ATOMIC POSTERIOR ENSEMBLES FOR SIMULATION-BASED INFERENCE

Sam Griesemer[1] **Willie Neiswanger**[1] **Yan Liu**[1]
[1]Thomas Lord Department of Computer Science, University of Southern California
{samgriesemer,neiswang,yanliu.cs}@usc.edu

ABSTRACT

Approximating parameter posteriors in likelihood-free settings is a practical challenge common to many scientific disciplines. While recent advances in both computer simulation and generative modeling have paved the way for tractable inference in high-fidelity environments, they often require prohibitively large sample sizes in practice. Sequential posterior estimation methods attempt to mitigate this by iteratively producing proposal distributions that refine the inverse model, but they lack explicit selection mechanisms for reducing information overlap in proposed simulations. In this work, we introduce a mutual information-based acquisition scheme for identifying informative simulation parameters, operating on disagreement in the parameter space across a weighted posterior ensemble of atomic proposals. Our approach crucially leverages only an inverse model, making it compatible with existing direct posterior estimation procedures. We demonstrate the potential of this method on several common simulation-based inference (SBI) benchmark tasks, and observe performance advantages over non-ensemble counterparts in low-data regimes.

1 INTRODUCTION

The intractability of likelihood functions is a common barrier to Bayesian inference in complex, real-world settings. While high-fidelity simulators may be readily available as models of the underlying generative process, they often do not admit a closed-form likelihood. Simulation-based inference (SBI) methods work around this limitation by assuming such models can only generate noisy samples, and attempt to compute or learn a posterior distribution from the resulting simulation data Cranmer et al. (2020).

Early success in this direction involved easy-to-use methods like extensions of kernel density estimation or Approximate Bayesian Computation (ABC) Rubin (1984); Beaumont et al. (2002). These methods struggle to scale with the dimensionality of most real-world applications, however, and neural network-based methods Papamakarios & Murray (2016); Lueckmann et al. (2017); Greenberg et al. (2019) have since been proposed to better address this challenge. Newer methods offer greater flexibility in the approximated probabilistic form used by the inference pipeline, (e.g., the posterior Papamakarios & Murray (2016), the likelihood Papamakarios et al. (2019), and likelihood ratio Hermans et al. (2020); Durkan et al. (2020)), as well as the underlying neural density estimator (NDE), including mixture density networks Bishop (1994) and normalizing flows Rezende & Mohamed (2016); Papamakarios et al. (2021), along with popular extensions (e.g., Real NVP Dinh et al. (2017), MAE Germain et al. (2015), MAF Papamakarios et al. (2018), etc).

Background Simulation-based inference seeks to approximate the posterior distribution $p(\theta|x)$ under a stochastic model $p(x|\theta)$. We assume $p(x|\theta)$ is defined implicitly via a simulation-based program, and while samples $x \sim p(x|\theta)$ can be drawn, direct evaluation of the likelihood value $p(x|\theta)$ is not possible. We further focus on the sequential (non-amortized) case, wherein an observational data point of interest x_o is known ahead of time, and we place particular emphasis on learning a high-quality approximation of $p(\theta|x_o)$.

Neural Posterior Estimation (NPE) methods approximate the posterior distribution directly by training an NDE $q_\phi(\theta|x)$ via maximum likelihood on samples $\{(\theta_i, x_i)\}_{i=1}^N$, where $\theta_i \sim p(\theta)$ and $x_i \sim$

$p(x|\theta_i)$, i.e., by minimizing the loss

$$\mathcal{L}(\phi) = \mathbb{E}_{\theta\sim p(\theta)}\mathbb{E}_{x\sim p(x|\theta)}\left[-\log q_\phi(\theta|x)\right],$$

where ϕ are the model's learnable parameters. So long as q_ϕ is sufficiently expressive, by Proposition 1 of Papamakarios & Murray (2016) $q_\phi(\theta|x)$ will converge to the true posterior $p(\theta|x)$ in the limit as $N \to \infty$.

Sequential Neural Posterior Estimation (SNPE) methods divide the inference process into multiple iterations of NPE, improving sample efficiency by leveraging the observation that $p(\theta|x = x_o)$ is typically much narrower than $p(\theta)$. A fully amortized method producing accurate $p(\theta|x)$ for any $x \in \mathcal{X}$ often requires impractically large numbers of simulation samples, including those from parameter values with low posterior density under x_o. Such an approximation is warranted in some settings, but when x_o is known, SNPE methods address this practical challenge by drawing θ values expected to be more informative about $p(\theta|x_o)$, using a proposal distribution $\tilde{p}(\theta)$ at each round that reflects the model's current posterior approximation $q_\phi(\theta|x = x_o)$.

Training the NDE q_ϕ on samples $\theta \sim \tilde{p}(\theta)$ when $\tilde{p}$ differs from the true prior results in convergence to a "proposal posterior,"

$$\tilde{p}(\theta|x) = p(\theta|x)\frac{\tilde{p}(\theta)p(x)}{\tilde{p}(x)p(\theta)} \tag{1}$$

rather than the true posterior (where $\tilde{p}(x) = \int_\Theta p(x|\theta)\tilde{p}(\theta)d\theta$). Different SNPE variants correct for this in distinct ways: ***SNPE-A*** Papamakarios & Murray (2016) trains $q_\phi(\theta|x)$ to approximate $\tilde{p}(\theta|x)$ at each round and applies importance reweighting afterward; ***SNPE-B*** Lueckmann et al. (2017) minimizes an importance-weighted loss directly alongside calibration kernels and Bayesian MDNs; and ***SNPE-C*** Greenberg et al. (2019), also known as Automatic Posterior Transformation (APT), enables directly training posterior models under flexible atomic proposals.

Contributions We introduce two variations of the Automatic Posterior Transformation (APT) scheme for sequential simulation-based inference, namely 1) importance-weighted posterior ensembles as a stable SBI approach and a tractable means of maintaining a posterior over model parameters $p(\phi|D)$, and 2) a mutual information-based adjustment to atomic proposals, prioritizing parameters expected to reduce model uncertainty while preserving convergence guarantees. We provide theoretical foundations for this approach and describe how it can be implemented in practice. We further demonstrate the method against APT without the proposed extensions in low-sample settings across several common SBI benchmarks.

2 METHODOLOGY

Atomic proposals with APT Greenberg et al. (2019) observe that minimizing the loss $\tilde{\mathcal{L}}(\phi) = -\sum_{i=1}^{N}\log\tilde{q}_\phi(\theta_i|x)$, where

$$\tilde{q}_\phi(\theta|x) = \frac{q_\phi(\theta|x)\left(\tilde{p}(\theta)/p(\theta)\right)}{\int_\Theta q_\phi(\theta|x)\left(\tilde{p}(\theta)/p(\theta)\right)d\theta} \tag{2}$$

produces $q_\phi(\theta|x) \to p(\theta|x)$ as $N \to \infty$. This is (again) by virtue of Prop. 1 of Papamakarios & Murray (2016); $\tilde{\mathcal{L}}(\phi)$ is minimized only when $\tilde{q}_\phi(\theta|x) = \tilde{p}(\theta|x)$, and thus $q_\phi(\theta|x) = p(\theta|x)$ by Eq. 1.

The authors further extend this scheme to arbitrary "atomic proposals," subverting the need for closed-form normalization constants in $\tilde{q}_\phi(\theta|x)$. The atomic loss scheme sets a uniform proposal prior $\tilde{p}(\theta) = U_\Theta$ over a fixed batch of parameters $\Theta = \{\theta_1, \ldots, \theta_M\}$, producing a categorical $\tilde{q}_\phi(\theta|x)$:

$$\tilde{q}_\phi(\theta|x) = \frac{q_\phi(\theta|x)/p(\theta)}{\sum_{\theta'\in\Theta} q_\phi(\theta'|x)/p(\theta')}$$

The loss $\tilde{\mathcal{L}}$ now no longer relies on any earlier choice of proposal prior (once Θ has been fixed), and as before, $\mathbb{E}_{\theta\sim U_\Theta, x\sim p(x|\theta)}[\tilde{\mathcal{L}}]$, is minimized when $\tilde{q}_\phi(\theta|x)$ is the true proposal posterior $\tilde{p}(\theta|x)$.

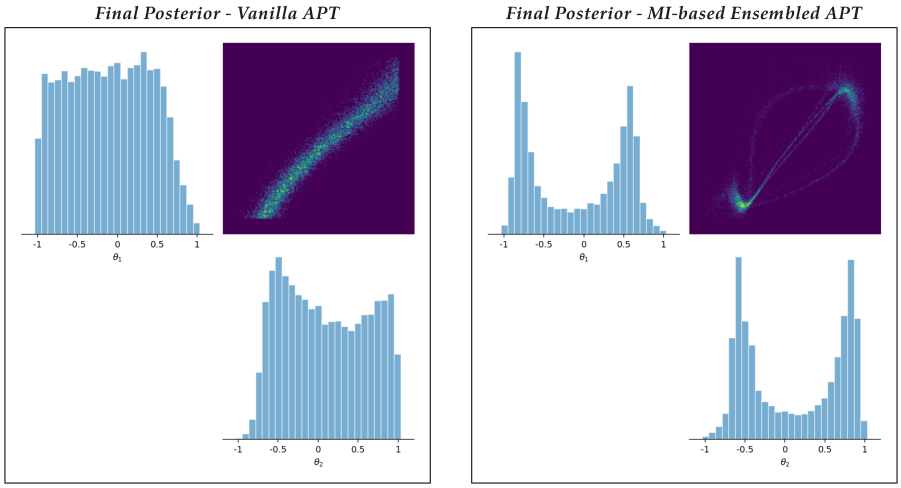


Figure 1: Visualization of the final-round posterior approximations under regular APT/SNPE-C (left) and the proposed MI-based ensembled APT variant (right) on the Two Moons task. Both runs were executed under the same random seed (and thus draw identical initial samples from the prior), and took place over four rounds, with 64 samples per round. These plots represent the final posterior models produced by each method from a randomly selected trial in the benchmark evaluations discussed in Section 3.

Importance Weighted Posterior Ensembles While APT ensures the NDE model converges to the true posterior in the limit as $N \to \infty$, in practice we often face several challenges: the NDE can get "stuck" in certain regions of the parameter space (and miss modes, for instance), or simply require many samples before the true posterior shape begins to emerge. Inspired by extreme variability in early rounds of SNPE-C (especially in low-data settings), we seek to stabilize round-by-round posterior estimates by ensembling several posterior approximations.

In particular, we define some number of NDE models K to treat as individual atomic proposals. The distribution from which each round's parameters are drawn i.i.d. is then the weighted ensemble of these proposals. That is, for round $r+1$, we have the proposal prior

$$\bar{q}_{\Phi}^{(r+1)}(\theta|x) = \sum_{j=1}^{K} w_j q_{\phi_j}(\theta|x)$$

where the component weight

$$w_j = \frac{\prod_i^{N_r} q_{\phi_j}\left(\theta_i^{(r)}|x_i^{(r)}\right)}{\sum_{k=1}^{K}\prod_i^{N_r} q_{\phi_k}\left(\theta_i^{(r)}|x_i^{(r)}\right)},$$

i.e., the self-normalized joint support of the j^{th} "component posterior" on the observed simulation data $\{(\theta_i, x_i)\}_{i=1}^{N_r}$ from the previous round (where N_r is the number of samples observed at round r).

Note that we generally view this scheme as an approximation of a true "marginal posterior"

$$q(\theta|x, D) = \int_{\Phi} q_{\phi}(\theta|x)p(\phi|D)d\phi,$$

independent of any particular selection of model weights ϕ, conditional on all observed simulation data D. Rather than explicitly maintain a model weight posterior $p(\phi|D)$, we effectively importance re-weight our K fixed atomic proposal samples ϕ_j according to how likely they would be under $p(\phi|D^{(r)})$ at each round r.

Once samples are drawn and simulation results are collected for a given round, the shared, fixed batch Θ is used to update each model independently according to the standard APT loss. The shared parameter "suggestion" step takes place when the ensembled proposal prior is constructed

pre-simulation, and the collective "wisdom" is disseminated back to each model post-simulation. By Eq. 2 we know that each model converges to the true posterior, but due to variability in ϕ_j, this may take place at different rates. Figure 1 highlights this in part, empirically demonstrating how several underlying models can contribute to a steady rate of convergence while any singular model may converge more slowly on its own. We discuss this in greater detail in Section 3.

We additionally note that the primary cost of maintaining several independent posterior NDEs is the increased training burden. Round-by-round model re-training is a bottleneck generally in the inference process, however, as we can't proceed to the next round until the NDE re-training is entirely complete. Training several models in parallel during this stage is thus a fairly simple way to minimize the runtime impact of additional models, and yields virtually identical wallclock times as single-model APT when K is reasonably small (e.g., $K = 4$ or $K = 8$, for instance, the latter being used in reported results from Section 3).

Prospective Mutual Information As noted by the APT authors Greenberg et al. (2019), training with an atomic loss can be intuitively likened to "quizzing" the model $q_\phi(\theta|x)$ with multiple choice questions. This can be seen with Eq. 2: given the M possible options in the batch Θ, the model must learn to correctly assign the most mass to the "correct" provided (θ, x) pair.

As an extension to this analogy, we posit the following: which questions are the best to ask? Put another way, which questions stand to "teach" our model the most about the target posterior $p(\theta|x_o)$? Here we turn to mutual information as a general measure of how much we expect to learn from certain observations, and weight the *prospective* impact of those outcomes by how likely the model currently believes them to be.

In particular, we want to calculate

$$\mathbb{I}[\phi; \theta|x_o, D'] = \mathbb{E}_{p(\phi|x_o, D')}\left[\mathcal{D}_{KL}\left(p(\theta|\phi, x_o, D')||p(\theta|x_o, D')\right)\right] \tag{3}$$

$$= \mathbb{H}\left[\theta|x_o, D'\right] - \mathbb{E}_{\phi\sim p(\phi|D')}\left[\mathbb{H}\left[\theta|\phi, x_o\right]\right], \tag{4}$$

i.e., the mutual information between simulation parameters θ and posterior model parameters ϕ, conditional on observational data of interest x_o and a prospective new dataset $D' = D \cup \{(\theta', x_o)\}$. This term in part mirrors the "active SBI" motivations of Griesemer et al. (2024), but we crucially aim to approximate the mutual information directly as a strategy for sampling guidance (as opposed to strict acquisition optimization of an alternative objective). By calculating this term under various simulation parameter candidates θ', we're attempting to identify those points that leave relatively little *remaining* information to learn about ϕ through subsequent draws of θ. We additionally note the direct connection (additional details in Appendix A.1) between the mutual information and the expected divergence between the ensemble posterior $p(\theta|x_o, D')$ and component models $p(\theta|x_o, \phi)$; parameters θ' that cause a significant shift represent regions of the parameter space where component models disagree about how to assign posterior density. This coincides with our aims of identifying valuable "questions" to pose when training via an atomic loss, as model disagreement indicates some difficulty in assigning a stable likelihood to that region across NDE models.

Note that D' makes a concrete assumption as to the outcome of any candidate θ': because we only assume access to an inverse model $q_\phi(\theta|x)$, we cannot generate alternative possible simulation outputs (e.g., $x' \sim p(x|\theta')$, as we might if a surrogate likelihood was available, for instance). Instead, we assume θ' will yield x_o, and simply weight the resulting mutual information estimate according to $q_\phi(\theta'|x_o)$, the currently available posterior belief of such an outcome. We can therefore naturally re-weight this model before its use as a proposal prior in a manner that preserves the full (true) prior support, e.g.,

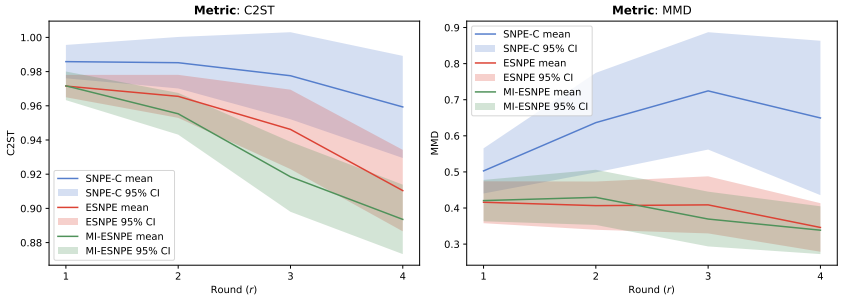
$$\tilde{p}(\theta') \propto q_\phi(\theta'|x_o)\exp\left(-\mathbb{I}[\phi; \theta|x_o, D']\right),$$

where we note the negative exponentiation of the mutual information reflects the goal of minimizing this term in the face of our beliefs. This preserves posterior convergence guarantees broadly applicable to atomic proposals, as $\tilde{p}(\theta') > 0$ when θ' is in the prior support.

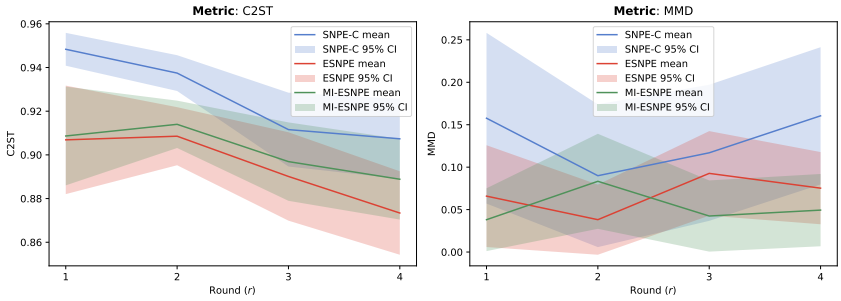
A critical practical challenge when calculating this term is the need to leverage only the current model (conditional dependence on D), rather than an explicit dependence on a newly trained model. That is, terms conditional on D' are problematic if we can't expect to re-train NDE models under *every* parameter candidate θ' for which we'd like to estimate the mutual information, and this becomes quickly infeasible for even small SBI tasks. We provide a detailed way to approach this in Appendix A.1.

3 EXPERIMENTS

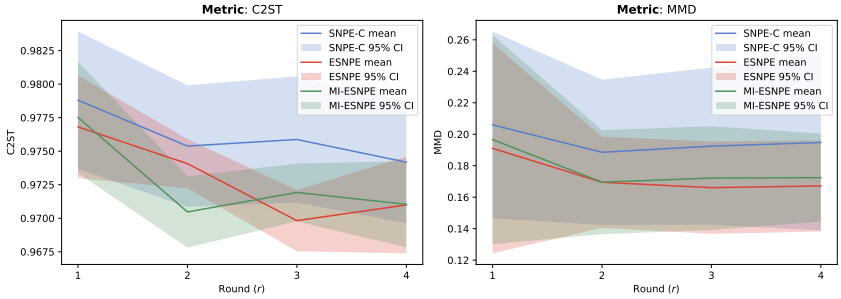
SBI Benchmarks To quantify the practical impact of our proposed contributions, we compare the performance of vanilla APT, ensembled APT, and the MI-based selection variant on three common SBI benchmarks: Two Moons, SLCP Distractors, and Bernoulli GLM Lueckmann et al. (2021). These tasks vary in difficulty for direct posterior estimation methods, and each present with a known posterior over which we can calculate key measures that indicate distribution fit. We report the maximum mean discrepancy (MMD) and a classifier 2-sample test (C2ST) across each method and setting in Figure 2. Note that smaller values are better for each metric (C2ST ranges between $0.5-1.0$). Appendix B provides additional details of NDE model hyperparameters and software/hardware details used to carry out experiments.



(a)



(b)



(c)

Figure 2: C2ST and MMD scores for vanilla APT/SNPE-C, ensembled APT (ESNPE), and MI-based selection variant (MI-ESNPE) across various SBI benchmark tasks. Subplot **(a)** shows results for the *Bernoulli GLM* task, subplot **(b)** for the *Two Moons* task, and subplot **(c)** for *SLCP Distractors*. Each plot depicts the mean and 95% confidence interval for each method and metric calculated across 10 randomly seeded runs.

Discussion Figure 2 reports results across few-sample inference runs: we perform four rounds of sequential inference and collect just 64 samples per round (256 total samples encountered after the final round). We repeat this across ten trials, using a fixed set of ten seeds to ensure all methods get

a "fair start," each seeing the same initial samples from the prior for a given seed. The ensemble methods comprised 8 individual NDE models each.

Across all settings and on both metrics, we find the mean score of APT/SNPE-C never once outperforms the mean score of either ensemble-based approaches, on any inference round. This demonstrates a fairly clear performance advantage, and suggests ensembled methods may be a strong choice more broadly, especially when sample efficiency is a concern: having several posterior models appears to balance noise encountered in early rounds and facilitate more targeted simulation samples. Although not reported here, it's perhaps worth noting the impact of the importance weighting scheme for updating the ensemble; when just maintaining a uniform mixture of APT proposals and routinely re-training each round, the resulting method regularly underperforms compared to individual APT runs. That is to say, drawing samples according to collections of NDE models that each individually converge to the true posterior is generally not a surefire means of attaining better results.

While the performance advantage varies across metric and task, it is perhaps most stark on the Bernoulli GLM task. The marginal difference among the ensemble-based approaches, however, is relatively minimal by comparison: neither appears to dominate across any particular dimension, and while the MI-based scheme outperforms on Bernoulli GLM, it is less consistent on the other tasks.

4 Conclusion

Recent advances in SBI methods have significantly improved the feasibility of performing accurate likelihood-free inference in challenging real-world settings. Operating in low-data settings remains a core practical consideration, however, especially when working with slow/expensive simulation systems. In this work, we proposed a general ensembling scheme for collectively training and combining groups of atomic proposal priors across rounds of sequential posterior estimation. We extended this scheme by proposing a means of approximating the information content of prospective simulation parameter candidates, further capitalizing on model uncertainty and guiding sampling toward useful regions of the parameter space. Across several common SBI benchmarks, we observe the ensemble schemes outperform standard APT on two relevant measures of posterior quality when operating under tight simulation budgets. These results indicate carefully balanced ensemble methods may be a strong contender for tackling challenging SBI tasks in a more stable, sample-efficient manner.

REFERENCES

Mark A. Beaumont, Wenyang Zhang, and David J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. ISSN 0016-6731. URL https://www.genetics.org/content/162/4/2025.

Christopher M. Bishop. Mixture density networks. Technical report, 1994.

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

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

Conor Durkan, Iain Murray, and George Papamakarios. On contrastive learning for likelihood-free inference, 2020.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation, 2015.

David S. Greenberg, Marcel Nonnenmacher, and Jakob H. Macke. Automatic posterior transformation for likelihood-free inference, 2019.

Sam Griesemer, Defu Cao, Zijun Cui, Carolina Osorio, and Yan Liu. Active sequential posterior estimation for sample-efficient simulation-based inference. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2412.05590.

Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free mcmc with amortized approximate ratio estimators, 2020.

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, 2017.

Jan-Matthis Lueckmann, Jan Boelts, David Greenberg, Pedro Goncalves, and Jakob Macke. Benchmarking simulation-based inference. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 343–351. PMLR, 13–15 Apr 2021.

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, 2018.

George Papamakarios, David C. Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows, 2019.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

Donald B. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172, 1984. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2240995.

Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, 2020. doi: 10.21105/joss.02505. URL https://doi.org/10.21105/joss.02505.

A Appendix

A.1 Mutual Information Approximation

In this section, we decompose the mutual information as presented in Eq. 3 and arrive at a means of approximating entropy terms when we only permit a conditional dependence on the current dataset D (i.e., assuming we can't explicitly perform model updates under every candidate θ' in practice).

Let $D' = D \cup \{(\theta', x_o)\}$. Then we have the mutual information between inference parameters θ and posterior model parameters ϕ:

$$\begin{aligned}
\mathbb{I}[\phi; \theta | x_o, D'] &= \int_\Phi \int_\Theta p(\theta, \phi | x_o, D') \log \left(\frac{p(\theta, \phi | x_o, D')}{p(\theta | x_o, D') p(\phi | x_o, D')} \right) d\theta d\phi \\
&= \int_\Phi \int_\Theta p(\theta | \phi, x_o, D') p(\phi | x_o, D') \log \left(\frac{p(\theta | \phi, x_o, D') p(\phi | x_o, D')}{p(\theta | x_o, D') p(\phi | x_o, D')} \right) d\theta d\phi \\
&= \int_\Phi p(\phi | x_o, D') \left[\int_\Theta p(\theta | \phi, x_o, D') \log \left(\frac{p(\theta | \phi, x_o, D')}{p(\theta | x_o, D')} \right) d\theta \right] d\phi \\
&= \int_\Phi p(\phi | x_o, D') \left[\mathcal{D}_{KL} \left(p(\theta | \phi, x_o, D') || p(\theta | x_o, D') \right) \right] d\phi \\
&= \mathbb{E}_{p(\phi | x_o, D')} \left[\mathcal{D}_{KL} \left(p(\theta | \phi, x_o, D') || p(\theta | x_o, D') \right) \right].
\end{aligned}$$

The mutual information can also be expressed in terms of entropies:

$$\begin{aligned}
\mathbb{I}[\phi; \theta | x_o, D'] &= \int_\Theta \int_\Phi p(\phi, \theta | x_o, D') \log \left(\frac{p(\phi, \theta | x_o, D')}{p(\phi | x_o, D') p(\theta | x_o, D')} \right) d\phi d\theta \\
&= \int_\Theta \int_\Phi p(\phi, \theta | x_o, D') \log \left(\frac{p(\phi | \theta, x_o, D') p(\theta | x_o, D')}{p(\phi | x_o, D') p(\theta | x_o, D')} \right) d\phi d\theta \\
&= \int_\Theta \int_\Phi p(\phi, \theta | x_o, D') \log p(\phi | \theta, x_o, D') d\phi d\theta - \int_\Theta \int_\Phi p(\phi, \theta | x_o, D') \log p(\phi | x_o, D') d\phi d\theta \\
&= -H \left[\phi | \theta, x_o, D' \right] - \int_\Theta \int_\Theta p(\theta | \phi, x_o, D') p(\phi | x_o, D') \log p(\phi | x_o, D') d\phi d\theta \\
&= -H \left[\phi | \theta, x_o, D' \right] - \int_\Theta p(\phi | x_o, D') \log p(\phi | x_o, D') \left[\int_\Theta p(\theta | \phi, x_o, D') d\theta \right] d\phi \\
&= -\mathbb{H} \left[\phi | \theta, x_o, D' \right] + \mathbb{H} \left[\phi | x_o, D' \right] \\
&= \mathbb{H} \left[\phi | x_o, D' \right] - \mathbb{E}_{\theta \sim p(\theta | D')} \left[\mathbb{H} \left[\phi | \theta, x_o \right] \right]
\end{aligned}$$

By symmetry of the joint factorization, we have generally

$$\begin{aligned}
\mathbb{I}[\phi; \theta | x_o, D'] &= \mathbb{E}_{p(\theta | x_o, D')} \left[\mathcal{D}_{KL} \left(p(\phi | \theta, x_o, D') || p(\phi | x_o, D') \right) \right] \\
&= \mathbb{E}_{p(\phi | x_o, D')} \left[\mathcal{D}_{KL} \left(p(\theta | \phi, x_o, D') || p(\theta | x_o, D') \right) \right] \\
&= \mathbb{H} \left[\theta | x_o, D' \right] - \mathbb{E}_{\phi \sim p(\phi | D')} \left[\mathbb{H} \left[\theta | \phi, x_o \right] \right] \\
&= \mathbb{H} \left[\phi | x_o, D' \right] - \mathbb{E}_{\theta \sim p(\theta | D')} \left[\mathbb{H} \left[\phi | \theta, x_o \right] \right]
\end{aligned}$$

We then seek to approximate

$$\mathbb{I}[\phi; \theta | x_o, D'] = \mathbb{H} \left[\theta | x_o, D' \right] - \mathbb{E}_{\phi \sim p(\phi | D')} \left[\mathbb{H} \left[\theta | \phi, x_o \right] \right]$$

We first note that, by Bayes' rule,

$$p(\phi|D') = \frac{p(\theta'|x_o, \phi)p(\phi|D)}{p(\theta'|x_o, D)},$$

which we use to rewrite a posterior dependence on D' in terms of D:

$$\begin{aligned}
p(\theta|x_o, D') &= \int p(\theta|x_o, \phi)p(\phi|D')d\phi \\
&= \int p(\theta|x_o, \phi)\left[\frac{p(\theta'|x_o, \phi)p(\phi|D)}{p(\theta'|x_o, D)}\right]d\phi \\
&= \frac{1}{p(\theta'|x_o, D)}\int p(\theta|x_o, \phi)p(\theta'|x_o, \phi)p(\phi|D)d\phi \\
&= \frac{1}{p(\theta'|x_o, D)}\mathbb{E}_{\phi|D}\left[p(\theta|x_o, \phi)p(\theta'|x_o, \phi)\right],
\end{aligned}$$

The following terms can then be used to approximate the mutual information $\mathbb{I}[\phi; \theta|x_o, D']$ when we just have access to a model $p(\theta|x_o, D)$ and model weight posterior $p(\phi|D)$ trained up to data D (rather than D' explicitly). For the "marginal entropy":

$$\begin{aligned}
\mathbb{H}\left[\theta|x_o, D'\right] &= \mathbb{E}_{\theta|x_o, D'}\left[-\log p(\theta|x_o, D')\right] \\
&= -\mathbb{E}_{\theta|x_o, D}\left[\left(\frac{p(\theta|x_o, D')}{p(\theta|x_o, D)}\right)\log p(\theta|x_o, D')\right] \\
&= -\mathbb{E}_{\theta|x_o, D}\left[\left(\frac{\mathbb{E}_{\phi|D}\left[p(\theta'|x_o, \phi)p(\theta|x_o, \phi)\right]}{p(\theta'|x_o, D)p(\theta|x_o, D)}\right)\log\left(\frac{\mathbb{E}_{\phi|D}\left[p(\theta'|x_o, \phi)p(\theta|x_o, \phi)\right]}{p(\theta'|x_o, D)}\right)\right] \\
&\approx -\frac{1}{M}\sum_{\theta_i \sim \theta|x_o, D}^{M}\left[\left(\frac{P'_K}{p(\theta'|x_o, D)p(\theta_i|x_o, D)}\right)\log\left(\frac{P'_K}{p(\theta'|x_o, D)}\right)\right],
\end{aligned}$$

where

$$P'_K = \frac{1}{K}\sum_{\phi_j \sim \phi|D}^{K}\left[p(\theta'|x_o, \phi_j)p(\theta_i|x_o, \phi_j)\right],$$

and the "expected conditional entropy"

$$\begin{aligned}
\mathbb{E}_{\phi|D'}\left[\mathbb{H}\left[\theta|\phi, x_o\right]\right] &= \mathbb{E}_{\phi|D'}\left[\mathbb{E}_{\theta|x_o, \phi}\left[-\log p(\theta|\phi, x_o)\right]\right] \\
&= \mathbb{E}_{\phi|D}\left[\frac{p(\theta'|x_o, \phi)}{p(\theta'|x_o, D)}\mathbb{E}_{\theta|x_o, \phi}\left[-\log p(\theta|\phi, x_o)\right]\right] \\
&\approx -\frac{1}{MK}\sum_{\phi_i \sim \phi|D}^{K}\left[\frac{p(\theta'|x_o, \phi_i)}{p(\theta'|x_o, D)}\sum_{\theta_j \sim \theta|x_o, \phi_i}^{M}\log p(\theta_j|\phi_i, x_o)\right]
\end{aligned}$$

B Experimental Details

The NDE model used for all experiments is a conditional masked autoregressive flow (MAF) comprised of 5 MADE transforms, each with two 50-unit residual blocks. Each MAF was trained with 10 atoms, and we use most of the training core as implemented in the `sbi` Python package Tejero-Cantero et al. (2020). No embedding networks were employed across any of the tasks, and the

`sbibm` package Lueckmann et al. (2021) was adapted to provide true posterior references and metric calculations for each reported task.

All experimentation code was written and packaged in Python 3.11. Experiments were performed on our own local hardware, a Linux-based machine running an Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz 64GB memory and NVIDIA GeForce RTX 2080 Ti.