potentially noisy) mechanism by which observable quantities $\boldsymbol{x}$ arise from hidden parameters $\boldsymbol{\theta}$ and independent noise $\boldsymbol{\xi}$

$$\boldsymbol{x} = g_j\big(\boldsymbol{\theta}_j, \boldsymbol{\xi}\big) \quad \text{with} \quad \boldsymbol{\theta}_j \in \Theta_j \tag{1}$$

where Θ_j is the corresponding parameter space of model g_j and the subscript j explicates that each model might be specified over a different parameter space. We assume that the functional or algorithmic form of each g_j is known and that we have a sample (dataset) $\{\boldsymbol{x}_i\}_{i=1}^{N} := \boldsymbol{x}_{1:N}$ of N (multivariate) observations $\boldsymbol{x}_n \in \mathcal{X}$ generated from an unknown process p^*. The task of Bayesian model selection is to choose the model in $\mathcal{M}$ that best describes the observed data by balancing simplicity (sparsity) and predictive performance.

B. Likelihood Function

A central object in Bayesian inference is the *likelihood function*, denoted as $p(\boldsymbol{x} \,|\, \boldsymbol{\theta}_j, \mathcal{M}_j)$. Broadly speaking, the likelihood returns the relative probability of an observation $\boldsymbol{x}$ (or a sequence of observations $\boldsymbol{x}_{1:N}$) given a parameter configuration $\boldsymbol{\theta}_j$ and model assumptions $\mathcal{M}_j$. When the parameters are systematically varied and the data held constant, the likelihood can be used to quantify how well each model instantiation fits the observed data.

If the likelihood of a generative model can be associated with a known probability density function (PDF) (e.g., Gaussian), the model can be formulated entirely in terms of the likelihood and the likelihood can be evaluated analytically or numerically for any pair $(\boldsymbol{x}, \boldsymbol{\theta})$. On the other hand, if the likelihood is unknown or intractable, as is the case when dealing with complex models, one can still generate random samples from the model by running the simulation program with a random configuration of its parameters.

This is due to the fact that each stochastic model, viewed as a Monte Carlo simulator, defines an implicit likelihood given by the relationship

$$\boldsymbol{p}\big(\boldsymbol{x} \mid \boldsymbol{\theta}_j, \mathcal{M}_j\big) = \int_{\Xi} \delta\big(\boldsymbol{x} - g_j(\boldsymbol{\theta}_j, \boldsymbol{\xi})\big)\, p\big(\boldsymbol{\xi} \mid \boldsymbol{\theta}_j\big)\, d\boldsymbol{\xi} \tag{2}$$

where $\delta(\cdot)$ is the Dirac delta function and the integral runs over all possible execution paths of the stochastic simulation for a fixed $\boldsymbol{\theta}_j$. For most complex models, this integral is analytically intractable or too expensive to approximate numerically, so it is much easier to specify the model directly in terms of the simulation program g_j instead of deriving the likelihood $p(\boldsymbol{x} \mid \boldsymbol{\theta}_j, \mathcal{M}_j)$. Importantly, we can still *sample* from the likelihood by running the simulator with different Monte Carlo realizations of $\boldsymbol{\xi}$, that is, for a fixed $\boldsymbol{\theta}_j$, we have the following equivalence:

$$\boldsymbol{x}_n \sim p\big(\boldsymbol{x} \mid \boldsymbol{\theta}_j, \mathcal{M}_j\big) \iff \boldsymbol{x}_n = g_j\big(\boldsymbol{\theta}_j, \boldsymbol{\xi}_n\big),\ \boldsymbol{\xi}_n \sim p(\boldsymbol{\xi}). \tag{3}$$

C. Bayes Factors

How does one assign preferences to competing models using a Bayesian toolkit? The canonical measure of evidence for a given model is the *marginal likelihood*

$$p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_j\big) = \int_{\Theta_j} p\big(\boldsymbol{x}_{1:N} \mid \boldsymbol{\theta}_j, \mathcal{M}_j\big)\, p\big(\boldsymbol{\theta}_j \mid \mathcal{M}_j\big)\, d\boldsymbol{\theta}_j \tag{4}$$

which is, in general, intractable to compute for nontrivial models. Importantly, the dependence on the prior over model $\mathcal{M}_j$'s parameters introduces a probabilistic version of Occam's razor, which expresses our preference for a simpler model over a more complex one, given that both models can account for the data equally well. The marginal likelihood thus focuses on *prior predictions* and penalizes the prior complexity of a model (i.e., the prior acts as a weight on the likelihood). This is in contrast to *posterior predictions*, which require marginalization over the parameter posterior $p(\boldsymbol{\theta}_j \mid \boldsymbol{x}_{1:N}, \mathcal{M}_j)$ and can be used to select the model which best predicts new data.

Provided that the marginal likelihood can be efficiently approximated, one can compute the ratio of marginal likelihoods for two models $\mathcal{M}_j$ and $\mathcal{M}_k$ via

$$\mathrm{BF}_{jk} = \frac{p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_j\big)}{p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_k\big)}. \tag{5}$$

This famous ratio is called a Bayes factor (BF) and is used in Bayesian settings for quantifying relative model preference. Thus, a $\mathrm{BF}_{jk} > 1$ indicates preference for model j over model k, given a set of observations $\boldsymbol{x}_{1:N}$. Alternatively, one can directly focus on the (marginal) posterior probability of a model $\mathcal{M}_j$

$$p\big(\mathcal{M}_j \mid \boldsymbol{x}_{1:N}\big) \propto p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_j\big)\, p\big(\mathcal{M}_j\big) \tag{6}$$

which equips the model space itself with a prior distribution $p(\mathcal{M})$ over the considered model space encoding potential preferences for certain models before collecting any data. Such a prior might be useful if a model embodies extraordinary claims (e.g., telekinesis) and thus requires extraordinary evidence supporting it. However, if no prior reasons can be given for favoring some models over others (i.e., one prefers not to prefer), a uniform model prior $p(\mathcal{M}) = 1/J$ can be assumed.

The ratio of posterior model probabilities is called the *posterior odds* and is connected to the BF via the corresponding model priors

$$\frac{p\big(\mathcal{M}_j \mid \boldsymbol{x}_{1:N}\big)}{p\big(\mathcal{M}_k \mid \boldsymbol{x}_{1:N}\big)} = \frac{p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_j\big)}{p\big(\boldsymbol{x}_{1:N} \mid \mathcal{M}_k\big)} \times \frac{p\big(\mathcal{M}_j\big)}{p\big(\mathcal{M}_k\big)}. \tag{7}$$

If two models are equally likely *a priori*, the posterior odds equal the BF. In this case, if the BF, or, equivalently, the posterior odds equal one, the observed data provide no decisive evidence for one of the models over the other. However, a relative evidence of one does not allow to distinguish whether the data are equally likely or equally unlikely under both models, as this is a question of absolute evidence. Needless to say, the distinction between relative and absolute evidence is of paramount importance for model comparison, so we now turn our attention to this distinction.

D. $\mathcal{M}$-Frameworks

In Bayesian inference, the relationship between the true generative process p^* and the model list $\mathcal{M}$ can be classified into three categories: $\mathcal{M}$-closed, $\mathcal{M}$-complete, and $\mathcal{M}$-open [58]. Closely related to the distinction between relative and absolute evidence is the distinction between $\mathcal{M}$-closed and $\mathcal{M}$-complete frameworks. Under an $\mathcal{M}$-closed framework, the true model is assumed to be in the predefined set of competing models $\mathcal{M}$, so relative evidence *is* identical to absolute evidence. Under an $\mathcal{M}$-complete framework, a true model is assumed to exist but is not necessarily assumed to be a member of $\mathcal{M}$. However, one still focuses on the models in $\mathcal{M}$ due to computational or conceptual limitations.[1]

Deciding on the particular $\mathcal{M}$-framework under which a model comparison problem is tackled is often a matter of prior theoretical considerations. However, since in most nontrivial research scenarios $\mathcal{M}$ is a finite set and candidate models in $\mathcal{M}$ are often simpler approximations to the true process, there will be *uncertainty* as to whether the observed data could have been generated by one of these models. In the following, we will refer to this uncertainty as *epistemic uncertainty*.

Our method uses a data-driven way to calibrate its epistemic uncertainty, in addition to model posterior probabilities, through simulations performed within an $\mathcal{M}$-closed framework. Consequently, given real observed data, a researcher can obtain a measure of uncertainty with regard to whether the generative model of the data is likely to be in $\mathcal{M}$ or not. From this perspective, our method lies somewhere between an $\mathcal{M}$-closed and an $\mathcal{M}$-complete framework as it provides information from both viewpoints. In this way, our approach to model misspecification differs from *likelihood-tempering* methods, which require an explicit evaluation of a *tilted* likelihood (raised to a power $0 < t < 1$) to prevent overconfident Bayesian updates [17].

III. Related Work

Bayesian methods for model comparison can be categorized as either posterior predictive or prior predictive

[1] In this work, we delegate the discussion of whether the concept of a true model has any ontological meaning to philosophy. See also [58] for a discussion of an $\mathcal{M}$-open framework, in which no true model is assumed to exist.

approaches [11], with our method falling into the latter category. Posterior predictive approaches are concerned with predicting new data using models trained on the current data. In prior predictive approaches, models are conditioned only on prior information but not on the current data. Accordingly, all current data count as new data for the purpose of prior predictive methods.

Naturally, cross-validation (CV) procedures are the main approach for posterior predictive comparisons [53]. Examples for widely applied methods that fall into this category are approximate CV procedures using Pareto-smoothed importance sampling [5], [52], information criterion approaches such as the widely applicable information criterion (WAIC; [56]), or stacking of posterior predictive distributions [58].

All these methods require the ability not only to evaluate the likelihood of each model for each observation during parameter estimation but also for new observations during prediction. What is more, if application of exact CV methods is required because approximations are insufficient or unavailable, models need to be estimated several times based on different datasets or subsets of the original dataset. This renders such methods practically infeasible when working with complex simulators for which estimating models even once is already very slow. Thus, even a single intractable model in the model set suffices to disproportionately increase the difficultly of performing model comparison.

In contrast, our proposed method circumvents explicit parameter estimation and focuses directly on the efficient approximation of BFs (and posterior model probabilities). Moreover, it overcomes two major sources of intractability that stand in the way of Bayesian model comparison via BFs: the likelihood (3) and the marginal likelihood (4).

When the likelihood can be computed in closed form, sophisticated algorithms for efficiently approximating the (intractable) marginal likelihood have been proposed in the Bayesian universe, such as *bridge sampling* and *path sampling* [12], [16]. However, these methods still depend on the ability to evaluate the likelihood $p(\boldsymbol{x} \,|\, \boldsymbol{\theta}_j, \mathcal{M}_j)$ for each candidate model. If, in addition, the likelihood itself is intractable, as is the case with complex simulators, researchers need to resort to expensive simulation-based methods [34], [37], [49], [50].

A standard set of tools for Bayesian simulation-based inference is offered by approximate Bayesian computation (ABC) methods [35], [47]. ABC methods approximate the model posterior by repeatedly sampling parameters from each proposal (prior) distribution and then simulating multiple datasets by running each simulator with the sampled parameters. A predefined similarity criterion determines whether a simulated dataset (or a summary statistic thereof) is sufficiently similar to the actually observed dataset. The model that most frequently generates synthetic observations matching those in the observed dataset is the one favored by ABC model comparison.

Despite being simple and elegant, standard ABC methods involve a crucial trade-off between accuracy and efficiency. In other words, stricter similarity criteria yield more accurate approximations of the desired posteriors at the price of higher and oftentimes intolerable rejection rates. What is more, most ABC methods require multiple *ad hoc* decisions from the method designer, such as the choice of similarity criterion or the summary statistics of the data (e.g., moments of empirical distributions) [34]. However, there is no guarantee that hand-crafted summaries extract all relevant information and model comparison with insufficient summary statistics can dramatically deteriorate the resulting model posteriors [44]. More scalable developments from the ABC family (ABC-SMC, ABC-MCMC, ABC neural networks, and the recently proposed ABC random forests) offer great efficiency boosts but still rely on hand-crafted summary statistics [24], [34], [46].

Recently, a number of promising innovations from the machine learning and deep learning literature have entered the field of simulation-based inference [6]. For instance, the sequential neural likelihood (SNL, [37]), the automatic posterior transformation (APT, [15]), the amortized ratio estimation [18], or the BayesFlow method [41] all implement powerful neural density estimators to overcome the shortcomings of standard ABC methods. Moreover, these methods involve some degree of *amortization*, which ensures extremely efficient inference after a potentially costly upfront training phase. However, neural density estimation focuses solely on efficient Bayesian parameter estimation instead of scaling up Bayesian model comparison. With certain caveats, neural density estimators can be adapted for Bayesian model comparison by post-processing the samples from an approximate posterior/likelihood over each model's parameters. However, such an approach will involve training a separate neural estimator for each model in the candidate set and has not yet been systematically investigated. In addition, most of these methods also rely on fixed summary statistics [37] and few applications using raw data directly exist [15], [41].

Alongside advancements in simulation-based inference, there has been an upsurge in the development of methods for uncertainty quantification in deep learning applications. For instance, much work has been done on the efficient estimation of Bayesian neural networks [19], [31], [32] since the pioneering work of [33]. Parallel to the establishment of novel variational methods [27], [28], these ideas have paved the way toward more interpretable and trustworthy neural network inference. Moreover, the need for distinguishing between different sources of uncertainty and the overconfidence of deep neural networks in classification and regression tasks has been demonstrated quite effectively [26], [45]. Our current work draws on recent methods for evidence and uncertainty representation in classification tasks [45]. However, our goal is to efficiently approximate BFs between competing mechanistic models using non-Bayesian neural networks, not to estimate neural network parameters (e.g., weights) via Bayesian methods.

Our method combines latest ideas from simulation-based inference and uncertainty quantification for building efficient and uncertainty-aware estimators for amortized Bayesian model comparison. As such, it is intended to complement the toolbox of the simulation-based methods for parameter estimation with crucial model comparison capabilities and incorporates some unique features beyond the scope of the standard ABC methods. In the following, we describe the building blocks of our method.

IV. Evidential Networks for Bayesian Model Comparison

A. Model Comparison as Classification

In line with previous simulation-based approaches to model comparison, we will use the fact that we can generate arbitrary

Algorithm 1 Monte Carlo Generation of Synthetic Datasets for Model Comparison

```
Require: $p(\mathcal{M})$ - prior over models, $\{p(\boldsymbol{\theta}_j \mid \mathcal{M}_j)\}$ - list of priors over model parameters, $\{g_j\}$ - list of stochastic simulators, $\{p_j(\boldsymbol{\xi})\}$ - list of noise distributions (RNGs), $p(N)$ - distribution over dataset sizes, $B$ - number of datasets to generate (batch size)
Draw dataset size: $N \sim p(N)$
for $b = 1, \ldots, B$ do
    Draw model index from model prior: $\mathcal{M}_j^{(b)} \sim p(\mathcal{M})$
    Draw model parameters from prior: $\boldsymbol{\theta}_j^{(b)} \sim p(\boldsymbol{\theta}_j \mid \mathcal{M}_j^{(b)})$
    for $n = 1, \ldots, N$ do
        Sample noise instance: $\boldsymbol{\xi}_n \sim p_j(\boldsymbol{\xi})$
        Run simulator $j$ to obtain $n$th synthetic observation: $\boldsymbol{x}_n = g_j(\boldsymbol{\theta}_j^{(b)}, \boldsymbol{\xi}_n)$
    end for
    Encode model index as a one-hot-encoded vector: $\boldsymbol{m}^{(b)} = \text{OneHotEncode}(\mathcal{M}_j^{(b)})$
    Store pair $(\boldsymbol{m}^{(b)}, \boldsymbol{x}_{1:N}^{(b)})$ in $\mathcal{D}_N^{(B)}$
end for
return mini-batch $\mathcal{D}_N^{(B)} := \{\boldsymbol{m}^{(b)}, \boldsymbol{x}_{1:N}^{(b)})\}_{b=1}^{B}$
```

amounts of data via (3) for each model $\mathcal{M}_j$. Following [34], [40], we cast the problem of model comparison as a probabilistic classification task. In other words, we seek a parametric mapping $f_{\boldsymbol{\phi}} : \mathcal{X}^N \to \Delta^J$ from an arbitrary data space $\mathcal{X}^N$ to a probability simplex Δ^J containing the posterior model probabilities $p(\mathcal{M} \mid \boldsymbol{x}_{1:N})$. Previously, different learning algorithms (e.g., random forests [34]) have been used to tackle model comparison as classification. Following recent developments in algorithmic alignment and probabilistic symmetry [2], [57], our method parameterizes $f_{\boldsymbol{\phi}}$ via a specialized neural network with trainable parameters $\boldsymbol{\phi}$ which is aligned to the probabilistic structure of the observed data (see the Network Architectures section in Appendix A for a detailed description of the used networks' structure).

In addition, our method differs from previous classification approaches to model comparison in the following aspects. First, it requires no hand-crafted summary statistics, since the most informative summary statistics are learned directly from data. Second, it uses online learning (i.e., on-the-fly simulations) and requires no storage of large reference tables or data grids. Third, the addition of new competing models does not require changing the architecture or re-training the network from scratch, since the underlying data domain remains the same. In line with the transfer learning literature, only the last layer of a pretrained network needs to be changed and training can be resumed from where it had stopped. Finally, our method is uncertainty-aware, as it returns a higher order distribution over posterior model probabilities. From this distribution, one can extract both absolute and relative evidences as well as quantify the model selection uncertainty implied by the observed data.

To set up the model classification task, we run Algorithm 1 repeatedly to construct training batches with B simulated datasets of size N and B model indices of the form $\mathcal{D}_N^{(B)} := \{(\boldsymbol{m}^{(b)}, \boldsymbol{x}_{1:N}^{(b)})\}_{b=1}^{B}$. We then feed each batch to a neural network which takes as input simulated data with variable sizes and returns a distribution over posterior model probabilities. The neural network parameters are optimized via standard backpropagation. Upon convergence, we can apply the pretrained network to arbitrarily many datasets of the form $\boldsymbol{x}_{1:N}^{(\text{obs})}$ to obtain a vector of probabilities $p_{\boldsymbol{\phi}}(\boldsymbol{m} \mid \boldsymbol{x}_{1:N}^{(\text{obs})})$ which approximates the true model posterior $p(\mathcal{M} \mid \boldsymbol{x}_{1:N}^{(\text{obs})})$. Note that this procedure incurs no memory overhead, as the training batches need not be stored in memory all at once.

Intuitively, the connection between data and models is encoded in the network's weights. Once trained, the evidential network can be reused to perform instant model comparison on multiple real observations. As mentioned above, the addition of new models involves simply adjusting the pretrained network, which requires much less time than retraining the network from scratch. We now describe how model probabilities and evidence are represented by the evidential network.

B. Evidence Representation

To obtain a measure of absolute evidence by considering a finite number of competing models, we place a Dirichlet distribution over the estimated posterior model probabilities [45]. This corresponds to modeling second-order probabilities in terms of the theory of subjective logic (SL) [25]. These second-order probabilities represent an uncertainty measure over quantities which are themselves probabilities. We use the second-order probabilities to capture epistemic uncertainty about whether the observed data have been generated by one of the candidate models considered during training.

The PDF of a Dirichlet distribution is given by

$$\text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=J}^{J} \pi_j^{\alpha_j - 1} \tag{8}$$

where $\boldsymbol{\pi}$ belongs to the unit $J - 1$ simplex (i.e., $\boldsymbol{\pi} \in \Delta^J := \{\boldsymbol{\pi} \mid \sum_{j=1}^{J} \pi_j = 1\}$ and $B(\boldsymbol{\alpha})$ is the multivariate beta function. The Dirichlet density is parameterized by a vector of *concentration parameters* $\boldsymbol{\alpha} \in \mathbb{R}_+^J$ which can be interpreted as evidences in the ST framework [25]. The sum of the individual evidence components $\alpha_0 = \sum_{j=1}^{J} \alpha_j$ is referred to as the Dirichlet strength, and it affects the precision of the higher order distribution in terms of its variance. Intuitively, the Dirichlet strength governs the *peakedness* of the distribution, with larger values leading to more peaked densities (i.e., most of the density being concentrated in a smaller region of the simplex). We can use the mean of the Dirichlet distribution,

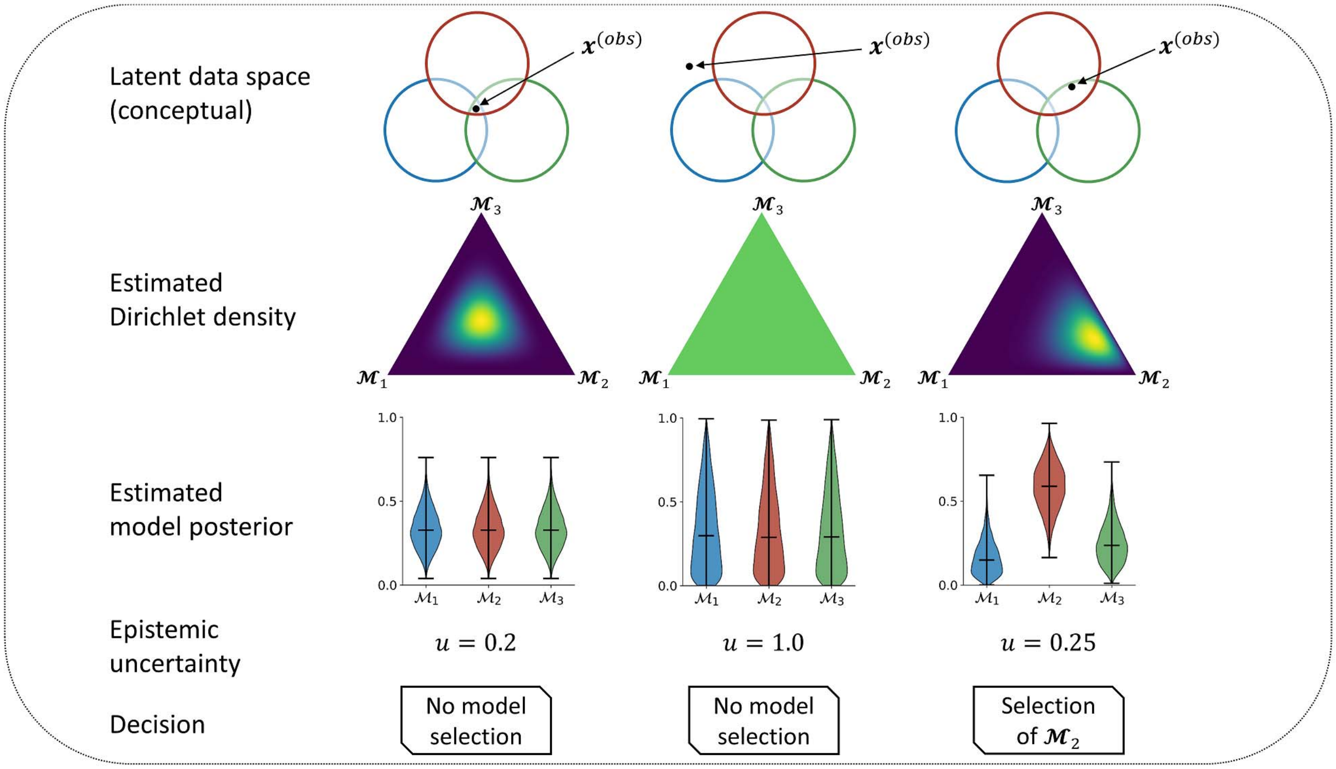


Fig. 2. Three different hypothetical model comparison scenarios with different observations. First column: observing a dataset which is equally probable under all models. In this case, the best candidate model cannot be selected and the Dirichlet density peaks in the middle of the simplex. Second column: dataset which is beyond the generative scope of all models and no model selection decision is possible. The Dirichlet density in this case is flat which indicates total uncertainty. Third column: observed dataset which is most probable under model 2, so the Dirichlet simplex is peaked toward the corner encoding model 2, and the corresponding model posterior for model 2 is highest.

which is a vector of probabilities given by

$$\mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\boldsymbol{\pi}] = \boldsymbol{\alpha}\,\frac{1}{\alpha_0} \tag{9}$$

to approximate the posterior model probabilities $p(\mathcal{M} \mid \boldsymbol{x}_{1:N})$, as will become clearer later in this section. A crucial advantage of such a Dirichlet representation is that it allows to look beyond model probabilities by inspecting the vector of computed evidences. For instance, imagine a scenario with three possible models. If $\boldsymbol{\alpha} = (5, 5, 5)$, the data provide equally strong evidence for all models (Fig. 2, first column)—all models explain the data well. If, on the other hand, $\boldsymbol{\alpha} = (1, 1, 1)$, then the Dirichlet distribution reduces to a uniform on the simplex indicating no evidence for any of the models (Fig. 2, second column)—no model explains the observations well. Note that in either case one cannot select a model on the basis of the data, because posterior model probabilities are equal, yet the interpretation of the two outcomes is very different: The second-order Dirichlet distribution allows one to distinguish between *equally likely* (first case) and *equally unlikely* (second case) models. The last column of Fig. 2 illustrates a scenario with $\boldsymbol{\alpha} = (2, 7, 3)$ in which case one can distinguish between all models (see also Fig. 6 for a scenario with data simulated from an actual complex model).

We can further quantify this distinction by computing an uncertainty score given by

$$u = \frac{J}{\alpha_0} \tag{10}$$

where J is the number of candidate models. Importantly, in our framework, individual concentration parameters (resp. neural network outputs) are lower bounded by 1. Thus, the uncertainty score ranges between 0 (total certainty) and 1 (total uncertainty) and has a straightforward interpretation. Accordingly, the total uncertainty is given when $\alpha_0 = J$, which would mean that the data provide no evidence for any of the J candidate models. On the other hand, $u \ll 1$ implies a large Dirichlet strength $\alpha_0 \gg J$, which would read that the data provide plenty of evidence for one or more models in question. The uncertainty score corresponds to the concept of *vacuity* (i.e., epistemic uncertainty) in the terminology of SL [25]. We argue that epistemic uncertainty should be a crucial aspect in model comparison and selection, as it quantifies the strength of evidence and, consequently, the strength of the theoretical conclusions we can draw given the observed data.

Consequently, model comparison in our framework consists in inferring the parameters of a Dirichlet distribution given an observed dataset. The problem of inferring posterior model probabilities can then be formulated as

$$p(\mathcal{M} \mid \boldsymbol{x}_{1:N}) \approx p_{\boldsymbol{\phi}}(\boldsymbol{m} \mid \boldsymbol{x}_{1:N}) = \mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}\left(f_{\boldsymbol{\phi}}(\boldsymbol{x}_{1:N})\right)}[\boldsymbol{\pi}] \tag{11}$$

where $f_{\boldsymbol{\phi}}$ is a neural network with positive outputs greater than one, $f_{\boldsymbol{\phi}} : \mathcal{X}^N \rightarrow [1, \infty]^J$. Additionally, we can also obtain a measure of absolute model evidence by considering the uncertainty encoded by the full Dirichlet distribution (10).

C. Learning Evidence in an $\mathcal{M}$-Closed Framework

How do we ensure that the outputs of the neural network match the true unknown model posterior probabilities? Consider, for illustrational purposes, a dataset with a single observation, that is, $N = 1$ such that $\boldsymbol{x}_{1:N} = \boldsymbol{x}$. As per Algorithm 1, we have unlimited access to training samples from $p(\mathcal{M}, \boldsymbol{x}) = \int p(\mathcal{M}) p(\boldsymbol{\theta} \,|\, \mathcal{M}) p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \mathcal{M}) d\,\boldsymbol{\theta}$. We use the mean of the Dirichlet distribution $p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x})$ parameterized by an evidential neural network with parameters $\boldsymbol{\phi}$ to approximate $p(\mathcal{M} \,|\, \boldsymbol{x})$. To optimize the parameters of the neural network, we can minimize some loss $\mathcal{L}$ in expectation over all possible datasets

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\, \mathbb{E}_{(\boldsymbol{m},\boldsymbol{x}) \sim p(\mathcal{M},\boldsymbol{x})}\big[\mathcal{L}\big(p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x}), \boldsymbol{m}\big)\big] \tag{12}$$

where $\boldsymbol{m}$ is a one-hot encoded vector of the true model index $\mathcal{M}_j$. We also require that $\mathcal{L}$ be a *strictly proper loss* [14]. A loss function is strictly proper if and only if it attains its minimum when $p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x}) = p(\mathcal{M} \,|\, \boldsymbol{x})$ [14]. When we choose the Shannon entropy $\mathbb{H}(p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x})) = -\sum_j p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x})_j \log p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x})_j$ for $\mathcal{L}$, we obtain the strictly proper logarithmic loss

$$\begin{aligned} \mathcal{L}\big(p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x}), \boldsymbol{m}\big) &= -\sum_{j=1}^{J} m_j \log p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x})_j && (13)\\ &= -\sum_{j=1}^{J} m_j \log\left(\frac{f_{\boldsymbol{\phi}}(\boldsymbol{x})_j}{\sum_{j'=1}^{J} f_{\boldsymbol{\phi}}(\boldsymbol{x})_{j'}}\right) && (14) \end{aligned}$$

where $m_j = 1$ when j is the true model index and 0 otherwise. Thus, to estimate $\boldsymbol{\phi}$, we can minimize the expected logarithmic loss over all simulated datasets where $f_{\boldsymbol{\phi}}(\boldsymbol{x})_j$ denotes the j-th component of the Dirichlet density given by the evidential neural network. Since we use a strictly proper loss, the evidential network yields the true model posterior probabilities over all possible datasets when perfectly converged.

Intuitively, the logarithmic loss encourages high evidence for the true model and low evidences for the alternative models. Correspondingly, if a dataset with certain characteristics can be generated by different models, evidence for these models will jointly increase. Additionally, the model which generates these characteristics most frequently will accumulate the most evidence and thus be preferred. However, we also want low evidence, or, equivalently, high epistemic uncertainty, for datasets which are implausible under all models. We address this problem in the next section.

D. Learning Absolute Evidence Through Regularization

We now propose a way to address the scenario in which no model explains the observed data well. In this case, we want the evidential network to estimate low evidence for all models in the candidate set. To attenuate evidence for datasets which are implausible under all models considered, we incorporate a Kullback–Leibler (KL) divergence into the criterion in Eq.13. We compute the KL divergence between the Dirichlet density generated by the neural network and a uniform Dirichlet density implying total uncertainty. Thus, the KL shrinks evidences which do not contribute to correct model assignments during training, so an implausible dataset at inference time will lead to low evidence under all models. This type of regularization has been used for capturing out-of-distribution (OOD) uncertainty in image classification [45]. Accordingly, our modified optimization criterion becomes

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\, \mathbb{E}_{(\boldsymbol{m},\boldsymbol{x}) \sim p(\mathcal{M},\boldsymbol{x})}\big[\mathcal{L}\big(p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x}), \boldsymbol{m}\big) + \lambda \Omega(\tilde{\boldsymbol{\alpha}})\big] \tag{15}$$

with $\Omega(\tilde{\boldsymbol{\alpha}}) = \mathbb{KL}[\mathrm{Dir}(\tilde{\boldsymbol{\alpha}}) \,||\, \mathrm{Dir}(\mathbf{1})]$. The term $\tilde{\boldsymbol{\alpha}} = \boldsymbol{m} + (1 - \boldsymbol{m}) \odot \boldsymbol{\alpha}$ represents the estimated evidence vector after removing the evidence for the true model. This is possible, because we know the true model during simulation-based training. For application on real datasets after training, knowing the ground truth is not required anymore as $\boldsymbol{\phi}$ has already been obtained. The KL regularizer penalizes evidences for the false models and drives these evidences toward unity. Equivalently, KL acts as a *ground-truth preserving prior* on the higher order Dirichlet distribution which preserves evidence for the true model and attenuates misleading evidences for the false models. The hyperparameter λ controls the weight of regularization and encodes the tolerance of the algorithm to accept implausible (OOD) datasets during inference. With large values of λ, it becomes possible to detect cases where all models are deficient; with $\lambda = 0$, only relative evidence is generated. Note that in the latter case, we recover our original proper criterion without penalization. The KL weight λ can be selected through prior empirical considerations on how well the simulations cover the plausible set of real-world datasets.

Importantly, the introduction of the KL regularizer renders the loss no longer *strictly proper*. Therefore, a large regularization weight λ would lead to poorer calibration of the approximate model posteriors, as the regularized loss is no longer minimized by the true model posterior. However, since the KL prior is ground-truth preserving, the accuracy of recovering the true model should not be affected. Indeed, we observe this behavior throughout our experiments.

To make optimization tractable, we use the fact that we can easily simulate batches of the form $\mathcal{D}_N^{(B)} = \{(\boldsymbol{m}^{(b)}, \boldsymbol{x}_{1:N}^{(b)})\}_{b=1}^{B}$ via Algorithm 1 and approximate (15) via standard backpropagation by minimizing the following loss:

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\phi})\\ &= \frac{1}{B}\sum_{b=1}^{B}\left[-\sum_{j=1}^{J} m_j^{(b)} \log\left(\frac{f_{\boldsymbol{\phi}}\Big(\boldsymbol{x}_{1:N}^{(b)}\Big)_j}{\sum_{j'=1}^{J} f_{\boldsymbol{\phi}}\Big(\boldsymbol{x}_{1:N}^{(b)}\Big)_{j'}}\right) + \lambda\Omega\big(\tilde{\boldsymbol{\alpha}}^{(b)}\big)\right] \end{aligned} \tag{16}$$

over multiple batches to converge at a Monte Carlo estimator $\widehat{\boldsymbol{\phi}}$ of $\boldsymbol{\phi}^*$. In practice, convergence can be determined as the point at which the loss stops decreasing, a criterion similar to *early stopping*. Alternatively, the network can be trained for a predefined number of epochs. Note that at least in principle, the network can be trained arbitrarily long, since we assume that we can access the joint distribution $p(\mathcal{M}, \boldsymbol{x}, N)$ through simulation [cf. Fig. 1, (left)].

E. Implicit Preference for Simpler Models

Remembering that $p_{\boldsymbol{\phi}}(\boldsymbol{m} \,|\, \boldsymbol{x}_{1:N}) \propto p(\boldsymbol{x}_{1:N} \,|\, \mathcal{M}) p(\mathcal{M})$, we note that perfect convergence implies that preference for simpler models (Bayesian Occam's razor) is automatically encoded by our method. This is due to the fact that we are approximating an expectation over all possible datasets, parameters, and models. Accordingly, datasets generated by a simpler model tend to be more similar compared with

Algorithm 2 Training Phase and Inference Phase for Amortized Bayesian Model Comparison

```
Require: $f_{\boldsymbol{\phi}}$ - evidential neural network, $\{\boldsymbol{x}^{(\mathrm{obs})}_{1:N_i}\}_{i=1}^{I}$ - list of $I$ observed datasets for inference, $\lambda$ - regularization weight, $B$ - number of simulations at each iteration (batch size)
1:  Training phase:
2:  repeat
3:      Generate a training batch $\mathcal{D}_N^{(B)} = \{(\boldsymbol{m}^{(b)}, \boldsymbol{x}^{(b)}_{1:N})\}_{b=1}^{B}$ via Algorithm 1
4:      Compute evidences for each simulated dataset in $\mathcal{D}_N^{(B)}$: $\boldsymbol{\alpha}^{(b)} = f_{\boldsymbol{\phi}}(\boldsymbol{x}^{(b)}_{1:N})$
5:      Compute loss according to Eq.16
6:      Update neural network parameters $\boldsymbol{\phi}$ via backpropagation
7:  until convergence to $\widehat{\boldsymbol{\phi}}$
8:  Amortized inference phase:
9:  for $i = 1, \ldots, I$ do
10:     Compute model evidences $\boldsymbol{\alpha}_i^{(\mathrm{obs})} = f_{\widehat{\boldsymbol{\phi}}}(\boldsymbol{x}^{(\mathrm{obs})}_{1:N_i})$
11:     Compute epistemic uncertainty $u_i = J / \sum_{j=1}^{J} \alpha^{(\mathrm{obs})}_{i,j}$
12:     Approximate true model posterior probabilities $p(\mathcal{M} \mid \boldsymbol{x}^{(\mathrm{obs})}_{1:N_i})$ via $p_{\boldsymbol{\phi}}(\boldsymbol{m} \mid \boldsymbol{x}_{1:N_i}) = \boldsymbol{\alpha}_i^{(\mathrm{obs})} / \sum_{j=1}^{J} \alpha^{(\mathrm{obs})}_{i,j}$
13: end for
14: Choose further actions
```

those from a more complex competitor. Therefore, during training, certain datasets which are plausible under both models will be generated more often by the simpler model than by the complex model. Thus, a perfectly converged evidential network will capture this behavior by assigning higher posterior probability to the simpler model (assuming equal prior probabilities). Therefore, at least in theory, our method captures complexity differences arising purely from the generative behavior of the models and does not presuppose an *ad hoc* measure of complexity (e.g., number of parameters).

F. Putting It All Together

The essential steps of our evidential method are summarized in Algorithm 2. Note that steps 2–7 and 9–13 can be executed in parallel with GPU support to dramatically accelerate convergence and inference. In sum, we propose to cast the problem of model comparison as evidence estimation and learn a Dirichlet distribution over posterior model probabilities directly via simulations from the competing models. To this end, we train an evidential neural network which approximates posterior model probabilities and further quantifies the epistemic uncertainty as to whether an observed dataset is within the generative scope of the candidate models. Moreover, once trained on simulations from a set of models, the network can be reused and extended to new models across a research domain, essentially *amortizing* the model comparison process. Accordingly, if the priors over model parameter do not change, multiple researchers can reuse the same network for multiple applications. If the priors over model parameters change or additional models need to be considered, the parameters of a pretrained network can be adjusted or the network augmented with additional output nodes for the new models.

V. Experiments

In this section, we demonstrate the utility of our method on a toy example and relevant models from chemistry, cognitive science, and neurobiology. A further toy example with 400 models and details for neural network training, architectures, performance metrics, and forward models are to be found in the Appendix.

A. Experiment 1: Beta-Binomial Model With Known Analytical Marginal Likelihood

As a basic proof of concept for our evidential method, we focus on a toy model comparison scenario with an analytically tractable marginal likelihood. Thereby, we pursue a couple of goals. First, we want to demonstrate that the estimated posterior probabilities closely approximate the analytic model posteriors. To show this, we compare the analytically computed with the estimated BFs. In addition, we want to show that the accuracy of recovery matches closely the accuracy obtained by the analytic BFs across all N. For this, we consider the simple beta-binomial model given by

$$\theta \sim \mathrm{Beta}(\alpha, \beta) \tag{17}$$

$$x_n \sim \mathrm{Bernoulli}(\theta) \quad \text{for} \quad n = 1, \ldots, N. \tag{18}$$

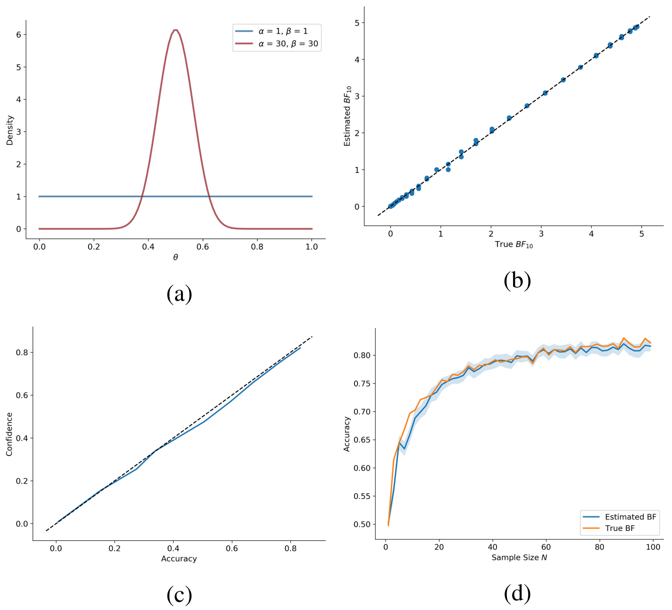
The analytical marginal likelihood of the beta-binomial model is

$$p(x_{1:N}) = \binom{N}{K} \frac{\mathrm{Beta}(\alpha + K, \beta + N - K)}{\mathrm{Beta}(\alpha, \beta)} \tag{19}$$

where K denotes the number of successes in the N trials. For this example, we will consider a model comparison scenario with two models, one with a flat prior $\mathrm{Beta}(1, 1)$ on the parameter θ and another with a sharp prior $\mathrm{Beta}(30, 30)$. The two prior densities are depicted in Fig. 3(a).

We train a small permutation invariant evidential network with batches of size $B = 64$ until convergence. For each batch, we draw the samples size from a discrete uniform distribution $N \sim \mathcal{U}_D(1, 100)$ and input the raw binary data to the network. We validate the network on 5000 separate validation datasets for each N. Convergence took approximately 15 min, whereas inference on all 5000 validation datasets took less than 2 s.

Our results demonstrate that the estimated BFs closely approximate the analytic BFs [Fig. 3(b)]. We also observe no systematic under- or overconfidence in the estimated BFs, which is indicated by the calibration curve resembling a straight line [Fig. 3(c)]. Finally, the accuracy of recovery achieved with the estimated BFs closely matches that of the analytic BFs across all sample sizes N [Fig. 3(d)].



(a) (b) (c) (d)

Fig. 3. (a) Prior densities of the θ parameter for both models of Experiment 1. (b) True versus estimated BFs obtained from the network-induced Dirichlet distribution at $N = 100$. (c) Calibration curve at $N = 100$ indicating very good calibration (dotted line represents perfect calibration). (d) Accuracy at all N achieved with both the analytic and estimated BFs (the shaded region represents a 95% bootstrap confidence interval around the accuracies of the evidential network). (a) Models' priors. (b) True versus estimated BFs at $N = 100$. (c) Calibration curve at $N = 100$. (d) Accuracy across all N.

B. Experiment 2: Markov Jump Process of Stochastic Chemical Reaction Kinetics

In this experiment, we apply our evidential method to a simple model of nonexchangeable chemical molecule concentration time series data. Furthermore, we demonstrate the efficiency benefits of our amortized learning method compared with a nonamortized ABC-SMC algorithm. We define two Markov jump process models $\mathcal{M}_1$ and $\mathcal{M}_2$ for conversion of (chemical) species z into species y

$$\mathcal{M}_1 : z + y \overset{\theta_1}{\rightarrow} 2y \tag{20}$$

$$\mathcal{M}_2 : z \overset{\theta_2}{\rightarrow} y. \tag{21}$$

Each model has a single rate parameter θ_i. We use a Gillespie simulator to generate simulated time series from the two models with an upper time limit of 0.1 s. Both models start with initial concentrations $x_0 = 40$ and $y_0 = 3$, so they only differ in terms of their reaction kinetics. The input time series $\boldsymbol{x}_{1:N}$ consist of a time vector $t_{1:N}$ and two vectors of molecule concentrations for each species at each time step, $z_{1:N}$ and $y_{1:N}$, which we stack together. We place a wide uniform prior over each rate parameter: $\theta_i \sim \mathcal{U}(0, 100)$.

We train an evidential sequence network for 50 epochs of 1000 minibatch updates and validate its performance on 500 previously unobserved time series. Wall clock training time was approximately 52.3 min. In contrast, the wall clock inference time on the 500 validation time series was 254 ms, leading to dramatic gains due to amortization. The bootstrap accuracy of recovery was 0.98 ($SD = 0.01$) over the entire validation set.

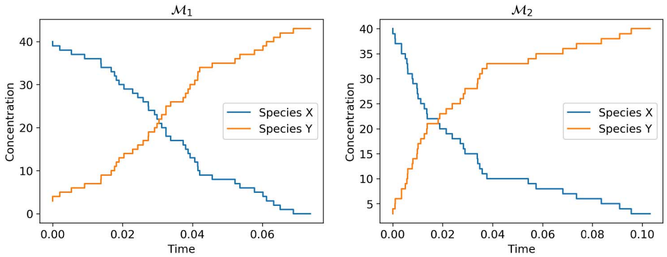


Fig. 4. Observed concentration time series from both Markov jump models of Experiment 2 with $\theta = 2.0$.

We also apply the ABC-SMC algorithm available from the *pyABC* [29] library to a single dataset $\boldsymbol{x}_{1:N}^{(\text{obs})}$ generated from model 1 ($\mathcal{M}_1$) with rate parameter $\theta_1 = 2.0$. Fig. 4 depicts time series generated from model 1 (left) and time series generated from model 2 with $\theta_2 = 2.0$. Notably, the generative differences implied by the two models are subtle and not straightforward to explicitly quantify.

For the ABC-SMC method, we set the minimum rejection threshold ϵ to 0.7 and the maximum number of populations to 15, as these settings lead to perfect recovery of the true model. As a distance function, we use the L_2 norm between the raw concentration time series of species z, evaluated at 20 time points.[2]

The convergence of the ABC-SMC algorithm on the single dataset took 12.2-min wall clock time. Thus, inference on the 500 validation datasets would have taken more than four days to complete. Accordingly, we see that the training effort with our method is worthwhile even after as few as five datasets. As for recovery on the single test dataset, ABC-SMC selects the true model with a probability of 1, whereas our evidential network outputs a probability of 0.997 which results in a negligible difference of 0.003 between the results from two methods.

C. Experiment 3: Stochastic Models of Decision Making

In this experiment, we apply our evidential method to compare several nontrivial nested stochastic *evidence accumulator models* (EAMs) from the field of human decision-making [42], [51]. With this experiment, we want to demonstrate the performance of our method in terms of accuracy and posterior calibration on exchangeable data obtained from complex cognitive models. Additionally, we want to demonstrate how our regularization scheme can be used to capture absolute evidence by artificially rendering the data implausible under all models.

1) Model Comparison Setting: EAMs describe the dynamics of decision-making via different neurocognitively plausible parameters (i.e., speed of information processing, decision threshold, bias/pre-activation, etc.). EAMs are most often applied to choice reaction times (RTs) data to infer neurocognitive processes underlying generation of RT distributions in cognitive tasks. The most general form of an EAM is given by a stochastic differential equation

$$dx = v dt + c d\xi \tag{22}$$

where dx denotes a change in activation of an accumulator, v denotes the average speed of information accumulation

[2] These settings were picked from the original pyABC documentation available at https://pyabc.readthedocs.io/en/latest/examples/chemical_reaction.html

(often termed the drift rate), and $d\xi$ represents a stochastic additive component with $d\xi \sim \mathcal{N}(0, c^2)$.

Multiple flavors of the above-stated basic EAM form exist throughout the literature [3], [42], [50], [51]. Moreover, most EAMs are intractable with standard Bayesian methods [3], so model selection is usually hard and computationally cumbersome. With this example, we pursue several goals. First, we want to demonstrate the utility of our method for performing model selection on multiple nested models. Second, we want to empirically show that our method implements Occam's razor. Third, we want to show that our method can indeed provide a proxy for absolute evidence.

To this end, we start with a very basic EAM defined by four parameters $\boldsymbol{\theta} = (v_1, v_2, a, t_0)$ with v_i denoting the speed of information processing (drift rate) for two simulated RT experimental tasks $i \in \{1, 2\}$, a denoting the decision threshold, and t_0 denoting an additive constant representing the time required for nondecisional processes such as motor reactions. We then define five more models with increasing complexity by successively *freeing* the parameters z_r (bias), α (heavy-taildness of noise distribution), s_{t_0} (variability of nondecision time), s_v (varibaility of drift-rate), and s_{zr} (variability of bias). Note that the inclusion of non-Gaussian diffusion noise renders an EAM model intractable, since in this case no closed-form likelihood is available (see [55] for more details). Table S1 in Appendix E lists the priors over model parameters and fixed parameter values.

The task of model selection is thus to choose among six nested EAM models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5, \mathcal{M}_6\}$, each able of capturing increasingly complex behavioral patterns. Each model j is able to account for all datasets generated by the previous models $i < j$, since the previous models are nested within the jth model. For instance, model $\mathcal{M}_6$ can generate all datasets possible under the other models at the cost of increased functional and parametric complexity. Therefore, we need to show that our method encodes Occam's razor purely through the generative behavior of the models.

To show that our regularization method can be used as a proxy to capture absolute evidence, we perform the following experiment. We define a temporal shifting constant $K \in (0, 10)$ (in units of seconds) and apply the shift to each response time in each validation dataset. Therefore, as K increases, each dataset becomes increasingly implausible under all models considered. For each K, we compute the average uncertainty over all shifted validation datasets and plot as a function of K. Here, we only consider the maximum number of trials $N = 300$.

We train three evidential neural networks with different KL weights: $\lambda \in \{0.0, 0.1, 1.0\}$ to investigate the effects of λ on accuracy, calibration, and uncertainty. All networks were trained with variable number of trials $N \sim \mathcal{U}_D(1, 300)$ per batch for a total of 50 000 iterations. The training of each network took approximately half a day on a single computer. In contrast, performing inference on 5000 datasets with a pre-trained network took less than 10 s.

2) Validation Results: To quantify the global performance of our method, we compute the accuracy of recovery as a function of the number of observations (N) for each of the models. We also compute the epistemic uncertainty as a function of N. To this end, we generate 500 new datasets for each N and compute the accuracy of recovery and average uncertainty. These results are depicted in Fig. 5.

a) Accuracy: We observe that the accuracy of recovery increases with increasing sample size and begins to flatten out around $N = 100$, independently of the regularization weight λ [Fig. 5(a)]. This behavior is desirable, as selecting the true model should become easier when more information is available. Furthermore, since the models are nested, perfect recovery is not possible, as the models exhibit a large shared data space.

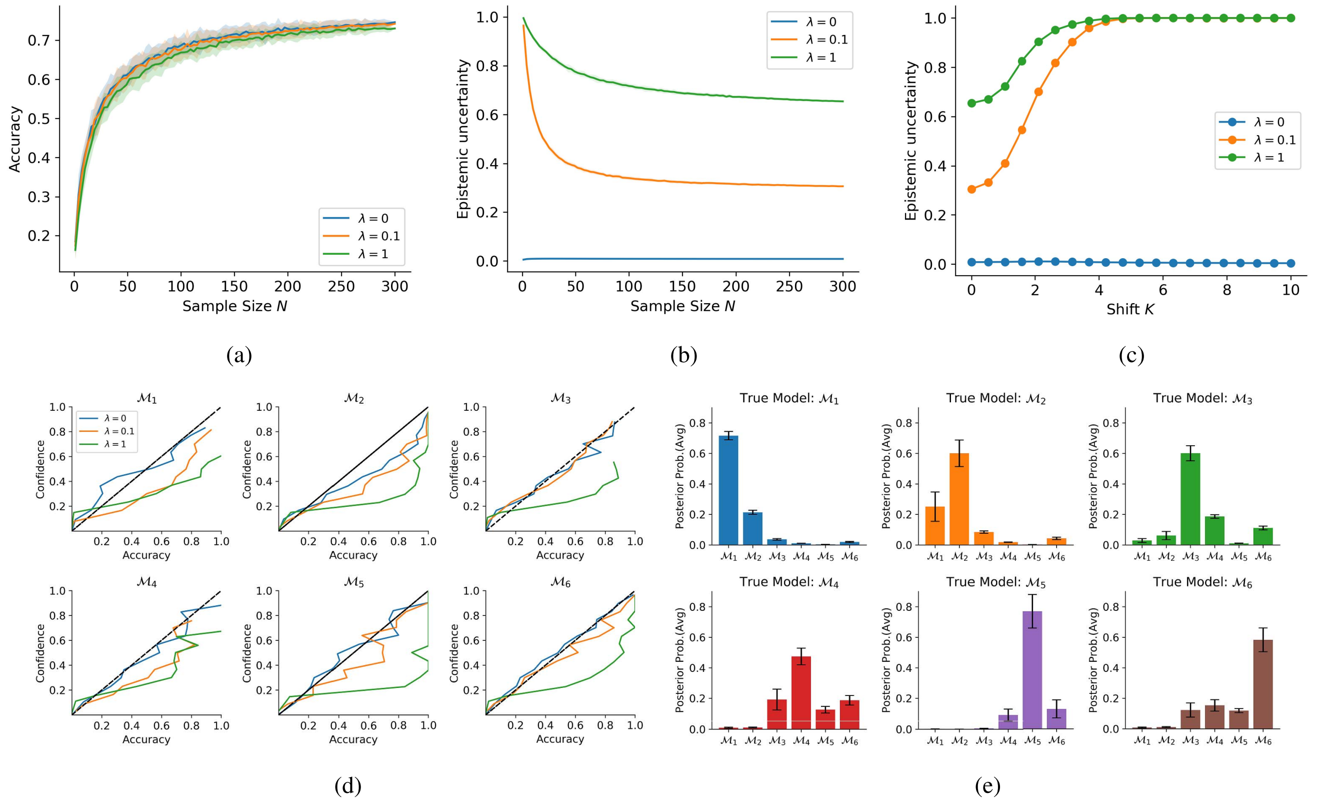
b) Calibration: Fig. 5(d) depicts the calibration curves for each model and each regularization value. The unregularized network appears to be very well-calibrated, whereas the regularized networks become increasingly underconfident with increasing regularization weight. This is due to the fact that the regularized networks are encouraged to generate zero evidence for the wrong models, so model probabilities become miscalibrated. Importantly, none of the networks shows overconfidence.

c) Occam's razor: We also test Occam's razor by generating 500 datasets from each model with $N = 300$ and compute the average predicted model posterior probabilities by the unregularized network. Thus, all datasets generated by model j are plausible under the remaining models $\mathcal{M}_i, i > j$. These average model probabilities are depicted in Fig. 5(e). Even though the datasets generated by the nested simpler models are plausible under the more complex models, we observe that Occam's razor is encoded by the behavior of the network, which, on average, consistently selects the simpler model when it is the true data-generating model. We also observe that this behavior is independent of regularization [the results for $\lambda = 0.1$ and $\lambda = 1$ are not depicted in Fig. 5(e)].

d) Epistemic uncertainty and absolute evidence: Epistemic uncertainty over different trial numbers (N) is zero when no KL regularization is applied ($\lambda = 0$). On the other hand, both small ($\lambda = 0.1$) or large ($\lambda = 1.0$) regularization weights lead to nonzero uncertainty over all possible N [Fig. 5(b)]. This pattern reflects a reduction in epistemic uncertainty with increasing amount of information and mirrors the inverse of the recovery curve. Note that the value at which epistemic uncertainty begins to flatten out is larger for the highly regularized model, as it encodes more cautiousness with respect to the challenging task of selecting a true nested model. Finally, the results on shifted datasets are depicted in Fig. 5(c). Indeed, we observe that the regularized networks are able to detect implausible datasets and output total uncertainty around $K > 4$ for all manipulated datasets. Uncertainty increases faster for high regularization. On the other hand, the unregularized model does not have any way of signaling impossibility of a decision, so its uncertainty remains at 0 over all K.

D. Experiment 4: Stochastic Models of Single-Neuron Activity

In this experiment, we apply our evidential method to complex nested spiking neuron models describing the properties of biological cells in the nervous system. The purpose of this experiment is threefold. First, we want to assess the ability of our method to classify models deploying a variety of spiking patterns which might account for different cortical and sub-cortical neuronal activities. Second, we want to challenge the network's ability to detect biologically implausible data patterns as accounted by epistemic uncertainty. Finally, we compare our method with other viable neural network



(a) (b) (c)

(d) (e)

Fig. 5. Detailed validation results from Experiment 3. (a) Accuracy over all N. (b) Epistemic uncertainty over all N. (c) Epistemic uncertainty over all K. (d) Calibration curves at $N = 300$. (e) Occam's razor at $N = 300$.

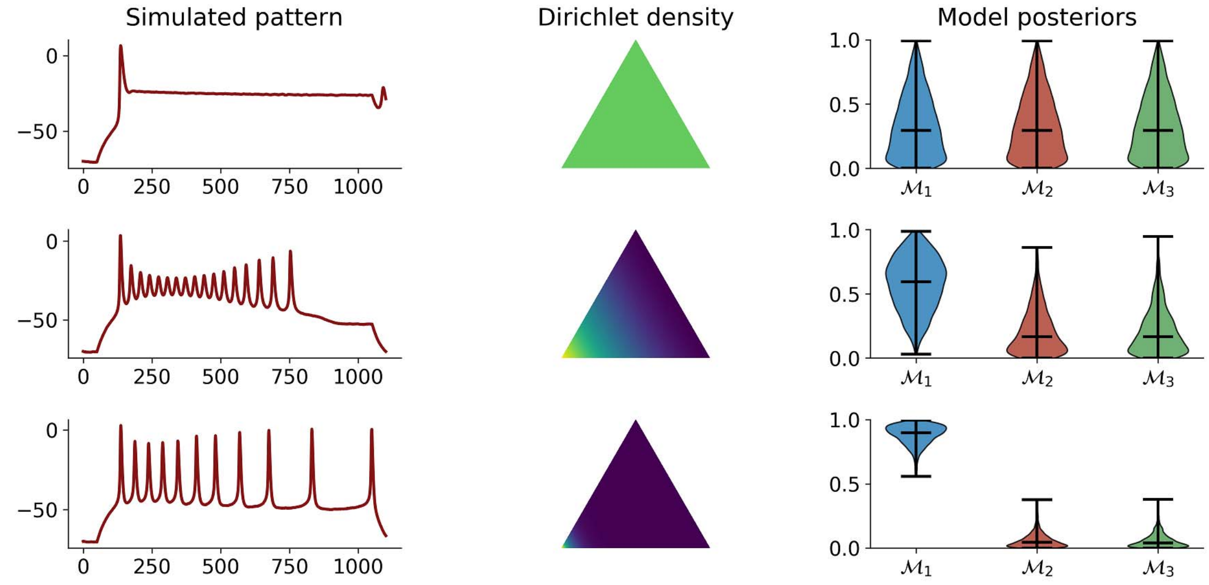


Fig. 6. Three simulated firing patterns, corresponding estimated Dirichlet densities, and model posteriors. Each row illustrates a different value of the parameter $\bar{g}_K$: $\bar{g}_K = 0.1$, $\bar{g}_K = 0.5$, and $\bar{g}_K = 0.75$, respectively. An increase in the parameter $\bar{g}_K$ is accompanied by a decrease in epistemic uncertainty [as measured via (10)]. An implausible value of $\bar{g}_k$ (first row) results in a flat density as an index of total uncertainty (uniform green areas). As the parameter value surpasses the plausible boundary (second and third rows), the Dirichlet simplex becomes peaked toward the bottom left edge encoding $\mathcal{M}_1$.

architectures that are able to perform amortized model comparison as classification. To this aim, we rely on a renowned computational model of biological neuronal dynamics.

1) Model Comparison Setting: In computational neuroscience, mathematical modeling of neuroelectric dynamics serves as a basis to understand the functional organization of the brain from both single-cell and large-scale network processing perspectives [1], [4], [9], [20], [21]. A plurality of different neural models has been proposed during the past decades, spanning from completely abstract to biologically detailed models. The former offers a simplified mathematical representation able to account for the main functional properties of spiking neurons, and the latter provides a detailed analogy between models' state variables and ion channels in biological neurons [38]. Importantly, these computational models differ in their capability to reproduce firing patterns observed in real cortical neurons [22].

The model family we consider here is a Hodgkin-Huxley-type model of cerebral cortex and thalamic neurons [20], [39]. The forward model is formulated as a set of five ordinary differential equations (ODEs) describing how the neuron membrane potential $V(t)$ unfolds in time as a function of an injected current $I_{inj}(t)$ and ion channels' properties. See Appendix D for more details regarding the forward simulation process.

To set up the model comparison problem, we treat different types of conductance, g_L, $\bar{g}_{Na}$, $\bar{g}_K$, and $\bar{g}_M$, as free parameters, and formulate three neural models based on different parameter configurations. In particular, we consider three models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$ defined by the parameter sets $\boldsymbol{\theta}_1 = (\bar{g}_{Na}, \bar{g}_K)$, $\boldsymbol{\theta}_2 = (\bar{g}_{Na}, \bar{g}_K, \bar{g}_M)$, and $\boldsymbol{\theta}_3 = (\bar{g}_{Na}, \bar{g}_K, \bar{g}_M, g_L)$. When not treated as free parameters, we set $\bar{g}_M$ and g_L to default values, such that $\bar{g}_M = 0.07$ and $g_L = 0.1$.

We compare the performance of our evidential method with the following methods: a standard softmax classifier, a classifier with Monte Carlo dropout [10], and two variational classifiers using a KL [27] and a maximum mean discrepancy (MMD, [59]) latent space regularizer, respectively. We also train three evidential networks with $\lambda = 0$ (no regularization), $\lambda = 0.5$, and $\lambda = 1.0$ to better quantify the effects of performing regularized model comparison. Here, we do not consider nonamortized methods, such as ABC or ABC-MCMC, as implemented in [35], since they would have taken an infeasible amount of time to validate on hundreds of datasets.

2) Validation Results: To assess performance, we train an unregularized evidential network for 60 epochs resulting in 60 000 minibatch updates. For each batch, we draw a random input current duration $T \sim \mathcal{U}_D(100, 400)$ (in units of milliseconds), with the same constant input current, I_{inj}, for each dataset simulation. Here, T reflects the physical time window in which biological spiking patterns can unfold. Since the sampling rate of the membrane potential is fixed ($dt = 0.2$), T affects both the span of observable spiking behavior and the number of simulated data points. The entire training phase with online learning took approximately 2.5 h. On the other hand, model comparison on 5000 validation time series took approximately 0.7 s, which highlights the extreme efficiency gains obtainable via globally amortized inference.

Regarding model selection, we observe accuracies above 0.92 across all T, with no gains in accuracy for increasing T, which shows that even short input currents are sufficient for performing reliable model selection for these complex models. Furthermore, the mean bootstrap calibration curves and accuracies on 5000 validation datasets are depicted in Figure S4d. We observe good calibration for all three models, with calibration errors less than 0.1. Notably, the overconfidence was 0 for all three models. The normalized *confusion matrix* is depicted in Fig. S4b.

TABLE I

COMPARISON RESULTS FROM EXPERIMENT 4

Neural Architecture	Accuracy	Calibration Error
Evidential ($\lambda = 0$)	0.919 ± 0.004	$\mathbf{0.078 \pm 0.010}$
Evidential ($\lambda = 0.5$)	0.917 ± 0.006	0.097 ± 0.012
Evidential ($\lambda = 1.0$)	0.900 ± 0.006	0.095 ± 0.011
Softmax classifier	0.913 ± 0.006	0.105 ± 0.015
MC Dropout classifier	0.885 ± 0.005	0.087 ± 0.009
MMD-VAE classifier	0.906 ± 0.006	0.091 ± 0.012
VAE classifier	$\mathbf{0.924 \pm 0.005}$	0.096 ± 0.012

To assess how well we can capture epistemic uncertainty for biologically implausible firing patterns, we train another evidential network with a gradually increasing regularization weight up to $\lambda = 1.0$. We then fix the parameter $\bar{g}_{Na} = 4.0$ of model $\mathcal{M}_1$ and gradually increase its second parameter $\bar{g}_K$ from 0.1 to 2.0. Since spiking patterns observed with low values of $\bar{g}_K$ are quite implausible and have not been observed during training, we expect uncertainty to gradually decrease. Three example firing patterns and the corresponding posterior estimates are depicted in Fig. 6. On the other hand, changing the sign of the output membrane potential, which results in biologically implausible firing patterns, leads to a trivial selection of $\mathcal{M}_3$. This is contrary to the expectations and shows that absolute evidence is also relative to what features the evidential network has learned during training.

Finally, Table I presents the comparison results in terms of accuracy and calibration error (all methods achieved 0 overconfidence). We train each neural network method for 30 epochs with identical optimizer settings and the same recurrent network architecture for ease of comparison. We then compute validation metrics on 3000 simulated neural firing patterns and report means and standard errors. Our unregularized evidential network ($\lambda = 0$) achieves the lowest calibration error, followed by the MC dropout classifier. In terms of accuracy, the KL variational classifier performs slightly better than our unregularized evidential network (but still within one standard deviation). Overall, the performance of all amortized methods considered in this experiment is similar, which highlights the viability of amortizing Bayesian model comparison in general. Note that training of each method took less than 1.5 h, and bootstrap validation on 3000 less than a minute. The latter would have been impossible to achieve within a reasonable time-frame using nonamortized methods.

VI. Discussion

In this article, we introduced a novel simulation-based method for approximate Bayesian model comparison based on specialized evidential neural networks. We demonstrated that our method can successfully deal with both exchangeable and nonexchangeable (time-dependent) sequences with variable numbers of observations without relying on fixed summary statistics. Furthermore, we presented a way to amortize the process of model comparison for a given family of models by splitting it into a potentially costly global training phase and a cheap inference phase. In this way, pretrained evidential networks can be stored, shared, and reused across multiple datasets and model comparison applications. Finally, we demonstrated a way to obtain a measure of absolute evidence in spite of operating in an $\mathcal{M}$-closed framework during the simulation phase. In the following, we reiterate the main advantages of our method.

A. Theoretical Guarantee

Using a strictly proper loss [14], we showed that our method can closely approximate analytic model posterior probabilities and BFs in theory and practice. In other words, posterior probability estimates are perfectly calibrated to the true model posterior probabilities when the strictly proper logarithmic loss is globally minimized. Indeed, our experiments confirm that the network outputs are well-calibrated. However, when optimizing the regularized version of the logarithmic loss, we are no longer working with a strictly proper loss, so calibration declines at the cost of capturing implausible datasets. However, we demonstrated that the accuracy of recovery (i.e., selecting the most plausible model in the set of considered models) does not suffer when training with regularization. In any case, perfect convergence is never guaranteed in finite sample scenarios, so validation tools such as calibration and accuracy curves are indispensable in practical applications.

B. Amortized Inference

Following ideas from *inference compilation* [30] and *prepaid parameter estimation* [36], our method avoids fitting each candidate model to each dataset separately. Instead, we cast the problem of model comparison as a supervised learning of absolute evidence and train a specialized neural network architecture to assign model evidences to each possible dataset. This requires only the specification of plausible priors over each model's parameters and the corresponding forward process, from which simulations can be obtained on the fly. During the upfront training, we use online learning to avoid storage overhead due to large simulated grids or reference tables [34], [36]. Importantly, the separation of model comparison into a costly upfront training phase and a cheap inference phase ensures that subsequent applications of the pretrained networks to multiple observed datasets are very efficient. Indeed, we showed in our experiments that inference on thousands of datasets can take less than a second with our method. Moreover, by sharing and applying a pretrained network for inference within a particular research domain, the results will be highly reproducible, since the *settings* of the method will be held constant in all applications.

C. Raw Data Utilization and Variable Sample Size

The problem of insufficient summary statistics has plagued the field of ABC for a long time, so as to deserve being dubbed the *curse of insufficiency* [34]. Using suboptimal summary statistics can severely compromise the quality of posterior approximations and validity of conclusions based on these approximations [44]. Our method avoids using hand-crafted summaries by aligning the architecture of the evidential neural network to the inherent probabilistic symmetry of the data [2]. Using specialized neural network architectures, such as permutation invariant networks or a combination of recurrent and convolutional networks, we also ensure that our method can deal with datasets containing variable numbers of observations. Moreover, by minimizing the strictly proper version of the logarithmic loss, we ensure that perfect convergence implies maximal data utilization by the network.

D. Absolute Evidence and Epistemic Uncertainty

Besides point estimates of model posterior probabilities, our evidential networks yield a full higher order probability distribution over the posterior model probabilities themselves. By choosing a Dirichlet distribution, we can use the mean of the Dirichlet distribution as the best approximation of model posterior probabilities. Beyond that, following ideas from the study of SL [25] and uncertainty quantification in classification tasks [45], we can extract a measure of *epistemic uncertainty*. We use epistemic uncertainty to quantify the impossibility of making a model selection decision based on a dataset, which is classified as implausible under all candidate models. Therefore, the epistemic uncertainty serves as a proxy to measure absolute evidence, in contrast to relative evidence, as given by BFs or posterior odds. This is an important practical advantage, as it allows us to conclude that all models in the candidate set are a poor approximation of the data-generating process of interest. Indeed, our initial experiments confirm that our measure of epistemic uncertainty increases when datasets no longer lie within the range of the considered models. However, extensive validation is needed to explore and understand which aspects of an observed sample lead to model misfit. Furthermore, exploring connections to approaches using auxiliary probabilistic classifiers for detecting model misspecifications, such as the recent CARMEN method [48], seems to be an interesting avenue for future research.

These advantageous properties notwithstanding our method have certain limitations. First, our regularized optimization criterion induces a trade-off between calibration and epistemic uncertainty, as confirmed by our experiments. This trade-off is due to the fact that we capture epistemic uncertainty via a special form of KL regularization during the training phase, which renders the optimized loss function no longer strictly proper. We leave it to future research to investigate whether this trade-off is fundamental and whether there are more elegant ways to quantify absolute evidence from a simulation-based perspective.

Second, our method is intended for model comparison from a prior predictive (marginal likelihood) perspective. However, since we do not explicitly fit each model to data, we cannot perform model comparison/selection based on posterior predictive performance. We note that in certain scenarios, posterior predictive performance might be a preferable metric for model comparison, so in this case simulation-based sampling methods should be used (e.g., ABC or neural density estimation, [7], [37]).

Third, perfect convergence might be hard to achieve in real-world applications. In this case, approximation error will propagate into model posterior estimates. Therefore, it is important to use performance diagnostic tools, such as calibration curves, accuracy of recovery, and overconfidence bounds, to detect potential estimation problems. Finally, even though our method exhibited excellent performance on the domain examples considered in this article, it might break down in high-dimensional parameter spaces. Future research should focus on applications to even more challenging model comparison scenarios, for instance, hierarchical Bayesian models with intractable likelihoods, or neural network models.

Acknowledgment

The authors would like to thank D. Izydorczyk and M. Sensi for reading this article and providing constructive feedback.

References

[1] L. F. Abbott and T. B. Kepler, “Model neurons: From Hodgkin–Huxley to hopfield,” in *Statistical Mechanics of Neural Networks*. Berlin, Germany: Springer, 1990, pp. 5–18.

[2] B. Bloem-Reddy and Y. W. Teh, "Probabilistic symmetries and invariant neural networks," 2019, *arXiv:1901.06082*.

[3] S. D. Brown and A. Heathcote, "The simplest complete model of choice response time: Linear ballistic accumulation," *Cogn. Psychol.*, vol. 57, no. 3, pp. 153–178, 2008.

[4] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, Jul. 2006.

[5] P.-C. Bürkner, J. Gabry, and A. Vehtari, "Approximate leave-future-out cross-validation for Bayesian time series models," *J. Stat. Comput. Simul.*, vol. 90, no. 14, pp. 2499–2523, Sep. 2020.

[6] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 48, pp. 30055–30062, Dec. 2020.

[7] J. M. J. D. Costa, H. R. B. Orlande, and W. B. D. Silva, "Model selection and parameter estimation in tumor growth models using approximate Bayesian computation-ABC," *Comput. Appl. Math.*, vol. 37, no. 3, pp. 2795–2815, Jul. 2018.

[8] T. S. Deisboeck and G. S. Stamatakos, *Multiscale Cancer Modeling*. Boca Raton, FL, USA: CRC Press, 2010.

[9] S. Dura-Bernal *et al.*, "NetPyNE, a tool for data-driven multiscale modeling of brain circuits," *ELife*, vol. 8, Apr. 2019, Art. no. e44494.

[10] Y. Gal and Z. Ghahramani, "Dropout as Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[11] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2013.

[12] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Stat. Sci.*, vol. 13, no. 2, pp. 163–185, May 1998.

[13] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[14] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *J. Amer. Statist. Assoc.*, vol. 102, no. 477, pp. 359–378, 2007.

[15] D. S. Greenberg, M. Nonnenmacher, and J. H. Macke, "Automatic posterior transformation for likelihood-free inference," 2019, *arXiv:1905.07488*.

[16] Q. F. Gronau *et al.*, "A tutorial on bridge sampling," *J. Math. Psychol.*, vol. 81, pp. 80–97, Dec. 2017.

[17] P. Grünwald and T. van Ommen, "Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it," *Bayesian Anal.*, vol. 12, no. 4, pp. 1069–1103, Dec. 2017.

[18] J. Hermans, V. Begy, and G. Louppe, "Likelihood-free MCMC with amortized approximate ratio estimators," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4239–4248.

[19] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.

[20] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952.

[21] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[22] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[23] E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[24] W. Wong, B. Jiang, T.-Y. Wu, and C. Zheng, "Learning summary statistic for approximate Bayesian computation via deep neural network," *Statistica Sinica*, vol. 27, no. 4, pp. 1595–1618, 2017.

[25] A. Jsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Cham, Switzerland: Springer, 2018.

[26] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5580–5590.

[27] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2575–2583.

[28] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[29] E. Klinger, D. Rickert, and J. Hasenauer, "PyABC: Distributed, likelihood-free inference," *Bioinformatics*, vol. 34, no. 20, pp. 3591–3593, Oct. 2018.

[30] T. A. Le, A. G. Baydin, and F. Wood, "Inference compilation and universal probabilistic programming," 2016, *arXiv:1610.09900*.

[31] Y. Li and Y. Gal, "Dropout inference in Bayesian neural networks with alpha-divergences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2052–2061.

[32] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational Bayesian neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2218–2227.

[33] D. J. C. MacKay, "Bayesian neural networks and density networks," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detectors Associated Equip.*, vol. 354, no. 1, pp. 73–80, 1995.

[34] J.-M. Marin, P. Pudlo, A. Estoup, and C. Robert, *Likelihood-Free Model Choice*. Boca Raton, FL, USA: CRC Press, 2018.

[35] U. K. Mertens, A. Voss, and S. Radev, "ABrox—A user-friendly Python module for approximate Bayesian computation with a focus on model comparison," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, Art. no. e0193981.

[36] M. Mestdagh, S. Verdonck, K. Meers, T. Loossens, and F. Tuerlinckx, "Prepaid parameter estimation without likelihoods," *PLOS Comput. Biol.*, vol. 15, no. 9, Sep. 2019, Art. no. e1007181.

[37] G. Papamakarios, D. C. Sterratt, and I. Murray, "Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows," 2018, *arXiv:1805.07226*.

[38] M. Pospischil, Z. Piwkowska, T. Bal, and A. Destexhe, "Comparison of different neuron models to conductance-based post-stimulus time histograms obtained in cortical pyramidal cells using dynamic-clamp *in vitro*," *Biol. Cybern.*, vol. 105, no. 2, pp. 167–180, 2011.

[39] M. Pospischil *et al.*, "Minimal Hodgkin–Huxley type models for different classes of cortical and thalamic neurons," *Biol. Cybern.*, vol. 99, nos. 4–5, pp. 427–441, 2008.

[40] P. Pudlo, J.-M. Marin, A. Estoup, J.-M. Cornuet, M. Gautier, and C. P. Robert, "Reliable ABC model choice via random forests," *Bioinformatics*, vol. 32, no. 6, pp. 859–866, Mar. 2016.

[41] S. T. Radev, U. K. Mertens, A. Voss, L. Ardizzone, and U. Kothe, "BayesFlow: Learning complex stochastic models with invertible neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 18, 2020, doi: 10.1109/TNNLS.2020.3042395.

[42] R. Ratcliff and G. McKoon, "The diffusion decision model: Theory and data for two-choice decision tasks," *Neural Comput.*, vol. 20, no. 4, pp. 873–922, 2008.

[43] O. Ratmann, C. Andrieu, C. Wiuf, and S. Richardson, "Model criticism based on likelihood-free inference, with an application to protein network evolution," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 26, pp. 10576–10581, Jun. 2009.

[44] C. P. Robert, J.-M. Cornuet, J.-M. Marin, and N. S. Pillai, "Lack of confidence in approximate Bayesian computation model choice," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 37, pp. 15112–15117, 2011.

[45] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3179–3189.

[46] S. A. Sisson and Y. Fan, *Likelihood-Free MCMC*. New York, NY, USA: Chapman & Hall, 2011.

[47] M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, and C. Dessimoz, "Approximate Bayesian computation," *PLoS Comput. Biol.*, vol. 9, no. 1, 2013, Art. no. e1002803.

[48] O. Thomas and J. Corander, "Diagnosing model misspecification and performing generalized Bayes' updates via probabilistic classifiers," 2019, *arXiv:1912.05810*.

[49] B. M. Turner and P. B. Sederberg, "A generalized, likelihood-free method for posterior estimation," *Psychonomic Bull. Rev.*, vol. 21, no. 2, pp. 227–250, 2014.

[50] B. M. Turner, P. B. Sederberg, and J. L. McClelland, "Bayesian analysis of simulation-based models," *J. Math. Psychol.*, vol. 72, pp. 191–199, Jun. 2016.

[51] M. Usher and J. L. McClelland, "The time course of perceptual choice: The leaky, competing accumulator model," *Psychol. Rev.*, vol. 108, no. 3, p. 550, 2001.

[52] A. Vehtari, A. Gelman, and J. Gabry, "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC," *Statist. Comput.*, vol. 27, no. 5, pp. 1413–1432, 2017.

[53] A. Vehtari and J. Ojanen, "A survey of Bayesian predictive methods for model assessment, selection and comparison," *Statist. Surv.*, vol. 6, pp. 142–228, Dec. 2012.

[54] R. Verity *et al.*, "Estimates of the severity of coronavirus disease 2019: A model-based analysis," *Lancet Infectious Diseases*, vol. 20, no. 6, pp. 669–677, 2020.

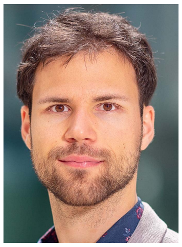
[55] A. Voss, V. Lerche, U. Mertens, and J. Voss, "Sequential sampling models with variable boundaries and non-normal noise: A comparison of six models," *Psychonomic Bull. Rev.*, vol. 26, no. 3, pp. 813–832, 2019.

[56] S. Watanabe, "WAIC and WBIC are information criteria for singular statistical model evaluation," in *Proc. Workshop Inf. Theoretic Methods Sci. Eng.*, 2013, pp. 90–94.

[57] K. Xu, J. Li, M. Zhang, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka, "What can neural networks reason about?" 2019, *arXiv:1905.13211*.

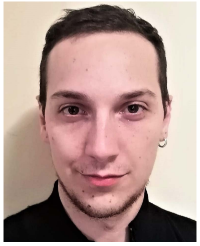
[58] Y. Yao, A. Vehtari, D. Simpson, and A. Gelman, "Using stacking to average Bayesian predictive distributions (with discussion)," *Bayesian Anal.*, vol. 13, no. 3, pp. 917–1007, 2018.

[59] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information maximizing variational autoencoders," 2017, *arXiv:1706.02262*.



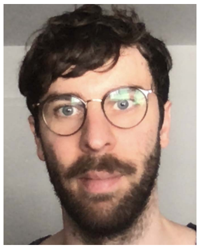
Stefan T. Radev received the Ph.D. degree in statistical modeling in psychology from Heidelberg University, Heidelberg, Germany, in 2021.

His research focuses on the convergence of Bayesian inference, deep learning, and mathematical models of complex processes at the intersection of cognitive science and artificial intelligence.



Marco D'Alessandro received the Ph.D. degree in cognitive modeling from the University of Trento, Trento, Italy, in 2021.

His research interests focus on developing computational models and statistical methods to investigate how human cognitive systems internally represent and manipulate information in uncertain environments.



Ulf K. Mertens received the Ph.D. degree in psychology/quantitative research methods from Heidelberg University, Heidelberg, Germany, in 2019.

He is currently a Data Scientist with Blue Yonder, Karlsruhe, Germany, where he develops transformer-based models for time series forecasting. His research interests include Bayesian inference, generative modeling, and natural language processing.

Andreas Voss received the Ph.D. degree in psychology from the University of Trier, Trier, Germany, in 2004.

He is currently a Professor of quantitative research methods with the Faculty of Psychology, Heidelberg University, Heidelberg, Germany. His research interests include statistical modeling of cognitive processes, Bayesian methods, and machine learning.

Ullrich Köthe received the Diploma degree in physics from the University of Rostock, Rostock, Germany, in 1991, and the Ph.D. degree in computer science from the University of Hamburg, Hamburg, Germany, in 2000.

He is currently an Adjunct Professor of computer science with the Interdisciplinary Center for Scientific Computing, Heidelberg University, Heidelberg, Germany. His research focuses on the connection between machine learning and the sciences from a methodological perspective and an application perspective and, in particular, on the interpretability of machine learning results.

Paul-Christian Bürkner received the Ph.D. degree in psychology from the University of Münster, Münster, Germany, in 2017.

He is a Statistician currently working as a Junior Research Group Leader for Bayesian Statistics at the Cluster of Excellence SimTech, University of Stuttgart, Stuttgart, Geramny. He is interested in a wide range of research topics most of which involve the development, evaluation, implementation, or application of Bayesian methods. His most recent work focuses, among others, on deep learning approaches to simulation-based inference.