Psychological Methods

A Deep Learning Method for Comparing Bayesian Hierarchical Models

Lasse Elsemüller, Martin Schnuerch, Paul-Christian Bürkner, and Stefan T. Radev

A Deep Learning Method for Comparing Bayesian Hierarchical Models

Lasse Elsemüller[1], Martin Schnuerch[2], Paul-Christian Bürkner[3], and Stefan T. Radev[4]

[1] Institute of Psychology, Heidelberg University
[2] Department of Psychology, University of Mannheim
[3] Department of Statistics, TU Dortmund University
[4] Cluster of Excellence STRUCTURES, Heidelberg University

Abstract

Bayesian model comparison (BMC) offers a principled approach to assessing the relative merits of competing computational models and propagating uncertainty into model selection decisions. However, BMC is often intractable for the popular class of hierarchical models due to their high-dimensional nested parameter structure. To address this intractability, we propose a deep learning method for performing BMC on any set of hierarchical models which can be instantiated as probabilistic programs. Since our method enables amortized inference, it allows efficient re-estimation of posterior model probabilities and fast performance validation prior to any real-data application. In a series of extensive validation studies, we benchmark the performance of our method against the state-of-the-art bridge sampling method and demonstrate excellent amortized inference across all BMC settings. We then showcase our method by comparing four hierarchical evidence accumulation models that have previously been deemed intractable for BMC due to partly implicit likelihoods. Additionally, we demonstrate how transfer learning can be leveraged to enhance training efficiency. We provide reproducible code for all analyses and an open-source implementation of our method.

Keywords: Bayesian statistics, model comparison, hierarchical modeling, deep learning, cognitive modeling

Supplemental materials: https://doi.org/10.1037/met0000645.supp

Hierarchical models (HMs) or multilevel models play an increasingly important methodological role in the social and cognitive sciences (Farrell & Lewandowsky, 2018; Rouder et al., 2017). HMs embody probabilistic and structural information about nested data occurring frequently in various settings, such as educational research (Ulitzsch et al., 2020), experimental psychology (Vandekerckhove et al., 2011), epidemiology (Jalilian & Mateu, 2021) or astrophysics (Hinton et al., 2019), to name just a few. Crucially, HMs can often extract more information from rich data structures than their nonhierarchical counterparts (e.g., aggregate analyses), while retaining a relatively high intrinsic interpretability of their structural components

(i.e., parameters). Moreover, viewed as formal instantiations of scientific hypotheses, HMs can be employed to systematically assign preferences to these hypotheses by means of formal model comparison. For example, Haaf and Rouder (2017) proposed a powerful framework based on Bayesian HMs for formulating and testing competing theoretical positions on quantitative versus qualitative individual differences.

We consider Bayesian model comparison (BMC) as a principled framework for comparing and ranking competing HMs via Occam's razor (Kass & Raftery, 1995; Lotfi et al., 2022; MacKay, 2003). However, standard BMC is analytically intractable for nontrivial HMs, as it requires marginalization over high-dimensional parameter

Lasse Elsemüller ![ORCID] https://orcid.org/0000-0003-0368-720X
Martin Schnuerch ![ORCID] https://orcid.org/0000-0001-6531-2265
Paul-Christian Bürkner ![ORCID] https://orcid.org/0000-0001-5765-8995
Stefan T. Radev ![ORCID] https://orcid.org/0000-0002-6702-9559

Correspondence concerning this article should be addressed to Lasse Elsemüller, Institute of Psychology, Heidelberg University, Hauptstraße 47, 69117 Heidelberg, Germany. Email: lasse.elsemueller@gmail.com

spaces. Moreover, BMC for complex HMs without explicit likelihoods (i.e., HMs available only as randomized simulators) becomes increasingly hopeless and precludes many interesting applications in the rapidly expanding field of simulation-based inference (Cranmer et al., 2020).

In this work, we propose to tackle the problem of BMC for arbitrarily complex HMs from a simulation-based perspective using deep learning. In particular, we build on the BayesFlow framework (Radev, D'Alessandro, et al., 2021; Radev et al., 2020) for simulation-based Bayesian inference and propose a novel hierarchical neural network architecture for approximating Bayes factors (BFs) and posterior model probabilities (PMPs) for any collection of HMs.

Our neural approach circumvents the steps of explicitly fitting all models and marginalizing over the parameter space of each model. Thus, it is applicable to both HMs with explicit likelihood functions and HMs accessible only through Monte Carlo simulations (i.e., with implicit likelihood functions). Moreover, our neural networks come with an efficient way to compute their calibration error (Guo et al., 2017), which provides an important diagnostic for self-consistency. Lastly, trained networks can be adapted to related tasks, substantially reducing the computational burden when dealing with demanding simulators.

The remainder of this article is organized as follows. In the Theoretical Background section, we introduce the theoretical background and related work on (hierarchical) BMC. We then present the rationale and details of our deep learning method in the Method section. In the Experiments section, we first present two validation studies of the proposed method: one that includes toy models for illustrative purposes and one that includes two popular classes of models from the field of cognitive psychology. We then apply our method to compare hierarchical evidence accumulation models with partly intractable likelihoods on a real data set. Finally, the Discussion section summarizes our contributions and discusses future perspectives.

Theoretical Background

Bayesian Hierarchical Modeling

In order to streamline statistical analyses, researchers rely on assumptions about the probabilistic structure or symmetry of the assumed data-generating process. For instance, the canonical assumption of independent and identically distributed (IID) data in psychological modeling states that (multivariate) observations are independent of each other and sampled from the same latent probability distribution (Nicenboim et al., 2022; Singmann & Kellen, 2019).

However, more complex dependencies may arise in a variety of contexts. For instance, if there are repeated measurements per participant or participants belong to different natural groups (e.g., school classes and working groups), the respective observations exhibit higher correlations within those clusters than across them. Ignoring this nested structure in statistical analyses may result in biased conclusions (Singmann & Kellen, 2019). Bayesian HMs formalize this structural knowledge by assuming that observations are sampled from a multilevel generative process (Gelman, 2006).

For instance, the generative recipe for a two-level Bayesian HM can be written as:

$$\boldsymbol{\eta} \sim p(\boldsymbol{\eta}), \tag{1}$$

$$\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta}|\boldsymbol{\eta}) \quad \text{for } m = 1, \ldots, M, \tag{2}$$

$$\boldsymbol{x}_{mn} \sim p(\boldsymbol{x}|\boldsymbol{\theta}_m) \quad \text{for } n = 1, \ldots, N_m, \tag{3}$$

where $\boldsymbol{\eta}$ denotes the group-level parameters, $\boldsymbol{\theta}_m$ denotes the individual parameters in group m, and $\boldsymbol{x}_{mn}$ represents the nth observation in group m. Such a model suggests the following (nonunique) factorization of the joint distribution:

$$p(\boldsymbol{\eta}, \{\boldsymbol{\theta}_m\}, \{\boldsymbol{x}_{mn}\}) = p(\boldsymbol{\eta}) \prod_{m=1}^{M} p(\boldsymbol{\theta}_m|\boldsymbol{\eta}) \prod_{n=1}^{N_m} p(\boldsymbol{x}_{mn}|\boldsymbol{\theta}). \tag{4}$$

The set notation $\{\boldsymbol{\theta}_m\}$ and $\{\boldsymbol{x}_{mn}\}$ implies that the number of groups and observations in each group can vary across simulations, data sets and experiments and that these quantities are exchangeable.

HMs can be considered as a compromise between a separate analysis of each group (no-pooling) that neglects the information contained in the rest of the data and an aggregate analysis of the data (complete pooling) that loses the distinction between intragroup and intergroup variability (Hox et al., 2017). The partial pooling of information induced by HMs leads to more stable and accurate individual estimates through the shrinkage properties of multilevel priors, whereby single estimates inform each other (Bürkner, 2017; Gelman, 2006).

Despite having desirable properties, hierarchical modeling comes at the cost of increased complexity and computational demands. These increased demands make it hard or even impossible to compare competing HMs within the probabilistic framework of BMC. Before we highlight these challenges, we first describe the basics of BMC for nonhierarchical models.

Bayesian Model Comparison

The starting point of BMC is a collection of J competing generative models $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_J\}$. Each $\mathcal{M}_j$ is associated with a prior $p(\boldsymbol{\theta}_j|\mathcal{M}_j)$ on the parameters $\boldsymbol{\theta}_j$ and a generative mechanism, which is either defined analytically through a (tractable) likelihood density function $p(\boldsymbol{x}|\boldsymbol{\theta}_j, \mathcal{M}_j)$ or realized as a Monte Carlo simulation program $g_j(\boldsymbol{\theta}, \boldsymbol{z})$ with random states $\boldsymbol{z}$. Together, the prior and the likelihood define the Bayesian joint model

$$p(\boldsymbol{\theta}_j, \boldsymbol{x}|\mathcal{M}_j) = p(\boldsymbol{\theta}_j|\mathcal{M}_j)p(\boldsymbol{x}|\boldsymbol{\theta}_j, \mathcal{M}_j), \tag{5}$$

which is also tacitly defined for simulator-based models by marginalizing the joint distribution $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}_j, \mathcal{M}_j)$ over all possible execution paths (i.e., random states) of the simulation program to obtain the implicit likelihood

$$p(\boldsymbol{x}|\boldsymbol{\theta}_j, \mathcal{M}_j) = \int p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}_j, \mathcal{M}_j)\mathrm{d}\boldsymbol{z}. \tag{6}$$

This integral is typically intractable for complex simulators (Cranmer et al., 2020), which makes it impossible to evaluate the likelihood and use standard Bayesian methods for parameter inference or model comparison.

The likelihood function, be it explicit or implicit, is a key object in Bayesian inference. When the parameters $\boldsymbol{\theta}$ are systematically varied and the data $\boldsymbol{x}$ held constant, the likelihood quantifies the relative fit of each model instantiation (defined by a fixed configuration $\boldsymbol{\theta}$) to the observed data.

When we marginalize the Bayesian joint model (Equation 5) over its parameter space, we obtain the marginal likelihood or Bayesian

evidence (see MacKay, 2003, Chapter 28):

$$p(\boldsymbol{x}|\mathcal{M}_j) = \int p(\boldsymbol{x}|\boldsymbol{\theta}_j, \mathcal{M}_j)p(\boldsymbol{\theta}_j|\mathcal{M}_j)\mathrm{d}\boldsymbol{\theta}_j. \qquad (7)$$

The marginal likelihood can be interpreted as the probability that we would generate data $\boldsymbol{x}$ from model $\mathcal{M}_j$ when we randomly sample from the model's parameter prior $p(\boldsymbol{\theta}_j|\mathcal{M}_j)$. Moreover, the marginal likelihood is a central quantity for prior predictive hypothesis testing or model selection (Kass & Raftery, 1995; O'Hagan, 1995; Rouder & Morey, 2012). It is well-known that the marginal likelihood encodes a notion of Occam's razor arising from the basic principles of probability (Kass & Raftery, 1995, see also Figure 1). Thus, the marginal likelihood provides a foundation for the widespread use of Bayes factors (Heck et al., 2022) or PMPs (Congdon, 2006) for BMC.

The relative evidence for a pair of models can be computed through the ratio of marginal likelihoods for the two competing models $\mathcal{M}_j$ and $\mathcal{M}_k$,

$$\mathrm{BF}_{jk} = \frac{p(\boldsymbol{x}|\mathcal{M}_j)}{p(\boldsymbol{x}|\mathcal{M}_k)}. \qquad (8)$$

This ratio is called Bayes factor (BF) and is widely used for quantifying pairwise model preference in Bayesian settings (Heck et al., 2022; Kass & Raftery, 1995). Accordingly, a $\mathrm{BF}_{jk} > 1$ indicates the preference for model j over model k given available data $\boldsymbol{x}$. Alternatively, one can directly focus on the (marginal) posterior probability of a model $\mathcal{M}_j$,

$$p(\mathcal{M}_j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{M}_j)p(\mathcal{M}_j)}{\sum_{j=1}^{J} p(\boldsymbol{x}|\mathcal{M}_j)p(\mathcal{M}_j)}, \qquad (9)$$

where $p(\mathcal{M}_j)$ is a categorical (typically uniform) prior distribution encoding a researcher's prior beliefs regarding the plausibility of each considered model. This prior distribution is then updated with the information contained in the marginal likelihood $p(\boldsymbol{x}|\mathcal{M}_j)$ to obtain the corresponding PMP, $p(\mathcal{M}_j|\boldsymbol{x})$. Occasionally in the text, we will refer to the vector of PMPs for all J models as $\boldsymbol{\pi}$ and to the individual PMPs as π_j. The ratio of two PMPs, known as posterior odds, is in turn connected to the BF via the corresponding model priors:

$$\frac{p(\mathcal{M}_j|\boldsymbol{x})}{p(\mathcal{M}_k|\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\mathcal{M}_j)}{p(\boldsymbol{x}|\mathcal{M}_k)} \times \frac{p(\mathcal{M}_j)}{p(\mathcal{M}_k)}. \qquad (10)$$

Despite its intuitive appeal, the marginal likelihood (and thus BFs and PMPs) represents a well-known and widely appreciated source of intractability in Bayesian workflows, since it typically involves a multidimensional integral (Equation 7) over potentially unbounded parameter spaces (Gronau, Sarafoglou, et al., 2017; Lotfi et al., 2022). Furthermore, the marginal likelihood becomes doubly intractable when the likelihood function is itself not available (e.g., in simulation-based settings), thereby making the comparison of such models a challenging and sometimes, up to this point, hopeless endeavor.

Unsurprisingly, estimating the marginal likelihood (Equation 7) in the context of hierarchical models becomes even more challenging, since the number of parameters over which we need to perform marginalization grows dramatically (i.e., parameters at all hierarchical levels enter the computation). These computational demands render the probabilistic comparison of HMs based on BFs or PMPs analytically intractable even for relatively simple models with explicit (analytical) likelihoods. Therefore, researchers need to resort to costly, approximate methods which typically only work for models with explicit likelihoods (Gelman & Meng, 1998; Gronau, Sarafoglou, et al., 2017; Meng & Schilling, 2002).

Approximate BMC

Explicit Likelihoods

The most efficient approximate methods to date require all candidate models to possess explicitly available likelihood functions. For the most simple scenario in which two HMs are nested (e.g., through an equality constraint on a parameter), the Savage–Dickey density ratio (Dickey & Lientz, 1970) provides a convenient approximation of the BF (Wagenmakers et al., 2010). Typically, however, the candidate models are not nested but exhibit notable structural differences. Thus, a general-purpose method is needed to encompass the entire plethora of model comparison scenarios arising in practical applications.

A more general method, and the current state-of-the-art for comparing HMs in psychological and cognitive modeling (Gronau et al.,

Figure 1
Hypothetical Bayesian Model Comparison Setting With a Simple Model $\mathcal{M}_1$ and a More Complex Model $\mathcal{M}_2$
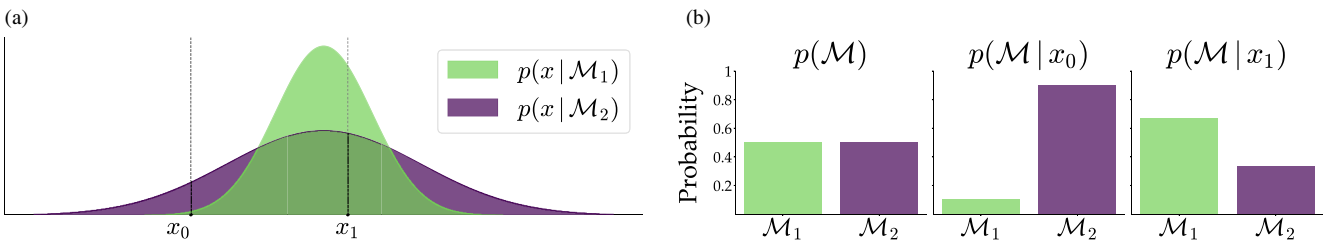
Note. (a) Marginal likelihoods: The complex model which accounts for a broader range of observations needs to spread its marginal likelihood to cover its larger generative scope. It does so at the cost of diminished sharpness. Thus, even though observation x_1 is well within its generative scope, the simpler model $\mathcal{M}_1$ yields a higher marginal likelihood and is therefore preferred. In contrast, observation x_0 has a higher marginal likelihood under model $\mathcal{M}_2$, as it is very unlikely to be generated by the simpler model $\mathcal{M}_1$. (b) The corresponding posterior model probabilities given a uniform model prior. See the online article for the color version of this figure.

2020, 2019; Schad et al., 2023), is given by bridge sampling (Bennett, 1976; Meng & Wong, 1996). Bridge sampling has enabled comparisons within families of complex process models, such as multinomial processing trees (MPTs; Gronau et al., 2019) or evidence accumulation models (EAMs; Gronau et al., 2020), and serves as a simple add-on for Markov chain Monte Carlo (MCMC)-based Bayesian workflows.

Crucially, bridge sampling relies on the posterior draws generated by an MCMC sampler (e.g., Stan; Carpenter et al., 2017) to efficiently approximate the marginal likelihood of each respective model (Gronau, Sarafoglou, et al., 2017). Note, however, that bridge sampling requires considerably more random draws for stable results than standard parameter estimation (usually about an order of magnitude more; Gronau, Singmann, et al., 2017). Moreover, the approximation quality of bridge sampling is dependent on the convergence of the MCMC chains (Gronau et al., 2020). Finally, there are no strong theoretical guarantees that the approximations are unbiased and accurately reflect the true marginal likelihoods (Schad et al., 2023).

Implicit Likelihoods

With the rise of complex, high-resolution models, intractable likelihood functions (i.e., functions that do not admit a closed form or are too costly to evaluate) become more and more common in statistical modeling. Such models are not limited to psychology and cognitive science (Nicenboim et al., 2022; Van Rooij et al., 2019), but are also common in fields such as neuroscience (Gonçalves et al., 2020), epidemiology (Radev, Graw, et al., 2021), population genetics (Pudlo et al., 2016), or astrophysics (Hermans et al., 2021). Despite the common term likelihood-free, simulator-based models still possess an implicitly defined likelihood (see Bayesian Model Comparison section) from which we can obtain random draws through Monte Carlo simulations. This enables model comparison through simulation-based methods, usually by means of approximate Bayesian computation (ABC; Marin et al., 2018; Mertens et al., 2018; Pudlo et al., 2016).

Traditional (rejection-based) ABC methods for BMC repeatedly simulate data sets from the specified generative models, retaining only those simulations that are sufficiently similar to the empirical data. To enable the calculation of this (dis-)similarity even in high-dimensional cases, the information contained in the simulated data sets is reduced by computing hand-crafted summary statistics, such as the mean and variance (Csilléry et al., 2010; Sunnåker et al., 2013). The resulting acceptance rates of the candidate models represent the approximations of their PMPs (Marin et al., 2018; Mertens et al., 2018).

Even for nonhierarchical models, ABC methods are known to be notoriously inefficient and highly dependent on the concrete choice of summary statistics (Cranmer et al., 2020; Marin et al., 2018). This choice is even more challenging for HMs, as modelers now have to retain an optimal amount of information on multiple levels. Moreover, the rapidly growing number of summary statistics reduces the probability that a simulated data set is similar enough to the empirical data, which vastly increases the number of required simulations (Beaumont, 2010; Marin et al., 2018).

Regardless of the number of summary statistics, their manual computation carries the danger of insufficiently summarizing the simulations and thereby producing biased approximations (a phenomenon known as the curse of insufficiency; Marin et al., 2018).

While many improvements of rejection-based ABC have been proposed, most notably ABC-MCMC (Marjoram et al., 2003; Turner & Sederberg, 2014), ABC-SMC (Sisson et al., 2007), as well as Gibbs ABC (Turner & Van Zandt, 2014) for Bayesian hierarchical modeling in particular (see also G. Clarté et al., 2021; Fengler et al., 2021), these advancements are still limited by their dependence on hand-crafted summary statistics or kernel density estimation methods.

Recent developments, such as ABC-RF (Pudlo et al., 2016), combine ABC with machine learning methods to build more expressive approximators for BMC problems. Accordingly, model comparison is treated as a supervised learning problem—the simulated data encompasses a training set for a machine learning algorithm that learns to recognize the true generative model from which the data set was simulated. The machine learning approach reduces the inefficiency problem that haunts rejection-based ABC methods, but does not alleviate the curse of insufficiency (Marin et al., 2018).

BMC With Neural Networks

Recently, Radev, D'Alessandro, et al. (2021) explored a method for simulation-based BMC using specialized neural networks. The authors proposed to jointly train two specialized neural networks using Monte Carlo simulations from each candidate model in $\mathcal{M}$: a summary network and an evidential network. The goal of the summary network is to extract maximally informative (in the optimal case, sufficient) summary statistics from complex data sets. The goal of the evidential network is to approximate PMPs as accurately as possible and, optionally, to quantify their epistemic uncertainty.

Importantly, simulation-based training of neural networks enables amortized inference for both implicit and explicit likelihood models. Amortization is a property that ensures rapid inference for an arbitrary amount of data sets after a potentially high computational investment in simulation and training (Mestdagh et al., 2019; Radev, D'Alessandro, et al., 2021; Radev et al., 2020). As a consequence, the calibration (Guo et al., 2017; Talts et al., 2018) or the inferential adequacy (Schad et al., 2021, 2023) of an amortized Bayesian method are embarrassingly easy to validate in practice.

In contrast, nonamortized methods, such as ABC-MCMC (Turner & Sederberg, 2014) or ABC-SMC (Sisson et al., 2007) need to repeat all computations from scratch for each observed data set. Thereby, it is often infeasible to assess their calibration or inferential adequacy in the predata phase of a Bayesian workflow (Gelman et al., 2020).

Unfortunately, the evidential method proposed by Radev, D'Alessandro, et al. (2021) is not applicable to HMs due to their nested probabilistic structure which cannot be tackled via previous summary networks. This severely limits the applicability of the method in quantitative research, where HMs have been advocated as a default choice (Lee, 2011; McElreath, 2020; Rouder et al., 2017). In the following, we describe how to extend the original method to enable amortized BMC for HMs.

Method

At its core, our method involves a multilevel permutation invariant neural network which is aligned to the probabilistic symmetry of the underlying HMs (see Figure 2 for a visualization). We hold that any method which does not rely on ad hoc summary statistics should

Figure 2

Our Proposed Hierarchical Neural Network Architecture for Encoding Permutation Invariance in the Transformation of Nested, Two-Level Data Into Posterior Model Probabilities
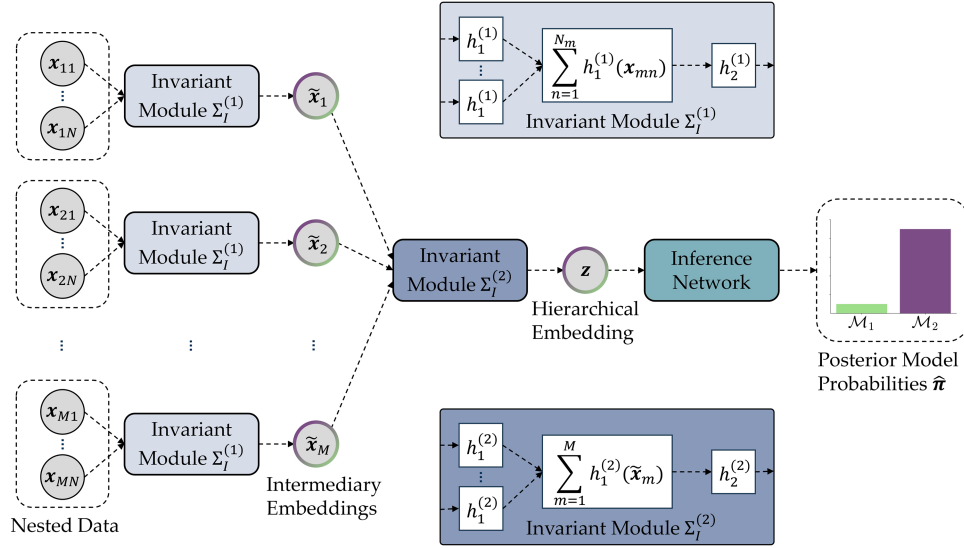
Note. A first invariant module $\Sigma_I^{(1)}$ reduces all N_m observations within each of the M groups to a single intermediary embedding vector $\tilde{x}_m$. For readability of the figure, we display $N_m = N$ as constant across each group. A second invariant module $\Sigma_I^{(2)}$ reduces all intermediary embedding vectors to a hierarchical embedding vector z, which gets passed through an inference network to arrive at the final vector $\hat{\pi}$ of approximated posterior model probabilities. See the online article for the color version of this figure.

take this probabilistic symmetry (e.g., exchangeability) into account in order to ensure the structural faithfulness of its approximations. Moreover, respecting the probabilistic symmetry implied by a generative model can not only make simulation-based training easier but also suggests a particular architecture for building neural Bayesian approximators.

Permutation Invariance

Permutation invariance is the functional equivalent of the probabilistic notion of exchangeability (Bloem-Reddy & Teh, 2020; Gelman, 2006), which roughly states that the order of random variables should not influence their joint probability.

To illustrate this point, consider the model in Equation 4, which has two exchangeable levels by design, indexed by $m \in \{1, \dots, M\}$ and $n \in \{1, \dots, N_m\}$. In a setting familiar to social scientists, we might have M individuals, each of whom provides N_m (multivariate) responses on some scale or in repeated trials of an experiment. Now, suppose that we want to compare a set of HMs $\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_J\}$ of the form given by Equation 4 that might differ in various ways (e.g., different prior/hyperprior assumptions or disparate likelihoods). Due to the structure of the models, the PMPs $p(\mathcal{M}|\{x_{mn}\})$ depend on neither the ordering of the individuals nor the ordering of their responses (which also holds true for the corresponding BFs).

More precisely, if $\mathbb{S}(\cdot)$ is an arbitrary permutation of an index set, then

$$p(\mathcal{M}|\{x_{mn}\}) = p(\mathcal{M}|\mathbb{S}(\{x_{mn}\})), \qquad (11)$$

for any $\mathbb{S}(\cdot)$ acting on $\{1, \dots, M\} \times \{1, \dots, N_1\} \times \cdots \times \{1, \dots, N_M\}$, where $\times$ denotes the Cartesian product of two (index) sets. Note that this notation implies that only permuting each m and

permuting each n within, but not across each group m is allowed. The property of permutation invariance is immediately obvious from the right-hand side of Equation 4 that involves two nested products (products being permutation invariant transformations when seen as functions operating on sets). Naturally, learning permutation invariance directly from data or simulations is hardly feasible with standard neural networks, even for nonnested data. Indeed, for nonhierarchical generative models, Radev, D'Alessandro, et al. (2021) proposed to use composite permutation invariant networks as employed by Zaheer et al. (2017). In the following section, we generalize this architectural concept to the hierarchical setting.

Hierarchical Invariant Neural Network Architecture

Permutation invariant networks differ from standard feedforward networks in that they can process inputs of different sizes and encode the probabilistic symmetry of the data directly (i.e., remove the need to learn the symmetry implicitly during training by supervised learning alone).

For the purpose of BMC with HMs, we realize a hierarchical permutation invariant function via a stack of invariant modules $\Sigma_I^{(l)}$ for each hierarchical level $l = 1, \dots, L$ of the Bayesian model (see Figure 2). Each invariant module performs an equivariant nonlinear transformation $h_1^{(l)}$ acting on the individual data points, followed by a pooling operator (e.g., sum or max) and a further nonlinear transformation $h_2^{(l)}$ acting on the pooled data.

In order to preserve hierarchical symmetry, we apply each $\Sigma_I^{(l)}$ independently to each nested sequence of data points. To make this point concrete, consider the two-level model given by Equation 4 and let data point x_{mn} denote the multivariate response

of person m in trial n of some data collection experiment. Accordingly, the first invariant module $\Sigma_I^{(1)}$ operates by reducing the trial data $\{x_n\}_m$ of each person m to a single person-vector $\tilde{x}_m$ of fixed size:

$$\tilde{x}_m = \Sigma_I^{(1)}(\{x_n\}_m) = h_2^{(1)}\left(\sum_{n=1}^{N_m} h_1^{(1)}(x_{mn})\right), \qquad (12)$$

where h_1 and h_2 are implemented as simple feedforward neural networks with trainable parameters suppressed for clarity. The second invariant module $\Sigma_I^{(2)}$ then compresses all person vectors to a final vector z of fixed size:

$$z = \Sigma_I^{(2)}(\{x_m\}) = h_2^{(2)}\left(\sum_{m=1}^{M} h_1^{(2)}(\tilde{x}_m)\right). \qquad (13)$$

In this way, the architecture becomes completely independent of the number of persons M or number of trials per person N_m, which could vary arbitrarily across persons. The vector z, whose dimensionality represents a tunable hyperparameter, can be interpreted as encoding learned summary statistics for the BMC task at hand (to be discussed shortly). Moreover, it is easy to see that z is independent of the ordering of persons or the ordering of trials within persons, as necessitated by the model formulation in Equation 4. Thus, the composition $\Sigma_I^{(2)} \bigcirc \Sigma_I^{(1)}(\{x_{mn}\})$ reduces a hierarchical data set with two levels to a single vector z which respects the probabilistic symmetry implied by the particular HM formulation.

Increasing the Capacity of Invariant Networks

Encoding an entire hierarchical data set $\{x_{mn}\}$ into a single vector z forces the composite neural network to perform massive data compression, creating a potential information bottleneck. For complex generative models, this task can become rather challenging and will depend highly on the representational capacity of the neural network (i.e., its ability to extract informative data set embeddings). Fortunately, we can enhance the simple architecture described in the preceding paragraph by using insights from Zaheer et al. (2017) and Bloem-Reddy and Teh (2020).

In order to increase the capacity of the previously introduced invariant transformation, we can stack together multiple equivariant modules $\Sigma_E^{(l)}$. Each equivariant module implements a combination of equivariant and invariant transformations. For instance, focusing on our two-level model example (Equation 4), the transformations at level 1 for each person m are now given by:

$$\tilde{x}_m = h_2^{(1)}\left(\sum_{n=1}^{N_m} h_1^{(1)}(x_{mn})\right), \qquad (14)$$

$$\tilde{x}_{mn} = h_3^{(1)}\left([x_{mn}, \tilde{x}_m]\right) \quad \text{for } n = 1, \ldots, N_m, \qquad (15)$$

where h_3 is also implemented as a simple feedforward neural network. In this way, each intermediary output $\tilde{x}_{mn}$ of the equivariant module now contains information from all data points, so the network can learn considerably more flexible transformations. Moreover, we can stack K equivariant modules followed by an invariant module, in order to obtain a deep invariant module,

which for the first hierarchical level ($l = 1$) takes the following form:

$$\tilde{x}_m = (\Sigma_I^{(1)} \bigcirc \Sigma_E^{(K,1)} \bigcirc \cdots \bigcirc \Sigma_E^{(1,1)})(\{x_n\}_m). \qquad (16)$$

Compared to the simple invariant module from Equation 12, the deep invariant module involves a larger number of computations but allows the network to learn more expressive representations. Accordingly, the transformation for the second hierarchical level ($l = 2$), which yields the final summary representation z, is given by:

$$z = (\Sigma_I^{(2)} \bigcirc \Sigma_E^{(K',2)} \bigcirc \cdots \bigcirc \Sigma_E^{(1,2)})(\{\tilde{x}_m\}), \qquad (17)$$

where the number of equivariant modules K' for level 2 can differ from the number of equivariant modules K for level 1. In our experiments, reported in the Experiments section, we observe a clear advantage of using deep invariant networks over their simple counterparts. Furthermore, for two-level models, we find that the performance of the networks is largely insensitive to the choice of K or K'.

Learning the Model Comparison Problem

In order to get from the learned summary representation z to an approximation of the analytic PMPs $\hat{\pi}$, we apply a final neural classifier (i.e., the inference network) $\mathcal{I}(z) = \hat{\pi}$, as visualized in Figure 2. We deviate from the Dirichlet-based setting by Radev, D'Alessandro, et al. (2021), since we found that implementing the inference network as a standard softmax classifier (Grathwohl et al., 2019) provides slightly better calibration and leads to more stable training in the specific context of HMs.

Denoting the entire hierarchical neural network as $f_\phi(\{x\}) = \hat{\pi}$ and an arbitrary hierarchical data set as $\{x\}$, we aim to minimize the expected logarithmic loss

$$\min_{\phi} \mathbb{E}_{p(\mathcal{M}, \{x\})}\left[-\sum_{j=1}^{J} \mathbb{I}_{\mathcal{M}_j} \cdot \log f_\phi(\{x\})_j\right], \qquad (18)$$

where ϕ represents the vector of trainable neural network parameters (e.g., weights and biases), $\mathbb{I}_{\mathcal{M}_j}$ is the indicator function for the "true" model. The expectation runs over the joint generative (mixture) distribution of all models $p(\mathcal{M}, \{x\})$, which we access through Monte Carlo simulations. Since the logarithmic loss is a strictly proper loss (Gneiting & Raftery, 2007), it drives the outputs of $f_\phi(\{x\})$ to estimate the actual PMPs $p(\mathcal{M}|\{x\})$ as best as possible. Thus, perfect convergence in theory guarantees that the network outputs the analytically correct PMPs which asymptotically select the "true" model in the closed world or the model that minimizes the Kullback-Leibler divergence to the "true" data generating process in the open world (Barron et al., 1999).

In practice, we approximate Equation 18 over a training set of B simulations from the competing HMs. Each entry b for $b = 1, \ldots, B$ in this training set represents a hierarchical data set $\{x^{(b)}\}$ itself along with a corresponding one-hot encoded vector for the "true" model index $\mathcal{M}_j^{(b)}$. The latter denotes the model from which the data set was generated and serves as the "ground truth" for supervised learning.

Similarly to Radev, D'Alessandro, et al. (2021), our neural method encodes an implicit preference for simpler HMs (i.e., Occam's razor) inherent in all marginal likelihood-based methods

(see MacKay, 2003, Chapter 28). Since our simulation-based training approximates an expectation over the marginal likelihoods of all HMs $p(\mathcal{M})p(\boldsymbol{x}|\mathcal{M})$, data sets generated by a simpler HM will tend to be more similar compared to those generated by a more complex one (cf. Figure 1). Thus, data sets that are plausible under both HMs will be generated more often by the simpler model than by the more complex model. A sufficiently expressive neural network will capture this behavior by assigning a higher PMP for the simpler model,[1] thereby capturing complexity differences arising directly from the generative behavior of the HMs.

Finally, to increase training efficiency when working under a limited simulation budget, we also explore a novel pretraining method inspired by transfer learning (Bengio et al., 2009; Torrey & Shavlik, 2010). First, we train the networks on data sets with a reduced number of exchangeable units (e.g., reducing the number of observations at level $l = 1$). This procedure accelerates training since it uses fewer simulator calls and the forward pass through the networks becomes cheaper. In the second step, we generate data with a realistic number of exchangeable units. Crucially, since we can use the pretrained network from step one as a better-than-random initialization, we need considerably fewer simulations than if we trained the network from scratch. Indeed, our Application: Hierarchical Evidence Accumulation Models section demonstrates the utility of this training method.

Experiments

In this section, we first conduct two simulation studies in which we extensively test the approximation performance of our hierarchical neural method. We start with a comparison of two nested toy HMs in the Validation Study 1: Hierarchical Normal Models section, followed by a comparison of two complex nonnested HMs of cognition in the Validation Study 2: Hierarchical SDT Versus MPT Models section. For both validation studies, we test our method internally by examining the calibration of the approximated PMPs. Additionally, we validate our method externally by benchmarking its performance against the current state-of-the-art for comparing HMs, namely, bridge sampling (Gelman & Meng, 1998; Gronau, Singmann, et al., 2017). To enable this challenging benchmark, we limit our validation studies to the comparison of models with explicit likelihoods to which bridge sampling is applicable.

Finally, in the Application: Hierarchical Evidence Accumulation Models section, we use our deep learning method to compare four hierarchical EAMs of response times data. Two of these models have no analytic likelihood, which makes the entire BMC setup intractable with current state-of-the-art methods (e.g., bridge sampling). Moreover, with this example, we also address the utility of a novel EAM, the Lévy flight model (Voss et al., 2019), that has previously been impossible to investigate directly using Bayesian HMs.

For all experiments, we assume uniform model priors $p(\mathcal{M}_j) = 1/J$. All computations are performed on a single-graphics processing unit machine with an NVIDIA RTX 3,070 graphics card and an AMD Ryzen 5 5600X processor. The reported computation times are measured as wall-clock times. Details on the implementation of our neural networks and the employed training procedures are provided in Appendix A. Code for reproducing all results from this article is freely available at https://github.com/bayesflow-org/Hierarchical-Model-Comparison. Additionally, our proposed method

is implemented in the BayesFlow Python library for amortized Bayesian workflows (Radev et al., 2023).

Validation Study 1: Hierarchical Normal Models

In this first experiment, we examine a simple and controllable model comparison setup to examine the behavior of our method under various conditions, before moving on to more complex scenarios. Inspired by Gronau (2021), we compare two hierarchical normal models $\mathcal{M}_1$ and $\mathcal{M}_2$ that share the same hierarchical structure

$$\tau^2 \sim \text{Normal}_+(0, 1), \tag{19}$$

$$\sigma^2 \sim \text{Normal}_+(0, 1), \tag{20}$$

$$\theta_m \sim \text{Normal}(\mu, \sqrt{\tau^2}) \quad \text{for } m = 1, \ldots, M, \tag{21}$$

$$x_{mn} \sim \text{Normal}(\theta_m, \sqrt{\sigma^2}) \quad \text{for } n = 1, \ldots, N_m, \tag{22}$$

with $\text{Normal}_+(\cdot)$ denoting a zero-truncated normal distribution. The models differ with respect to the parameter μ that describes the location of the individual-level parameters θ_m: whereas $\mathcal{M}_1$ assumes the location of θ_m to be fixed at 0, the more flexible $\mathcal{M}_2$ allows for μ to vary

$$\mathcal{M}_1\colon \mu = 0, \tag{23}$$

$$\mathcal{M}_2\colon \mu \sim \text{Normal}(0, 1). \tag{24}$$

Calibration

The most important properties of an approximate inference method are the trustworthiness of its results and, more pragmatically, whether we can diagnose the lack of trustworthiness in a given application. A useful proxy for trustworthiness is the calibration of a probabilistic classifier, which measures how closely the predicted probabilities of outcomes match their true underlying probabilities (Guo et al., 2017; Schad et al., 2023).

However, computing the calibration of a BMC procedure is hardly feasible in a nonamortized setting, since it involves applying the method to a large number of simulated data sets. For bridge sampling, for example, that would imply re-fitting the models via MCMC and running bridge sampling on at least hundreds, if not thousands of simulated data sets. The calibration of our networks, on the other hand, can be determined almost immediately after training due to their amortization property (Radev, D'Alessandro, et al., 2021).

In the following experiments, we assess the calibration of our networks visually (via calibration curves) and numerically (via a measure of calibration error). For generating a calibration curve (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005), we first sort the predicted PMPs $\hat{\pi}_j^{(s)}$ on S simulated data sets $s = 1, \ldots, S$, which we then partition into I equally spaced probability bins $i = 1, \ldots, I$ (we use $I = 15$ bins for all validation experiments). For each model j and each bin i containing a set $\mathcal{B}_{ij}$ of predicted model indices, we compute the mean prediction for the model (predicted probability [PP]) and the actual fraction of this model being

---

[1] Assuming equal prior model probabilities.

true (true probability[TP]) as follows:

$$PP(\mathcal{B}_{ij}) := \frac{1}{|\mathcal{B}_{ij}|} \sum_{b \in \mathcal{B}_{ij}} \widehat{\pi}_j^{(b)}, \tag{25}$$

$$TP(\mathcal{B}_{ij}) := \frac{1}{|\mathcal{B}_{ij}|} \sum_{b \in \mathcal{B}_{ij}} \mathbb{I}_{\mathcal{M}_j^{(b)}}, \tag{26}$$

where $\mathbb{I}$ again denotes the indicator function for the "true model." These two quantities varying over the bins form the X- and Y-axis of a calibration curve. A well-calibrated model comparison method with an agreement in each bin (as indicated by a diagonal line) thus yields approximations that reflect the true probabilities of the compared models (Guo et al., 2017). We further summarize this information via the expected calibration error (ECE; Naeini et al., 2015) as a single number bounded between 0 and 1, which we estimate by averaging the individual deviations between PP and TP in each bin:

$$\widehat{ECE}_j := \sum_{i=1}^{I} \frac{|\mathcal{B}_{ij}|}{S} \left| PP(\mathcal{B}_{ij}) - TP(\mathcal{B}_{ij}) \right|. \tag{27}$$

If follows from Equation 27 that a perfect ECE can be achieved by always predicting indifferent probabilities (e.g., $\widehat{\pi}_1 = \widehat{\pi}_2 = .5$ when comparing two models). We therefore complement our calibration assessment by measuring the accuracy of recovery, for which we dichotomize the predicted PMPs $\widehat{\pi}_j^{(s)}$ on S simulated data sets into one-versus-rest model predictions $\widehat{\mathcal{M}}_j^{(s)}$:

$$Acc_j := \frac{1}{S} \mathbb{I}_{\widehat{\mathcal{M}}_j^{(s)} = \mathcal{M}_j^{(s)}}. \tag{28}$$

Thus, in our BMC context, accuracy roughly is to ECE what sharpness is to posterior calibration in Bayesian parameter estimation (Bürkner et al., 2022; L. Clarté et al., 2022).[2]

Fixed Data Set Sizes. In the first calibration experiment, we examine the performance of our method for the most simple application case of learning a model comparison problem on a specific (fixed) data set size. Here, all data sets simulated for training and validating, the network consist of $M = 50$ groups and $N_m = 50$ observations for each group $m = 1, \ldots, M$.

We train the network for 10,000 backpropagation steps, taking 6 min. Subsequently, we calculate its calibration on 5,000 held-out validation data sets and repeat this process 25 times to obtain stable results with uncertainty quantification. Figure 3a depicts the resulting median calibration curve. Its close alignment to the dashed diagonal line (representing perfect calibration) indicates that the approximate PMPs are well-calibrated (median ECE over all repetitions of $\widehat{ECE} = 0.014$). The curve's coverage of the full range of PPs and the median accuracy of $\widehat{Acc} = 0.89$ confirm that the excellent calibration does not stem from indifferent predictions. The subsequent comparison of our method to bridge sampling suggests that this accuracy is indeed close to the upper bound imposed by the aleatoric uncertainty in the model-implied data.

Data Sets With Varying Numbers of Observations. We now train our hierarchical network to approximate BMC over a range of hierarchical data sets with varying number of observations within groups N_m. This amortization over observation sizes would provide a substantial efficiency gain if a researcher desires to compare HMs on multiple data sets with differing N_m, as only a single network

would have to be trained for all data sets.[3] In our validation setup, each simulated data set still consists of $M = 50$ groups, but now the number of observations within those groups varies in $N_m = 1, \ldots, 100$.

We train the network for 20,000 training steps, taking 13 min. At each training step, we draw the number of observations for the current batch of simulations from a discrete uniform distribution $N_m \sim \text{Uniform}_D(1, 100)$. For each N_m used during training, we evaluate the calibration 25 times on 5,000 held-out simulated validation data sets. This repetition procedure allows us to quantify the uncertainty of our ECE estimates.

Figure 3b plots the median ECE values for each observation size. The neural network achieves high calibration with a median ECE over all observation sizes (and repetitions) of $\widehat{ECE} = 0.012$. Moreover, the unsystematic pattern of the median curve and the homoscedastic variation between the observation sizes indicate that the network has learned the model comparison task equally well for all settings (with the ECE only rising slightly for the poorly identifiable $N_m = 1$ setting). Together, the low calibration error and the accurate model predictions (median accuracy $\widehat{Acc} = 0.88$) indicate that our method incurs no trade-off between calibration and accuracy. We additionally observe no bias towards a model in all but the smallest observation sizes (see Figure B1 for accuracy and bias examinations in all settings).

Data Sets with Varying Numbers of Groups and Observations. In the third calibration experiment, we test the ability of the network to learn a model comparison problem over a range of data sets with varying number of groups M and varying observations per group N_m. This training scheme allows for amortized model comparison on multiple data sets with different sizes, which can be especially useful for a priori sample size determination on simulated data. Additionally, the trained network can be stored and reused on future data sets with yet-unknown sample sizes. For this experiment, training and validation data sets are simulated with $M = 1, \ldots, 100$ groups and $N_m = 1, \ldots, 100$ observations, resulting in a vast variability of data set sizes between 1 up to 10,000 data points.

Given the complexity of the learning task, we now train the network for 40,000 training steps, taking 36 min. At each training step, we draw the number of groups and observations from discrete uniform distributions $M \sim \text{Uniform}_D(1, 100)$ and $N_m \sim \text{Uniform}_D(1, 100)$. We estimate calibration on 5,000 held-out simulations for each combination of M and N_m. As this implies simulating 50,000,000 data sets, we forego the repetition procedure employed in the previous experiments.

Figure 4 depicts the calibration and accuracy results for all combinations of M and N_m. We observe low ECEs for the vast majority of settings in Figure 4a (median ECE over all settings of $\widehat{ECE} = 0.013$). In other words, the trained network is capable of generating highly calibrated PMPs over a broad range of data set sizes. Moreover, the BMC results are sensitive to the number of nested observations N_m, but not to the number of groups M, in our experimental setups. The only systematic drop in calibration occurs for data sets containing just a few nested observations

---

[2] We focus on the accuracy since we use a uniform model prior $p(\mathcal{M})$, but other metrics of predictive performance, such as the logarithmic scoring rule, would have been expedient as well.

[3] Note that we refer to variability between data sets. We describe an approach for handling within data set variability of nested trials in the Application: Hierarchical Evidence Accumulation Models section.

Figure 3
Validation Study 1: Calibration Results for the First Two Experiments
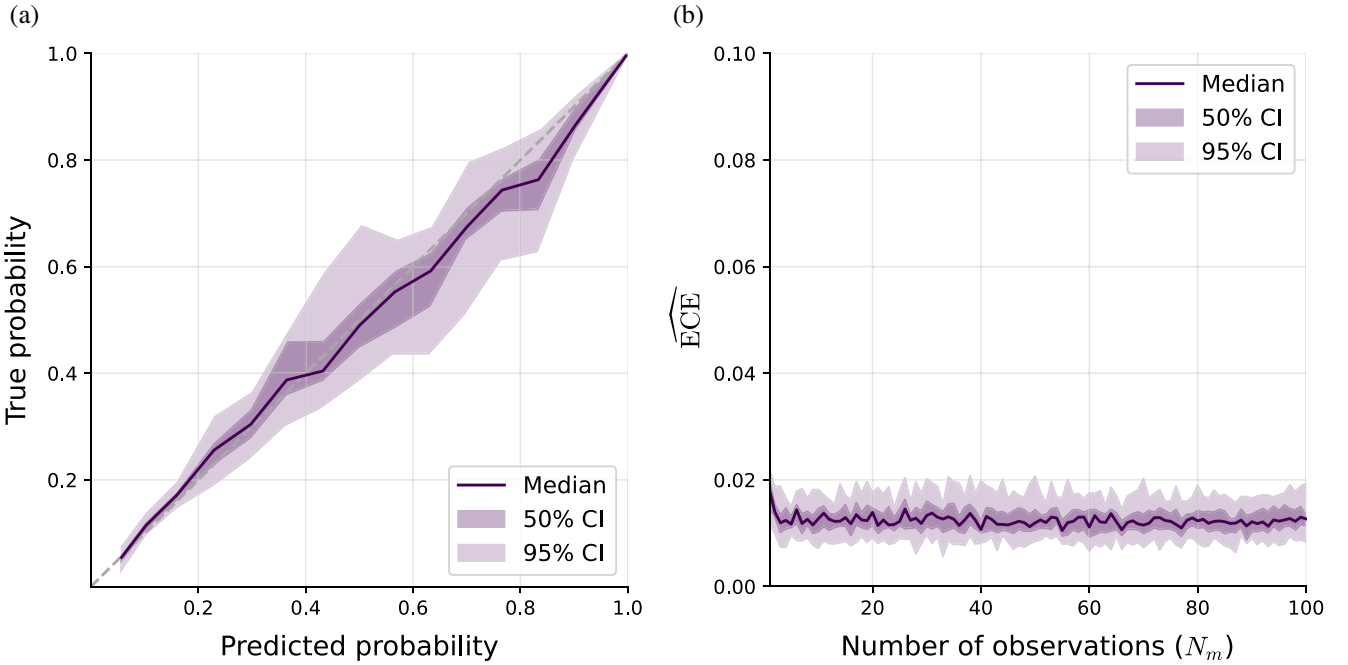
Note. (a) Results for the neural network trained on fixed data set sizes: Median calibration curve and CIs for data sets of $M = 50$ groups with $N_m = 50$ observations within each group. (b) Results for the neural network trained on data sets with the varying number of observations: Median ECEs and CIs for data sets of $M = 50$ groups with differing numbers of observations N_m within each group. Medians and CIs for all results are computed over 25 repetitions. CI = confidence interval; ECE = expected calibration error. See the online article for the color version of this figure.

($N_m \leq 5$). Considering that we observed better calibration for this low number of observations in a network trained on data sets with varying N_m (see Figure 3b), we surmise that the drop in the edge areas in Figure 4a arises from the challenging learning task over vastly different data set sizes (a phenomenon known as amortization gap; Cremer et al., 2018). The overall low (i.e., good) ECEs for all cases but the poorly identifiable $N_m = 1$ setting suggest that the networks' approximations are generally trustworthy. This is further confirmed in Figure 4b, where the observable accuracy pattern assures that this high calibration does not arise from a trade-off with predictive performance. Despite the demanding amortization setting, the network achieves an excellent median accuracy of $\widehat{\text{Acc}} = 0.88$, similar to the earlier experiments. We also find no indication of bias in any of the test settings except the $N_m = 1$ setting (see Figure B2). Marginal diagnostic plots for all metrics are provided in Figure B3.

Bridge Sampling Comparison

After validating the general trustworthiness of our method, we now benchmark it against the current gold standard for comparing HMs, namely, bridge sampling, as implemented by Gronau, Singmann, et al. (2017). As the nonamortized nature of bridge sampling restricts the feasible number of test sets, we conduct the benchmarking on 100 test sets which are simulated equally from $\mathcal{M}_1$ and $\mathcal{M}_2$. All simulated data sets consist of $M = 50$ groups and $N_m = 50$ observations per group. The fixed sample sizes of the test sets allow us to compare the two most distinct networks from the Calibration section to bridge sampling: the fixed network that is trained for this specific sample size

and the more complex variable network that is trained for amortized model comparison over variable sample sizes between $M = 1, \ldots, 100$ groups and $N_m = 1, \ldots, 100$ observations per group.

For bridge sampling, we first run four parallel MCMC chains with a warm-up period of 1,000 draws and 49,000 postwarm-up posterior draws per chain in Stan (Carpenter et al., 2017; Stan Development Team, 2019). We assess convergence through a visual inspection of the MCMC chains and an assessment of the $\widehat{R}$, bulk ESS, and tail ESS metrics (Vehtari et al., 2021). Afterward, we use the posterior draws to approximate PMPs and BFs with the bridgesampling R package (Gronau, Singmann, et al., 2017). We confirm the sufficiency of the total of 196,000 posterior draws by assessing the variability between multiple runs as by Schad et al. (2023), which yields highly similar results. Further insights via our calibration diagnostics are precluded by bridge sampling being a nonamortized method.

Approximation Performance. As we compare approximate PMPs, we can use a number of complementary metrics commonly employed to evaluate the quality of probabilistic predictions. First, we quantify the fraction of times the correct model $\mathcal{M}_j^{(s)}$ underlying a simulated data set s was detected, that is, the accuracy of recovery (see Equation 28). Second, we assess the mean absolute error (MAE) to investigate the average deviation of the approximated model probabilities $\widehat{\pi}_j^{(s)}$ from a perfect classification:

$$\text{MAE}_j := \frac{1}{S} \sum_{s=1}^{S} \left| \widehat{\pi}_j^{(s)} - \mathbb{I}_{\mathcal{M}_j}^{(s)} \right|. \tag{29}$$

Figure 4

Validation Study 1: Results for the Neural Network Trained and Tested Over Variable Data Set Sizes

(a)

(b)
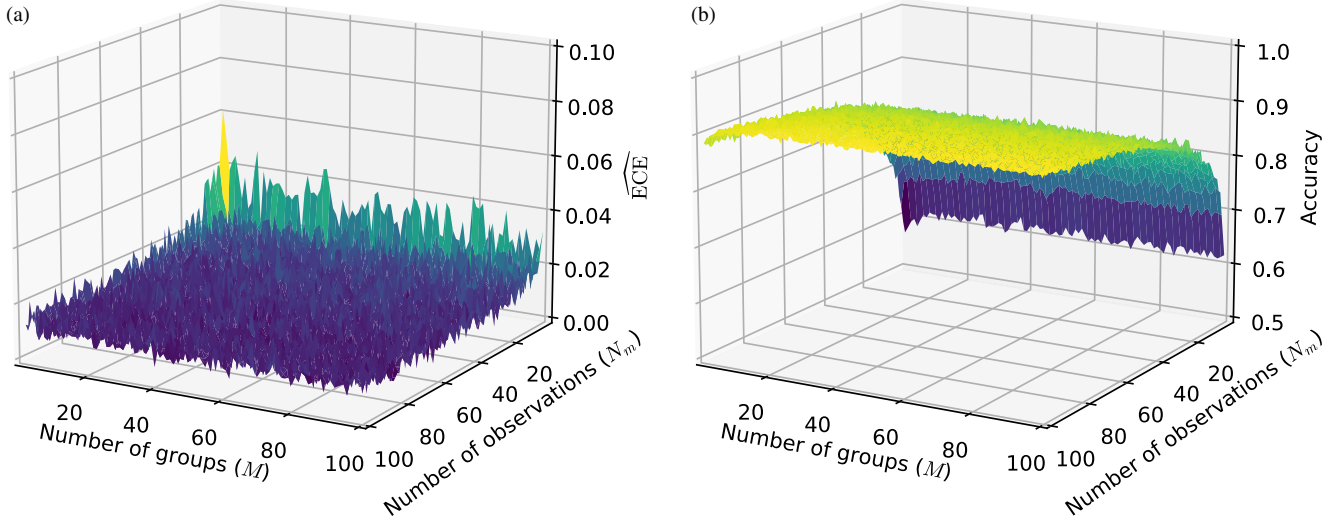
Note. (a) ECE and (b) accuracy of recovery. ECE = expected calibration error. See the online article for the color version of this figure.

Third, we measure the root-mean-square error (RMSE), which places particular emphasis on large prediction errors, to detect whether one method produces highly incorrect approximations more frequently than the other:

$$\text{RMSE}_j := \sqrt{\frac{1}{S}\sum_{s=1}^{S}\left(\widehat{\pi}_j^{(s)} - \mathbb{I}_{\mathcal{M}_j}^{(s)}\right)^2}. \quad (30)$$

Fourth, we calculate the Log-Score following the logarithmic scoring rule:

$$\text{LogScore}_j := -\frac{1}{S}\sum_{s=1}^{S}\left[\mathbb{I}_{\mathcal{M}_j}^{(s)} \cdot \log\widehat{\pi}_j^{(s)}\right]. \quad (31)$$

Its property as a strictly proper scoring rule implies that it is asymptotically minimized if and only if the approximate probabilities equal the true probabilities (Gneiting & Raftery, 2007). Lastly, we measure simulation-based calibration (SBC; Talts et al., 2018) as adapted by Schad et al. (2023) for model inference by the difference between the prior probability for a model and its average posterior probability in the test sets:

$$\text{SBC}_j := \text{p}(\mathcal{M}_j) - \frac{1}{S}\sum_{s=1}^{S}\widehat{\pi}_j^{(s)}. \quad (32)$$

We evaluate all metrics for $\mathcal{M}_2$ so that a bias towards $\mathcal{M}_1$ is indicated by positive SBC values and a bias towards $\mathcal{M}_2$ by negative SBC values.

Table 1 depicts the comparison results for our experimental setting. All metrics show equal performances for bridge sampling and the two neural network variants, with any differences being well within the range of the standard errors.

Approximation Convergence. In the following, we analyze the degree of convergence between the two methods at the level of individual data sets. We explore this visually by contrasting the PMP and (natural logarithmic) BF approximations of bridge sampling with the two neural network variants in Figure 5. We observe that the two methods' PMP approximations agree for the easy cases where the true underlying model is clearly classifiable. Thus, discrepancies between the two methods arise mainly for data sets with predicted PMPs close to $\widehat{\pi} = 0.5$. Even for the data sets with the largest discrepancies, the two methods do not map to qualitatively different decisions: $\widehat{\pi}_2^{\text{(bridge)}} = 0.67$ and $\widehat{\pi}_2^{\text{(neural)}} = 0.79$ for the fixed network, $\widehat{\pi}_2^{\text{(bridge)}} = 0.32$ and $\widehat{\pi}_2^{\text{(neural)}} = 0.25$ for the variable network. Most importantly, we detect no systematic pattern in these deviations.

As BFs represent the ratio of marginal likelihoods, they allow for a closer inspection of the degree of agreement between the methods in those edge cases with PMPs close to 0 or 1. We observe a close convergence for data sets classified as stemming from $\mathcal{M}_1$. Considering the predictions favoring $\mathcal{M}_2$, there are discrepancies for data sets with log BFs > 9.49. Since this corresponds to BFs >13,000 and PMPs > 0.9999, it is not visible in the PMP approximation plots. We obtain such extreme results only for $\mathcal{M}_2$, as this model allows for deviations of the group level parameters' location from 0 and enables the occurrence of extreme evidence in its favor. The divergence in this area of extreme evidence emerges most likely from the loss function employed for training the neural networks: the logarithmic loss obtained from a minuscule deviation of the PMP from 1 is near 0, which results in a negligible incentive for further optimization of the network's weights. We could reject a competing explanation based on limited floating-point precision, since training with an increased floating-point precision from 32-bit to 64-bit resulted in identical patterns. For visibility purposes, we exclude the 27 data sets for which bridge sampling approximated a BF > 1,000,000 for the BF plots in Figure 5, all continuing the observed plateau pattern. Plots with all 100 data sets are provided in Appendix B.

Table 1

Validation Study 1: Performance Metrics for the Comparison Between Hierarchical Normal Models

	Accuracy	MAE	RMSE	Log-Score	SBC
Bridge sampling	0.86 (0.03)	0.19 (0.03)	0.32 (0.03)	0.32 (0.06)	−0.02 (0.04)
Fixed network	0.84 (0.04)	0.19 (0.03)	0.32 (0.03)	0.32 (0.06)	−0.01 (0.04)
Variable network	0.86 (0.03)	0.19 (0.03)	0.32 (0.03)	0.31 (0.06)	−0.01 (0.04)

Note. Bootstrapped mean values and standard errors (in parentheses) are presented. We use 1,000 bootstrap versions of the test data sets and estimate the standard errors from the bootstrap standard deviations of the metrics. MAE = mean absolute error; RMSE = root-mean-square error; SBC = simulation-based calibration.

The divergence we encountered provides insights into the technical nature of our method but only arises in cases of extreme evidence. Thus, it is far from altering the substantive conclusions derived from the simulated BMC setting. Considering the convergence between the two methods in the realm of practical relevance, we can conclude that our method produces highly similar approximations to bridge sampling in this scenario.

Approximation Time. Both bridge sampling and our deep learning method can be divided into two computational phases. For bridge sampling, the first phase consists of drawing from the posterior parameter distributions (taking 52 s per data set on average). Bridge sampling itself takes place in the second phase (taking 38 s on average). Notably, in contrast to amortized inference with neural networks, both phases need to be repeated for each (simulated or observed) data set. Taking the initial compilation time of 42 s into account, bridge sampling consequently took 152 min for BMC on our 100 test data sets.

For the neural networks, the first phase (training) is resource-intensive (taking 6 min for the fixed network and 36 min for the variable network). The second phase (inference) is then performed in near real-time (inference on all 100 test data sets took 0.0004 s for the fixed network and 0.007 s for the variable network) and thus amortizes the training cost over multiple applications. For the simple HMs compared here, the amortization gains of our networks over bridge sampling come into effect after performing BMC on four (fixed network) or 24 (variable network) data sets.

We acknowledge our likely suboptimal choices of computational steps for the bridge sampling workflow or the neural networks and hence wish to stress the general patterns of nonamortized versus amortized methods demonstrated here. In general, we expect an advantage of bridge sampling in terms of efficiency in situations where only one or a few data sets are available and obtaining a large number of posterior draws is feasible. The demonstrated amortization property of our method might not be so relevant for inference on a single hierarchical data set, but it becomes crucial for performing calibration or recovery studies, which necessitate multiple refits of the same model (Schad et al., 2023).

Validation Study 2: Hierarchical SDT Versus MPT Models

We now extend our validation experiments from the simple setup with nested HMs to the comparison of nonnested HMs of cognition. In this simulation study, we examine the ability of our method to distinguish between data sets generated either from an HM based on a signal detection theory model (SDT model; Green & Swets, 1966) or

a hierarchical multinomial processing tree model (MPT model; Riefer & Batchelder, 1988). For illustrative purposes, we embed our simulation study within an old–new recognition scenario, where participants indicate whether or not a stimulus was previously presented to them.

We ensure a challenging model comparison setting via three design aspects: first, we specify both models to possess a similar generative behavior, that is, hardly distinguishable prior predictive distributions of hit rates and false alarm rates (prior predictive plots are provided in Appendix C). Second, data sets of old–new recognition typically contain low information as they only consist of binary variables indicating the stimulus type and response, respectively. Third, we further amplify the information sparsity of the data sets by choosing a particularly small size for all data sets of $M = 25$ simulated participants and $N_m = 50$ observations per participant.

A major difference between the compared cognitive model classes lies in the assumption of a continuous latent process by the SDT model and discrete processes (or states) by the MPT model. Our specification of the SDT model follows the hierarchical formulation of the standard equal-variance model by Rouder and Lu (2005). As the competing MPT model, we specify a hierarchical latent-trait two-high-threshold model (Klauer, 2010), which, in contrast to the SDT model, explicitly models correlations between its parameters. We follow the convention of restricting the parameters that describe the probability of recognizing a previously presented stimulus as old and a distractor stimulus as new to be equal, $D_O = D_N$, to render the MPT model identifiable (Erdfelder et al., 2009; Singmann & Kellen, 2013). Our prior choices for the parameters of both models are described in Appendix C.

We train the neural network for 50,000 training steps. As in the Validation Study 1: Hierarchical Normal Models section, we first leverage the amortization property of our method to inspect its calibration for the current model comparison task. Figure 6a shows that the trained neural network generates well-calibrated PMP approximations (median ECE over 25 repetitions of $\widehat{ECE} = 0.009$).

Next, we assess whether the observed calibration of the network translates into a competitive performance relative to bridge sampling. The benchmarking setup (50 simulated data sets from each model) and the implementation of the bridge sampling workflow follow the procedure described in the Bridge Sampling Comparison section.

The classification metrics depicted in Table 2 reveal the excellent performance of both methods, despite the challenging BMC scenario. We further observe a high degree of convergence between approximate PMPs derived by the two methods (cf. Figure 6b).

Figure 5

*Validation Study 1: Comparison of Approximation Results*
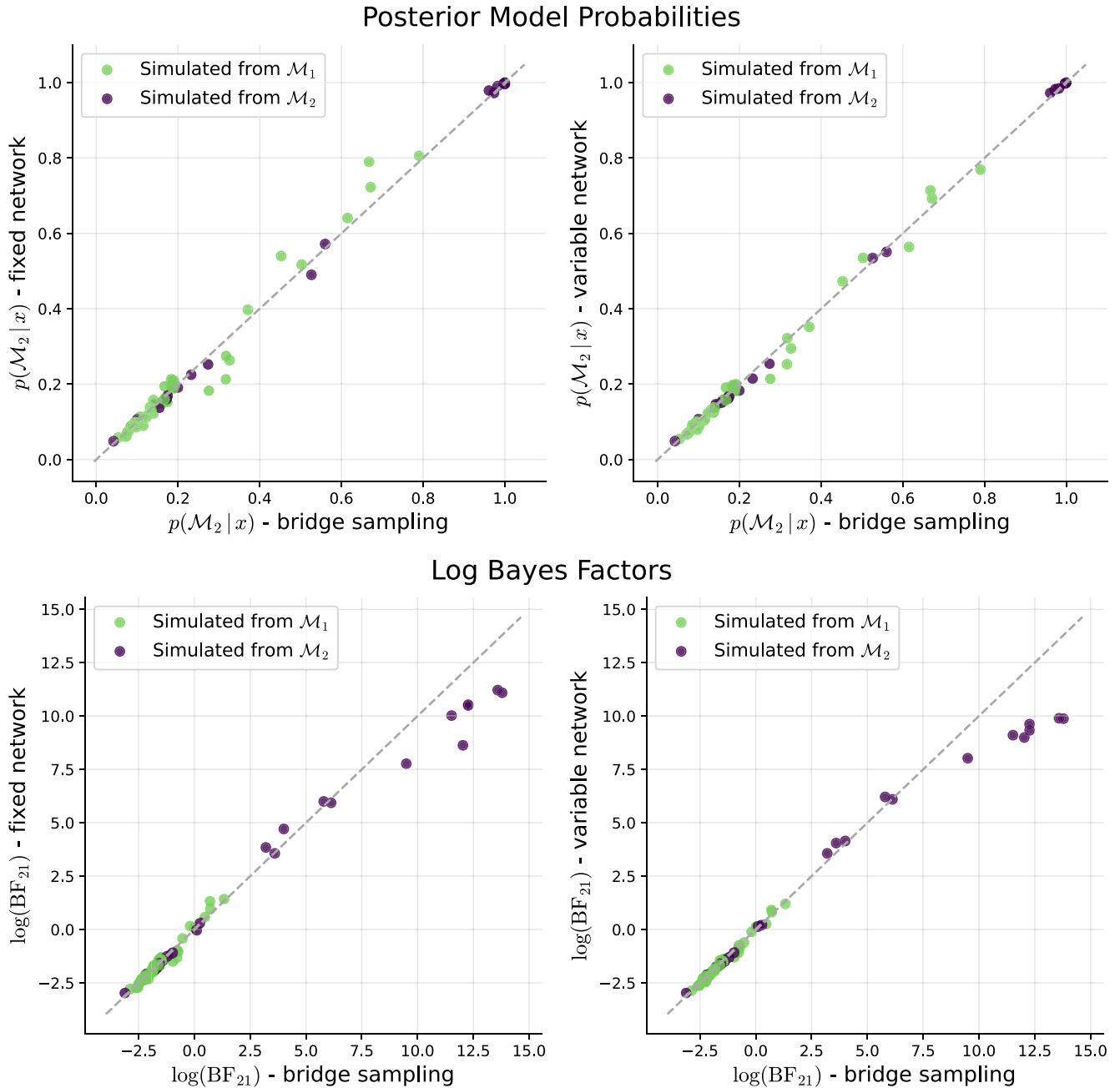
Note. Comparing bridge sampling versus the neural network trained on fixed data set sizes (left) and the neural network trained on variable data set sizes (right). For visibility purposes, the BF plots include only those 73 data sets for which bridge sampling approximated a $BF_{21} < 1,000,000$ (plots with all data sets are provided in Appendix B). BF = Bayes factor. See the online article for the color version of this figure.

Again, we find discrepancies between bridge sampling and our method in areas of extreme evidence (see Figure C2 for log BFs). As depicted in Figure 6c, obtaining PMP approximations for the 100 test data sets took more than 6 hr for bridge sampling and 55 min for the neural network. For this comparison of more complex cognitive models, the amortization advantage of our method emerges when analyzing 15 or more data sets. Note that this

advantage would quickly show up in validation studies involving multiple model refits (e.g., bootstrap, sensitivity analysis, or cross-validation).

All validation experiments so far have been set up in an $\mathcal{M}$-closed setting, with validation data simulated from the set $\mathcal{M}$ of models under consideration (Bernardo & Smith, 1994). Therefore, as a final validation, we test whether our method also behaves sensibly

Figure 6

Validation Study 2: Results for the Comparison Between Hierarchical SDT and MPT Models
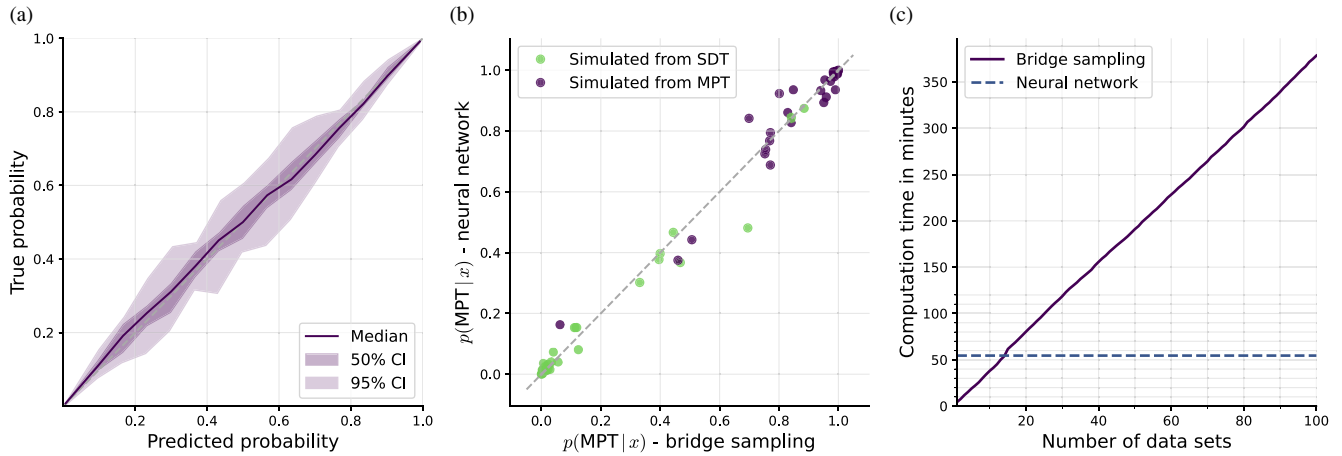
Note. (a) Calibration of the neural network over 25 repetitions with 5,000 data sets each, (b) convergence of approximate PMPs, and (c) computation times. PMP = posterior model probability; SDT = signal detection theory; MPT = multinomial processing tree; CIs = confidence intervals. See the online article for the color version of this figure.

in an $\mathcal{M}$-open setting, where none of the models generated the test data. For this, we simulate 100 noise data sets with the same hierarchical structure as before but generate the binary values for stimulus types and responses from a Bernoulli distribution with $p = 0.5$. Our neural method agrees with bridge sampling by assigning very high PMPs to the SDT model for all noise data sets ($\bar{\pi}_{SDT}^{(bridge)} = 0.999965$; $\bar{\pi}_{SDT}^{(neural)} = 0.999958$). Correspondingly, the deviations between both methods are minimal. We thus observe a close alignment between bridge sampling and our neural method in both a well-specified and a misspecified scenario. This tentative result suggests that our amortized estimates are faithful approximations not only in an $\mathcal{M}$-closed but also an $\mathcal{M}$-open setting, at least for this BMC scenario.

The converging results from the two validation studies demonstrate that our neural method generates well-calibrated and accurate PMP approximations. Despite our method only accessing the likelihood function indirectly via simulations, it can successfully compete with bridge sampling, which has direct access to the likelihood function.

Application: Hierarchical Evidence Accumulation Models

In the following, we showcase the utility of our method by comparing complex hierarchical EAMs in a real-data situation where

likelihood-based methods such as bridge sampling would not be applicable. More precisely, we seek to test the explanatory power of different stochastic diffusion model formulations proposed by Voss et al. (2019) for experimental response time data.

The so-called Lévy flight model increases the flexibility of the standard Wiener diffusion model (Ratcliff et al., 2016) but renders its likelihood function intractable with standard numerical approximations (Voss & Voss, 2007). The complete incorporation of all information through hierarchical modeling and the realization of BMC has consequently been infeasible so far. Thus, in a recent study, Wieschen et al. (2020) had to resort to a separate computation of the Bayesian information criterion (BIC) for each participant with subsequent aggregation. We aim to extend the study of Wieschen et al. (2020) by comparing fully hierarchical EAMs through PMPs and BFs. Moreover, we intend to answer the question formulated by Wieschen et al. (2020) as to whether the superior performance of the more complex models in their study stems from an insufficient punishment of model flexibility by the BIC. In addition to addressing a substantive research question in this application, we also demonstrate multiple advantages of our deep learning method on empirical data:

- Compare HMs with intractable likelihoods: As our method is simulation-based, including models with intractable likelihood functions in the comparison set does not alter its feasibility.

Table 2

Validation Study 2: Performance Metrics for the Comparison Between Hierarchical SDT and MPT Models

	Accuracy	MAE	RMSE	Log-Score	SBC
Bridge sampling	0.95 (0.02)	0.1 (0.02)	0.21 (0.04)	0.15 (0.04)	0.0 (0.04)
Neural network	0.95 (0.02)	0.09 (0.02)	0.21 (0.03)	0.14 (0.04)	0.0 (0.04)

Note. Bootstrapped mean values and standard errors (in parentheses) are presented. We use 1,000 bootstrap versions of the test data sets and estimate the standard errors from the bootstrap standard deviations of the metrics. MAE = mean absolute error; RMSE = root-mean-square error; SBC = simulation-based calibration.

- Adequately model nested data: Our method alleviates computational challenges that prevent modelers from adequately capturing the information contained in nested data structures through HMs.
- Reuse trained networks via fine-tuning: We accelerate the training of our neural network by pretraining it on less complex simulated data and subsequently fine-tuning it on simulated data resembling the actual experimental setting.
- Handle missing data: We train a neural network that can handle varying amounts of missing data by randomly masking simulated data during the training process.
- Validate a trained network on simulated data: The amortized nature of our method allows for extensive validation of a trained network prior to its application to empirical data.

Model Specification

For this application, we consider a Lévy flight model with nonGaussian noise (Voss et al., 2019). The Lévy flight process is driven by the following stochastic ordinary differential equation:

$$dx = v\,dt + \sigma\,d\xi \tag{33}$$

$$\xi \sim \text{AlphaStable}\left(\alpha, \mu = 0, \sigma = \frac{1}{\sqrt{2}}, \beta = 0\right), \tag{34}$$

which represents a Lévy walk characterized by a fat-tailed stable noise distribution.[4] In the above equation, x denotes the accumulated (perceptual) evidence, v denotes the rate of accumulation, and α controls the tail exponent of the noise variate ξ. Voss et al. (2019) and Wieschen et al. (2020) argued that the more abrupt changes in the information accumulation process that this model allows for could provide a better

description of human decision-making than a Gaussian noise. The addition of Lévy noise renders the standard numerical approximation of the diffusion model likelihood intractable (Voss & Voss, 2007). Consequently, neither standard MCMC nor bridge sampling are applicable for Bayesian parameter estimation and BMC, respectively.

There is an ongoing debate about the inclusion of additional parameters that account for intertrial variability in the diffusion model parameters: while they can provide a better model fit, the estimation of intertrial variability parameters is often difficult and can result in unstable results (Boehm et al., 2018; Lerche & Voss, 2016). Thus, Wieschen et al. (2020) also compared basic (without intertrial variability parameters) to full (with intertrial variability parameters) versions of the drift-diffusion and Lévy flight models.

Consequently, the set of candidate models considered here consists of four EAMs with increasing flexibility (i.e., the scope of possible data patterns that they can generate):

- $\mathcal{M}_1$, the most parsimonious basic diffusion model with the parameter v describing the mean rate of information uptake, the parameter a describing the threshold at which a decision is made, the parameter z_r describing a bias of the starting point towards one decision alternative and the parameter t_0 describing the nondecision time, that is, the time spent encoding the stimulus and executing the decision.

---

[4] An earlier version of this work used the original formulation by Voss et al. (2019), which sets $\sigma = 1$. For the special case of $\alpha = 2.0$, which is equivalent to the Wiener diffusion model, $\sigma = 1$ leads to an unusual diffusion constant (standard deviation of Gaussian noise) of $\sqrt{2}$, whereas $\sigma = \frac{1}{\sqrt{2}}$ ensures the conventional diffusion constant of 1. Notably, model comparison results are highly sensitive to the choice of σ.

Figure 7

Real-Data Application: Validation Results for the Evidence Accumulation Models on 2,000 Simulated Data Sets per Model
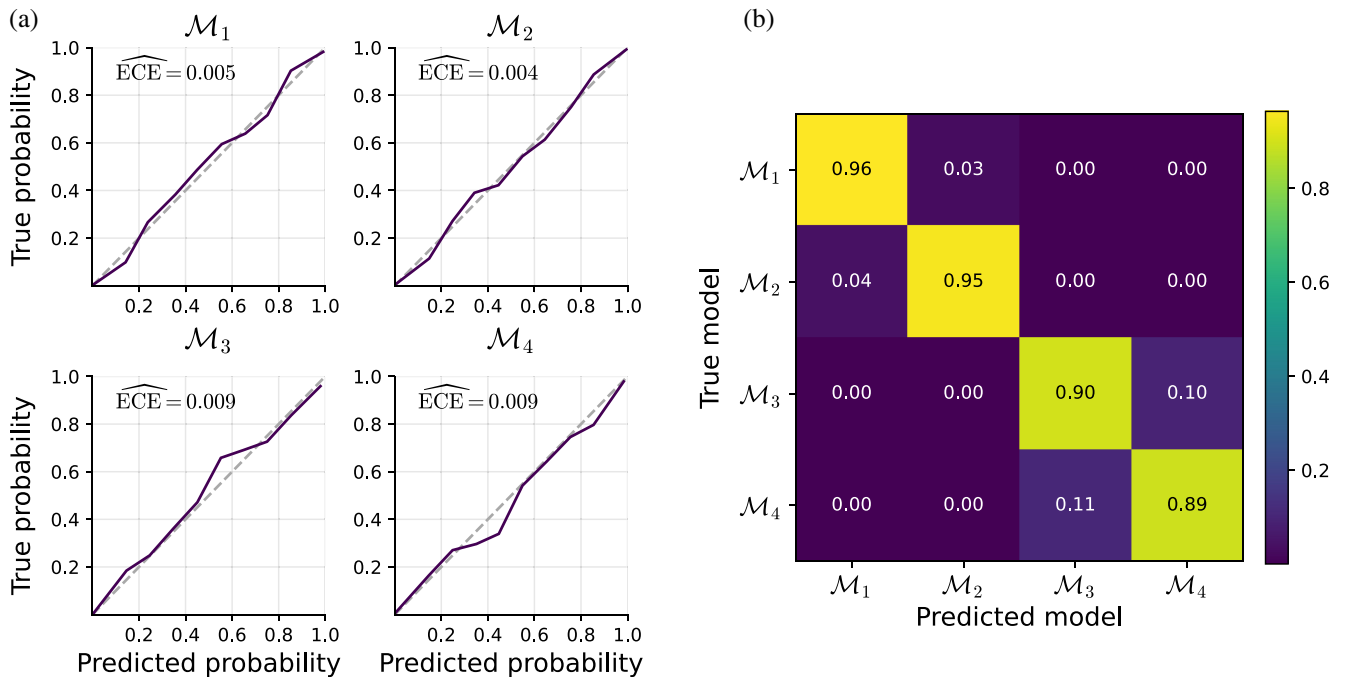
Note. (a) Calibration curves and (b) confusion matrix. ECE = expected calibration error. See the online article for the color version of this figure.

Table 3

Real-Data Application: BFs and PMPs Estimated From Data by Wieschen et al. (2020)

	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
BF_{j3}	9.63×10^{-5}	0.01	[a]	0.27
BF_{3j}	1.04×10^{4}	78	[a]	3.71
PMP	7.51×10^{-5}	1.00×10^{-2}	0.78	0.21

Note. BFs = Bayes factors; PMPs = posterior model probabilities.
[a] The preferred model.

- $\mathcal{M}_2$, the basic Lévy flight model, in which the assumption of a Wiener diffusion process with Gaussian noise is replaced by the above introduced Lévy flight process. The additional free parameter α governs the tail behavior of the noise distribution. The setting $\alpha = 2$ is equivalent to a Gaussian distribution, whereas $\alpha = 1$ reduces to a Cauchy distribution.
- $\mathcal{M}_3$, the full diffusion model, which extends $\mathcal{M}_1$ with the parameters s_{v_m}, s_{z_m}, and s_{t_m} that denote the variability (i.e., standard deviations) of drift rate, starting point bias, and nondecision time, respectively, between trials.
- $\mathcal{M}_4$, the full Lévy flight model, that possesses the largest flexibility by including intertrial variability parameters as well as the flexible Lévy noise distribution controlled by α.

Parameter priors and prior predictive checks are provided in Appendix D.

Data

The reanalyzed data set by Wieschen et al. (2020) contains 40 participants who completed a total of 900 trials of binary decision tasks (color discrimination and lexical decision) each. On average, 3.17% of trials per participant were excluded due to extremely short or long reaction times.

Simulation-Based Training

Since simulating data from EAMs can be challenging, especially when they include nonGaussian noise, we leverage the advantage that neural networks are capable of transfer learning as described in the Method section. Transfer learning describes the utilization of representations that had been previously learned by a neural network in a particular task for a new, related task (e.g., Ng et al., 2015). In this way, neural networks can be applied in small data settings (e.g., a limited simulation budget) by reusing the training knowledge encoded from structurally similar (possibly big data) settings.

For the purpose of model comparison, we first pretrain the network for 20 epochs (i.e., passes over the whole training data) on 10,000 simulated data sets per model. These data sets resemble the empirical data in that they consist of 40 simulated participants, but differ in that the number of trials is reduced by a factor of 9 (100 instead of 900 trials per participant). Afterwards, we fine-tune the network for additional 30 epochs on 2,000 simulated data sets per model that match the empirical data set with 40 simulated participants and 900 trials per participant. Thereby, we considerably reduce the computational demand of the training process. We further speed up the training phase by simulating all data prior to the training of the network in the high-performance programming language Julia (Bezanson et al., 2017). Pretraining took 10 min for the simulations and 11 min for training the networks. Fine-tuning took 18 min for the simulations and 16 min for training the networks, resulting in a total of 55 min for the training phase.

To fully adapt the network to the characteristics of the empirical data, we also simulate missing data during fine-tuning. In each training epoch, we generate a random binary mask f coding the simulated missing values. We sample the number of masked trials from a (discretized) normal distribution truncated between one and the number of trials, 900. The distributions' mean and standard deviation match the amount and variability of missing trials in the empirical data. We then perform an element-wise multiplication $\tilde{x} = x \otimes f$ and feed the "contaminated" data $\tilde{x}$ to the network. This procedure results in a robust network that can process various proportions of missing data. We find rank stability of our results in the presence of up to 25% missing data in Appendix D.

Results

Before applying our trained network to the empirical data, we validate it on 2,000 simulated data sets per model. First, the individual

Figure 8

Real-Data Application: Model Posteriors on the Empirical Data Set With Uncertainty Under Different Data Perturbations
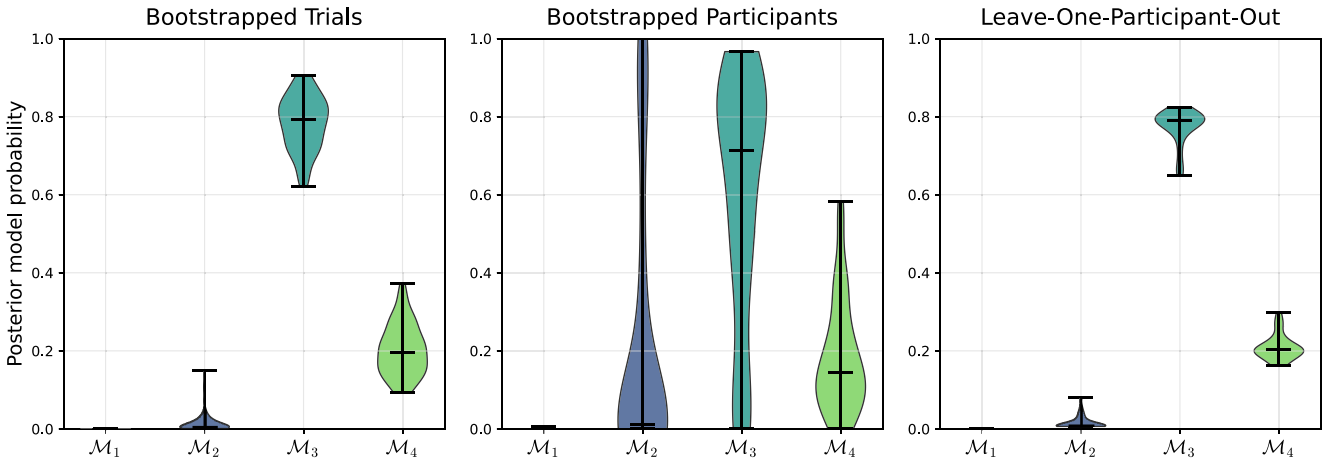
Note. We use 100 bootstrap samples for the bootstrapped results. See the online article for the color version of this figure.

calibration curves in Figure 7a show an excellent calibration for all models with $\widehat{\text{ECE}}$s close to 0. The calibration curves now consist of 10 instead of 15 intervals to obtain stable results despite the smaller amount of validation data sets per model. Second, we evaluate the accuracy of recovery and patterns of misclassification through the confusion matrix depicted in Figure 7b. The confusion matrix confirms that the excellent calibration of the network does not stem from chance performance. It also reveals that the selection of the "true" model becomes more difficult with increasing model complexity, which is a direct consequence of the Occam's razor property inherent in BMC (cf. Figure 1).

Table 3 presents the model comparison results on the empirical data set. Additionally, Figure 8 displays the model posteriors under different data perturbations. Consistent with the results of the nonhierarchical BIC approach by Wieschen et al. (2020), we find little evidence for both the basic diffusion model $\mathcal{M}_1$ and the basic Lévy flight model $\mathcal{M}_2$. This implies that the additional complexity of allowing parameters to vary between trials in $\mathcal{M}_3$ and $\mathcal{M}_4$ is, even under the strict penalization of prior-predictive flexibility in BMC, outweighed by better model fit. Also in agreement with Wieschen et al. (2020), we observe evidence for both $\mathcal{M}_3$ and $\mathcal{M}_4$, but, in contrast to Wieschen et al. (2020), our results slightly favor the full diffusion model $\mathcal{M}_3$ over the full Lévy flight model $\mathcal{M}_4$. Figure 8 confirms both the slight advantage of $\mathcal{M}_3$ over $\mathcal{M}_4$ and the substantial uncertainty associated with these results.

Discussion

Nested data are ubiquitous in the quantitative sciences, including psychological and cognitive research (Farrell & Lewandowsky, 2018). Yet, to avoid dealing with the complex dependencies resulting from these data, researchers often resort to simpler analyses, ignoring potentially important structural information. HMs provide a flexible way to represent the multilevel structure of nested data, but this flexibility can make BMC a daunting undertaking.

In this work, we proposed a powerful remedy to this problem: building on the BayesFlow framework (Radev et al., 2020), we developed a neural network architecture that enables approximate BMC for arbitrarily complex HMs. In two simulation studies, we showed that our deep learning method is well-calibrated and performs as accurately as bridge sampling, which is the current state-of-the-art for comparing HMs with simple likelihoods. Moreover, in a subsequent real-data application, we compared the relatively new Lévy flight model with existing evidence accumulation models. Thus, we argue that our method is well-suited to enhance the applicability of (complex) HMs in psychological research. Below, we summarize the key properties and limitations of our method while also outlining future research directions.

Amortized Inference

Our method offloads the computational demands for comparing HMs onto the training phase of a custom neural network, allowing for near real-time model comparison using the trained network. The resulting amortization offers several advantages over nonamortized methods.

First, it enables thorough validation of a trained network on thousands of simulated data sets, allowing large-scale simulation-based

diagnostics to become an integral part of the BMC workflow (Gelman et al., 2020; Schad et al., 2023). Second, the trained and validated networks can be used not only for point estimates of BFs or PMPs on empirical data but also for exploring the robustness of the results against multiple data perturbations, as showcased in our real-data application.

Third, we demonstrated the feasibility of amortizing over variable data set sizes in our first validation study. This is particularly advantageous in the context of HMs since nested data sets often contain multiple exchangeable levels with variable sizes (e.g., different number of clusters, participants, and observations). Analyzing multiple hierarchical data sets with variable sizes only requires a single network that has seen different data set sizes during training. The same network could also be used for various simulation studies, such as the challenging task of designing maximally informative experiments in a hierarchical BMC setting (Heck & Erdfelder, 2019; Myung & Pitt, 2009).

Lastly, we showed that researchers do not even need to consider all possible shapes of future data sets when training such a network, as they can use transfer learning to efficiently adapt a trained network to a related setting. Beyond allowing more flexibility in reusing networks across experiments, researchers or even fields, transfer learning can also considerably reduce the computational demands associated with comparing complex HMs. As demonstrated in our real-data application, a network can be pretrained on simulated data sets with reduced size and fine-tuned afterwards on sizes matching the empirical data.

Independence From Explicit Likelihoods

Unlike other popular methods for performing BMC on HMs, such as the Savage–Dickey density ratio or bridge sampling, our method is not constrained by the availability of an explicit likelihood function for all competing models. As long as the models in question can be implemented as simulators, the neural network can be trained to perform BMC on these models. The value of such a method is evident, as it decouples the substantive task of model specification from concerns about the feasibility of estimation methods.

Statistical models are instantiations of substantive knowledge or hypotheses. As such, we argue that model specification should not be unduly restricted by considerations of computational tractability—a sentiment that is closely related to what Haaf et al. (2021) call the "specification-first-principle." Our proposed deep learning method satisfies this principle, as model specification may be guided exclusively by substantive arguments with few concerns about tractability. Thus, we believe that our method makes a contribution to the recent upsurge of innovative psychological models (Collins & Shenhav, 2022; Ghaderi-Kangavari et al., 2023; Heathcote & Matzke, 2022) by allowing for an efficient assessment of their incremental value in a hierarchical setting.

Limitations and Outlook

One of the main challenges of approximate methods and, more broadly, statistical inference, is ensuring the faithfulness of the obtained results. The outlined possibilities for validating amortized model comparison and examining the robustness of the results are important contributions of our method, but they come with open

questions. Concerning the validation of the network, framing model comparison as a supervised learning problem allows us to draw on the rich literature on classification performance and calibration metrics. Nevertheless, determining a "good enough" score for an approximate BMC method remains challenging, as the optimally possible performance is application-specific and usually unknown.

Concerning the application of the network to empirical data, we showed in Validation Study 2 that our method produces, at least in this scenario, reasonable results when confronted with data not stemming from the models under consideration. Moreover, our robustness checks are a practical proxy for measuring the reliability of BMC results in a closed-world setting. However, these checks cannot possibly capture the (lack of) absolute evidence for an HM: as a relative method, BMC may indicate that one model fits the data better than a set of competing models, but it does not provide any measure of how well (or poorly) the model itself approximates the underlying data-generating process. Additionally, it has been shown that severe model misspecification, with fundamental gaps between simulated and empirical data (e.g., parameter priors that exclude essential regions), can lead to unreliable simulation-based inference (Cranmer et al., 2020; Frazier & Drovandi, 2021). A promising direction to address this limitation could be the combination of our method with the recently proposed meta-uncertainty framework for BMC (Schmitt et al., 2022), which can be greatly accelerated with amortized deep learning methods. This combination could provide a principled delineation of different uncertainty sources, enabling the detection of model misspecification cases where none of the competing HMs can explain the observed data. Still, further research is needed to determine whether meta-uncertainty can provide reliable evidence for the open versus closed world assumption in the context of HMs.

Since BMC is a marginal likelihood (i.e., prior predictive) approach, priors over model parameters should be informed by scientific theory and will thus have a decisive influence on the results (Vanpaemel, 2010). We do not intend to re-iterate the ongoing discussion about this property of BMC (Gronau & Wagenmakers, 2019a,2019b; Haaf et al., 2021; Vehtari et al., 2019), but want to highlight a specific difficulty that arises for HMs: Parameter priors of an HM are connected via multilevel dependencies, increasing the risk that poor prior choices lead to nonintended model behavior (for a recent discussion of this problem in cognitive modeling, see Sarafoglou et al., 2022). Therefore, prior predictive checks and prior sensitivity analyses become especially important when conducting BMC on competing HMs. While transfer learning reduces the computational demands of retraining a neural network for sensitivity analyses, another avenue for future research would be the amortization over different prior choices, enabling immediate prior sensitivity assessment.

Finally, it should be noted that the version of our method explored here can only compare HMs assuming exchangeable data at each hierarchical level. Although the majority of HMs in social science research follow this probabilistic symmetry, some researchers may want to compare nonexchangeable HMs, for example, to study within-person dynamics (Driver & Voelkle, 2018; Lodewyckx et al., 2011; Schumacher et al., 2022). Fortunately, the modularity of our method allows easy adaptation of the neural network architecture to handle nonexchangeable HMs. To compare hierarchical time-series models with temporal dependencies at the lowest level, for instance, the first invariant module could be exchanged for a recurrent network, as proposed by Radev, D'Alessandro et al. (2021) for nonhierarchical models. Thus, future research could extend and validate our method in BMC settings involving nonexchangeable HMs.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., & Devin, M. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. arXiv preprint arXiv:1603.04467.

Barron, A., Schervish, M. J., & Wasserman, L. (1999). The consistency of posterior distributions in nonparametric problems. *The Annals of Statistics*, 27(2), 536–561. https://doi.org/10.1214/aos/1018031206

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41(1), 379–406. https://doi.org/10.1146/ecolsys.2010.41.issue-1

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). *Curriculum learning* [Conference session]. In Proceedings of the 26th Annual International Conference on Machine Learning.

Bennett, C. H. (1976). Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22(2), 245–268. https://doi.org/10.1016/0021-9991(76)90078-4

Bernardo, J. M., & Smith, A. F. (1994). *Bayesian theory*. John Wiley & Sons.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98. https://doi.org/10.1137/141000671

Bloem-Reddy, B., & Teh, Y. W. (2020). Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(1), 1–61.

Boehm, U., Annis, J., Frank, M. J., Hawkins, G. E., Heathcote, A., Kellen, D., Krypotos, A.-M., Lerche, V., Logan, G. D., Palmeri, T. J., van Ravenzwaaij, D., Servant, M., Singmann, H., Starns, J. J., Voss, A., Wiecki, T. V., Matzke, D., & Wagenmakers, E.-J. (2018). Estimating across-trial variability parameters of the diffusion decision model: Expert advice and recommendations. *Journal of Mathematical Psychology*, 87, 46–75. https://doi.org/10.1016/j.jmp.2018.09.004

Bürkner, P. C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bürkner, P. C., Scholz, M., & Radev, S. T. (2022). *Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy*. arXiv preprint arXiv:2209.02439.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 1–32. https://doi.org/10.18637/jss.v076.i01

Clarté, G., Robert, C. P., Ryder, R. J., & Stoehr, J. (2021). Componentwise approximate Bayesian computation via Gibbs-like steps. *Biometrika*, 108(3), 591–607. https://doi.org/10.1093/biomet/asaa090

Clarté, L., Loureiro, B., Krzakala, F., & Zdeborová, L. (2022). *A study of uncertainty quantification in overparametrized high-dimensional models*. arXiv preprint arXiv:2210.12760.

Collins, A. G., & Shenhav, A. (2022). Advances in modeling learning and decision-making in neuroscience. *Neuropsychopharmacology*, 47(1), 104–118. https://doi.org/10.1038/s41386-021-01126-y

Congdon, P. (2006). Bayesian model choice based on Monte Carlo estimates of posterior model probabilities. *Computational Statistics & Data Analysis*, 50(2), 346–357. https://doi.org/10.1016/j.csda.2004.08.001

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48), 30055–30062. https://doi.org/10.1073/pnas.1912789117

Cremer, C., Li, X., & Duvenaud, D. (2018). *Inference suboptimality in variational autoencoders* [Conference session]. In Proceedings of the 35th International Conference on Machine Learning.

Csilléry, K., Blum, M. G., Gaggiotti, O. E., & François, O. (2010). Approximate Bayesian computation (ABC) in practice. *Trends in Ecology & Evolution*, *25*(7), 410–418. https://doi.org/10.1016/j.tree.2010.04.001

DeGroot, M. H., & Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *32*(1–2), 12–22. https://doi.org/10.2307/2987588

Dickey, J. M., & Lientz, B. (1970). The weighted likelihood ratio, sharp hypotheses about chances, the order of a Markov chain. *The Annals of Mathematical Statistics*, *41*(1), 214–226. https://doi.org/10.1214/aoms/1177697203

Driver, C. C., & Voelkle, M. C. (2018). Hierarchical Bayesian continuous time dynamic modeling. *Psychological Methods*, *23*(4), 774–799. https://doi.org/10.1037/met0000168

Erdfelder, E., Auer, T. S., Hilbig, B. E., Aßfalg, A., Moshagen, M., & Nadarevic, L. (2009). Multinomial processing tree models: A review of the literature. *Zeitschrift für Psychologie/Journal of Psychology*, *217*(3), 108–124. https://doi.org/10.1027/0044-3409.217.3.108

Farrell, S., & Lewandowsky, S. (2018). *Computational modeling of cognition and behavior*. Cambridge University Press.

Fengler, A., Govindarajan, L. N., Chen, T., & Frank, M. J. (2021). Likelihood approximation networks (LANs) for fast inference of simulation models in cognitive neuroscience. *eLife*, *10*, Article e65074. https://doi.org/10.7554/eLife.65074

Frazier, D. T., & Drovandi, C. (2021). Robust approximate Bayesian inference with synthetic likelihood. *Journal of Computational and Graphical Statistics*, *30*(4), 958–976. https://doi.org/10.1080/10618600.2021.1875839

Gelman, A. (2006). Multilevel (hierarchical) modeling: What it can and cannot do. *Technometrics*, *48*(3), 432–435. https://doi.org/10.1198/004017005000000661

Gelman, A., & Meng, X. L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 163–185. https://www.jstor.org/stable/2676756

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020). *Bayesian workflow*. arXiv preprint arXiv:2011.01808.

Ghaderi-Kangavari, A., Rad, J. A., & Nunez, M. D. (2023). A general integrative neurocognitive modeling framework to jointly describe EEG and decision-making on single trials. *Computational Brain & Behavior*, 6, 317–376. https://doi.org/10.1007/s42113-023-00167-4

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, *102*(477), 359–378. https://doi.org/10.1198/016214506000001437

Gonçalves, P. J., Lueckmann, J. M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P., & Greenberg, D. S. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, *9*, Article e56261. https://doi.org/10.7554/eLife.56261

Grathwohl, W., Wang, K. C., Jacobsen, J. H., Duvenaud, D., Norouzi, M., & Swersky, K. (2019). *Your classifier is secretly an energy based model and you should treat it like one*. arXiv preprint arXiv:1912.03263.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics* (Vol. 1). Wiley.

Gronau, Q. F. (2021). *Hierarchical normal example (Stan)*. https://cran.csiro.au/web/packages/bridgesampling/vignettes/bridgesampling_example_stan.html

Gronau, Q. F., Heathcote, A., & Matzke, D. (2020). Computing Bayes factors for evidence-accumulation models using Warp-III bridge sampling. *Behavior Research Methods*, *52*(2), 918–937. https://doi.org/10.3758/s13428-019-01290-6

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J., & Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology*, *81*, 80–97. https://doi.org/10.1016/j.jmp.2017.09.005

Gronau, Q. F., Singmann, H., & Wagenmakers, E. J. (2017). *Bridgesampling: An R package for estimating normalizing constants*. arXiv preprint arXiv:1710.08162.

Gronau, Q. F., & Wagenmakers, E. J. (2019a). Limitations of Bayesian leave-one-out cross-validation for model selection. *Computational Brain & Behavior*, *2*(1), 1–11. https://doi.org/10.1007/s42113-018-0011-7

Gronau, Q. F., & Wagenmakers, E. J. (2019b). Rejoinder: More limitations of Bayesian leave-one-out cross-validation. *Computational Brain & Behavior*, *2*(1), 35–47. https://doi.org/10.1007/s42113-018-0022-4

Gronau, Q. F., Wagenmakers, E. J., Heck, D. W., & Matzke, D. (2019). A simple method for comparing complex models: Bayesian model comparison for hierarchical multinomial processing tree models using Warp-III bridge sampling. *Psychometrika*, *84*(1), 261–284. https://doi.org/10.1007/s11336-018-9648-3

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). *On calibration of modern neural networks* [Conference session]. In Proceedings of the 34th International Conference on Machine Learning.

Haaf, J. M., Klaassen, F., & Rouder, J. N. (2021). *Bayes factor vs. posterior-predictive model assessment: Insights from ordinal constraints*. PsyArXiv preprint.

Haaf, J. M., & Rouder, J. N. (2017). Developing constraint in Bayesian mixed models. *Psychological Methods*, *22*(4), 779–798. https://doi.org/10.1037/met0000156

Heathcote, A., & Matzke, D. (2022). Winner takes all! What are race models, and why and how should psychologists use them? *Current Directions in Psychological Science*, *31*(5), 383–394. https://doi.org/10.1177/09637214221095852

Heck, D. W., Boehm, U., Böing-Messing, F., Bürkner, P. -C., Derks, K., Dienes, Z., Fu, Q., Gu, X., Karimova, D., Kiers, H. A., Klugkist, I., Kuiper, R. M., Lee, M. D., Leenders, R., Leplaa, H. J., Linde, M., Ly, A., Meijerink-Bosman, M., Moerbeek, M., … Palfi, B. (2022). A review of applications of the Bayes factor in psychological research. *Psychological Methods*, *28*(3), 558–579. https://doi.org/10.1037/met0000454

Heck, D. W., & Erdfelder, E. (2019). Maximizing the expected information gain of cognitive modeling via design optimization. *Computational Brain & Behavior*, *2*(3), 202–209. https://doi.org/10.1007/s42113-019-00035-0

Hermans, J., Banik, N., Weniger, C., Bertone, G., & Louppe, G. (2021). Towards constraining warm dark matter with stellar streams through neural simulation-based inference. *Monthly Notices of the Royal Astronomical Society*, *507*(2), 1999–2011. https://doi.org/10.1093/mnras/stab2181

Hinton, S. R., Davis, T., Kim, A. G., Brout, D., D'Andrea, C. B., Kessler, R., Lasker, J., Lidman, C., Macaulay, E., Möller, A., & Sako, M. (2019). Steve: A hierarchical Bayesian model for supernova cosmology. *The Astrophysical Journal*, *876*(1), Article 15. https://doi.org/10.3847/1538-4357/ab13a3

Hox, J. J., Moerbeek, M., & Van de Schoot, R. (2017). *Multilevel analysis: Techniques and applications*. Routledge.

Jalilian, A., & Mateu, J. (2021). A hierarchical spatio-temporal model to analyze relative risk variations of COVID-19: A focus on Spain, Italy and Germany. *Stochastic Environmental Research and Risk Assessment*, *35*(4), 797–812. https://doi.org/10.1007/s00477-021-02003-2

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*(430), 773–795. https://doi.org/10.1080/01621459.1995.10476572

Kingma, D. P., & Ba, J. L. (2015). *Adam: A method for stochastic optimization* [Conference session]. In 3rd International Conference on Learning Representations.

Klauer, K. C. (2010). Hierarchical multinomial processing tree models: A latent-trait approach. *Psychometrika*, *75*(1), 70–98. https://doi.org/10.1007/s11336-009-9141-0

Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical Bayesian models. *Journal of Mathematical Psychology*, *55*(1), 1–7. https://doi.org/10.1016/j.jmp.2010.08.013

Lerche, V., & Voss, A. (2016). Model complexity in diffusion modeling: Benefits of making the model more parsimonious. *Frontiers in Psychology*, 7, Article 1324. https://doi.org/10.3389/fpsyg.2016.01324

Lodewyckx, T., Tuerlinckx, F., Kuppens, P., Allen, N. B., & Sheeber, L. (2011). A hierarchical state space approach to affective dynamics. *Journal of Mathematical Psychology*, 55(1), 68–83. https://doi.org/10.1016/j.jmp.2010.08.004

Lotfi, S., Izmailov, P., Benton, G., Goldblum, M., & Wilson, A. G. (2022). *Bayesian model selection, the marginal likelihood, and generalization*. arXiv preprint arXiv:2202.11678.

MacKay, D. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.

Marin, J. M., Pudlo, P., Estoup, A., & Robert, C. (2018). *Likelihood-free model choice*. Chapman & Hall/CRC Press.

Marjoram, P., Molitor, J., Plagnol, V., & Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26), 15324–15328. https://doi.org/10.1073/pnas.0306899100

McElreath, R. (2020). *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman & Hall/CRC Press.

Meng, X. L., & Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3), 552–586. https://doi.org/10.1198/106186002457

Meng, X. L., & Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 6(4), 831–860.

Mertens, U. K., Voss, A., & Radev, S. (2018). ABrox—A user-friendly Python module for approximate Bayesian computation with a focus on model comparison. *PLoS ONE*, 13(3), Article e0193981. https://doi.org/10.1371/journal.pone.0193981

Mestdagh, M., Verdonck, S., Meers, K., Loossens, T., & Tuerlinckx, F. (2019). Prepaid parameter estimation without likelihoods. *PLoS Computational Biology*, 15(9), Article e1007181. https://doi.org/10.1371/journal.pcbi.1007181

Myung, J. I., & Pitt, M. A. (2009). Optimal experimental design for model discrimination. *Psychological Review*, 116(3), 499–518. https://doi.org/10.1037/a0016104

Naeini, M. P., Cooper, G., & Hauskrecht, M. (2015). *Obtaining well calibrated probabilities using Bayesian binning*. In Proceedings of the twenty-ninth AAAI conference on artificial intelligence (pp. 2901–2907). AAAI Press.

Ng, H. W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015). *Deep learning for emotion recognition on small datasets using transfer learning* [Conference session]. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction.

Nicenboim, B., Schad, D. J., & Vasishth, S. (2022). *An introduction to Bayesian data analysis for cognitive science*. https://vasishth.github.io/bayescogsci/book/.

Niculescu-Mizil, A., & Caruana, R. (2005). *Predicting good probabilities with supervised learning*. In Proceedings of the 22nd international conference on machine learning (pp. 625–632).

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1), 99–118.

Pudlo, P., Marin, J. M., Estoup, A., Cornuet, J. M., Gautier, M., & Robert, C. P. (2016). Reliable ABC model choice via random forests. *Bioinformatics*, 32(6), 859–866. https://doi.org/10.1093/bioinformatics/btv684

Radev, S. T., D'Alessandro, M., Mertens, U. K., Voss, A., Köthe, U., & Bürkner, P. C. (2021). Amortized Bayesian model comparison with evidential deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8), 4903–4917. https://doi.org/10.1109/TNNLS.2021.3124052

Radev, S. T., Graw, F., Chen, S., Mutters, N. T., Eichel, V. M., Bärnighausen, T., & Köthe, U. (2021). OutbreakFlow: Model-based Bayesian inference of disease outbreak dynamics with invertible neural networks and its application to the COVID-19 pandemics in Germany. *PLoS Computational Biology*, 17(10), Article e1009472. https://doi.org/10.1371/journal.pcbi.1009472

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4), 1452–1466. https://doi.org/10.1109/TNNLS.2020.3042395

Radev, S. T., Schmitt, M., Schumacher, L., Elsemüller, L., Pratz, V., Schälte, Y., Köthe, U., & Bürkner, P. C. (2023). *BayesFlow: Amortized Bayesian workflows with neural networks*. arXiv preprint arXiv:2306.16015.

Ratcliff, R., Smith, P. L., Brown, S. D., & McKoon, G. (2016). Diffusion decision model: Current issues and history. *Trends in Cognitive Sciences*, 20(4), 260–281. https://doi.org/10.1016/j.tics.2016.01.007

Riefer, D. M., & Batchelder, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, 95(3), 318–339. https://doi.org/10.1037/0033-295X.95.3.318

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, 12(4), 573–604. https://doi.org/10.3758/BF03196750

Rouder, J. N., & Morey, R. D. (2012). Default Bayes factors for model selection in regression. *Multivariate Behavioral Research*, 47(6), 877–903. https://doi.org/10.1080/00273171.2012.734737

Rouder, J. N., Morey, R. D., & Pratte, M. S. (2017). Bayesian hierarchical models of cognition. In W. H. Batchelder, H. Colonius, E. N. Dzhafarov, & J. Myung (Eds.), *New handbook of mathematical psychology: Foundations and methodology* (pp. 504–551). Cambridge University Press.

Sarafoglou, A., Kuhlmann, B. G., Aust, F., & Haaf, J. M. (2022). *Theory-informed refinement of Bayesian hierarchical MPT modeling*. PsyArXiv preprint.

Schad, D. J., Betancourt, M., & Vasishth, S. (2021). Toward a principled Bayesian workflow in cognitive science. *Psychological Methods*, 26(1), 103–126. https://doi.org/10.1037/met0000275

Schad, D. J., Nicenboim, B., Bürkner, P. C., Betancourt, M., & Vasishth, S. (2023). Workflow techniques for the robust use of Bayes factors. *Psychological Methods*, 28(6), 1404–1426. https://doi.org/10.1037/met0000472

Schmitt, M., Radev, S. T., & Bürkner, P. C. (2022). *Meta-uncertainty in Bayesian model comparison*. arXiv preprint arXiv:2210.07278.

Schumacher, L., Bürkner, P. C., Voss, A., Köthe, U., & Radev, S. T. (2022). *Neural superstatistics: A Bayesian method for estimating dynamic models of cognition*. arXiv preprint arXiv:2211.13165.

Singmann, H., & Kellen, D. (2013). MPTinR: Analysis of multinomial processing tree models in R. *Behavior Research Methods*, 45(2), 560–575. https://doi.org/10.3758/s13428-012-0259-0

Singmann, H., & Kellen, D. (2019). An introduction to mixed models for experimental psychology. In D. H. Spieler & E. Schumacher (Eds.), *New methods in cognitive psychology* (pp. 4–31). Routledge.

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6), 1760–1765. https://doi.org/10.1073/pnas.0607208104

Stan Development Team. (2019). *Stan modeling language users guide and reference manual* (Version 2.21.0). https://mc-stan.org

Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., & Dessimoz, C. (2013). Approximate Bayesian computation. *PLoS Computational Biology*, 9(1), Article e1002803. https://doi.org/10.1371/journal.pcbi.1002803

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). *Validating Bayesian inference algorithms with simulation-based calibration*. arXiv preprint arXiv:1804.06788.

Tieleman, T., & Hinton, G. (2012). *Lecture 6.5-rmsprop, coursera: Neural networks for machine learning* (Technical Report 6). University of Toronto.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264). IGI global.

Tran, N. H., van Maanen, L., Heathcote, A., & Matzke, D. (2021). Systematic parameter reviews in cognitive modeling: Towards a robust and

cumulative characterization of psychological processes in the diffusion decision model. *Frontiers in Psychology*, *11*, Article 608287. https://doi.org/10.3389/fpsyg.2020.608287

Turner, B. M., & Sederberg, P. B. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic Bulletin & Review*, *21*(2), 227–250. https://doi.org/10.3758/s13423-013-0530-0

Turner, B. M., & Van Zandt, T. (2014). Hierarchical approximate Bayesian computation. *Psychometrika*, *79*(2), 185–209. https://doi.org/10.1007/s11336-013-9381-x

Ulitzsch, E., von Davier, M., & Pohl, S. (2020). A multiprocess item response model for not-reached items due to time limits and quitting. *Educational and Psychological Measurement*, *80*(3), 522–547. https://doi.org/10.1177/0013164419878241

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response times. *Psychological Methods*, *16*(1), 44–62. https://doi.org/10.1037/a0021765

Vanpaemel, W. (2010). Prior sensitivity in theory testing: An apologia for the Bayes factor. *Journal of Mathematical Psychology*, *54*(6), 491–498. https://doi.org/10.1016/j.jmp.2010.07.003

Van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability: A guide to classical and parameterized complexity analysis*. Cambridge University Press.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P. C. (2021). Rank-normalization, folding, and localization: an improved R̂ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, *16*(2), 667–718. https://doi.org/10.1214/20-BA1221

Vehtari, A., Simpson, D. P., Yao, Y., & Gelman, A. (2019). Limitations of "Limitations of Bayesian leave-one-out cross-validation for model selection". *Computational Brain & Behavior*, *2*(1), 22–27. https://doi.org/10.1007/s42113-018-0020-6

Voss, A., Lerche, V., Mertens, U., & Voss, J. (2019). Sequential sampling models with variable boundaries and non-normal noise: A comparison of six models. *Psychonomic Bulletin & Review*, *26*(3), 813–832. https://doi.org/10.3758/s13423-018-1560-4

Voss, A., & Voss, J. (2007). Fast-dm: A free program for efficient diffusion model analysis. *Behavior Research Methods*, *39*(4), 767–775. https://doi.org/10.3758/BF03192967

Wagenmakers, E. J., Lodewyckx, T., Kuriyal, H., & Grasman, R. (2010). Bayesian hypothesis testing for psychologists: A tutorial on the Savage–Dickey method. *Cognitive Psychology*, *60*(3), 158–189. https://doi.org/10.1016/j.cogpsych.2009.12.001

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the drift-diffusion model in Python. *Frontiers in Neuroinformatics*, *7*, Article 14. https://doi.org/10.3389/fninf.2013.00014

Wieschen, E. M., Voss, A., & Radev, S. (2020). Jumping to conclusion? A Lévy flight model of decision making. *The Quantitative Methods for Psychology*, *16*(2), 120–132. https://doi.org/10.20982/tqmp.16.2.p120

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. *Advances in Neural Information Processing Systems*, *30*. https://papers.nips.cc/paper_files/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html

Appendix A

Neural Network Implementation and Training

The neural networks are implemented in the Python library TensorFlow (Abadi et al., 2015) and jointly optimized via backpropagation. During training, we use mini-batch gradient descent with batches of size $B = 32$ per backpropagation update (training step). We employ the Adam optimizer (Kingma & Ba, 2015) with a cosine decay schedule in Validation Study 1 (initial learning rate of 5×10^{-4}) and the real-data application (initial learning rate of 5×10^{-4} for pretraining and 5×10^{-5} for fine-tuning). In Validation Study 2, we use the RMSprop optimizer (Tieleman & Hinton, 2012) with an initial learning rate of 2.5×10^{-4} and a cosine decay schedule, which we found to work better for the unusually sparse binary data. In all validation studies, we use online training, that is, simulate new training data sets flexibly right before each training step. In the real-data application, we simulate all data sets efficiently a priori in the Julia programing language and therefore use offline training, that is, training with a predetermined amount of data sets.

We use the following neural network architectures: the hierarchical summary network is composed of two deep invariant modules, each consisting of $K = 2$ equivariant modules followed by an invariant module. The inference network is realized via a standard feedforward network with three fully connected layers followed by a softmax output layer. We did not conduct a thorough search for optimal hyperparameter settings of the neural networks and the training process.

Appendix B

Validation Study 1 Details

Calibration

Additional results for the scenario containing data sets with varying number of observations are depicted in Figure B1. Accuracy and SBC (median of $\widehat{\text{SBC}} = -0.0006$) are stable across nearly all settings, only slightly dropping for data sets with few observations.

Concerning the scenario containing data sets with varying number of groups and nested observations, Figure B2 presents generally unbiased SBC results with a median of $\widehat{\text{SBC}} = 0.0004$. Figure B3 shows the marginal plots corresponding to the three-dimensional plots for all metrics.

Bridge Sampling Comparison

Figure B4 displays the log BFs approximated by bridge sampling and the neural network variants for all 100 test data sets, including those 27 data sets for which bridge sampling approximated a BF > 1,000,000 and that were therefore excluded in Figure 5 for visibility purposes.
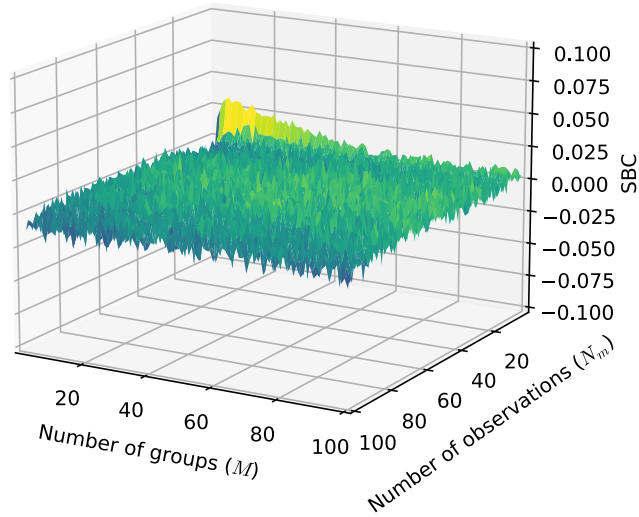
Figure B1

Validation Study 1: Additional Results for the Neural Network Trained and Tested on Data Sets With Varying Number of Observations

(a)

(b)
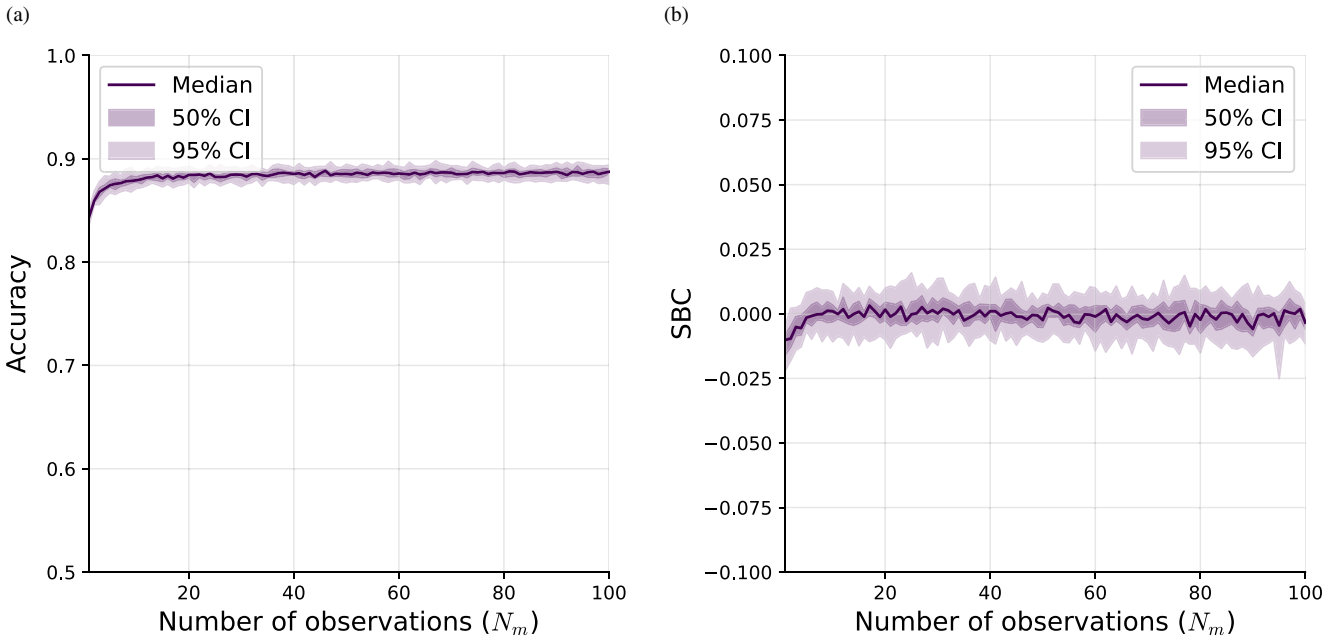
Note. (a) Accuracy of recovery and (b) SBC. SBC = simulation-based calibration; CI = confidence interval. See the online article for the color version of this figure.

Figure B2

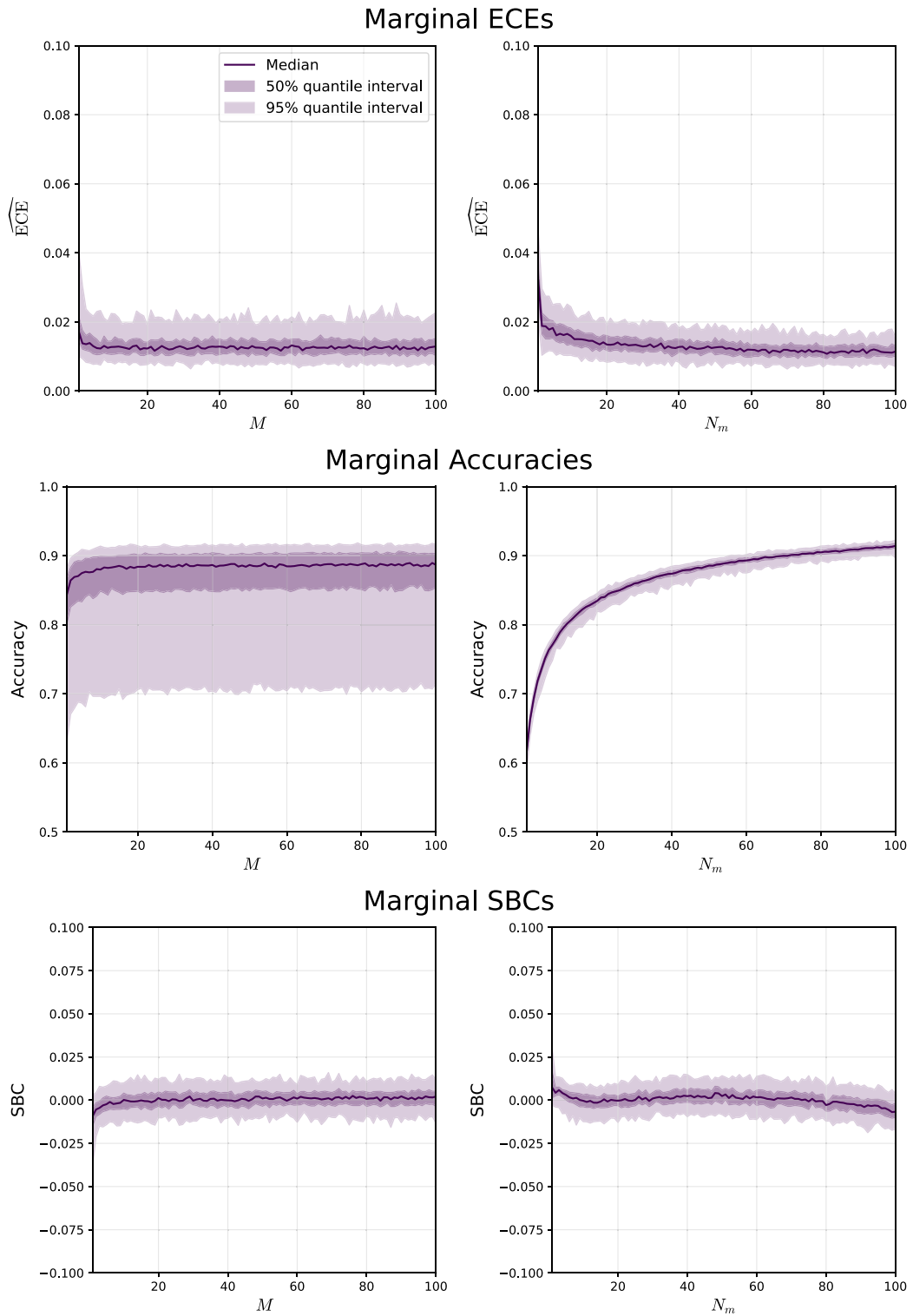*Validation Study 1: SBC Results for the Neural Network Trained and Tested Over Variable Data Set Sizes*

Note. SBC = simulation-based calibration. See the online article for the color version of this figure.

Figure B3

Validation Study 1: Marginal Plots for the Neural Network Trained and Tested Over Variable Data Set Sizes
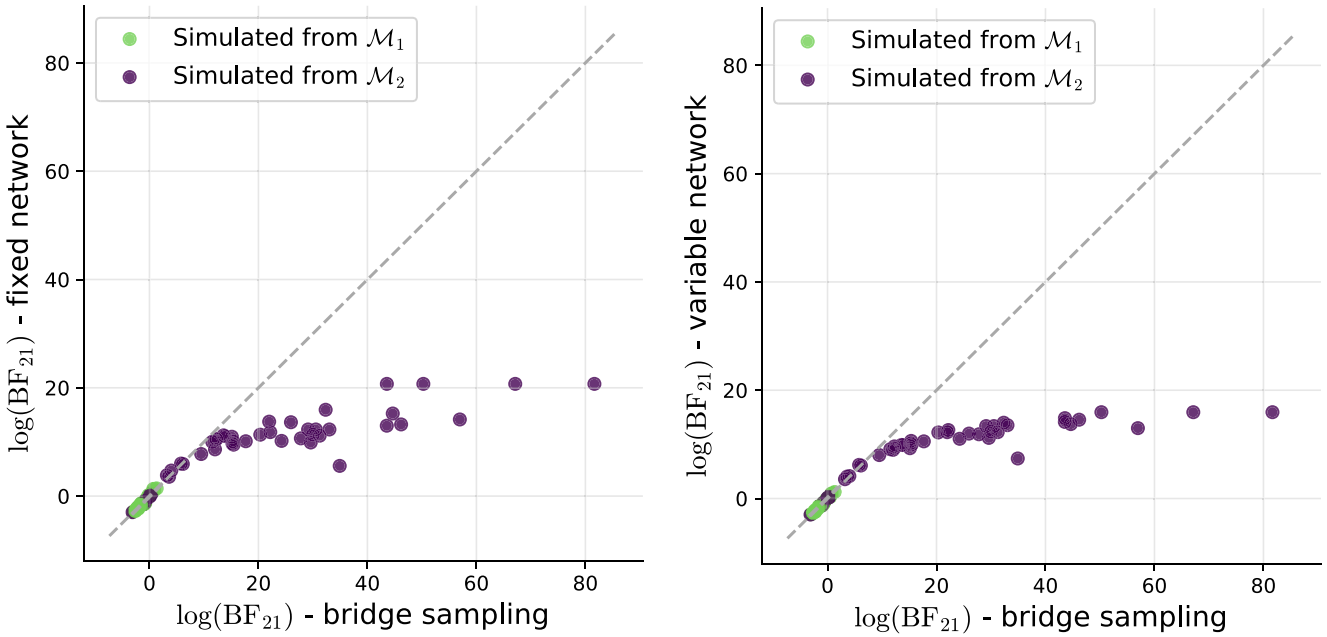

Note. ECE = expected calibration error; SBC = simulation-based calibration. See the online article for the color version of this figure.

(*Appendices continue*)

Figure B4

Validation Study 1: Full Comparison Results for the Log BFs (All 100 Test Data Sets)

Note. BFs = Bayes factors. See the online article for the color version of this figure.

Appendix C

Validation Study 2 Details

Here, we provide details on our model specifications and prior choices. We reformulate the observation-level structure of the MPT model as a binomial instead of a multinomial process to obtain identical response generation implementations for both models

$$x_{mn}^h \sim \text{Bernoulli}(h_m) \quad \text{for } n = 1, \ldots, N_m, \tag{C1}$$

$$x_{mn}^f \sim \text{Bernoulli}(f_m) \quad \text{for } n = 1, \ldots, N_m, \tag{C2}$$

where h_m denotes the probability of detecting an old item as old ("hit") and f_m denotes the probability of detecting a new item as

old ("false alarm"). The generating processes of these probabilities with our distributional choices are described in Tables C1 and C2 for the SDT model and Tables C3 and C4 for the MPT models. Figure C1 shows the prior predictive patterns of hit rates and false alarm rates arising from 5,000 simulated data sets for each model.

Figure C2 presents the log BFs approximated by bridge sampling and the neural network, showing slight discrepancies in areas of extreme evidence. In contrast to the nested models in Validation Study 1, the SDT and MPT models being nonnested allows for extreme evidence for both models.

Table C1

Validation Study 2: Hyperprior Distributions of the SDT Model

Parameter	Symbol	Prior distribution
Probit-transformed hit probability	$\mu_{h'}$	Normal(1, 0.5)
	$\sigma_{h'}$	Gamma(1, 1)
Probit-transformed false alarm probability	$\mu_{f'}$	Normal(−1, 0.5)
	$\sigma_{f'}$	Gamma(1, 1)

Note. SDT = signal detection theory.

Table C2

Validation Study 2: Group-Level Prior Distributions and Transformations of the SDT Model

Parameter	Symbol	Prior distribution/transformation
Probit-transformed hit probability	h'_m	Normal($\mu_{h'}$, $\sigma_{h'}$)
Probit-transformed false alarm probability	f'_m	Normal($\mu_{f'}$, $\sigma_{f'}$)
Hit probability	h_m	$\Phi(h'_m)$
False alarm probability	f_m	$\Phi(f'_m)$

Note. SDT = signal detection theory.

Table C3

Validation Study 2: Hyperprior Distributions and Transformations of the MPT Model

| Parameter	Symbol	Prior distribution/transformation
Probit-transformed recognition probability	$h_{d'}$	Normal(0, 0.25)
Probit-transformed guessing probability	$h_{g'}$	Normal(0, 0.25)
	$\lambda_{d'}$	Uniform(0, 2)
	$\lambda_{g'}$	Uniform(0, 2)
Covariance matrix	Q	InvWishart(3, $\mathbb{I}$)
	Σ	Diag($\lambda_{d'}$, $\lambda_{g'}$) Q Diag($\lambda_{d'}$, $\lambda_{g'}$)

Note. MPT = multinomial processing tree.

Table C4

Validation Study 2: Group-Level Prior Distributions and Transformations of the MPT Model

Parameter	Symbol	Prior distribution/transformation
Probit-transformed recognition probability	d'_m	Normal$\left(\begin{bmatrix} \mu_{d'} \\ \mu_{g'} \end{bmatrix}, \Sigma\right)$
Probit-transformed guessing probability	g'_m	
Recognition probability	d_m	$\Phi(d'_m)$
Guessing probability	g_m	$\Phi(g'_m)$
Hit probability	h_m	$d_m + (1 - d_m)*g_m$
False alarm probability	f_m	$(1 - d_m)*g_m$

Note. MPT = multinomial processing tree.

Figure C1

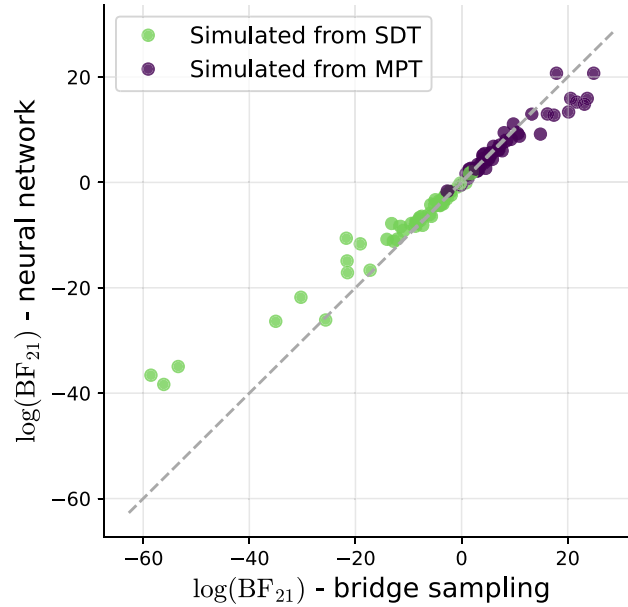*Validation Study 2: Prior Predictive Checks for the SDT and the MPT Model*
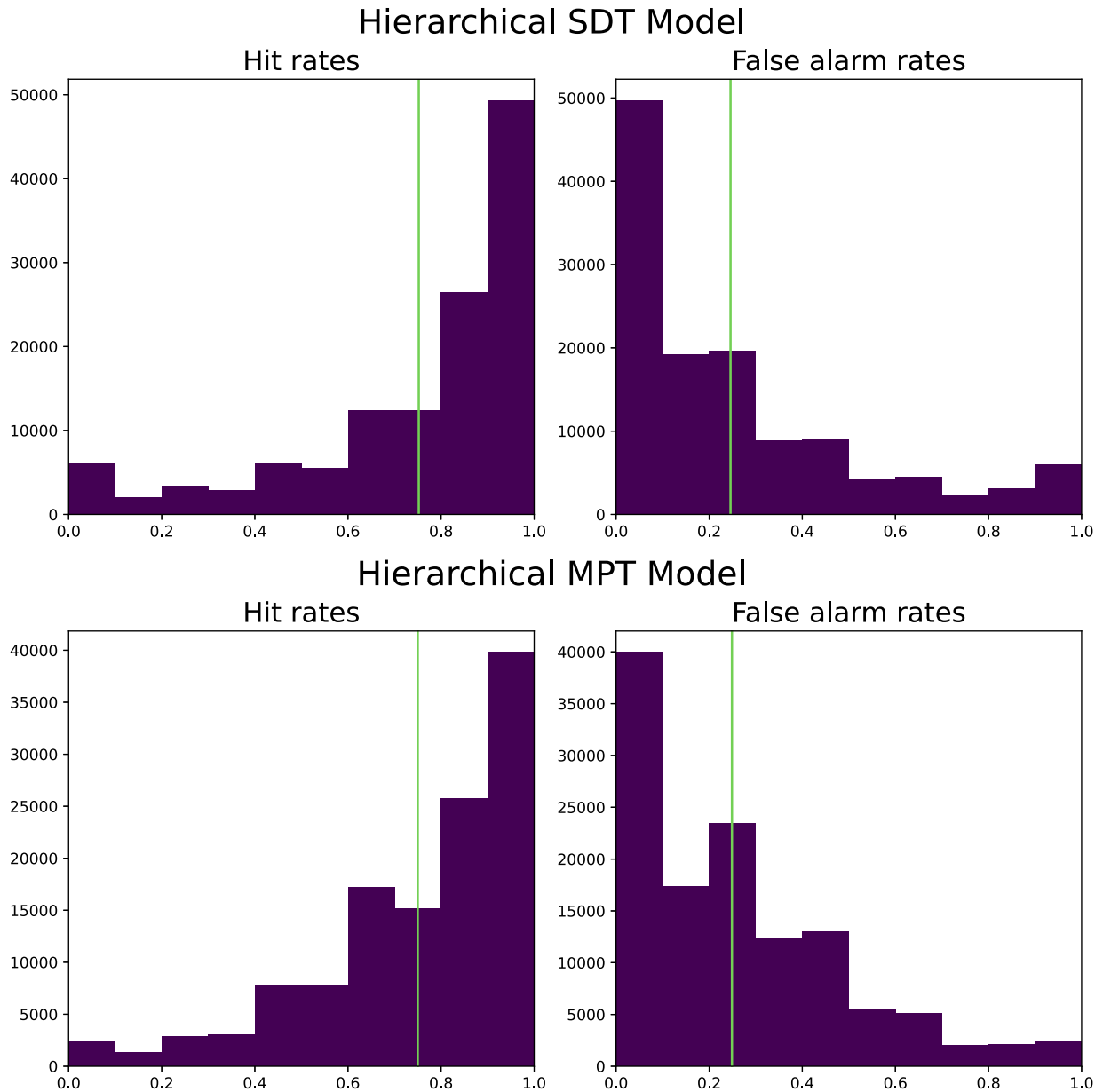
Note. The vertical (green) lines indicate the mean. SDT = signal detection theory; MPT = multinomial processing tree. See the online article for the color version of this figure.

Figure C2

Validation Study 2: Full Comparison Results for the Log BFs (All 100 Test Data Sets)

Note. SDT = signal detection theory; MPT = multinomial processing tree; BFs= Bayes factors. See the online article for the color version of this figure.

Appendix D

Application Details

Parameter Priors and Prior Predictive Checks

We base our priors upon the comprehensive collection of diffusion model parameter estimates by Tran et al. (2021). For the Lévy flight models, $\mathcal{M}_2$ and $\mathcal{M}_4$, we inform the prior on the additional α parameter by the estimates for comparable tasks (those completed under speed instructions) by Voss et al. (2019). For the intertrial variability parameters included in $\mathcal{M}_3$ and $\mathcal{M}_4$, we follow the nonhierarchical priors that Wiecki et al. (2013) suggest to use in hierarchical drift-diffusion models, but choose a nonpooling approach with individual parameters instead of a complete-pooling approach. Table D1 contains the hyperprior choices and Table D2 the group-level priors.

To ensure that the informed priors for our HMs accurately reflect prior knowledge at both levels, we conduct prior

predictive checks based on 10,000 simulations (displayed in Figures D1–D3).

Robustness Against Artificial Noise

Here, we inspect the stability of our neural network against additional noise injection. Figure D4 displays the model comparison results as increasing percentages of trials per participant are artificially masked as missing. We repeat the random masking of trials 100 times per percentage step to assess the sensitivity of the results to specific parts of the empirical data. Consistent with our main results, there is a clear separation between low evidence for $\mathcal{M}_1$ and $\mathcal{M}_2$ and substantial evidence for $\mathcal{M}_3$ and $\mathcal{M}_4$ across all settings. Despite our network being trained on the empirical amount of missing data, 3.17% over both tasks, we observe rank stability of the model comparison results up until 25% missing data per participant.
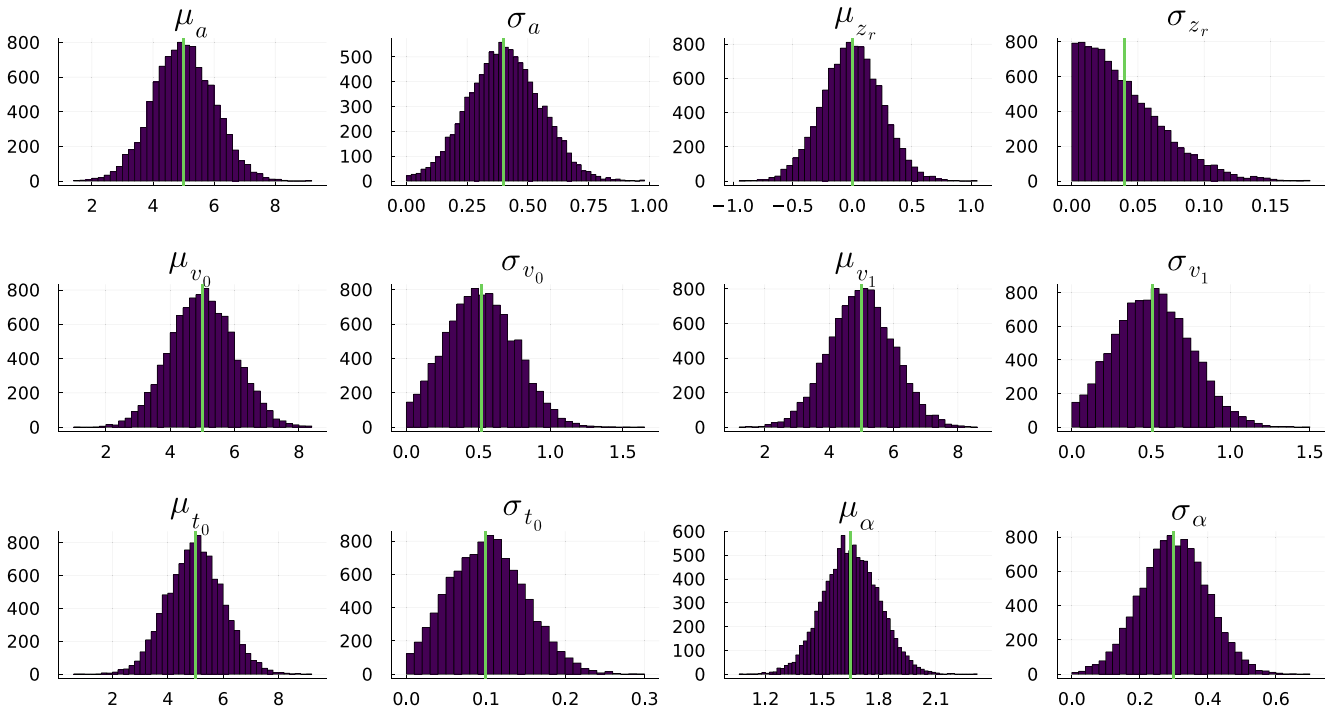
(Appendices continue)

Table D1

Real-Data Application: Hyperprior Distributions of the Evidence Accumulation Models

Parameter	Symbol	Prior distribution
Threshold separation	μ_a	Normal(5, 1)
	σ_a	Normal$_+$(0.4, 0.15)
Relative starting point	μ_{zr}	Normal(0, 0.25)
	σ_{zr}	Normal$_+$(0, 0.05)
Drift rate for blue/nonword stimuli	μ_{v_0}	Normal(5, 1)
	σ_{v_0}	Normal$_+$(0.5, 0.25)
Drift rate for orange/word stimuli	μ_{v_1}	Normal(5, 1)
	σ_{v_1}	Normal$_+$(0.5, 0.25)
Nondecision time	μ_{t_0}	Normal(5, 1)
	σ_{t_0}	Normal$_+$(0.1, 0.05)
Stability parameter of the noise distribution	μ_α	Normal(1.65, 0.15)
	σ_α	Normal$_+$(0.3, 0.1)

Table D2

Real-Data Application: Group-Level Prior Distributions of the Evidence Accumulation Models

Parameter	Symbol	Prior distribution
Threshold separation	a_m	Gamma(μ_a, σ_a)
Relative starting point	zr_m	invlogit(Normal(μ_{zr}, σ_{zr}))
Drift rate for blue/nonword stimuli	v_{0_m}	$-$Gamma(μ_{v_0}, σ_{v_0})
Drift rate for orange/word stimuli	v_{1_m}	Gamma(μ_{v_1}, σ_{v_1})
Nondecision time	t_{0_m}	Gamma(μ_{t_0}, σ_{t_0})
Stability parameter of the noise distribution	α_m	TruncatedNormal(μ_α, σ_α, 1, 2)
Intertrial variability of starting point	s_{z_m}	Beta(1, 3)
Intertrial variability of drift	s_{v_m}	Normal$_+$(0, 2)
Intertrial variability of nondecision time	s_{t_m}	Normal$_+$(0, 0.3)

Note. Normal$_+(\cdot)$ denotes a zero-truncated normal distribution that only allows for positive values. Truncated-Normal($\cdot$) denotes a truncated normal distribution with the lower and upper limits given by the last two values.
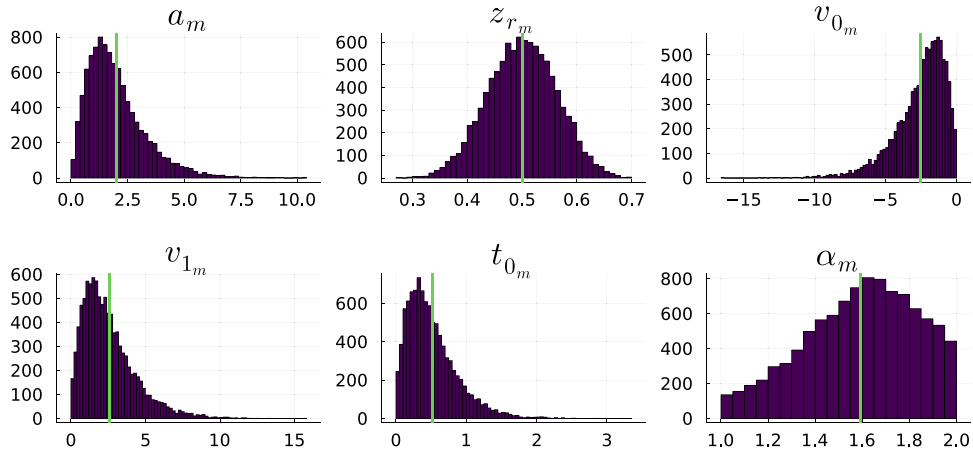
Figure D1

Real-Data Application: Prior Predictive Checks for the Hyperpriors in the Comparison of Evidence Accumulation Models

Note. The vertical (green) lines indicate the mean. See the online article for the color version of this figure.
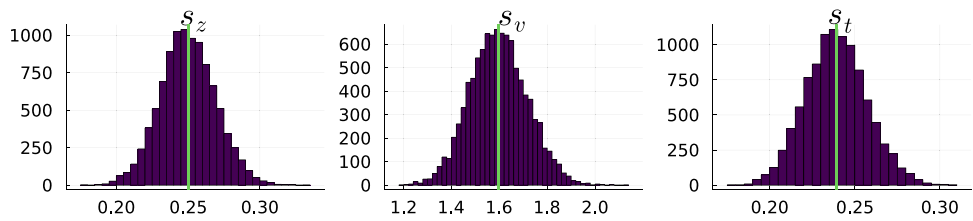
Figure D2

Real-Data Application: Prior Predictive Checks for the Hierarchical Group-Level Priors in the Comparison of Evidence Accumulation Models

Note. The vertical (green) lines indicate the mean. See the online article for the color version of this figure.

Figure D3

Real-Data Application: Prior Predictive Checks for the Nonhierarchical Group-Level Priors in the Comparison of Evidence Accumulation Models
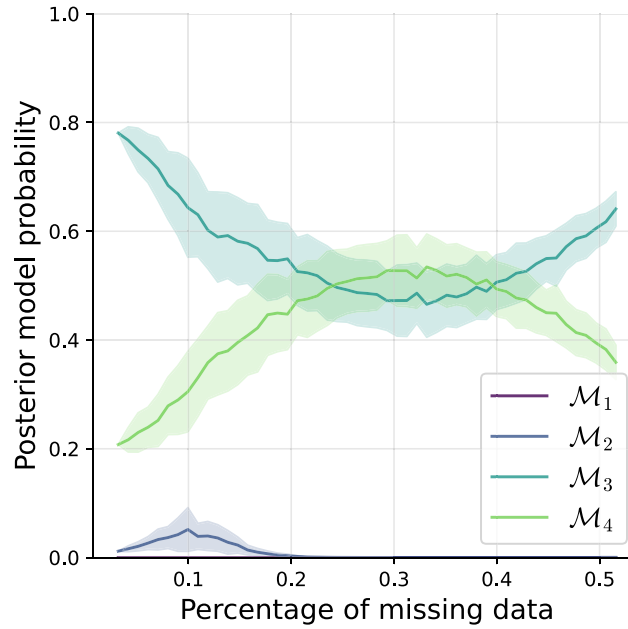

Note. The vertical (green) lines indicate the mean. See the online article for the color version of this figure.

(*Appendices continue*)

Figure D4

Real-Data Application: Robustness of the Model Comparison Results Against Increasing Amounts of Artificially Injected Random Noise

Note. The lines represent the average probabilities of 100 repetitions per percentage step (in each repetition masking a random subset of the empirical data), whereas the shaded areas indicate the standard deviation between these repetitions. See the online article for the color version of this figure.