Is Learning Summary Statistics Necessary for Likelihood-free Inference?

Is Learning Summary Statistics Necessary for Likelihood-free Inference?

Yanzhi Chen [1] **Michael U. Gutmann** [2] **Adrian Weller** [1]

Abstract

Likelihood-free inference (LFI) is a set of techniques for inference in implicit statistical models. A longstanding question in LFI has been how to design or learn good summary statistics of data, but this might now seem unnecessary due to the advent of recent end-to-end (i.e. neural network-based) LFI methods. In this work, we rethink this question with a new method for learning summary statistics. We show that learning sufficient statistics may be easier than direct posterior inference, as the former problem can be reduced to a set of low-dimensional, easy-to-solve learning problems. This suggests us to explicitly decouple summary statistics learning from posterior inference in LFI. Experiments on diverse inference tasks with different data types validate our hypothesis.

1. Introduction

Many data generating processes in science and engineering can be well described by a parametric statistical model that allows forward simulation: $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$ but does not admit a tractable likelihood $p(\mathbf{x}|\boldsymbol{\theta})$. These models are called *implicit models* (Diggle & Gratton, 1984) and have applications as diverse as physics (Sjöstrand et al., 2008), genetics (Järvenpää et al., 2018), computer graphics (Mansinghka et al., 2013), robotics (Lopez-Guevara et al., 2017), finance (Bansal & Yaron, 2004), economics (Dyer et al., 2022), cosmology (Weyant et al., 2013; Alsing et al., 2018), ecology (Wood, 2010) and epidemiology (Chinazzi et al., 2020).

An important question is how to perform Bayesian inference in implicit models. *Likelihood-free inference* (LFI) techniques facilitate inference in such circumstances. LFI does not need to evaluate the likelihood function. Rather, it only requires us to sample (i.e. simulate) data from the model. Traditional methods such as approximate Bayesian computation (ABC) work by repeatedly simulating data from the model, and use a small subset of the simulated data closest to the observed data to construct the posterior (Pritchard et al., 1999; Marjoram et al., 2003; Beaumont et al., 2009). Recent advances make use of flexible neural networks to approximate the intractable likelihood (Papamakarios et al., 2019), the density ratio (Hermans et al., 2020; Durkan et al., 2020) or directly the posterior (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019).

A historically important ingredient in likelihood-free inference was the design/choice of suitable summary statistics, which was believed to be essential (Blum et al., 2013; Fearnhead & Prangle, 2012; Sisson et al., 2018; Papamakarios et al., 2019). This motivated the development of many approaches aiming at the automatic design of summary statistics (Fearnhead & Prangle, 2012; Chan et al., 2018; Alsing et al., 2018; Wiqvist et al., 2019; Brehmer et al., 2020; Chen et al., 2021; Dyer et al., 2021; Pacchiardi & Dutta, 2022). The necessity of summary statistics was, however, recently challenged due to the advent of end-to-end LFI methods (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019; Hermans et al., 2020). In these methods, the input data $\mathbf{x}$ can be directly passed to an encoder trained jointly with the posterior estimator, so that the summary statistics is learned implicitly. In this sense, explicit summary statistics learning seems no more necessary.

In this work, we rethink whether it is necessary or not to learn summary statistics explicitly in likelihood-free inference. We address this question from two perspectives. First, we show that the sufficient statistics of an implicit model may indeed be easier to learn than its likelihood or posterior. Second, we show that recent end-to-end inference approaches in LFI (e.g. SNPE-C (Greenberg et al., 2019), SNR (Hermans et al., 2020)) can be unreliable in some cases. Both discoveries suggest that explicit (i.e. separate) learning of summary statistics still has a market. We highlight the following contributions:

- We propose a new method, SSS, for learning sufficient statistics, where neither exact estimation of the posterior $p(\boldsymbol{\theta}|\mathbf{x})$ nor of the mutual information $I(\mathbf{x};\boldsymbol{\theta})$ is needed;
- Based on our method, we develop a new LFI algorithm, SNL + SSS, which is shown to outperform state-of-the-art end-to-end inference algorithms (e.g. SNPE-C, SNR).

[1]Department of Engineering, Cambridge University, UK [2]School of Informatics, The University of Edinburgh, UK. Correspondence to: Yanzhi Chen <yc514@cam.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

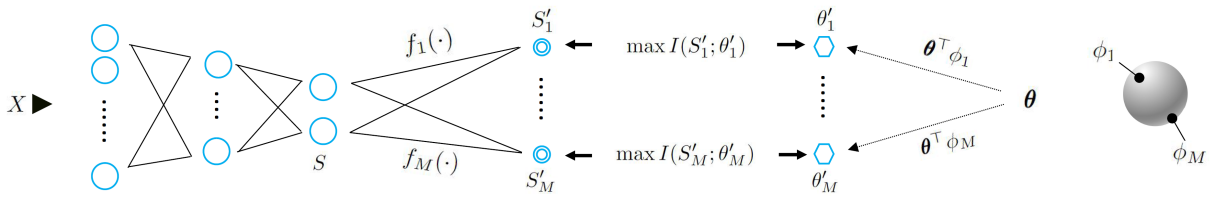


Figure 1. An overview of the proposed method for learning sufficient statistics $S = \arg\max_s I(s(X);\boldsymbol{\theta})$, $S \in \mathbb{R}^d$. Our method does not directly estimate the mutual information $I(s(X);\boldsymbol{\theta})$. Rather, it optimises a set of mutual informations $I(S'_i;\theta'_i), i \in \{1,2,...M\}$ between M pairs of *low-dimensional* random variables $\{(S'_i,\theta'_i)\}_{i=1}^M$. Here $\theta'_i \in \mathbb{R}$ is the 'sliced' version of $\boldsymbol{\theta}$ after the i-th slicing operation (dashed arrow in the figure). $S'_i \in \mathbb{R}^{d'}$ is the 'secondary' sufficient statistics tailored for θ'_i computed as $S'_i = f_i(S)$. Note that $d' \ll d$. Each slicing directions $\boldsymbol{\phi}_i$ is uniformly sampled from the surface of a unit hypersphere. Objects in black solid lines are to learn.

2. Background

Likelihood-free inference. LFI considers Bayesian inference for implicit statistical models (Diggle & Gratton, 1984) where the evaluation of the likelihood function of the model is intractable but sampling from the model is possible:

$$\pi(\boldsymbol{\theta}|\mathbf{x}^o) \propto \pi(\boldsymbol{\theta})\underbrace{p(\mathbf{x}^o|\boldsymbol{\theta})}_{?}, \tag{1}$$

where $\mathbf{x}^o$ is the observed data, $\pi(\boldsymbol{\theta})$ is the prior over the model parameters $\boldsymbol{\theta}$, $p(\mathbf{x}^o|\boldsymbol{\theta})$ is the intractable likelihood function and $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ is the posterior over $\boldsymbol{\theta}$. Despite that we do not have access to the exact likelihood, we assume that can still sample data from the model: $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$. The task is then to infer $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ given $\mathbf{x}^o$ and the sampled data: $\mathcal{D} = \{\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^n$, $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}), \mathbf{x}^{(i)} \sim p(\mathbf{x}|\boldsymbol{\theta}^{(i)})$. Note that the proposal $p(\boldsymbol{\theta})$ can be different from the prior $\pi(\boldsymbol{\theta})$.

Different (neural) LFI methods use different strategies to learn the posterior (1) from $\mathcal{D}$. For example, *sequential neural posterior estimate* (SNPE) (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019) learns the conditional distribution $p(\boldsymbol{\theta}|\mathbf{x})$ in $\mathcal{D}$, whereas *sequential neural likelihood* (SNL) (Papamakarios et al., 2019) learns the likelihood $p(\mathbf{x}|\boldsymbol{\theta})$, both using a neural density estimator (Papamakarios et al., 2017; Durkan et al., 2019). The posterior $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ can then be obtained from the learned $p(\boldsymbol{\theta}|\mathbf{x}^o)$ or $p(\mathbf{x}^o|\boldsymbol{\theta})$ in conjunction with the prior $\pi(\boldsymbol{\theta})$. Alternatively, one may learn the ratio $r(\mathbf{x},\boldsymbol{\theta}) = p(\mathbf{x},\boldsymbol{\theta})/p(\mathbf{x})p(\boldsymbol{\theta})$ by a network r. This leads to the *sequential neural ratio estimate* (SNR) method (Hermans et al., 2020; Thomas et al., 2022).

Many LFI methods work in a *sequential* regime where the posterior $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ is learned in multiple rounds. One option is to use active learning or Bayesian decision theory to guide the process (Gutmann & Corander, 2016; Järvenpää et al., 2019; 2021; Oliveira et al., 2021). Another option is to use the posterior estimated in the rth round as the proposal distribution for $\boldsymbol{\theta}$ in the next round: $p_{r+1}(\boldsymbol{\theta}) = \hat{\pi}_r(\boldsymbol{\theta}|\mathbf{x}^o)$. Such a strategy is often used in neural LFI (Lueckmann et al., 2017; Papamakarios et al., 2019; Greenberg et al., 2019; Lueckmann et al., 2021) as it allows one to quickly focus on the plausible region of the posterior, greatly accelerating inference. However, different LFI methods have different affinities to sequential learning. For example, likelihood and ratio learning approaches have been shown to be highly compatible with sequential learning, whereas posterior targeting methods e.g. (Papamakarios & Murray, 2016; Lueckmann et al., 2017) may have issues in sequential learning regime.

Summary statistics for LFI. It is well-known that if good statistics of the data are available, Bayesian inference can be done with the statistics $S(\mathbf{x})$ instead of the raw data $\mathbf{x}$:

$$\pi(\boldsymbol{\theta}|\mathbf{x}) \approx \pi(\boldsymbol{\theta}|S(\mathbf{x})) \propto \pi(\boldsymbol{\theta})p(S(\mathbf{x})|\boldsymbol{\theta}), \tag{2}$$

where $S : \mathcal{X} \to \mathcal{S}$ is a deterministic function. Inference based on sufficient statistics has many benefits, see e.g. the Rao-Blackwell theorem. However, it is often hard to find suitable statistics for an implicit model due to the intractable likelihood function. One way to find S without accessing the likelihood is to learn it from the data $\mathcal{D}$ by e.g. the infomax principle (Chen et al., 2021): $S = \arg\max_s I(s(\mathbf{x});\boldsymbol{\theta})$. However, infomax learning can be challenging if I is high as $\mathbb{V}[\hat{I}] = O(e^I)$ (Poole et al., 2019; Song & Ermon, 2019). On the other hand, there exist simpler methods to learn summary statistics without estimating information (Fearnhead & Prangle, 2012; Alsing et al., 2018; Brehmer et al., 2020), but they are generally insufficient.

A natural question to ask is whether we really need to learn summary statistics *explicitly* in LFI. For SNL, the answer is clearly yes, as modelling the low-dimensional distribution $p(S|\boldsymbol{\theta})$ is much easier than modelling $p(\mathbf{x}|\boldsymbol{\theta})$. For SNPE and SNR, however, this is believed to be unnecessary. This is because the neural networks in these methods can be made end-to-end for $\mathbf{x}$, so that they can automatically transform $\mathbf{x}$ to some low-dimensional representation. In this sense, the internal layers in SNPE and SNR learn the sufficient statistics *implicitly*. In fact, the method by Chen et al. (2021) for learning S can also be seen as a variant of SNR where the design of the ratio estimator is sufficient statistics aware, so there seems no need to learn sufficient statistics explicitly.

3. Sufficient statistics learning

In this section, we answer the previous question from a different perspective: we show that the learning of sufficient statistics may be easier than inference itself. More specifically, to learn sufficient statistics, we only need to solve low-dimensional classification or metric learning problems.

3.1. Slice sufficient statistics

The core of our method is Theorem 1, which is inspired by recent sliced techniques in machine learning and statistics (Goldfeld & Greenewald, 2021; Chen et al., 2022).

Theorem 1. *Let $\mathbf{x} \in \mathbb{R}^D$ and $\boldsymbol{\theta} \in \mathbb{R}^K$ be two random variables and $S : \mathbb{R}^D \to \mathbb{R}^d$ be a deterministic function. Then $S(\mathbf{x})$ is a sufficient statistics if and only if $S(\mathbf{x})$ maximises $SI(S(\mathbf{x}); \boldsymbol{\theta})$ as defined below:*

$$SI(S(\mathbf{x}); \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\phi} \sim \mathbb{S}^{K-1}}[I(S(\mathbf{x}); \boldsymbol{\phi}^\top \boldsymbol{\theta})], \tag{3}$$

where $\boldsymbol{\phi} \in \mathbb{S}^{K-1}$ is a vector uniformly sampled from the surface of a K-dimensional unit sphere $\mathbb{S}^{K-1}$.

Proof. See Appendix A. □

Theorem 1 is non-trivial, as by the data processing inequality we know that any deterministic function $F(\cdot)$ will lose information about $\boldsymbol{\theta}$, so maximising $I(S(\mathbf{x}); F(\boldsymbol{\theta}))$ with $F(\boldsymbol{\theta}) = \boldsymbol{\phi}^\top \boldsymbol{\theta}$ seems not enough to maximise $I(S(\mathbf{x}); \boldsymbol{\theta})$. However, Theorem 1 says if $I(S(\mathbf{x}); \boldsymbol{\phi}^\top \boldsymbol{\theta})$ is maximised *for all* $\boldsymbol{\phi} \in \mathbb{S}^{K-1}$, so is $I(S(\mathbf{x}); \boldsymbol{\theta})$.

A Monte Carlo estimate to the objective (3) is:

$$SI(S(\mathbf{x}); \boldsymbol{\theta}) \approx \frac{1}{M} \sum_{i=1}^{M} I(S(\mathbf{x}); \boldsymbol{\phi}_i^\top \boldsymbol{\theta}), \quad \boldsymbol{\phi}_i \sim \mathbb{S}^{K-1} \tag{4}$$

where each $I(S(\mathbf{x}); \boldsymbol{\phi}_i^\top \boldsymbol{\theta})$ can be expressed using the infomax principle

$$I(S(\mathbf{x}); \boldsymbol{\phi}_i^\top \boldsymbol{\theta}) = \sup_{f_i} I(\underbrace{f_i(S(\mathbf{x}))}_{S_i'}; \underbrace{\boldsymbol{\phi}_i^\top \boldsymbol{\theta}}_{\theta_i'}) \tag{5}$$

where $S_i' \in \mathbb{R}^{d'}$ is a 'secondary' sufficient statistic tailored for $\theta_i' \in \mathbb{R}$. The introduction of $f_i : \mathbb{R}^d \to \mathbb{R}^{d'}$ is motivated by the fact that the sufficient statistic S_i' is different for each θ_i', and their dimensionalities d' should satisfy $d' \ll d$. We note that estimating $I(S_i'; \theta_i')$ is easier than estimating $I(S(\mathbf{x}); \boldsymbol{\theta})$ as (a) it is a lower-dimensional mutual information estimation problem; (b) as each $I_i = I(S_i'; \theta_i') \leq I(S(\mathbf{x}), \boldsymbol{\theta}) = I_{\text{all}}$, we know $\mathbb{V}[\hat{I}_i] \ll \mathbb{V}[\hat{I}_{\text{all}}]$ (recall that $\mathbb{V}[\hat{I}] = O(e^I)$), so estimating $\hat{I}_i$ is more sample-efficient.

The problem now boils down to how to quantify $I(S_i', \theta_i')$. While any estimator can be used in principle, we focus on the following two non-KL proxies to mutual information, which are either more robust or faster to compute than KL:

Jensen-Shannon divergence proxy (JSD). This proxy corresponds to interpreting mutual information as the distributional discrepancy between $p(S_i', \theta_i')$ and $p(S_i')p(\theta_i')$ and replacing the KL divergence with *Jensen-Shannon divergence*, which was shown to be more robust (Hjelm et al., 2018). It is defined as follows:

$$\begin{aligned}\hat{I}(S_i', \theta_i') = \sup_{T_i : \mathbb{R} \times \mathbb{R}^{d'} \to \mathbb{R}} & \mathbb{E}_{p(\theta_i', S_i')}\left[-\operatorname{sp}(-T_i(\theta_i', S_i'))\right] \\ & -\mathbb{E}_{p(\theta_i')p(S_i')}\left[\operatorname{sp}(T_i(\theta_i', S_i'))\right]\end{aligned} \tag{6}$$

where $\operatorname{sp}(u) = \log(1 + \exp(u))$ is the softplus function and T_i is a deterministic function often known as the critic. This can be seen as training a classifier T_i to distinguish samples $(S_i', \theta_i') \sim p(S_i', \theta_i')$ v.s. samples $(S_i', \theta_i') \sim p(S_i')p(\theta_i')$.

Distance correlation proxy (dCorr). This proxy corresponds to interpreting mutual information as a dependency metric and replacing it with *distance correlation*, a non-parametric dependency metric that is fast to compute (Székely et al., 2014). It is defined as:

$$\hat{I}(S_i'; \theta_i') = \frac{\mathbb{E}_{p(\theta_i', S_i')p(\tilde{\theta}_i', \tilde{S}_i')}[h(\theta_i', \tilde{\theta}_i')h(S_i', \tilde{S}_i')]}{\sqrt{\mathbb{E}_{p(\theta_i')p(\tilde{\theta}_i')}[h^2(\theta_i', \tilde{\theta}_i')]\mathbb{E}_{p(S_i')p(\tilde{S}_i')}[h^2(S_i', \tilde{S}_i')]}} \tag{7}$$

where $h(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\| - \mathbb{E}_{p(\tilde{\mathbf{b}})}[\|\mathbf{a} - \tilde{\mathbf{b}}\|] - \mathbb{E}_{p(\tilde{\mathbf{a}})}[\|\tilde{\mathbf{a}} - \mathbf{b}\|] + \mathbb{E}_{p(\tilde{\mathbf{a}})p(\tilde{\mathbf{b}})}[\|\tilde{\mathbf{a}} - \tilde{\mathbf{b}}\|]$ is the doubly centred distance. This can be seen as learning S_i' whose pairwise distances highly correlate with the pairwise distances of θ_i'.

In terms of the parameterisation of the secondary encoders $\{f_1, ..., f_M\}$ and the critic networks $\{T_1, ..., T_M\}$, we use an *amortised* strategy, where only two networks $f : \mathbb{R}^d \times \mathbb{Z}^K \to \mathbb{R}^{d'}$ and $T : \mathbb{R}^{d'} \times \mathbb{R} \times \mathbb{Z}^K \to \mathbb{R}$ need to be trained:

$$f_i(S) = f(S, \boldsymbol{\phi}_i), \quad T_i(S_i', \theta_i') = T(S_i', \theta_i', \boldsymbol{\phi}_i).$$

With such strategy, learning now amounts to the training of three neural networks f, S, T. The whole sliced-based statistics learning procedure is summarised in Algorithm 1.

We further draw a connection between our slice-based approach and existing summary statistics learning methods:

- *Infomax statistics* (Chen et al., 2021). This corresponds to learning S by directly maximising the mutual information between $S(\mathbf{x})$ and $\boldsymbol{\theta}$:

  $$S = \arg\max_s I(s(\mathbf{x}); \boldsymbol{\theta}).$$

  As shown in Theorem 1, our method recovers the goal of this approach when the number of slices $M \to \infty$. However, unlike this approach, we need not estimate $I(S; \boldsymbol{\theta})$ in the original space explicitly, which can be difficult if $I(S; \boldsymbol{\theta})$ is high (Poole et al., 2019; Song & Ermon, 2019).

Algorithm 1 Slice sufficient statistics learning

Input: simulated data $\mathcal{D} = \{\boldsymbol{\theta}^{(j)}, \mathbf{x}^{(j)}\}_{j=1}^n$
Output: Statistics function $S = \arg\max_s I(s(\mathbf{x}); \boldsymbol{\theta})$
Parameters: encoders S, f, MI estimation network T
Hyperparams: number of slices M, learning rate η
while not converge **do**
  sample a minibatch $\mathcal{B} \subset \mathcal{D}$;
  for i in 1 to M **do**
    sample $\boldsymbol{\phi}_i \sim \mathcal{U}(\mathbb{S}^{K-1})$;
    compute $S_i' = f(S(\mathbf{x}), \boldsymbol{\phi}_i)$ and $\theta_i' = \boldsymbol{\phi}_i^\top \boldsymbol{\theta}$;
    compute $\hat{I}_i = \hat{I}(S_i'; \theta_i')$ by (6) or (7) with $\mathcal{B}$;
  end for
  $S \leftarrow S - \eta \nabla_S \frac{1}{M} \sum_{i=1}^M \hat{I}_i$;
  $f \leftarrow f - \eta \nabla_f \frac{1}{M} \sum_{i=1}^M \hat{I}_i$;
  $T \leftarrow T - \eta \nabla_T \frac{1}{M} \sum_{i=1}^M \hat{I}_i$; // if (6) was used for $\hat{I}_i$
end while
return $S(\cdot)$

Algorithm 2 SNL with slice sufficient statistics

Input: prior $\pi(\boldsymbol{\theta})$, observed data $\mathbf{x}^o$
Output: estimated posterior $\hat{\pi}(\boldsymbol{\theta}|\mathbf{x}^o)$
Parameters: neural density estimators q, proxy q'
Initialization: $\mathcal{D} = \varnothing, p_1(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$
for r in 1 to R **do**
  repeat
    sample $\boldsymbol{\theta}^{(i)} \sim p_r(\boldsymbol{\theta})$;
    simulate $\mathbf{x}^{(i)} \sim p(\mathbf{x}|\boldsymbol{\theta}^{(i)})$;
  until n' samples
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{n'}$
  learn statistics $S(\cdot)$ with $\mathcal{D}$ by Algorithm 1;
  $\hat{p}(S|\boldsymbol{\theta}) = \arg\max_q \sum_{i=1}^n \log q(S(\mathbf{x}^{(i)})|\boldsymbol{\theta}^{(i)})$;
  $\hat{\pi}(\boldsymbol{\theta}|S^o) \propto \pi(\boldsymbol{\theta}) \cdot \hat{p}(S(\mathbf{x}^o)|\boldsymbol{\theta})$;
  $p_{r+1}(\boldsymbol{\theta}) \leftarrow q'(\boldsymbol{\theta})$ where $q'(\boldsymbol{\theta})$ is learned by (12);
end for
return $\hat{\pi}(\boldsymbol{\theta}|S^o)$

- *Moment statistics* (Fearnhead & Prangle, 2012). This corresponds to taking S as the posterior mean $\hat{\mathbb{E}}[\boldsymbol{\theta}|\mathbf{x}]$:

$$S = \arg\min_s \mathbb{E}[\|s(\mathbf{x}) - \boldsymbol{\theta}\|_2^2].$$

  This can be seen as a degenerate case of our method where exactly K one-hot slices $\boldsymbol{\phi} \in \{0, 1\}^K$ are used. To see this, remark minimising $\sum_k \mathbb{E}[\|s(\mathbf{x})_k - \theta_k\|_2^2]$ is equivalent to maximising a lower bound of $I(s(\mathbf{x})_k; \theta_k), \forall k$, and the latter equals $I(s(\mathbf{x})_k; \boldsymbol{\phi}_k^\top \boldsymbol{\theta})$ if $\boldsymbol{\phi}_k$ is one-hot.

3.2. Determining the dimensionality of statistics

A remaining problem is how to determine d, the dimensionality of the sufficient statistics S. The Pitman-Koopman-Darmois theorem (Koopman, 1936) shows that sufficient statistics with fixed dimensionality only exist for the exponential family, so there is no way to pre-define d. We hence use a data-driven way to determine d based on Theorem 2:

Theorem 2. *Let $\mathbf{x} \in \mathbb{R}^D$ and $\boldsymbol{\theta} \in \mathbb{R}^K$ be two random variables. Consider optimising the following objective function w.r.t a deterministic function $s : \mathbb{R}^D \to \mathbb{R}^J$:*

$$\max_s \sum_{j=1}^J I(s(\mathbf{x})_{\leq j}; \boldsymbol{\theta}), \tag{8}$$

where $s(\mathbf{x})_{\leq j}$ denotes the first j dimensions of $s(\mathbf{x})$. Let $S = s(\mathbf{x})$ be the random variable induced by $s(\cdot)$ learned in (8) and S_j be its jth dimension. We then have

$$I(S_j; \boldsymbol{\theta}|S_{<j}) \leq I(S_{j-1}; \boldsymbol{\theta}|S_{<j-1}).$$

Proof. See Appendix A. □

That is, similar to PCA, the dimensions in S as learned by (8) will be *ordered*, with most information about $\boldsymbol{\theta}$ concentrating on the leading dimensions of S. This allows us to choose d by inspecting the contribution of each dimension. For example, if we discover the informativeness of the first K and the first $2K$ dimensions of S are very similar, we know $d = K$ is enough. Equivalently, we can also compare the posteriors yielded by the first K and the first $2K$ dimensions of S, and set $d = K$ if we find them similar[1] (here we only consider $d = K$ v.s. $d = 2K$ as we find that $d \leq 2K$ is often enough for achieving sufficiency in practice). Under this setting, the objective (8) can be simplified as

$$\max_s SI(s(\mathbf{x})_{\leq K}; \boldsymbol{\theta}) + SI(s(\mathbf{x})_{\leq 2K}; \boldsymbol{\theta}), \tag{9}$$

which has only two terms. Here we have replaced I by SI.

4. Posterior inference

4.1. Algorithm

SNL with sufficient statistics. Once S is learned, we can use it to replace the raw data $\mathbf{x}$ in inference. Sequential Neural Likelihood (SNL) (Papamakarios et al., 2019) is used here as the inference method. Given data $\mathcal{D} = \{\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^n$ and a statistics function $S(\cdot)$, SNL first approximates the likelihood function $p(S|\boldsymbol{\theta})$ as:

$$\hat{p}(S|\boldsymbol{\theta}) = \arg\max_q \sum_{i=1}^n \log q(S(\mathbf{x}^{(i)})|\boldsymbol{\theta}^{(i)}), \tag{10}$$

[1] A Gaussian copula approximation to $\hat{\pi}(\boldsymbol{\theta}|S^o)$ can be used for this purpose, which allows interpretable comparison of both the marginal distributions and the dependency structure. We will detail in Sec. 4.1 how to obtain such a Gaussian copula approximation.

Table 1. A summary of the inference tasks considered.

	g-and-k	Bayesian LR	Ricker model	OU Process	Ising model
parameters	$\boldsymbol{\theta} \in \mathbb{R}^9$	$\boldsymbol{\theta} \in \mathbb{R}^{12}$	$\boldsymbol{\theta} \in \mathbb{R}^3$	$\boldsymbol{\theta} \in \mathbb{R}^6$	$\boldsymbol{\theta} \in \mathbb{R}^2$
data type	i.i.d	i.d	time-series	time-series	image
true posterior by	numerically	analytic	particle filtering (SMC)	analytic	ABC with known S^*

where q is a neural density estimator (Papamakarios et al., 2017; Durkan et al., 2019). It then estimates the posterior $\pi(\boldsymbol{\theta}|S^o)$ by Bayes rule:

$$\hat{\pi}(\boldsymbol{\theta}|S^o) \propto \pi(\boldsymbol{\theta}) \cdot \hat{p}(S^o|\boldsymbol{\theta}). \tag{11}$$

As mentioned in Sec. 2, to accelerate inference, learning in SNL can be made sequential where the current estimate of the posterior $\hat{\pi}(\boldsymbol{\theta}|S^o)$ is used as the proposal distribution $p(\boldsymbol{\theta})$ in the future. This procedure is shown in Algorithm 2.

Neural copula proxy. A question in the above sequential learning procedure is how to efficiently sample $\boldsymbol{\theta}$ from the unnormalised posterior (11). To facilitate fast sampling from (11), following Glöckler et al. (2022), we learn an easy-to-sample proxy $q'(\boldsymbol{\theta})$ to the unnormalised posterior:

$$q'(\boldsymbol{\theta}) = \arg\min_q \mathbb{E}_{q(\boldsymbol{\theta})}\Big[\log \pi(\boldsymbol{\theta})\hat{p}(S^o|\boldsymbol{\theta}) - \log q(\boldsymbol{\theta})\Big] \tag{12}$$

which is equivalent to minimising $\mathrm{KL}[q'(\boldsymbol{\theta})\|\hat{\pi}(\boldsymbol{\theta}|S^o)] + C$ w.r.t q' where C is a constant unrelated to q'. Here, we choose to model q' by a *neural Gaussian copula*:

$$\boldsymbol{\theta} \sim q'(\boldsymbol{\theta}) \;\; \Leftrightarrow \;\; \theta_l = g_l(\epsilon_l), \epsilon \sim \mathcal{N}(\epsilon; 0, \mathbf{V}),$$

where $g_l(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonic neural network. We note that q' needs not to be a very accurate approximation as the goal here is fast sampling. In addition to fast sampling, this neural Gaussian copula also supports easy inspection of both the marginal distributions $q'(\boldsymbol{\theta}_l)$ and the dependency structure $\mathbf{V}$, making it also an interpretable proxy.

4.2. Analysis

Below we discuss why the above 'slice sufficient statistics + SNL' method may be more preferable than other strategies.

SNL + SSS v.s. SNPE. One merit of SNL is its affinity to sequential learning, where one can readily use any proposal distribution $p(\boldsymbol{\theta})$ for $\boldsymbol{\theta}$. As comparison, methods like SNPE-A (Papamakarios & Murray, 2016) and SNPE-B (Lueckmann et al., 2017) may suffer from several issues during sequential learning (SNPE-A: numerical instability; SNPE-B: high variance of the objective due to importance weighting), see e.g. (Greenberg et al., 2019) for an analysis.

SNL + SSS v.s. SNR. Unlike SNPE-A and SNPE-B, recent LFI approaches e.g. SNPE-C (Greenberg et al., 2019) and SNR (Hermans et al., 2020) naturally fit well with sequential learning. These methods learn the posterior by contrastive learning (Durkan et al., 2020) and are in essence ratio estimators (Gutmann & Hyvärinen, 2009; Gutmann & Hyvärinen, 2012; Thomas et al., 2022) where the ratio $p(\boldsymbol{\theta}, \mathbf{x})/p(\boldsymbol{\theta})p(\mathbf{x})$ is learned. However, ratio estimation by contrastive learning can be unreliable if the two distributions $p(\boldsymbol{\theta}, \mathbf{x})$ and $p(\boldsymbol{\theta})p(\mathbf{x})$ are too distinct (Rhodes et al., 2020; Choi et al., 2022; Gutmann et al., 2022) (e.g. high-dimensional cases). In our experiments, we show how these methods can struggle in high-dimensional ratio estimation.

5. Experiments

5.1. Setup

Evaluation metric. We assess inference quality by the discrepancy between the true posterior $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ and the inferred posterior $\hat{\pi}(\boldsymbol{\theta}|\mathbf{x}^o)$:

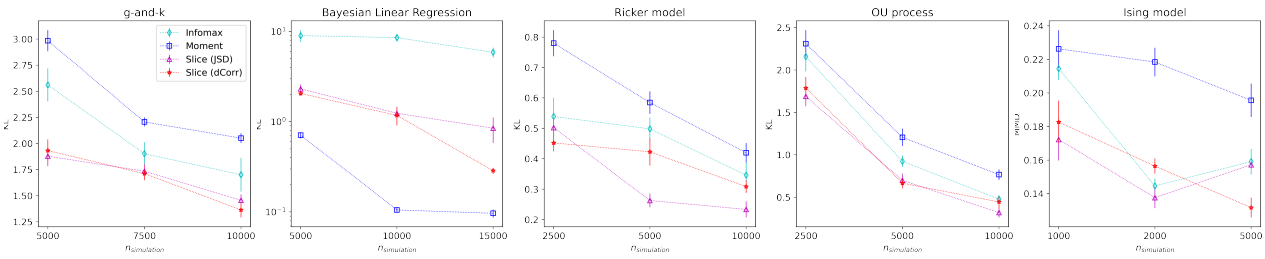
$$\Delta(\pi(\boldsymbol{\theta}|\mathbf{x}^o), \hat{\pi}(\boldsymbol{\theta}|\mathbf{x}^o)),$$

where $\Delta(p, q)$ is some discrepancy between two distributions p and q. Δ is taken as either (a) the KL divergence when the true posterior $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ is analytically available or can be approximated up to high precision; or (b) MMD when we can readily collect samples $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}|\mathbf{x}^o)$ from the true posterior and the dimensionality of $\boldsymbol{\theta}$ is low, depending on the problem.[2] See Appendix B. We note that the evaluation of LFI methods is still an open problem (Lueckmann et al., 2021; Forrow & Baker, 2021; Hermans et al., 2021).

Baselines. We compare the proposed method from two angles, namely against other summary statistics and against other inference strategies. In more details:

- *Summary statistics*. We compare our slice-based sufficient statistics with the classic moment-based statistics (Fearnhead & Prangle, 2012) and the recent infomax statistics (Chen et al., 2021). All methods use SNL in inference.
- *Inference methods*. We also compare the proposed 'SNL + SSS' algorithm with other neural LFI algorithms: SNL (Papamakarios et al., 2019), SNPE-C (Greenberg et al., 2019) and SNR (Hermans et al., 2020).

[2] Classifier two sample test (Lueckmann et al., 2021; Gutmann et al., 2018) may also be used. See Appendix B for a discussion.



(a) g-and-k (b) Bayesian LR (c) Ricker model (d) OU process (e) Ising model

Figure 2. Comparing different summary statistics learning approaches. **x-axis**: the simulation budget. **y-axis**: the discrepancy between the inferred and the true posteriors. *Infomax* corresponds to learning the summary statistics via the infomax principle (Chen et al., 2021): $S = \arg\max_s I(s(\mathbf{x}); \boldsymbol{\theta})$. *Moment* is the classic method which takes the predicted posterior mean as the summary statistics (Fearnhead & Prangle, 2012): $S(\mathbf{x}) = \hat{\mathbb{E}}[\boldsymbol{\theta}|\mathbf{x}]$. Standard error of the mean is reported in the figure. Results are obtained from 20 independent runs.

Note that SNPE-C/SNR can also be seen as infomax learners where different estimators of $I(\mathbf{x}; \boldsymbol{\theta})$ are used (SNPE-C: InfoNCE (Oord et al., 2018); SNR: JSD (Hjelm et al., 2018)). On the other hand, the infomax statistics method can be seen as a variant to SNR where the role of sufficient statistics is explicitly considered in the design of network architecture by the use of an encoder. In the experiments below, we also use the same encoder for SNPE-C/SNR, so different LFI methods only differ in the way they learn the posterior.

Hyperparams. Throughout the experiments we use $M = 8$ slices and set $d = K$ (except for the experiments where we select d according to Section 3.2) and $d' = 2$. An ablation study on the effect of the number of slices is in Appendix B.

5.2. Results

Figure 2 shows an overview of the results and Table 1 summarises the properties of the tasks considered. The tasks cover different data types and were chosen such that the true posteriors are known or can be approximated accurately.

Multivariate g-and-k model. The first model we consider is a well-known benchmark in LFI. The data in this model is generated as

$$x_l = Q(z_l; \boldsymbol{\theta}), \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{V}),$$

$$Q(z_l; \boldsymbol{\theta}) = A_l + B_l\Big(1 + 0.8 \cdot \frac{1 - e^{-g_l z_l}}{1 + e^{g_l z_l}}\Big)(1 + z_l^2)^{g_l} z_l.$$

where $l = 1, 2$ and $\mathbf{V} = [[1, \rho], [\rho, 1]]$. The parameter of interest is $\boldsymbol{\theta} = \{\{A_l, B_l, g_l, k_l\}_{l=1}^2, \rho\}$ satisfying $B_l > 0, g_l > -0.5, k > 0, \rho \in (-1, 1)$. This model has been shown to be very flexible in approximating many 2D distributions with only a few parameters, though its likelihood is not analytic. The data here is a population of 100 i.i.d. samples drawn from the model: $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{100}$. This data is pre-processed by computing a low-level statistics $\mathbf{X}'$ containing (a) 12 equally-spaced marginal quantiles and (b) the Spearman's rank correlation between the two marginals, and we infer the posterior $\pi(\boldsymbol{\theta}|\mathbf{X}'^o) \propto \pi(\boldsymbol{\theta})p(\mathbf{X}'^o|\boldsymbol{\theta})$ given the observed $\mathbf{X}'^o$. While the likelihood function of this model is not analytic, it can be approximated accurately by inverting $\hat{z}_d = Q^{-1}(x_d; \boldsymbol{\theta})$ numerically (e.g. by using gradient descent: $z_d \leftarrow z_d - \eta \nabla_{z_d} \|Q(z_d; \boldsymbol{\theta}) - x_d\|_2^2$).

Figure 2.(a) shows the results of different summary statistics learning strategies. In this 9-dimensional problem, the proposed slice-based method clearly outperforms the existing infomax approach. This may be because, unlike the infomax approach, our slice-based method does not need to estimate $I(S; \boldsymbol{\theta})$ in the original space. In Table 2, we further compare the proposed SNL + SSS method with SNR and SNPE-C, using KL divergence as the discrepancy metric. We see that our method is not only more accurate but also more robust, as indicated by the smaller standard deviation in the table.

Bayesian linear regression. The goal of Bayesian linear regression is to infer the parameters $\boldsymbol{\theta}$ of a linear map from noisy observations of outputs at known inputs. The setup is

$$\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \mathbf{I}), p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{U}) = \prod_{l=1}^{L} \mathcal{N}(x_l; \boldsymbol{\theta}^\top \mathbf{u}_l, \sigma^2)$$

where $\mathbf{x} = \{x_l\}_{l=1}^L \in \mathbb{R}^{L \times 1}$and $\mathbf{U} = \{\mathbf{u}_l\}_{l=1}^L \in \mathbb{R}^{L \times K}$ are the output and input data respectively. Each input $\mathbf{u}_l$ is generated as $\mathbf{u}_l \sim \mathcal{N}(0, \mathbf{V})$ with $\mathbf{V}_{ij} = 0.4$ if $i \neq j$ and $\mathbf{V}_{ij} = 1$ otherwise. We wish to infer $\pi(\boldsymbol{\theta}|\mathbf{x}^o, \mathbf{U}) \propto \pi(\boldsymbol{\theta})p(\mathbf{x}^o|\boldsymbol{\theta}, \mathbf{U})$ given $\mathbf{x}^o$. Here $L = 50$ and $K = 12$. Since the prior and the likelihood function are both multivariate Gaussian distributions, the posterior is analytically known. We use this task to illustrate a typical failure mode of ratio-estimating methods (SNR, SNPE-C).

We first compare different summary statistics learning approaches in Figure 2.(b). From the figure we see that infomax statistics are unreliable for this problem: the KL between the inferred posterior $\hat{\pi}(\boldsymbol{\theta}|S(\mathbf{x}^o))$ and the true poste-

Table 2. A comparison of different neural LFI algorithms. The numbers show the discrepancies $\Delta(\pi(\boldsymbol{\theta}|\mathbf{x}_o), \hat{\pi}(\boldsymbol{\theta}|s(\mathbf{x}_o)))$ averaged over 20 runs ($\pm$ standard deviations). Here the results for SNL + SSS correspond to the case where the JSD proxy (6) is used for MI estimate.

	g-and-k	**Bayesian LR**	**Ricker model**	**OU Process**	**Ising model**
SNL + SSS	1.591 ± 0.189	1.231 ± 1.011	0.261 ± 0.101	0.479 ± 0.198	0.137 ± 0.028
SNL	1.992 ± 0.517	0.598 ± 0.179	1.887 ± 0.792	1.745 ± 0.447	0.917 ± 0.224
SNPE-C	2.082 ± 0.325	13.45 ± 3.846	0.413 ± 0.156	1.428 ± 0.457	0.152 ± 0.055
SNR	1.903 ± 0.346	8.534 ± 3.702	0.498 ± 0.164	1.009 ± 0.580	0.144 ± 0.019
Δ	KL	KL	KL	KL	MMD
n. simulations	7,500	10,000	5,000	5,000	2,000

rior $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ is at least an order larger than other approaches (note that the y-axis in this figure is in log scale). One reason may be that the underlying mutual information $I(S(\mathbf{x}); \boldsymbol{\theta})$ is relatively high, making it very difficult to estimate from $n \leq 10,000$ samples. This result also echoes recent studies in mutual information estimation (Song & Ermon, 2019; Rhodes et al., 2020; Choi et al., 2022). On the other hand, we also see that the moment method, which solves a regression problem instead of mutual information estimate, works very well for this Gaussian problem.[3] Our method, as analysed in Sec. 3.1, can be seen as in the middle of these two methods, so its performance is in-between.

The second column in Table 2 compares the performance of our SNL+SSS algorithm and other LFI algorithms. As expected, we see SNPE-C and SNR not performing well. This coincides with recent studies (Song & Ermon, 2019; Rhodes et al., 2020; Choi et al., 2022) reporting the unreliability of ratio estimation/mutual information methods in medium-to-high dimensional settings. On the other hand, we discover that SNL actually works better than our method. This may be because (a) the dimensionality of the data $\mathbf{x} \in \mathbb{R}^{50}$ in this problem is not so high, so SNL works reasonably well; (b) there is information loss incurred by slicing in our method.

Ricker model. The next model we consider is a state-space model where the transition function is non-linear and non-Gaussian. It is widely used in ecology to describe the evolution of an animal population (Wood, 2010). Given parameters of interest $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$, the data generating process in this model can be described as:

$$v_t = v_{t-1} e^{\theta_1 + \theta_2 \epsilon_t - v_{t-1}}, \quad \epsilon_t \sim \mathcal{N}(\epsilon_t; 0, 1),$$

$$x_t \sim \text{Poisson}(\theta_3 v_t),$$

where only $\mathbf{x} = \{x_t\}_{t=1}^T$ is observed. This results in an intractable likelihood function $p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{v}|\boldsymbol{\theta}) d\mathbf{v}$ for this model. While intractable, the likelihood function $p(\mathbf{x}|\boldsymbol{\theta})$ of this model can be numerically approximated by the particle filtering algorithm (Andrieu et al., 2010) up to high precision using a large number of particles (e.g. 10^6) and adaptive proposals (Paige & Wood, 2016). Here we set $T = 30$, so that $\mathbf{x} \in \mathbb{R}^{30}$. The goal is to infer the posterior $\pi(\boldsymbol{\theta}|\mathbf{x}^o) \propto \pi(\boldsymbol{\theta}) p(\mathbf{x}^o|\boldsymbol{\theta})$ under a uniform prior.

As the data $\mathbf{x}$ in this model takes the form of a time series, it is natural to use a 1D convolution neural network as the backbone for $S(\cdot)$ (the same architecture is also used for SNPE-C and SNR). Figure 2.(c) summarises the results that compare different summary statistics learning methods. For this model, the improvement brought by slicing is considerable, especially when the JSD estimator is used. The 3rd column in Table 2 tells the same story. The poor performance of moment statistics may be due to its insufficiency.

Ornstein-Uhlenbeck process. The fourth model considered is a time-series model where the data generating process is governed by a stochastic differential equation (SDE):

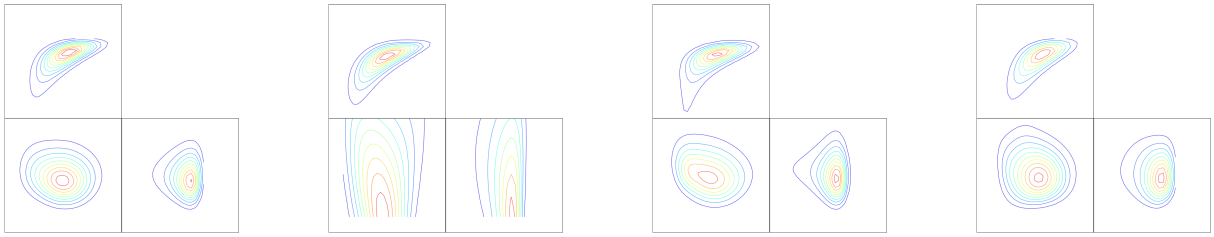
$$d\mathbf{x}_t = f(\mathbf{x}_t, \boldsymbol{\theta}) dt + g(\mathbf{x}_t, \boldsymbol{\theta}) \odot (\mathbf{A} dW_t),$$

$$f_k(\mathbf{x}_t, \boldsymbol{\theta}) = \theta_{2k-1}(e^{\theta_{2k}} - x_{t,k}), \qquad g_k(\mathbf{x}_t, \boldsymbol{\theta}) = \theta_{4+k}$$

where $k = 1, 2$ and $\mathbf{A}\mathbf{A}^\top = [[1, 0.55], [0.55, 1]]$. This SDE can easily be simulated by the Euler-Maruyama method, and we simulate it for an overall time of $T = 10$. We record the simulated data after every $\Delta_t = 0.2$ time units, resulting in an observed time-series $\mathbf{x}^o \in \mathbb{R}^{2\times 50}$. The goal is to infer $\pi(\boldsymbol{\theta}|\mathbf{x}^o)$ under a uniform prior. Since we do not observe the full trajectories, the likelihood of SDEs is generally intractable, but the model here has an analytic likelihood.

Similar to the setting in the Ricker model, we use a 1D convolution neural network to process the time-series data in this task. See Appendix B for its architecture. Figure 2.(d) summarises the results that compare different summary statistics learning methods. We see that for this 6-dimensional problem, the proposed slice-based method again performs much better than the infomax method. This is especially the case for small data regimes (e.g. $n \leq 5000$),

[3] It can be shown that the ground-truth sufficient statistics for this problem is exactly the conditional mean: $S^*(\mathbf{x}) = \mathbb{E}[\boldsymbol{\theta}|\mathbf{x}]$.



(a) True posterior (b) SNPE-C (c) SNL + SSS, $d = K$ (d) SNL + SSS, $d = 2K$

Figure 3. OU process, example contour plots for inferred posteriors. Inference is done with 5,000 samples. The figures show the marginal posterior $\hat{\pi}(\theta_i, \theta_j|\mathbf{x}^o)$ for $i, j \in \{1, 2, 3\}$. The plots (c) and (d) visualise the contours of the neural Gaussian copula proxy in (12).

though as more data is available (e.g. $n = 10000$) the performance gap becomes minor. The comparison in Table 2 again highlights the advantage over SNL, SNPE-C, SNR.

In Figure 3 we further compare the true posterior (panel (a)), with the approximation by SNPE-C (panel (b)), and the posterior approximated by the neural Gaussian copula proxy (panels (c) and (d)). We see that the neural copula proxy does a good job in approximating the posterior, while supporting easy visualisation of marginal distributions, in addition to fast sampling. Figure 3 also provides an example for how to use the neural copula proxy to determine d. In this example, a practitioner can set $d = K$, as $d = K$ and $d = 2K$ yield very similar posteriors.

Ising model. The last model we consider originated from statistical physics and describes the states of atomic spins on a 8×8 lattice. Each spin has two states described by a discrete random variable $x_i \in \{-1, +1\}$. Given parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$, the probability mass function of this model is

$$p(\mathbf{x}|\boldsymbol{\theta}) \propto \exp(-H(\mathbf{x}; \boldsymbol{\theta})),$$

$$H(\mathbf{x}; \boldsymbol{\theta}) = -\theta_1 \sum_{\langle i,j \rangle} x_i x_j - \theta_2 \sum_i x_i.$$

where $\langle i, j \rangle$ denotes that spin i and spin j are adjacent. Being an energy-based model, the likelihood function of this model is not analytic due to the intractable normalising constant $Z(\boldsymbol{\theta}) = \sum_{\mathbf{x} \in \{-1,1\}^{m \cdot m}} \exp[-H(\mathbf{x}; \boldsymbol{\theta})]$. However, it is possible to sample $\mathbf{x}$ from the model by MCMC. Note that the sufficient statistics are known for this model: $S^*(\mathbf{x}) = \{\sum_{\langle i,j \rangle} x_i x_j, \sum_i x_i\}$. The true posterior can be approximated by rejection ABC algorithm due to the existence of low-dimensional sufficient statistics $S^*(\mathbf{x}) \in \mathbb{Z}^2$.

As the data $\mathbf{x} \in \{-1, +1\}^{8 \times 8}$ in this task takes the form of an image, we use a 2D convolutional network to process the data. Figure 2.(e) compares the performance of different summary statistics. It can be seen that for this low-dimensional problem, our slice method is close to the infomax method for the cases $n \geq 2000$, though it is better for smaller values of n. We believe this is because the parameter space in this task is only two-dimensional, so the benefit brought by slicing is not notable. In fact, as slicing always incurs information loss, the advantages brought by slicing (e.g. sample efficiency) may be cancelled out.

6. Conclusion

This work presents slice sufficient statistics (SSS), a new method for constructing summary statistics in likelihood-free inference (LFI). The main message is that the learning of sufficient statistics may be easier than direct posterior inference. Motivated by this observation, we further develop a new LFI algorithm, SNL+SSS, which is shown to outperform state-of-the-art inference strategies (e.g. SNPE-C, SNR) on diverse inference tasks. As a byproduct, we shed light on shortcomings of SNPE-C and SNR, namely that these ratio estimation-based methods can be unreliable when the parameter of the inference task is high-dimensional.

Our work highlights the importance of considering what is easier to learn for an implicit model. If some objects (e.g. summary statistics) are easier to learn than the others (e.g. density), then one should learn the former first to help learning the latter. In this regard, score estimator (Pacchiardi & Dutta, 2022) is also worth considering in the future.

It should be noted that, while powerful, the proposed slice-based method is not a silver bullet: it comes at the price of a higher computational cost and a potential loss of sufficiency. Nonetheless, as the method strikes a better trade-off between sufficiency and sample efficiency, we found it among the best performing methods in a wide range of settings.

While focusing on LFI, due to the connection between sufficient statistics and the infomax principle, we believe the proposed slice method can also generalise to other infomax representation learning tasks (Hjelm et al., 2018; Oord et al., 2018; Chen et al., 2020), and we leave this to future works.

Acknowledgements

WAW acknowledges support from a Turing AI Fellowship under grant EP/V025279/1 and the Leverhulme Trust via CFI. YC acknowledges support from the Cambridge Trust.

References

Alsing, J., Wandelt, B., and Feeney, S. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology. *Monthly Notices of the Royal Astronomical Society*, 477(3):2874–2885, 2018.

Andrieu, C., Doucet, A., and Holenstein, R. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

Bansal, R. and Yaron, A. Risks for the long run: A potential resolution of asset pricing puzzles. *The journal of Finance*, 59(4):1481–1509, 2004.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

Blum, M. G., Nunes, M. A., Prangle, D., Sisson, S. A., et al. A comparative review of dimension reduction methods in approximate bayesian computation. *Statistical Science*, 28(2):189–208, 2013.

Brehmer, J., Louppe, G., Pavez, J., and Cranmer, K. Mining gold from implicit models to improve likelihood-free inference. *Proceedings of the National Academy of Sciences*, 117(10):5242–5249, 2020.

Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S., and Song, Y. A likelihood-free inference framework for population genetic data using exchangeable neural networks. In *Advances in Neural Information Processing Systems*, pp. 8594–8605, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, Y., Zhang, D., Gutmann, M. U., Courville, A. C., and Zhu, Z. Neural approximate sufficient statistics for implicit models. In *ICLR*, 2021.

Chen, Y., Li, Y., Weller, A., et al. Scalable infomin learning. *Advances in Neural Information Processing Systems*, 35: 2226–2239, 2022.

Chinazzi, M., Davis, J. T., Ajelli, M., Gioannini, C., Litvinova, M., Merler, S., y Piontti, A. P., Mu, K., Rossi, L., Sun, K., et al. The effect of travel restrictions on the spread of the 2019 novel coronavirus (covid-19) outbreak. *Science*, 368(6489):395–400, 2020.

Choi, K., Meng, C., Song, Y., and Ermon, S. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pp. 2552–2573. PMLR, 2022.

Diggle, P. J. and Gratton, R. J. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B*, pp. 193–227, 1984.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. *arXiv preprint arXiv:1906.04032*, 2019.

Durkan, C., Murray, I., and Papamakarios, G. On contrastive learning for likelihood-free inference. *arXiv preprint arXiv:2002.03712*, 2020.

Dyer, J., Cannon, P. W., and Schmon, S. M. Deep signature statistics for likelihood-free time-series models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

Dyer, J., Cannon, P., Farmer, J. D., and Schmon, S. Black-box bayesian inference for economic agent-based models. *arXiv preprint arXiv:2202.00625*, 2022.

Fearnhead, P. and Prangle, D. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.

Forrow, A. and Baker, R. E. Measuring the accuracy of likelihood-free inference. *arXiv preprint arXiv:2112.08096*, 2021.

Glöckler, M., Deistler, M., and Macke, J. H. Variational methods for simulation-based inference. *arXiv preprint arXiv:2203.04176*, 2022.

Goldfeld, Z. and Greenewald, K. Sliced mutual information: A scalable measure of statistical dependence. *Advances in Neural Information Processing Systems*, 34:17567–17578, 2021.

Greenberg, D., Nonnenmacher, M., and Macke, J. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pp. 2404–2414, 2019.

Gutmann, M. U. and Corander, J. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125): 1–47, 2016.

Gutmann, M. U. and Hyvärinen, A. Learning features by contrasting natural images with noise. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2009.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

Gutmann, M. U., Kleinegesse, S., and Rhodes, B. Statistical applications of contrastive learning. *Behaviormetrika*, pp. 1–25, 2022.

Hermans, J., Begy, V., and Louppe, G. Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pp. 4239–4248. PMLR, 2020.

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., and Louppe, G. Averting a crisis in simulation-based inference. *arXiv preprint arXiv:2110.06581*, 2021.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

Järvenpää, M., Gutmann, M., Vehtari, A., and Marttinen, P. Gaussian process modeling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria. *Annals of Applied Statistics*, 2018.

Järvenpää, M., Gutmann, M., Vehtari, A., and Marttinen, P. Efficient acquisition rules for model-based approximate Bayesian computation. *Bayesian Analysis*, 14(2):595–622, 2019. doi: doi:10.1214/18-BA1121.

Järvenpää, M., Gutmann, M. U., Vehtari, A., and Marttinen, P. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16 (1):147–178, 2021.

Koopman, B. O. On distributions admitting a sufficient statistic. *Transactions of the American Mathematical society*, 39(3):399–409, 1936.

Lopez-Guevara, T., Taylor, N., Gutmann, M., Ramamoorthy, S., and Subr, K. Adaptable pouring: Teaching robots not to spill using fast but approximate fluid simulation. In Levine, S., Vanhoucke, V., and Goldberg, K. (eds.), *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, volume 78 of *Proceedings of Machine Learning Research*, pp. 77–86, November 2017.

Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, pp. 1289–1299, 2017.

Lueckmann, J.-M., Boelts, J., Greenberg, D. S., Gonçalves, P. J., and Macke, J. H. Benchmarking simulation-based inference. *arXiv preprint arXiv:2101.04653*, 2021.

Mansinghka, V. K., Kulkarni, T. D., Perov, Y. N., and Tenenbaum, J. Approximate bayesian image interpretation using generative probabilistic graphics programs. In *Advances in Neural Information Processing Systems*, pp. 1520–1528, 2013.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

Oliveira, R., Ott, L., and Ramos, F. No-regret approximate inference via bayesian optimisation. In de Campos, C. and Maathuis, M. H. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 2082–2092, 2021.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pacchiardi, L. and Dutta, R. Score matched neural exponential families for likelihood-free inference. *J. Mach. Learn. Res.*, 23:38–1, 2022.

Paige, B. and Wood, F. Inference networks for sequential monte carlo in graphical models. In *International Conference on Machine Learning*, pp. 3040–3049. PMLR, 2016.

Papamakarios, G. and Murray, I. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pp. 1028–1036, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Papamakarios, G., Sterratt, D., and Murray, I. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 837–848. PMLR, 2019.

Poole, B., Ozair, S., Oord, A. v. d., Alemi, A. A., and Tucker, G. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798, 1999.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.

Sisson, S., Fan, Y., and Beaumont, M. *Handbook of Approximate Bayesian Computation.*, chapter Overview of Approximate Bayesian Computation. Chapman and Hall/CRC Press, 2018.

Sjöstrand, T., Mrenna, S., and Skands, P. A brief introduction to pythia 8.1. *Computer Physics Communications*, 178(11):852–867, 2008.

Song, J. and Ermon, S. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019.

Székely, G. J., Rizzo, M. L., et al. Partial distance correlation with methods for dissimilarities. *Annals of Statistics*, 42 (6):2382–2412, 2014.

Thomas, O., Dutta, R., Corander, J., Kaski, S., and Gutmann, M. U. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31, 2022.

Weyant, A., Schafer, C., and Wood-Vasey, W. M. Likelihood-free cosmological inference with type ia supernovae: approximate Bayesian computation for a complete treatment of uncertainty. *The Astrophysical Journal*, 764(2), 2013.

Wiqvist, S., Mattei, P.-A., Picchini, U., and Frellsen, J. Partially exchangeable networks and architectures for learning summary statistics in approximate bayesian computation. In *International Conference on Machine Learning*, pp. 6798–6807, 2019.

Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102, 2010.

Supplementary Materials

Yanzhi Chen [1] **Michael U. Gutmann** [2] **Adrian Weller** [1]

A. Theorem proofs

Theorem 1. *Let $\mathbf{x} \in \mathbb{R}^D$ and $\boldsymbol{\theta} \in \mathbb{R}^K$ be two random variables and $S : \mathbb{R}^D \to \mathbb{R}^d$ be a deterministic function. Then $S(\mathbf{x})$ is a sufficient statistics if and only if $S(\mathbf{x})$ maximises $SI(S(\mathbf{x}); \boldsymbol{\theta})$,*

$$SI(S(\mathbf{x}); \boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\phi} \sim \mathbb{S}^{K-1}}[I(S(\mathbf{x}); \boldsymbol{\phi}^\top \boldsymbol{\theta})], \tag{1}$$

where $\boldsymbol{\phi} \in \mathbb{S}^{K-1}$ is a vector uniformly sampled from the surface of a K-dimensional unit sphere $\mathbb{S}^{K-1}$.

Proof: The key of the proof is to show if $S(X)$ is the sufficient statistics of $\boldsymbol{\theta}$, then it is also the sufficient statistics for $\phi^\top \boldsymbol{\theta}, \forall \phi \in \mathbb{S}^{K-1}$ and vice versa. We first recall a recent result in statistics (Proposition 1, (Nadjahi et al., 2020)):

$$KL[p(\boldsymbol{\theta}) \| q(\boldsymbol{\theta})] = 0 \;\; \Leftrightarrow \;\; KL[p(\phi^\top \boldsymbol{\theta}) \| q(\phi^\top \boldsymbol{\theta})] = 0, \forall \phi \in \mathbb{S}^{K-1}$$

This means that for a particular X and a particular function $S(\cdot)$,

$$KL[p(\boldsymbol{\theta}|X) \| p(\boldsymbol{\theta}|S(X))] = 0 \;\; \Leftrightarrow \;\; KL[p(\phi^\top \boldsymbol{\theta}|X) \| p(\phi^\top \boldsymbol{\theta}|S(X))] = 0, \forall \phi \in \mathbb{S}^{K-1}$$

Therefore,

$$KL[p(\boldsymbol{\theta}|X) \| p(\boldsymbol{\theta}|S(X))] = 0, \forall X \;\; \Leftrightarrow \;\; KL[p(\phi^\top \boldsymbol{\theta}|X) \| p(\phi^\top \boldsymbol{\theta}|S(X))] = 0, \forall X, \forall \phi \in \mathbb{S}^{K-1}$$

Note that the LHS of the above formula implies that $S(X)$ is a sufficient statistics for $\boldsymbol{\theta}$ and the RHS implies that $S(X)$ is a sufficient statistics for $\phi^\top \boldsymbol{\theta}, \forall \phi \in \mathbb{S}^{d-1}$. This essentially means that

$$S = \arg\max_s I(\boldsymbol{\theta}, s(X)) = \arg\max_s I(\phi^\top \boldsymbol{\theta}; s(X)), \forall \phi \in \mathbb{S}^{K-1}$$

i.e. $I(\boldsymbol{\theta}; S(X))$ is maximised if and only if for $\forall \phi \in \mathbb{S}^{K-1}$, $I(\phi^\top \boldsymbol{\theta}; S(X))$ is also maximised. It is then easy to verify $\arg\max_s I(\boldsymbol{\theta}; s(X)) = \arg\max_s \mathbb{E}_{\phi \sim \mathbb{S}^{K-1}}[I(s(X); \phi^\top \boldsymbol{\theta})]$, which completes the proof. □

Theorem 2. *Let $\mathbf{x} \in \mathbb{R}^D$ and $\boldsymbol{\theta} \in \mathbb{R}^K$ be two random variables. Consider optimising the following objective function w.r.t a deterministic function $s : \mathbb{R}^D \to \mathbb{R}^J$:*

$$\max_s \sum_{j=1}^{J} I(s(\mathbf{x})_{\leq j}; \boldsymbol{\theta}), \tag{2}$$

where $s(\mathbf{x})_{\leq j}$ denotes the first j dimensions of $s(\mathbf{x})$. Let $S = s(\mathbf{x})$ be the random variable induced by $s(\cdot)$ learned in (2) and S_j be its jth dimension. We then have

$$I(S_j; \boldsymbol{\theta}|S_{<j}) \leq I(S_{j-1}; \boldsymbol{\theta}|S_{<j-1}).$$

[1]Department of Engineering, University of Cambridge, UK [2]School of Informatics, The University of Edinburgh, UK. Correspondence to: Yanzhi Chen <yc514@cam.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Proof:

$$\underbrace{I(\boldsymbol{\theta}; S_{\leq j})}_{A_j} = I(\boldsymbol{\theta}; [S_{<j}, S_j]) = \underbrace{I(\boldsymbol{\theta}; S_j | S_{<j})}_{I_j} + I(\boldsymbol{\theta}; S_{<j})$$

Therefore

$$A_j = I_j + A_{j-1} = I_j + I_{j-1} + A_{j-2} = ... = \sum_{k=1}^{j} I_k$$

So

$$\max_s \sum_{j=1}^{J} I(s(X)_{\leq j}; \boldsymbol{\theta}) = \sum_{j=1}^{J} A_j = \sum_{j=1}^{J} \sum_{k=1}^{j} I_k = \sum_{j=1}^{J} (J - j + 1) I_j = C \cdot \sum_{j=1}^{J} p_j I_j$$

where the constants p_j and C are

$$p_j = \frac{J - j + 1}{\sum_{k=1}^{J} J - k + 1}, \qquad C = \sum_{k=1}^{J} J - k + 1$$

Note that $\sum_j^J p_j = 1$. Therefore the maximisation of $\max_s \sum_{j=1}^J I(s(X)_{\leq j}; \boldsymbol{\theta})$ is equivalent to the following constraint optimisation problem (note that here $\sum_j I_j = I(S; \boldsymbol{\theta})$, so it is also a constant):

$$\max_{I_j} \sum_{j=1}^{J} p_j I_j,$$

$$s.t. \ \sum_j p_j = 1, \qquad \sum_j^J I_j = I(S; \boldsymbol{\theta})$$

$$p_1 > p_2 > ... > p_J$$

It immediately comes out that $I_1 \geq I_2 \geq ...I_J$, which completes the proof. □

B. Experiment details

KL divergence between the true and the estimate posteriors. In the paper, we use $\mathrm{KL}[p(\boldsymbol{\theta}) \| q(\boldsymbol{\theta})]$ as the evaluation metric. There are two challenges: (a) numerical integration is difficult in high-dimensional cases; (b) the two distributions p and q may be known only up to a normalising constant (for example, in SNL/SNR the posterior is unnormalised). Let $p'(\boldsymbol{\theta})$ and $q'(\boldsymbol{\theta})$ be the unnormalised versions of $p(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ respectively and Z_p and Z_q be their corresponding normalising constants: $p(\boldsymbol{\theta}) \propto p'(\boldsymbol{\theta}), q(\boldsymbol{\theta}) \propto q'(\boldsymbol{\theta}), Z_p = \int p'(\boldsymbol{\theta}) d\boldsymbol{\theta}, Z_q = \int q'(\boldsymbol{\theta}) d\boldsymbol{\theta}$. We compute $\mathrm{KL}[p \| q]$ from p', q' as follows.

Step 1. Proxy distribution learning. We first find an easy-to-sample proxy $t(\boldsymbol{\theta})$ to $p(\boldsymbol{\theta})$ satisfying $t(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta})$. The role of $t(\boldsymbol{\theta})$ is to serve as a good proposal in importance sampling which is used to calculate the normalising constants Z_p, Z_q and also $\mathrm{KL}[p(\boldsymbol{\theta}) \| q(\boldsymbol{\theta})]$ itself later. We do so by running a single round SNPE-A algorithm (Papamakarios & Murray, 2016) with considerable simulation budget (e.g. $n = 10^5$):

$$t(\boldsymbol{\theta}) = \arg\max_Q \frac{1}{n} \sum_{i=1} \log Q(\boldsymbol{\theta}^{(i)} | \mathbf{x}^{(i)}), \qquad \boldsymbol{\theta}^{(i)} \sim \pi(\boldsymbol{\theta}), \mathbf{x}^{(i)} \sim p(\mathbf{x}^{(i)} | \boldsymbol{\theta}^{(i)})$$

We highlight that $t(\boldsymbol{\theta})$ as learned in this way will not suffer from the same problem as in contrastive learning-based methods (i.e. SNPE-C/SNR) mentioned in the main text. Here, we model t by a mixture density network. Note that here $t(\boldsymbol{\theta})$ needs not to be a very precise approximation to $p(\boldsymbol{\theta})$; serving as a good proposal is sufficient.

Step 2. Estimating normalising constants with proxy. After getting the proxy $t(\boldsymbol{\theta})$, we then use this easy-to-sample proxy to estimate the normalising constants by importance sampling:

$$\boldsymbol{\theta}^{(i)} \sim t(\boldsymbol{\theta})$$

$$Z_p = \int p'(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int t(\boldsymbol{\theta}) \frac{p'(\boldsymbol{\theta})}{t(\boldsymbol{\theta})} d\boldsymbol{\theta} \approx \frac{1}{m} \sum_{i=1}^{m} \frac{p'(\boldsymbol{\theta}^{(i)})}{t(\boldsymbol{\theta}^{(i)})}, \quad Z_q = \int q'(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int t(\boldsymbol{\theta}) \frac{q'(\boldsymbol{\theta})}{t(\boldsymbol{\theta})} d\boldsymbol{\theta} \approx \frac{1}{m} \sum_{i=1}^{m} \frac{q'(\boldsymbol{\theta}^{(i)})}{t(\boldsymbol{\theta}^{(i)})}.$$

Step 3. KL computation with proxy. Once we have obtained the the unnormalised $p(\boldsymbol{\theta}) = p'(\boldsymbol{\theta})/Z_p$ and $q(\boldsymbol{\theta}) = q'(\boldsymbol{\theta})/Z_q$, we can compute KL by importance sampling:

$$\mathrm{KL}[p(\boldsymbol{\theta})\|q(\boldsymbol{\theta})] = \int p(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} = \int t(\boldsymbol{\theta}) \underbrace{\frac{p(\boldsymbol{\theta})}{t(\boldsymbol{\theta})}}_{w(\boldsymbol{\theta})} \cdot \log \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \approx \frac{1}{m} \sum_{i=1}^{m} \Big[w(\boldsymbol{\theta}^{(i)}) \cdot \log \frac{p(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})} \Big], \;\; \boldsymbol{\theta}^{(i)} \sim t(\boldsymbol{\theta}).$$

We note that the above estimator for $\mathrm{KL}[p\|q]$ is *biased*. However, it is not difficult to show that this estimator is *consistent* provided that the true normalising constants Z_p and Z_q are bounded (which is often the case). In the experiments, we use m =2,000 samples to estimate Z_p, Z_q and KL. Further increasing the number of samples does not see notable difference.

Evaluation metric: KL vs C2ST. Here we briefly discuss why we prefer KL rather than an alternative metric namely *classifier two samples test* (C2ST) (Gutmann et al., 2018; Lueckmann et al., 2021) when evaluating inference quality. In C2ST, a (neural network-based) classifier is trained to distinguish samples from $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ v.s. samples from $\boldsymbol{\theta} \sim q(\boldsymbol{\theta})$. The classification accuracy $\rho \in [0, 1]$ is then served as the discrepancy between p and q. The pros and cons of such method is:

- *Advantage*. (a) It is interpretable and easy to understand; (b) it is a powerful test there are sufficient samples from p and q.
- *Disadvantage*. It is often difficult to obtain samples from p and q when they are only known up to a normalising constant (and this is unfortunately the case for SNL/SNR). If the samples obtained are inaccurate, the accuracy is questionable.

Note that for the disadvantage above, it can not be resolved by using an easy-to-sample proxy $t(\boldsymbol{\theta})$ (as the one used in the KL metric). This is because this will require $t(\boldsymbol{\theta})$ to approximate $p(\boldsymbol{\theta})$ very well, or we need to resort to importance sampling.

Evaluation metric: MMD. When MMD is used as the discrepancy between the true and the inferred posterior, we use a Gaussian kernel for $\mathrm{MMD}(P, Q)$, with the bandwidth being the median of the pairwise distance $\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|$, $\boldsymbol{\theta}_k \sim P(\boldsymbol{\theta})$.

Neural network architectures. We here provide the details of the architecture of the neural networks used in experiments.

- *Neural density estimator* q. Throughout the paper, we use masked autoregressive flow (MAF) (Papamakarios et al., 2017) for density estimation. Following existing works, we use 5 blocks for this MAF, with 50 tanh units in each block.
- *Neural Gaussian copula* q'. Each marginal transformation function $g_k(\epsilon_k)$ is modelled by a 3-layers monotonic neural network: $g_k(\epsilon_k) = \sum_l^L A_{kl} \tanh(B_{kl}\epsilon_k + C_{kl}) + D_k$ where $A_{kl} \in \mathbb{R}^+, B_{kl} \in \mathbb{R}^+, C_{kl} \in \mathbb{R}, D_k \in \mathbb{R}$. Here $L = 10$.
- *Summary statistics network* $S(\cdot)$. The architecture of $S(\cdot)$ depends on the data type. (a) For i.i.d data, we use a 3-layers MLP with 100 units in each hidden layers; (b) For time-series data, we use a 1D convolution network with 2 convolution layers, with the kernel size being (2, 1) and the number of filters being (50, 100) in the two layers respectively; (c) For image data, we use a 2D convolution network with 2 convolution layers, with the kernel size being (2, 1) and the number of filters being (50, 100) in the two layers respectively. In all cases, the computed features will be fed to a MLP with architecture 100-100-d where d is the dimensionality of the output i.e. $S(\mathbf{x}) \in \mathbb{R}^d$. ReLU is used as the non-linearity. This summary statistics network is also used in SNPE-C and SNR to pre-process the data before the posterior estimators.
- *Secondary encoder network* $f(S, \phi_i)$. For this network, which is used to compute the secondary sufficient statistics $S_i' \in \mathbb{R}^{d'}$ from S and is amortised among different slicing directions $\phi_1, ...\phi_M$, we use a $(d+K) - 100K - d'$ architecture for it. Here '+K' in the first layer corresponds to amortisation. d' is set to be 2 i.e. $d' = 2$. ReLU is used.
- *Ratio estimator* t *in SNR*. For this network, we design its architecture to be $t(\mathbf{x}, \boldsymbol{\theta}) = \mathrm{MLP}(S(\mathbf{x}), \boldsymbol{\theta})$ where $S(\cdot)$ is as above and MLP is a fully-connected network with architecture $(d+K) - 200 - 200 - 1$. ReLU is used as the non-linearity.
- *Mutual information estimator* $T(S_i', \theta_i', \phi_i)$ *for estimating* $I(S_i', \theta_i')$. For this network, which is used to estimate the information between $S_i' \in \mathbb{R}^{d'}$ and $\boldsymbol{\theta}_i' \in \mathbb{R}$, we design it to be a fully-connected network with architecture $(d' + 1 + K) - (100K + K) - 100K - 1$. Here, the two '+$K$' in the first two layers accounts for amortisation. ReLU is used.

Neural network training. For all our experiments, we use early stopping to train all neural networks, where we use 80% of the data in training and 20% in validation (the patience threshold is 500 iterations). All neural nets are trained using Adam (Kingma & Ba, 2014) with its default settings. The learning rate is 5×10^{-4}. A batch size of 200 is used for all networks.

The effect of number of slices M. We investigate here how the number of slices affects performance and execution time.

- *Computational cost*. In Table 3, we compare the execution time of our method under different number of slices. The reported numbers are the training time (in seconds) per mini-batch. One can see that our method is actually not expensive to run compared to training a NDE (the NDE here is a MAF with 5 blocks), especially when the dCorr estimator is used.

- *Performance*. In Table 4 and 5, we investigate how the number of slices affects the performance of our method. Results for both estimators (JSD, dCorr) are reported, along with that of SNPE-C. We see that our method is fairly robust w.r.t different choice of M, with $M \geq 4$ generally working well. One possible explanation for why our method is insensitive to the number of slices is as follows. While using only a small number of slices is insufficient, we consider different slices in different mini-batches across large number of iterations, which compensates for the small number of slices used in each mini-match. Moreover, as the learning of different slicing directions is amortised, the learning of each slicing direction also mutually helps each other, further reducing the required number of slices. As $M = 8$ achieves good performance in general, and is still affordable to run (compared to training a NDE), we recommend it as the default choice in practice.

Table 1. The execution time under different slice numbers.

	$M = 2$	$M = 4$	$M = 6$	$M = 8$	$M = 10$	NDE
JSD estimator	0.011	0.016	0.023	0.029	0.032	0.021
dCorr estimator	0.007	0.009	0.011	0.016	0.019	0.021

Table 2. Inference quality w.r.t. the number of slices when JSD estimator is used.

	$M = 2$	$M = 4$	$M = 6$	$M = 8$	$M = 10$	SNPE-C
g-and-k	1.62 ± 0.21	1.63 ± 0.22	1.73 ± 0.21	1.59 ± 0.19	1.58 ± 0.16	2.08 ± 0.33
Bayesian LR	2.50 ± 3.37	2.47 ± 2.91	2.01 ± 2.12	1.23 ± 1.01	1.25 ± 1.10	13.5 ± 3.85
Ricker's model	0.30 ± 0.14	0.26 ± 0.08	0.25 ± 0.11	0.26 ± 0.10	0.25 ± 0.11	0.41 ± 0.16
OU Process	0.95 ± 0.74	0.63 ± 0.26	0.47 ± 0.19	0.69 ± 0.35	0.68 ± 0.35	1.43 ± 0.46

Table 3. Inference quality w.r.t. the number of when dCorr estimator is used.

	$M = 2$	$M = 4$	$M = 6$	$M = 8$	$M = 10$	SNPE-C
g-and-k	1.68 ± 0.32	1.62 ± 0.23	1.71 ± 0.19	1.58 ± 0.12	1.61 ± 0.14	2.08 ± 0.33
Bayesian LR	1.67 ± 1.62	1.69 ± 1.39	1.06 ± 1.28	1.17 ± 1.18	0.70 ± 0.54	13.5 ± 3.85
Ricker's model	0.46 ± 0.24	0.38 ± 0.14	0.38 ± 0.26	0.42 ± 0.20	0.41 ± 0.19	0.41 ± 0.16
OU Process	0.83 ± 0.34	0.70 ± 0.32	0.67 ± 0.25	0.66 ± 0.27	0.72 ± 0.32	1.43 ± 0.46

Detail settings for the inference tasks. We summarise in Table 1 the prior and the true parameter for the models considered.

- *Multivariate g-and-k model*. The true parameter $\boldsymbol{\theta}^* = \{\{A_l^*, B_l^*, g_l^*, k_l^*\}_{l=1}^2, \rho^*\}$ is $\{\{3, 1, 2, 0.5\}, \{3, 1, 0.5, 0.5\}, 0.75\}$. The prior $\pi(\boldsymbol{\theta})$ is uniform: $A_l \sim \mathcal{U}(2.5, 3.5)$, $B_l \sim \mathcal{U}(0.5, 1.5)$, $g_l \sim \mathcal{U}(-0.2, 2)$, $k_l \sim \mathcal{U}(0.0, 1.0)$, $\rho \sim \mathcal{U}(-0.9, 0.9)$.

- *Bayesian linear regression*. The true parameter $\boldsymbol{\theta}^*$ is $\theta_l^* = 0.2 + l\Delta$ where $\Delta = 0.8/K$ where K is the dimensionality of $\boldsymbol{\theta}$. Here $K = 12$. The prior $\pi(\boldsymbol{\theta})$ is a factorised Gaussian: $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \mathbf{I})$.

- *Ricker model*. The true parameter $\boldsymbol{\theta}^* = \{3.8, 0.5, 10\}$. The prior $\pi(\boldsymbol{\theta})$ is $\theta_1 \sim \mathcal{U}(3, 5)$, $\theta_2 \sim \mathcal{U}(0.25, 0.8)$, $\theta_3 \sim \mathcal{U}(8, 11)$. The prior for θ_2 is carefully chosen so that particle filtering can approximate the likelihood well (Fasiolo et al., 2016).

- *OU process*. The true parameter $\boldsymbol{\theta}^* = \{\theta_1^*, \theta_2^*, \theta_3^*, \theta_4^*, \theta_5^*, \theta_6^*\}$ is $\{0.5, 1.0, 0.5, 0.5, 1.0, 0.5\}$. The prior is $\theta_1 \sim \mathcal{U}(0, 1)$, $\theta_4 \sim \mathcal{U}(0, 1)$, $\theta_2 \sim \mathcal{U}(-2, 2)$, $\theta_5 \sim \mathcal{U}(-2, 2)$, $\theta_3 \sim \mathcal{U}(0.1, 1.4)$, $\theta_6 \sim \mathcal{U}(0.1, 1.4)$.

- *Ising model*. The true parameter $\boldsymbol{\theta}^* = \{0.3, 0.1\}$ and the prior $\pi(\boldsymbol{\theta})$ here is uniform: $\theta_1 \sim \mathcal{U}(0, 1)$, $\theta_2 \sim \mathcal{U}(-0.2, 0.4)$.

Sampling from the unnormalised posterior. As mentioned in the main text, some LFI methods e.g. SNL and SNR does not support readily sampling from the estimated posterior. For controlled comparison, we also use the neural Gaussian copula proxy to support fast sampling in SNL/SNR, so that different methods only differ in the ways they infer the posterior.

It is worth noticing that when using the neural copula proxy $q'(\boldsymbol{\theta})$, due to the mode-seeking nature of the reverse KL objective used, the learned proxy $q'(\boldsymbol{\theta})$ is typically 'narrower' than the target distribution i.e. the estimated posterior $\hat{\pi}(\boldsymbol{\theta}|\mathbf{x}^o)$. This can be risky, as it can mis-guide simulation (e.g. focusing on a possibly wrong region of the posterior too early and fail to explore other areas of the posterior — a common issue in approximate sampling). We propose two simple ways to fix it.

- *Mixture proposal*. One simple way to fix this issue is to consider a mixture proposal between the prior $\pi(\boldsymbol{\theta})$ and $q'(\boldsymbol{\theta})$:

$$p(\boldsymbol{\theta}) = \lambda\pi(\boldsymbol{\theta}) + (1-\lambda)q'(\boldsymbol{\theta})$$

  where $\lambda \in [0,1]$ e.g. $\lambda = 0.5$. We find this simple strategy very robust, and we recommend it to be the default setting.

- *Inflated proposal*. Alternatively, we can consider an 'inflated' version of $q'(\boldsymbol{\theta})$ where we adjust each sample $\boldsymbol{\theta} \sim q'(\boldsymbol{\theta})$ as:

$$\theta_l = \mathbb{E}_{q'}[\theta_l] + \lambda \cdot (\theta_l - \mathbb{E}_{q'}[\theta_l])$$

  where $\lambda \geq 1$. This is equivalent to scale the variance $\mathbb{V}_{q'}[\boldsymbol{\theta}]$ of $q'(\boldsymbol{\theta})$ by a factor of λ^2, hence exploring more regions.

References

Fasiolo, M., Pya, N., and Wood, S. N. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. *Statistical Science*, pp. 96–118, 2016.

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lueckmann, J.-M., Boelts, J., Greenberg, D. S., Gonçalves, P. J., and Macke, J. H. Benchmarking simulation-based inference. *arXiv preprint arXiv:2101.04653*, 2021.

Nadjahi, K., Durmus, A., Chizat, L., Kolouri, S., Shahrampour, S., and Simsekli, U. Statistical and topological properties of sliced probability divergences. *Advances in Neural Information Processing Systems*, 33:20802–20812, 2020.

Papamakarios, G. and Murray, I. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pp. 1028–1036, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.