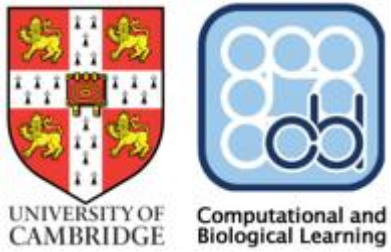


Is It Necessary to Learn Summary Statistics for LFI?

Yanzhi Chen[1], **Michael Gutmann**[2], **Adrian Weller**[1,3]

[1]Cambridge University, [2]University of Edinburgh, [3]Alan Turing Institute

Key message

1. In LFI, summary statistics may be easier to learn than the posterior itself

2. To learn sufficient statistics, we can maximise mutual information in projected spaces

3. End-to-end inference methods (e.g. SNPE-C) may be unreliable in some cases

Contents

- Background on LFI and summary statistics

- New method, slice sufficient statistics (S^3), for learning summary statistics in LFI

- Application to inference in implicit statistical models

Contents

- Background on LFI and summary statistics

- New method, slice sufficient statistics (S^3), for learning summary statistics in LFI

- Application to inference in implicit statistical models

Background

- **Likelihood-free inference (LFI)**

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

posterior prior likelihood

likelihood intractable, but we can still sample data from model

- **Summary statistics for LFI**

compressive representation ***S*** of data ***x*** such that

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

or equivalently[0, 1],

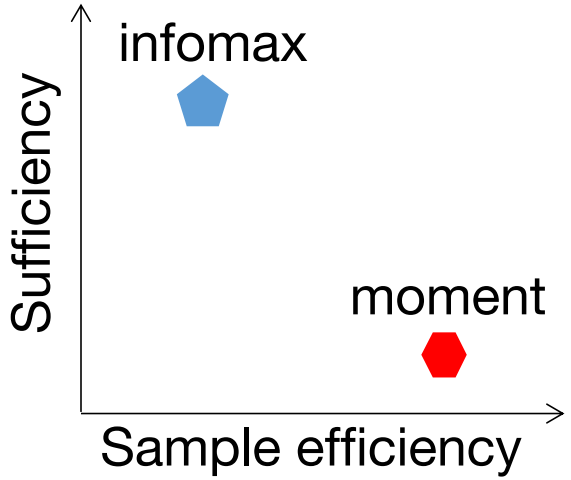
$$I(S(\mathbf{x});\boldsymbol{\theta}) \approx I(\mathbf{x};\boldsymbol{\theta})$$

i.e. representation of data that preserves all information about parameters

Methods for learning summary statistics

- **Moment statistics**[2]

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



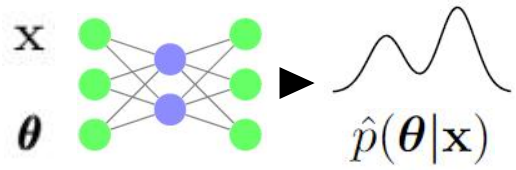
- **Infomax statistics**[1]

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

*other methods: score-matching[9, 12]; Fisher information maximization[11] auto-encoder[10]

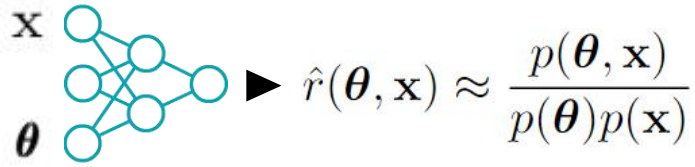
End-to-end inference in LFI

- **Sequential neural posterior estimate (SNPE)**[6, 7]



$\hat{p}(\boldsymbol{\theta}|\mathbf{x})$ neural density estimator, learned by MLE

- **Sequential neural ratio estimate (SNR)**[8, 13]



$\hat{r}(\boldsymbol{\theta}, \mathbf{x})$ neural ratio estimator, learned by contrasitive learning

*$\mathbf{x}$ directly fed to an encoder jointly trained with the neural posterior/ratio estimators

Necessity of learning summary statistics

We already have end-to-end inference algorithms

- seems like we no more need to learn summary statistics separately

Principles for learning summary statistics

- (a) its learning is easier than inference itself; (b) it is approximately sufficient

Contents

- Background on LFI and summary statistics

- New method for learning approximate sufficient statistics in LFI

- Application to inference in implicit statistical models

Overview

Main idea: learn $S = \arg\max_s I(s(\mathbf{x}); \boldsymbol{\theta})$ without estimating *I* explicitly

divide-and-conquer strategy[3]

$$\max_S I(S; \boldsymbol{\theta}) \iff \max_{S'_k} I(S'_k, \boldsymbol{\theta}'_k), \forall k$$

high-dim (hard)

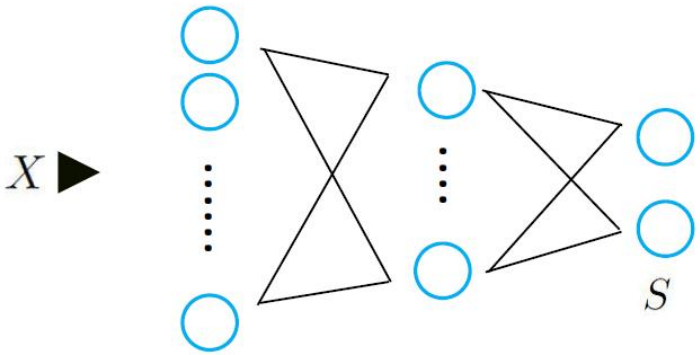
low-dim (easy)

*These low-dimensional MI is as easy to learn as 2D classification/metric learning

Method

Sliced infomax learning

$$\max_{S} I(S;\boldsymbol{\theta}) \iff \max_{S'_k} I(S'_k, \boldsymbol{\theta}'_k), \forall k$$

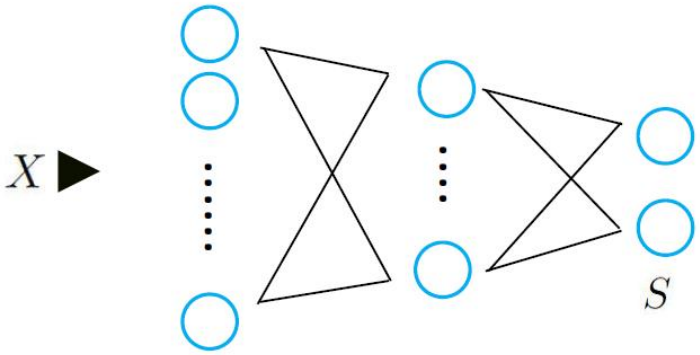


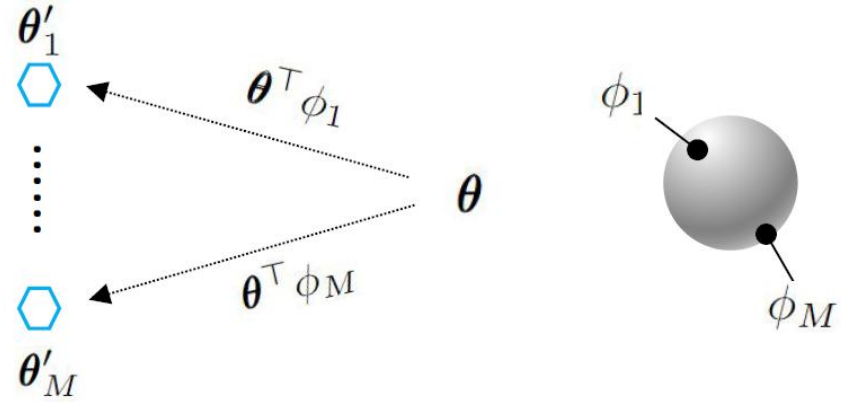
$\boldsymbol{\theta}$

Method

Sliced infomax learning

$$\max_{S} I(S;\boldsymbol{\theta}) \iff \max_{S'_k} I(S'_k, \boldsymbol{\theta}'_k), \forall k$$



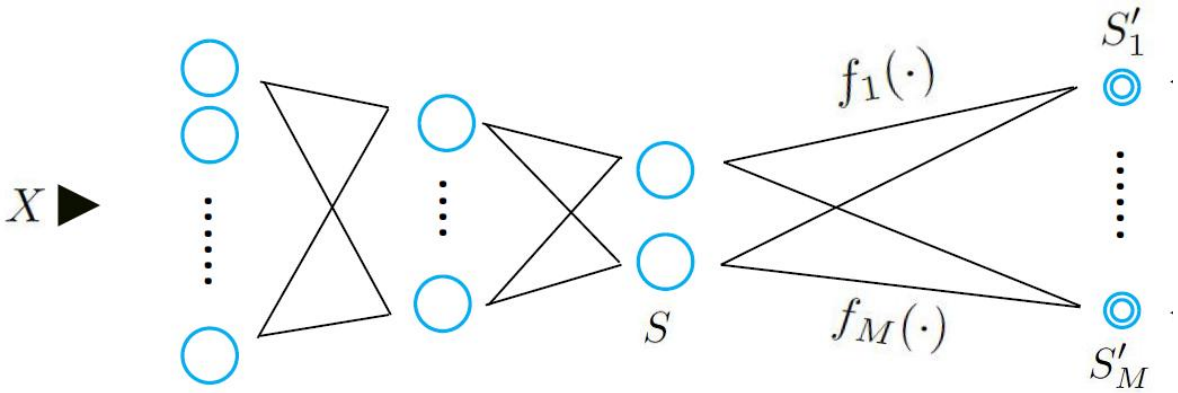


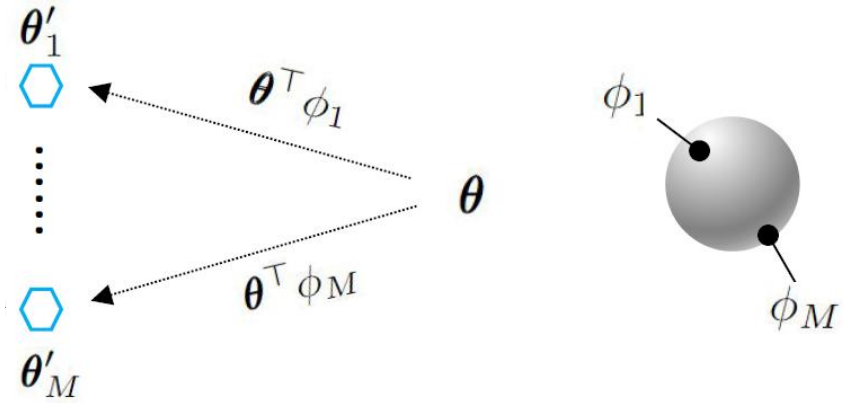
1. generating sliced version of $\boldsymbol{\theta}$

Method

Sliced infomax learning

$$\max_{S} I(S;\boldsymbol{\theta}) \iff \max_{S'_k} I(S'_k, \boldsymbol{\theta}'_k), \forall k$$



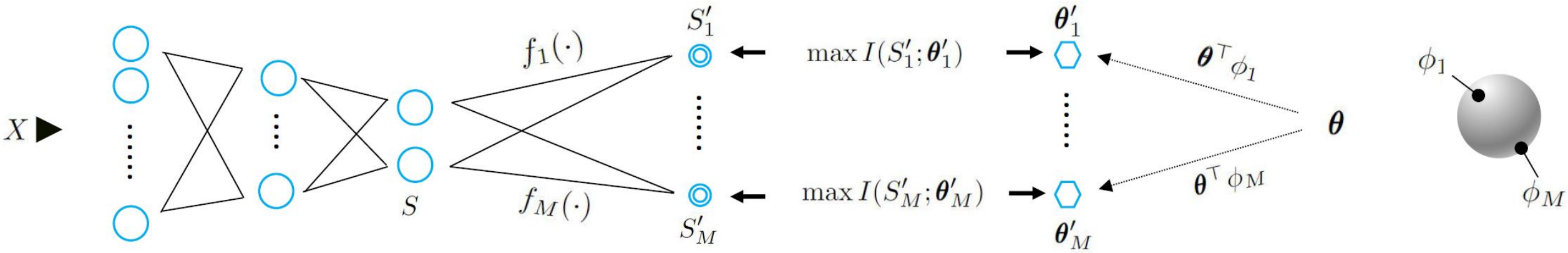


2. compute 2nd-level summary statistics S'_k

Method

Sliced infomax learning

$$\max_{S} I(S;\boldsymbol{\theta}) \iff \max_{S'_k} I(S'_k, \boldsymbol{\theta}'_k), \forall k$$



3. maximise low-dimensional MI

Theory

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. □

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

How to estimate/quantify MI

Estimating MI is still not so easy

We are interested in $\arg\max_{S'_k} I(S'_k(\mathbf{x}), \theta'_k)$ rather than knowing its exact value

Consider proxies to MI that have better properties

How to estimate/quantify MI

- **MI as distributional discrepancy between joint and marginal**

replacing KL with JSD[4]

$$\hat{I}(S'_i, \theta'_i) = \sup_{T_i : \mathbb{R} \times \mathbb{R}^{d'} \to \mathbb{R}} \mathbb{E}_{p(\theta'_i, S'_i)} \left[-\mathrm{sp}(-T_i(\theta'_i, S'_i)) \right] - \mathbb{E}_{p(\theta'_i)p(S'_i)} \left[\mathrm{sp}(T_i(\theta'_i, S'_i)) \right]$$

T: a neural network

*classifier problem to classify samples from joint vs marginals

How to estimate/quantify MI

- **MI as statistical dependence metric**

replacing MI with distance correlation[5]

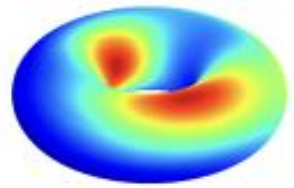
$$\hat{I}(S'_i;\theta'_i) = \frac{\mathbb{E}_{p(\theta'_i,S'_i)p(\tilde{\theta}'_i,\tilde{S}'_i)}[h(\theta'_i,\tilde{\theta}'_i)h(S'_i,\tilde{S}'_i)]}{\sqrt{\mathbb{E}_{p(\theta'_i)p(\tilde{\theta}'_i)}[h^2(\theta'_i,\tilde{\theta}'_i)]\mathbb{E}_{p(S'_i)p(\tilde{S}'_i)}[h^2(S'_i,\tilde{S}'_i)]}}$$

h: some distance function

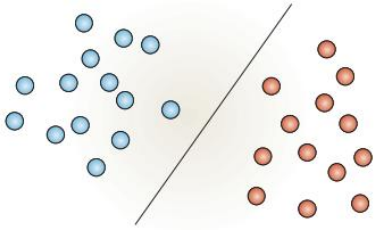
*metric learning problem where the pairwise distance of S'_i correlates with that of θ'_i

Recap

- Original inference problem
  high-dimensional density/ratio estimation (hard)



- Summary statistics learning problem
  low-dimensional classification/metric learning (easy)



Connection to other summary statistics

- **Slice statistics (ours)**

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

- **Moment statistics**[1,2,3]

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

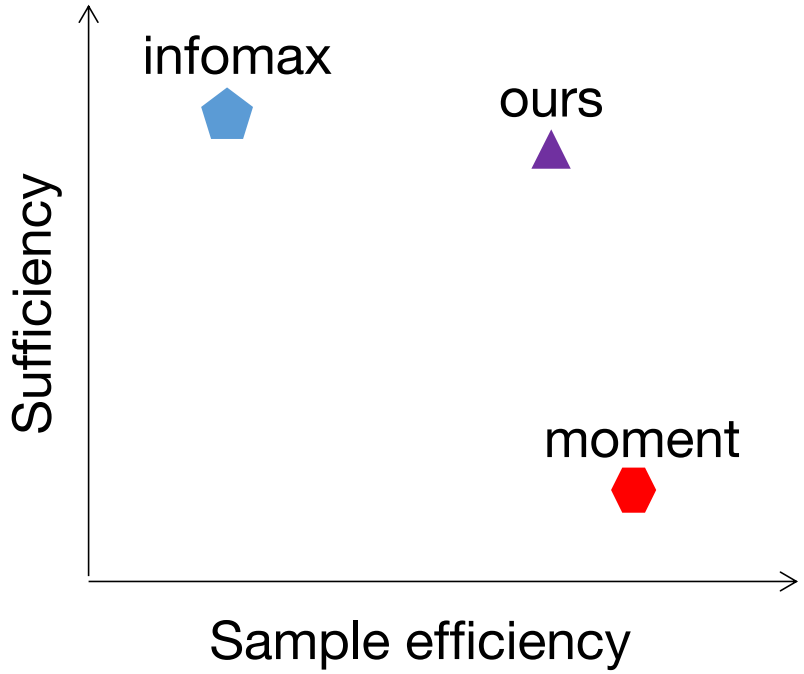
degenerated case of our method with only *K* one-hot slices

- **Infomax statistics**[4]

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

we recover the goal of this approach as #slices goes to infinity

Connection to other summary statistics



*we take the best from the two worlds

Inference

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 $\hat{\pi}(\boldsymbol{\theta}|S^o)$

← learn s.s

Contents

- Background on summary statistics in LFI

- New method for learning approximate sufficient statistics in LFI

- Application to inference in implicit statistical models

Inference tasks

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^*

*true posterior either known or can be approximated up to very high precision

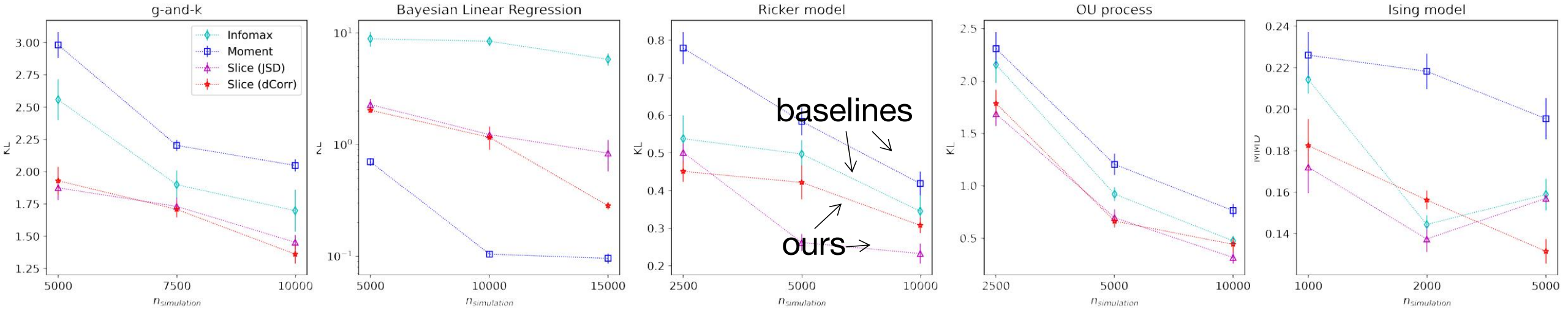
Baselines

- **VS. other summary statistics**
  - moment statistics
  - infomax statistics

- **VS. end-to-end inference algorithms**
  - SNPE-C
  - SNR

Comparison to other summary statistics

All methods use SNL in inference



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

x-axis: simulation budget

y-axis: discrepancy(true, learned)

Comparison to other end-to-end inference method

	g-and-k	Bayesian LR	Ricker model	OU Process	Ising model
SNL + SSS (ours)	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 (end-to-end)	2.082 ± 0.325	13.45 ± 3.846	0.413 ± 0.156	1.428 ± 0.457	0.152 ± 0.055
SNR (end-to-end)	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

*digits are discrepancy (true, learned posterior)

Take-aways

- Summary statistics may be easier to learn than the posterior itself

- End-to-end inference strategies can occationally be less accurate in high-dimensional cases

- Rethink what objects are easier to learn (likelihood, posterior, ratio, statistics, score) in LFI

- Infomax representation learning can be done in low-dimensional projected spaces.

Reference

[0]. Learning and generalization with the information bottleneck. Theoretical Computer Science

[1]. Neural approximate sufficient statistics for implicit models, ICLR 2021

[2]. Constructing summary statistics for approximate Bayesian computation, JRSS-B 2009

[3]. Sliced mutual information, NeurIPS 2021

[4]. Learning deep representations by mutual information estimation and maximization, ICLR, 2019

[5]. Partial distance correlation with methods for dissimilarities, Annals of Statistics, 2014

[6]. Fast ε-free Inference of Simulation Models with Bayesian Conditional Density Estimation, NeurIPS 2016

[7]. Flexible statistical inference for mechanistic models of neural dynamics, NeurIPS 2017

[8]. Automatic Posterior Transformation for Likelihood-free Inference, ICML 2020

Reference

[9]. Score matched neural exponential families for likelihood-free inference, JMLR 2022

[10]. Learning Summary Statistics for Bayesian Inference with Autoencoders, arxiv 2201.12059

[11]. Massive optimal data compression and density estimation for scalable, likelihood-free inference in cosmology, MNRAS 2018

[12]. Mining gold from implicit models to improve likelihood-free inference, PNAS 2020

[13]. Likelihood-free MCMC with Amortized Approximate Ratio Estimators, ICML 2020

[14]. Markov chain Monte Carlo without likelihoods, PNAS 2003

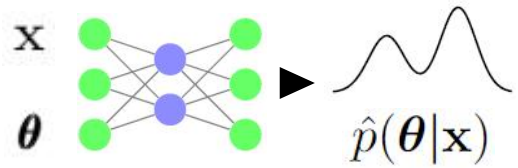
[15]. Telescoping density-ratio estimation, Neurips 2020

[16]. Density ratio estimation via infinitesimal classification, AISTATS 2022

[17]. Estimating the Density Ratio between Distributions with High Discrepancy using Multinomial Logistic Regression, TMLR 2023

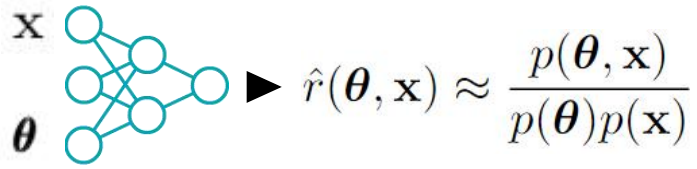
Appendix - Pitfall of end-to-end inference

- **Sequential neural posterior estimate (SNPE)**[6,7]



less compatible to sequential learning[8]

- **Sequential neural ratio estimate (SNR)**[8,13]

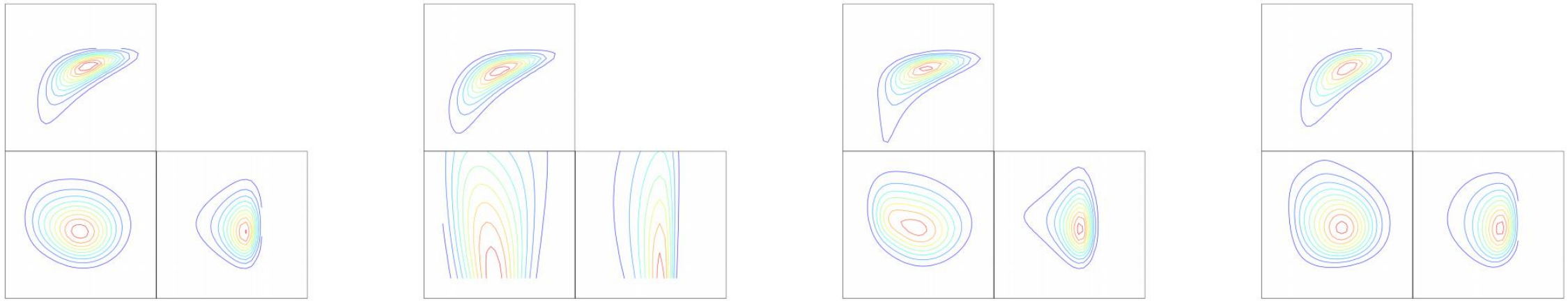


not so reliable if $p(\boldsymbol{\theta})p(\mathbf{x})$ and $p(\boldsymbol{\theta}, \mathbf{x})$ too distinct[15, 16, 17]

Appendix - Neural Copula Proxy

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

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



(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).

Appendix - Determining the dimensionality *d*

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* j*th dimension. We then have*

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

like PCA, each dimension in the learned *S* is *sorted*

▼

determine the optimal *d* by inspecting information loss