Quit While You're Ahead: Sequential Training Can Harm Simulation-Based Inference

Kai Samaroo[1] **Yunyi Shen**[2] **Robin J. Ryder**[1] **Tamara Broderick**[2]

[1]Imperial College London
[2]Massachusetts Institute of Technology (MIT)

Abstract

Simulation-based inference (SBI) is widely used to infer parameters of simulators whose likelihood functions are intractable to evaluate. Recent work has suggested that sequential, multi-round training schemes can improve upon a single round. While these conclusions are supported by evaluations on models with at most a dozen parameters, simulators of practical interest can exhibit parameter dimensionalities approaching 100. In the present work, we systematically compare single-round and sequential SBI methods as parameter dimensionality increases to 100. We show empirically that sequential methods are often less accurate than single-round methods in dimensions higher than those previously studied.

1 Introduction

Many applications of statistical modeling require the specification of a simulator that generates synthetic data given a parameter setting, with the goal of mimicking the structure and dynamics of the underlying process being investigated. Ongoing advances in computational technology have enabled scientists to develop increasingly complex simulators that represent underlying processes with unprecedented fidelity. Examples arise in cosmology (Schafer and Freeman, 2012), neuroscience (Hashemi et al., 2023), glaciology (Moss et al., 2025), and computer graphics (Mansinghka et al., 2013). However, this increased complexity often renders the simulator's likelihood function intractable: the simulator can be sampled from for any parameter setting, but the likelihood function cannot be evaluated. This renders conventional Bayesian inference algorithms such as Markov Chain Monte Carlo (MCMC, Hoffman and Gelman, 2014) and Variational Inference (VI, Blei et al., 2017) inapplicable, as they crucially rely on likelihood evaluations. Simulation-based inference (SBI, Cranmer et al., 2020) provides a framework for Bayesian parameter inference in this setting by learning the distributional relationship between parameters and data from repeated model simulations. Modern neural-network-based SBI methods are now widely used by practitioners to infer parameters of real-world simulators (Deistler et al., 2025; SBI Developers, 2024a).

Many neural SBI methods have both *amortized* and *sequential* implementations. Sequential methods employ a multi-round training procedure in which the posterior approximation obtained at each round is used to refine the approximation at the next round. The sequential rounds aim to improve both accuracy and simulation efficiency by concentrating training simulations in regions relevant to the observed data. For each family of SBI methods, the amortized variant can be viewed as the sequential variant with a single round.

The SBI literature often recommends employing sequential methods when a highly accurate posterior approximation is required for a specific observation (Papamakarios and Murray, 2016; Greenberg et al., 2019; Papamakarios et al., 2019; Lueckmann et al., 2021; Deistler et al., 2025). However, exist-

ing comparisons of amortized and sequential SBI methods that we are aware of consider parameter spaces with at most 12 dimensions (a similar bound holds even beyond comparisons of sequential and amortized methods; see Section 3). Thus, it is plausible a priori that sequential methods behave differently in high-dimensional parameter spaces in ways not captured by the existing literature.

Our contributions: We systematically compare the accuracy of leading sequential and amortized SBI methods across parameter dimensionalities reaching 100 in two separate model classes. Surprisingly, **we find that sequential methods often *degrade* performance relative to their amortized counterparts.** We observe this phenomenon most frequently in the sequential neural posterior estimation methods we consider (Greenberg et al., 2019; Deistler et al., 2022), which are among the most widely used sequential SBI approaches in practice.

2 Background

Setup and notation: We assume access to a simulator that generates synthetic data $\boldsymbol{x} \in \mathbb{R}^{d_{\boldsymbol{x}}}$ given a parameter setting $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$. Since the generation process is stochastic, it implicitly defines a likelihood function $\boldsymbol{\theta} \mapsto \pi(\boldsymbol{x}|\boldsymbol{\theta})$ for each $\boldsymbol{x}$. We assume that samples $\boldsymbol{x}$ can be obtained by forward simulation, which we denote by $\boldsymbol{x} \sim \pi(\boldsymbol{x}|\boldsymbol{\theta})$, but the likelihood function $\boldsymbol{\theta} \mapsto \pi(\boldsymbol{x}|\boldsymbol{\theta})$ is intractable and cannot be evaluated. The simulator, combined with a user-specified proper prior $\pi(\boldsymbol{\theta})$ and experimentally observed data $\boldsymbol{x}_{\text{obs}}$, defines an exact posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}}) \propto_{\boldsymbol{\theta}} \pi(\boldsymbol{x}_{\text{obs}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$.

Simulation-based inference: SBI algorithms learn from large collections of synthetic parameter–data pairs generated from the joint distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta})$. In particular, we let $\mathcal{D} := \{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N$ denote the SBI training dataset, where each pair $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ is obtained by first sampling $\boldsymbol{\theta}_n \sim \pi(\boldsymbol{\theta})$ and then generating synthetic data $\boldsymbol{x}_n \sim \pi(\boldsymbol{x}|\boldsymbol{\theta}_n)$ from the simulator conditioned on $\boldsymbol{\theta}_n$. If the joint distribution is sufficiently regular, we might hope to approximate its marginals well using this dataset; e.g., one form of SBI performs what might be seen as regression (with a functional output) to learn the mapping $\boldsymbol{x} \mapsto \pi(\boldsymbol{\theta}|\boldsymbol{x})$. Each SBI method we consider has an amortized (single-round) and sequential (multi-round) variant. The sequential variants use multi-round training schemes in which, rather than training on parameter-data pairs from $\mathcal{D}$ at every round, the posterior approximation $\tilde{\pi}_r(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ obtained at round r is used to influence new training pairs at round $r+1$. Training pairs thus become more concentrated across rounds; the goal is to focus computational effort on regions of the joint space relevant to the observed data $\boldsymbol{x}_{\text{obs}}$. When performing R rounds of sequential training, the posterior approximation is taken to be $\tilde{\pi}_R(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$, the final-round posterior approximation.

SBI methods we consider: The main methodological approaches to neural-network-based SBI (Cranmer et al., 2020) differ in the aspect of the parameter-data relationship they seek to approximate. Neural posterior estimation (NPE, Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Deistler et al., 2022) targets the posterior distribution directly, neural likelihood estimation (NLE, Papamakarios et al., 2019; Glockler et al., 2022) targets the likelihood function, and neural ratio estimation (NRE, Hermans et al., 2020; Durkan et al., 2020; Miller et al., 2022) targets likelihood ratios. Unlike NPE, both NLE and NRE require a conventional Bayesian inference subroutine to generate approximate posterior samples. We consider SNPE-C (Greenberg et al., 2019) and TSNPE (Deistler et al., 2022) as representative NPE methods, NLE-MCMC (Papamakarios et al., 2019) and NLE-VI (Glockler et al., 2022) as representative NLE methods, and SNRE-B (Durkan et al., 2020) as a representative NRE method. We provide further details on each algorithm, together with the rationale for its selection, in Appendix B.

3 Related Work

Parameter dimensionalities considered in studies proposing SBI methods: The original papers proposing key SBI methods include evaluations on a range of models but do not examine the impact of high parameter dimensionality on posterior approximation quality. Instead, many focus

on the challenges posed by high-dimensional data. Table 1 provides a summary of the maximum parameter dimensionalities considered in key SBI papers. For example, Greenberg et al. (2019); Papamakarios et al. (2019); Dirmeier et al. (2025); Hermans et al. (2020); Durkan et al. (2020); Miller et al. (2022) evaluate their methods on simulators with at most 12 parameters, often considerably fewer. Other examples include Deistler et al. (2022); Glockler et al. (2022); Sharrock et al. (2024), who evaluate their algorithms on a 31-parameter neuroscience model (Prinz et al., 2004). However, these experiments were primarily used to compare runtime performance against a competing method, rather than to evaluate posterior approximation quality.

Parameter dimensionalities considered in existing SBI evaluation studies: One of the most widely cited SBI evaluation studies is Lueckmann et al. (2021), who introduced ten benchmark models and compare both sequential and amortized methods. Notably, they found that sequential methods outperformed amortized approaches, particularly on more challenging inference problems. However, all benchmark models considered in their study have at most 10 parameters. A subsequent evaluation study, Hermans et al. (2022), introduced additional benchmark models and evaluated popular SBI methods using the expected coverage metric. However, due to the computational cost of evaluating this metric, which scales with parameter dimensionality, the study was restricted to models with at most 5 parameters. Both studies leave open the question of whether their conclusions extend to higher-dimensional parameter spaces. To our knowledge, the only systematic exploration of SBI performance in high-dimensional parameter spaces is Hashemi et al. (2023), who run SBI on an epilepsy model with close to 100 parameters. They evaluate performance in terms of classification accuracy across varying parameter dimensionalities and other relevant SBI settings. Notably, they evaluate only amortized methods (excluding sequential variants) and report strong performance. A natural question is how the performance gap between amortized and sequential methods evolves in high-dimensional settings. The study of Hashemi et al. (2023) also raises the question of whether the extra expense of sequential training is worthwhile in high parameter dimensions.

4 Experimental Methods

We analyze a simple uniform Gaussian model class and a more complex time series model class. We focus on the former in the main text and leave detailed results for the latter to Appendix E.

Uniform Gaussian model: To investigate the effect of increasing parameter dimensionality, we study a variant of the following uniform Gaussian model: $\boldsymbol{\theta} \sim \text{Unif}[-L, L]^d$ and $\boldsymbol{x}|\boldsymbol{\theta} \sim N(\boldsymbol{\theta}, I_d)$, where $d = d_{\boldsymbol{\theta}} = d_{\boldsymbol{x}}$ denotes the dimensionality of both the data space and the parameter space, L controls the width of the prior, and I_d is the $d \times d$ identity matrix. We condition on observing $\boldsymbol{x}_{\text{obs}} = \mathbf{0}$, so our exact posterior becomes

$$\boldsymbol{\theta}|\{\boldsymbol{x}_{\text{obs}} = \mathbf{0}\} \sim \mathcal{N}_{[-L,L]^d}(\mathbf{0}, I_d), \tag{1}$$

where $\mathcal{N}_{[-L,L]^d}$ denotes a truncated Gaussian distribution on the hypercube $[-L, L]^d$. Namely, our model has a ground truth posterior that can be sampled from using a simple rejection sampler. Finally, we allow $L := L_d$ to depend on d, for reasons that we describe next.

Choice of prior width: In models with high-dimensional parameter spaces, the posterior is often highly concentrated relative to the prior. Recall from Section 2 that SBI methods populate the parameter space with samples and learn from these samples. We therefore expect concentrated posteriors to pose challenges for SBI methods, which are being tasked with locating a "needle" (the posterior mass) within a "haystack" (the prior support). Furthermore, when employing sequential methods in these settings, it is plausible that early-round posterior approximations may fail to locate the narrow region of posterior mass, which would negatively impact future rounds by focusing training simulations on suboptimal regions. In contrast, amortized methods do use intermediate posterior approximations and thus avoid this amplification of early-round errors.

Therefore, we aim to quantify how the posterior concentrates relative to the prior as the parameter dimensionality increases. We expect posterior concentration to increase with increasing d, even

when L is held fixed. In the fixed-L setting, we were unable to obtain a tractable expression for the volume ratio of posterior and prior credible regions. But by allowing $L := L_d$ as a function of d (see definition 1), we were able to characterize posterior concentration analytically, and in fact establish super-exponential decay. In particular, defining the hypersphere $\mathcal{S}_{L_d} := \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq L_d\}$, we prove in Appendix C.1 that $\mathcal{S}_{L_d}$ is a 99% credible[1] region of the exact posterior (Eq. (1)). Moreover, the ratio of the volume of this 99% credible region to the volume of the prior support is:

$$\frac{\mathrm{vol}\left(\mathcal{S}_{L_d}\right)}{\mathrm{vol}([-L_d, L_d]^d)} = \left(\frac{\sqrt{\pi}}{2}\right)^d \frac{1}{\Gamma(1 + d/2)}. \tag{2}$$

which decays to zero super-exponentially fast as $d \to \infty$, as shown in Fig. 4. In fact, we show in Theorem 6 that the volume ratio equals the probability that a parameter sample $\boldsymbol{\theta}_n$ from the SBI training dataset $\mathcal{D}$ lands in the 99% posterior credible region $\mathcal{S}_{L_d}$. In the experiments, we decrease the volume ratio from 0.52 to below 10^{-70} by increasing the dimension d from 3 to 100.

Justifying increasing data dimension: The aim of this experiment is to compare how each SBI method scales with increasing parameter dimensionality d, where we expect performance to degrade due to the decreasing volume ratio (Eq. (2)). Since $d_{\boldsymbol{x}} = d_{\boldsymbol{\theta}} = d$, a reader might be concerned that we confound our experiments by scaling the data dimension with the parameter dimension. We note, though, that existing work suggests data dimensionalities below ∼100 pose little challenge to neural SBI methods (Greenberg et al., 2019; Papamakarios et al., 2019; Durkan et al., 2020). Since we take d up to 100 (which is considered high-dimensional for parameters but not for data), we expect negligible confounding from the data dimensionality.

Evaluating quality: We report the empirical distribution of $\|\boldsymbol{\theta}\|_2^2 := \sum_{i=1}^d \theta_i^2$ for samples $\boldsymbol{\theta}$ drawn from each algorithm's posterior approximation. We compare these distributions visually to the sampling distribution of $\|\boldsymbol{\theta}\|_2^2$ under the exact posterior. The posterior expectation of $\|\boldsymbol{\theta}\|_2^2$ equals the sum over the marginal variances of the posterior, meaning that accurate recovery of the squared norm distribution requires the approximation to accurately quantify uncertainty across dimensions. We discuss evaluation in greater detail in Appendix C.2.

Experimental design: We conduct all experiments in Python 3.9.6, using the `sbi` package (Boelts et al., 2025) to implement all SBI algorithms. We run each sequential method for 8 rounds, with 5,000 simulations per round. We take the amortized (single-round) posterior approximations as $\tilde{\pi}_1(\boldsymbol{\theta}|\mathbf{0})$, namely the first-round posterior approximation from the sequential methods. See Appendix F for further implementation details. Note that we are, if anything, overly generous toward the multi-round variants in this design, since they are trained on 40,000 simulations whereas the single-round variants receive just 5,000.

A more complex posterior geometry: To assess whether our conclusions extend beyond the simple uniform Gaussian model, we also study a linear Gaussian state space model (LGSSM). This model has two global parameters, and we increase the number of local parameters to vary the overall parameter dimensionality. See Appendix E for full model and experimental details.

5 Experimental Results

Single-round training frequently outperforms multi-round training in dimensions above 10: Throughout this section, we focus on the uniform Gaussian model before briefly discussing the LGSSM results at the end. Despite being trained on eight times as many simulations and requiring more runtime (see Appendix D.1), we find that the final approximations from the sequential methods often perform worse than the first-round (i.e. amortized) approximations when $d > 10$. In particular, we see that the approximate posterior mean and quantiles of $\|\boldsymbol{\theta}\|_2^2$ for both NPE methods (the first two columns of Fig. 1) are substantially farther from the ground truth in the sequential

1. We define a $100(1-\alpha)$% credible region as a region that contains at least $1-\alpha$ of the posterior mass.

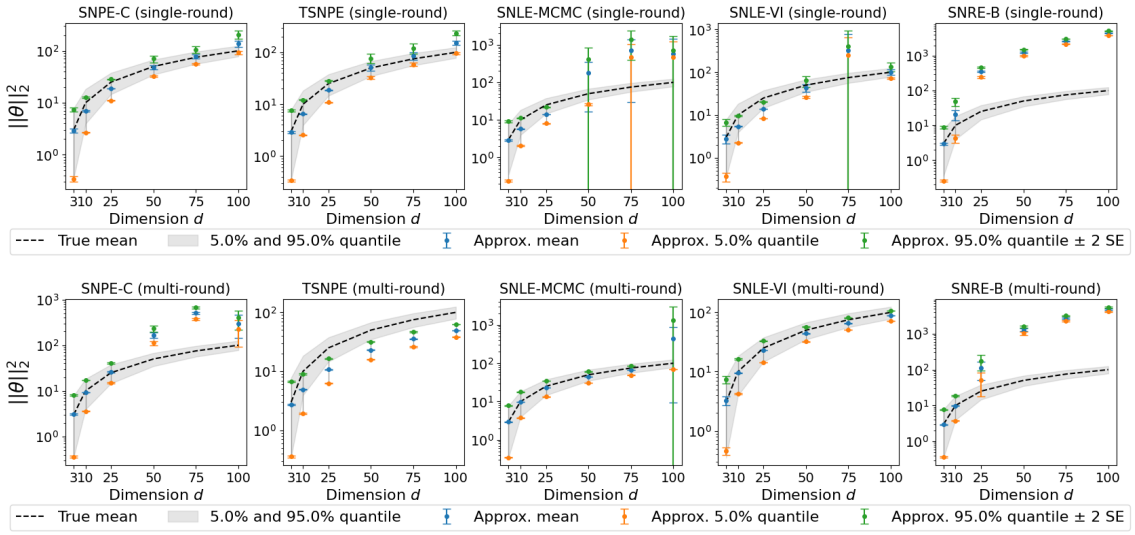

Figure 1. *Upper row*: Amortized (single-round) performance for the uniform Gaussian model. *Lower row*: Sequential (multi-round) performance. Each *column* shows a different SBI method (vertical axes not shared). In each *panel*, we plot the exact posterior mean of $\|\boldsymbol{\theta}\|_2^2$ (dashed black line) and exact 5% to 95% quantile region (gray shading) across parameter dimensions d (horizontal axis: 3, 10, 25, 50, 75, 100). We plot the SBI method's mean approximation (blue dot), 5% quantile (orange dot), and 95% quantile (green dot). Each mean and quantile is averaged over 10 repeated experiments and is accompanied by an error bar showing ± two standard errors. The closer the vertical lines lie to the shaded gray region, the more closely the approximate posterior matches the exact posterior. Vertical lines above/below the shaded region suggest over-/under-dispersion, respectively. Refer to Fig. 4 to convert values of d into posterior-prior volume ratios.

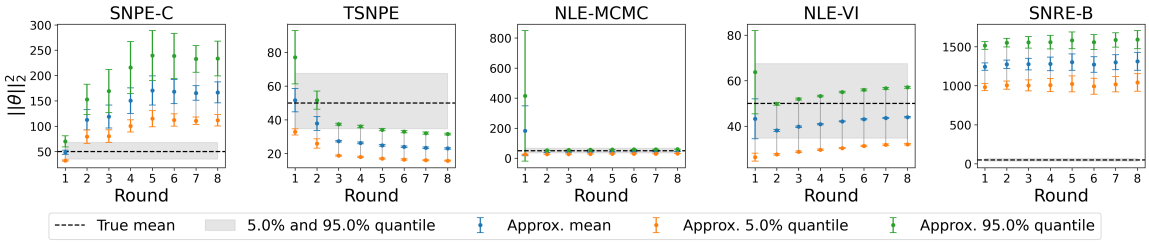

Figure 2. We use the same visual indicators and columns as in Fig. 1, but now each panel shows the performance of the sequential SBI method across rounds (horizontal axis) for a fixed dimensionality $d = 50$.

setting than in the amortized setting, as evidenced by their poorer agreement with the gray shaded region. Amortized NLE-VI achieves somewhat similar performance to its sequential counterpart in dimensions above 10, with the exception of a divergent variational run at $d = 75$ (discussed in more detail later). NLE-MCMC performs worse in the amortized setting. Finally, in dimensions above 25, SNRE-B performs comparably poorly in both settings, showing little evidence of improvement from sequential training.

Our findings coincide with previous low-dimensional studies: As shown in Fig. 1, the sequential methods outperform their amortized counterparts when the parameter dimensionality is below 10 (apart from TSNPE due to the failure mode addressed below). This finding coincides with the existing evaluation studies reviewed in Section 3, which focus on low-dimensional settings and report advantages from sequential training.

TSNPE becomes increasingly under-dispersed across sequential rounds when $d > 3$: The second panel of Fig. 2 shows that the TSNPE posterior approximations become increasingly under-

dispersed across rounds when $d = 50$, as indicated by the vertical lines lying consistently below the gray shaded region after round 1. Fig. 6 shows that this phenomenon occurs in all dimensions $d > 3$. This behaviour appears to arise from the sequential truncation mechanism (see Appendix B.2). In particular, unlike the other sequential methods, TSNPE constructs each proposal distribution by *truncating* the prior to an estimated highest probability region of the posterior approximation from the previous round. Therefore, errors in early-round posterior approximations can cause truncation regions to exclude regions of non-negligible posterior mass. Once excluded, these regions are assigned zero probability by all subsequent proposal distributions and are therefore never revisited by future training simulations. As a result, later rounds concentrate mass into increasingly narrow regions of the parameter space, leading to under-dispersed posterior approximations.

Extra rounds of SNPE-C degrade performance when $d > 25$: We observe in the left row of Fig. 1 that SNPE-C performs substantially worse than its amortized counterpart in dimensions above 25. By $d = 50$, Fig. 2 shows that the SNPE-C posterior approximations become increasingly over-dispersed across sequential rounds, with performance plateauing at a poor level by round 4. Moreover, Fig. 6 indicates that this failure mode becomes more severe with dimensionality; by $d = 75$, performance degrades monotonically across rounds, with no evidence of plateau or recovery.

NLE-MCMC and NLE-VI exhibit instabilities: The amortized and sequential variants of NLE (using either MCMC or VI subroutines) show comparable performance across dimensions. We observe a few exceptions in which the NLE algorithm becomes unstable ($d = 100$ for multi-round with MCMC, $d \geq 50$ for single-round with MCMC, $d = 75$ for single-round with VI), which manifest as inflated error bars in Fig. 1. A closer inspection of the experimental runs confirms that, in each case, at most one or two out of the ten repetitions produce extremely large squared norms, over an order of magnitude larger than expected from the true distribution, while the other runs appear well-behaved. We initially hypothesized that the NLE-MCMC instabilities observed in Fig. 1 may be caused by the axis-aligned slice sampling procedure used to sample from the approximate NLE posterior. To test this, we re-ran the experiment using the No-U-Turn Sampler (NUTS, Hoffman and Gelman, 2014), which is known to perform robustly in high-dimensional settings. We compare the results to those obtained using slice sampling in Fig. 7. We find that the instabilities observed in amortized NLE-MCMC at $d = 75$ and $d = 100$ disappear when using NUTS instead of slice sampling, although the other instabilities persist. Combined with the similar failure of NLE-VI at $d = 75$, these results suggest that irregularities in the learned surrogate likelihood produced by NLE may induce instabilities even in otherwise robust, state-of-the-art MCMC/VI subroutines.

Linear Gaussian state space model: In Appendix E, our experiments for the LGSSM reach similar conclusions: sequential methods often underperform their amortized counterparts in parameter dimensionalities beyond those seen in the previous studies. We also find similar results when we give the amortized and sequential methods the same overall simulation budget.

6 Conclusions and Future Work

Our central finding is that **for models with high-dimensional parameter spaces, sequential SBI methods often produce less accurate posterior approximations than their amortized counterparts**. We most frequently observed this phenomenon in the sequential neural posterior estimation methods SNPE-C and TSNPE, which is noteworthy given that SNPE-C is one of the most widely used sequential SBI approaches (SBI Developers, 2024a), closely followed by TSNPE.

Future work could build upon our experiments by disentangling the effects of posterior concentration from those of parameter dimensionality, which were coupled in our experiments. This would allow us to determine whether posterior concentration is responsible for the observations in Section 5, or whether they are instead driven by other factors, such as raw parameter dimensionality or quantities that scale with it. Other future work could repeat our experiments with a larger number of independent runs to better quantify the frequency and prevalence of the instabilities observed in NLE, as well as the similar instabilities observed for SNPE-C and TSNPE in Appendix E.

Acknowledgments

This work was supported in part by the NSF TRIPODS program (award DMS-2022448). RJR received funding from the European Union under the GA 101071601, through the 2023–2029 ERC Synergy grant OCEAN. We also thank Iain Murray and Dennis Prangle for helpful discussions and feedback throughout this work.

References

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017.

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, 2025.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, May 2020.

Michael Deistler, Pedro J Goncalves, and Jakob H Macke. Truncated proposals for scalable and hassle-free simulation-based inference. *Advances in Neural Information Processing Systems*, 35: 23135–23149, 2022.

Michael Deistler, Jan Boelts, Peter Steinbach, Guy Moss, Thomas Moreau, Manuel Gloeckler, Pedro LC Rodrigues, Julia Linhart, Janne K Lappalainen, Benjamin Kurt Miller, et al. Simulation-based inference: A practical guide. *arXiv preprint arXiv:2508.12939*, 2025.

Simon Dirmeier, Carlo Albert, and Fernando Perez-Cruz. Simulation-based inference for high-dimensional data using surjective sequential neural likelihood estimation. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025.

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

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, pages 2771–2781. PMLR, 2020.

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, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11098–11116. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/geffner23a.html.

Manuel Glockler, Michael Deistler, and Jakob H. Macke. Variational methods for simulation-based inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=kZ0UYdhqkNY.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR, 2019.

Meysam Hashemi, Anirudh N. Vattikonda, Jayant Jha, Viktor Sip, Marmaduke M. Woodman, Fabrice Bartolomei, and Viktor K. Jirsa. Amortized Bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators. *Neural Networks*, 163:178–194, 2023. URL https://www.sciencedirect.com/science/article/pii/S0893608023001752.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pages 4239–4248. PMLR, 2020.

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*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=LHAbHkt6Aq.

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(1):1593–1623, 2014.

Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, and Nicolas Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 2015.

Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/addfa9b7e234254d26e9c7f2af1005cb-Paper.pdf.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, pages 343–351. PMLR, 2021.

Vikash Mansinghka, Tejas Kulkarni, Yura Perov, and Joshua Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. *Advances in Neural Information Processing Systems*, 06 2013.

Benjamin Kurt Miller, Christoph Weniger, and Patrick Forré. Contrastive neural ratio estimation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=kOIaB1hzaLe.

Guy Moss, Vjeran Višnjević, Olaf Eisen, Falk M Oraschewski, Cornelius Schröder, Jakob H Macke, and Reinhard Drews. Simulation-based inference of surface accumulation and basal melt rates of an antarctic ice shelf from isochronal layers. *Journal of Glaciology*, 71:e44, 2025.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd international conference on artificial intelligence and statistics*, pages 837–848. PMLR, 2019.

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.

Astrid A Prinz, Dirk Bucher, and Eve Marder. Similar network activity from disparate circuit parameters. *Nature neuroscience*, 7(12):1345–1352, 2004.

Walter Rudin. Real and complex analysis. 1974.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

SBI Developers. API of implemented methods. https://sbi.readthedocs.io/en/latest/tutorials/16_implemented_methods.html, 2024a. Accessed: 2026-03-20.

SBI Developers. What should I do when my 'posterior samples are outside the prior support' in SNPE-C? https://sbi.readthedocs.io/en/latest/faq/question_01_leakage.html, 2024b. Accessed: 2026-03-20.

Chad Schafer and Peter Freeman. Likelihood-free inference in cosmology: Potential for the estimation of luminosity functions. *Lecture Notes in Statistics*, 209:3–19, 05 2012. doi: 10.1007/978-1-4614-3520-4-1.

Louis Sharrock, Jack Simons, Song Liu, and Mark Beaumont. Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models, 2024. URL https://arxiv.org/abs/2210.04872.

Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. *Advances in Neural Information Processing Systems*, 36:16837–16864, 2023.

Appendix A. Parameter Dimensionalities of Existing Studies

In Table 1, we show the maximum parameter dimensionality considered in accuracy evaluations across a collection of SBI papers.

Paper	Max $d_{\boldsymbol{\theta}}$	Comments
SNPE-A (Papamakarios and Murray, 2016)	6	-
SNPE-B (Lueckmann et al., 2017)	12	-
SNPE-C (Greenberg et al., 2019)	10	-
NLE(-MCMC) (Papamakarios et al., 2019)	12	-
NLE(-VI) (Glockler et al., 2022)	31	For $d_{\boldsymbol{\theta}} = 31$: no amortized & only did PPCs
SSNL (Dirmeier et al., 2025)	10	-
SNRE-A (Hermans et al., 2020)	5	-
SNRE-B (Durkan et al., 2020)	5	-
SNRE-C (Miller et al., 2022)	10	-
TSNPE (Deistler et al., 2022)	31	For $d_{\boldsymbol{\theta}} = 31$: no amortized & only did PPCs
NPSE (Geffner et al., 2023)	10	-
SNPSE (Sharrock et al., 2024)	31	For $d_{\boldsymbol{\theta}} = 31$: no amortized & only did PPCs
Benchmarking SBI (Lueckmann et al., 2021)	10	Evaluation study
Trust Crisis in SBI (Hermans et al., 2022)	5	Evaluation study

Table 1. Maximum parameter dimensionalities considered in key SBI papers. Existing studies predominantly focus on low- to moderate-dimensional parameter spaces, with limited analysis of how performance scales as parameter dimensionality increases.

Appendix B. Neural Simulation-Based Inference

Neural SBI methods learn from large collections of synthetic parameter–data pairs generated from the joint distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta})$. In particular, we let $\mathcal{D} := \{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^{N}$ denote the SBI training dataset, where each pair $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ is obtained by first sampling $\boldsymbol{\theta}_n \sim \pi(\boldsymbol{\theta})$ and then generating synthetic data $\boldsymbol{x}_n \sim \pi(\boldsymbol{x}|\boldsymbol{\theta}_n)$ from the simulator conditioned on $\boldsymbol{\theta}_n$, yielding samples from the joint distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta})$. Intuition from ABC tells us that $\mathcal{D}$ contains information about the posterior distribution: parameter settings $\boldsymbol{\theta}_n$ that generate synthetic data $\boldsymbol{x}_n$ similar to the observed dataset $\boldsymbol{x}_{\text{obs}}$ should receive high posterior probability.

Neural SBI algorithms train neural networks to learn the distributional relationship between parameters and data encoded by $\mathcal{D}$. The main methodological approaches to neural SBI (Cranmer et al., 2020) differ in the aspect of this relationship they seek to approximate. Neural posterior estimation (NPE, Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Deistler et al., 2022) targets the posterior distribution directly, neural likelihood estimation (NLE, Papamakarios et al., 2019; Glockler et al., 2022) targets the likelihood function, and neural ratio estimation (NRE, Hermans et al., 2020; Durkan et al., 2020; Miller et al., 2022) targets likelihood ratios. These algorithms use neural networks to approximate their target densities (or density ratios in the case of NRE). However, neural networks cannot directly be used to represent probability densities, since valid densities must be non-negative and integrate to one; the latter constraint being difficult to enforce for an arbitrary neural network. To address this, many neural SBI methods employ conditional density estimators, which we introduce in Appendix B.1.

B.1 Conditional Density Estimators

A key challenge in probabilistic machine learning is how to parameterize probability distributions using neural networks. While neural networks are highly flexible function approximators, their

outputs do not define valid probability densities, which must be non-negative and integrate to one. Constructing neural-network-based probabilistic models therefore requires architectures that enforce these constraints while retaining the expressivity of neural networks.

A conditional density estimator is a learnable function that represents the probability distribution over a target variable given a conditioning variable. In the context of neural posterior estimation, the target variable is $\boldsymbol{\theta}$ and the conditioning variable is $\boldsymbol{x}$, yielding a conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ with learnable parameters ϕ. These learnable parameters are optimized using gradient descent (Bishop and Nasrabadi, 2006) to make $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ approximate the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. In the case of neural likelihood estimation, the same construction is used with $\boldsymbol{\theta}$ and $\boldsymbol{x}$ reversed, resulting in an approximation $q_\phi(\boldsymbol{x}|\boldsymbol{\theta})$ to the (intractable) likelihood function $\pi(\boldsymbol{x}|\boldsymbol{\theta})$. In both cases, inference reduces to optimizing the parameters ϕ of the conditional density estimator.[2]

Conditional normalizing flows (CNFs, Papamakarios et al., 2021) are among the most widely used conditional density estimators in modern SBI. A CNF is defined as the pushforward of a simple base distribution through a learnable diffeomorphic[3] map. The map acts on the space of the target variable and is parameterized by a neural network that takes the conditioning variable as additional input. CNFs can model complex conditional distributions while admitting efficient sampling and tractable density evaluation via the change of variables formula[4] (Rudin, 1974). Moreover, when the mapping is sufficiently flexible, CNFs are universal density approximators (a proof can be found in Section 2.2 of Papamakarios et al. (2021)). Owing to their expressiveness, efficient sampling procedures, and tractable density evaluation, CNFs have become the default family for conditional density estimators in many neural SBI algorithms. Popular CNF variants used in SBI include masked autoregressive flows (Papamakarios et al., 2017, 2019; Papamakarios and Murray, 2016) and neural spline flows (Durkan et al., 2019). For a comprehensive overview of normalizing flows, we refer the reader to Papamakarios et al. (2021).

B.2 Neural Posterior Estimation

Neural Posterior Estimation (NPE) methods (Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Deistler et al., 2022; Geffner et al., 2023; Sharrock et al., 2024) comprise a suite of SBI methods that train a conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ to approximate the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ directly. We consider two popular NPE methods, SNPE-C (Greenberg et al., 2019) and TSNPE (Deistler et al., 2022), both of which have amortized (single-round) and sequential (multi-round) variants which we introduce below.

Amortized NPE trains the density estimator using pairs drawn directly from the SBI training dataset $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N$. The objective functions differ between SNPE-C and TSNPE, and we introduce them after presenting their sequential variants, from which the objectives are most naturally motivated. Since the training dataset $\mathcal{D}$ consists of parameter-data pairs drawn from the joint distribution $\pi(\boldsymbol{x}, \boldsymbol{\theta})$, amortized training exposes the density estimator to a diverse collection of pairs $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ distributed throughout the full support of the model. As a result, the amortized posterior approximation $\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{x})$ derived from[5] the learned density estimator $q_{\phi^*}(\boldsymbol{\theta}|\boldsymbol{x})$ typically performs reasonably well when conditioned on any observation $\boldsymbol{x}$ supported by the prior predictive distribution, since it has been trained on observations sampled from that distribution. This is advantageous when rapid posterior inference is required for many different observed datasets, but is often considered computationally wasteful when the goal is accurate inference for a specific dataset $\boldsymbol{x}_{\text{obs}}$, since the

2. We remark that neural ratio estimation trains a neural network classifier to estimate likelihood ratios directly, and therefore does not rely on conditional density estimators (see Appendix B.4).
3. A diffeomorphic map is an invertible mapping that is continuously differentiable and whose inverse is also continuously differentiable.
4. The change of variables formula allows one to obtain the density of the complex probability distribution in terms of the base density and the Jacobian determinant of the inverse mapping.
5. The density estimator is not always equal to the posterior approximation. For example, when the prior has bounded support, it is often necessary to truncate the density estimator to the support of the prior, yielding $\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{x}) \propto q_{\phi^*}(\boldsymbol{\theta}|\boldsymbol{x}) \cdot \mathbb{I}\{\boldsymbol{\theta} \in \text{supp}(\pi(\boldsymbol{\theta}))\}$.

amortized density estimator gets trained in regions of the data space that are far from $\boldsymbol{x}_{\text{obs}}$ (Cranmer et al., 2020; Papamakarios and Murray, 2016).

Sequential NPE algorithms aim to improve computational efficiency by iteratively concentrating training pairs in regions of the joint space relevant to the observed dataset $\boldsymbol{x}_{\text{obs}}$. These methods use multi-round training schemes in which, rather than training q_ϕ on parameter-data pairs from $\mathcal{D}$ at every round, the posterior approximation $\tilde{\pi}_r(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ obtained at round r is used to influence new training pairs at round $r+1$. The nature of this influence differs between the two NPE methods we consider (see below). As a result of the increasingly concentrated proposals, computational effort is focused on training the density estimator in regions of high posterior mass, rather than being spent on regions that are unlikely to explain the observed data. When performing R rounds of sequential NPE, the posterior approximation is taken to be $\tilde{\pi}_R(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$, the final-round posterior approximation.

Among the most widely used sequential NPE methods is sequential SNPE-C (Greenberg et al., 2019; SBI Developers, 2024a), which takes the proposal distribution for generating training parameters $\boldsymbol{\theta}_n$ at each round to equal the posterior approximation from the previous round.[6] However, by generating training pairs using proposal distributions that differ from the prior, standard objectives such as maximum likelihood are no longer guaranteed to yield a density estimator that converges to the exact posterior (see Section 2.3 from Greenberg et al. (2019)). SNPE-C addresses this issue through a classification-based training objective: given N training pairs $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ from *any* distribution, SNPE-C derives its learning signal from N multiple-choice questions of the form: "Which parameter among $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$ generated the observation $\boldsymbol{x}_n$?" for $n = 1, ..., N$. For each training pair $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$, the correct answer is $\boldsymbol{\theta}_n$, whereas all other candidate parameters $\{\boldsymbol{\theta}\}_{\neq n}$ are treated as negative examples since they did not generate $\boldsymbol{x}_n$. Using a clever reparameterization of the density estimator q_ϕ, the authors develop the following cross-entropy-based objective function

$$\phi^* = \arg\min_\phi \left[-\sum_{n=1}^{N} \log \left(\frac{q_\phi(\boldsymbol{\theta}_n|\boldsymbol{x}_n)/\pi(\boldsymbol{\theta}_n)}{\sum_{j=1}^{N} q_\phi(\boldsymbol{\theta}_j|\boldsymbol{x}_n)/\pi(\boldsymbol{\theta}_j)} \right) \right] \tag{3}$$

Interestingly, Greenberg et al. (2019) show that this inherently *discrete* objective can be used to train the *continuous* conditional density estimator to asymptotically converge to the exact posterior under regularity assumptions (see Appendix A.4 from Greenberg et al. (2019)). The full SNPE-C algorithm is presented in Appendix A.2 of Greenberg et al. (2019).

Unlike maximum-likelihood training, the SNPE-C objective Eq. (3) does not constrain the amount of mass the density estimator q_ϕ assigns within the support of bounded priors. Therefore, SNPE-C is susceptible to a failure mode known as *leakage* (Greenberg et al., 2019; Deistler et al., 2022; Durkan et al., 2020; Glockler et al., 2022), whereby the density estimator assigns increasing amounts of mass to regions outside the prior support. Since SNPE-C must truncate q_ϕ within the prior support to generate samples, leakage can cause prohibitively long runtimes.

When leakage becomes problematic, sequential TSNPE (Deistler et al., 2022) is a commonly used alternative designed to mitigate this issue. Sequential TSNPE takes the proposal distribution at round $r+1$ to be the prior distribution $\pi(\boldsymbol{\theta})$ truncated to the $1-\epsilon$ highest probability region[7] (HPR) of the round r posterior approximation $\tilde{\pi}_r(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$, for some small $\epsilon > 0$.[8] Like sequential SNPE-C, this focuses training pairs into regions relevant to the observed dataset $\boldsymbol{x}_{\text{obs}}$, but does so through a different proposal mechanism. Unlike SNPE-C, which requires a classification-based training objective to ensure asymptotic convergence to the exact posterior, TSNPE can be trained

6. As before, the corresponding training simulations $\boldsymbol{x}_n$ are generated from the simulator conditioned on $\boldsymbol{\theta}_n$.
7. The $1-\epsilon$ highest probability region of a probability distribution is the smallest region that contains $1-\epsilon$ of its mass.
8. The authors found that $\epsilon = 10^{-4}$ worked well for all benchmark problems they tested (Deistler et al., 2022).

using the maximum likelihood objective $\sum_{n=1}^{N} \log q_\phi(\boldsymbol{\theta}_n|\boldsymbol{x}_n)$ at each round. Furthermore, Deistler et al. (2022) prove in (their) Appendix 6.2 that the density estimator asymptotically converges to the exact posterior under this sequential procedure. The full TSNPE algorithm is presented in Algorithm 1 of Deistler et al. (2022).

Difference between amortized TSNPE and amortized SNPE-C: Recall that, for a given SBI method, the amortized version corresponds to the sequential version with a single round, in which training pairs are drawn directly from the SBI training dataset $\mathcal{D}$ rather than being refined using intermediate proposal distributions. Therefore, the amortized variants of SNPE-C and TSNPE differ only through their training objectives. Amortized TSNPE trains the density estimator q_ϕ using maximum-likelihood, and is the standard training procedure used in amortized NPE. In contrast, amortized SNPE-C trains the density estimator q_ϕ using the classification-based objective Eq. (3). This classification objective is required to overcome theoretical challenges arising in the multi-round setting, but is not necessary in the amortized case. Nevertheless, we consider both in our experiments, but emphasize that amortized TSNPE is used almost exclusively instead of amortized SNPE-C in practice (Deistler et al., 2025).

B.3 Neural Likelihood Estimation

In contrast to NPE, neural likelihood estimation (NLE) methods (Papamakarios et al., 2019; Glockler et al., 2022; Dirmeier et al., 2025) train a conditional density estimator $q_\phi(\boldsymbol{x}|\boldsymbol{\theta})$ to approximate the (intractable) likelihood function $\pi(\boldsymbol{x}|\boldsymbol{\theta})$. Likelihood-based Bayesian inference algorithms such as MCMC (Hoffman and Gelman, 2014) or VI (Blei et al., 2017) can then be employed to generate posterior samples from the Bayesian model that pairs this approximate likelihood $q_\phi(\boldsymbol{x}|\boldsymbol{\theta})$ with the prior. To be precise, this MCMC/VI subroutine is tasked with sampling from the following posterior distribution

$$\frac{q_{\phi^*}(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_\Theta q_{\phi^*}(\boldsymbol{x}|\boldsymbol{\theta}')\pi(\boldsymbol{\theta}')d\boldsymbol{\theta}'}$$

which is itself an approximation to the exact posterior.[9] We refer to the MCMC and VI variants of NLE as NLE-MCMC (Papamakarios et al., 2019) and NLE-VI (Glockler et al., 2022), respectively.

Amortized NLE begins by training a density estimator $q_\phi(\boldsymbol{x}|\boldsymbol{\theta})$ to approximate the likelihood function $\pi(\boldsymbol{x}|\boldsymbol{\theta})$. It achieves this by optimizing the maximum likelihood objective $\sum_{n=1}^{N} \log q_\phi(\boldsymbol{x}_n|\boldsymbol{\theta}_n)$ with respect to ϕ, where the training pairs $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ are taken from $\mathcal{D}$. It is straightforward to show that the resulting density estimator asymptotically converges to the true likelihood (see Section 3 of Papamakarios et al. (2019)). The trained likelihood approximation $q_{\phi^*}(\boldsymbol{x}|\boldsymbol{\theta})$ is then used within MCMC or VI to obtain approximate posterior samples.

Sequential NLE, analogously to sequential NPE, uses multiple rounds to concentrate training pairs in regions of the joint space that are informative for the observed dataset $\boldsymbol{x}_{\text{obs}}$. Like SNPE-C, the parameter proposal distribution is taken to equal the posterior approximation from the previous round (which must now be sampled from using MCMC or VI). Similarly to TSNPE, the likelihood approximations at each round can be trained using maximum likelihood without the need for a modified objective (see Section 3 of Papamakarios et al. (2019)). The complete sequential NLE-MCMC algorithm is outlined in Algorithm 1 of Papamakarios et al. (2019), while the sequential NLE-VI algorithm is described in Algorithm 1 of Glockler et al. (2022). The amortized versions correspond to setting $R = 1$ in both algorithms.

9. We note that the subroutine targets an *approximation* to the exact posterior, since the learned likelihood $q_{\phi^*}(\boldsymbol{x}|\boldsymbol{\theta})$ does not equal the true likelihood $\pi(\boldsymbol{x}|\boldsymbol{\theta})$ in general.

B.4 Neural Ratio Estimation

Neural Ratio Estimation (NRE) methods (Hermans et al., 2020; Durkan et al., 2020; Miller et al., 2022) train neural network classifiers to learn density ratios of the Bayesian model, bypassing the need for conditional density estimators. We consider SNRE-B (Durkan et al., 2020) as the NRE method in our experiments, as recommended by the `sbi` Python package (see Appendix B.5). SNRE-B trains a neural network classifier[10] $f_\phi(\boldsymbol{x}, \boldsymbol{\theta})$ by minimizing the cross entropy loss function

$$\phi^* = \arg\min_\phi \left[-\sum_{n=1}^{N} \log \left(\frac{\exp\left(f_\phi(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\right)}{\sum_{j=1}^{N} \exp\left(f_\phi(\boldsymbol{x}_n, \boldsymbol{\theta}_j)\right)} \right) \right] \tag{4}$$

where the training pairs $(\boldsymbol{x}_n, \boldsymbol{\theta}_n)$ are generated by sampling $\boldsymbol{\theta}_n$ from any proposal distribution, followed by $\boldsymbol{x}_n \sim \pi(\boldsymbol{x}|\boldsymbol{\theta}_n)$. Durkan et al. (2020) show that this objective results in f_ϕ learning the log density ratio between the posterior and prior

$$f_{\phi^*}(\boldsymbol{x}, \boldsymbol{\theta}) = \log \left(\frac{\pi(\boldsymbol{\theta}|\boldsymbol{x})}{\pi(\boldsymbol{\theta})} \right) + c(\boldsymbol{x}) \tag{5}$$

up to an intractable normalizing constant $c(\boldsymbol{x})$ that arises from the fact that the cross entropy loss Eq. (4) is invariant to shifts $f_\phi \mapsto f_\phi + c(\boldsymbol{x})$. Rearranging Eq. (5) yields an unnormalized posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x}) \propto_{\boldsymbol{\theta}} \exp\left(f_{\phi^*}(\boldsymbol{x}, \boldsymbol{\theta})\right) \pi(\boldsymbol{\theta})$ that can be sampled from using likelihood-based Bayesian inference algorithms. In particular, the authors suggest using MCMC to sample from this unnormalized posterior, and we therefore follow this approach in our experiments.

Equivalence between SNRE-B and SNPE-C: The cross-entropy loss Eq. (4) can be interpreted as training the classifier f_ϕ to solve a set of N multiple-choice questions of the form: "Which parameter among $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$ generated the observation $\boldsymbol{x}_n$?" for $n = 1, ..., N$. It is therefore intimately related to the training objective Eq. (3) used in SNPE-C, which derives its learning signal from the same set of questions (see Appendix B.2). In fact, Durkan et al. (2020) show that SNRE-B and SNPE-C are both special cases of a unified contrastive learning framework (see Algorithm 1 therein). In particular, they observed that parameterizing the SNRE-B classifier $f_\phi(\boldsymbol{x}, \boldsymbol{\theta})$ in terms of a conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ as follows:

$$f_\phi(\boldsymbol{x}, \boldsymbol{\theta}) = \log \left(\frac{q_\phi(\boldsymbol{\theta}|\boldsymbol{x})}{\pi(\boldsymbol{\theta})} \right) \tag{6}$$

recovers the SNPE-C algorithm from Appendix B.2. To see this, simply replace $f_\phi(\boldsymbol{x}, \boldsymbol{\theta})$ with $\log\left(q_\phi(\boldsymbol{\theta}|\boldsymbol{x})/\pi(\boldsymbol{\theta})\right)$ in the SNRE-B objective Eq. (4) and observe that this recovers the SNPE-C objective Eq. (3). Thus, SNRE-B and SNPE-C can be viewed as two instantiations of the same underlying contrastive learning framework: SNRE-B learns a log density ratio via a classifier f_ϕ and subsequently constructs an unnormalized posterior which can be normalized using MCMC, whereas SNPE-C reparameterizes this classifier using Eq. (6) such that it approximates the posterior directly using a conditional density estimator q_ϕ. The resulting training objectives are mathematically equivalent under this reparameterization Eq. (6).

Amortized and sequential SNRE-B: Due to the equivalence outlined above, we omit a separate discussion of the amortized and sequential variants of SNRE-B to avoid repetition. Instead, we refer the reader to the corresponding SNPE-C variants in Appendix B.2, as well as Algorithm 1 in Durkan et al. (2020), which provides a unified framework for both methods.

10. Strictly speaking, $f_\phi(\boldsymbol{x}, \boldsymbol{\theta})$ is a real-valued neural network that outputs unnormalized scores (logits). The actual multiclass classifier is obtained by applying a softmax over these scores across the candidate parameter set $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$ (see Eq. (4)), yielding a discrete probability distribution over which parameter generated a given observation $\boldsymbol{x}$. The function f_ϕ should therefore be interpreted as a learned scoring function, rather than a probability-valued classifier. The literature, however, frequently refers to f_ϕ as a classifier, and we maintain this convention.

B.5 Choice of Benchmark Algorithms

The `sbi` Python package (Boelts et al., 2025) has emerged as one of the most widely used software libraries for simulation-based inference among both researchers and practitioners. The package provides amortized and sequential implementations of popular NPE, NLE, and NRE variants and highlights those that are currently regarded as state of the art. To ensure that our empirical study reflects common practitioner workflows, we focus on the methods recommended and supported by `sbi`. For NPE, SNPE-C is recommended (Greenberg et al., 2019), but SBI Developers (2024b) advise using TSNPE (Deistler et al., 2022) if SNPE-C fails, so we consider both. For NLE, the package defaults to MCMC for posterior sampling, but recommends switching to VI in cases where MCMC becomes prohibitively slow. We therefore consider both NLE-MCMC (Papamakarios et al., 2019) and NLE-VI (Glockler et al., 2022). For NRE, we consider SNRE-B (Durkan et al., 2020), the package's recommended method, together with its default MCMC subroutine for posterior sampling. Together, these methods constitute the primary SBI algorithms exposed to users of the `sbi` package and therefore represent common practitioner workflows. To ensure that our evaluation reflects typical usage, we use each method's default settings (e.g. hyperparameter settings and neural network architectures) unless stated otherwise.

Recently, alternative approaches based on flow matching and diffusion models have been proposed for SBI. While promising, these approaches either do not support sequential inference (Wildberger et al., 2023) or are at an early stage of development (Geffner et al., 2023; Sharrock et al., 2024). For this reason, we focus our empirical evaluation on the more established methods outlined above, leaving an investigation of these newer approaches as future work.

Appendix C. Further Details on the Uniform Gaussian Model

The goal of this section is to expand upon the uniform Gaussian model used in our experiments.

C.1 Deriving the Volume Ratio

We begin by defining the width L_d of the uniform prior $\text{Unif}[-L_d, L_d]^d$ such that the hypersphere $\mathcal{S}_{L_d}$ of radius L_d is a 99% credible region for the exact posterior $\mathcal{N}_{[-L_d,L_d]^d}(\mathbf{0}, I_d)$. To achieve this in a clean and tractable way, we choose L_d as an increasing function of the dimension d (see definition 1) such that $\mathcal{S}_{L_d}$ contains 99% of the mass of the *untruncated* version of the posterior, $\mathcal{N}(\mathbf{0}, I_d)$. Because this hypersphere lies within the prior support $[-L_d, L_d]^d$ (see Lemma 2), it follows that $\mathcal{S}_{L_d}$ also contains at least 99% of the mass of the exact posterior (see Theorem 4). This construction allows us to compute the ratio Eq. (2) of the volume of $\mathcal{S}_{L_d}$ divided by the volume of the prior support (see Lemma 5). In fact, since the prior is uniform, this volume ratio equals the probability of a prior sample $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$ landing in the 99% posterior credible region $\mathcal{S}_{L_d}$ (see Theorem 6).

Definition 1. *For any $d \in \mathbb{N}^{>0}$, we define the width of the prior as $L_d := \sqrt{F^{-1}_{\chi^2_d}(0.99)}$, where $F^{-1}_{\chi^2_d}$ denotes the inverse CDF of the χ^2_d distribution.*

A plot of L_d against d is shown in Fig. 3. The rationale for this choice of prior width will become clear in the subsequent discussion.

Lemma 2. *For any $d \in \mathbb{N}^{>0}$ and any $L > 0$, the hypersphere $\mathcal{S}_L := \{\boldsymbol{\theta} \in \mathbb{R}^d : ||\boldsymbol{\theta}||_2 \leq L\}$ is a subset of the hypercube $[-L, L]^d$. Formally, $\mathcal{S}_L \subseteq [-L, L]^d$.*

Proof.

$$\underbrace{||\boldsymbol{\theta}||_2 \leq L}_{\boldsymbol{\theta} \text{ in hypersphere } \mathcal{S}_L} \implies \sum_{i=1}^{d} \theta_i^2 \leq L^2 \implies \forall i, \ |\theta_i| \leq L \implies \underbrace{\boldsymbol{\theta} \in [-L, L]^d}_{\boldsymbol{\theta} \text{ in hypercube } [-L,L]^d}$$

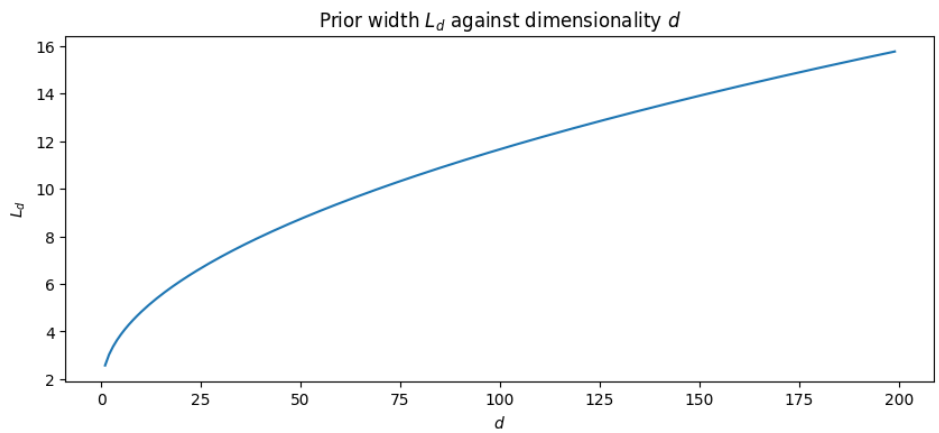

Figure 3. Prior width $L_d = \sqrt{F^{-1}_{\chi^2_d}(0.99)}$ against d.

□

Lemma 3. *For any $d \in \mathbb{N}^{>0}$, a sample $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, I_d)$ satisfies $||\boldsymbol{\theta}||_2^2 \sim \chi^2_d$.*

Proof. By definition, $||\boldsymbol{\theta}||_2^2 := \sum_{i=1}^d \theta_i^2$ is a sum of d i.i.d standard Gaussian random variables, which is precisely the definition of a χ^2_d random variable. □

Theorem 4. *For any $d \in \mathbb{N}^{>0}$, let L_d be defined as in definition 1. Then the hypersphere $\mathcal{S}_{L_d} := \{\boldsymbol{\theta} \in \mathbb{R}^d : ||\boldsymbol{\theta}||_2 \leq L_d\}$ is a 99% credible region for the exact posterior Eq. (1).*

Proof. Using Lemma 3, for a sample $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, I_d)$, we can write

$$P\left(||\boldsymbol{\theta}||_2 \leq L_d\right) = P\left(||\boldsymbol{\theta}||_2^2 \leq L_d^2\right) = F_{\chi^2_d}(L_d^2)$$

where $F_{\chi^2_d}$ denotes the CDF of the χ^2_d distribution. Recalling that $L_d = \sqrt{F^{-1}_{\chi^2_d}(0.99)}$, this simplifies to

$$P\left(||\boldsymbol{\theta}||_2 \leq L_d\right) = F_{\chi^2_d}(L_d^2) = F_{\chi^2_d}(F^{-1}_{\chi^2_d}(0.99)) = 0.99$$

Thus, the hypersphere $\mathcal{S}_{L_d}$ contains exactly 99% of the mass of the $\mathcal{N}(\mathbf{0}, I_d)$ distribution:

$$\int_{\mathcal{S}_{L_d}} \phi(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0.99$$

where $\phi(\boldsymbol{\theta})$ denotes the density of a $\mathcal{N}(\mathbf{0}, I_d)$. However, recall that the exact posterior Eq. (1) of our model is a truncated standard Gaussian $\mathcal{N}_{[-L_d, L_d]^d}(\mathbf{0}, I_d)$, meaning its density is a re-normalized version of ϕ:

$$\pi(\boldsymbol{\theta}|\mathbf{0}) = \frac{\phi(\boldsymbol{\theta}) \cdot \mathbb{I}\{\boldsymbol{\theta} \in [-L_d, L_d]^d\}}{\int_{[-L_d, L_d]^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}'}$$

where the (intractable) normalization constant $\int_{[-L_d, L_d]^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}'$ is bounded above by 1, since:

$$[-L_d, L_d]^d \subset \mathbb{R}^d \implies \int_{[-L_d, L_d]^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}' < \int_{\mathbb{R}^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}' = 1$$

This implies that $\mathcal{S}_{L_d}$ is a 99% credible region for the exact posterior Eq. (1), since it contains at least 99% of the mass of the exact posterior:

$$\mathbb{P}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta}|\mathbf{0})}[\boldsymbol{\theta} \in \mathcal{S}_{L_d}] = \int_{\mathcal{S}_{L_d}} \pi(\boldsymbol{\theta}|\mathbf{0}) d\boldsymbol{\theta} = \int_{\mathcal{S}_{L_d}} \frac{\phi(\boldsymbol{\theta}) \cdot \mathbb{I}\{\boldsymbol{\theta} \in [-L_d, L_d]^d\}}{\int_{[-L_d,L_d]^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}'} d\boldsymbol{\theta}$$

$$\overset{\text{Lemma 2}}{=} \int_{\mathcal{S}_{L_d}} \frac{\phi(\boldsymbol{\theta})}{\int_{[-L_d,L_d]^d} \phi(\boldsymbol{\theta}') d\boldsymbol{\theta}'} d\boldsymbol{\theta} > \int_{\mathcal{S}_{L_d}} \phi(\boldsymbol{\theta}) d\boldsymbol{\theta} = 0.99$$

as required.

□

Lemma 5. *For any $d \in \mathbb{N}^{>0}$ and any $L > 0$, the volume of the hypersphere $\mathcal{S}_L := \{\boldsymbol{\theta} \in \mathbb{R}^d : ||\boldsymbol{\theta}||_2 \leq L\}$ divided by the volume of the hypercube $[-L, L]^d$ is:*

$$\left(\frac{\sqrt{\pi}}{2}\right)^d \frac{1}{\Gamma(1 + d/2)}$$

which is independent of L, and converges to zero super-exponentially as $d \to \infty$ (see Figure 4).

Proof.

$$\frac{\text{vol}\left(\mathcal{S}_L\right)}{\text{vol}([-L, L]^d)} = \frac{\text{vol}\left(\{\boldsymbol{\theta} : ||\boldsymbol{\theta}||_2 \leq L\}\right)}{\text{vol}([-L, L]^d)} = \frac{\pi^{d/2} L^d / \Gamma(1 + d/2)}{(2L)^d} = \left(\frac{\sqrt{\pi}}{2}\right)^d \frac{1}{\Gamma(1 + d/2)}$$

where $\text{vol}\left(\mathcal{S}_L\right)$ is computed using the classical formula for the volume of a d-dimensional hypersphere, attributed to Eugène Catalan (1839). □

Theorem 6. *For any $d \in \mathbb{N}^{>0}$, the probability of a prior sample $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$ lying in the 99% credible region $\mathcal{S}_{L_d}$ of the exact posterior Eq. (1) is given by the volume ratio Eq. (2)*

$$\mathbb{P}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left(\boldsymbol{\theta} \in \mathcal{S}_{L_d}\right) = \left(\frac{\sqrt{\pi}}{2}\right)^d \frac{1}{\Gamma(1 + d/2)}$$

Equivalently, the expected number of parameter samples $\boldsymbol{\theta}_n$ from the SBI training dataset $\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N$ lying inside the 99% credible region $\mathcal{S}_{L_d}$ is N times Eq. (2).

Proof. Since the prior is uniform on $[-L_d, L_d]^d$, the probability of a prior sample $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$ lying inside $\mathcal{S}_{L_d}$ is equal to the volume of $\mathcal{S}_{L_d}$ divided by the volume of $[-L_d, L_d]^d$. By Lemma 5, this volume ratio is given by

$$\left(\frac{\sqrt{\pi}}{2}\right)^d \frac{1}{\Gamma(1 + d/2)}$$

as required. □

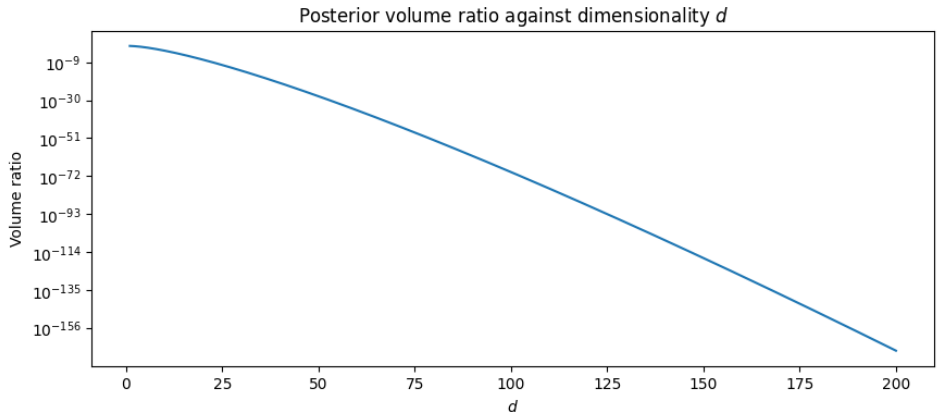

Figure 4. Volume ratio Eq. (2) against d. The vertical axis is on $\log_{10}$ scale.

C.2 Squared Euclidean Norm as an Evaluation Metric

Recall that each SBI algorithm is trained to approximate the exact posterior $\pi(\boldsymbol{\theta}|\mathbf{0})$ (defined in Eq. (1)), and we denote this approximation by $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{0})$. To assess how well these algorithms approximate the exact posterior, we compare the distribution of squared Euclidean norms under the approximate posterior with that under the exact posterior. The purpose of this section is to justify why calibration of squared norms provides a practically meaningful way to evaluate how well the algorithms recover key aspects of the exact posterior.

Why Calibration of Squared Norms Matters in Practice. The expectation of $\|\boldsymbol{\theta}\|_2^2$ for $\boldsymbol{\theta}$ sampled from the true posterior (Eq. (1)) is a sum over the marginal posterior variances:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right] = \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left[\theta_i^2\right] = \sum_{i=1}^{d} \mathrm{Var}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left(\theta_i\right) \tag{7}$$

Therefore, accurately estimating (Eq. (7)) implies that the SBI algorithm accurately estimates the average marginal variance across all dimensions, which is simply $\frac{1}{d}$ times (Eq. (7)). Since practitioners commonly report variances of posterior marginals, accurate recovery of the average posterior variance across all dimensions provides a practically useful indicator of whether an SBI method is quantifying uncertainty accurately. Accurate uncertainty quantification of marginal parameters is paramount for practitioners, since overconfident posterior approximations (i.e. those that underestimate variances) can lead to false scientific discoveries.

Despite being the sum of marginal variances, the squared norm expectation (Eq. (7)) can also detect approximations that learn incorrect means. To see this, suppose the posterior approximation $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{0})$ correctly infers the variances, but learns at least one incorrect mean:

$$\mathrm{Var}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left(\theta_i\right) = \mathrm{Var}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left(\theta_i\right) \qquad \text{for all } i \in \{1, 2, ..., d\}$$

$$\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left(\theta_j\right) \neq \mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left(\theta_j\right) = 0 \qquad \text{for at least one } j \in \{1, 2, ..., d\}$$

In this case, the expectation of the squared norm of a sample $\boldsymbol{\theta} \sim \tilde{\pi}(\boldsymbol{\theta}|\mathbf{0})$ is given by

$$\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right] = \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i^2\right] = \sum_{i=1}^{d} \left[\mathrm{Var}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left(\theta_i\right) + \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i\right]^2\right]$$

$$= \sum_{i=1}^{d} \mathrm{Var}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left(\theta_i\right) + \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i\right]^2 \overset{\text{Correct Var}}{=} \sum_{i=1}^{d} \mathrm{Var}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left(\theta_i\right) + \sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i\right]^2$$

$$\overset{(Eq.\ (7))}{=} \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right]}_{\text{Correct value}} + \underbrace{\sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i\right]^2}_{\text{Error } >0}$$

We see that $\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right]$ differs from the exact expectation $\mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right]$ by an error term $\sum_{i=1}^{d} \mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\theta_i\right]^2$, which is strictly positive because there exists some j where the approximate mean for θ_j is incorrect (i.e. not equal to 0). The derivation above assumes that the approximate variances $\mathrm{Var}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left(\theta_i\right)$ are correct; if this assumption is violated (which is almost always the case in practice), the resulting error is expected to be even larger[11].

Lastly, we note that our squared norm evaluation metric does not merely ensure that the expectation in (Eq. (7)) is correctly calibrated; it assesses whether the entire *distribution* of $\|\boldsymbol{\theta}\|_2^2$ matches the exact distribution. This is a stronger criterion, capable of detecting more subtle discrepancies between the moments of the approximate and exact posteriors.

Appendix D. Further Experimental Results for the Uniform Gaussian Model

D.1 Runtimes

Fig. 5 depicts the total runtime (sampling + training time for all 8 rounds) of each algorithm as a function of dimensionality d.

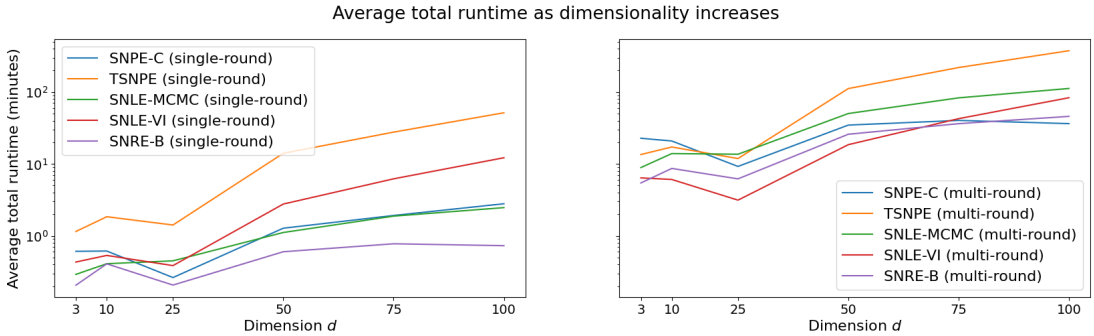

Figure 5. Total runtime of each algorithm (in minutes) as a function of dimensionality d, averaged across 10 repeated experiments. The left panel shows single-round variants, and the right panel shows 8-round variants.

D.2 Squared norms across rounds

Fig. 6 depicts the mean, 0.05 quantile, and 0.95 quantile of the empirical distribution of the squared norms of samples from the approximate posterior at each round of training, for $d \in \{3, 10, 25, 50, 75, 100\}$.

11. Note that it is theoretically possible for errors in the posterior means and variances to offset one another, resulting in $\mathbb{E}_{\boldsymbol{\theta}\sim\tilde{\pi}(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right]$ matching $\mathbb{E}_{\boldsymbol{\theta}\sim\pi(\boldsymbol{\theta})}\left[\|\boldsymbol{\theta}\|_2^2\right]$ by chance. However, such cancellations are highly unlikely and there is no principled reason to expect them to occur in practice.

D.3 Comparing NUTS and Slice Sampling in NLE

Fig. 7 compares amortized and sequential NLE-MCMC using two different MCMC subroutines: axis-aligned slice sampling (two leftmost plots) and NUTS (two rightmost plots).

Appendix E. Linear Gaussian State Space Model

E.1 Experimental Methods

We now turn to the linear Gaussian state space model (LGSSM, see Section E.1.1), which exhibits a more complex and realistic posterior geometry than the uniform Gaussian model introduced in Section 4. In this formulation, two static parameters interact with a sequence of latent state parameters that evolve according to an AR(1) time series process. This induces a local–global parameter structure in which global parameters govern the dynamics of the latent trajectory, leading to strong dependencies both across time and between parameter subsets. Unlike the simple uniform Gaussian model, the posterior mass of the LGSSM is distributed along a curved, high-dimensional manifold induced by the latent state evolution. As a result, inference requires simultaneously recovering global static parameters and local latent states that are tightly coupled through the hierarchical structure. Despite this increased structural complexity, the exact posterior remains fully tractable via the Kalman filter and Rauch–Tung–Striebel smoother (Särkkä and Svensson, 2023), providing exact ground-truth samples that can be used to derive evaluation metrics (see Section E.1.4). We also derive a notion of posterior concentration relative to the prior (see Section E.1.2), analogous to the volume ratios introduced in Section 4 for the uniform Gaussian model.

E.1.1 The Model

We consider the linear Gaussian state space model (LGSSM) as described in Section 7 of Kantas et al. (2015). The model is defined by the following generative process. Given static parameters (ρ, τ), we generate the latent time series trajectory $z_{0:T}$ according to the AR(1) process[12]

$$z_0 \sim N(0, 1)$$

$$z_t \sim N(\rho z_{t-1}, \tau^2) \qquad t = 1, ..., T$$

for some time horizon $T > 0$. Conditional on the latent trajectory, observations $x_{0:T}$ are generated via pointwise Gaussian noise

$$x_t \sim N(z_t, \sigma^2) \qquad t = 0, ..., T$$

for some observation noise $\sigma > 0$ that we assume is fixed and known throughout.

As a Bayesian Model: We treat the LGSSM as a Bayesian model with parameters $\boldsymbol{\theta} = (\rho, \tau, z_{0:T}) \in \mathbb{R}^{T+3}$ and data $\boldsymbol{x} = x_{0:T} \in \mathbb{R}^{T+1}$. This model has parameter dimensionality $d_{\boldsymbol{\theta}} = T+3$, which we vary in our experiments. We place the following hierarchical prior over the Bayesian parameters

$$\pi(\rho, \tau, z_{0:T}) = \pi(\rho)\pi(\tau)\pi(z_{0:T}|\rho, \tau)$$

where the priors over the static parameters are given by

$$\rho \sim \text{Unif}[0, 1]$$

$$\tau \sim N_{[0,2]}(1, 1)$$

12. We use $N(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2, and $\mathcal{N}(x|\mu, \sigma^2)$ to denote the corresponding probability density function evaluated at x.

and the conditional prior for the latent trajectory given the static parameters has an AR(1) factorization

$$\pi(z_{0:T}|\rho,\tau) = \mathcal{N}(z_0|0,1)\prod_{t=1}^{T}\mathcal{N}(z_t|\rho z_{t-1},\tau^2)$$

Since the observations $x_{0:T}$ are generated by adding pointwise Gaussian noise with fixed variance σ^2 to $z_{0:T}$, the likelihood is given by

$$\pi(x_{0:T}|\rho,\tau,z_{0:T}) = \prod_{t=0}^{T}\mathcal{N}(x_t|z_t,\sigma^2)$$

which is independent of the static parameters.

Observed data: We generate a single synthetic observation sequence $x_{0:100}^{\text{obs}}$ from the LGSSM with ground-truth parameters $\rho^* = 0.8$, $\tau^* = 1$, and a fixed observation noise $\sigma = 0.5$. We use four observed datasets in our experiments, corresponding to sections $x_{0:T}^{\text{obs}}$ of this observation sequence $x_{0:100}^{\text{obs}}$ for $T \in \{10, 25, 50, 100\}$. Thus, the dataset for $T = 10$ is a subset of the dataset for $T = 25$, the dataset for $T = 25$ is a subset of the dataset for $T = 50$, and so on. Using nested datasets in this way ensures that differences in SBI approximation quality can be attributed to increasing T rather than to random variation arising from independently generated observation sequences. These four datasets are used in all subsequent LGSSM experiments. For each T, the resulting posterior $\pi(\rho,\tau,z_{0:T}|x_{0:T}^{\text{obs}})$ can be sampled from exactly using the Kalman filter (Särkkä and Svensson, 2023), which we implement in Python using standard libraries. Samples generated by these algorithms serve as ground-truth samples when developing the evaluation metrics in Section E.1.4.

E.1.2 Quantifying Posterior Concentration

The goal of this section is to define a notion of volume ratio for the LGSSM posterior relative to the prior. This serves as a measure of posterior concentration, with values close to zero indicating that the posterior is highly concentrated relative to the prior. Unlike in the uniform Gaussian setting from Section 4, the exact volume ratio is not available in closed form, and must be approximated.

We begin by defining a conditional volume ratio for the posterior of latent states given fixed static parameters, and then average this quantity over the exact posterior of the static parameters to obtain a global concentration metric.

Volume Ratio Conditional on Static Parameters: Conditional on the values of the static parameters (ρ,τ), the LGSSM prior and posterior are both multivariate Gaussian:

$$\pi(z_{0:T}|\rho,\tau) = \mathcal{N}(z_{0:T}|\mu_{\rho,\tau}^{\text{prior}},\Sigma_{\rho,\tau}^{\text{prior}}) \tag{8}$$

$$\pi(z_{0:T}|\rho,\tau,x_{0:T}^{\text{obs}}) = \mathcal{N}(z_{0:T}|\mu_{\rho,\tau}^{\text{posterior}},\Sigma_{\rho,\tau}^{\text{posterior}}) \tag{9}$$

where $\mu_{\rho,\tau}^{\text{prior}}$ and $\Sigma_{\rho,\tau}^{\text{prior}}$ correspond to the mean and covariance matrix of the AR(1) prior process, and $\mu_{\rho,\tau}^{\text{posterior}}$ and $\Sigma_{\rho,\tau}^{\text{posterior}}$ are computed in closed form using the Kalman filter and Rauch–Tung–Striebel smoother (Särkkä and Svensson, 2023). For $\alpha \in (0,1)$, the Gaussianity of the conditional prior Eq. (8) results in an elliptical $100\cdot\alpha\%$ credible region of the form

$$\{z \in \mathbb{R}^{T+1} : (z-\mu_{\rho,\tau}^{\text{prior}})^T(\Sigma_{\rho,\tau}^{\text{prior}})^{-1}(z-\mu_{\rho,\tau}^{\text{prior}}) \le \chi^2_{T+1}(\alpha)\}$$

where $\chi^2_{T+1}(\alpha)$ is the α-quantile of the chi-squared distribution with $T+1$ degrees of freedom. This region has volume given by

$$\frac{\pi^{\frac{T+1}{2}}}{\Gamma\left(\frac{T+1}{2}+1\right)}\left(\chi^2_{T+1}(\alpha)\right)^{\frac{T+1}{2}}\sqrt{\det\left(\Sigma_{\rho,\tau}^{\text{prior}}\right)}$$

One can derive the volume of the $100 \cdot \alpha\%$ credible region of the conditional posterior Eq. (9) by simply replacing $\Sigma_{\rho,\tau}^{\text{prior}}$ with $\Sigma_{\rho,\tau}^{\text{posterior}}$ in the formula above. By dividing the volume of the $100 \cdot \alpha\%$ credible region of the conditional posterior by that of the conditional prior, we obtain the conditional volume ratio:

$$V_{\rho,\tau} := \frac{\frac{\pi^{\frac{T+1}{2}}}{\Gamma(\frac{T+1}{2}+1)} \left(\chi^2_{T+1}(\alpha)\right)^{\frac{T+1}{2}} \sqrt{\det\left(\Sigma_{\rho,\tau}^{\text{posterior}}\right)}}{\frac{\pi^{\frac{T+1}{2}}}{\Gamma(\frac{T+1}{2}+1)} \left(\chi^2_{T+1}(\alpha)\right)^{\frac{T+1}{2}} \sqrt{\det\left(\Sigma_{\rho,\tau}^{\text{prior}}\right)}}$$

$$= \sqrt{\frac{\det(\Sigma_{\rho,\tau}^{\text{posterior}})}{\det(\Sigma_{\rho,\tau}^{\text{prior}})}} \tag{10}$$

which is independent of the choice of α.

Approximate Volume Ratio: The conditional volume ratio $V_{\rho,\tau}$ in Eq. (10) is the volume ratio of the posterior over latent states conditional on a specific (ρ, τ). This only defines a notion of concentration for slices across the latent dimension of the posterior distribution for fixed (ρ, τ). However, we wish to construct a notion of volume ratio on a global scale across the entire posterior $\pi(\rho, \tau, z_{0:T}|x_{0:T}^{\text{obs}})$. To achieve this, we average over (ρ, τ) sampled from the exact posterior $\pi(\rho, \tau|x_{0:T}^{\text{obs}})$

$$V := \mathbb{E}_{(\rho,\tau)\sim\pi(\rho,\tau|x_{0:T}^{\text{obs}})}\left[V_{\rho,\tau}\right] = \int\int V_{\rho,\tau}\pi(\rho,\tau|x_{0:T}^{\text{obs}})d\rho d\tau \approx \frac{1}{M}\sum_{i=1}^{M} V_{\rho_i,\tau_i} \tag{11}$$

where $(\rho_i, \tau_i) \sim \pi(\rho, \tau|x_{0:T}^{\text{obs}})$ for $i = 1, ..., M$. In practice, we use $M = 10{,}000$, as this was found to yield low-variance estimates of the volume ratio Eq. (11).

The top row of Table 2 reports the volume ratios defined in Eq. (11) for the posterior distributions corresponding to the four observed datasets used in our experiments (see Section E.1.1). The other two rows correspond to larger observation noises $\sigma = 1$ and $\sigma = 100$. We observe that the volume ratios decrease with increasing T, indicating that the posterior becomes increasingly concentrated relative to the prior as the dimensionality of the parameter space grows. This behaviour is common in high-dimensional Bayesian inference. We also observe that the volume ratios decrease with decreasing observation noise σ, which is expected since a low observation noise produces a highly informative likelihood that tightly constrains the posterior. For the large observation noise $\sigma = 100$, the volume ratio is approximately 1 for all T, indicating that the posterior closely resembles the prior.

	T			
Observation noise σ	10	25	50	100
0.5	$\mathbf{7.92 \times 10^{-4}}$	$\mathbf{1.16 \times 10^{-9}}$	$\mathbf{1.76 \times 10^{-16}}$	$\mathbf{3.05 \times 10^{-33}}$
1	0.0936	2.27×10^{-3}	7.69×10^{-8}	2.14×10^{-13}
100	0.999	0.997	0.993	0.984

Table 2. Approximate volume ratios Eq. (11) for the LGSSM posterior conditioned on datasets with different values of T and observation noise levels σ. The bold row corresponds to the volume ratios of the posteriors used in our experiments (see Appendix E.2). Smaller values indicate a more concentrated posterior relative to the prior.

E.1.3 Justifying Increasing Data Dimension

The goal of this experiment is to compare how each SBI method scales with increasing parameter dimensionality, which in the LGSSM is equal to $d_{\boldsymbol{\theta}} = T + 3$ (2 static parameters and $T + 1$ latent states). Since the data dimension is $d_{\boldsymbol{x}} = T + 1$, a reader might be concerned that increasing T simultaneously increases both the parameter and data dimensionality, potentially confounding the results. However, we refer the reader to Section 4, where we argue that the data dimensionalities (up to 101) considered in our experiments are unlikely to introduce noticeable confounding effects.

E.1.4 Evaluation Metrics

For each SBI method, we wish to develop evaluation metrics that measure the quality of the posterior approximation at each time series length T. We consider three evaluation metrics, one for ρ, one for τ, and one for $z_{0:T}$.

Evaluation metrics for ρ and τ: For each value of T, we can compute the mean, 5% and 95% quantiles of the exact posterior over ρ using the Kalman filter. These are compared to the corresponding mean and quantiles obtained by each SBI posterior approximation to evaluate how well each SBI algorithm recovers the posterior over ρ. An analogous evaluation is performed for τ.

Evaluation metric for $z_{0:T}$: Having developed evaluation metrics for ρ and τ, we now wish to derive a metric that evaluates how well the SBI approximations recover the posterior over the latent trajectory $z_{0:T}$. Recall from Eq. (9) that, conditional on a fixed (ρ, τ), the posterior over latent states is Gaussian with mean $\mu^{\text{posterior}}_{\rho,\tau}$ and covariance matrix $\Sigma^{\text{posterior}}_{\rho,\tau}$ (which can both be computed in closed form). We can standardize a sample $z_{0:T}$ from this conditional posterior $\pi(z_{0:T}|\rho, \tau, x^{\text{obs}}_{0:T})$ as follows:

$$\left(\Sigma^{\text{posterior}}_{\rho,\tau}\right)^{-1/2} \left(z_{0:T} - \mu^{\text{posterior}}_{\rho,\tau}\right) \sim N(\mathbf{0}, I_{T+1}) \tag{12}$$

By Lemma 3, the squared norm of this standardized sample follows a χ^2_{T+1} distribution:

$$\left\| \left(\Sigma^{\text{posterior}}_{\rho,\tau}\right)^{-1/2} \left(z_{0:T} - \mu^{\text{posterior}}_{\rho,\tau}\right) \right\|_2^2 \sim \chi^2_{T+1} \tag{13}$$

We use the statistic in Eq. (13) to evaluate the accuracy of latent-state recovery for each SBI method: given a posterior approximation $\tilde{\pi}(\rho, \tau, z_{0:T}|x^{\text{obs}}_{0:T})$, we draw evaluation samples and compute this statistic for each sample. We then compare the mean, 5%, and 95% quantiles of these statistics to those of the reference χ^2_{T+1} distribution. Agreement with the reference distribution indicates accurate recovery of the latent-state posterior.

Due to its similarity to the squared norm evaluation metric introduced in Section 4, the interpretation of Eq. (13) is largely analogous. In particular, Eq. (13) is the squared norm of Eq. (12), where the latter is the latent state trajectory that has been standardized using the exact posterior mean $\mu^{\text{posterior}}_{\rho,\tau}$ and covariance $\Sigma^{\text{posterior}}_{\rho,\tau}$. Therefore, similarly to the squared norm metric used in Section 4 for the uniform Gaussian model, Eq. (13) can be used to detect posterior approximations that exhibit under- or over-dispersion in the latent-state dimensions.

E.2 Experimental Results

In this section, we consider the linear Gaussian state space model. Recall that the variable T controls the parameter dimensionality, since the model has 2 static (global) parameters and $T+1$ latent states (local parameters), resulting in a total parameter dimensionality of $T+3$. We vary T from 10 to 100, corresponding to posterior-prior volume ratios between 7.92×10^{-4} and 3.05×10^{-33} (see Table 2). To investigate how accurately each SBI method recovers the posterior distribution as T increases, we compare methods using the evaluation metrics outlined in Section E.1.4.

We note that the MCMC sampling required by NLE-MCMC and SNRE-B was prohibitively slow, necessitating their exclusion from the analysis; we discuss this further in Section E.2.2.

Plots we discuss in this section: Fig. 8 and Fig. 9 show, for each value of T, the marginal posterior approximations for the global parameters ρ and τ, respectively. Fig. 10 shows the distribution of the local parameter evaluation statistic Eq. (13) as a function of T for each method. To provide further insight into the behaviour of the sequential methods in high dimensions, Fig. 11 shows how the distribution of this statistic evolves across the eight rounds of sequential training for $T = 50$. Analogous plots for $T = 10, 25$, and 100 are depicted in Fig. 13. We compare all evaluation metrics to their ground-truth values (obtained using the Kalman filter). We also plot the average runtime (sampling + training time) of each algorithm as a function of T in Fig. 12. The following subsections analyze these plots in detail and summarize our main experimental findings.

E.2.1 High-Level Observations on Posterior Approximation Quality

Under a matched simulation budget, sequential methods often recover global parameters less accurately than amortized methods for $T > 10$: When compared against amortized methods trained on the same budget of 40,000 total simulations, the sequential variants of SNPE-C and TSNPE often recover the global parameters ρ and τ less accurately for $T > 10$. For TSNPE, Fig. 8 shows that, for $T > 10$, the single-round posterior approximations over ρ remain reasonably accurate, whereas the multi-round approximations become inflated (often extending close to the bounds of the Unif$[0, 1]$ prior) and are significantly more biased. Similarly, Fig. 9 shows that the multi-round TSNPE posterior approximations over τ are more biased than the amortized approximations. For SNPE-C, the multi-round posterior approximation for ρ at $T = 100$ is noticeably more biased than its single-round counterpart (see Fig. 8). For smaller values of T, however, the amortized and sequential SNPE-C posterior approximations over ρ appear comparable. Moreover, Fig. 9 shows that the multi-round SNPE-C posterior approximations for τ are noticeably less accurate than their amortized counterparts, particularly for $T = 50$, where the sequential posterior becomes substantially more biased. The amortized and sequential variants of NLE-VI exhibit similarly poor recovery of the global parameters across all values of T. Notably, the inflated error bars in the bottom left panels of Fig. 8 and Fig. 9, relative to the top left panels, indicate that sequential NLE-VI approximations exhibit greater variability than those from amortized NLE-VI across experimental runs.

Under a matched simulation budget, sequential methods often recover local parameters less accurately than amortized methods for $T > 10$: Similarly to the global parameters, Fig. 10 shows that, when compared against amortized methods trained on the same budget of 40,000 total simulations, the multi-round variants of SNPE-C and TSNPE often recover the local parameters (latent states) $z_{0:T}$ less accurately for $T > 10$. This is evidenced by the bottom right and bottom middle panels in Fig. 10, which exhibit poorer agreement with the gray shaded region than the corresponding top right and top middle panels for $T > 10$. For TSNPE, the amortized posterior approximations remain highly accurate for $T < 100$. In contrast, the corresponding multi-round approximations are less reliable: at $T = 10$, multi-round TSNPE exhibits an instability that is discussed further in Section E.2.2, while for $T = 25$ and $T = 50$ the posterior becomes under-dispersed. At $T = 100$, the amortized posterior is over-dispersed whereas the multi-round posterior is under-dispersed, suggesting inaccuracies in both approaches. For SNPE-C, the degradation in performance between the amortized and multi-round methods is more pronounced. In particular, for all T above 10, the sequential SNPE-C posterior approximation becomes less accurate than the amortized version, often becoming increasingly over-dispersed in comparison. We observe an instability in multi-round SNPE-C at $T = 25$ which we discuss further in Section E.2.2. The amortized NLE-VI approximations are poor for all values of T, and the sequential variants fail to yield an improvement, often inducing instabilities (see Section E.2.2).

Additional rounds of sequential training often degrade approximation quality when $T > 10$: We have already established that, when trained on the same total number of simulations, sequential variants can underperform amortized ones. We now examine a related but distinct question: how posterior approximation quality evolves over successive rounds of sequential training. This analysis considers only the per-round posterior approximations from the sequential methods and is therefore distinct from the matched-budget comparison presented above. For $T = 50$, Fig. 11 shows that additional rounds of training frequently degrade approximation quality of TSNPE and SNPE-C as measured by the local parameter evaluation statistic Eq. (13). We address each method separately in Section E.2.2. We note that, for NLE-VI, the plot shows that the posterior approximations exhibit similarly poor performance across all rounds.

Our findings coincide with previous low-dimensional studies: For $T = 10$, corresponding to a parameter dimensionality of $T + 3 = 13$, we observe that sequential methods tend to perform at least as well as amortized methods. The only exception is the TSNPE experiment, in which inflated error bars at $T = 10$ in Fig. 10 indicate an instability in the sequential training. In Section E.2.2, we show that this instability is driven entirely by a single divergent TSNPE run among the 10 used in the plot. When this run is excluded, the $T = 10$ results for sequential TSNPE align closely with the gray shaded region, indicating that the method otherwise produces accurate posterior approximations at this dimensionality. With the exception of this divergent TSNPE run, the results of this experiment are consistent with the findings of the low-dimensional studies reviewed in Section 3, mirroring the conclusions drawn for the uniform Gaussian model in Section 5.

E.2.2 In-Depth Analysis of Individual Methods

Sequential TSNPE exhibits the same progressive under-dispersion observed in the uniform Gaussian model: The middle panel of Fig. 11 and the second panel of Fig. 2 depict the per-round performance of sequential TSNPE applied to the LGSSM and uniform Gaussian models, respectively. Namely, the former corresponds to the LGSSM with $T = 50$, while the latter corresponds to the uniform Gaussian model with $d = 50$. Despite the different models, the two panels exhibit strikingly similar failure modes. This comparison is particularly meaningful because the evaluation metrics used in the two settings are closely related: both are squared norms of vectors derived from joint posterior samples (see Section 4 and Section E.1.4). In particular, we observe that the TSNPE posterior approximation becomes increasingly under-dispersed across rounds in both models, suggesting that early stopping at an intermediate round would be beneficial. In Fig. 2, termination at round 1 appears optimal, whereas Fig. 11 suggests that the round 2 approximation is most accurate. In practice, however, it is unlikely that a practitioner would have such insight into when to terminate training.

Sequential SNPE-C shows signs of recovering from over-dispersion, in contrast to the uniform Gaussian model at a comparable dimensionality: The rightmost panel of Fig. 11 and the leftmost panel of Fig. 2 depict the per-round performance of sequential SNPE-C applied to the LGSSM and uniform Gaussian models, respectively. Namely, the former corresponds to the LGSSM with $T = 50$, while the latter corresponds to the uniform Gaussian model with $d = 50$. In both plots, we observe a substantial increase in dispersion from the first to the second round, with the second-round posterior approximation becoming significantly over-dispersed in each case. This is followed either by a gradual correction (LGSSM, Fig. 11) or by further amplification (uniform Gaussian model, Fig. 2) of this over-dispersion across the remaining rounds. In both figures, the first-round posterior approximations appear to be the most accurate. However, extrapolating the trend observed in the LGSSM experiment (rightmost panel of Fig. 11) suggests that additional rounds of training beyond the eight considered here may eventually correct the over-dispersion sufficiently for the final-round posterior to become optimal.

The finding above was initially surprising, since at comparable parameter dimensionalities (50 and 53, respectively), one would expect SNPE-C to be less prone to failure on the simpler uniform

Gaussian model; yet it instead exhibits no signs of recovery from over-dispersion, whereas it does for the more complex LGSSM. Examining the posterior–prior volume ratios of the two models reveals that the uniform Gaussian model has a much smaller ratio of 10^{-28}, compared to 1.76×10^{-16} for the LGSSM (see Fig. 4 and Table 2, respectively). Therefore, despite its simpler posterior geometry, the $d = 50$ uniform Gaussian model may pose a greater challenge to sequential SNPE-C than the more complex $T = 50$ LGSSM, due to its posterior being substantially more concentrated (see Section 4 for further discussion).

There appears to be a threshold dimension beyond which sequential SNPE-C deteriorates monotonically across rounds: The leftmost column of Fig. 6 shows how the squared norm evaluation metric evolves across training rounds for sequential SNPE-C in the uniform Gaussian model, for each value of d. The rightmost column of Fig. 13 shows the analogous plots for the LGSSM, displaying the evolution of the latent-state evaluation metric Eq. (13) across SNPE-C training rounds for each value of T. Despite representing two different models, these plots reveal a common pattern: there appears to be a threshold dimensionality separating two distinct regimes. Below this threshold, the over-dispersion introduced by sequential SNPE-C tends to be gradually corrected across training rounds. Above it, however, the posterior approximations become increasingly over-dispersed from round to round, with no evidence of recovery. For the uniform Gaussian model, this threshold appears to lie between $d = 25$ and $d = 50$, corresponding to posterior-prior volume ratios between 2.85×10^{-11} and 1.54×10^{-28} (see Fig. 4). For the LGSSM, the threshold appears to lie above $T = 100$, corresponding to a volume ratio above 3.05×10^{-33} (see Table 2), since the $T = 100$ plot shows signs of entering the regime of amplified over-dispersion. We note that these threshold dimensionalities lie strictly above those considered in existing work, suggesting that this failure mode of sequential SNPE-C may only become apparent in higher-dimensional settings than those explored previously.

All sequential methods exhibit instabilities across runs: Fig. 10 shows that the three sequential methods considered in this experiment each exhibit instability for at least one value of T. This is evidenced by the inflated error bars: for TSNPE, this occurs at $T = 10$; for SNPE-C, at $T = 25$; and for NLE-VI, at both $T = 10$ and $T = 25$. A closer inspection of the 10 experimental runs reveals that the instabilities observed in TSNPE and SNPE-C are driven by a single anomalous run in each case, which produced a statistic more than an order of magnitude larger than that expected under the true distribution, while the remaining nine runs appear well-behaved. For NLE-VI, however, the observed instabilities in the sequential variant are driven by multiple divergent runs, a finding that is consistent with the NLE divergences reported in Section 5. Notably, these divergences occur in lower dimensions than in the uniform Gaussian model, suggesting that the more complex posterior geometry in the LGSSM exacerbates such failure modes.

E.2.3 Computational Cost

Methods with an MCMC subroutine were prohibitively slow: NLE-MCMC and SNRE-B (which also relies on an MCMC subroutine) were prohibitively slow to run on our computing cluster, likely due to the high dimensionality and complex posterior geometries of the MCMC targets. This remained the case even after experimenting with both slice sampling and NUTS as MCMC subroutines and using up to 16 CPU cores for parallel sampling. We therefore exclude these methods from the analysis. This exclusion reflects a practical limitation of these methods: if inference is infeasible on the computational resources available for this study, it is likely to be similarly infeasible for practitioners working with similar computational resources.

Under a matched simulation budget, sequential methods incur higher computational costs than amortized methods: Fig. 12 shows that sequential methods require substantially longer runtimes than their amortized counterparts for all values of T, despite both being trained on 40,000 simulations. In particular, sequential TSNPE consistently exhibits the highest average runtime, reaching over 8 hours at $T = 100$, compared to approximately 1 hour and 18 minutes

for the amortized version. In contrast, sequential SNPE-C consistently has the lowest runtime among the sequential methods, taking about 1 hour at $T = 100$, with its amortized counterpart being even faster at approximately 25 minutes. This shows that multi-round training introduces a substantial computational overhead beyond what would be expected from using the same total number of simulations in a single training round.

Appendix F. Implementation Details

All experiments were conducted in Python 3.9.6, using the `sbi` package (Boelts et al. (2025)) to implement the SBI algorithms. The `sbi` toolbox provides an interface for the main SBI algorithms (NPE, NLE, NRE) along with built-in support for sequential and amortized training using PyTorch. Its modular design allows researchers and practitioners to experiment with different density estimators, neural architectures, embedding networks, and training schemes in a reproducible manner.

To ensure that our experiments reflect typical usage by practitioners, we use the recommended defaults from the `sbi` package unless stated otherwise. We stress that practitioner-focused validation studies like ours should refrain from selecting hyperparameters based on performance on benchmarks with known ground truth. Not only does this provide an unrealistic reflection of typical usage, but it also risks overfitting to specific benchmark tasks, which would limit the generalizability of any conclusions drawn about the algorithm. Thus, we adopt settings that are intended to generalize across all tasks we consider.

In practical applications of SBI algorithms, hyperparameters are typically chosen by monitoring metrics such as training and validation losses, with adjustments made to ensure stable training. When there were architectural choices to be made, such as selecting embedding networks for different types of data, we followed standard machine learning guidelines. For example, when working with high-dimensional raw images, it would be natural to use a CNN embedding network to process the data.

We leverage insights from other validation studies (Lueckmann et al. (2021); Hermans et al. (2022); Deistler et al. (2025)) where possible. For example, Lueckmann et al. (2021) found that using vectorized axis-aligned slice sampling with multiple chains as the MCMC method in SNLE-MCMC led to strong results across tasks (see their Appendix H.3).

We note that TSNPE failed to produce a single sample within a tractable time when using rejection sampling to truncate the prior when $d \geq 10$. This is because the round 1 posterior approximation concentrates significantly relative to the prior, meaning the truncation set occupies only a tiny fraction of the prior's volume, resulting in an extremely low acceptance rate for rejection sampling. To avoid this, we follow the approach of the original authors (Deistler et al. (2022)) and employ sampling importance resampling (SIR) to draw from the truncated prior for all dimensions d.

Code for all experiments can be found at:

https://github.com/kaisamaroo/SBC-SBI

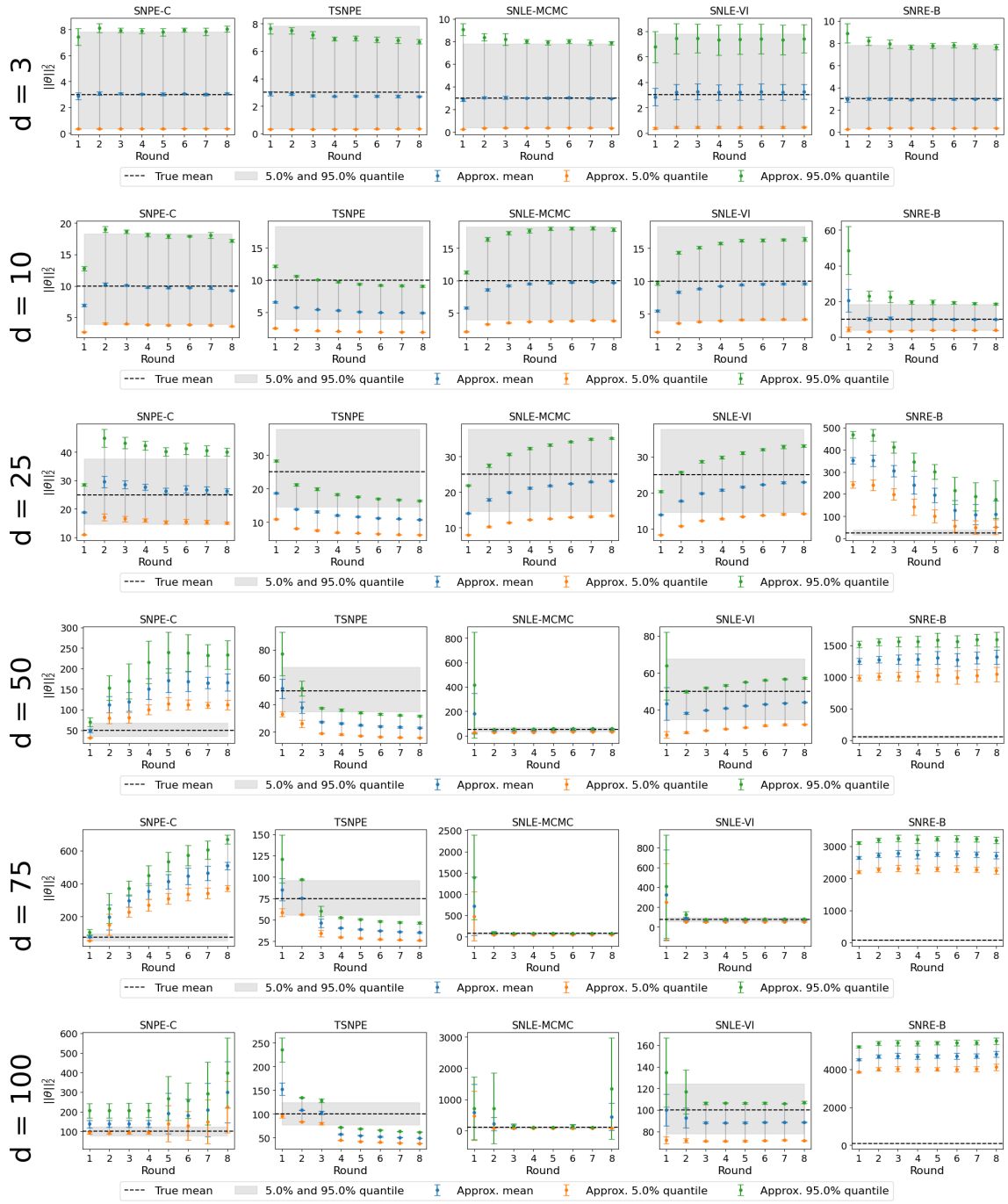

Figure 6. The mean, 0.05 quantile, and 0.95 quantile of the empirical distribution of the squared norms of samples from the approximate posterior at each round of training, for $d \in \{3, 10, 25, 50, 75, 100\}$. The true mean and quantiles are the black dashed line and shaded region, respectively. Each mean and quantile is accompanied by an error bar representing $\pm$ two standard errors from 10 repeated experiments. The closer the vertical lines lie to the shaded gray region, the more closely the approximate posterior matches the true posterior. Vertical lines above/below the shaded region suggest over-/under-dispersion, respectively.

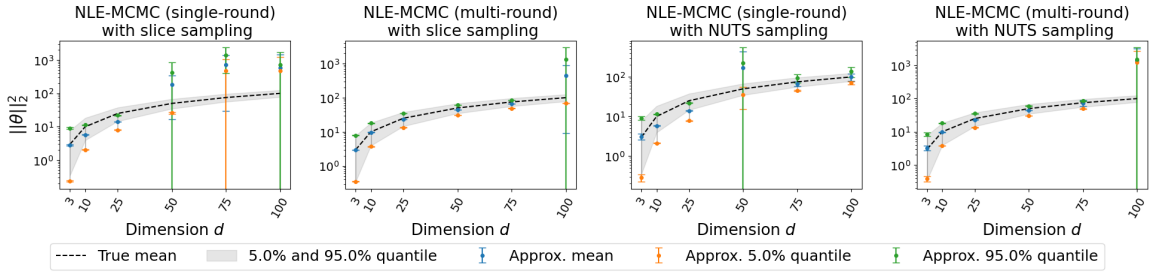

Figure 7. Comparison of amortized and sequential NLE using different MCMC subroutines. The two left panels use the axis-aligned slice sampling subroutine, while the two right panels use the NUTS subroutine. All panels use the same visual indicators and columns as in Fig. 1.

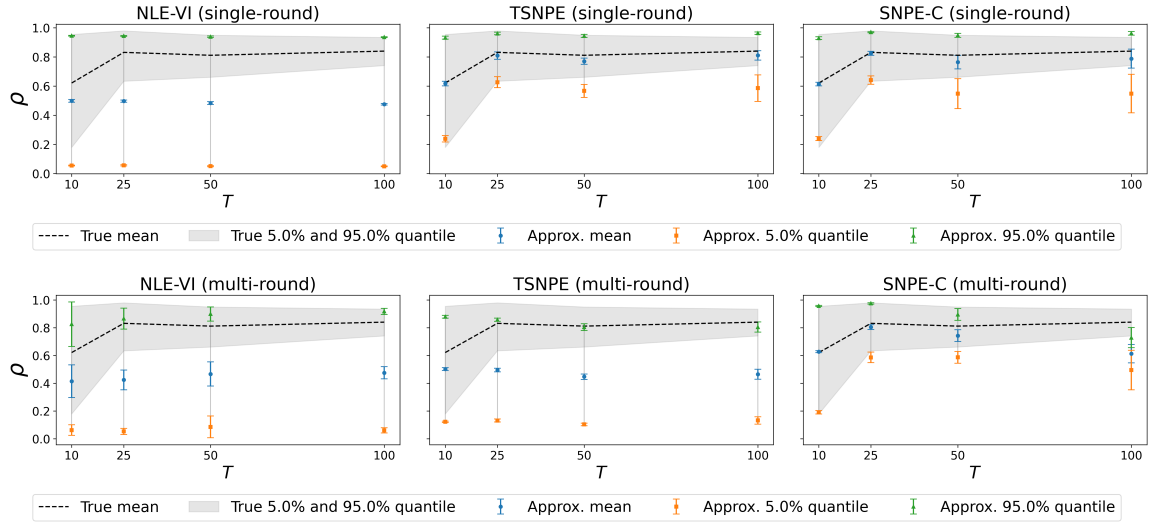

Figure 8. *Upper row*: Amortized (single-round) performance when trained on the full 40,000 simulation budget. *Lower row*: Sequential (multi-round) performance. Each *column* corresponds to a different SBI method. In each *panel*, we plot the exact posterior mean of ρ (dashed black line) and exact 5% to 95% quantile region (gray shading) against T (horizontal axis: 10, 25, 50, 100). We plot the SBI method's mean approximation (blue dot), 5% quantile (orange dot), and 95% quantile (green dot) for ρ. Each mean and quantile is shown as an average over 10 repeated experiments and is accompanied by an error bar representing $\pm$ two standard errors. Refer to Table 2 to convert values of T into posterior-prior volume ratios.

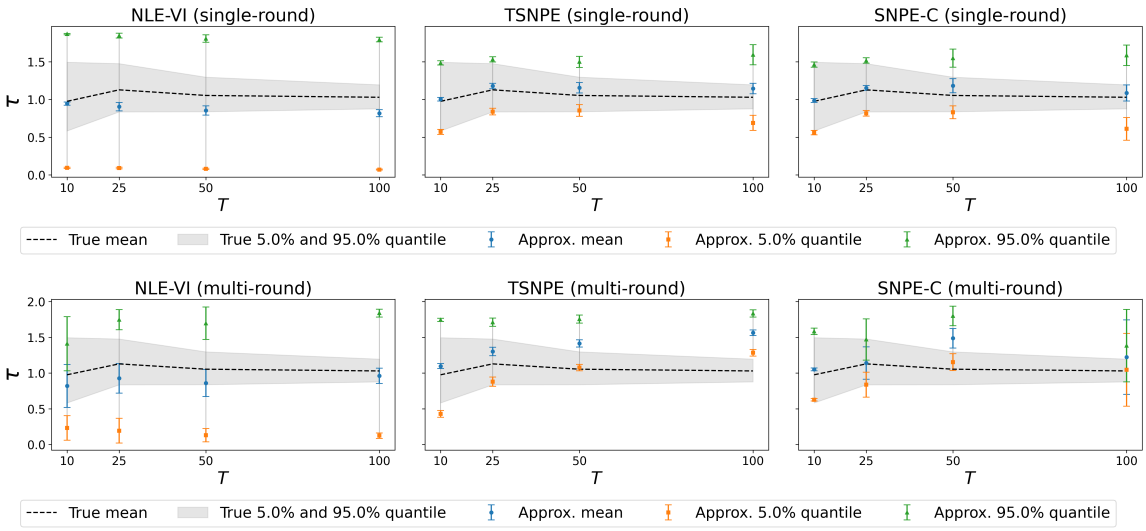

Figure 9. We use the same visual indicators and column layout as in Fig. 8, but now display the posterior distributions over τ.

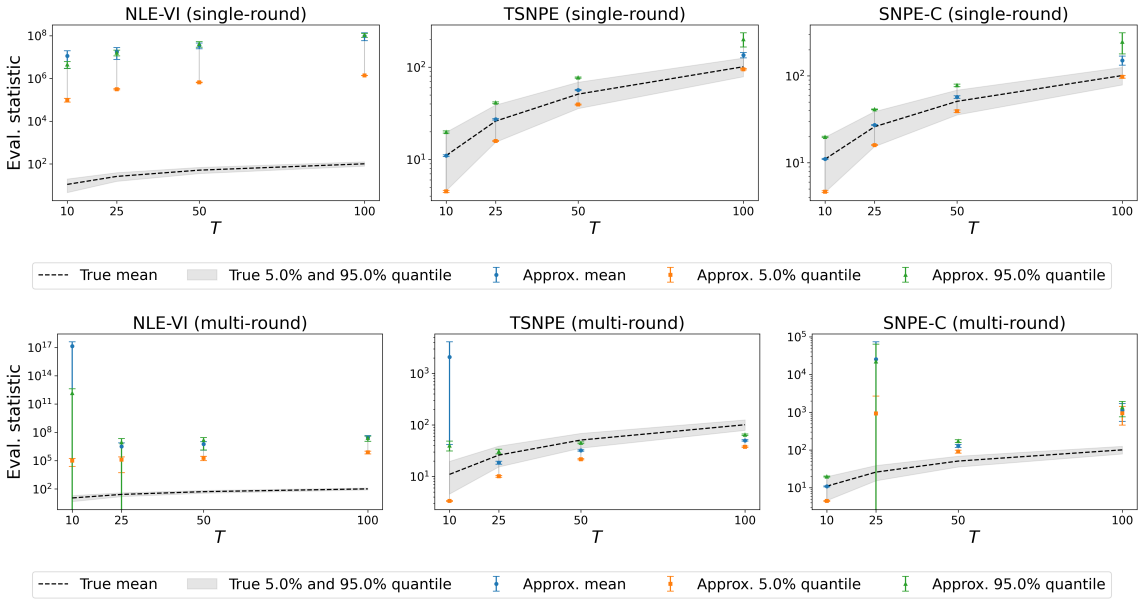

Figure 10. *Upper row*: Amortized (single-round) performance when trained on the full 40,000 simulation budget. *Lower row*: Sequential (multi-round) performance. Each *column* corresponds to a different SBI method (vertical axes not shared). In each *panel*, we plot the exact posterior mean of the local parameter evaluation statistic Eq. (13) (dashed black line) and exact 5% to 95% quantile region (gray shading) against T (horizontal axis: 10, 25, 50, 100). We plot the SBI method's mean approximation (blue dot), 5% quantile (orange dot), and 95% quantile (green dot). Each mean and quantile is shown as an average over 10 repeated experiments and is accompanied by an error bar representing $\pm$ two standard errors. The closer the vertical lines lie to the shaded gray region, the more closely the approximate posterior matches the exact posterior. Vertical lines above/below the shaded region suggest over-/under-dispersion, respectively. Refer to Table 2 to convert values of T into posterior-prior volume ratios.

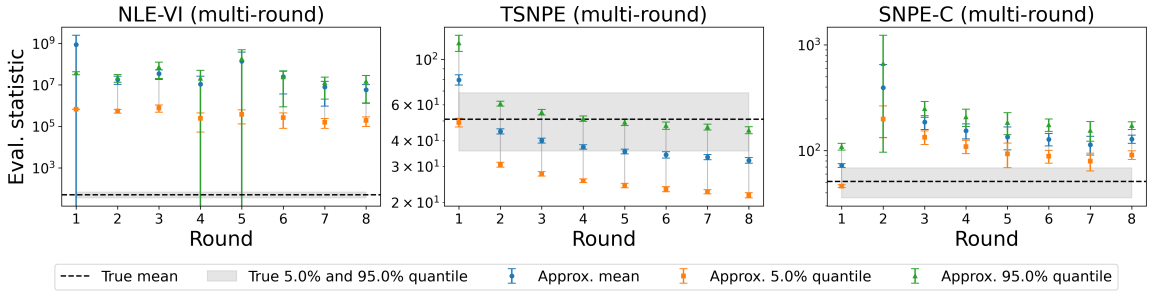

Figure 11. We use the same visual indicators and columns as in Fig. 10, but now each panel shows the performance of the sequential SBI method across rounds (horizontal axis) for a fixed $T = 50$, corresponding to a posterior-prior volume ratio of 1.76×10^{-16}. The first-round posterior approximations can be viewed as amortized posterior approximations trained using 5,000 simulations.

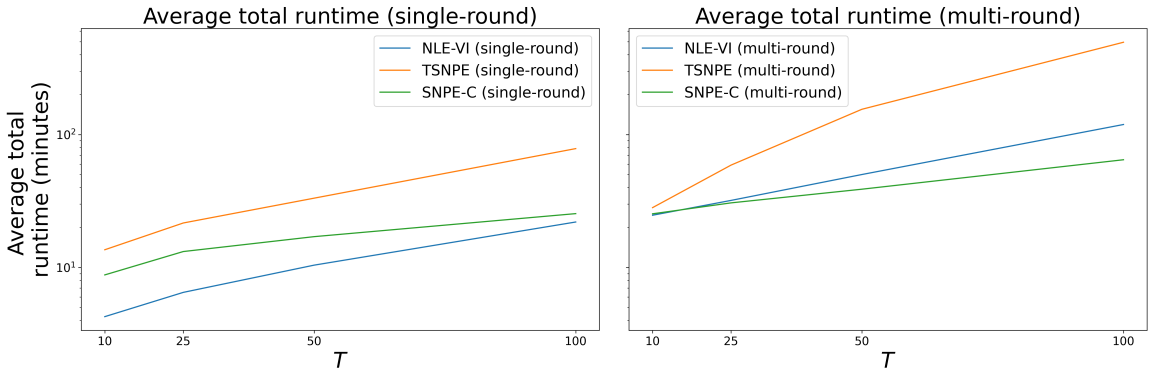

Figure 12. Total runtime of each algorithm (in minutes) as a function of T, averaged across 10 repeated experiments. The left panel shows single-round variants (trained on the full 40,000 simulation budget), and the right panel shows 8-round variants. The vertical axis is shown on a logarithmic scale.

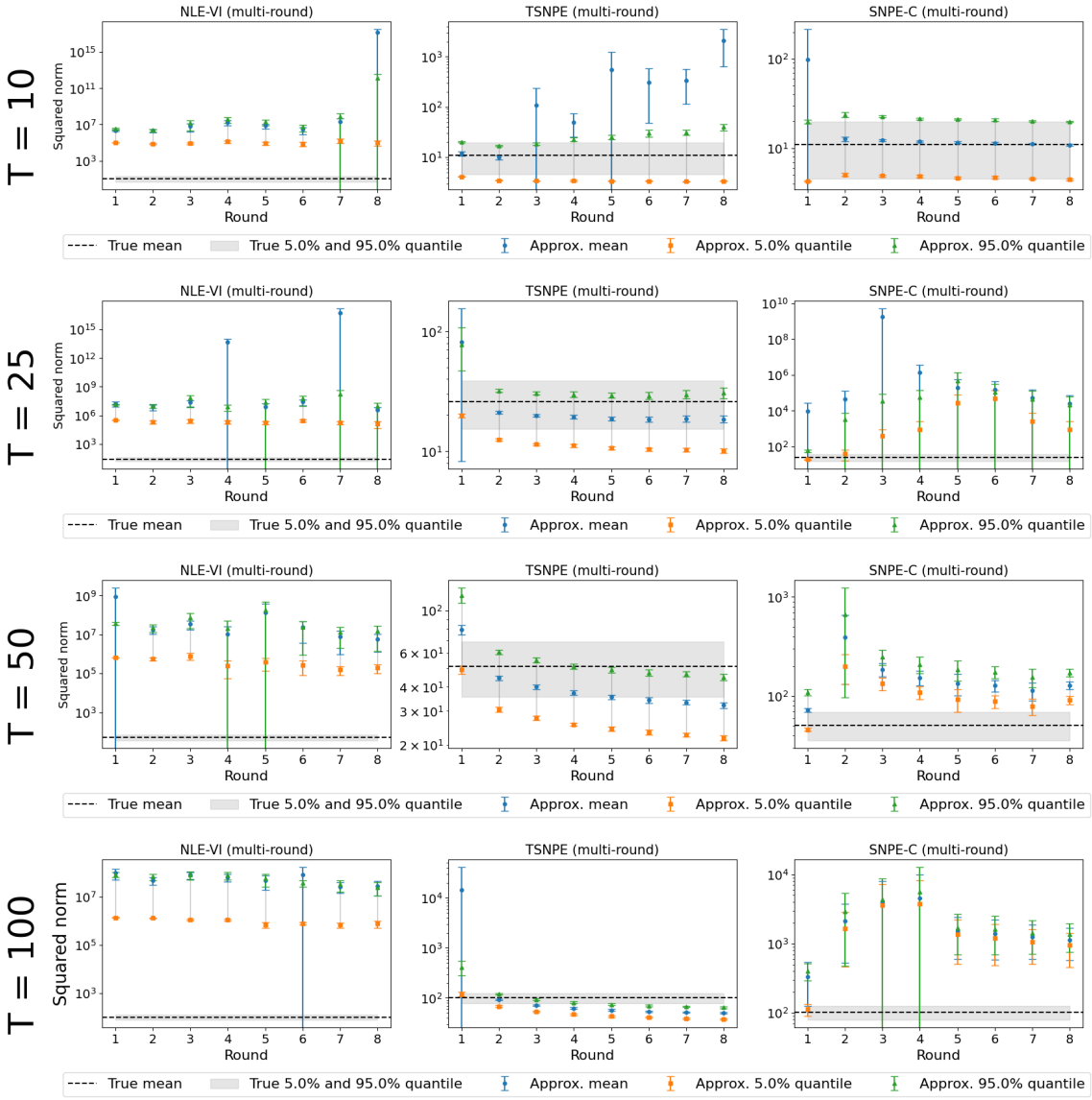

Figure 13. The mean, 5% quantile, and 95% quantile of the empirical distribution of the statistic in Eq. (13) for samples drawn from the approximate posterior at each round of training, for $T \in \{10, 25, 50, 100\}$. The true mean and quantiles (of a χ^2_{T+1} distribution) are the black dashed line and shaded region, respectively. Each mean and quantile is shown as an average over 10 repeated experiments and is accompanied by an error bar representing $\pm$ two standard errors. The closer the vertical lines lie to the shaded gray region, the more closely the approximate posterior matches the exact posterior. Vertical lines above/below the shaded region suggest over-/under-dispersion, respectively.