Amortized Bayesian Workflow

Chengkun Li *chengkun.li@helsinki.fi*
University of Helsinki

Aki Vehtari *aki.vehtari@aalto.fi*
ELLIS Institute Finland, Aalto University

Paul-Christian Bürkner *paul.buerkner@tu-dortmund.de*
TU Dortmund University

Stefan T. Radev *radevs@rpi.edu*
Rensselaer Polytechnic Institute

Luigi Acerbi *luigi.acerbi@helsinki.fi*
University of Helsinki

Marvin Schmitt *mail.marvinschmitt@gmail.com*
Independent Scientist

Reviewed on OpenReview: *https://openreview.net/forum?id=osV7adJlKD*

Abstract

Bayesian inference often faces a trade-off between computational speed and sampling accuracy. We propose an adaptive workflow that integrates rapid amortized inference with gold-standard MCMC techniques to achieve a favorable combination of both speed and accuracy when performing inference on many observed datasets. Our approach uses principled diagnostics to guide the choice of inference method for each dataset, moving along the Pareto front from fast amortized sampling via generative neural networks to slower but guaranteed-accurate MCMC when needed. By reusing computations across steps, our workflow synergizes amortized and MCMC-based inference. We demonstrate the effectiveness of this integrated approach on several synthetic and real-world problems with tens of thousands of datasets, showing efficiency gains while maintaining high posterior quality.

1 Introduction

In many statistical modeling applications, from finance to biology and neuroscience, we often aim to infer unknown parameters θ from observables y modeled as a joint distribution $p(\theta, y)$ (e.g., Raulo et al., 2023; Seaton et al., 2023; George et al., 2022; Landmeyer et al., 2020; Chen et al., 2019; Malén et al., 2022; Schneider et al., 2018; Tsilifis & Ghosh, 2022). The posterior $p(\theta \mid y)$ is the statistically optimal solution to this inverse problem, and there are different computational approaches to approximate this target distribution.

Markov chain Monte Carlo (MCMC) methods constitute the most popular family of posterior sampling algorithms and still remain the gold standard for modern Bayesian inference due to their theoretical guarantees and powerful diagnostics (Gelman et al., 2013; 2020). MCMC methods yield autocorrelated draws conditional on a fixed dataset y_{obs}. As a consequence, the probabilistic model has to be re-fit for each new dataset, which involves repeating the entire MCMC procedure from scratch. Modern implementations equip MCMC with state-of-the-art extensions, for example, through Hamiltonian dynamics (HMC; Neal, 2011), by minimizing the required tuning by users (NUTS; Hoffman & Gelman, 2014), or by parallelizing thousands of chains on GPU hardware (ChEES-HMC; Hoffman et al., 2021). The well-established *Bayesian workflow* (Gelman et al., 2020) leverages these tools in an iterative process of model specification, fitting, evaluation, and revision.

While powerful, this approach becomes computationally burdensome when applied independently to large collections of datasets.

Differently, *amortized Bayesian inference* (ABI) aims to learn a direct mapping from observables y to the corresponding posterior $p(\theta \mid y)$, using flexible function approximators such as deep neural networks (Cranmer et al., 2020; Radev et al., 2020; Greenberg et al., 2019; Papamakarios et al., 2021; Wildberger et al., 2023; Sharrock et al., 2024; Zammit-Mangion et al., 2025). Amortized inference typically follows a two-stage approach: (i) a training stage, where neural networks learn to distill information from the probabilistic model based on simulated examples of observations and parameters $(\theta, y) \sim p(\theta)\, p(y \mid \theta)$; and (ii) an inference stage where the neural networks approximate the posterior distribution for an unseen dataset y_{obs} in near-instant time without repeating the training stage. In other words: The upfront training cost is *amortized* by negligible inference cost on arbitrary amounts of unseen test data. Owing to its reliance on simulated data, amortized inference in this form overlaps with *simulation-based inference* (Cranmer et al., 2020), which originated from posterior computations for models with intractable likelihood.

However, amortized inference lacks the powerful diagnostics and gold-standard guarantees associated with MCMC samplers in the standard Bayesian workflow (Gelman et al., 2020). Yet, applying a standard workflow is computationally prohibitive at scale. In modern Bayesian computation, MCMC and ABI occupy different ends of a Pareto frontier (see Figure 1): the former provides reliable accuracy at high cost, while the latter offers near-instant inference speed with limited per-dataset reliability (Hermans et al., 2022; Schmitt et al., 2023; Lueckmann et al., 2021).

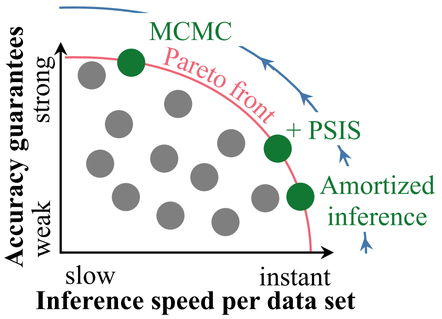


Figure 1: Our workflow adaptively moves along the Pareto front and reuses previous computations.

In this paper, we propose an adaptive workflow that consistently yields high-quality posterior draws while remaining computationally efficient. Our proposed workflow *moves along the Pareto front*, enabling fast-and-accurate inference when possible, and slow-but-guaranteed-accurate inference when necessary (see Figure 1). It combines the strengths of ABI and MCMC by incorporating diagnostic checks to guide inference decisions and reuse computations wherever possible. The resulting *amortized Bayesian workflow* therefore offers a principled, scalable, and diagnostic-driven approach for efficient posterior inference on many observed datasets; see Figure 2 for a conceptual overview.[1] To summarize, our contributions are:

- Design of—and systematic guidance through—an adaptive Bayesian workflow for accelerating Bayesian inference, which combines the strengths of amortized inference, importance sampling, and MCMC in a theoretically motivated and modular manner.
- Empirical validation of the workflow and of its inference speedup, demonstrating the applicability of the workflow on both synthetic and large-scale, real-world problems.

2 Integrating amortized inference into the Bayesian workflow

Our adaptive workflow starts with neural network training to enable subsequent amortized inference on a large number of unseen datasets—typically well into tens of thousands. This training phase is conceptually identical to standalone amortized inference training (e.g., Radev et al., 2020; Cranmer et al., 2020). For the inference phase, however, we develop a principled control flow that guides the analysis. Based on state-of-the-art diagnostics that are tailored to each step along the workflow, we propose decision criteria to select the appropriate inference algorithm for each observed dataset. In order to optimize the overall efficiency, our workflow contains mechanisms to reuse previous computations along the way.

[1] The software implementation is available at `https://github.com/pipme/amortized-Bayesian-workflow`.

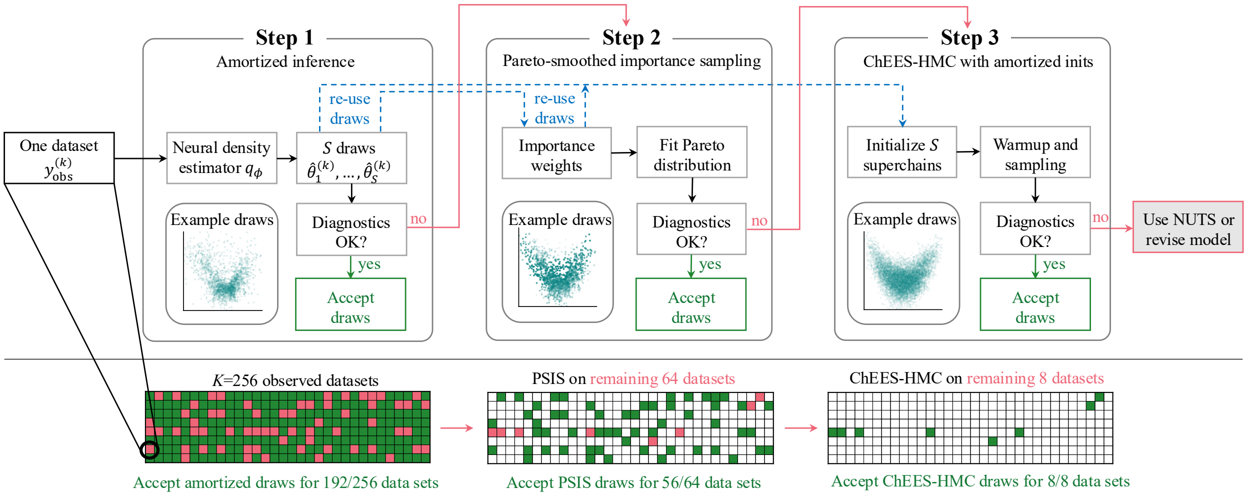

Step 1
Amortized inference
One dataset $y_{\text{obs}}^{(k)}$
Neural density estimator q_ϕ
S draws $\hat{\theta}_1^{(k)}, \dots, \hat{\theta}_S^{(k)}$
re-use draws
Example draws
Diagnostics OK?
no
yes
Accept draws
Step 2
Pareto-smoothed importance sampling
re-use draws
Importance weights
Fit Pareto distribution
Example draws
Diagnostics OK?
no
yes
Accept draws
Step 3
ChEES-HMC with amortized inits
Initialize S superchains
Warmup and sampling
Example draws
Diagnostics OK?
no
yes
Accept draws
Use NUTS or revise model
K=256 observed datasets
PSIS on remaining 64 datasets
ChEES-HMC on remaining 8 datasets
Accept amortized draws for 192/256 data sets
Accept PSIS draws for 56/64 data sets
Accept ChEES-HMC draws for 8/8 data sets

Figure 2: Our adaptive workflow leverages near-instant amortized posterior sampling when possible and gradually resorts to slower—but more accurate—sampling algorithms. As indicated by the blue dashed arrows, we reuse the S draws from the amortized posterior in Step 1 for the subsequent steps in the form of PSIS proposals (Step 2) and initial values in ChEES-HMC (Step 3).

2.1 Training phase: simulation-based optimization

In ABI, a neural estimator q_ϕ with trainable parameters ϕ typically minimizes a strictly proper scoring rule $\mathcal{S}$ (Gneiting & Raftery, 2007; Pacchiardi & Dutta, 2022) in expectation over the joint model $p(\theta, y) = p(\theta)p(y \,|\, \theta)$,

$$\phi = \arg\min_{\phi} \mathbb{E}_{(\theta,y)\sim p(\theta,y)} \left[\mathcal{S}\big(q_\phi(\cdot \,|\, y),\, \theta\big)\right]. \tag{1}$$

A popular choice is the logarithmic scoring rule, $\mathcal{S}\big(q_\phi(\cdot \,|\, y),\, \theta\big) := -\log q_\phi(\theta \,|\, y)$, which amounts to the forward Kullback-Leibler (KL) objective used for training normalizing flows in ABI (Greenberg et al., 2019; Radev et al., 2020). Score-based formulations that target a time-dependent gradient $\nabla_{\theta_t} \log p(\theta_t \,|\, y)$ are also possible (Sharrock et al., 2024; Gloeckler et al., 2024). Since most Bayesian models are generative by design, we can readily simulate M synthetic tuples of parameters and corresponding observations from the joint probabilistic model,

$$(\theta^{(m)}, y^{(m)}) \sim p(\theta, y) \quad \Leftrightarrow \quad \theta^{(m)} \sim p(\theta),\; y^{(m)} \sim p(y \,|\, \theta) \;\text{ for } m = 1, \dots, M, \tag{2}$$

which results in the training set $\{(\theta^{(m)}, y^{(m)})\}_{m=1}^{M}$ for optimizing Eq. 1. Throughout this paper, we use coupling-based normalizing flows (Durkan et al., 2019; Papamakarios et al., 2021) as a flexible conditional density estimator q_ϕ and the forward KL divergence as the training objective. However, our proposed workflow is agnostic to the specific choice of generative backbone used for amortization, as long as the model supports efficient sampling (see Section 2.2.1) and density evaluations (see Section 2.2.2).

Diagnostics. Since the neural network training algorithm hinges on simulated data, we cannot evaluate the amortized posterior estimator on real data just yet. However, we can easily simulate a synthetic *test set* $\{(\theta_\star^{(j)}, y^{(j)})\}_{j=1}^{J}$ of size J from the joint model via Eq. 2. In this *closed-world* setting, we know which "true" parameter vector $\theta_\star^{(j)}$ generated each simulated test dataset $y^{(j)}$. A key diagnostic for evaluating the amortized posterior estimator is *simulation-based calibration checking* (SBC; Talts et al., 2018; Säilynoja et al., 2022; Modrák et al., 2025; Yao & Domke, 2023). Formally, SBC involves (1) defining a test quantity $f: \Theta \times Y \to \mathbb{R}$ (e.g., marginal projections θ or the log likelihood $p(y \,|\, \theta)$), (2) computing this statistic for the true data-generating parameter $\theta_\star^{(j)}$, and (3) comparing it to the empirical distribution of the same statistic derived from amortized posterior draws given $y^{(j)}$ (Modrák et al., 2025). The rank of the true statistic within the posterior draws should be uniformly distributed if the amortized posterior estimator is well-calibrated.

We recommend assessing uniformity using the graphical approach by Säilynoja et al. (2022), which reveals the type of miscalibration present (e.g., bias or over-/under-dispersion) and is therefore useful for guiding improvements to amortized training. The choice of test quantity in SBC determines the sensitivity of the check; for example, the log-likelihood test quantity is typically more sensitive at detecting discrepancies than marginal projections (Modrák et al., 2025); using expressive neural classifiers is also possible (Yao & Domke, 2023). We further note a trade-off: imposing stricter criteria can improve the fidelity of the amortized estimator but will also tend to reject otherwise practically useful amortized estimators.

By default, we use marginal projections as the test quantities for SBC and complement SBC checking with *parameter recovery checking*, where parameter estimates are compared against known ground-truth parameters via direct visualization (Radev et al., 2020; 2023). Parameter recovery checking provides practical insight into whether the learned inverse mapping from y to θ is effective and helps mitigate a known failure mode of SBC with marginal projections as test quantities, in which the posterior approximation simply recovers the prior. We refer to Appendix A for further details and corresponding pseudocodes.

Note. Amortized inference lies at the intersection of Bayesian modeling and deep learning, unlocking massive potential for scalable posterior inference. However, this also comes with the practical challenges inherent to training deep neural networks. While a detailed treatment of neural architecture design and optimization exceeds the scope of this paper, practitioners can use established simulation-based inference libraries like `sbi` (Boelts et al., 2025) or `BayesFlow` (Radev et al., 2023), which provide modern plug-and-play components as well as sensible defaults for a wide range of applications. We summarize a set of best practices and actionable recommendations for training amortized posterior estimators in Appendix B.

Training phase: If simulation-based calibration checking and parameter recovery diagnostics pass, proceed to Step 1. Otherwise, tune the training hyperparameters (e.g., simulation budget, training epochs, learning rate, or neural network architecture) and re-train the amortized network.

2.2 Inference phase: posterior approximation on observed datasets

Once the amortized estimator is capable of yielding sufficiently accurate posterior draws in closed-world settings (i.e., in-distribution), we use the pre-trained neural network to achieve rapid amortized posterior inference on a total of K observed datasets $\{y_{\text{obs}}^{(k)}\}_{k=1}^{K}$. Recall that a given pre-trained amortized neural estimator may be perfectly suitable for some real datasets while it is utterly untrustworthy for others. Therefore, we want to assess on a per-dataset basis whether the amortized posterior draws are trustworthy and should be accepted, or whether we should proceed to a slower algorithm with stronger accuracy guarantees. The diagnostics in the inference phase are evaluated conditionally on each observed dataset, with the ultimate goal of determining whether the set of current posterior draws is acceptable for that specific dataset.

2.2.1 Step 1: Amortized posterior draws

We want to exploit the rapid sampling capabilities of the amortized posterior estimator q_ϕ as much as possible, as long as the sampled posteriors are trustworthy according to a set of principled diagnostics. Therefore, the natural first step for each observed dataset $y_{\text{obs}}^{(k)}$ is to query the amortized posterior and sample S posterior draws $\hat{\theta}_1^{(k)}, \ldots, \hat{\theta}_S^{(k)} \sim q_\phi(\theta \,|\, y^{(k)})$ in near-instant time (see Figure 2, first panel).

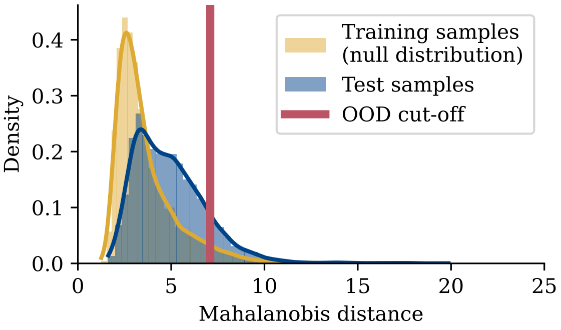


Figure 3: Illustration of our sampling-based hypothesis test that flags OOD datasets (to the right of the OOD cut-off).

Diagnostics. Like other neural network approaches (Yang et al., 2024), amortized inference may yield unfaithful results under distribution shifts (Schmitt et al., 2023; Ward et al., 2022; Huang et al., 2023). To address this, we detect whether an observed dataset y_{obs} is out-of-distribution (OOD) relative to the data-generating process

$p(\theta, y)$. We first compute a low-dimensional summary statistic $s(y) \in \mathbb{R}^d$ for each dataset.[2] The summary statistics from the training dataset $\{y^{(m)}\}_{m=1}^M$ are used to approximate the Mahalanobis distance by estimating their empirical mean μ_s and covariance Σ_s. Then, for any test dataset y, its Mahalanobis distance to the training set is:

$$D_M(y) = \sqrt{(s(y) - \mu_s)^\top \Sigma_s^{-1} (s(y) - \mu_s)}. \tag{3}$$

We compute $\{D_M(y^{(m)})\}_{m=1}^M$ for all training datasets to establish a frequentist sampling distribution of distances under the null hypothesis (i.e., of in-distribution datasets). Given a new observed dataset y_{obs}, we compare its Mahalanobis distance $D_M(y_{\text{obs}})$ to the empirical distribution of training distances. We define the OOD rejection rule as:

$$\text{Reject}_{\text{OOD}}(y_{\text{obs}}) = \mathbb{I}\left\{D_M(y_{\text{obs}}) > \text{Quantile}_{1-\alpha}\left(\{D_M(y^{(m)})\}_{m=1}^M\right)\right\}, \tag{4}$$

where α is by default set to 0.05 and we flag datasets whose Mahalanobis distances fall in the right α tail of the empirical training distances as out-of-distribution (see Figure 3). The type-I error rate α (false rejection) of this test can be set relatively high to obtain a conservative test that will flag many datasets for detailed investigation in further steps of our workflow.

In a nutshell, this is a sampling-based hypothesis test for distribution shifts, similar in spirit to the kernel-based test proposed by Schmitt et al. (2023). Since the amortized estimator has no guarantees nor known error bounds for data outside of the empirical support of the joint model $p(\theta, y)$(Elsemüller et al., 2024; Schmitt et al., 2023; Frazier et al., 2024; Elsemüller et al., 2025), we propagate such out-of-distribution datasets to Step 2. It is worth noting that a smaller Mahalanobis distance does not necessarily imply better posterior quality and that this OOD test is only intended to filter out datasets that are most likely to be problematic for the amortized estimator—specifically, those requiring extrapolation outside the ellipsoid defined by the training summary statistics.

Alternative diagnostics. In addition to the proposed out-of-distribution test, more sophisticated data-conditional diagnostics can further assess the accuracy of amortized posterior draws for individual datasets and enhance the reliability of accepted amortized draws. Examples include posterior simulation-based calibration checking (posterior SBC; Säilynoja et al., 2025) or the local classifier two-sample test (L-C2ST; Linhart et al., 2023), to name a few. These diagnostics each offer distinct advantages and limitations, but typically require substantially more computation than the OOD test.

Posterior SBC is conceptually straightforward and offers necessary conditions for the accuracy of amortized posterior samples by assessing consistency. However, it requires additional simulations for each test dataset and requires training the amortized estimator on inputs that effectively double the size of the original observations. L-C2ST, which trains classifiers to distinguish between $q_\phi(\theta \mid y)\, p(y)$ and the joint distribution $p(\theta, y)$, provides theoretically sufficient and necessary conditions for amortized inference accuracy. In practice, however, its effectiveness can be very sensitive to several factors, including classifier design choices (e.g., data pre-processing and optimization strategies), classifier calibration, and the relative sizes of the simulation budgets allocated to classifier training and amortized estimator training.

The choice to apply these additional diagnostics depends on context-specific factors, including the number of observed datasets, the relative computational cost of simulations versus likelihood evaluations,[3] and the dimensionality of the observations. Ultimately, whether amortized posterior draws are deemed acceptable hinges on the accuracy requirements of the specific application. By default, we recommend the OOD test for its simplicity, efficiency, and suitability as a first-line diagnostic.

[2] The summary statistic can be either learned by the amortized estimator q_ϕ in the training phase or be based on domain knowledge.

[3] For example, if likelihood evaluations are relatively cheap, instead of applying sophisticated diagnostics in Step 1, it is often worthwhile to process directly to Step 2, where Pareto-smoothed importance sampling can offer more informative and powerful diagnostics.

Step 1: If the observed dataset passes the OOD test (i.e., Mahalanobis distance is below the threshold), accept the amortized draws; otherwise, proceed to Step 2.

2.2.2 Step 2: Pareto-smoothed importance sampling

In this step, we use Pareto-smoothed importance sampling (PSIS) (Vehtari et al., 2024) to both improve and assess the quality of the amortized posterior draws of datasets which have previously been rejected (see Figure 2, second panel). Based on the amortized posterior draws from Step 1, PSIS computes importance weights $w_s^{(k)} = p(y^{(k)} \,|\, \hat{\theta}_s)\, p(\hat{\theta}_s)/q_\phi(\hat{\theta}_s \,|\, y^{(k)})$ for each observed dataset $y^{(k)}$ (as in default importance sampling). Then, PSIS fits a generalized Pareto distribution to the largest importance weights, which in turn is used to smooth the tail of the weight distribution (Vehtari et al., 2024). Finally, these smoothed importance weights are used for computing posterior expectations and for improving the posterior draws with the sampling importance resampling (SIR) scheme (Rubin, 1988). While the utility of standard importance sampling for improving neural posterior draws has previously been investigated (Dax et al., 2023), we specifically use the PSIS algorithm, which is self-diagnosing (see **Diagnostics** below) and therefore better suited for a principled workflow. Further details of PSIS are provided in Appendix A.

Diagnostics. We use the Pareto-$\hat{k}$ diagnostic to gauge the fidelity of the PSIS-refined posterior draws. Pareto-$\hat{k}$ is the estimated shape parameter of the generalized Pareto distribution and quantifies the tail heaviness of the largest importance weights. According to Vehtari et al. (2024), for moderate sample size ($S > 2000$), Pareto-$\hat{k} \leq 0.7$ indicates that PSIS estimates are reliable;[4] when $\hat{k} > 0.7$, the minimum sample size for obtaining a reliable Monte Carlo estimate through (Pareto-smoothed) importance sampling rapidly grows infeasibly large in practice, implying that the amortized posterior is a poor proposal for importance sampling correction and the corresponding dataset should be routed to Step 3. This $\hat{k}$ threshold is consistent with the established practice of using PSIS to improve and assess the quality of posterior approximations obtained from variational inference (Yao et al., 2018; Dhaka et al., 2021; Zhang et al., 2022).

Note. The posterior estimator in ABI is typically mode-covering since it optimizes the *forward* KL divergence in Eq. 1. When the neural network training is insufficient (e.g., small simulation budget or poorly optimized network), this may lead to overdispersed posteriors. Fortunately, this tends to err in the right direction, and PSIS can generally mitigate overdispersed *mode-covering* draws in low to moderate dimensions (Dhaka et al., 2021). In contrast, variational inference typically optimizes the *reverse* KL divergence (Rezende & Mohamed, 2015), which implies *mode-seeking* behavior that is less favorable for importance sampling.

Step 2: If Pareto-$\hat{k} \leq 0.7$, accept the importance sampling results; otherwise, proceed to Step 3.

2.2.3 Step 3: Many-chains MCMC with amortized initializations

If PSIS does not yield satisfactory results, we resort to an MCMC sampling scheme as a safe fallback option. In our amortized workflow, the MCMC step is augmented by reusing computations from the previous steps as initialization values. In principle, this step can incorporate any MCMC algorithm suited to the problem at hand. Examples include slice sampling for models with non-differentiable likelihoods (Neal, 2003), or HMC (Neal, 2011) samplers when gradients are available.

In this work, we use the ChEES-HMC algorithm (Hoffman et al., 2021) as an instantiation of MCMC. Most notably, ChEES-HMC supports the execution of thousands of parallel chains on a GPU for high-throughput sampling (Sountsov et al., 2024). Amortized posterior draws from previous steps provide a natural and convenient choice for initializing MCMC chains to accelerate convergence (Figure 4). This approach is conceptually similar to using methods like parallel quasi-Newton variational inference (i.e., *Pathfinder*; Zhang et al., 2022) to obtain initial values for MCMC chains. However, the amortized initial values are drawn in parallel in near-instant time, while Pathfinder requires re-fitting the variational approximation for each new observed dataset. For the purpose of ChEES-HMC initializations

[4] For small sample size ($S < 2000$), the threshold of Pareto-$\hat{k}$ is $\min(1 - 1/\log_{10}(S), 0.7)$.

with multimodal posterior distributions, it is again desirable that the amortized posterior draws are typically mass-covering (cf. Step 2). See Appendix A for additional details on the ChEES-HMC algorithm.

Diagnostics. In this last step, we use the nested $\widehat{R}$ diagnostic (Margossian et al., 2024), which is specifically designed to assess the convergence of the *many-but-short* MCMC chains.[5] If the diagnostics in this step indicate unreliable inference, we recommend resorting to the overarching Bayesian workflow (Gelman et al., 2020) and addressing the computational issues that even persist when using the (ChEES-)HMC algorithm. This could involve increasing the number of warmup iterations, using the established NUTS-HMC algorithm (Hoffman & Gelman, 2014; Carpenter et al., 2017), or revising the Bayesian model specification and parametrization.

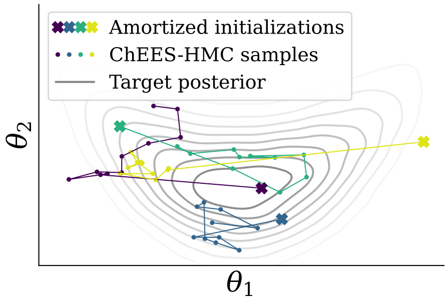


Figure 4: We initialize many ChEES-HMC chains with amortized draws.

Step 3: If (nested) $\widehat{R}$ is below the convergence threshold (e.g., 1.01), accept the MCMC draws. Otherwise, increase warm-up or revise the model according to the standard Bayesian workflow (Gelman et al., 2020).

2.3 Related work

Both simulation-based inference and amortized inference have seen rapid progress over the past decade (Zammit-Mangion et al., 2025; Cranmer et al., 2020; Lavin et al., 2021), driven by the need to perform Bayesian inference in complex models with intractable likelihoods (e.g., Dingeldein et al., 2024; Wehenkel et al., 2024; Zhou et al., 2024; Ghaderi-Kangavari et al., 2023; von Krause et al., 2022; Bieringer et al., 2021; Radev et al., 2021). These advances have been fueled by modern generative modeling, such as normalizing flows (Papamakarios et al., 2021; Radev et al., 2020; Greenberg et al., 2019), transformers (Müller et al., 2022; Chang et al., 2025; Whittle et al., 2025), diffusion models (Song et al., 2021; Sharrock et al., 2024; Linhart et al., 2024; Geffner et al., 2023; Gloeckler et al., 2024), consistency models (Song et al., 2023; Schmitt et al., 2024b), and flow matching (Lipman et al., 2023; Wildberger et al., 2023). Practical software toolkits such as `BayesFlow` (Radev et al., 2023) and `sbi` (Boelts et al., 2025) further make these simulation-based inference techniques accessible to practitioners in user-friendly interfaces.

To address the potential systematic errors of (amortized) neural posteriors, several works propose corrections using importance reweighting schemes (Dax et al., 2023; Starostin et al., 2025), augmented training objectives (Delaunoy et al., 2022; Mishra et al., 2025; Orozco et al., 2025; Schmitt et al., 2024a), or post-hoc corrections (Siahkoohi et al., 2023). Simultaneously, hybrid approaches that combine density estimators with MCMC have gained traction (Salimans et al., 2015; Hoffman et al., 2019; Gabrié et al., 2022; Midgley et al., 2022; Arbel et al., 2021; Cabezas et al., 2024; Grenioux et al., 2023). These include using variational approximations or learned flows as preconditioners for MCMC (Hoffman et al., 2019; Cabezas & Nemeth, 2023), adaptive proposal mechanisms (Parno & Marzouk, 2018; Gabrié et al., 2022), and initialization strategies to accelerate convergence or improve diagnostics (Zhang et al., 2022; Wang et al., 2023; Starostin et al., 2025).

More broadly, automated Bayesian inference has been a central design goal of probabilistic programming systems such as Stan (Carpenter et al., 2017), PyMC (Oriol et al., 2023), (Num)Pyro (Bingham et al., 2019; Phan et al., 2019). These libraries provide general-purpose inference engines—typically gradient-based MCMC and variational inference—that can be applied to a wide range of likelihood-based models and are accompanied by well-developed diagnostic recommendations and workflow guidelines (Gelman et al., 2020). However, they do not natively support amortized inference across many datasets, and inference must be rerun from scratch for each dataset.

Our proposed workflow builds on and complements these lines of work by integrating amortized inference, likelihood-based correction, and many-chain MCMC into a unified, modular, and diagnostic-driven pipeline for accelerating Bayesian inference. It dynamically adapts the inference strategy to the dataset at hand—using

[5] In more conventional settings involving long MCMC chains, the standard $\widehat{R}$ diagnostics (Gelman & Rubin, 1992; Vehtari et al., 2021) can be applied.

amortized posterior draws when they are adequate and escalating to PSIS and MCMC otherwise—thereby improving the robustness of amortized inference and the overall efficiency of posterior computation. This modular design provides a practical foundation for principled amortized inference across diverse data regimes.

3 Experiments

In this section, we empirically evaluate the effectiveness of our proposed amortized Bayesian workflow across various synthetic and real-world problems. We also examine how reusing amortized posterior draws in subsequent steps can improve the downstream sampling performance. The source code to reproduce all experiments is available in the supplementary material.

3.1 Procedure

Training settings. For each problem, we begin by training the amortized posterior estimator on simulated parameter-observation pairs (i.e., simulation-based training). We verify that the model performance is satisfactory in a closed-world setting, as diagnosed by simulation-based calibration and parameter recovery checking (see Section 2.1). Details on diagnostic results, simulation budgets, and training hyperparameters are provided in Appendix C.

Inference settings. For the out-of-distribution diagnostics in Step 1, we use the $\alpha = 0.05$ as the rejection threshold. We compute Mahalanobis distances in the summary statistics using 10,000 training simulations. We draw 2,000 posterior samples from the amortized posterior q_ϕ at Step 1. In Step 2, we correct the amortized draws using PSIS, rejecting draws if Pareto-$\hat{k} > 0.7$. Step 3 uses ChEES-HMC with convergence determined by nested $\widehat{R} < 1.01$. We run 2048 chains in parallel (16 superchains, each with 128 subchains), with 200 warmup steps and *a single sampling step*, for a total of 2048 posterior draws.

Evaluation metrics. To assess the quality of posterior draws from our workflow, we compare them to reference posterior draws using two evaluation metrics: the 1-Wasserstein distance (W1) and the mean marginal total variation distance (MMTV). The W1 distance quantifies the overall discrepancy between full joint distributions. MMTV measures the lack of overlap between marginal distributions and takes value in the range $[0, 1]$; for example, an MMTV value of 0.2 implies that, on average, the approximate posterior draws and reference draws share an 80% overlap for their marginal distributions. For both metrics, lower values indicate better posterior approximation quality. As a rule of thumb, MMTV values below 0.2 indicate good posterior approximation fidelity (Acerbi, 2020; Li et al., 2025).

3.2 Applications

We apply the proposed workflow to four posterior inference problems, including both simulated benchmarks and real-world experimental datasets. These case studies were chosen to reflect a range of commonly encountered statistical inference scenarios, including classical distributional parameter estimation and analyses of large-scale datasets arising in psychology and cognitive modeling. We describe each problem briefly below, with further details provided in Appendix C.

Generalized extreme value distribution (GEV). We consider parameter inference for the generalized extreme value (GEV) distribution, which models the maxima of samples from a distribution family. Each observation y_i is modeled as:

$$y_i \sim \mathrm{GEV}(\mu, \sigma, \xi), \tag{5}$$

where $\mu \in \mathbb{R}$ is the location, $\sigma \in \mathbb{R}_{>0}$ is the scale, and $\xi \in \mathbb{R}$ is the shape parameter. We follow the prior specification from Caprani (2021). For each dataset, we collect $N = 65$ i.i.d. observations and infer the posterior distribution over the parameter vector $\theta = (\mu, \sigma, \xi)$. We generate a total of $K = 1000$ test datasets by deliberately simulating from a model with a 2× wider prior distribution to emulate out-of-distribution settings in real applications (see Appendix C for details).

Bernoulli GLM. The Bernoulli generalized linear model (GLM) is a classical model with binary outcomes, included in the SBI benchmark suite (Lueckmann et al., 2021). Each observation $y_i \in \{0, 1\}$ is modeled as:

$$y_i \sim \text{Bernoulli}(\sigma(v_i^\top \theta)), \tag{6}$$

where $v_i \in \mathbb{R}^{10}$ is a fixed input vector, $\theta \in \mathbb{R}^{10}$ is the parameter vector, and $\sigma(\cdot)$ denotes the logistic function. We generate $K = 10,000$ in-distribution test datasets by sampling parameters from the model prior and simulating corresponding observations $\{y_i\}_{i=1}^{100}$ (Lueckmann et al., 2021).

Psychometric curve fitting. Psychometric functions are widely used in perceptual and cognitive science to characterize the relationship between stimulus intensity and the probability of a specific response (Wichmann & Hill, 2001). We use the overdispersed hierarchical model from Schütt et al. (2016), where the number of correct trials y_i at stimuli level x_i is modeled as:

$$y_i \sim \text{Binomial}(n_i, p_i), \quad p_i \sim \text{Beta}\left(\left(\frac{1}{\eta^2} - 1\right)\bar{p}_i,\ \left(\frac{1}{\eta^2} - 1\right)(1 - \bar{p}_i)\right), \tag{7}$$

where n_i is the number of trials, $\eta \in [0, 1]$ controls overdispersion, and $\bar{p}_i = \psi(x_i; m, w, \lambda, \gamma)$ is the expected success probability given by the psychometric function $\psi(x; m, w, \lambda, \gamma) = \gamma + (1 - \lambda - \gamma)\, S(x; m, w)$, where S is a sigmoid function (e.g., cumulative normal), m is the threshold, w is the width, λ is the lapse rate for infinitely high stimulus levels, and γ is the guess rate for infinitely low stimulus levels. In total, the model parameters are $\theta = (m, w, \lambda, \gamma, \eta)$. Our empirical evaluation uses 8,526 mouse behavioral datasets from the International Brain Laboratory public database (The International Brain Laboratory et al., 2021).

Decision model. The drift-diffusion model (DDM) is a popular evidence accumulation model for psychological models of human decision making (Ratcliff & McKoon, 2008). It describes a two-choice decision task as a stochastic process in which noisy evidence accumulates over time until it reaches one of the decision boundaries. The evolution of the decision variable $z(t)$ is modeled as

$$\mathrm{d}z(t) = v\, \mathrm{d}t + \sigma\, \mathrm{d}W(t), \tag{8}$$

where v is the drift rate (the average rate of evidence accumulation), σ is the noise scale, and $W(t)$ denotes a standard Wiener process. A decision is made when $z(t)$ reaches either a positive or negative boundary, typically placed symmetrically at $\pm a$, where a is the boundary separation. The model also includes a non-decision time parameter τ, capturing processes that are not part of the decision process. We adopt the model specification from von Krause et al. (2022), which extends the standard DDM to incorporate experimental condition effects via six parameters: $\theta = (v_1, v_2, a_1, a_2, \tau_c, \tau_n)$. The test datasets consist of 15,000 participants from the online implicit association test (IAT) database (Xu et al., 2014; von Krause et al., 2022), providing a large-scale, real-world benchmark for Bayesian inference in cognitive modeling.

3.3 Main Results

Table 1 summarizes the performance of the proposed amortized Bayesian workflow across the four problems described in Section 3.2. Step 1 (ABI) exhibits extremely low time per accepted dataset (TPA), with most of the cost incurred as a one-time expense during the training phase—including prior simulation, model training, and diagnostic evaluation. Once trained, ABI incurs negligible marginal cost ($\ll$ 1sec) when applied to a new dataset. Datasets flagged as out-of-distribution in Step 1 are forwarded to Step 2 for correction via PSIS. PSIS is highly effective, successfully correcting most rejected amortized draws and substantially reducing the number of datasets requiring full MCMC. Only a small subset of datasets progresses to Step 3, where ChEES-HMC is used for high-fidelity sampling. As the most computationally expensive component, ChEES-HMC is applied selectively, allowing the workflow to retain both accuracy and efficiency. Overall, the amortized workflow completes inference for nearly all datasets.[6] Compared to using ChEES-HMC for all datasets, our workflow achieves substantial computational savings—approximately over 5×, 120×, 60×, and 15× faster for the GEV, Bernoulli GLM, psychometric curve, and decision model tasks, respectively.

[6] A small number of datasets with particularly difficult properties require extended MCMC runs to converge.

Table 1: Summary of our amortized Bayesian workflow across four problems. For each step, we report the number of accepted datasets, wall-clock time (minutes), and **t**ime **p**er **a**ccepted dataset (TPA) in seconds. The time for the training phase includes amortized estimator training, simulations, and diagnostics evaluations. The time for Step 1 includes amortized posterior draws and the OOD test. The TPA for Step 1 accounts for both the training phase and Step 1. "Workflow total" aggregates the results of our method across all steps. As a baseline reference, "Baseline workflow total" is an estimate of the total required runtime for ChEES-HMC on all datasets.

Problem	**Step**	**Accepted datasets**	**Time (min)**	**TPA (s)**
GEV	Training phase	—	3	0.4
	Step 1: Amortized inference	523/1000	0.1	
	Step 2: Amortized + PSIS	357/477	0.8	0.1
	Step 3: ChEES-HMC w/ inits	87/120	11	7
	Workflow total (ours)	967/1000	15	0.9
	Baseline workflow total	—	85	—
Bernoulli GLM	Training phase	—	0.8	0.007
	Step 1: Amortized inference	9519/10000	0.3	
	Step 2: Amortized + PSIS	425/481	0.4	0.06
	Step 3: ChEES-HMC w/ inits	56/56	4	4
	Workflow total (ours)	10000/10000	5	0.03
	Baseline workflow total	—	688	—
Psychometric curve	Training phase	—	6	0.06
	Step 1: Amortized inference	7213/8526	0.4	
	Step 2: Amortized + PSIS	1215/1313	4	0.2
	Step 3: ChEES-HMC w/ inits	69/98	26	22
	Workflow total (ours)	8497/8526	37	0.3
	Baseline workflow total	—	2217	—
Decision model	Training phase	—	85	0.4
	Step 1: Amortized inference	13498/15000	1	
	Step 2: Amortized + PSIS	827/1502	47	3
	Step 3: ChEES-HMC w/ inits	554/675	526	57
	Workflow total (ours)	14879/15000	659	3
	Baseline workflow total	—	11594	—

Figure 5 presents the quality of posterior draws using the W1 distance (top row) and MMTV distance (bottom row), comparing draws from each step of the workflow against reference posteriors obtained via well-tuned NUTS. Rejected amortized draws (ABI✗) exhibit markedly worse performance than accepted ones (ABI✓), confirming the effectiveness of the OOD diagnostics.[7] PSIS-corrected draws offer accuracy comparable to ChEES-HMC samples, with only a slight decrease in quality. While amortized draws accepted in Step 1 are less accurate than those produced by PSIS or ChEES-HMC, they still provide high-quality approximations across the majority of datasets, as implied by the W1 and MMTV metrics. These results demonstrate that the proposed workflow not only scales efficiently but also consistently produces high-quality posterior estimates.

3.4 Advantage of amortized initializations for MCMC

One major goal of our workflow is to minimize reliance on expensive MCMC by maximizing the reuse of computations. Even when ABI and the PSIS refinement fail to yield acceptable posterior draws after Step 2, we can still leverage the amortized outputs to accelerate MCMC in Step 3.

To evaluate whether amortized posterior estimates remain useful in such cases, we test their effectiveness as initializations for ChEES-HMC chains. We conduct experiments on 20 randomly selected test datasets that progress to Step 3 of the workflow. This indicates that both the amortized posterior draws and their Pareto-

[7] For the Bernoulli GLM, the rejected amortized draws appear of good quality because the test datasets are drawn directly from the same prior used during training (i.e., in-distribution).

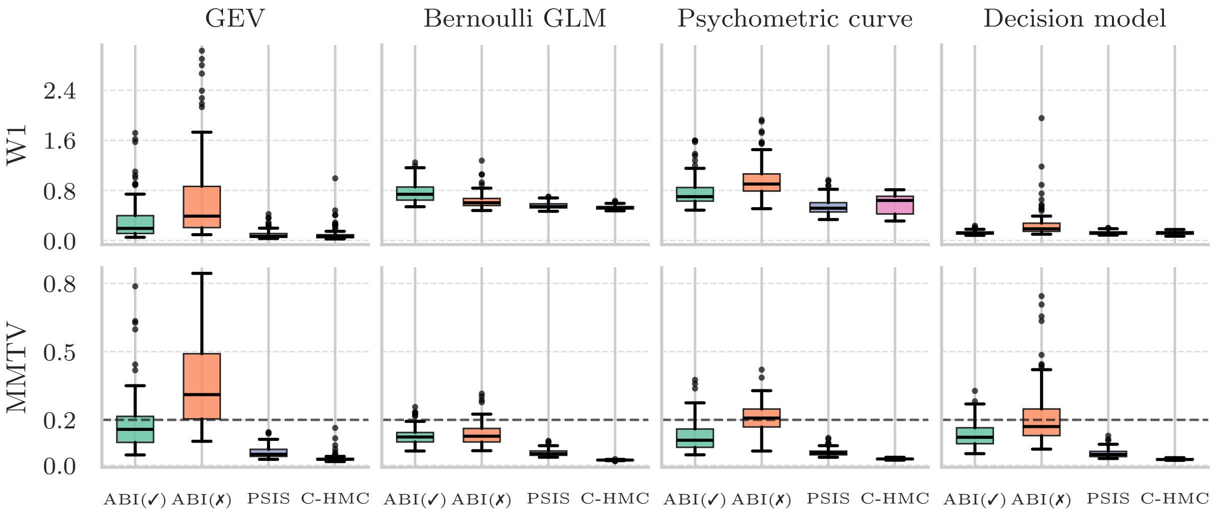


Figure 5: Evaluation of posterior draws across four problems based on two metrics: W1 distance (top row) and MMTV distance (bottom row). Lower values indicate better posterior approximation. ABI(✓) and ABI(✗) denote accepted and rejected draws, respectively, from amortized Bayesian inference in Step 1. PSIS denotes importance-weighted draws accepted in Step 2, and C-HMC denotes draws accepted via ChEES-HMC in Step 3. Metrics are computed on up to 100 datasets for each type of draws.

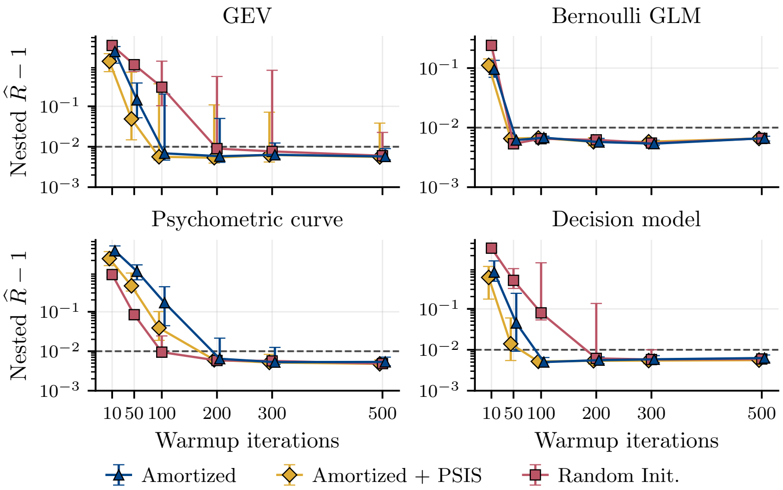


Figure 6: Using amortized posterior draws as initializations for ChEES-HMC reduces the required warmup in the GEV and decision model tasks. We show median±IQR across 20 test datasets in Step 3.

smoothed refinement are deemed unacceptable, as quantified by Pareto-$\hat{k} > 0.7$ in Step 2. We compare three initialization methods for ChEES-HMC chains: (1) amortized posterior draws, (2) PSIS-refined amortized draws, and (3) a random initialization scheme similar to Stan (Carpenter et al., 2017). We run the chains for varying numbers of warmup iterations, followed by a single sampling iteration. As described in Section 2, we use the nested $\widehat{R}$ value to gauge whether the chains converged appropriately during the warmup stage, as quantified by the common $\widehat{R}-1$ threshold of 0.01 (Vehtari et al., 2021).

Figure 6 shows that amortized posterior draws (and their PSIS-refined counterparts) can significantly reduce the required number of warmup iterations to achieve ChEES-HMC chain convergence, *even though the draws themselves have previously been flagged as unacceptable*. For the GEV problem and the decision model, chains initialized with amortized draws converge faster than those using random initialization. In the Bernoulli GLM, all methods perform similarly. For the psychometric curve model, random initialization leads to faster convergence for the early stage, but amortized draws still reach the convergence threshold at a similar speed at iteration 200, indicating competitive performance. These findings are particularly relevant in the many-short-chains regime, where computational cost is dominated by the warmup phase. For instance, with 2048 parallel chains, every single post-warmup step yields 2048 posterior samples, leading to enormous efficiency gains from shorter warmup.

Overall, these results demonstrate that amortized inference may provide suitable initializations for ChEES-HMC. However, the added benefit of initializing chains with PSIS-refined amortized draws (Step 2) instead of raw amortized draws (Step 1) remains unclear. While PSIS often accelerates convergence, it occasionally degrades worst-case performance (see upper error bounds for GEV task in Figure 6). We further study the impact of initialization for the popular NUTS sampler (Hoffman & Gelman, 2014), with similar results: amortized initializations reduce the required warmup in most cases (see Appendix E).

4 Discussion

We presented an adaptive Bayesian workflow to combine the rapid speed of amortized inference with the undisputed sampling quality of MCMC. Our amortized workflow enables a fundamental shift in the scale and feasibility of Bayesian inference. Applying traditional MCMC (e.g., ChEES-HMC) within a standard Bayesian workflow to every dataset independently would require approximately 10 days of GPU computation across our experimental suite. In contrast, our amortized workflow completes inference in half a day, achieving speedups ranging from over 5× to 120× depending on the problem. Crucially, high-quality posterior draws are retained through a cascade of diagnostics and selective escalation to PSIS and MCMC. In conclusion, our workflow efficiently uses resources by (i) applying fast amortized inference when the results are accurate; (ii) refining draws with PSIS when possible; and (iii) amortized initializations of slower but accurate MCMC chains when needed.

Modularity and practical flexibility. A key strength of the proposed workflow lies in its modular structure, which allows practitioners to tailor each component to the specific constraints and objectives of their application. In cases where preliminary analysis or low-latency decision-making is essential (e.g., real-time experimental pipelines) or where likelihood evaluations are computationally expensive, the workflow can operate in a lightweight mode using amortized inference with out-of-distribution rejection alone (i.e., Step 1 in our workflow). Conversely, in high-stakes applications where accuracy is paramount, analysts can enforce escalation of all amortized draws through PSIS and, if needed, proceed to full MCMC to guarantee statistical robustness. The choice of MCMC sampler in Step 3 is also fully interchangeable: alternative algorithms such as slice sampling, ensemble samplers (e.g., `emcee`; Foreman-Mackey et al., 2013), or NUTS can be substituted if the model is non-differentiable, multimodal, or requires richer exploration.

Furthermore, while our paper focuses on the trade-off between wall-clock inference speed and posterior quality, practical deployments may also involve additional factors, such as inference cost (e.g., monetary expense for GPU/CPU hours) and hardware availability. Consequently, the most suitable workflow variant can differ across settings. For example, when GPU resources are limited, launching parallel MCMC chains on CPUs offers a practical alternative, making the workflow more accessible for a broader range of users.

Applicability, limitations and future directions. Our proposed workflow targets likelihood-based Bayesian models for which prior predictive simulation and likelihood evaluation are possible. It is most beneficial in repeated-inference regimes (many datasets or frequent re-fits), with moderate effective dimensionality, and when a good amortized estimator can be trained once and subsequently reused. Hence, it is not universally suitable and does not yield inference speedup gains for all Bayesian models. Training amortized models requires upfront investment in optimization and simulation. In our experiments, we found that default neural network hyperparameter settings, such as normalizing flow architectures, summary network configurations, and optimizer settings, generally yield good performance.

However, in more challenging cases, such as the GEV problem, adjustments may be necessary, guided by training-phase diagnostics. The simulation burden can be exacerbated in high-dimensional ($\gtrsim 10$ parameters) or weakly identifiable models, where neural estimators may struggle to approximate complex inverse maps. Alternative amortized inference approaches (see, e.g., Mittal et al., 2025) could be explored in future work to complement simulation-based amortized inference in such scenarios. In settings where likelihood evaluations are particularly expensive, iterative refinements of the amortized estimator on individual datasets (Glöckler et al., 2022) may also be a practical alternative to likelihood-based corrections in Steps 2 and 3.

Moreover, while our diagnostic for the amortized posterior draws in Step 1 is effective and highly efficient in practice, it remains an imperfect proxy for the true posterior approximation error and can occasionally result in the acceptance of poor-quality amortized draws (cf. Figure 5). An additional empirical study in Appendix D shows that (1) the Mahalanobis distance and the posterior quality metrics are positively correlated when OOD datasets are present; (2) however, some low-distance datasets still yield poor metrics, highlighting the limitation that the OOD diagnostic cannot fully guarantee accuracy and motivating enforced escalation to PSIS (Step 2) when higher accuracy is a requirement. Future work could explore even more effective discrepancy measures, potentially tailored to the task at hand.

More broadly, the workflow supports a compelling vision of training amortized models once and reusing them across tasks or studies—a strategy well suited to applications ranging from psychology to computational biology, among others. In such settings, our layered diagnostics and selective escalation are crucial for maintaining reliability and efficiency. This positions the workflow as a practical bridge between amortized inference and traditional Bayesian rigor, enabling scalable yet trustworthy inference.

Acknowledgments

We thank Alexander Fengler for the help with PyMC implementation of the drift-diffusion model. CL and LA were supported by the Research Council of Finland (grants number 356498 and 358980 to LA). AV acknowledges the Research Council of Finland Flagship program: Finnish Center for Artificial Intelligence, and Academy of Finland project 340721. STR is supported by the National Science Foundation under Grant No. 2448380. MS and PB acknowledge support of Cyber Valley Project CyVy-RF- 2021-16, the DFG under Germany's Excellence Strategy – EXC-2075 – 390740016 (the Stuttgart Cluster of Excellence SimTech). PB additionally acknowledges the support of DFG Collaborative Research Cluster 391 "Spatio-Temporal Statistics for the Transition of Energy and Transport" – 520388526. MS acknowledges travel support from the European Union's Horizon 2020 research and innovation programme under grant agreements No 951847 (ELISE) and No 101070617 (ELSA), and support from the Aalto Science-IT project. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI) for supporting this project with computational and data storage resources. The authors also acknowledge the research environment provided by ELLIS Institute Finland.

References

Luigi Acerbi. Variational Bayesian Monte Carlo with noisy likelihoods. *Advances in Neural Information Processing Systems*, 33:8211–8222, 2020.

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport Monte Carlo. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 318–330. PMLR, July 2021.

Sebastian Bieringer, Anja Butter, Theo Heimel, Stefan Höche, Ullrich Köthe, Tilman Plehn, and Stefan T Radev. Measuring qcd splittings with invertible networks. *SciPost Physics*, 10(6):126, 2021.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(28):1–6, 2019. ISSN 1533-7928.

Jan Boelts, Michael Deistler, Manuel Gloeckler, Álvaro Tejero-Cantero, Jan-Matthis Lueckmann, Guy Moss, Peter Steinbach, Thomas Moreau, Fabio Muratore, Julia Linhart, Conor Durkan, Julius Vetter, Benjamin Kurt Miller, Maternus Herold, Abolfazl Ziaeemehr, Matthijs Pals, Theo Gruner, Sebastian Bischoff, Nastya Krouglova, Richard Gao, Janne K. Lappalainen, Bálint Mucsányi, Felix Pei, Auguste Schulz, Zinovia Stefanidi, Pedro Rodrigues, Cornelius Schröder, Faried Abu Zaid, Jonas Beck, Jaivardhan Kapoor, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. Sbi reloaded: A toolkit for simulation-based inference workflows. *Journal of Open Source Software*, 10(108):7754, April 2025. ISSN 2475-9066. doi: 10.21105/joss.07754.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Alberto Cabezas and Christopher Nemeth. Transport elliptical slice sampling. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 3664–3676. PMLR, April 2023.

Alberto Cabezas, Louis Sharrock, and Christopher Nemeth. Markovian flow matching: Accelerating MCMC with continuous normalizing flows. *Advances in Neural Information Processing Systems*, 37:104383–104411, December 2024.

Colin Caprani. Generalized extreme value distribution. https://www.pymc.io/projects/examples/en/latest/case_studies/GEV.html, 2021. PyMC Examples. Accessed: 2025-05-13.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Paul E Chang, Nasrulloh Loka, Daolang Huang, Ulpu Remes, Samuel Kaski, and Luigi Acerbi. Amortized probabilistic conditioning for optimization, simulation and inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2025.

Gang Chen, Paul-Christian Bürkner, Paul A. Taylor, Zhihao Li, Lijun Yin, Daniel R. Glen, Joshua Kinnison, Robert W. Cox, and Luiz Pessoa. An integrative Bayesian approach to matrix-based analysis in neuroimaging. *Human Brain Mapping*, 40(14):4072–4090, 2019. doi: 10.1002/hbm.24686.

François Chollet et al. Keras. https://keras.io, 2015.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 2020.

Maximilian Dax, Stephen R. Green, Jonathan Gair, Michael Pürrer, Jonas Wildberger, Jakob H. Macke, Alessandra Buonanno, and Bernhard Schölkopf. Neural importance sampling for rapid and reliable gravitational-wave inference. *Phys. Rev. Lett.*, 130:171403, Apr 2023. doi: 10.1103/PhysRevLett.130.171403. URL https://link.aps.org/doi/10.1103/PhysRevLett.130.171403.

Arnaud Delaunoy, Joeri Hermans, François Rozet, Antoine Wehenkel, and Gilles Louppe. Towards reliable simulation-based inference with balanced neural ratio estimation. *Advances in Neural Information Processing Systems*, 35:20025–20037, December 2022.

Akash Kumar Dhaka, Alejandro Catalina, Manushi Welandawe, Michael R Andersen, Jonathan Huggins, and Aki Vehtari. Challenges and opportunities in high dimensional variational inference. In *Advances in Neural Information Processing Systems*, volume 34, pp. 7787–7798. Curran Associates, Inc., 2021.

Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. TensorFlow distributions, 2017.

Lars Dingeldein, David Silva-Sánchez, Luke Evans, Edoardo D'Imprima, Nikolaus Grigorieff, Roberto Covino, and Pilar Cossio. Amortized template-matching of molecular conformations from cryo-electron microscopy images using simulation-based inference. *bioRxiv*, pp. 2024.07.23.604154, 2024. doi: 10.1101/2024.07.23.604154. URL http://biorxiv.org/content/early/2024/07/31/2024.07.23.604154.abstract.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Lasse Elsemüller, Hans Olischläger, Marvin Schmitt, Paul-Christian Bürkner, Ullrich Koethe, and Stefan T. Radev. Sensitivity-aware amortized Bayesian inference. *Transactions on Machine Learning Research*, 2024.

Lasse Elsemüller, Valentin Pratz, Mischa von Krause, Andreas Voss, Paul-Christian Bürkner, and Stefan T. Radev. Does unsupervised domain adaptation improve the robustness of amortized Bayesian inference? a systematic evaluation, 2025.

Alexander Fengler, Yang Xu, Bera Krishn, Aisulu Omar, and Michael J. Frank. HSSM: A generalized toolbox for hierarchical Bayesian estimation of computational models in cognitive neuroscience. Manuscript in preparation, 2025. URL https://github.com/lnccbrown/HSSM.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC hammer. *PASP*, 125: 306–312, 2013. doi: 10.1086/670067.

David T. Frazier, Ryan Kelly, Christopher Drovandi, and David J. Warne. The statistical accuracy of neural posterior and likelihood estimation, 2024.

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, March 2022. doi: 10.1073/pnas.2109420119.

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th international conference on machine learning*, volume 202 of *Proceedings of machine learning research*, pp. 11098–11116. PMLR, 2023. URL https://proceedings.mlr.press/v202/geffner23a.html.

Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7(4):457–472, November 1992. ISSN 0883-4237, 2168-8745. doi: 10.1214/ss/1177011136.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, Philadelphia, PA, 3 edition, 2013.

Andrew Gelman, Aki Vehtari, Daniel Simpson, et al. Bayesian workflow. *arXiv preprint*, 2020.

J.-P. George, P.-C. Bürkner, T. G. M. Sanders, M. Neumann, C. Cammalleri, J. V. Vogt, and M. Lang. Long-term forest monitoring reveals constant mortality rise in European forests. *Plant Biology*, 24(7): 1108–1119, 2022. doi: 10.1111/plb.13469.

Amin Ghaderi-Kangavari, Jamal Amani Rad, and Michael D. Nunez. A general integrative neurocognitive modeling framework to jointly describe EEG and decision-making on single trials. *Computational Brain & Behavior*, 6(3):317–376, 2023. ISSN 2522-0861, 2522-087X. doi: 10.1007/s42113-023-00167-4. URL https://link.springer.com/10.1007/s42113-023-00167-4.

Manuel Glöckler, Michael Deistler, and Jakob H. Macke. Variational methods for simulation-based inference. In *International Conference on Learning Representations*, 2022.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. All-in-one simulation-based inference. In *Proceedings the International Conference on Machine Learning (ICML)*, pp. 15735–15766, 2024.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Pedro J Gonçalves, Jan-Matthis Lueckmann, Michael Deistler, Marcel Nonnenmacher, Kaan Öcal, Giacomo Bassetto, Chaitanya Chintaluri, William F Podlaski, Sara A Haddad, Tim P Vogels, David S Greenberg, and Jakob H Macke. Training deep neural density estimators to identify mechanistic models of neural dynamics. *eLife*, 9:e56261, September 2020. ISSN 2050-084X. doi: 10.7554/eLife.56261.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, 2019.

Louis Grenioux, Alain Oliviero Durmus, Eric Moulines, and Marylou Gabrié. On sampling with approximate transport maps. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 11698–11733. PMLR, July 2023.

Joeri Hermans, Arnaud Delaunoy, François Rozet, Antoine Wehenkel, Volodimir Begy, and Gilles Louppe. A crisis in simulation-based inference? Beware, your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*, September 2022. ISSN 2835-8856.

Matthew Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport, March 2019.

Matthew Hoffman, Alexey Radul, and Pavel Sountsov. An adaptive-MCMC scheme for setting trajectory lengths in Hamiltonian Monte Carlo. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3907–3915. PMLR, 13–15 Apr 2021.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

Daolang Huang, Ayush Bharti, Amauri Souza, Luigi Acerbi, and Samuel Kaski. Learning robust statistics for simulation-based inference under model misspecification, 2023.

Ravin Kumar, Colin Carroll, Ari Hartikainen, and Osvaldo Martin. Arviz a unified library for exploratory analysis of bayesian models in python. *Journal of Open Source Software*, 4(33):1143, 2019. doi: 10.21105/joss.01143. URL https://doi.org/10.21105/joss.01143.

Nils C. Landmeyer, Paul-Christian Bürkner, Heinz Wiendl, Tobias Ruck, Hans-Peter Hartung, Heinz Holling, and Meuth. Disease-modifying treatments and cognition in relapsing-remitting multiple sclerosis: A meta-analysis. *Neurology*, 94(22):2373–2383, 2020. doi: 10.1212/WNL.0000000000009522.

Alexander Lavin, Hector Zenil, Brooks Paige, et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint*, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753, 2019.

Chengkun Li, Bobby Huggins, Petrus Mikkola, and Luigi Acerbi. Normalizing flow regression for Bayesian inference with offline likelihood evaluations. In *7th Symposium on Advances in Approximate Bayesian Inference*, 2025.

Julia Linhart, Alexandre Gramfort, and Pedro L. C. Rodrigues. L-C2ST: Local diagnostics for posterior approximations in simulation-based inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Julia Linhart, Gabriel Victorino Cardoso, Alexandre Gramfort, Sylvain Le Corff, and Pedro L. C. Rodrigues. Diffusion posterior sampling for simulation-based inference in tall data settings, June 2024. URL http://arxiv.org/abs/2404.07593. arXiv:2404.07593 [cs, stat].

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 343–351. PMLR, March 2021.

Tuulia Malén, Tomi Karjalainen, Janne Isojärvi, Aki Vehtari, Paul-Christian Bürkner, Vesa Putkinen, Valtteri Kaasinen, Jarmo Hietala, Pirjo Nuutila, Juha Rinne, and Lauri Nummenmaa. Atlas of type 2 dopamine receptors in the human brain: Age and sex dependent variability in a large PET cohort. *NeuroImage*, 255: 119149, 2022. doi: 10.1016/j.neuroimage.2022.119149.

Charles C. Margossian, Matthew D. Hoffman, Pavel Sountsov, Lionel Riou-Durand, Aki Vehtari, and Andrew Gelman. Nested $\widehat{R}$: Assessing the convergence of Markov chain Monte Carlo when running many short chains. *Bayesian Analysis*, pp. 1 – 28, 2024. doi: 10.1214/24-BA1453. URL https://doi.org/10.1214/24-BA1453.

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*, September 2022.

Aayush Mishra, Daniel Habermann, Marvin Schmitt, Stefan T. Radev, and Paul-Christian Bürkner. Robust amortized Bayesian inference with self-consistency losses on unlabeled data, 2025.

Sarthak Mittal, Niels Leif Bracher, Guillaume Lajoie, Priyank Jaini, and Marcus Brubaker. Amortized In-Context Bayesian Posterior Estimation, February 2025.

Martin Modrák, Angie H. Moon, Shinyoung Kim, Paul Bürkner, Niko Huurre, Kateřina Faltejsková, Andrew Gelman, and Aki Vehtari. Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity. *Bayesian Analysis*, 20(2):461–488, June 2025. ISSN 1936-0975, 1931-6690. doi: 10.1214/23-BA1404.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, June 2003. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1056562461.

Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011. ISBN 978-0-429-13850-8.

Abril-Pla Oriol, Andreani Virgile, Carroll Colin, Dong Larry, Fonnesbeck Christopher J., Kochurov Maxim, Kumar Ravin, Lao Jupeng, Luhmann Christian C., Martin Osvaldo A., Osthege Michael, Vieira Ricardo, Wiecki Thomas, and Zinkov Robert. PyMC: A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, 9:e1516, 2023. doi: 10.7717/peerj-cs.1516.

Rafael Orozco, Ali Siahkoohi, Mathias Louboutin, and Felix J Herrmann. Aspire: iterative amortized posterior inference for Bayesian inverse problems. *Inverse Problems*, 41(4), 2025.

Lorenzo Pacchiardi and Ritabrata Dutta. Likelihood-free inference with generative neural networks via scoring rule minimization, May 2022. arXiv:2205.15784.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Matthew D. Parno and Youssef M. Marzouk. Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682, January 2018. doi: 10.1137/17M1134640.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Stefan T Radev, Frederik Graw, Simiao Chen, Nico T Mutters, Vanessa M Eichel, Till Bärnighausen, and Ullrich Köthe. 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*, 2021.

Stefan T. Radev, Marvin Schmitt, Lukas Schumacher, Lasse Elsemüller, Valentin Pratz, Yannik Schälte, Ullrich Köthe, and Paul-Christian Bürkner. BayesFlow: Amortized Bayesian workflows with neural networks. *Journal of Open Source Software*, 8(89):5702, 2023. doi: 10.21105/joss.05702. URL https://doi.org/10.21105/joss.05702.

Roger Ratcliff and Gail McKoon. The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20(4):873–922, 2008.

Aura Raulo, Paul-Christian Bürkner, Jarrah Dale, English, Curt Lamberth, Josh A. Firth, and Coulson. Social and environmental transmission spread different sets of gut microbes in wild mice. *bioRxiv preprint*, 2023. doi: 10.1101/2023.07.20.549849.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Donald B. Rubin. Using the SIR algorithm to simulate posterior distributions. In *Bayesian statistics 3. Proceedings of the third Valencia international meeting, 1-5 June 1987*, pp. 395–402. Clarendon Press, 1988.

Teemu Säilynoja, Paul-Christian Bürkner, and Aki Vehtari. Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison. *Statistics and Computing*, 32 (2):1–21, 2022.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1218–1226. PMLR, June 2015.

Marvin Schmitt, Paul-Christian Bürkner, and Köthe. Detecting model misspecification in amortized Bayesian inference with neural networks. *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2023.

Marvin Schmitt, Desi R. Ivanova, Daniel Habermann, Ullrich Koethe, Paul-Christian Bürkner, and Stefan T. Radev. Leveraging self-consistency for data-efficient amortized Bayesian inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st international conference on machine learning*, volume 235 of *Proceedings of machine learning research*, pp. 43723–43741. PMLR, 2024a. URL https://proceedings.mlr.press/v235/schmitt24a.html.

Marvin Schmitt, Valentin Pratz, Ullrich Köthe, Paul-Christian Bürkner, and Stefan T. Radev. Consistency models for scalable and fast simulation-based inference. In *Proceedings of the 38th international conference on neural information processing systems*, 2024b. URL http://arxiv.org/abs/2312.05440.

Ilona Schneider, Harald Kugel, Ronny Redlich, Dominik Grotegerd, Christian Bürger, Paul-Christian Bürkner, Nils Opel, Katharina Dohm, Dario Zaremba, Susanne Meinert, et al. Association of serotonin transporter gene AluJb methylation with major depression, amygdala responsiveness, 5-HTTLPR/rs25531 polymorphism, and stress. *Neuropsychopharmacology*, 43(6):1308–1316, 2018. doi: 10.1038/npp.2017.273.

Heiko H. Schütt, Stefan Harmeling, Jakob H. Macke, and Felix A. Wichmann. Painfree and accurate Bayesian estimation of psychometric functions for (potentially) overdispersed data. *Vision Research*, 122:105–123, May 2016. ISSN 0042-6989. doi: 10.1016/j.visres.2016.02.002.

Fiona M. Seaton, David A. Robinson, Don Monteith, Inma Lebron, Paul-Christian Bürkner, Sam Tomlinson, Bridget A. Emmett, and Simon M. Smart. Fifty years of reduction in sulphur deposition drives recovery in soil pH and plant communities. *Journal of Ecology*, 111(2):464–478, 2023. doi: 10.1111/1365-2745.14039.

Louis Sharrock, Jack Simons, Song Liu, and Mark Beaumont. Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st international conference on machine learning*, volume 235 of *Proceedings of machine learning research*, pp. 44565–44602. PMLR, 2024. URL https://proceedings.mlr.press/v235/sharrock24a.html.

Ali Siahkoohi, Gabrio Rizzuti, Rafael Orozco, and Felix J. Herrmann. Reliable amortized variational inference with physics-based latent distribution correction. *GEOPHYSICS*, 88(3), 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International conference on learning representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th international conference on machine learning*, volume 202 of *Proceedings of machine learning research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Pavel Sountsov, Colin Carroll, and Matthew D. Hoffman. Running Markov chain Monte Carlo on modern hardware and software, November 2024.

Vladimir Starostin, Maximilian Dax, Alexander Gerlach, Alexander Hinderhofer, Álvaro Tejero-Cantero, and Frank Schreiber. Fast and reliable probabilistic reflectometry inversion with prior-amortized neural posterior estimation. *Science Advances*, 11(11):eadr9668, March 2025. ISSN 2375-2548. doi: 10.1126/sciadv.adr9668.

Teemu Säilynoja, Marvin Schmitt, Paul-Christian Bürkner, and Aki Vehtari. Posterior SBC: Simulation-based calibration checking conditional on data. *arXiv:2502.03279*, 2025.

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint*, 2018.

The International Brain Laboratory, Valeria Aguillon-Rodriguez, Dora Angelaki, Hannah Bayer, Niccolo Bonacchi, Matteo Carandini, Fanny Cazettes, Gaelle Chapuis, Anne K Churchland, Yang Dan, Eric Dewitt, Mayo Faulkner, Hamish Forrest, Laura Haetzel, Michael Häusser, Sonja B Hofer, Fei Hu, Anup Khanal, Christopher Krasniak, Ines Laranjeira, Zachary F Mainen, Guido Meijer, Nathaniel J Miska, Thomas D Mrsic-Flogel, Masayoshi Murakami, Jean-Paul Noel, Alejandro Pan-Vazquez, Cyrille Rossant, Joshua Sanders, Karolina Socha, Rebecca Terry, Anne E Urai, Hernando Vergara, Miles Wells, Christian J Wilson, Ilana B Witten, Lauren E Wool, and Anthony M Zador. Standardized and reproducible measurement of decision-making in mice. *eLife*, 10:e63711, May 2021. ISSN 2050-084X. doi: 10.7554/eLife.63711.

Panagiotis Tsilifis and Sayan Ghosh. Inverse design under uncertainty using conditional normalizing flows. In *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, January 2022. doi: 10.2514/6.2022-0631. URL http://dx.doi.org/10.2514/6.2022-0631.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2), 2021. ISSN 1936-0975. doi: 10.1214/20-ba1221. URL http://dx.doi.org/10.1214/20-BA1221.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25(72):1–58, 2024.

Mischa von Krause, Stefan T. Radev, and Andreas Voss. Mental speed is high until age 60 as revealed by analysis of over a million participants. *Nature Human Behaviour*, 6(5):700–708, May 2022. ISSN 2397-3374. doi: 10.1038/s41562-021-01282-7.

Yu Wang, Mikolaj Kasprzak, and Jonathan H. Huggins. A targeted accuracy diagnostic for variational approximations. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pp. 8351–8372. PMLR, April 2023.

Daniel Ward, Patrick Cannon, Mark Beaumont, Matteo Fasiolo, and Sebastian Schmon. Robust neural posterior estimation and statistical model criticism. *Advances in Neural Information Processing Systems*, 35:33845–33859, 2022.

Antoine Wehenkel, Jens Behrmann, Andrew C. Miller, Guillermo Sapiro, Ozan Sener, Marco Cuturi Cameto, and Jörn-Henrik Jacobsen. Simulation-based inference for cardiovascular models. In *NeurIPS workshop*, 2024. URL https://arxiv.org/abs/2307.13918.

George Whittle, Juliusz Ziomek, Jacob Rawling, and Michael A Osborne. Distribution transformers: Fast approximate Bayesian inference with on-the-fly prior adaptation. *arXiv preprint arXiv:2502.02463*, 2025.

Felix A. Wichmann and N. Jeremy Hill. The psychometric function: I. Fitting, sampling, and goodness of fit. *Perception & Psychophysics*, 63(8):1293–1313, November 2001. ISSN 1532-5962. doi: 10.3758/BF03194544.

Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in neural information processing systems*, volume 36, pp. 16837–16864, 2023.

Kaiyuan Xu, Brian Nosek, and Anthony G. Greenwald. Psychology data from the race implicit association test on the project implicit demo website. *Journal of Open Psychology Data*, 2014. doi: 10.5334/jopd.ac.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

Yuling Yao and Justin Domke. Discriminative calibration: Check bayesian computation from simulations and flexible classifier. *Advances in Neural Information Processing Systems*, 36:36106–36131, 2023.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5581–5590. PMLR, 10–15 Jul 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Andrew Zammit-Mangion, Matthew Sainsbury-Dale, and Raphaël Huser. Neural methods for amortized inference. *Annual Review of Statistics and Its Application*, 12(Volume 12, 2025):311–335, 2025. ISSN 2326-831X. doi: 10.1146/annurev-statistics-112723-034123.

Lu Zhang, Bob Carpenter, Andrew Gelman, and Aki Vehtari. Pathfinder: Parallel quasi-Newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49, 2022. URL http://jmlr.org/papers/v23/21-0889.html.

Lingyi Zhou, Stefan T. Radev, William H. Oliver, Aura Obreja, Zehao Jin, and Tobias Buck. Evaluating sparse galaxy simulations via out-of-distribution detection and amortized Bayesian model comparison, October 2024.

Appendix

This appendix provides additional details and analyses to complement the main text, included in the following sections:

- Background, Appendix A
- Best practices for training amortized estimators, Appendix B
- Experiment details, Appendix C
- Additional experimental study of the OOD diagnostic in Step 1, Appendix D
- Amortized initialization for NUTS, Appendix E

A Background

This section provides a concise overview of the diagnostics and algorithms used in our workflow, including simulation-based calibration checking, parameter recovery checking, out-of-distribution diagnostic with Mahalanobis distance, Pareto-smoothed importance sampling, the ChEES-HMC algorithm, and the nested $\widehat{R}$ convergence diagnostic. Pseudocodes are also given for reference.

Simulation-based calibration checking. Simulation-based calibration (SBC; Talts et al., 2018; Modrák et al., 2025) is a principled technique for assessing the calibration of posterior distributions estimated by Bayesian inference procedures, particularly useful in simulation-based amortized inference settings. SBC is based on the idea that if the posterior $p(\theta \mid y)$ is correctly specified, then the rank of the true parameter $\theta_\star$ among posterior draws should follow a uniform distribution. Formally, SBC defines a test statistic $f : \Theta \times Y \to \mathbb{R}$ (e.g., a component of θ, or the log-likelihood $p(y \mid \theta)$). For each simulated dataset $y^{(j)}$ generated from the joint model $p(\theta, y)$, the test statistic is evaluated at the ground-truth parameter $\theta_\star^{(j)}$ and compared to the same statistic evaluated over posterior samples $\{\theta_s^{(j)}\} \sim q_\phi(\theta \mid y^{(j)})$. The rank of $f(\theta_\star^{(j)}, y^{(j)})$ among $\{f(\theta_s^{(j)}, y^{(j)})\}$ is recorded. Repeating this process for all simulated datasets yields a distribution of rank statistics, which should be uniform under well-calibrated inference. Deviations from uniformity signal systematic bias (e.g., over/under-dispersion) in the posterior approximation. We use the graphical approach proposed by Säilynoja et al. (2022) to assess the uniformity of the rank statistics in SBC. This method provides visual diagnostics for identifying systematic biases or miscalibrations in the posterior approximation by plotting the empirical cumulative distribution function (ECDF) and confidence bands (95%). The pseudocode of SBC is provided in Algorithm 1. Examples of SBC checking results using this approach are provided in Appendix C.

Algorithm 1 SBC diagnostic

```
Require: Joint model $p(\theta, y)$; amortized posterior $q_\phi(\theta \mid y)$; scalar test function $f$; number of simulated datasets $J$ (e.g., 200); posterior draws $S$ (e.g., 1000)
for $j = 1, \ldots, J$ do
    Sample $\theta_\star^{(j)} \sim p(\theta)$, $y^{(j)} \sim p(y \mid \theta_\star^{(j)})$
    Draw $\theta_s^{(j)} \sim q_\phi(\theta \mid y^{(j)})$ for $s = 1, \ldots, S$
    Compute $T_\star^{(j)} = f(\theta_\star^{(j)}, y^{(j)})$, $T_s^{(j)} = f(\theta_s^{(j)}, y^{(j)})$
    Compute rank $r^{(j)} = \sum_{s=1}^{S} \mathbb{I}[T_s^{(j)} < T_\star^{(j)}] + \text{uniform}(0, \sum_{s=1}^{S} \mathbb{I}[T_s^{(j)} = T_\star^{(j)}])$
end for
Compare empirical ranks $r^{(j)}$ against the uniform$(0, S)$ distribution using the graphical approach of Säilynoja et al. (2022) to identify miscalibration patterns. Alternatively, the uniformity test based on the scalar statistic in Eq. 7 of Modrák et al. (2025) can also be applied.
```

Parameter recovery checking. Parameter recovery is a complementary diagnostic to SBC and provides a direct visualization of posterior approximation in recovering true generative parameters (Radev et al., 2020; 2023). The idea is to simulate a collection of datasets $\{y^{(j)}\}$ along with their corresponding ground-truth parameters $\{\theta_\star^{(j)}\}$ from the joint model, and assess whether the posterior distributions $q_\phi(\theta \mid y^{(j)})$ effectively recover these known values. In our workflow, we compare the posterior median extracted from each posterior to the corresponding ground-truth values, along with the median absolute deviation to indicate uncertainty. These comparisons are visualized using scatter plots, with correlation coefficients quantifying the strength of recovery. While not a direct measure of posterior calibration or correctness, parameter recovery provides important practical insight into whether the learned inverse mapping from y to θ is effective. The pseudocode of the parameter recovery diagnostic is provided in Algorithm 2. Examples of parameter recovery checking results using this approach are provided in Appendix C.

Algorithm 2 Parameter recovery diagnostic

```
Require: Joint model $p(\theta, y)$; amortized posterior $q_\phi(\theta \mid y)$; number of datasets $J$; posterior draws $S$
for $j = 1, \dots, J$ do
    Sample $\theta_\star^{(j)} \sim p(\theta)$, $y^{(j)} \sim p(y \mid \theta_\star^{(j)})$
    Draw $\theta_s^{(j)} \sim q_\phi(\theta \mid y^{(j)})$ for $s = 1, \dots, S$
    for each parameter component $k$ do
        Compute posterior median $\hat{\theta}_k^{(j)} = \mathrm{median}_s[\theta_{s,k}^{(j)}]$
        Optionally compute a dispersion measure (e.g., median absolute deviation) for $\theta_{s,k}^{(j)}$
    end for
end for
For each $k$, plot $\hat{\theta}_k^{(j)}$ vs. $\theta_{\star,k}^{(j)}$ and report correlation
```

OOD diagnostic with Mahalanobis distance. The out-of-distribution (OOD) diagnostic used in Step 1 tests whether an observed dataset y_{obs} falls outside the support of the prior predictive distribution (i.e., the training distribution for the amortized estimator). We work with low-dimensional summary statistics $s(y) \in \mathbb{R}^d$, which are either learned (e.g., via a summary network h_ψ) or hand-crafted with domain knowledge. In the latter case, the amortized estimator q_ϕ must be trained using these same hand-crafted statistics. The pseudocode for computing the Mahalanobis distance and checking whether an observed dataset y_{obs} is OOD is provided in Algorithm 3.

Algorithm 3 OOD diagnostic with Mahalanobis distance (Step 1)

```
Require: Training datasets $\{y^{(m)}\}_{m=1}^M$ (e.g., $M = 10000$); summary statistic function $s(\cdot)$; rejection level $\alpha$
    (e.g., $\alpha = 0.05$)
Compute $s^{(m)} = s(y^{(m)})$ for all $m$
Compute empirical mean $\mu_s = \frac{1}{M}\sum_{m=1}^{M} s^{(m)}$ and covariance $\Sigma_s = \frac{1}{M}\sum_{m=1}^{M}(s^{(m)} - \mu_s)(s^{(m)} - \mu_s)^\top$
For each $m$, compute Mahalanobis distance $D_M(y^{(m)}) = \sqrt{(s^{(m)} - \mu_s)^\top \Sigma_s^{-1}(s^{(m)} - \mu_s)}$
Let $T_\alpha$ be the empirical $(1 - \alpha)$-quantile of $\{D_M(y^{(m)})\}_{m=1}^M$

procedure TestOOD($y_{\text{obs}}$)
    Compute $s_{\text{obs}} = s(y_{\text{obs}})$ and $D_M(y_{\text{obs}})$
    return $\mathbb{I}_{D_M(y_{\text{obs}}) > T_\alpha}$                    ▷ 1 = OOD, 0 = in-distribution
end procedure
```

Pareto-smoothed importance sampling. Pareto-smoothed importance sampling (PSIS; Vehtari et al., 2024) is a robust method for improving the stability and reliability of importance sampling (IS) estimates. Given a target distribution $p(y \mid \theta)\, p(\theta)$ and a proposal distribution $q_\phi(\theta)$, with samples $\hat{\theta}_s \sim q_\phi(\theta)$, PSIS stabilizes the raw importance weights $w_s = p(y \mid \hat{\theta}_s)\, p(\hat{\theta}_s) / q_\phi(\hat{\theta}_s \mid y)$ by modeling the tail behavior of the importance weights. Specifically, the distribution of extreme importance weights can be approximated by a generalized Pareto distribution (GPD):

$$p(t \mid u, \sigma, k) = \begin{cases} \frac{1}{\sigma}\left(1 + k\left(\frac{t-u}{\sigma}\right)\right)^{-\frac{1}{k}-1}, & k \neq 0 \\ \frac{1}{\sigma}\exp\left(\frac{t-u}{\sigma}\right), & k = 0, \end{cases} \tag{9}$$

where k is the shape parameter, u is the location parameter and σ is the scale parameter. The number of finite fractional moments of the importance weight distribution depends on k: a generalized Pareto distribution has $1/k$ finite moments when $k > 0$. To stabilize the importance sampling estimate, the extreme importance weights are replaced with well-spaced order statistics drawn from the fitted generalized Pareto distribution, leading to a more stable and efficient IS estimator. The estimated shape parameter $\hat{k}$ serves as a diagnostic for the reliability of the importance sampling estimate. The pseudocode of PSIS is provided in Algorithm 4.

Algorithm 4 PSIS weights and Pareto-$\hat{k}$ diagnostic (Step 2)

```
Require: Observed data $y_{\text{obs}}$; log-likelihood $\log p(y \mid \theta)$; log-prior $\log p(\theta)$; amortized posterior $q_\phi(\theta \mid y)$; draws $\theta_s \sim q_\phi$, $s = 1, \ldots, S$
for $s = 1, \ldots, S$ do
    $\ell_s = \log p(y_{\text{obs}} \mid \theta_s) + \log p(\theta_s) - \log q_\phi(\theta_s \mid y_{\text{obs}})$
end for
$\tilde{\ell}_s = \ell_s - \max_s \ell_s$
$w_s = \exp(\tilde{\ell}_s)$
Sort $w_s$ to obtain $w(1) \leq \cdots \leq w(S)$
Choose tail size $M = \lfloor \min(0.2S, 3\sqrt{S}) \rfloor$ and define tail $w_{(S-M+1)}, \ldots, w_{(S)}$
Fit a GPD to the $M$ largest importance weights $w_{(S-M+1)}, \ldots, w_{(S)}$ and obtain shape estimate $\hat{k}$
Replace the $M$ largest weights with smoothed values from the fitted GPD to obtain stabilized weights $\tilde{w}_s$
Normalize: $\bar{w}_s = \tilde{w}_s / \sum_{r=1}^{S} \tilde{w}_r$
Use $\theta_s, \bar{w}_s$ as weighted PSIS-corrected posterior draws; treat them as reliable if $\hat{k} \leq \min(1 - 1/\log_{10}(S), 0.7)$
```

Given the PSIS-stabilized weights $\bar{w}_s$ from Algorithm 4, one can either compute weighted Monte Carlo estimates directly (Vehtari et al., 2024) or apply the SIR procedure in Algorithm 5 to obtain approximately unweighted posterior draws for downstream use (e.g., visualization or MCMC initialization).

Algorithm 5 Sampling importance resampling (SIR) using PSIS weights

```
Require: PSIS-corrected weighted sample $\{(\theta_s, \bar{w}_s)\}_{s=1}^{S}$; desired number of resampled draws $S'$ (e.g., $S' = S$)
Define a categorical distribution on indices $s \in \{1, \ldots, S\}$ with probabilities $\bar{w}_1, \ldots, \bar{w}_S$
for $j = 1, \ldots, S'$ do
    Sample index $I_j \sim \text{Categorical}(\bar{w}_1, \ldots, \bar{w}_S)$ ▷ Typically with replacement; weighted sampling without replacement is useful for generating unique initializations for MCMC chains
    Set $\tilde{\theta}_j = \theta_{I_j}$
end for
Return unweighted PSIS-corrected posterior draws $\{\tilde{\theta}_j\}_{j=1}^{S'}$
```

ChEES-HMC algorithm. The ChEES-HMC algorithm (Hoffman et al., 2021) is a massively parallel and adaptive extension of Hamiltonian Monte Carlo (HMC) designed to leverage single-instruction multiple-data (SIMD) hardware accelerators such as GPUs. This enables rapid generation of posterior draws following an initial warm-up phase. During warm-up, ChEES-HMC adaptively tunes the trajectory length T and step size ϵ by maximizing the "Change in the Estimator of the Expected Square" (ChEES), a heuristic that serves as a proxy for reducing autocorrelation in the second moments of the Markov chain. ChEES-HMC is particularly suitable for our amortized workflow, as we can easily generate a large number of good starting points (amortized draws) to launch many short MCMC chains. For our experiments, we used ChEES-HMC to run 2048 parallel chains, organized into 16 superchains with 128 subchains each. This configuration is essential for computing the nested $\widehat{R}$ diagnostic (Margossian et al., 2024), which assesses convergence across a large number of short chains. The pseudocode explaining the use of ChEES-HMC in Step 3 is provided in Algorithm 6.

Algorithm 6 Use of ChEES-HMC in Step 3

```
Require: Log-posterior density $\log p(\theta, y_{\text{obs}}) = \log p(y_{\text{obs}} \mid \theta) + \log p(\theta)$ and its gradient w.r.t. $\theta$; number of superchains $K$; number of subchains per superchain $M$; warm-up length $N_{\text{warmup}}$ (e.g., $N_{\text{warmup}} = 200$); after warm-up sampling length $N$ (e.g., $N = 1$)
Collect $K$ unique amortized posterior draws or PSIS-corrected draws for chain initialization; each of these $K$ draws must have a finite log-posterior density value.
For each superchain group $k = 1, \dots, K$, initialize $M$ subchains at the same initial state (total $K \times M$ chains)
Run ChEES-HMC warm-up for $N_{\text{warmup}}$ iterations to adapt step size $\epsilon$ and trajectory length $L$
Fix $(\epsilon, L)$ and run $N$ iterations to collect draws from $KM$ parallel chains
Compute nested $\widehat{R}$ for each parameter component of $\theta$
If nested $\widehat{R}$ is below a chosen threshold (e.g. $< 1.01$), accept the combined draws as Step 3 posterior draws; otherwise increase warmup length, use alternative MCMC algorithm (e.g., NUTS-HMC) or revise the model
```

Nested $\widehat{R}$ diagnostic. The potential scale reduction factor $\widehat{R}$ (Gelman & Rubin, 1992; Vehtari et al., 2021) is arguably the most popular diagnostic for assessing the convergence of MCMC chains. The basic idea is that multiple MCMC chains starting from overdispersed initial points should produce similar Monte Carlo estimators if they have converged, i.e., the impact of initialization vanishes as the chains converge to the stationary distribution. Nested $\widehat{R}$ diagnostic (Margossian et al., 2024) extends the classical $\widehat{R}$ diagnostic for monitoring convergence of many-short-chain MCMC samplers such as the ChEES-HMC algorithm. It requires organizing chains into K superchains, each consisting of M subchains that share the same initial point. Thus, one can assess convergence through comparing the variability between superchains and the variability within superchains, similar to the standard $\widehat{R}$ diagnostic. We provide the pseudocode for computing the nested $\widehat{R}$ diagnostic in Algorithm 7 for reference.

Algorithm 7 Nested $\widehat{R}$ diagnostic

```
Require: Posterior draws $\{\theta^{(nmk)}\}$ after warm-up, where $k \in \{1, \dots, K\}$ (superchains), $m \in \{1, \dots, M\}$ (subchains), $n \in \{1, \dots, N\}$ (draws); scalar function of interest $f$
Compute scalar values $f^{(nmk)} \leftarrow f(\theta^{(nmk)})$ for all $n, m, k$.
Compute subchain means: $\bar{f}^{(\cdot mk)} \leftarrow \frac{1}{N} \sum_{n=1}^{N} f^{(nmk)}$
Compute superchain means $\bar{f}^{(\cdot\cdot k)} \leftarrow \frac{1}{M} \sum_{m=1}^{M} \bar{f}^{(\cdot mk)}$ and overall mean $\bar{f}^{(\cdot\cdot\cdot)} \leftarrow \frac{1}{K} \sum_{k=1}^{K} \bar{f}^{(\cdot\cdot k)}$
for each superchain $k = 1, \dots, K$ do
    Compute between-chain variance $\tilde{B}_k$:
        If $M > 1$, $\tilde{B}_k \leftarrow \frac{1}{M-1} \sum_{m=1}^{M} (\bar{f}^{(\cdot mk)} - \bar{f}^{(\cdot\cdot k)})^2$; else $\tilde{B}_k \leftarrow 0$.
    Compute within-chain variance $\tilde{W}_k$:
        If $N > 1$, $\tilde{W}_k \leftarrow \frac{1}{M} \sum_{m=1}^{M} \left( \frac{1}{N-1} \sum_{n=1}^{N} (f^{(nmk)} - \bar{f}^{(\cdot mk)})^2 \right)$; else $\tilde{W}_k \leftarrow 0$.
end for
Compute between-superchain variance: $\widehat{B}_\nu \leftarrow \frac{1}{K-1} \sum_{k=1}^{K} (\bar{f}^{(\cdot\cdot k)} - \bar{f}^{(\cdot\cdot\cdot)})^2$
Compute within-superchain variance: $\widehat{W}_\nu \leftarrow \frac{1}{K} \sum_{k=1}^{K} (\tilde{B}_k + \tilde{W}_k)$
Return nested $\widehat{R} \leftarrow \sqrt{\frac{\widehat{W}_\nu + \widehat{B}_\nu}{\widehat{W}_\nu}}$
```

B Best practices for training amortized estimators

Amortized inference approaches problems of Bayesian modeling with methods from deep learning. While the precise training setup naturally depends on the concrete problem at hand, some general principles have proven help across a wide range of amortized inference applications. We summarize these here as initial guidance for applied practitioners.

Rely on established tooling. Modern libraries for amortized inference such as `sbi` (Boelts et al., 2025) and `BayesFlow` (Radev et al., 2023) provide well-tested neural density estimators, training loops, and data pipelines. In many cases, their default architectures and optimization settings already yield strong performance without manual tuning. First and foremost, we strongly recommend starting from these defaults and only introducing additional complexity if the diagnostics indicate deficiencies.

Monitor the training process. In many amortized inference settings, data are generated on the fly by a forward simulation program (see Eq. 2), and training does not rely on a fixed dataset. In this case, classical data splits into training and validation set are less meaningful, since each minibatch effectively constitutes fresh data from the joint model. Nonetheless, it is still important to monitor training progress with multiple signals, such as the training loss, calibration diagnostics, or summary statistics of posterior samples. These checks help assess whether the model continues to improve or has already reached a performance plateau after a short period. In other settings, amortized inference may be trained on a *fixed* set of simulated data, for example due to an expensive simulator or precomputed datasets. In such cases, holding out a validation set is strongly recommended to detect overfitting and guide selection of the amortized estimator.

Track multiple performance signals. Regardless of whether data are simulated on the fly or fixed, we recommend monitoring multiple indicators during training (e.g., after each epoch). Loss curves provide a coarse signal of optimization progress but are often insufficient on their own. For example, normalizing flows are usually trained with a negative log-likelihood loss, which does not account for mode coverage in multi-modal posterior distributions. Complementary diagnostics such as simulation-based calibration, parameter recovery, or posterior predictive checks on held-out datasets offer more direct insight into posterior quality and failure modes.

Assess and adjust model expressiveness. When training stagnates or diagnostics indicate systematic errors, the limiting factor is often model expressiveness rather than optimization details. Underexpressive models may show symptoms such as poor parameter recovery, persistent miscalibration, or posterior collapse toward the prior (i.e., data insensitivity), even when the training loss decreases. A pragmatic strategy is to begin with an overly expressive architecture (i.e., many trainable weights) to establish a performance baseline, and then gradually simplify the model. Conversely, if diagnostics remain unsatisfactory, increasing model capacity is often more effective than tuning hyperparameters of the optimizer.

Accept that training is iterative. Even with modern tooling, amortized inference training may require many iterations during development, especially for complex or weakly identifiable models. The objective is not to find a universally optimal neural architecture, but to reach a regime where the amortized posterior is reliable enough to enter the adaptive workflow proposed in this paper, where subsequent diagnostics and correction steps can take over.

C Experiment details

In this section, we provide additional experimental details omitted from the main text for brevity.

Evaluation metrics. To assess the quality of posterior approximations produced by each step of the workflow, we compare them against reference posterior draws obtained via a well-tuned No-U-Turn Sampler (NUTS). Specifically, we precomputed NUTS-based posterior samples for a subset of 5000 test datasets, which serve as a ground-truth reference for evaluation.[8] We then evaluate the 1-Wasserstein distance (W1) and the mean marginal total variation (MMTV) distance on up to 100 datasets from each inference step: Step 1 (amortized inference), Step 2 (amortized + PSIS), and Step 3 (ChEES-HMC with amortized initializations). These metrics are reported in the main text (Figure 5).

Neural network architecture for amortized inference. For all experiments, we use a coupling-based normalizing flow implemented in `BayesFlow` (Radev et al., 2023). The flow consists of 6 transformation layers, each comprising an invertible normalization, two affine coupling transformations, and a random permutation between elements. Before entering the coupling flow network as conditioning variables, the observed dataset y is encoded into a lower-dimensional summary statistic $h_\psi(y)$ via a summary network h_ψ. This summary network is implemented either as a DeepSet architecture (Zaheer et al., 2017) or a SetTransformer (Lee et al., 2019), depending on the problem setting. Both architectures are designed to handle permutation-invariant data structures. For the Bernoulli GLM, we bypass the summary network and directly use the known 10-dimensional sufficient statistics (Lueckmann et al., 2021). The specific choice of summary network for each application is described in the respective problem descriptions below.

Training-phase optimization. For all problems, the neural network is optimized via the AdamW optimizer (Loshchilov & Hutter, 2019) with weight decay of 10^{-3} and a cosine decay learning rate schedule (initial learning rate of 2.5×10^{-4}, a warmup target of 5×10^{-4}, $\alpha = 0$) as implemented in Keras (Chollet et al., 2015). A global gradient clip norm of 1.5 is applied. Training is performed with a batch size of 512 for 300 epochs,[9] with cosine decay steps set to the product of batch size and epochs. A held-out validation set is used to monitor optimization and select the best-performing model checkpoint.

Space transformation. Following standard practice in Bayesian computation (e.g., PyMC; Oriol et al., 2023, Stan; Carpenter et al., 2017), we transform parameters to an *unconstrained* space for inference. The amortized neural estimator is trained to estimate parameters in this unconstrained space. PSIS operates independently of the parameterization and thus remains unaffected by this transformation. ChEES-HMC also performs inference in the unconstrained space. All evaluation metrics (W1 and MMTV distances) are computed in this space. However, parameter recovery and simulation-based calibration plots are shown in the original constrained space for better interpretability.

Computing infrastructure and software. For all applications, the full workflow—including amortized training, inference, diagnostics, Pareto-smoothed importance sampling, and ChEES-HMC sampling—was conducted on a single NVIDIA V100 GPU (32GB), 8 cores of an AMD EPYC 7452 processor, and 8-16GB RAM. For runtime details across experiments, refer to Table 1 in the main text. The core code base was built using `BayesFlow` (Radev et al., 2023) (MIT license), PyMC (Oriol et al., 2023) (Apache-2.0 license), ArviZ (Kumar et al., 2019) (Apache-2.0 license) and JAX (Bradbury et al., 2018) (Apache-2.0 license). We used the implementation of ChEES-HMC provided by TensorFlow Probability (Dillon et al., 2017) (Apache-2.0 license).

Below, we provide details for each problem to complement the main text.

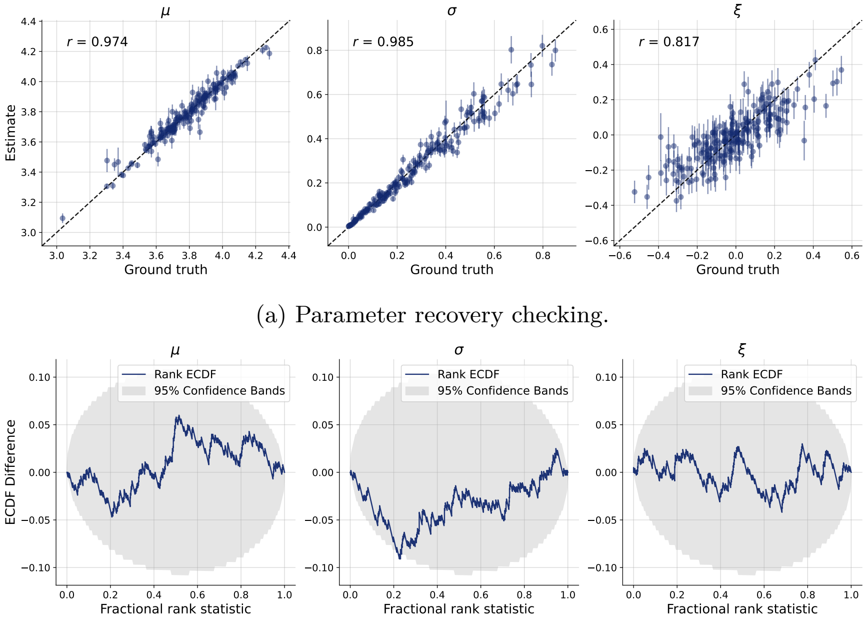
C.1 Generalized extreme value distribution

Problem description. Following Caprani (2021), the prior distribution for the parameters of the generalized extreme value distribution (GEV) is defined as:

$$\begin{aligned} &\mu \sim \mathcal{N}(3.8, 0.04) \\ &\sigma \sim \text{Half-Normal}(0, 0.09) \\ &\xi \sim \text{Truncated-Normal}(0, 0.04) \text{ with bounds } [-0.6, 0.6]. \end{aligned} \tag{10}$$

[8] For the generalized extreme value distribution problem, 1000 reference datasets were used, corresponding to the total number of test datasets.

[9] For Bernoulli GLM, we only train for 100 epochs.



(a) Parameter recovery checking.

(b) Simulation-based calibration checking.

Figure 7: Training-phase diagnostics for the GEV problem. The parameter recovery is strong for the parameters μ, σ, and good for the shape parameter ξ. Simulation-based calibration checking indicates good calibration for all parameters. Parameter recovery and simulation-based calibration checking indicate acceptable convergence of the amortized posterior estimator.

Simulation budgets. We use 10,000 simulated parameter–observation pairs for training the amortized estimator, 1000 for validation, and 200 for training-phase diagnostics, including parameter recovery and simulation-based calibration.

Summary network. We use a DeepSet as the summary network. The dimensionality of the learned summary statistics is 16. The DeepSet has a depth of 1, uses a *mish* activation, max inner pooling layers, 64 units in the equivariant and invariant modules, and 5% dropout.

Training-phase diagnostics. The closed-world diagnostics (parameter recovery and simulation-based calibration checking) in Figure 7 indicate that the neural network training has successfully converged to an acceptable posterior estimator within the scope of the training set.

Test datasets. In order to emulate distribution shifts that arise in real-world applications while preserving the controlled experimental environment, we simulate the "observed" datasets from a joint model whose prior is 2× wider (i.e., with 4× the variance) than the model used during training. More specifically, the prior is specified as:

$$\begin{aligned}
&\mu \sim \mathcal{N}(3.8, 0.16) \\
&\sigma \sim \text{Half-Normal}(0, 0.36) \\
&\xi \sim \text{Truncated-Normal}(0, 0.16) \text{ with bounds } [-1.2, 1.2].
\end{aligned} \tag{11}$$

C.2 Bernoulli GLM

Problem description. Following Lueckmann et al. (2021), we set the prior for θ as:

$$\theta \sim \mathcal{N}\left(0, \begin{bmatrix} 2 & 0 \\ 0 & (F^\top F)^{-1} \end{bmatrix}\right), \tag{12}$$

where the matrix F is defined such that $F_{i,i-2} = 1$, $F_{i,i-1} = -2$, $F_{i,i} = 1 + \sqrt{\frac{i-1}{9}}$, and $F_{i,j} = 0$ otherwise, for $1 \leq i, j \leq 9$. The task duration is set to $T = 100$, with fixed input vectors $\{v_i\}_{i=1}^{100}$, where each $v_i \in \mathbb{R}^{10}$. Corresponding observations are denoted by $\{y_i\}_{i=1}^{100}$. Further details can be found in Lueckmann et al. (2021); Gonçalves et al. (2020).

Simulation budgets. We use 10,000 simulated parameter–observation pairs for training the amortized estimator, 1000 for validation, and 200 for training-phase diagnostics, including parameter recovery and simulation-based calibration.

Summary network. For Bernoulli GLM, the 10-dimensional sufficient summary statistic for each dataset can be computed as $Vy^\top$ where $y = [y_1, \cdots, y_{100}]$ and $V = [v_1, \cdots, v_{100}]$. We therefore use this summary statistic for amortized training directly without relying on a separate summary neural network.

Training-phase diagnostics. The closed-world diagnostics (parameter recovery and simulation-based calibration checking) in Figure 8 indicate that the neural network training has successfully converged to an acceptable posterior estimator within the scope of the training set.

Test datasets. We generate $K = 10{,}000$ in-distribution test datasets by sampling parameters from the model prior and simulating corresponding observations $\{y_i\}_{i=1}^{100}$ from the Bernoulli distribution.

C.3 Psychometric curve fitting

Problem description. We adopt an overdispersed psychometric model (Schütt et al., 2016) with the error function (erf) as the sigmoid function in the psychometric function:

$$\psi(x; m, w, \lambda, \gamma) = \gamma + (1 - \lambda - \gamma)\ \mathrm{erf}(x; m, w), \tag{13}$$

where m is the threshold, w is the width, λ is the lapse rate, and γ is the guess rate.

The full probabilistic model is defined as follows:

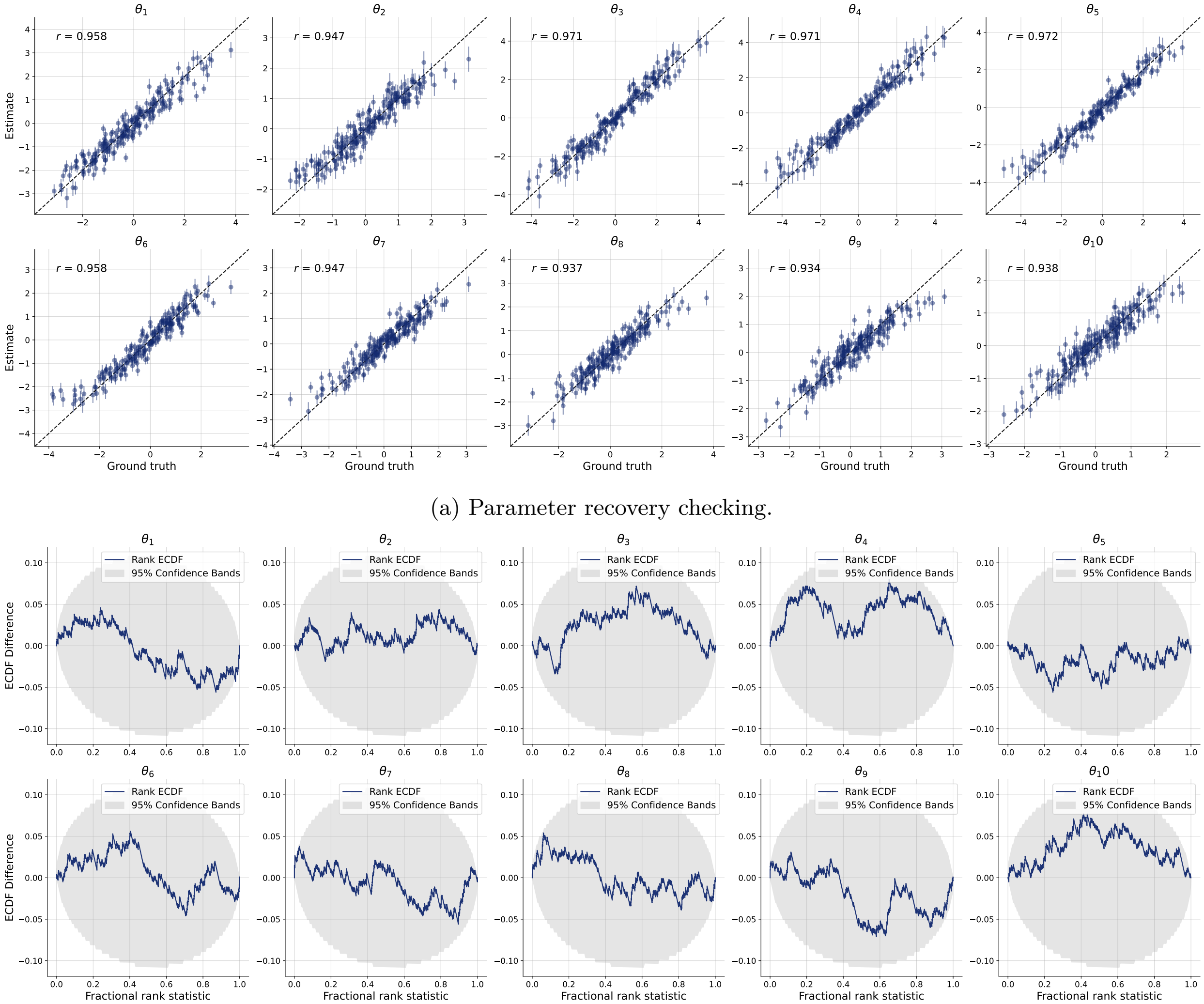
$$\begin{aligned}
\tilde{m} &\sim \mathrm{Beta}(2, 2), \\
w &\sim \text{Half-Normal}(0, 1), \\
\lambda, \gamma, \eta &\sim \mathrm{Beta}(1, 10), \\
m &= 2\tilde{m} - 1, \\
\bar{p}_i &= \psi(x_i; m, w, \lambda, \gamma), \\
p_i &\sim \mathrm{Beta}\left(\left(\frac{1}{\eta^2} - 1\right)\bar{p}_i,\ \left(\frac{1}{\eta^2} - 1\right)(1 - \bar{p}_i)\right), \\
y_i &\sim \mathrm{Binomial}(n_i, p_i),
\end{aligned} \tag{14}$$

where n_i denotes the number of trials, and x_i is the stimulus level. Stimuli are presented at nine fixed levels: $x_i \in \{-100.0,\ -25.0,\ -12.5,\ -6.25,\ 0.0,\ 6.25,\ 12.5,\ 25.0,\ 100.0\}$ and each value is further normalized by dividing by 100.

Simulation budgets. We use 50,000 simulated parameter–observation pairs for training the amortized estimator, 1000 for validation, and 200 for training-phase diagnostics, including parameter recovery and simulation-based calibration.

Summary network. We use a DeepSet as the summary network, which maps the input dataset to a 16-dimensional summary statistic. The DeepSet has a depth of 2, uses a *gelu* activation, mean inner pooling layers, 64 units in the equivariant and invariant modules, and 5% dropout.

Training-phase diagnostics. The closed-world diagnostics (parameter recovery and simulation-based calibration checking) in Figure 9 indicate that the neural network training has successfully converged to an acceptable posterior estimator within the scope of the training set.

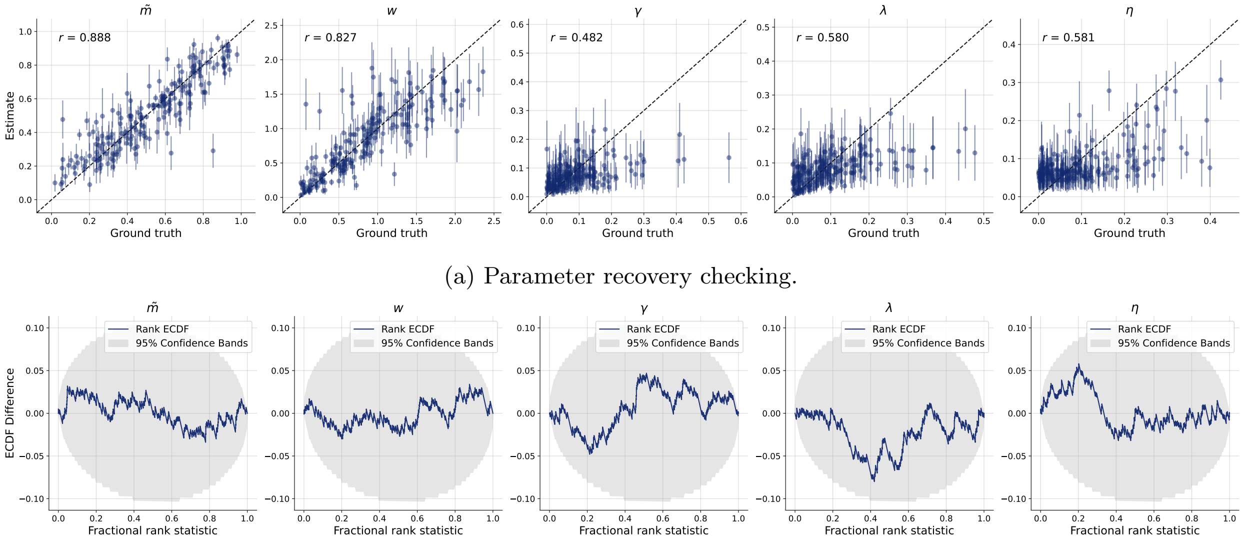


(a) Parameter recovery checking.

(b) Simulation-based calibration checking.

Figure 8: Training-phase diagnostics for the Bernoulli GLM problem. The parameter recovery is strong for all parameters. Simulation-based calibration checking indicates good calibration for all parameters. Parameter recovery and simulation-based calibration checking indicate acceptable convergence of the amortized posterior estimator.

Test datasets. Our empirical evaluation uses 8,526 mouse behavioral datasets from the International Brain Laboratory public database (The International Brain Laboratory et al., 2021). We retrieve the data using the provided API with the argument `task="biasedChoiceWorld"`, which corresponds to behavioral data collected after the mice have completed training. Each dataset is processed into an observation tensor of shape $(9, 3)$, where each row contains the number of correct trials y_i, the total number of trials n_i, and the stimulus level x_i.



(a) Parameter recovery checking.

(b) Simulation-based calibration checking.

Figure 9: Training-phase diagnostics for the psychometric curve fitting problem. Recovery is good for $\tilde{m}$ and w, while the other parameters exhibit weaker recoverability. Simulation-based calibration checking indicates excellent calibration for all parameters. Parameter recovery and simulation-based calibration checking indicate acceptable convergence of the amortized posterior estimator.

C.4 Decision model

Problem description. Following von Krause et al. (2022), we specify the prior distributions for the drift-diffusion model parameters as:

$$\begin{aligned} v_1, v_2 &\sim \text{Gamma}(2, 1), \\ a_1, a_2 &\sim \text{Gamma}(6, 0.15), \\ \tau_c &\sim \text{Gamma}(3, 0.15), \\ \tau_n &\sim \text{Gamma}(3, 0.5), \end{aligned} \tag{15}$$

where all Gamma distributions use the shape–scale parametrization.[10] We implement the drift-diffusion model likelihood using the `hssm` package (Fengler et al., 2025) and PyMC.

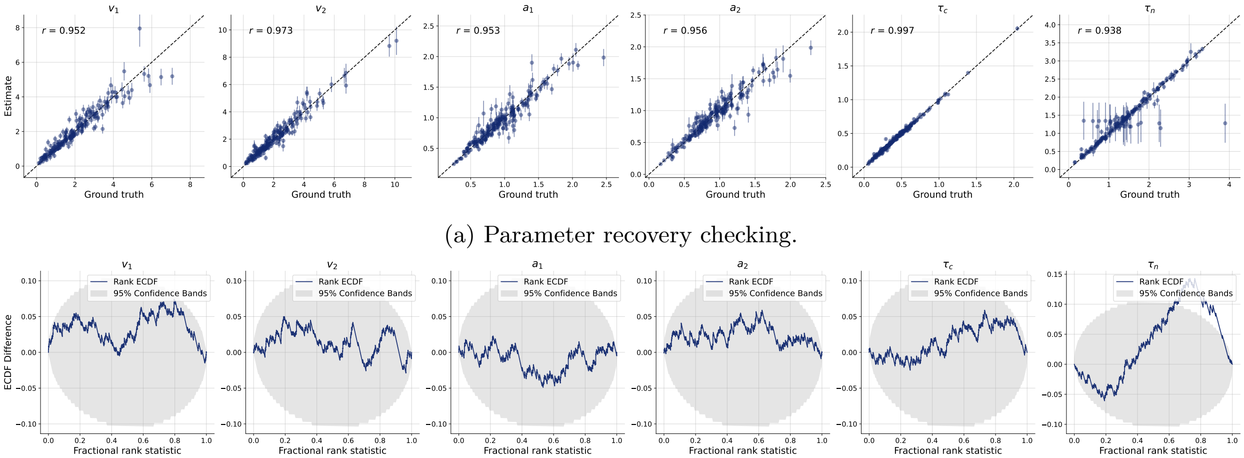
Simulation budgets. We use 100,000 simulated parameter–observation pairs for training the amortized estimator, 1000 for validation, and 200 for training-phase diagnostics, including parameter recovery and simulation-based calibration.

Summary network. We use a SetTransformer as the summary network, which maps the input dataset to a 16-dimensional summary statistic. The SetTransformer has two set attention blocks, followed by a pooling multi-head attention block and a fully connected output layer. Each multilayer perceptron (MLP) in the set blocks has two hidden layers of width 128, with *gelu* activation and 5% dropout.

Training-phase diagnostics. The closed-world diagnostics (parameter recovery and simulation-based calibration checking) in Figure 10 indicate that the neural network training has successfully converged to an acceptable posterior estimator within the scope of the training set.

Test datasets. The test datasets consist of 15,000 participants pre-processed from the online implicit association test (IAT) database (Xu et al., 2014; von Krause et al., 2022). Each test dataset is a tensor of

[10] The prior distributions for the boundary separation parameters a_1 and a_2 differ slightly from those in von Krause et al. (2022) due to a different parametrization of boundary separation.



(a) Parameter recovery checking.

(b) Simulation-based calibration checking.

Figure 10: Training-phase diagnostics for the decision model. Parameter recovery is strong for all parameters. Simulation-based calibration checking indicates good calibration for all parameters except τ_n, which shows mild deviations, suggesting occasional overestimation by the amortized estimator for this parameter. Parameter recovery and simulation-based calibration checking indicate acceptable convergence of the amortized posterior estimator.

shape $(120, 4)$, where each row corresponds to a single trial and contains the response time, missing data mask, experiment condition type, and stimulus type.

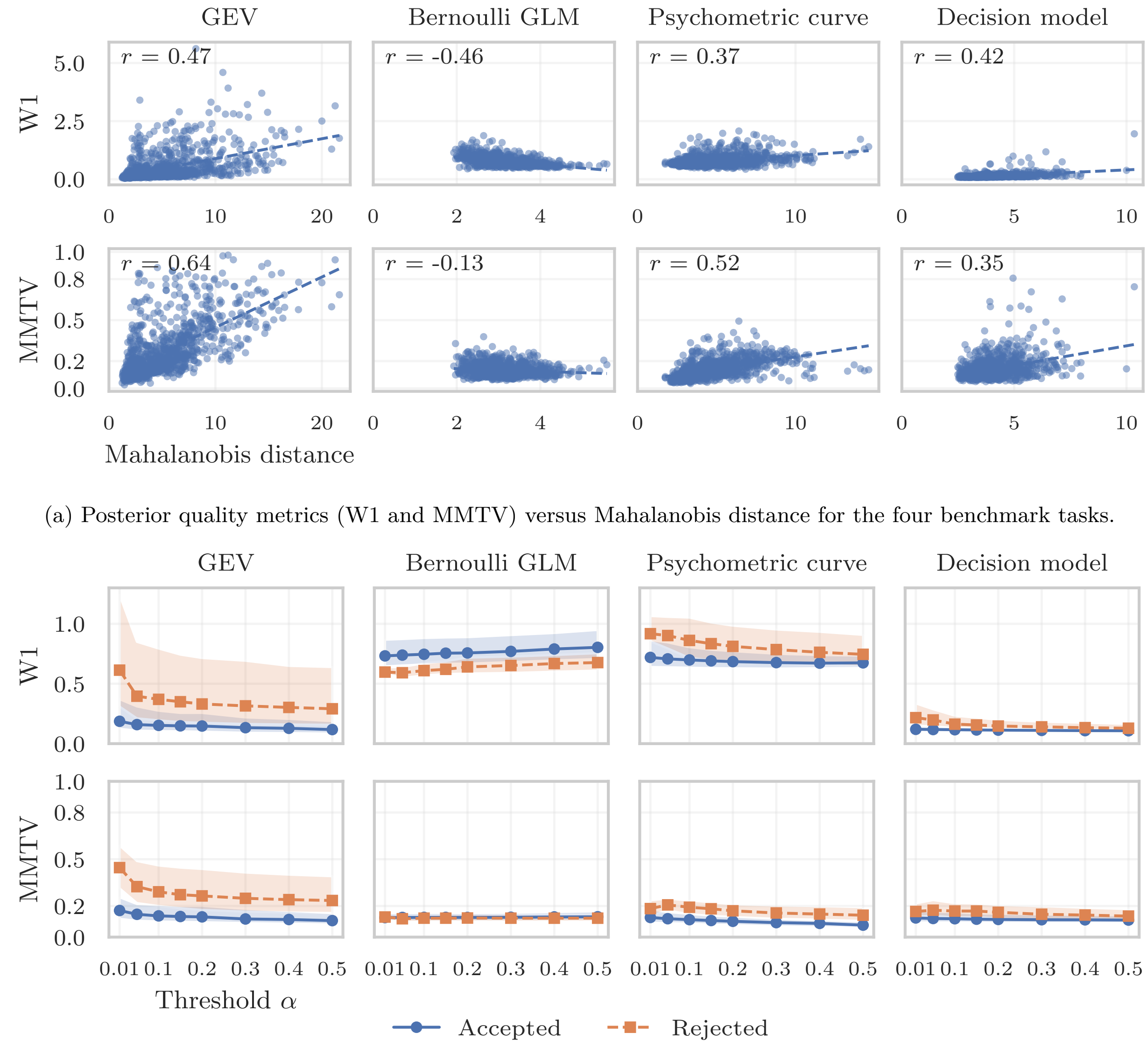
D Additional experimental study of the OOD diagnostic in Step 1

To further investigate the relationship between the Mahalanobis distance in the OOD diagnostic and the quality of the amortized posterior, we visualize this relationship using scatter plots in Figure 11a for the four tasks considered in the main text. For each task, we use around 1000 test datasets and compute the Pearson correlation coefficient r. The Mahalanobis distance is positively correlated with the two posterior quality metrics (W1 and MMTV) for the GEV, psychometric curve, and decision model tasks, where out-of-distribution test datasets are present. For the Bernoulli GLM, the correlation is negative; here, all test datasets were generated from the same distribution (prior simulations) as the training datasets and the Mahalanobis distance is not informative. From Figure 11a, we see a key limitation of the Step-1 OOD diagnostic: the Mahalanobis distance is clearly not a perfect proxy for the posterior quality. In particular, the amortized estimator may still yield low-quality posterior draws on a dataset with a smaller Mahalanobis distance, as also observed in Figure 5.

We next check the impact of the threshold α in the Step-1 OOD diagnostic by varying it from 0.01 to 0.5, specifically over the set $[0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5]$, as shown in Figure 11b. As α increases, more test datasets are rejected, and the overall posterior quality of accepted amortized posterior draws generally improves as measured by the median and IQR of the posterior metrics (lower posterior metric values indicate higher quality). The quality of rejected amortized posterior draws also improves as α increases, while remaining consistently worse than that of the accepted amortized draws.[11] Overall, the posterior quality of the accepted amortized draws, in terms of median and IQR of W1 and MMTV, is not very sensitive to the threshold α, and $\alpha = 0.05$ appears to be a reasonable default choice.

These results support the use of the Mahalanobis-distance-based OOD test as a lightweight first-line diagnostic in Step 1: it tends to flag the most problematic datasets and thereby improves the quality of accepted amortized posterior draws at negligible additional cost. At the same time, the residual low-quality posteriors at

[11] For Bernoulli GLM, where test datasets and training datasets come from the same distribution, we observe a slightly reverse trend as α increases. This is consistent with the corresponding result shown in Figure 11a and further suggests that the Mahalanobis distance is not a good measure for posterior quality for in-distribution datasets in this case.



(a) Posterior quality metrics (W1 and MMTV) versus Mahalanobis distance for the four benchmark tasks.

(b) Sensitivity of the rejection threshold α in the OOD test. The median ± IQR (shaded area) of the posterior quality metrics is shown separately for accepted and rejected datasets.

Figure 11: Relationship between amortized posterior quality metrics, Mahalanobis distance, and the OOD rejection threshold in Step 1. (a) Scatter plots of W1 and MMTV versus Mahalanobis distance, with Pearson correlation coefficient r reported in each panel. (b) Effect of varying the threshold α on the posterior quality of accepted and rejected amortized posterior draws. See text in Appendix D for details.

small Mahalanobis distances underscore that this diagnostic *cannot* guarantee accuracy. For applications that require tighter accuracy guarantees, it is therefore natural to enforce escalation to Step 2 (PSIS) irrespective of the OOD outcome, trading additional computation for a more robust posterior approximation.

E Amortized initialization for NUTS

In addition to ChEES-HMC, we evaluate the effectiveness of amortized posterior draws as initializations for the NUTS sampler. The experimental settings mirror those used for ChEES-HMC (Section 3.4), except that we launch only four chains, which is the typical configuration for NUTS. As shown in Figure 12, amortized initializations reduce the number of required warm-up iterations for both the GEV problem and the decision model. For the psychometric curve and Bernoulli GLM problems, all three initialization methods (amortized, PSIS-refined, and random) yield similar convergence behavior according to the $\hat{R}$ diagnostic (Vehtari et al., 2021).

Notably, NUTS generally requires fewer warm-up iterations than ChEES-HMC across the evaluated problems, suggesting that while amortized initializations are still beneficial, the relative gain is more pronounced for ChEES-HMC, which runs many short chains in parallel.

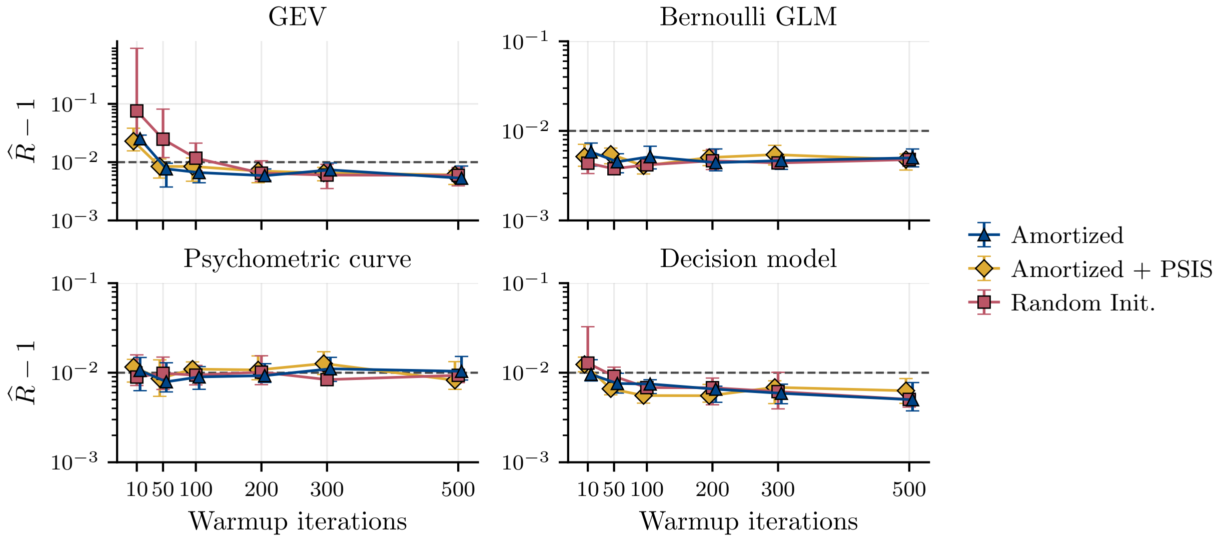


Figure 12: The effect of initialization for NUTS. The figure shows median±IQR across 20 test datasets. Using amortized posterior draws as initializations for NUTS reduces the required warmup in the GEV and decision model tasks.