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

Anonymous Authors

Abstract

Simulation-based inference (SBI) is widely used to infer parameters of simulators whose likelihood functions are intractable. Recent work has suggested that sequential, multi-round training can improve upon a single round. While these conclusions are supported by evaluations on benchmark tasks with at most 20 parameters, simulators of practical interest can exhibit parameter dimensionalities exceeding 100. In the present work, we systematically study the performance of both single-round and sequential SBI methods as parameter dimensionality increases to 100. We show empirically that sequential methods often perform worse than single-round methods in dimensions higher than 20.

1 Introduction

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), and computer graphics (Mansinghka et al., 2013). However, this increased complexity often renders the simulator's likelihood function intractable. Simulation-based inference (SBI, Cranmer et al., 2020) provides a framework for performing parameter inference in this setting, with modern neural SBI methods now widely used by practitioners to infer parameters of real-world simulators (Ramirez Sierra and Sokolowski, 2025; Hashemi et al., 2023; Hull et al., 2024). These methods have both *amortized* and *sequential* implementations. Sequential methods employ a multi-round training procedure in which the approximation obtained at each round is used to guide future rounds. The sequential rounds are designed to concentrate training samples in regions of the parameter space that generate data resembling the observed data. Amortized methods can be thought of as sequential methods with a single round. While amortized methods are cheaper (and often more stable) to train, the literature currently suggests employing sequential methods when high accuracy is required for a specific observation (Lueckmann et al., 2021; Deistler et al., 2025; Papamakarios and Murray, 2016).

However, almost every assessment of SBI approximation quality that we are aware of uses parameter dimensions less than 20 (this bound holds even beyond comparisons of sequential and amortized methods; see Section 2 for details). Thus, it is plausible a priori that sequential methods behave differently in high-dimensional settings in ways not captured by the existing literature. The only systematic exploration of performance in high-dimensional parameter spaces (to our knowledge) is Hashemi et al. (2023), who run SBI on an epilepsy model with hundreds of parameters. They compare classification accuracy across parameter dimension (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. In this work, we systematically compare the accuracy of leading sequential and amortized SBI methods across

dimensions up to 100 in a controlled, stylized model. Rather surprisingly, we find that sequential methods can sometimes *degrade* performance relative to their amortized counterparts.

2 Related Work

SBI Methods We Consider. We compare five SBI methods, which we choose as follows. The popular `sbi` Python package (Boelts et al., 2025) includes defaults for three popular SBI variants that all take different approaches to forming posterior approximations (sbi developers, 2024a): neural posterior estimation (NPE, Papamakarios and Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Deistler et al., 2022), neural likelihood estimation (NLE, Papamakarios et al., 2019; Glockler et al., 2022), and neural ratio estimation (NRE, Hermans et al., 2020; Durkan et al., 2020; Miller et al., 2022). Each have sequential variants which we refer to as **S**NPE, **S**NLE, and **S**NRE, respectively. In our experiments, we follow the default recommendations of the `sbi` package to ensure that state-of-the-art variants are used in each of the three cases: For SNPE, the default is SNPE-C (Greenberg et al., 2019), but sbi developers (2024b) recommend TSNPE (Deistler et al., 2022) if SNPE-C fails, so we consider both. MCMC and VI are subroutines of SNLE, so we include both (SNLE-MCMC, Papamakarios et al. (2019) and SNLE-VI, Glockler et al. (2022)) in our experiments. For SNRE, SNRE-B (Durkan et al., 2020) with an MCMC subroutine is the default.

Data Dimensionality. It is well established in the literature that SBI methods degrade when applied to simulators with high-dimensional *data* (Greenberg et al., 2019; Dirmeier et al., 2025; Papamakarios et al., 2019; Hermans et al., 2022; Lueckmann et al., 2021), largely due to the difficulty of extracting informative features from high-dimensional inputs. However, it is less well understood how SBI methods scale to high-dimensional *parameters*.

Parameter Dimensionality. The original papers proposing SBI methods include empirical evaluations on a range of tasks and simulators but do not examine the impact of high parameter dimensionality on accuracy, with many focusing instead on the challenges posed by high-dimensional data. For example, Greenberg et al. (2019); Papamakarios et al. (2019); Glockler et al. (2022); Dirmeier et al. (2025); Durkan et al. (2020); Miller et al. (2022) evaluate their methods on simulators with (often considerably) less than 20 parameters. Deistler et al. (2022) run their algorithm on two neuroscience simulators with parameter dimensionalities up to 31. However, these experiments were used to compare running time relative to SNPE-C; they did not assess posterior approximation quality.

Recent evaluation studies (Hermans et al., 2022; Lueckmann et al., 2021) propose a diverse range of benchmark simulators and evaluate both amortized and sequential methods. However, they only consider at most 10 parameter dimensions. Deistler et al. (2025) evaluate neural posterior estimation on a neuroscience simulator with 31 parameters, but restrict attention to the amortized setting, leaving open how a sequential variant would perform.

Appendix C contains a summary of the maximum parameter dimensionalities used in key SBI papers.

3 Experimental Methods

Setup and Notation. We assume access to a simulator that generates 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 $\pi(\boldsymbol{x}|\boldsymbol{\theta})$. We assume that samples $\boldsymbol{x}$ can be obtained by forward simulation, but the likelihood function $\pi(\boldsymbol{x}|\boldsymbol{\theta})$ is intractable and cannot be evaluated pointwise. The simulator, combined with a user-specified proper prior $\pi(\boldsymbol{\theta})$ and observed data $\boldsymbol{x}_{\text{obs}}$, defines an exact posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}}) \propto_{\boldsymbol{\theta}} \pi(\boldsymbol{x}_{\text{obs}}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. Practitioners often aim to approximate certain marginal means and variances of the exact posterior, as these characterize the distributions of scientifically relevant parameters. To approximate the posterior, SBI uses a training dataset $\{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N$ of N simulated samples from the joint distribution $\pi(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ (i.e. each $\boldsymbol{\theta}_n \sim \pi(\boldsymbol{\theta})$ followed by $\boldsymbol{x}_n \sim \pi(\boldsymbol{x}|\boldsymbol{\theta}_n)$).

The Model. For our experiment, we choose a model whose posterior can be sampled from directly using simple Monte Carlo; that is, we choose a model with a reference ground truth posterior.

Since all SBI methods are trained on samples from the prior, we suspect they may struggle when the region of high posterior mass occupies only a small fraction of the prior. Intuitively, a prior that is much more diffuse than the posterior results in most training parameters $\boldsymbol{\theta}_n$ lying far from the region of high posterior mass. In this case, only a small fraction of training simulations will be informative about the exact posterior. To study the impact of posterior concentration, we consider a sequence of models in which this fraction decreases as d increases. In particular, we use a uniform-Gaussian model, with likelihood and prior given by:

$$\boldsymbol{\theta} \sim \text{Unif}[-L_d, L_d]^d \quad \text{and} \quad \boldsymbol{x}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}, I_d),$$

where L_d controls the width of the prior, $d = d_{\boldsymbol{\theta}} = d_{\boldsymbol{x}}$ denotes the dimensionality of both the data space and the parameter space, and I_d is the $d \times d$ identity matrix. We condition on observing $\boldsymbol{x}_{\text{obs}} = \mathbf{0}$, so our posterior becomes

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

where $\mathcal{N}_{[-L_d, L_d]^d}$ denotes a truncated Gaussian distribution on the hypercube $[-L_d, L_d]^d$. The dependence of the prior width L_d on the dimension d is described below.

Choice of the Prior Width. We let L_d be such that over 99% of the posterior mass is inside a sphere of radius L_d; see Appendix B for further justification as well as details of our calculations throughout this section. With this choice of L_d, we observe that the volume of this $\geq$99% credible sphere divided by the volume of the prior is:

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

which goes to zero super-exponentially fast as $d \to \infty$ (see Fig. 3). Equivalently, since the prior is uniform, the expected number of training parameters $\boldsymbol{\theta}_n$ from the SBI training dataset $\{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N$ lying inside the $\geq$99% credible sphere is N times the quantity in Eq. (2).

We expect the SBI algorithms to struggle as d gets larger, since they are tasked with locating a needle (the posterior mass) in an exponentially growing haystack (the prior support).

Choice of the Dimension and Data Dimension Confounding. A reader might be concerned that we confound our experiments by presenting high-dimensional data at the same time we present high-dimensional parameters. 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 will take d only up to 100, we expect negligible confounding from the data dimensionality.

Evaluation Metric. Practitioners commonly report means and variances of key posterior marginals. The posterior mean of $\|\boldsymbol{\theta}\|_2^2 := \sum_{i=1}^d \theta_i^2$ is a sum over the marginal second moments of the posterior and is therefore a summary of performance across dimensions. In what follows, we report the empirical distributions of $\|\boldsymbol{\theta}\|_2^2$ for $\boldsymbol{\theta}$ sampled from each algorithm's posterior approximation. We compare these distributions to ground truth, which is shown in Appendix F to be indistinguishable from a χ_d^2 distribution.

Experimental Design. We conducted all our experiments in Python 3.9.6, using the `sbi` package (Boelts et al., 2025) for implementations of all SBI algorithms. We implement each sequential method using 8 rounds of training, with 5000 simulations per round. At each round $r \in \{1, 2, ..., 8\}$ of training, we compute the squared norms of 5000 samples from the round-r posterior approximation $\tilde{\pi}_r(\boldsymbol{\theta}|\mathbf{0})$. 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 G for further implementation details. Note that we are, if anything, overly generous toward the sequential methods in this design. The sequential methods are trained on 40,000 simulations whereas the amortized variants recieve just 5000.

4 Experimental Results

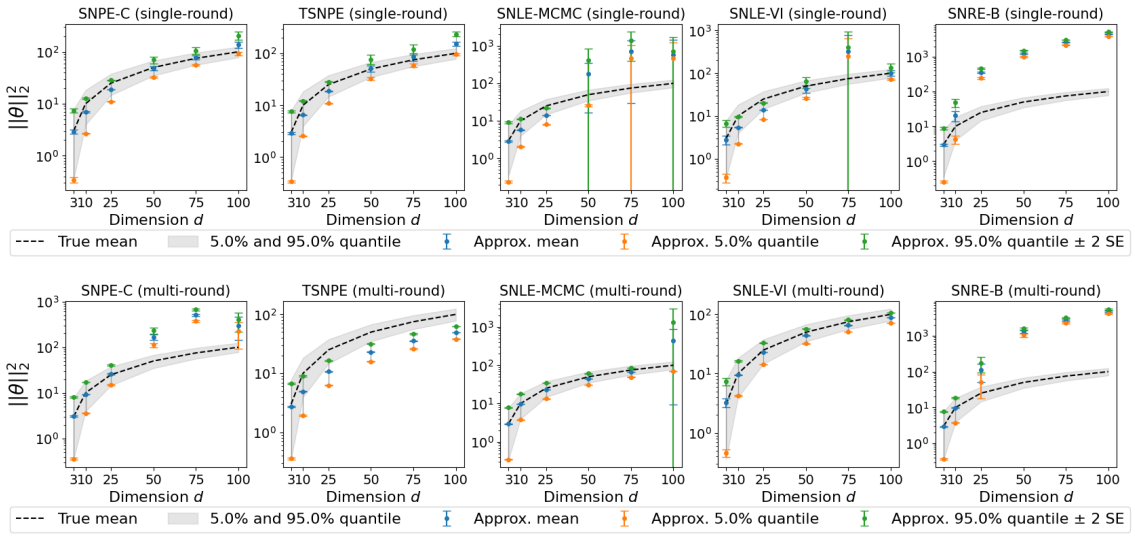

Figure 1. *Upper row*: Amortized (single round) performance. *Lower row*: Sequential (multi-round) performance. Each *column* corresponds to a different SBI method. In each *panel*, we plot the exact posterior mean of $\|\boldsymbol{\theta}\|_2^2$ (dashed black line) and exact 5% to 95% quantile region (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). We run 10 repeated experiments and, for each dot, include bars for ± two standard deviations across experiments.

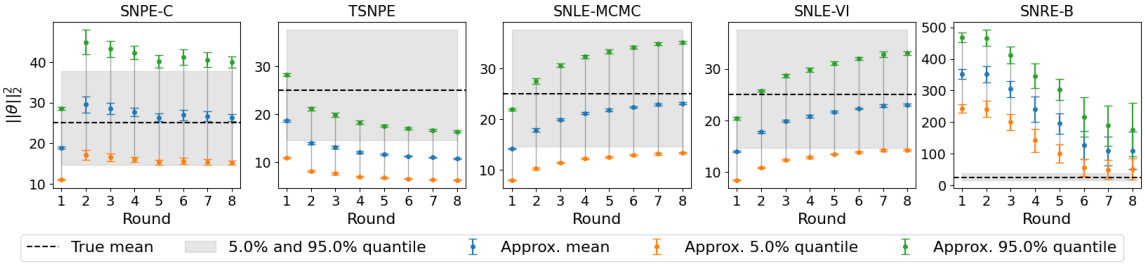

Figure 2. We use the same visual indicators and columns as in Fig. 1, but now each panel shows the performance of the sequential SBI method across rounds (horizontal axis) for a fixed dimensionality $d = 25$.

Amortized methods often beat sequential ones when parameter dimension is above 10. Despite having eight times as many simulations to learn from and more runtime (see Appendix D), we find that the sequential methods often perform worse than the amortized methods when d is above 10. Consider $d > 10$ in each panel of Fig. 1. We see that the approximate posterior mean and quantiles of $\|\boldsymbol{\theta}\|_2^2$ for both SNPE-C and TSNPE (first two columns) are much farther from ground truth in the sequential case than in the amortized case. SNLE-MCMC performs worse in the amortized case. Outside of a divergent VI run at $d = 75$, SNLE-VI performs comparably in the amortized and sequential cases. SNRE-B performs comparably poorly in the two cases.

Our results are consistent with past low-dimensional studies. In low dimensions ($d \leq 10$), Fig. 1 shows that we do see improved performance of sequential methods over amortized methods. This finding coincides with the previous low-dimensional evaluation studies from Section 2. That said, we might expect the amortized methods to perform better with a more comparable simulation budget.

Extra rounds of TSNPE fail since sequential truncation loses posterior mass. Even at $d = 10$, TSNPE begins to produce under-dispersed posteriors. This TSNPE failure mode arises because its first-round posterior approximation $\tilde{\pi}_1(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ is itself under-dispersed: Since TSNPE uses the $1 - \epsilon$ highest probability region of its first-round posterior approximation to truncate the prior, all subsequent proposal distributions assign zero probability mass outside this region. This action has the effect of "locking in" the poor start, and increasingly restrictive proposals amplify the under-dispersion. We see this behavior directly in the TSNPE panel of Fig. 2.

Extra rounds of SNPE-C fail for $d > 25$, and it remains to understand why. We see in the left row of Fig. 1 that SNPE-C performs substantially worse than the amortized equivalent for $d > 25$. Even though the sequential variant does not perform substantially worse at $d = 25$, Fig. 2 suggests a big dip in accuracy for SNPE-C between rounds 1 and 2 and seems to require the remaining rounds up to 8 to recover. Fig. 6 in Appendix E confirms that this behavior only worsens for higher dimensions; by $d = 75$, the approximation is monotonically becoming worse in each round.

SNLE-MCMC and SNLE-VI exhibit instabilities. The amortized and sequential variants of SNLE (using either MCMC or VI subroutines) show comparable performance across dimensions. All exceptions ($d = 100$ for multi-round with MCMC, $d \geq 50$ for amortized with MCMC, $d = 75$ for amortized with VI) seem to result from a small subset of unusual runs (Fig. 1). When we examine the runs more closely, we confirm in each case that at most a handful of the ten repetitions yield extremely large squared norms, over an order of magnitude larger than expected from the true distribution.

One possible cause is the axis-aligned slice sampling procedure used for the MCMC subroutine in SNLE-MCMC. Although we might expect this sampler to perform well for an axis-aligned Gaussian, the MCMC procedure is applied with the approximated likelihood (and exact prior) rather than the true Gaussian likelihood; the approximated likelihood may be sufficiently different from the exact likelihood as to cause these instabilities. If our future work confirms this hypothesis (see Section 5), we might mitigate the observed failures with an MCMC method expected to perform more robustly in high dimensions, e.g. (Hoffman and Gelman, 2014). Notably, though, the similar failure of VI at $d = 75$ opens up the possiblity of a more general SNLE issue.

5 Discussion

In this paper we have shown on a simple toy model that sometimes sequential (multi-round) SBI variants can perform worse than their amortized (single-round) counterparts. A number of directions remain to complete this work in progress. As an immediate set of next steps, we would like to:

- Compare sequential and amortized methods with an equal simulation budget.
- Compare SNRE-B with the VI subroutine; here we used the (default) MCMC subroutine.
- Check if the problematic SNLE runs can be detected with standard MCMC and VI diagnostics.

In the medium to long term, we plan to:

- Perform an analogous comparison on a model where we vary only the parameter dimension while keeping the data dimension fixed.
- Compare to state-of-the-art Markov chain Monte Carlo methods as a best-case baseline.
- Run on more complex and diverse models that still allow a ground truth for comparison.
- Identify and test posterior geometries that might be challenging for the MCMC and VI subroutines.
- Run on real simulators. While we will not have ground truth in these cases, we can use diagnostics such as simulation-based calibration (SBC) to compare.
- Carefully explore the failures of each method to ensure we understand the mechanisms underpinning each failure mode.
- Test MCMC samplers that are known to perform better in high dimensions as alternative MCMC subroutines in SNLE-MCMC (instead of axis-aligned slice sampling).

References

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018. URL https://arxiv.org/abs/1701.02434.

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

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.

R. Hull, E. Leonarduzzi, L. De La Fuente, H. Viet Tran, A. Bennett, P. Melchior, R. M. Maxwell, and L. E. Condon. Simulation-based inference for parameter estimation of complex watershed simulators. *Hydrology and Earth System Sciences*, 28(20):4685–4713, 2024. doi: 10.5194/hess-28-4685-2024. URL https://hess.copernicus.org/articles/28/4685/2024/.

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.

George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with Bayesian conditional density estimation. 29, 2016.

George Papamakarios, David C. Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows, 2019. URL https://arxiv.org/abs/1805.07226.

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.

Michael Alexander Ramirez Sierra and Thomas R Sokolowski. Comparing AI versus optimization workflows for simulation-based inference of spatial-stochastic systems. *Machine Learning: Science and Technology*, 6(1):010502, 2025.

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.

Appendix A. Algorithms

Neural SBI algorithms train powerful neural-network based conditional density[1] estimators on simulated parameter–data pairs in order to approximate the posterior (NPE, Papamakarios and Murray (2016), Lueckmann et al. (2017), Greenberg et al. (2019), Deistler et al. (2022)), the likelihood (NLE, Papamakarios et al. (2019), Glockler et al. (2022)), or likelihood ratios (NRE, Hermans et al. (2020), Durkan et al. (2020), Miller et al. (2022)), enabling inference without requiring likelihood evaluations. Most of these methods have both amortized and sequential implementations. Amortized methods allow rapid inference and can be repeatedly conditioned on multiple different observed datasets cheaply, but tend to be inefficient when high accuracy is required for a specific observation. When the goal is to accurately recover the posterior conditioned on a single observation $\boldsymbol{x}_{\text{obs}}$, sequential variants (which we shall refer to as **S**NPE, **S**NLE, **S**NRE) are often required. These methods employ a multi-round training procedure in which the posterior approximation obtained at round r is used to guide the parameter proposals at round $r+1$. This encourages training samples to accumulate in regions of the parameter space with high posterior mass, improving the quality and relevance of the training data: simulations are concentrated in areas that are most informative for $\boldsymbol{x}_{\text{obs}}$ rather than being wasted on regions of low posterior mass. For a fixed simulation budget, sequential methods have been shown to produce more accurate posterior approximations (Lueckmann et al. (2021)), especially for models with high dimensional data and/or parameter spaces.

SNPE methods sequentially train a conditional density estimator $q_\phi(\boldsymbol{\theta}|\boldsymbol{x})$ to approximate the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$ directly. SNPE-C (Greenberg et al. (2019)) remains the most widely used variant due to its stability and compatibility with flexible density estimators such as normalizing flows (Papamakarios et al. (2021)). However, density estimator leakage issues have been documented in SNPE-C, and TSNPE (Deistler et al. (2022)) aims to mitigate these by proposing parameters using a truncated version of the prior. We consider both in our experiments.

SNLE, in contrast to SNPE, trains a conditional density estimator $q_\phi(\boldsymbol{x}|\boldsymbol{\theta})$ to approximate the intractable likelihood function $\pi(\boldsymbol{x}|\boldsymbol{\theta})$. A conventional likelihood-based Bayesian inference algorithm such as MCMC (Hoffman and Gelman (2014)) or VI (Blei et al. (2017)) is then employed to approximate the posterior with this surrogate likelihood: $\tilde{\pi}(\boldsymbol{\theta}|\boldsymbol{x}) \propto q_\phi(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. We refer to these variants as SNLE-MCMC and SNLE-VI, respectively, and we consider both in our experiments.

SNRE trains a neural network classifier $f_\phi(\boldsymbol{x}, \boldsymbol{\theta})$ using a cross-entropy objective such that it converges to the log-density ratio $\log(\pi(\boldsymbol{x}|\boldsymbol{\theta})/\pi(\boldsymbol{x}))$ up to normalization. Using this learned ratio, SNRE-B (Durkan et al. (2020)) applies MCMC using the unnormalized posterior $\exp\left(f_{\phi^*}(\boldsymbol{x}_{\text{obs}}, \boldsymbol{\theta})\right)\pi(\boldsymbol{\theta})$ to generate samples from the posterior $\pi(\boldsymbol{\theta}|\boldsymbol{x}_{\text{obs}})$. Durkan et al. (2020) showed that SNRE-B and SNPE-C are closely related and can be unified within a contrastive learning framework.

Appendix B. Quantifying the Curse of Dimensionality

It is well known that the squared Euclidean norm of a sample $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, I_d)$ (i.e. from an un-truncated Gaussian with the same mean and covariance as (1)) follows a χ^2_d distribution. Using this fact, we can write

$$P\left(||\boldsymbol{\theta}||_2 \leq R\right) = P\left(||\boldsymbol{\theta}||_2^2 \leq R^2\right) = F_{\chi^2_d}(R^2) \quad \text{for } \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, I_d)$$

where $F_{\chi^2_d}$ denotes the CDF of the χ^2_d distribution. Thus, if we want a spherical region $\mathcal{S}_{R_{0.99}} := \{\boldsymbol{\theta} : ||\boldsymbol{\theta}||_2 \leq R_{0.99}\}$ that contains 99% of the mass of the $\mathcal{N}(\mathbf{0}, I_d)$ distribution, we can solve

$$F_{\chi^2_d}(R^2_{0.99}) = 0.99 \implies R_{0.99} = \sqrt{F^{-1}_{\chi^2_d}(0.99)} \tag{3}$$

1. Or density ratio estimators in the case of NRE (Hermans et al. (2020), Durkan et al. (2020), Miller et al. (2022)).

Taking the prior hypercube width L to equal the radius $R_{0.99}$ ensures that $\mathcal{S}_{R_{0.99}}$ is contained within $[-L, L]^d$, since:

$$\underbrace{||\boldsymbol{\theta}||_2 \leq R_{0.99} = L}_{\boldsymbol{\theta} \text{ in hypersphere } \mathcal{S}_L} \implies \sum_{i=1}^{d} \theta_i^2 \leq L^2 \implies \forall i, |\theta_i| \leq L \implies \underbrace{\boldsymbol{\theta} \in [-L, L]^d}_{\boldsymbol{\theta} \text{ in hypercube } [-L,L]^d}$$

Therefore, taking $L = R_{0.99}$ ensures that at least 99% of the probability mass of a $\mathcal{N}(\mathbf{0}, I_d)$ is contained in the hypercube $[-L, L]^d$. Moreover, since the exact posterior is a truncation of a $\mathcal{N}(\mathbf{0}, I_d)$ to $[-L, L]^d$, it is guaranteed that more than 99% of the mass of the exact posterior (1) is contained in $\mathcal{S}_L$ (for large d, it can be shown that this approaches 100% - see Figure 4). In other words, $\mathcal{S}_L$ is a 99% credible interval for the exact posterior $\pi(\boldsymbol{\theta}|\mathbf{0})$. This setup provides a controlled setting for us to analyze how accurately the SBI methods locate the region of high posterior mass when it occupies only a small fraction of the prior support. In particular, we can quantify how concentrated the posterior is relative to the prior by computing the probability of a prior sample $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$ landing in the 99% posterior credible interval $\mathcal{S}_L$. This probability provides a geometric measure of how challenging the simulation-based inference problem becomes as the dimensionality d increases (see Sections 1.3–1.4 of Betancourt (2018)). Since the prior is uniform, this probability is equal to the volume of $\mathcal{S}_L$ divided by the total volume of the prior support $[-L, L]^d$, and is given by[2]:

$$\begin{aligned} \mathbb{P}_{\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})} \left(\boldsymbol{\theta} \in \mathcal{S}_L \right) &= \frac{\text{vol}\left(\{ \boldsymbol{\theta} : ||\boldsymbol{\theta}||_2 \leq L \} \right)}{\text{vol}([-L, L]^d)} = \frac{\pi^{d/2} L^d / \Gamma(1 + d/2)}{(2L)^d} \\ &= \left(\frac{\sqrt{\pi}}{2} \right)^d \frac{1}{\Gamma(1 + d/2)} \end{aligned} \tag{4}$$

which goes to zero super-exponentially fast as $d \to \infty$ (see Figure 3). Equivalently, for an SBI training dataset $\{(\boldsymbol{x}_n, \boldsymbol{\theta}_n)\}_{n=1}^N \overset{\text{i.i.d}}{\sim} \pi(\boldsymbol{x}, \boldsymbol{\theta})$ of size N, the expected number of training parameters $\boldsymbol{\theta}_n$ lying inside $\mathcal{S}_L$ (i.e. the expected number of "informative" training samples) is N times the quantity in Eq. (4), which likewise decays super-exponentially with dimension d. Thus, we expect the SBI algorithms to struggle as d gets larger, since they are tasked with locating a needle (the posterior mass) in an exponentially growing haystack (the prior support).

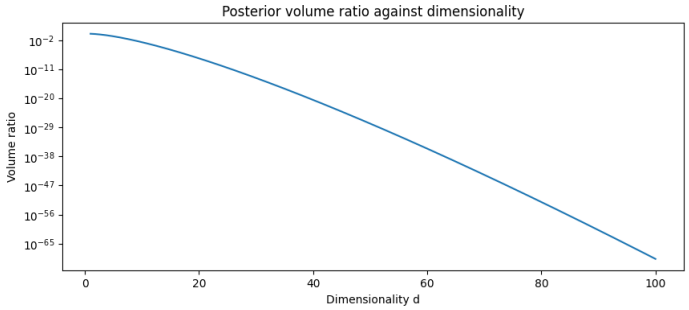

Figure 3. Volume ratio (4) against d. The vertical axis is on $\log_{10}$ scale.

Appendix C. Parameter Dimensionalities of Existing Studies

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

2. The numerator is the volume of a d-dimensional hypersphere, which is attributed to a paper published in 1839 by the Belgian mathematician Eugène Catalan.

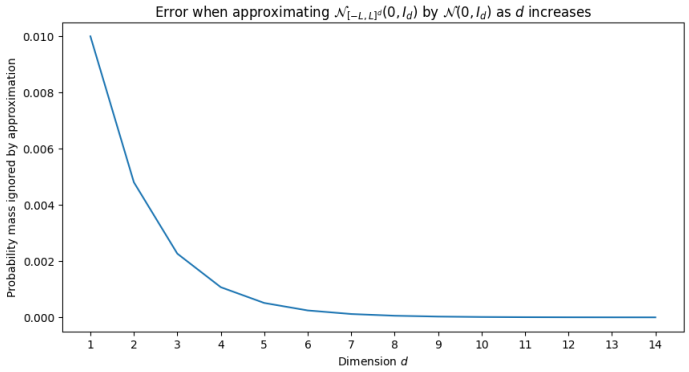

Figure 4. The amount of probability mass discarded when approximating a $\mathcal{N}_{[-L,L]^d}(\mathbf{0}, I_d)$ distribution by a $\mathcal{N}(\mathbf{0}, I_d)$ for various dimensions d (where L is computed using (3)). This mass discrepancy falls below 10^{-7} beyond the moderate dimension $d = 14$.

Paper	**Max $d_{\boldsymbol{\theta}}$**	**Comments**
SNPE-A (Papamakarios and Murray, 2016)	~ 5	-
SNPE-B (Lueckmann et al., 2017)	~ 12	-
SNPE-C (Greenberg et al., 2019)	~ 10	-
SNLE(-MCMC) (Papamakarios et al., 2019)	~ 12	-
SNLE(-VI) (Papamakarios et al., 2019)	~ 10	-
SSNL (Papamakarios et al., 2019)	~ 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)	~ 20	Ran $d_{\boldsymbol{\theta}} = 30$ but didn't check accuracy
Benchmarking SBI (Lueckmann et al., 2021)	~ 10	-
Trust Crisis in SBI (Hermans et al., 2022)	~ 5	-

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 D. Runtimes

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

Appendix E. 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 $d \in \{3, 10, 25, 50, 75, 100\}$.

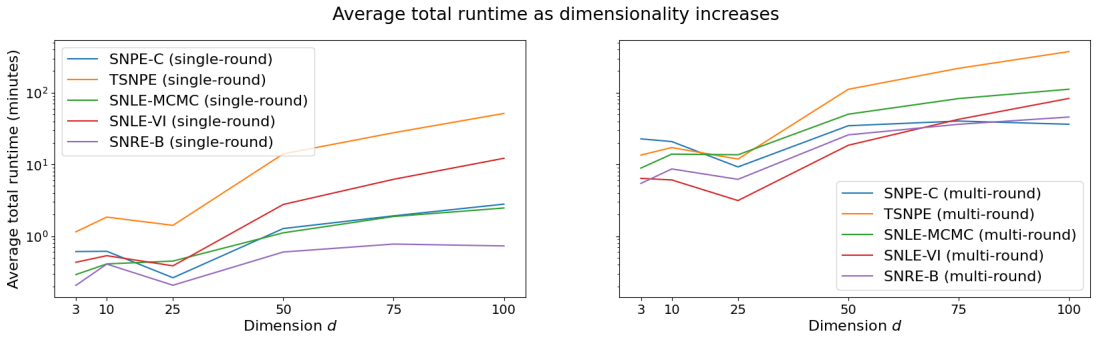

Figure 5. Total runtime of each algorithm (in minutes) as a function of dimensionality d, averaged across 10 repeated experiments. The left panel shows single-round variants, and the right panel shows 8-round variants.

Appendix F. Distribution of Squared Norms

Figure 7 shows a histogram representation of the distribution of $||\boldsymbol{\theta}||_2^2$ for 200000 draws $\boldsymbol{\theta} \sim \mathcal{N}_{[-L,L]^d}(\mathbf{0}, I_d)$ for various d. For all dimensions shown, the distribution of $||\boldsymbol{\theta}||_2^2$ is indistinguishable from a χ_d^2 distribution, which justifies the squared Euclidean norm metric used in our analysis.

Appendix G. 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:

[REDACTED FOR ANONYMITY. Link to repo will be provided in the camera-ready version.]

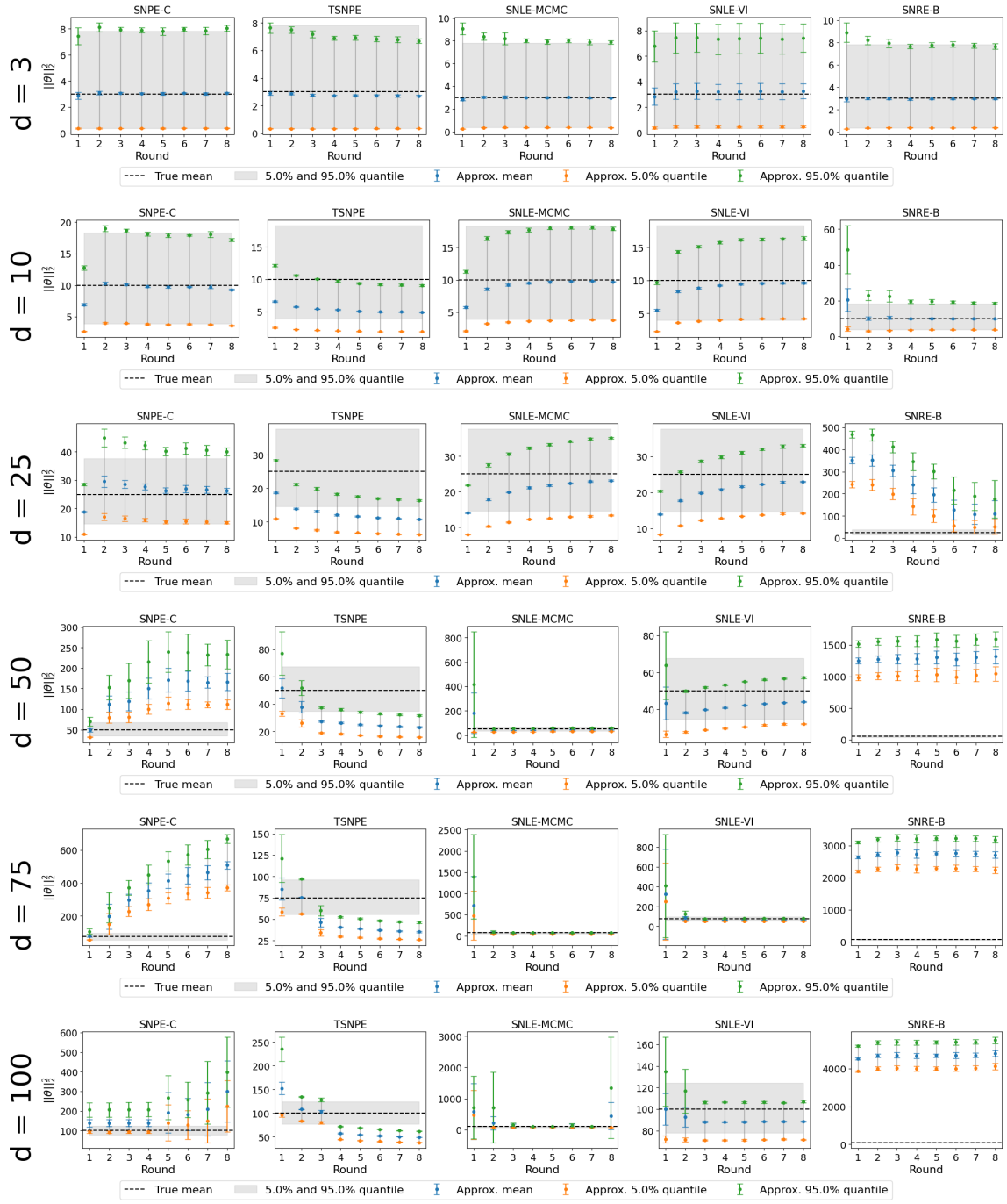

Figure 6. The mean, 0.05 quantile, and 0.95 quantile of the empirical distribution of the squared norms of samples from the approximate posterior at each round of training, for $d \in \{3, 10, 25, 50, 75, 100\}$. The true mean and quantiles (of a χ^2_d distribution) are the black dashed line and shaded 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.

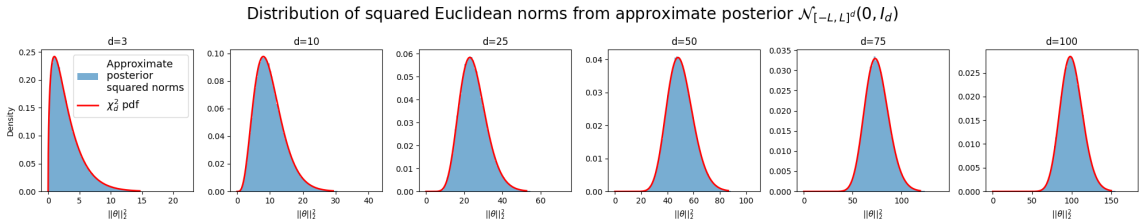

Figure 7. Histogram representing the distribution of $||\boldsymbol{\theta}||_2^2$ for 200000 draws $\boldsymbol{\theta} \sim \mathcal{N}_{[-L,L]^d}(\mathbf{0}, I_d)$ for various d. For all dimensions shown, the distribution of $||\boldsymbol{\theta}||_2^2$ is indistinguishable from a χ^2_d distribution.