Truncated Marginal Neural Ratio Estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Parametric stochastic simulators are ubiquitous in science, often featuring high-dimensional input parameters and/or an intractable likelihood. Performing Bayesian parameter inference in this context can be challenging. We present a neural simulator-based inference algorithm which simultaneously offers simulation efficiency and fast empirical posterior testability, which is unique among modern algorithms. Our approach is simulation efficient by simultaneously estimating low-dimensional marginal posteriors instead of the joint posterior and by proposing simulations targeted to an observation of interest via a prior suitably truncated by an indicator function. Furthermore, by estimating a locally amortized posterior our algorithm enables efficient empirical tests of the robustness of the inference results. Such tests are important for sanity-checking inference in real-world applications, which do not feature a known ground truth. We perform experiments on a marginalized version of the simulation-based inference benchmark and two complex and narrow posteriors, highlighting the simulator efficiency of our algorithm as well as the quality of the estimated marginal posteriors.

1 Introduction

Parametric stochastic simulators are ubiquitous in science [1–3] and using them to solve the Bayesian inverse problem is of general interest. Likelihood-based methods like Markov chain Monte Carlo (MCMC) [4, 5] or nested sampling [6] are applicable when the likelihood is tractable. It is equally common that the likelihood is only implicitly defined by the simulator or is inefficient to compute. For this so-called *likelihood-free* or *simulation-based* inference, the traditional approach is Approximate Bayesian Computation (ABC) [7].

Simulation-based inference (SBI) has been an open research topic for years. Deep learning has accelerated progress in the field [8–11]. Proposed algorithms that learn the likelihood [10] or the posterior [11–13] utilize a density estimator. The likelihood-to-evidence ratio [8] can be learned via a classification-based technique. Refs. [8] and [11] were brought into a unified framework by [14].

High-fidelity simulators often have many parameters and/or an intractable likelihood function, which can make inference notoriously difficult. Practitioners are usually faced with observational data and an expensive stochastic simulator, without access to the ground truth posterior. They want a testably accurate posterior estimate without extreme simulation expense. With existing methods, the practitioner must choose between increased accuracy per simulation or efficient empirical testability. We provide a method which offers both simultaneously with a balance that can be tuned by a hyperparameter. Three attributes contribute to this goal:

Targeted inference. Focusing simulations on the parameter regions that are most relevant for the inference problem and target data is more efficient. This is particularly true when most posterior density is concentrated compared to the prior's density.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Table 1: Comparison of SBI methods, including our proposed TMNRE. See text for details.

Method / Property	Likelihood-based	ABC	NRE	NPE	SNRE	SNPE	**TMNRE**
Targeted inference	✓	•	✗	✗	✓	✓	✓
Simulator efficient *direct* marginals	✗	✓	•	•	✗	✗	✓
(Local) amortization	✗	✗	✓	✓	✗	✗	✓

Marginal posteriors instead of the joint. Scientific insight is often based on a low dimensional marginalization of the posterior with nuisance parameters removed. The full joint posterior can be unnecessarily informative. Targeting marginals directly, by estimating only the marginal posterior for the parameters of interest, is simpler and sufficient for most purposes.

Consistency checks through local amortization. Practitioners are interested in testing the quality of inference methods [15–17]. One such test is to compare the empirical and nominal contained mass of estimated credible regions. *Amortized* methods learn the posterior for any data, generated by any parameter, facilitating empirical study of the nominal credible regions on fabricated data. Still, learning an amortized posterior is excessive if only a small subset of parameters are consistent with a target observation.

We propose the concept of *local amortization* to learn the posterior on said subset, combining simulator efficiency of targeted inference with the testability of amortization. Both are critical components for enabling trustworthy scientific results.

Our contribution. We propose an algorithm that simultaneously achieves all three of the above aspects: Truncated Marginal Neural Ratio Estimation (TMNRE) approximates the marginal likelihood-to-evidence ratio in a sequence of rounds. As a basis, we adopt likelihood-to-evidence estimation proposed in [8]. Our iterative scheme is loosely inspired by likelihood-based nested sampling [6, 18, 19] since we generate training data drawn from a nested sequence of truncated priors in multiple rounds. Our algorithm (a) preferentially generates simulations in relevant regions of the parameter space, (b) allows generation of all marginals of interest simultaneously and in parallel from the same training data, and (c) yields posteriors that are locally amortized in a constrained region around the posterior, enabling empirical self-consistency test of the inference results.

Related work. In Table 1, we compare the properties and features of a selection of deep-learning based simulator-based inference methods that are directly relevant for our work. Sampling from regions of highest probability density is baked into most *likelihood-based* methods [4–6, 18–22]. Amortization is generally not available with these methods because they sample from a particular posterior. *Approximate Bayesian Computation (*ABC*)* is a rejection sampling technique where proposed samples from the generative model are accepted based on a user defined distance criterion comparing generated data to the observation of interest. Two important methods include REJ-ABC [7] where the proposal distribution is simply the parameter prior and SMC-ABC [23, 24] where the proposal is iteratively refined.

Likelihood-free inference can be cast as a conditional density estimation problem targeting either the posterior directly [11–14] or the likelihood [10, 25]. This technique requires a density estimator, normally implemented as a mixture density network [26] or a normalizing flow [27, 28]. *Neural Likelihood Estimation* (NLE) performs well on benchmark tasks but must learn a density representation of the data in an unsupervised setting. Modern variants of *Neural Posterior Estimation* (NPE) [14] have become effective enough to offer an alternative marginal estimation method for practitioners. A simple implementation was discussed in [29].

*Amortized Approximate Ratio Estimators / Neural Ratio Estimation (*NRE*):* Binary classification allows estimation of the likelihood ratio between two hypotheses [9, 30–34] and was most famously applied to Generative Adversarial Networks [35]. Ref. [8] noted that naive application in the likelihood-free setting was unsatisfactory because the mathematically arbitrary choice of reference hypothesis significantly affected empirical MCMC results. Comparing likelihoods from jointly drawn $(\boldsymbol{x}, \boldsymbol{\theta}) \sim p(\boldsymbol{x}, \boldsymbol{\theta})$ and marginally drawn $(\boldsymbol{x}, \boldsymbol{\theta}) \sim p(\boldsymbol{x})p(\boldsymbol{\theta})$ samples, where $\boldsymbol{x}$ and $\boldsymbol{\theta}$ refer respectively to simulated data and simulator parameters, addresses the issue. Ref. [14] cast NRE and NPE in a unifying framework by adapting the loss function to contrast several possible hypothetical parameters.

In this paper we refer to the algorithm described in [8] as NRE or NRE_A while the likelihood ratio algorithm described in [14] is referred to as NRE_B.

Direcly estimating the marginal posterior distribution directly has been mentioned [8] and explored [29, 36]. Moment networks [29] produce the (central) moments of the posterior distribution, without calculating the density explicitly, via a hierarchy of neural networks trained on a regression problem. Ref. [29] also introduced a method which learns marginal posteriors with normalizing flows.

Sequential Methods: The neural likelihood-free methods generally offer a so-called sequential formulation that targets the posterior of a particular observation $\boldsymbol{x}_o$ [8, 10, 14]. Rather than drawing samples from the prior, the simulation budget is divided between rounds and the previous round's posterior is used as the new proposal distribution for the next round. This method increases simulation efficiency, but does not allow for amortization. Importantly, sequential methods can become highly inefficient when targeting multiple marginal posteriors directly because the previous round's marginal posterior does not update beliefs about the other parameters. A full parameter vector is necessary to run the simulator, thus defeating the purpose for all nuisance (marginalized-over) parameters.

2 Method

We aim to estimate any marginal posterior of interest using an approximate marginal likelihood-to-evidence ratio. Although we normally compute every one and two dimensional marginal posterior for visualization purposes, our method is not limited to this restriction. Let us define the object of study.

Let parametric stochastic simulator $\boldsymbol{g}$ be a nonlinear function that maps a vector of real parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_D)$ and a stochastic latent state $\boldsymbol{z}$ to an observation $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{z})$. The likelihood function is therefore $p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \int \delta(\boldsymbol{x} - \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{z}))\, p(\boldsymbol{z} \mid \boldsymbol{\theta})\, d\boldsymbol{z}$, where $\delta(\cdot)$ denotes the Dirac delta. We consider a *factorizable prior* $p(\boldsymbol{\theta}) = p(\theta_1) \cdots p(\theta_D)$ over the parameters. The joint posterior is given via Bayes' rule as $p(\boldsymbol{\theta} \mid \boldsymbol{x}) = p(\boldsymbol{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) / p(\boldsymbol{x})$, where $p(\boldsymbol{x})$ is the evidence. Our goal is to efficiently compute arbitrary marginal posteriors, $p(\boldsymbol{\vartheta} \mid \boldsymbol{x})$. Here, $\boldsymbol{\vartheta}$ are the parameters of interest, and we denote all other (nuisance) parameters by $\boldsymbol{\eta}$, such that $\boldsymbol{\theta} = (\boldsymbol{\vartheta}, \boldsymbol{\eta})$. The marginal posterior is obtained from the joint distribution $p(\boldsymbol{\vartheta}, \boldsymbol{\eta} \mid \boldsymbol{x}) \coloneqq p(\boldsymbol{\theta} \mid \boldsymbol{x})$ by integrating over all components of $\boldsymbol{\eta}$,

$$p(\boldsymbol{\vartheta} \mid \boldsymbol{x}) = \int p(\boldsymbol{\vartheta}, \boldsymbol{\eta} \mid \boldsymbol{x})\, d\boldsymbol{\eta} = \frac{\int p(\boldsymbol{x} \mid \boldsymbol{\vartheta}, \boldsymbol{\eta}) p(\boldsymbol{\eta})\, d\boldsymbol{\eta}}{p(\boldsymbol{x})} p(\boldsymbol{\vartheta}) \coloneqq \frac{p(\boldsymbol{x} \mid \boldsymbol{\vartheta})}{p(\boldsymbol{x})} p(\boldsymbol{\vartheta}). \tag{1}$$

where we used Bayes' rule, prior factorizibility, and defined the marginal likelihood $p(\boldsymbol{x} \mid \boldsymbol{\vartheta})$.

2.1 Marginal Neural Ratio Estimation (MNRE)

This paper considers the set of one and two dimensional marginal posteriors and their corresponding parameters of interest. Given parameter vector $\boldsymbol{\theta} \in \mathbb{R}^D$, define the set of all parameters associated with the one dimensional marginal posteriors by $\Theta_1 \coloneqq \{\theta_1, \dots, \theta_D\}$. We do something similar, up to symmetry, for all two dimensional marginal posteriors $\Theta_2 \coloneqq \big\{(\theta_i, \theta_j) \in \mathbb{R}^2 \;\big|\; i = 1, \dots, D,\, j = i+1, \dots, D\big\}$. We set our marginals of interest $\{\boldsymbol{\vartheta}_k\} \coloneqq \Theta_1 \cup \Theta_2$ but in the general case, $\{\boldsymbol{\vartheta}_k\}$ can be any set of marginals that the practitioner desires. For every $\boldsymbol{\vartheta}_k$ we use NRE [8] to estimate the corresponding marginal likelihood-to-evidence ratio

$$r_k(\boldsymbol{x}, \boldsymbol{\vartheta}_k) \coloneqq \frac{p(\boldsymbol{x} \mid \boldsymbol{\vartheta}_k)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}, \boldsymbol{\vartheta}_k)}{p(\boldsymbol{x}) p(\boldsymbol{\vartheta}_k)} = \frac{p(\boldsymbol{\vartheta}_k \mid \boldsymbol{x})}{p(\boldsymbol{\vartheta}_k)} \;. \tag{2}$$

To this end, we train binary classifiers $\hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)$ to distinguish jointly drawn parameter-simulation pairs $(\boldsymbol{x}, \boldsymbol{\vartheta}_k) \sim p(\boldsymbol{x}, \boldsymbol{\vartheta}_k)$ from marginally drawn parameter-simulation pairs $(\boldsymbol{x}, \boldsymbol{\vartheta}_k) \sim p(\boldsymbol{x}) p(\boldsymbol{\vartheta}_k)$, where ϕ represents the parameters of the classifier. A Bayes optimal classifier ρ_k would recover the density $\rho_k(\boldsymbol{x}, \boldsymbol{\vartheta}_k) = \frac{p(\boldsymbol{x}, \boldsymbol{\vartheta}_k)}{p(\boldsymbol{x}, \boldsymbol{\vartheta}_k) + p(\boldsymbol{x}) p(\boldsymbol{\vartheta}_k)}$. Then the ratios of interest can be estimated by

$$\hat{r}_k(\boldsymbol{x}, \boldsymbol{\vartheta}_k) \coloneqq \frac{\hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)}{1 - \hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)} \approx \frac{p(\boldsymbol{x}, \boldsymbol{\vartheta}_k)}{p(\boldsymbol{x}) p(\boldsymbol{\vartheta}_k)} = r_k(\boldsymbol{x}, \boldsymbol{\vartheta}_k) \;. \tag{3}$$

We train each ratio estimator $\hat{r}_k(\boldsymbol{x}, \boldsymbol{\vartheta})$ using Adam [37] to minimize the binary cross-entropy (BCE)

$$\ell_k = -\int \left[p(\boldsymbol{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \ln \hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k) + p(\boldsymbol{x}) p(\boldsymbol{\theta}) \ln \left(1 - \hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)\right) \right] d\boldsymbol{x}\, d\boldsymbol{\theta} \;. \tag{4}$$

In practice, we concatenate $\boldsymbol{x}$ with $\boldsymbol{\vartheta}_k$ as the input to $\hat{\rho}_{k,\phi}$. Since each classifier trains independently, it is trivial to train them all in parallel using the same underlying $(\boldsymbol{x}, \boldsymbol{\theta})$ pairs.

Practically, we parameterize the classifier by $\hat{\rho}_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k) = \sigma \circ f_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)$, where σ is the logistic sigmoid and $f_{k,\phi}$ is a neural network. The connection in Eq. (3) between the estimated ratio and the classifier implies that $\log \hat{r}_k(\boldsymbol{x}, \boldsymbol{\vartheta}_k) = f_{k,\phi}(\boldsymbol{x}, \boldsymbol{\vartheta}_k)$. We call the above technique MNRE.

When training data is limited, we found empirically (see Sec. 3.2 below) that the MNRE approach typically leads to conservative (i.e., not overconfident) likelihood-to-evidence ratio estimates, provided early stopping criteria are used to avoid over-fitting of the classifier. At its core MNRE solves a simple, supervised binary classification task rather than a complex, unsupervised density estimation problem. Classification tasks are generally easier to train [9], and can rely on battle-tested network architectures.

2.2 Truncated Marginal Neural Ratio Estimation (TMNRE)

MNRE and NRE estimate a (marginal) likelihood-to-evidence ratio agnostic to the observed data $\boldsymbol{x}$ or parameter $\boldsymbol{\theta}$, a so-called *amortized* estimate. In other words, MNRE is suitable when $\boldsymbol{x} \in \{\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{z}) \mid \boldsymbol{\theta} \in \Omega, \forall \boldsymbol{z}\}$ where Ω is the support of the prior. We propose an extension of this algorithm that enables targeted simulation of parameters relevant to a given target observation $\boldsymbol{x}_o$, and locally amortizes posteriors such that it enables empirical tests of the inference results. *Local amortization* implies that our proposed method is suitable when $\boldsymbol{x} \in \{\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{z}) \mid \boldsymbol{\theta} \in \Gamma, \forall \boldsymbol{z}\}$ where the parameter region $\Gamma \subset \Omega$ is a function of $\boldsymbol{x}_o$ and will be defined below.

We observe that values of $\boldsymbol{\theta}$ which could not have plausibly generated $\boldsymbol{x}_o$ evaluate to negligible posterior density, i.e. $p(\boldsymbol{\theta} \mid \boldsymbol{x}_o) \approx 0$, which suggests that the corresponding parameters $\boldsymbol{\theta}$ do not significantly contribute to the marginalization in Eq. (1). We denote a prior that is suitably constrained to parameters with non-negligible posterior density $p(\boldsymbol{\theta} \mid \boldsymbol{x}_o)$ by

$$p_\Gamma(\boldsymbol{\theta}) \coloneqq V^{-1} \mathbb{1}_\Gamma(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \ , \tag{5}$$

where $\mathbb{1}_\Gamma(\boldsymbol{\theta})$ is an indicator function that is unity on $\Gamma \subset \Omega$ and zero otherwise, and V^{-1} is a normalizing constant (which can be interpreted as the fractional volume of the truncated prior). The subscript $\square_\Gamma$ denotes quantities $\square$ based on a prior truncated by indicator function $\mathbb{1}_\Gamma$.

We define a rectangular indicator function $\mathbb{1}_{\Gamma^{\text{rec}}}$ by discarding parameters that lie in the far tails of the one dimensional marginal posteriors of our target observation $\boldsymbol{x}_o$, using a thresholding $\epsilon \ll 1$, via

$$\Gamma^{\text{rec}} = \left\{ \boldsymbol{\theta} \in \Omega \;\middle|\; \forall d = 1, \ldots, D : \frac{p(\theta_d \mid \boldsymbol{x}_o)}{\max_{\theta_d} p(\theta_d \mid \boldsymbol{x}_o)} > \epsilon \right\} \ . \tag{6}$$

For Gaussian joint posteriors, this scheme leads to one dimensional marginal posteriors $p_\Gamma(\theta_d \mid \boldsymbol{x}_o)$ that are truncated at their approximately $\pm\sqrt{-2 \ln \epsilon}\sigma$ tail. In general, truncation will lead to an approximation error that can be estimated as $p_{\Gamma^{\text{rec}}}(\boldsymbol{\theta} \mid \boldsymbol{x}_o) = p(\boldsymbol{\theta} \mid \boldsymbol{x}_o) + \mathcal{O}(\epsilon) \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \boldsymbol{x}_o)$, see Appendix C. Throughout this paper, we use $\epsilon = 10^{-6}$, which corresponds to $\pm 5.26\sigma$ for a Gaussian posterior. Those truncations do not affect the location of high-probability credible contours and have hence no practical effect on parameter inference tasks. We provide more exemplary error estimates for a range of cases in Appendix C.

Our algorithm defines a series of nested indicator functions $\mathbb{1}_{\Gamma^{(m)}}$ whose regions have the property

$$\Omega \coloneqq \Gamma^{(1)} \supset \Gamma^{(2)} \supset \cdots \supset \Gamma^{(M)} \supset \Gamma^{\text{rec}}. \tag{7}$$

They iteratively approximate the indicator function $\mathbb{1}_{\Gamma^{\text{rec}}}$ in multiple rounds $m = 1, \ldots, M$. This sequence is generated with the following steps:

- We initialize $\Gamma^{(1)} = \Omega$, meaning that we start with the unconstrained prior.
- Each round $1 \leq m \leq M$, we train D, one dimensional ratio estimators $\hat{r}_{d,\Gamma^{(m)}}(\boldsymbol{x} \mid \theta_d)$ using data from within the constrained region, $\boldsymbol{\theta} \in \Gamma^{(m)}$. The estimated marginal posterior is $\hat{p}_{\Gamma^{(m)}}(\theta_d \mid \boldsymbol{x}) = \hat{r}_{d,\Gamma^{(m)}}(\boldsymbol{x} \mid \theta_d) p_{\Gamma^{(m)}}(\theta_d)$. To this end, do MNRE, setting $\boldsymbol{\vartheta}_k = \theta_k, \ d \in \{1, 2, \ldots, D\}$ using the constrained prior $p_{\Gamma^{(m)}}(\boldsymbol{\theta})$ with $N^{(m)}$ training samples per round.
- For each round $m < M$, we estimate the indicator function for the next round using the approximated posteriors, via

$$\Gamma^{(m+1)} = \left\{ \boldsymbol{\theta} \in \Gamma^{(m)} \;\middle|\; \forall d : \frac{\hat{p}_{\Gamma^{(m)}}(\theta_d \mid \boldsymbol{x}_o)}{\max_{\theta_d} \hat{p}_{\Gamma^{(m)}}(\theta_d \mid \boldsymbol{x}_o)} > \epsilon \right\} \ . \tag{8}$$

Algorithm 1 Truncated Marginal Neural Ratio Estimation (TMNRE)

Inputs: Simulator $p(\boldsymbol{x} \mid \boldsymbol{\theta})$, factorizable prior $p(\boldsymbol{\theta})$, real observation $\boldsymbol{x}_0$, max rounds M, training data per round $N^{(m)}$, threshold ϵ, dimension of parameters D, mass ratio β, classifiers $\boldsymbol{\rho}_1(\boldsymbol{x}, \boldsymbol{\theta}) = \{\sigma \circ f_{\phi,d}(\boldsymbol{x}, \theta_d)\}_{d=1}^{D}$ and $\boldsymbol{\rho}_2(\boldsymbol{x}, \boldsymbol{\theta}) = \{\sigma \circ f_{\phi,d}(\boldsymbol{x}, \boldsymbol{\vartheta}_d)\}_{d=(1,1)}^{(D,D)}$.

Outputs: Parameterized classifiers $\boldsymbol{\rho}_1(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{\rho}_2(\boldsymbol{x}, \boldsymbol{\theta})$, constrained region Γ^{rec}.

procedure MNRE($\mathcal{D}$, $\boldsymbol{\theta}'$, $\boldsymbol{\rho}_\phi$)
  while $\boldsymbol{\rho}_\phi$ not converged **do**
    $\phi \leftarrow$ OPTIMIZER $\left(\phi, \nabla_\phi \sum_k \left[\text{BCE}(\hat{\rho}_{\phi,k}(\boldsymbol{x}, \boldsymbol{\vartheta}_k), 1) + \text{BCE}(\hat{\rho}_{\phi,k}(\boldsymbol{x}, \boldsymbol{\vartheta}'_k), 0)\right]\right)$
  return $\mathbf{f}_\phi$

Initialize: $\mathcal{D}^{(0)} \leftarrow \{\}$, $\Gamma^{(0)} \leftarrow \text{supp}(p(\boldsymbol{\theta}))$, $\alpha^{(0)} \leftarrow 0$, $m \leftarrow 1$.

procedure TMNRE
  while $\alpha^{(m-1)} \leq \beta$ and $m \leq M$ **do**
    $\mathcal{D}_\Gamma^{(m-1)} \leftarrow \left\{(\boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(n)}) \in \mathcal{D}^{(m-1)} \;\middle|\; \boldsymbol{\theta}^{(n)} \in \Gamma^{(m-1)}\right\}$ ▷ *Retain data in region*
    $N_{\text{simulate}}^{(m)} \leftarrow N^{(m)} - |\mathcal{D}_\Gamma^{(m-1)}|$ ▷ *Calculate num. necessary simulations*
    $\boldsymbol{\theta} \leftarrow \{\boldsymbol{\theta}^{(n)} \sim \mathbb{1}_{\Gamma^{(m-1)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})\}_{n=1}^{N_{\text{simulate}}^{(m)}}$ ▷ *Sample for jointly distributed pairs*
    $\boldsymbol{x} \leftarrow \{\boldsymbol{x}^{(n)} \sim p(\boldsymbol{x} \mid \boldsymbol{\theta}^{(n)})\}_{n=1}^{N_{\text{simulate}}^{(m)}}$ ▷ *Simulate jointly distributed pairs*
    $\boldsymbol{\theta}' \leftarrow \{\boldsymbol{\theta}^{(n)} \sim \mathbb{1}_{\Gamma^{(m-1)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})\}_{n=1}^{N^{(m)}}$ ▷ *Sample for marginally distributed pairs*
    $\mathcal{D}^{(m)} \leftarrow \mathcal{D}_\Gamma^{(m-1)} \cup \{(\boldsymbol{x}^{(n)}, \boldsymbol{\theta}^{(n)})\}_{n=1}^{N_{\text{simulate}}^{(m)}}$ ▷ *Aggregate training data*
    $\boldsymbol{\rho}_1 \leftarrow$ MNRE($\mathcal{D}^{(m)}$, $\boldsymbol{\theta}'$, $\boldsymbol{\rho}_1$)
    $\Gamma^{(m)} \leftarrow \left\{\boldsymbol{\theta} \in \Gamma^{(m-1)} \;\middle|\; \forall d : \frac{\hat{p}_{d,\Gamma^{(m)}}(\theta_d|\boldsymbol{x}_o)}{\max_{\theta_d} \hat{p}_{d,\Gamma^{(m)}}(\theta_d|\boldsymbol{x}_o)} > \epsilon\right\}$ ▷ *Find constrained region*
    $\alpha^{(m)} \leftarrow \int \mathbb{1}_{\Gamma^{(m)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} / \int \mathbb{1}_{\Gamma^{(m-1)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ ▷ *Update prior mass ratio*
    $m \leftarrow m + 1$ ▷ *Increment counter*
  $\boldsymbol{\rho}_2 \leftarrow$ MNRE($\mathcal{D}^{(m)}$, $\boldsymbol{\theta}'$, $\boldsymbol{\rho}_2$)
  return $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, $\Gamma^{(m)}$

- The last round is determined either when $m = M$ or when a stopping criterion is reached. The stopping criterion is defined by the ratio of consecutive truncated prior masses. It is satisfied when the sequence of truncated priors have the property $\int \mathbb{1}_{\Gamma^{(m)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} / \int \mathbb{1}_{\Gamma^{(m-1)}}(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} > \beta$. We often set $\beta = 0.8$.
- Using the data from this final constrained region, we train all one and higher-dimensional marginal posteriors of interest.

We briefly address failure modes. First, this algorithm relies on the assumption that posterior estimates $\hat{p}_{\Gamma^{(m)}}(\theta_d \mid \boldsymbol{x}_o)$ from MNRE provide a good approximation of $p(\theta_d \mid \boldsymbol{x}_o)$. An over-confident estimate would remove parameter ranges that are part of Γ^{rec}. In practice, we have not observed this effect. We give credit to early stopping and a conservative choice of ϵ, and provide further illustration and support in Sec. 3 below. Second, since the truncated posterior only agrees with the ground truth up to corrections of order ϵ, the iterative scheme will not converge to Eq. (6); rather to a similar expression where the right-hand side of the inequality in Eq. (6) receives additional $\mathcal{O}(\epsilon)$ corrections. Although these corrections mildly affect the truncations, they are of little practical relevance since we choose an ϵ which is very small. Both failure modes are diagnosed by checking whether high probability regions of the estimated posteriors intersect with the boundaries of the indicator function.

Like sequential methods [10, 14] the number of rounds M, the training data per round $N^{(m)}$, and any stopping criteria β are hyperparameters. For further discussion and default values see Appendix A, for bound derivations and limitations see Appendix C and D. We present TMNRE in Algorithm 1.

Properties of our algorithm. We discuss the properties of our algorithm in support of Table 1. First, our algorithm performs *targeted inference* by successively focusing on regions of the parameter space that are compatible with an observation $\boldsymbol{x}_o$. Second, since training data is always drawn from the prior, it is possible to efficiently *train arbitrary marginal posteriors* with the same training data generated for round M. Third, the algorithm trains *locally amortized posteriors* that are valid for

parameters $\boldsymbol{\theta} \in \Gamma^{(m)}$, facilitating empirical consistency checks of the estimated posteriors within this region. All these aspects will be demonstrated in the experiments in the following section.

3 Experiments

First, we perform experiments to compare TMNRE to other algorithms on standard benchmarks from the simulator-based inference literature. Next, we highlight useful aspects of our algorithm regarding targeted inference, marginalization and local amortization with two additional experiments. Further experiments, including application on a cosmology simulator, can be found in Appendix E.

3.1 Performance on standard tasks

We compare the performance of our algorithm with other traditional and neural simulation-based inference methods on a selection of problems from the SBI benchmark [38]. Each *task* is defined by a simulator, ten observations, a simulation budget, and 10,000 samples drawn from corresponding reference and approximate posterior distributions. The reference samples enable quantification of algorithmic accuracy on a range of performance metrics. We evaluate performance on each of the following tasks: Gaussian Linear Uniform, SLCP, SLCP Distractors, Gaussian Mixture, and Two Moons. Each has a uniform prior. Details in Appendix A.

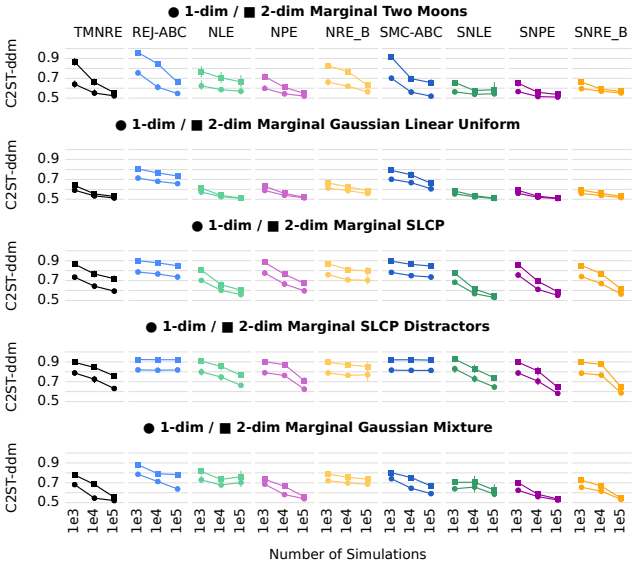

Figure 1: **Performance on marginalized posterior benchmark tasks.** Classification accuracy (C2ST-ddm) of our method, TMNRE, and REJ-ABC, NLE, NPE, NRE_B, SMC-ABC, SNLE, SNPE, and SNRE_B for 10 observations each, means and 95% confidence intervals. One and two dimensional scores are plotted. Lower scores imply better posteriors. Our simulation budget is approximate but within $\sim 5\%$ of reported value. The plot and tasks are derivative of [38].

Since our method estimates every one and two dimensional marginal posterior, we compare samples from our approximate marginal posteriors with samples from the reference joint posterior, marginalized over nuisance parameters. We quantify the results using the Classifier 2-Sample Test [39, 40]. We train a C2ST classifier for each of the $\binom{D}{1} + \binom{D}{2}$ possible one and two dimensional marginals, and report the mean values, and 95% confidence intervals, in Figure 1. We call this averaged performance metric *C2ST-ddm*, see Appendix B for more detail. The results are presented as grouped by dimensionality since learning difficulty increases with dimension and we expect this to be reflected in the C2ST-ddm scores.

For comparison, we computed the C2ST-ddm on the other benchmark methods' posterior samples. Unlike our method, which was trained on the marginals directly, the benchmark methods were trained to estimate the joint posterior. We note that since TMNRE trains a neural network for every marginal (efficiently, in parallel), our method has many times more parameters than any neural likelihood-free inference method that directly targets the joint. However, parameter count is not usually a scarce resource in this setting. Training hyperparameters can be found in Appendix A.

As shown in Figure 1, our method outperformed both REJ-ABC and SMC-ABC on all tasks. The maximum number of rounds before meeting the stopping criteria was four for Two Moons, Gaussian Linear Uniform, and Gaussian Mixture. TMNRE was competitive with sequential methods at high simulation budget on these tasks, while it performed only slightly better than non-sequential methods on the SLCP and SLCP Distractors. TMNRE stopped after only two and one round(s) on SLCP

and SLCP Distractors. Doing marginal inference directly offers increased efficiency compared to non-sequential methods across tasks. On the tasks with narrow posteriors, it offered comparable accuracy with sequential methods given large simulation budget, The benefit was diminished on tasks with wide posteriors like SLCP and SLCP Distractors–a limitation of the method. In general, TMNRE provides sufficient accuracy and local amortization. Neither sequential nor non-sequential methods offer both.

3.2 Efficient targeted inference: a 3-dim torus model

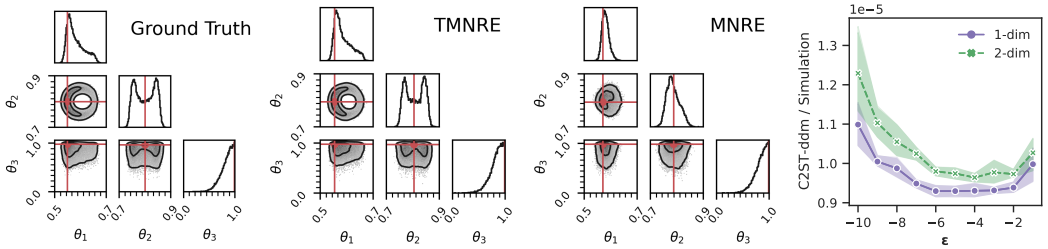

Figure 2: *First through Third Panel:* View of torus marginal posteriors as estimated by rejection sampling, TMNRE, and MNRE. In some dimensions, the posterior extends the full unit cube prior width, while in others it is very narrow. TMNRE easily finds the asymmetric details after constraining to the relevant region while MNRE does not. *Fourth Panel:* The results of a hyperparameter scan of ϵ on the torus task. The C2ST-ddm per simulation is reported versus ϵ. The mean and 95% confidence intervals are shown over five repetitions of the experiment. Lower values indicate better performance.

We define a task which highlights the effectiveness of truncating the prior, namely a simulator with a very small torus shaped posterior. We present an ablation study of the truncation method along with a hyperparameter scan of ϵ. The task's details are enumerated in Appendix A.

We ran Algorithm 1 which satisfied the stopping criterion after four rounds. We performed marginal likelihood-to-evidence ratio estimation on all one and two dimensional marginals for each step in the sequence of constrained regions, using the number of samples available that round. We also trained an estimator which used the same simulation budget but the samples were drawn from the unconstrained prior. We analyzed the prior volume, C2ST-ddm, and the sum of one dimensional KL divergences at each round for both methods. The posteriors are shown in Figure 2 and the performance metrics for the ablation study, are shown in Figure 3.

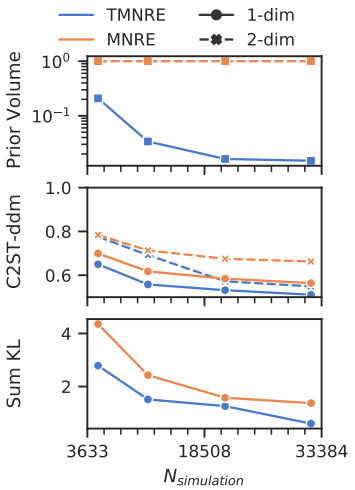

Figure 3: Performance metrics on MNRE and TMNRE versus simulation budget. Budgets determined by truncation algorithm. *T:* Prior volume. *M:* According to C2ST-ddm, TMNRE produces more accurate posteriors. *L:* KL divergence summed over 1-dim marginals; same result.

We found TMNRE very accurately approximated all marginals at the maximum simulation budget. MNRE placed mass in the correct region but missed the shape of the posterior entirely. TMNRE improved simulation efficiency compared with MNRE as indicated by the slope of the C2ST-ddm. The max-normalized posterior estimates at every round are plotted in Figure 4. We note that given the limited training data in early rounds, our method predicts wider posteriors than the ground truth. These are called conservative posterior estimates and they are the preferred failure mode for practitioners.

To determine the effects the hyperparameter ϵ, we performed a grid search between 10^{-10} to 10^{-1} using TMNRE on the same simulator. We reported the performance in terms of the C2ST-ddm per simulation in Figure 2 and repeated the experiment five times. We observe that the optimal value of ϵ was 10^{-6} since it was the most conservative value of ϵ that optimized the metric.

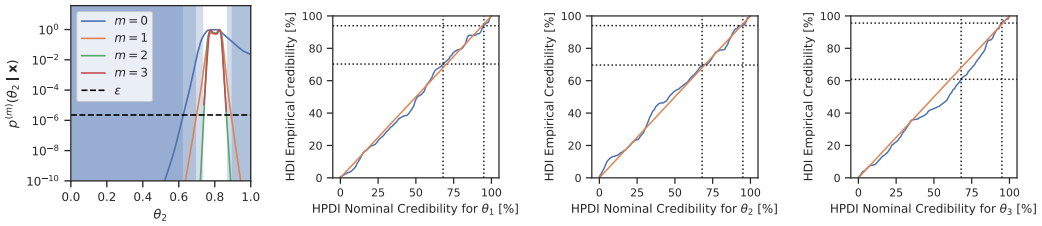

Figure 4: *First panel:* The ratio estimator for θ_2 in the 3-dim torus example, for several consecutive rounds. The estimator is conservative with limited data because it converges from above against the final results (red line). *Second to Fourth Panels:* Empirical versus nominal credibility for the highest posterior density intervals (HPDI) for each θ_d in the 3-dim torus.

3.3 Empirical tests of inference results through local amortization

Our algorithm locally amortizes the posterior for parameters drawn from the constrained prior $p_\Gamma(\boldsymbol{\theta})$. This opens the door for various experimental diagnostics to test the reliability of our trained inference networks with simulated data (which is also possible for NRE [41], but not for sequential methods that are exclusively targeting on one specific observation rather than a range of observations). We demonstrate this by comparing the empirical credibility to the nominal credibility for the highest posterior density intervals.

For the 3-dim torus example, we draw 10000 samples $(\boldsymbol{x}, \theta_d) \sim p(\boldsymbol{x} \mid \boldsymbol{\theta}) p_\Gamma(\boldsymbol{\theta})$ from the constrained generative model. For all samples, we generate marginal posteriors, $\hat{p}(\theta_d \mid \boldsymbol{x})$. For those marginal posteriors, we then derive the frequency with which $t\%$ highest density intervals contain the true value θ_d. The result is shown in Fig. 4. It provides an immediate check of the reliability of our trained inference networks *without knowing the ground truth*, and provides a safeguard against overconfident statements, which is critical for using the results of inference networks in a scientific context.

3.4 Efficient marginal posteriors: a 10-dim egg-box model

We define a posterior that, when plotted in two dimensions, looks like a top-down view of an 2×2 egg-box. Let $\boldsymbol{\theta}, \boldsymbol{g}(\boldsymbol{\theta}) \in \mathbb{R}^D$ and θ_k denote the kth element of $\boldsymbol{\theta}$, then the simulator for this problem is defined $g_k(\boldsymbol{\theta}) = \sin(\theta_k \cdot \pi)$. To fix the posterior shape, we set $\theta_{k,o} = \frac{1}{4}$, $k = 1, 2, \ldots, D$ and $\boldsymbol{x}_0 = \boldsymbol{g}(\boldsymbol{\theta}_o)$. The likelihood is determined by an additive noise model $p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{g}(\boldsymbol{\theta}), \sigma^2 \boldsymbol{I})$ with $\sigma = 0.1$. The total number of modes in our 10-dimensional model is $2^{10} = 1024$. Realistic models do not typically feature such a regular mode structure, but this pattern enables an analytic estimate of ground truth posteriors for comparison with the various algorithms.

Given 10,000 training samples drawn from the prior and a $D = 10$ dimensional parameter space, we trained MNRE to estimate all one and two dimensional marginals, the SBI [42] implementation of NRE and SNRE on the joint, and finally a marginalized version of SNRE (SMNRE). In SMNRE, we divided the samples across 10 rounds and each round proposed samples according to the previous round's posterior distribution for the predicted marginals, but the initial prior for the nuisance parameters. Since, in a general setting, SMNRE cannot use samples from another marginal estimator, we divided the 10,000 training samples evenly among the 55, one and two dimensional marginal estimators, each estimator receiving 181 training samples. 25,000 samples from each reported posterior are visible in Figure 5. Our method was the only one which recovered the structure of the ground truth marginal posteriors, providing empirical evidence that estimating marginals directly can provide high accuracy at low simulation budgets for complex high-dimensional posteriors.

4 Discussion and conclusions

We presented Truncated Marginal Neural Ratio Estimation (TMNRE), a simulation-based inference algorithm based on NRE. The core idea of our algorithm is to focus on most probable parameter regions by truncating marginal posteriors in their very low-probability tails. For Gaussian posteriors this is typically beyond 5σ and does not significantly affect the higher density contours. In addition to performing on par or better than existing algorithms on standard benchmarks, TMNRE is better

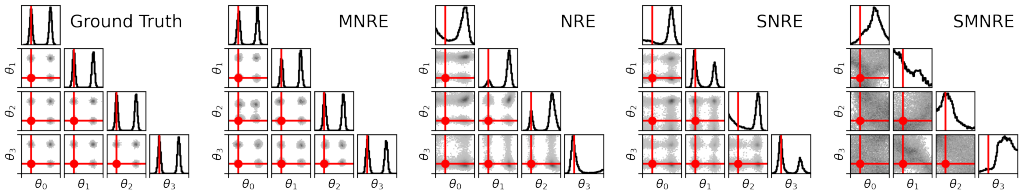

Figure 5: Posteriors from the 10-dim eggbox benchmark (only 4 parameters are shown for clarity). All methods received 10k training samples and produced 25k posterior samples. NRE and SNRE were trained jointly, while MNRE and SMNRE were trained marginally. SMNRE cannot share training samples between marginals so each estimator received an equal share of the total simulation budget, 181 samples. Our method is the only one which recovered the structure of the ground truth.

suited to the practitioner's needs than other algorithms because it offers simulation efficient marginal posterior estimation and the capacity to perform efficient consistency checks through local amortization. These features are particularly desirable to scientists whose simulators are expensive and rife with nuisance parameters.

TMNRE uses a sequence of training and sampling rounds to automatically produce parameters with high posterior density, i.e. relevant to a particular observation $\boldsymbol{x}_o$. The output of this sequence is a hyperrectangular approximation to a highest posterior density region, implicitly defined by hyperparameter ϵ. That means simulations from within this constrained region are likely informative while simulations from outside the region are likely uninformative. Using data drawn from this constrained region, TMNRE estimates any marginal posterior of interest directly using a marginal likelihood-to-evidence ratio; a simpler and more practical technique than estimating the entire joint posterior. Finally, by construction, our targeted inference method can accurately estimate posteriors of simulations from within the constrained parameter region. This freedom facilitates fast empirical studies of the nominal credible regions, which are of critical importance in real-life applications when there is no ground truth posterior to compare to.

On the SBI benchmark [38], we found that TMNRE is on par with the most effective SBI algorithms, such as SNRE [8], as measured by the C2ST performance metric, Fig. 1. We highlighted the benefits of TMNRE using two showcase tasks: a torus-shaped posterior and an eggbox-shaped posterior. The torus featured a very narrow posterior that TMNRE found and accurately learned while simple MNRE failed to do so, Fig. 3. We demonstrated validity of our iterative procedure, and the ability to perform important validation tests by testing the nominal credible intervals empirically, Fig. 4. The eggbox's joint posterior featured 1024 modes. MNRE efficiently estimated the marginal posteriors whereas other algorithms which estimate the joint posterior on the same task generally showed low performance, Fig. 5.

The presented algorithm is aimed at marginal posterior inference, which is a typical goal for scientific applications, but does not allow, e.g., to evaluate the posterior predictive distribution which requires the joint posterior. Furthermore, our algorithm particularly shines for high-dimensional problems with complex and/or narrow posteriors, whereas we expect that simpler problems could benefit from some of the other available SBI method. We address further limitations of our method in Appendix D.

We note that the hyperrectangular indicator function, defined in Eq. (8), is not optimal if some of the parameters are strongly correlated. However, it can be straightforwardly extended to more complex shapes. The challenge is to efficiently define the boundaries of the indicator function and sample from within it, a problem tackled by effective nested sampling algorithms [43].

This work is primarily foundational and the societal impacts, other than the cost of training machine learning models, would therefore be drawn from a hypothetical application. As this is an inference method, it would be possible to apply it to biased simulators which could reinforce unethical patterns. This could go unnoticed if the simulator was too complex to assess ethically. We urge caution when applying the method to problems which could reasonably affect living things. On the other hand, this work was inspired by necessity from astrophysical problems and is already being applied to simulators which produce general public engagement in science such as understanding gravitational waves and the history of the universe, see Appendix E. In general, the societal impacts are closely tied to the implications of the simulators themselves.

References

[1] Nilanjan Banik, Gianfranco Bertone, Jo Bovy, and Nassim Bozorgnia. Probing the nature of dark matter particles with stellar streams. *Journal of Cosmology and Astroparticle Physics*, 2018(07):061–061, Jul 2018.

[2] Justin Alsing, Tom Charnock, Stephen Feeney, and Benjamin Wandelt. Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, 488(3):4440–4458, 2019.

[3] Johann Brehmer, Kyle Cranmer, Gilles Louppe, and Juan Pavez. Constraining effective field theories with machine learning. *Physical review letters*, 121(11):111801, 2018.

[4] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092, June 1953.

[5] W K Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[6] John Skilling. Nested sampling for general bayesian computation. *Bayesian Anal.*, 1(4):833–859, December 2006.

[7] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[8] 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.

[9] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proc. Natl. Acad. Sci. U. S. A.*, May 2020.

[10] 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.

[11] 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.

[12] George Papamakarios and Iain Murray. Fast ϵ-free inference of simulation models with bayesian conditional density estimation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[13] Jan-Matthis Lueckmann, Pedro J Gonçalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1289–1299, 2017.

[14] 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.

[15] Gary J Feldman and Robert D Cousins. Unified approach to the classical statistical analysis of small signals. *Physical Review D*, 57(7):3873, 1998.

[16] Alex Geringer-Sameth, Savvas M Koushiappas, and Matthew G Walker. Comprehensive search for dark matter annihilation in dwarf galaxies. *Physical Review D*, 91(8):083535, 2015.

[17] Sara Algeri, Jan Conrad, and David A van Dyk. A method for comparing non-nested models with application to astrophysical searches for new physics. *Monthly Notices of the Royal Astronomical Society: Letters*, 458(1):L84–L88, 2016.

[18] F Feroz, M P Hobson, and M Bridges. MultiNest: an efficient and robust bayesian inference tool for cosmology and particle physics. *Mon. Not. Roy. Astron. Soc. 398: 1601-1614,2009*, September 2008.

[19] W J Handley, M P Hobson, and A N Lasenby. polychord : next-generation nested sampling. *Mon. Not. R. Astron. Soc.*, 453(4):4384–4398, September 2015.

[20] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.

[21] Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[22] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

[23] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P H Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface*, 6(31):187–202, February 2009.

[24] Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[25] Jan-Matthis Lueckmann, Giacomo Bassetto, Theofanis Karaletsos, and Jakob H Macke. Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. PMLR, 2019.

[26] Christopher M Bishop. Pattern recognition and machine learning (information science and statistics), 2006.

[27] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2335–2344, 2017.

[28] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[29] Niall Jeffrey and Benjamin D Wandelt. Solving high-dimensional parameter inference: marginal posterior densities & moment networks. *arXiv preprint arXiv:2011.05991*, 2020.

[30] Kyle Cranmer, Juan Pavez, and Gilles Louppe. Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv preprint arXiv:1506.02169*, 2015.

[31] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

[32] Michael U Gutmann, Jukka Corander, et al. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 2016.

[33] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, Michael U Gutmann, et al. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2016.

[34] Dustin Tran, Rajesh Ranganath, and David M Blei. Hierarchical implicit models and likelihood-free variational inference. *arXiv preprint arXiv:1702.08896*, 2017.

[35] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[36] Benjamin Kurt Miller, Alex Cole, Gilles Louppe, and Christoph Weniger. Simulation-efficient marginal posterior estimation with swyft: stop wasting your precious time. *arXiv preprint arXiv:2011.13951*, 2020.

[37] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR: International Conference on Learning Representations*, pages 1–15, 2015.

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

[39] Jerome H Friedman. On multivariate goodness–of–fit and two–sample testing. *STATISTICAL PROBLEMS IN PARTICLE PHYSICS, ASTROPHYSICS AND COSMOLOGY*, page 311, 2003.

[40] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

[41] Joeri Hermans, Nilanjan Banik, Christoph Weniger, Gianfranco Bertone, and Gilles Louppe. Towards constraining warm dark matter with stellar streams through neural simulation-based inference. *arXiv preprint arXiv:2011.14923*, 2020.

[42] 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.

[43] Joshua S Speagle. dynesty: A dynamic nested sampling package for estimating bayesian posteriors and evidences. April 2019.

[44] A F M Smith and A E Gelfand. Bayesian statistics without tears: A Sampling-Resampling perspective. *Am. Stat.*, 46(2):84–88, 1992.

[45] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[46] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[47] Aaditya Ramdas, Sashank Jakkam Reddi, Barnabás Póczos, Aarti Singh, and Larry Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[48] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[49] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[50] Thejs Brinckmann and Julien Lesgourgues. MontePython 3: boosted MCMC sampler and other features. *Phys. Dark Univ.*, 24:100260, 2019.

[51] Eleonora Di Valentino et al. Exploring cosmic origins with CORE: Cosmological parameters. *JCAP*, 04:017, 2018.

[52] Laurence Perotto, Julien Lesgourgues, Steen Hannestad, Huitzu Tu, and Yvonne Y. Y. Wong. Probing cosmological parameters with the CMB: Forecasts from full Monte Carlo simulations. *JCAP*, 10:013, 2006.

[53] Jonas Degrave and Ira Korshunova. Why machine learning algorithms are hard to tune and how to fix it. `https://www.engraved.blog/why-machine-learning-algorithms-are-hard-to-tune/`, Jan 2021.

[54] John Platt and Alan Barr. Constrained differential optimization. In D Anderson, editor, *Neural Information Processing Systems*. American Institute of Physics, 1988.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] In the introduction / contributions, we highlight three aspirations of a useful SBI method and claim that we achieve those goals. We define the technique in Sections 2.1 and 2.2, clarified with Algorithm 1. Finally, we highlight those specific aspects in the experiments from Sections 3.2-3.4.
   (b) Did you describe the limitations of your work? [Yes] We addressed some limitations during the exposition of the method throughout Section 2, on the benchmark problem in Section 3.1, and in the conclusion Section 4. We were specific in Section D.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] This is addressed at the end of the conclusion. It is primarily related to a speculative application.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the guidelines. We have considered potential negative societal impacts and believe our work is unlikely to directly produce negative societal impact beyond the costs of training machine learning models. We believe we have complied with the general ethical conduct guidelines in our practices and because our data involves no human subjects.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Theoretical contributions are limited in our paper. We described the assumptions about the prior in Section 2 and the method's relevance in Section 2.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] The claim that the effects of our method are of $\mathcal{O}(\epsilon)$ is derived step by step in Section C. Convergence of likelihood-to-ratio estimation was already proven in [8].
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code will be provided along with the supplemental material. In the event of publication, it will be made available online.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Some details are written within the experiments section 3, remaining available details are written within the corresponding subsection in Section A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figures 1 and 2. However, not all results contain multiple runs, namely the eggbox and the performance metrics in Figure 3 only report a single run.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Discussed in Section A including a total CO_2 estimate.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Figure 1. Mention of other general-propose software will go into the acknowledgements.
   (b) Did you mention the license of the assets? [Yes] See Section A.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We distribute posterior samples from our trained models and other training diagnostic information. This will be included in the code for reproducibility purposes.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We only used simulation data from a public, online benchmark distributed with an MIT License. The benchmark was intended to be used in precisely the manner in which we use it.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]