Mining gold from implicit models to improve likelihood-free inference

Johann Brehmer,[1] **Gilles Louppe,**[2] **Juan Pavez,**[3] **and Kyle Cranmer**[1]
[1] New York University, [2] University of Liège, [3] Federico Santa María Technical University
johann.brehmer@nyu.edu, g.louppe@uliege.be,
juan.pavezs@alumnos.usm.cl, kyle.cranmer@nyu.edu

Abstract

Simulators often provide the best description of real-world phenomena; however, they also lead to challenging inverse problems because the density they implicitly define is often intractable. We present a new suite of simulation-based inference techniques that go beyond the traditional Approximate Bayesian Computation approach, which struggles in a high-dimensional setting, and extend methods that use surrogate models based on neural networks. We show that additional information, such as the joint likelihood ratio and the joint score, can often be extracted from simulators and used to augment the training data for these surrogate models. Finally, we demonstrate that these new techniques are more sample efficient and provide higher-fidelity inference than traditional methods.

1 Introduction

In many areas of science, complicated real-world phenomena are best described through computer simulations. Typically, the simulators implement a stochastic generative process in the "forward" mode based on a well-motivated mechanistic model parameterized by θ. While the simulators can generate samples of observations $x \sim p(x|\theta)$, they typically do not admit a tractable likelihood (or density) $p(x|\theta)$. Probabilistic models defined only via the samples they produce are often called implicit models. Implicit models lead to intractable inverse problems, which is a barrier for statistical inference of the parameters θ given observed data. These problems arise in fields as diverse as particle physics, epidemiology, and population genetics, which has motivated the development of *likelihood-free inference* algorithms such as Approximate Bayesian Computation (ABC) [1, 2] and neural density estimation (NDE) techniques [3–20].

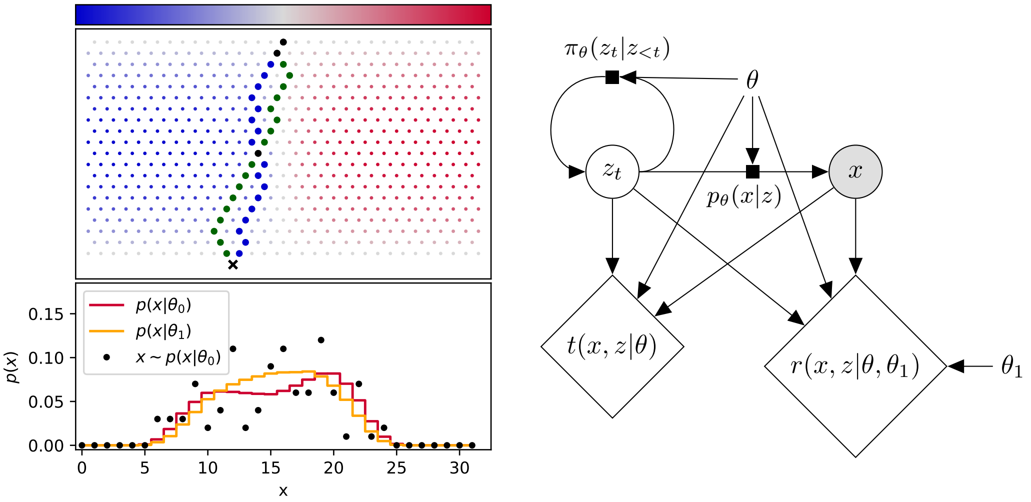
We present a suite of new techniques for likelihood-free inference, and introduce a broader class of simulation-based inference problems where additional information can be extracted from the simulator. This augmented data can be used to train neural network surrogates that estimate the likelihood ratio $r(x|\theta_0, \theta_1) = p(x|\theta_0)/p(x|\theta_1)$. This provides the key quantity needed for both frequentist and Bayesian inference procedures. The resulting methods provide a significant increase in sample efficiency and quality of the resulting inference compared to previous techniques.

1.1 Motivating example and key observation

Typically the setting of likelihood-free inference methods assumes that the only available output from the simulator are samples of observations $x \sim p(x|\theta)$. However, we will show that in many cases additional information can be extracted from the simulator, even though the implicit density is intractable. Let the generative process be characterized by some set of latent variables z such that

$$p(x|\theta) = \int \mathrm{d}z \; p(x, z|\theta) \,. \tag{1}$$

Preprint. Work in progress.



(a) Toy Simulator

(b) Graphical Model Abstraction

Figure 1: (a) A toy simulation generalizing the Galton board where the transitions are biased left (blue) or right (red) depending on the nail position and the value of at θ. Two example latent trajectories z are shown (blue and green) leading to the same observed value of x. Below, the distribution for $\theta_0 = -0.8$ and $\theta_1 = -0.6$ (orange and red histograms). Finally, an example empirical distribution from 100 runs of the simulator with θ_0 shows that the sample variance is much larger than the differences from θ_0 vs. θ_1. (b) A graphical model representation of a density defined implicitly by a stochastic simulator where the latent state z_t evolves sequentially according to a policy $p(z_t|z_{<t}, \theta)$ and final observation model $p(x|z, \theta)$. The joint score $t(x, z|\theta)$ and joint ratio $r(x, z|\theta, \theta_1)$ are tractable deterministic functions that can be extracted from the simulator code.

Often the likelihood is intractable exactly because the latent space z is enormous and it is infeasible to explicitly calculate this integral. In real-world scientific simulators, the trajectory for a single observation can involve many millions of latent variables.

As a motivating example, consider the simulation for a generalization of the Galton board, in which a set of balls is dropped through a lattice of nails ending in one of several bins denoted by x. The Galton board is commonly used to demonstrate the central limit theorem, and if the nails are uniformly placed such that the probability of bouncing to the left is p, the sum over the latent space is tractable analytically and the resulting distribution of x is a binomial distribution with N_{rows} trials and probability p of success. However, if the nails are not uniformly placed, and the probability of bouncing to the left is an arbitrary function of the nail position and some parameter θ, the resulting distribution requires an explicit sum over the latent paths z that might lead to a particular x. Such a distribution would become intractable as N_{rows}, the size of the lattice of nails, increases. Figure 1a shows an example of two latent trajectories that lead to the same x. In this toy example, the probability $p(z_h, z_v, \theta)$ of going left is given by $(1 - f(z_v))/2 + f(z_v)\sigma(5\theta(z_h - 1/2))$, where $f(z_v) = \sin(\pi z_v)$, σ is the sigmoid function, and z_h and z_v are the horizontal and vertical nail positions normalized to $[0, 1]$. This leads to a non-trivial $p(x|\theta)$, which can even be bimodal. Code for simulation and inference in this problem is available at Ref. [21].

While $p(x|\theta)$ is intractable, the *joint score*

$$t(x, z|\theta_0) \equiv \nabla_\theta \log p(x, z|\theta)\Big|_{\theta_0} . \tag{2}$$

can be computed by accumulating the factors $\nabla_\theta \log p(z_h, z_v|\theta)$ as the simulation runs forward through its control flow conditioned on the random trajectory z. A similar trick can be applied to extract the *joint likelihood ratio*

$$r(x, z|\theta_0, \theta_1) \equiv \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} . \tag{3}$$

Figure 1a shows that a large number of samples from the simulator are needed to reveal the differences in the distribution of x for small changes in θ – the number of samples needed grows like $(p/\Delta p)^2$. Moreover, this toy simulation is representative of many real-world simulators in that it is composed of non-differentiable control-flow elements. This poses a difficulty, making methods based on $\nabla_z x$ [22] and $\nabla_\theta x$ inapplicable, which previously motivated techniques such as AVO [18].

Figure 1b presents a graphical model that abstracts the simulation as a probabilistic sequence of latent states z_t. The mechanistic model implemented by the simulator describes a particular probabilistic transition $\pi_\theta(z_t|z_{<t})$, which can be viewed as a parameterized policy for taking the action z_t given the past $z_{<t}$. Finally, the simulation emits a sampled observation based on $p_\theta(x|z)$. While $p(x|\theta)$ is intractable due to the integration over the latent space, it is possible to calculate how much more or less likely a particular trajectory through the simulator would be if one changed θ. Moreover, this relative change can efficiently be accumulated as the simulation transitions from $z_t \to z_{t+1}$. This is essentially the same observation as the policy gradient used in REINFORCE [23]; however, instead of trying to optimize θ via a stochastic gradient estimate of some reward function, we will simply augment the data generated by the simulator with the joint score. Similarly, there is a large class of problems in which one can extract the joint likelihood ratio from the simulator even though $p(x|\theta)$ is intractable.

In Sec. 4 we will show how this simulated data, augmented with the joint score and joint likelihood ratio, can be used in a regression setting to estimate the intractable likelihood ratio $r(x|\theta_0, \theta_1) \equiv p(x|\theta_0)/p(x|\theta_1)$ and the intractable score function

$$t(x|\theta_0) \equiv \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} . \tag{4}$$

This score vector provides sufficient statistics, which fully characterizes the implicit model in some local neighborhood around θ_0, as we will discuss in more detail in Sec. 5.

2 Related work

Techniques for simulator-based inference can divided into two broad categories. The first category uses the simulator directly during inference, while the second uses the simulator to construct or train a tractable surrogate model that is used during inference. We also find strong connections between simulator-based inference and learning in implicit generative models [24], such as GANs, with a considerable amount of cross-pollination between these areas.

Approximate Bayesian Computation (ABC). A particularly ubiquitous method is Approximate Bayesian Computation [1, 2], a Bayesian sampling technique in which the likelihood is approximated by comparing data generated from the simulator to the observed data. This approach requires introducing a kernel $K_\epsilon(x, x_{\text{obs}})$, which defines a notion of distance between the simulated data x and the observed data x_{obs}. This is an approximate inference method that is exact in the limit $\epsilon \to 0$. It scales poorly when x is high-dimensional, thus much of the research in ABC is focused on finding appropriate summary statistics. Relevant works include classifier ABC [16], that relies on a classifier to estimate the discrepancy between the observed data the model distributions, and Hamiltonian ABC [25], which makes use of finite differences through the simulator to estimates gradients with respect to the θ. Reference [22] introduces an ϵ-free exact inference approach, but it is restricted to differentiable generative models.

Probabilistic programming (PPS). Probabilistic programming systems represent another class of methods that use the simulator directly during inference [26, 27]. These techniques are deeply integrated into the control flow of the program, but still require a tractable likelihood term or ABC-like kernel to compare the the simulated data x and the observed data x_{obs}. While our work will not utilize probabilistic programming, there is commonality in the notion of a non-standard interpretation of the simulator code to produce a non-standard output (i. e. the joint score and joint likelihood ratio).

The likelihood ratio trick (LRT). In the most general likelihood-free inference problem, the only quantity available from the simulator are samples of observables $x \sim p(x|\theta)$ for given parameters θ, from which we aim to estimate ratios of likelihoods. In this setting we can train a probabilistic classifier with decision function $\hat{s}(x)$ to discriminate between two equal-sized samples $\{x_i\} \sim p(x|\theta_0)$ and $\{x_i\} \sim p(x|\theta_1)$. The binary cross-entropy loss

$$L_{\text{XE}} = -\mathbb{E}_{p(x|\theta)\pi(\theta)} \left[\mathbb{1}(\theta = \theta_1) \log \hat{s}(x|\theta_0, \theta_1) + \mathbb{1}(\theta = \theta_0) \log(1 - \hat{s}(x|\theta_0, \theta_1)) \right] \tag{5}$$

is then minimized by the optimal decision function

$$s^*(x|\theta_0,\theta_1) = \frac{p(x|\theta_1)}{p(x|\theta_0) + p(x|\theta_1)} \qquad \Leftrightarrow \qquad r^*(x|\theta_0,\theta_1) \equiv \frac{p(x|\theta_0)}{p(x|\theta_1)} = \frac{1 - s(x|\theta_0,\theta_1)}{s(x|\theta_0,\theta_1)} \,. \tag{6}$$

We can therefore estimate the likelihood ratio between θ_0 and θ_1 by training a sufficiently expressive classifier to discriminate samples generated according to the two parameters. If it is trained well, the likelihood ratio can be estimated from the decision function $\hat{s}(x)$ as

$$\hat{r}(x|\theta_0,\theta_1) = \frac{1 - \hat{s}(x|\theta_0,\theta_1)}{\hat{s}(x|\theta_0,\theta_1)} \,. \tag{7}$$

This “likelihood ratio trick” or “density ratio trick” is widely appreciated [6, 17, 24, 28]. In Sec. 3, we discuss two new ideas introduced in Ref. [6, 29] that substantially expand and improve this approach to likelihood ratio estimation.

Neural density estimation (NDE). More recently, several methods for conditional density estimation have been proposed, often based on neural networks [3–5, 8–15, 17–20]. These provide flexible models for $\hat{p}(x|\theta)$, and training by maximum likelihood (minimizing the negative log likelihood)

$$L_{\text{MLE}} = -\mathbb{E}_{p(x|\theta)} \left[\log \hat{p}(x|\theta)\right] \tag{8}$$

with a sufficiently flexible density estimator will approximate $p^*(x|\theta) = p(x|\theta)$.

Novel contributions. The most important novel contribution that differentiates our work from the existing methods is the observation that additional information can be extracted from the simulator, and that this “augmented” data can dramatically improve sample efficiency and quality of likelihood-free inference. We playfully introduce the analogy of mining gold as this augmented data requires work to extract and is very valuable. This is the starting point of the techniques presented in Secs. 4 and 5. Concurrently, application of these methods to a specific class of problems in particle physics has been discussed in Refs. [30, 31]. This manuscript is meant to serve as the primary reference for these new techniques. It is addressed to the broader community and requires weaker assumptions than those made in the physics context. We also introduce a new variant called SCANDAL, for which we provide the first experimental results in the context of a generalized Galton board simulator.

In addition to the new approaches, we highlight two important ingredients of the CARL technique, originally presented in Refs. [6, 29]. This algorithm is based on the likelihood ratio trick but goes beyond this basic idea by introducing an additional calibration step and the use of parameterized estimators that learn the dependency of the likelihood ratio on the model parameters θ. The new approaches described in this work can also make use of calibration and parameterized estimators. We discuss these elements before moving to the new techniques that leverage the augmented data.

3 Calibrated discriminative classifiers

3.1 Calibration

In practice, not all probabilistic classifiers trained to separate samples from θ_0 and θ_1 learn the decision function given in Eq. (6). This may be due to using a different loss function, finite training sample size, insufficient capacity, or inefficiencies during training. Nevertheless, as long as the classifier decision function is a monotonic function of the likelihood ratio, we can use it to define a precise estimator for the likelihood ratio. This requires an additional calibration step, transforming a likelihood ratio estimator $\hat{r}_{\text{raw}}$ defined through Eq. (7) to

$$\hat{r}_{\text{cal}} = \frac{\hat{p}(\hat{r}_{\text{raw}}|\theta_0)}{\hat{p}(\hat{r}_{\text{raw}}|\theta_1)} \,. \tag{9}$$

Here the densities $\hat{p}(\hat{r}_{\text{raw}}|\theta)$ are estimated through univariate density estimation such as histograms or kernel density estimation. We use the term CARL to describe likelihood ratio estimators based on classifiers with a subsequent calibration.

3.2 Parameterized estimators

The simulators we consider in this work do not only implicitly define a single density $p(x)$, but a family of densities $p(x|\theta)$. The parameters θ may potentially belong to a high-dimensional parameter space. For the class of methods that attempt to learn a surrogate for the purpose of inference there are two broad strategies. The first is to estimate $p(x|\theta)$ or the likelihood ratio $r(x|\theta_0, \theta_1)$ for specific values of θ or pairs (θ_0, θ_1). This may be done via a pre-defined set of θ values or on-demand using an active-learning iteration.

We propose a different approach, in which one trains a *parameterized estimator* for the full model $\hat{r}(x|\theta_0, \theta_1)$ as a function of both the observables x and the parameters (θ_0, θ_1) [6, 32]. The training data then consists of a number of subsamples, each generated with different values of θ_0 and θ_1, and the parameter values are used as additional inputs to the discriminative classifier. Alternatively, the reference hypothesis θ_1 in the denominator of the likelihood ratio can be kept at a fixed reference value (or a composite hypothesis, integrated over different values of θ_1 with some prior $\pi(\theta_1)$), and only the θ_0 dependence is modeled by the network. This approach encourages the estimator to learn the typically smooth dependence of the likelihood ratio on the parameters of interest from the training data and borrow power from neighboring training data.

4 Learning from augmented data

Now we transition to the setting in which the simulator provides not only observations x, but also the joint likelihood ratio $r(x, z|\theta_0, \theta_1)$ and the joint score $t(x, z|\theta_0)$, both conditional on the unobservable latent variables z. How can this "augmented data" be used to estimate the likelihood ratio $r(x|\theta_0, \theta_1)$? The integral of the ratio is not the ratio of the integrals! Similarly, how can the joint score be used to estimate the score $t(x|\theta_0)$? The integral of the log is not the log of the integral!

Consider the squared error of a function $\hat{g}(x)$ that only depends on the observable x, but is trying to approximate a function $g(x, z)$ that also depends on the latent variable z,

$$L_{\text{MSE}} = \mathbb{E}_{p(x,z|\theta)} \left[\, (g(x, z) - \hat{g}(x))^2 \, \right] . \tag{10}$$

The minimum-mean-square-error prediction of g is given by the conditional expectation

$$g^*(x) = \mathbb{E}_{p(z|x,\theta)} \left[\, g(x, z) \, \right] . \tag{11}$$

Identifying $g(x, z)$ with the joint likelihood ratio $r(x, z|\theta_0, \theta_1)$ and $\theta = \theta_1$, we define

$$L_r = \mathbb{E}_{p(x,z|\theta_1)} \left[\, (r(x, z|\theta_0, \theta_1) - \hat{r}(x))^2 \, \right] \, , \tag{12}$$

which is minimized by

$$r^*(x) = \mathbb{E}_{p(z|x,\theta_1)} \left[\, r(x, z|\theta_0, \theta_1) \, \right] = r(x|\theta_0, \theta_1). \tag{13}$$

Similarly, by identifying $g(x, z)$ with the joint score $t(x, z|\theta_0)$ and setting $\theta = \theta_0$, we define

$$L_t = \mathbb{E}_{p(x,z|\theta_0)} \left[\, \left(t(x, z|\theta_0) - \hat{t}(x|\theta_0) \right)^2 \, \right] \, , \tag{14}$$

which is minimized by

$$t^*(x) = \mathbb{E}_{p(z|x,\theta_0)} \left[\, t(x, z|\theta_0) \, \right] = t(x|\theta_0) \, . \tag{15}$$

These loss functions are useful because they allow us to transform $t(x, z|\theta_0) \to t(x|\theta_0)$ and $r(x, z|\theta_0, \theta_1) \to r(x|\theta_0, \theta_1)$. That is, we are able to regress on these two intractable quantities! This is what makes the joint score and joint likelihood ratio the gold worth mining.

The augmented data is particularly powerful for enhancing the power of simulation-based inference for small changes in the parameter θ. When restricted to samples $x \sim p(x|\theta)$ the variance from the simulator is a challenge. The fluctuations in the empirical density scale with the square root of the number of samples, thus large numbers of samples are required before small changes in the implicit density can faithfully be distinguished. In contrast, each sample of the joint ratio and joint score provides an exact piece of information even for arbitrarily small changes in θ.

Table 1 summarizes six new approaches to simulator-based inference that leverage the augmented data and loss functions described above.

Table 1: A summary of simulator-based inference strategies including the traditional ABC method and approaches that use neural networks to learn a surrogate for amortized likelihood-free inference. Approaches based on neural density estimation and CARL only make use of the samples $x \sim p(x|\theta)$, while the six new methods leverage the augmented data and the loss functions L_r and L_t.

Method	L_{XE}	L_{MLE}	L_r	L_t	θ sampling
ABC (**A**pproximate **B**ayesian **C**omputation)					$\theta \sim \pi(\theta)$
NDE (**N**eural **d**ensity **e**stimation)		✓			$\theta \sim \pi(\theta)$
LRT / CARL (**L**ikelihood **r**atio **t**rick / **c**alibrated **a**pproximate **r**atios of **l**ikelihoods)	✓				$\theta \sim \pi(\theta)$
ROLR (**R**egression **o**n **l**ikelihood **r**atio)			✓		$\theta \sim \pi(\theta)$
SCANDAL (**Sc**ore **a**ugmented **n**eural **d**ensity **a**pproximates **l**ikelihood)		✓		✓	$\theta \sim \pi(\theta)$
CASCAL (**Ca**RL **and** **sc**ore **a**pproximate **l**ikelihood ratio).	✓			✓	$\theta \sim \pi(\theta)$
RASCAL (**Ra**tio **and** **sc**ore **a**pproximate **l**ikelihood ratio)			✓	✓	$\theta \sim \pi(\theta)$
SALLY (**S**core **a**pproximates **l**ikelihood **l**ocall**y**)				✓	$\theta = \theta_0$
SALLINO (**S**core **a**pproximates **l**ikelihood **l**ocally **in** **o**ne dimension)				✓	$\theta = \theta_0$

5 Locally sufficient statistics for implicit models

We can go further by considering an expansion of the implicit model around a reference point θ_{ref}. Up to linear order in $\theta - \theta_{\text{ref}}$, we find

$$p_{\text{local}}(x|\theta) = \frac{1}{Z(\theta)}\, p(t(x|\theta_{\text{ref}}) \,|\, \theta_{\text{ref}}) \, \exp[t(x|\theta_{\text{ref}}) \cdot (\theta - \theta_{\text{ref}})] \tag{16}$$

with some normalization factor $Z(\theta)$. This local approximation is in the exponential family and the score vector $t(x|\theta_{\text{ref}})$, defined in Eq. (4), are its sufficient statistics. For inference in a sufficiently small neighborhood around a reference point θ_{ref}, a precise estimator of the score $\hat{t}(x|\theta_{\text{ref}})$ therefore defines a vector of ideal summary statistics that contain all the information in an observation x on the parameters θ. Amazingly, the joint score together with Eqs. (14) and (15) allow us to extract sufficient statistics from an intractable, non-differentiable simulator, at least in the neighborhood of θ_{ref}. Moreover, this local model can be estimated by running the simulator at a single value θ_{ref} — it does not require scanning the θ space, which would suffer from the curse of dimensionality.

Based on this observation, we introduce the SALLY estimator for the likelihood ratio. By minimizing the squared error with respect to the joint score, see Eq. (14), we train a score estimator $\hat{t}(x|\theta_{\text{ref}})$. In a next step, we estimate the density $\hat{p}(\hat{t}(x|\theta_{\text{ref}})|\theta)$ through standard density estimation techniques, defining the likelihood ratio estimator $\hat{r}(x|\theta_0, \theta_1) = \hat{p}(\hat{t}(x|\theta_{\text{ref}})|\theta_0)/\hat{p}(\hat{t}(x|\theta_{\text{ref}})|\theta_1)$. This calibration procedure implicitly includes the effect of the normalizing constant $Z(\theta)$.

This inference method requires density estimation in the estimated score space, with typically $\dim \hat{t} \equiv \dim \theta \ll \dim x$. But in cases with large number of parameters, it is beneficial to reduce the dimensionality even further. In the local model of Eq. (16) the likelihood ratio $r(x|\theta_0, \theta_1)$ only depends on the scalar product between the score and $(\theta_0 - \theta_1)$ up to an x-independent constant related to $Z(\theta)$. Thus, given a score estimator $\hat{t}(x|\theta_{\text{ref}})$, we can define the scalar function $\hat{h}(x|\theta_0, \theta_1) \equiv \hat{t}(x|\theta_{\text{ref}}) \cdot (\theta_0 - \theta_1)$. In the local approximation and assuming a precise estimator $\hat{t}(x|\theta_{\text{ref}})$, this scalar is a sufficient statistic for the 1-dimensional parameter space connecting θ_0 and θ_1. This motivates the SALLINO technique: again, a neural network is trained on the joint score data from the simulator to estimate the score $\hat{t}(x|\theta_{\text{ref}})$. The likelihood ratio is then estimated through univariate density estimation on $\hat{h}$ as $\hat{r}(x|\theta_0, \theta_1) = \hat{p}(\hat{h}(x|\theta_0, \theta_1)|\theta_0)/\hat{p}(\hat{h}(x|\theta_0, \theta_1)|\theta_1)$.

The SALLY and SALLINO techniques are designed to work very well close to the reference point. The local model approximation may deteriorate further away, leading to a reduced sensitivity and weaker bounds. These approaches are simple and robust, and in particular the SALLINO algorithm scales exceptionally well to high-dimensional parameter spaces.

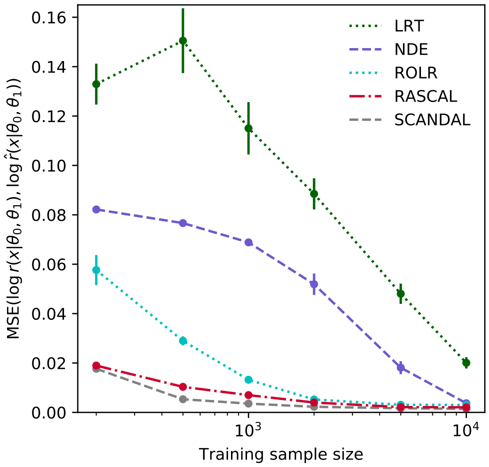


Figure 2: Galton board example. MSE on $\log r$ vs. training sample size. We show the mean and its error based on 15 runs.

6 Experiments

6.1 Generalized Galton board

We return to the motivating example in Sec. 1.1 and Fig. 1a. First, we compare neural density estimation (NDE) to the score-augmented neural density estimator SCANDAL. As the simulator defines a distribution over a discrete x, we use a neural network with a softmax output layer over the bins to model $\hat{p}(x|\theta)$. The network is explicitly parameterized in terms of θ, the parameter of the simulator that defines the position of the nails (i. e. it takes θ as an input). We use a simple network architecture with a single hidden layer, 10 hidden units, and $\tanh$ activations. Figure 2 shows the mean squared error between $\log \hat{r}(x|\theta_0, \theta_1)$ and the true $\log r(x|\theta_0, \theta_1)$ (estimated from histograms of $2 \cdot 10^4$ simulations from $\theta_0 = -0.8$ and $\theta_1 = -0.6$), summing over $x \in [5, 15]$, versus the training sample size. The training sample size refers to the total number of x samples, distributed over 10 values of $\theta \in [-1, -0.4]$. Fig. 2 shows that both SCANDAL and RASCAL are dramatically more sample efficient than pure neural density estimation and the likelihood ratio trick, which do not leverage the joint score. ROLR improves upon pure neural density estimation and achieves the same asymptotic error as SCANDAL, though more slowly.

6.2 Particle physics

Our second example is a real-world problem from particle physics. A simulator describes the production of a Higgs boson at the Large Hadron Collider experiments in the “weak boson fusion” production mode, followed by the decay into four electrons or muons, subsequent radiation patterns, the interaction with the detector elements, and the reconstruction procedure. Each recorded collision produces a single high-dimensional observable $x \in \mathbb{R}^{42}$, and the dataset consists of multiple iid observations of x. The goal is to infer confidence limits on two parameters $\theta \in [-1, 1]^2$ that characterize the effect of high-energy physics models on these interactions. We consider a synthetic observed dataset with 36 iid simulated observations of x drawn from $\theta = (0, 0)$.

The new inference techniques can accommodate state-of-the-art simulators, but in that setting we cannot compare them to the true likelihood function. We therefore work in a simplified setup and approximate the detector response such that the true likelihood function is tractable, providing us with a ground truth to compare the inference techniques to. As simulator we use a combination of MADGRAPH 5 [33] and MADMAX [34–36]. The setup and the results of this experiment are described at length in Ref. [31].

We are able to extract the joint score and joint likelihood ratio from the simulation and we test the sample efficiency and the quality of the inference for all of the new techniques except for SCANDAL.

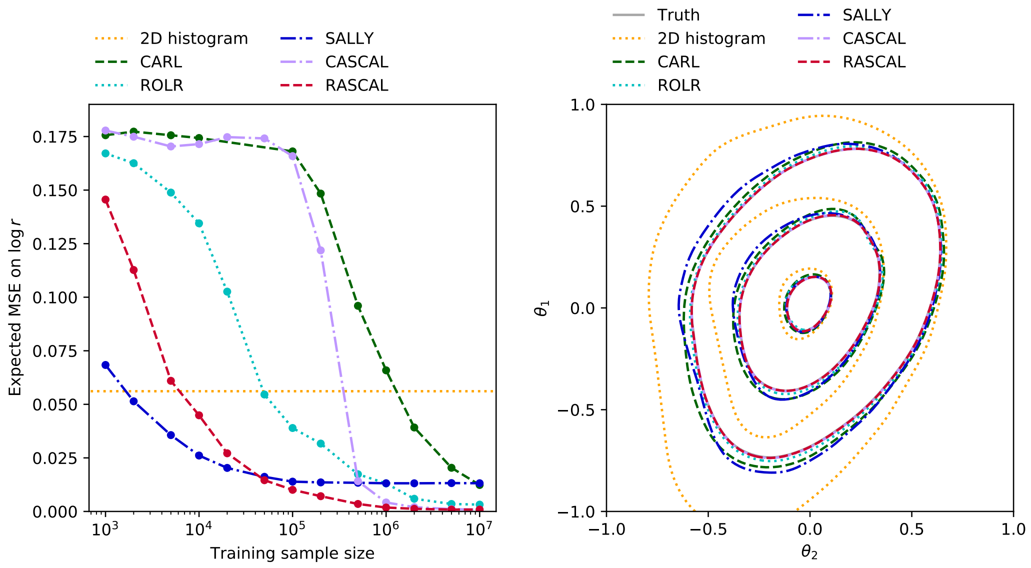


Figure 3: Particle physics example. Left: mean squared error as a function of the training sample size. Right: expected 68% / 95% / 99.7% confidence intervals for θ, including ground truth (grey).

In the left panel of Fig. 3 we show the expected mean squared error of the approximate $\log \hat{r}(x|\theta_0, \theta_1)$ as a function of the training sample size. We take the expectation over random values of θ_0, drawn from a Gaussian prior with mean $(0, 0)$ and covariance matrix $\text{diag}(0.2^2, 0.2^2)$. We compare the new techniques to the traditional inference method in particle physics: estimating densities through histograms, using two established kinematic variables as summary statistics (similar to ABC).

All new inference techniques outperform the traditional histogram method, provided that the training samples are sufficiently large. Using augmented data substantially decreases the amount of training data required for a good performance: the RASCAL method, which uses both the joint ratio and joint score information from the simulator, reduces the amount of training data by two orders of magnitude compared to the CARL technique, which uses only the samples $x \sim p(x|\theta)$. The particularly simple local techniques SALLY and SALLINO need even less data for a good performance. However, their performance eventually plateaus and does not asymptote to zero error. This is because the local model approximation breaks down further away from the reference point $\theta_{\text{ref}} = (0, 0)^T$, and the score is no longer the sufficient statistics. The right panel of Fig. 3 shows the resulting expected confidence intervals from the various methods. The CASCAL and RASCAL techniques enable the highest precision, leading to exclusion limits virtually indistinguishable from those based on the true likelihood ratio.

7 Conclusions

In this work, we have presented a range of new inference techniques for the setting in which the likelihood is only implicitly defined through a stochastic generative simulator. The new methods estimates ratios of likelihoods with data available from the simulator. For the most general case, in which only samples of observations are available, we have presented the CARL algorithm that estimates likelihood ratios through calibrated discriminative classifiers. While the basic concept of ratio estimation by comparison is well established, we have improved this approach by adding a calibration step, and by considering parameterized estimators that learn the dependency of the likelihood ratio on the theory parameters from data.

We then considered a class of problems in which the simulator does not only provide access to samples, but also to the joint likelihood ratio or the joint score, quantities conditional on the latent variables that characterize the data generation process. This scenario is common in the physical sciences, for instance in particle physics and cosmology. While these additional quantities often require work to be extracted, they also prove to be very valuable as they can dramatically improve sample effiency and quality of inference. Indeed, we have shown that this additional information lets us define loss functionals that are minimized by the likelihood ratio, which can in turn be used to efficiently guide the training of neural networks to precisely estimate likelihood ratios. This is the idea behind the new ROLR, CASCAL, and RASCAL inference techniques, which are differentiated by which pieces of information they incorporate.

A third class of techniques is motivated by a local approximation of the likelihood function around a reference point, where the score vector are the sufficient statistics. In the case where the simulator provides the joint score, we can use it to train a precise estimator of the score and use it as a optimal summary statistics. We introduce the SALLY technique, which estimates likelihood ratios through multivariate density estimation in the estimated score space. The SALLINO method takes the dimensionality reduction one step further lets us compress any observation into a scalar function without losing information on the likelihood ratio, at least in the local approximation. We have demonstrated in two experiments that the new inference techniques let us precisely estimate likelihood ratios. In turn, this enables parameter measurement with a higher precision and less training data than with established methods.

Finally, these results motivates the development of tools that provide a non-standard interpretation of the simulator code and automatically generate the joint score and joint ratio. These tools could borrow from recent developments in probabilistic programming and automatic differentiation [26, 27, 37–40]. Such tools would reduce the effort needed to mine the gold so valuable to simulator-based inference.

Acknowledgments

We would like to thank Cyril Becot and Lukas Heinrich, who contributed to this project at an early stage. We are grateful to Felix Kling, Tilman Plehn, and Peter Schichtel for providing the MADMAX code and helping us use it. KC wants to thank CP3 at UC Louvain for their hospitality. Finally, we would like to thank Atılım Güneş Baydin, Lydia Brenner, Joan Bruna, Kyunghyun Cho, Michael Gill, Siavash Golkar, Ian Goodfellow, Daniela Huppenkothen, Michael Kagan, Hugo Larochelle, Yann LeCun, Fabio Maltoni, Jean-Michel Marin, Iain Murray, George Papamakarios, Duccio Pappadopulo, Dennis Prangle, Rajesh Ranganath, Dustin Tran, Rost Verkerke, Wouter Verkerke, Max Welling, and Richard Wilkinson for interesting discussions.

JB, KC, and GL are grateful for the support of the Moore-Sloan data science environment at NYU. KC and GL were supported through the NSF grants ACI-1450310 and PHY-1505463. JP was partially supported by the Scientific and Technological Center of Valparaíso (CCTVal) under Fondecyt grant BASAL FB0821. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

[1] D. B. Rubin: ‘Bayesianly justifiable and relevant frequency calculations for the applied statistician’. Ann. Statist. 12 (4), p. 1151, 1984. URL `https://doi.org/10.1214/aos/1176346785`.

[2] M. A. Beaumont, W. Zhang, and D. J. Balding: ‘Approximate bayesian computation in population genetics’. Genetics 162 (4), p. 2025, 2002.

[3] Y. Fan, D. J. Nott, and S. A. Sisson: ‘Approximate Bayesian Computation via Regression Density Estimation’. ArXiv e-prints , 2012. arXiv:1212.1479.

[4] L. Dinh, D. Krueger, and Y. Bengio: ‘NICE: Non-linear Independent Components Estimation’. ArXiv e-prints , 2014. arXiv:1410.8516.

[5] D. Jimenez Rezende and S. Mohamed: ‘Variational Inference with Normalizing Flows’. ArXiv e-prints , 2015. arXiv:1505.05770.

[6] K. Cranmer, J. Pavez, and G. Louppe: ‘Approximating Likelihood Ratios with Calibrated Discriminative Classifiers’ , 2015. arXiv:1506.02169.

[7] K. Cranmer and G. Louppe: ‘Unifying generative models and exact likelihood-free inference with conditional bijections’. J. Brief Ideas , 2016.

[8] L. Dinh, J. Sohl-Dickstein, and S. Bengio: ‘Density estimation using Real NVP’. ArXiv e-prints , 2016. arXiv:1605.08803.

[9] G. Papamakarios and I. Murray: ‘Fast ε-free inference of simulation models with bayesian conditional density estimation’. In ‘Advances in Neural Information Processing Systems’, p. 1028–1036, 2016.

[10] B. Paige and F. Wood: ‘Inference Networks for Sequential Monte Carlo in Graphical Models’. ArXiv e-prints , 2016. arXiv:1602.06701.

[11] R. Dutta, J. Corander, S. Kaski, and M. U. Gutmann: ‘Likelihood-free inference by ratio estimation’. ArXiv e-prints , 2016. arXiv:1611.10242.

[12] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle: ‘Neural Autoregressive Distribution Estimation’. ArXiv e-prints , 2016. arXiv:1605.02226.

[13] A. van den Oord, S. Dieleman, H. Zen, et al.: ‘WaveNet: A Generative Model for Raw Audio’. ArXiv e-prints , 2016. arXiv:1609.03499.

[14] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu: ‘Conditional Image Generation with PixelCNN Decoders’. ArXiv e-prints , 2016. arXiv:1606.05328.

[15] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu: ‘Pixel Recurrent Neural Networks’. ArXiv e-prints , 2016. arXiv:1601.06759.

[16] M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander: ‘Likelihood-free inference via classification’. Statistics and Computing p. 1–15, 2017.

[17] D. Tran, R. Ranganath, and D. M. Blei: ‘Hierarchical Implicit Models and Likelihood-Free Variational Inference’. ArXiv e-prints , 2017. arXiv:1702.08896.

[18] G. Louppe and K. Cranmer: ‘Adversarial Variational Optimization of Non-Differentiable Simulators’. ArXiv e-prints , 2017. arXiv:1707.07113.

[19] G. Papamakarios, T. Pavlakou, and I. Murray: ‘Masked Autoregressive Flow for Density Estimation’. ArXiv e-prints , 2017. arXiv:1705.07057.

[20] G. Papamakarios, D. C. Sterratt, and I. Murray: ‘Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows’. ArXiv e-prints , 2018. arXiv:1805.07226.

[21] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez: ‘Code repository for the generalized Galton board example in the paper “Mining gold from implicit models to improve likelihood-free inference”’. `http://github.com/johannbrehmer/simulator-mining-example`, 2018.

[22] M. M. Graham, A. J. Storkey, et al.: ‘Asymptotically exact inference in differentiable generative models’. Electronic Journal of Statistics 11 (2), p. 5105, 2017.

[23] R. J. Williams: ‘Simple statistical gradient-following algorithms for connectionist reinforcement learning’. In ‘Reinforcement Learning’, Springer, p. 5–32, 1992.

[24] S. Mohamed and B. Lakshminarayanan: ‘Learning in Implicit Generative Models’. ArXiv e-prints , 2016. arXiv:1610.03483.

[25] E. Meeds, R. Leenders, and M. Welling: ‘Hamiltonian abc’. arXiv preprint arXiv:1503.01916 , 2015.

[26] F. Wood, J. W. van de Meent, and V. Mansinghka: ‘A new approach to probabilistic programming inference’. In ‘Proceedings of the 17th International conference on Artificial Intelligence and Statistics’, p. 1024-1032, 2014.

[27] T. A. Le, Baydin, A. G. Baydin, and F. Wood: ‘Inference compilation and universal probabilistic programming’. In ‘Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)’, volume 54 of Proceedings of Machine Learning Research, p. 1338–1348. PMLR, Fort Lauderdale, FL, USA, 2017.

[28] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al.: ‘Generative Adversarial Networks’. ArXiv e-prints , 2014. arXiv:1406.2661.

[29] G. Louppe, K. Cranmer, and J. Pavez: ‘carl: a likelihood-free inference toolbox’. J. Open Source Softw. , 2016.

[30] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez: ‘Constraining Effective Field Theories with Machine Learning’ , 2018. arXiv:1805.00013.

[31] J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez: ‘A Guide to Constraining Effective Field Theories with Machine Learning’ , 2018. arXiv:1805.00020.

[32] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, and D. Whiteson: ‘Parameterized neural networks for high-energy physics’. Eur. Phys. J. C76 (5), p. 235, 2016. arXiv:1601.07913.

[33] J. Alwall, R. Frederix, S. Frixione, et al.: ‘The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations’. JHEP 07, p. 079, 2014. arXiv:1405.0301.

[34] K. Cranmer and T. Plehn: ‘Maximum significance at the LHC and Higgs decays to muons’. Eur. Phys. J. C51, p. 415, 2007. arXiv:hep-ph/0605268.

[35] T. Plehn, P. Schichtel, and D. Wiegand: ‘Where boosted significances come from’. Phys. Rev. D89 (5), p. 054002, 2014. arXiv:1311.2591.

[36] F. Kling, T. Plehn, and P. Schichtel: ‘Maximizing the significance in Higgs boson pair analyses’. Phys. Rev. D95 (3), p. 035026, 2017. arXiv:1607.07441.

[37] B. Eli, J. P. Chen, M. Jankowiak, et al.: ‘Pyro: Deep probabilistic programming’. `https://github.com/uber/pyro`, 2017.

[38] D. Tran, M. D. Hoffman, R. A. Saurous, E. Brevdo, K. Murphy, and D. M. Blei: ‘Deep probabilistic programming’. arXiv preprint arXiv:1701.03757 , 2017.

[39] N. Siddharth, B. Paige, J.-W. van de Meent, et al.: ‘Learning disentangled representations with semi-supervised deep generative models’. In I. Guyon, U. V. Luxburg, S. Bengio, et al. (eds.), ‘Advances in Neural Information Processing Systems 30’, p. 5927–5937. Curran Associates, Inc., 2017.

[40] A. Gelman, D. Lee, and J. Guo: ‘Stan: A Probabilistic Programming Language for Bayesian Inference and Optimization’. Journal of Educational and Behavioral Statistics 40 (5), p. 530, 2015.