Direct Amortized Likelihood Ratio Estimation

Adam D. Cobb, Brian Matejek, Daniel Elenius, Anirban Roy, Susmit Jha
Neuro-Symbolic Computing and Intelligence (NuSCI) Research Group
Computer Science Laboratory, SRI International

Abstract

We introduce a new amortized likelihood ratio estimator for likelihood-free simulation-based inference (SBI). Our estimator is simple to train and estimates the likelihood ratio using a single forward pass of the neural estimator. Our approach directly computes the likelihood ratio between two competing parameter sets which is different from the previous approach of comparing two neural network output values. We refer to our model as the direct neural ratio estimator (DNRE). As part of introducing the DNRE, we derive a corresponding Monte Carlo estimate of the posterior. We benchmark our new ratio estimator and compare to previous ratio estimators in the literature. We show that our new ratio estimator often outperforms these previous approaches. As a further contribution, we introduce a new derivative estimator for likelihood ratio estimators that enables us to compare likelihood-free Hamiltonian Monte Carlo (HMC) with random-walk Metropolis-Hastings (MH). We show that HMC is equally competitive, which has not been previously shown. Finally, we include a novel real-world application of SBI by using our neural ratio estimator to design a quadcopter.

1 Introduction

In many scientific applications we rely on complex simulators to provide us with a set of observations, $\mathbf{x}$, for a corresponding set of parameters, $\boldsymbol{\theta}$. Common examples range from simulating Computational Fluid Dynamics (or CFD), to flight simulators and computational biology. This paradigm is often referred to as simulation-based inference (SBI) [1]. An objective of SBI is to perform Bayesian inference to learn the posterior over the parameters of a simulator, where all solutions require accounting for a lack of an analytical likelihood. They revolve around Bayes' rule and the corresponding Bayesian inference approaches that currently exist in the literature. There exists multiple neural-based approaches such as Neural Posterior Estimation [2], Neural Likelihood Estimation (NLE) [3], and Neural Ratio Estimation (NRE) [4]. NRE is closely tied to MCMC in that it is used to estimate the ratio in the MH step. While there have been many papers that have focused on comparison between all aforementioned approaches, including their sequential and amortized variants (e.g. Lueckmann et al. [5]), the focus of this paper is to explore within the existing range of neural ratio estimators.

Contributions. In this workshop paper we build on the work of [6, 7, 8] by introducing a new amortized neural likelihood ratio estimator that *directly* computes the likelihood ratio between two sets of parameters and only requires a single pass through the network to achieve this estimation. We derive a corresponding new Monte Carlo estimate of the posterior distribution when using DNRE. We also introduce a new gradient estimator that can be applied to both our approach and previous approaches. This gradient estimator is more numerically stable than the previous one. We benchmark DNRE along with the baselines both with and without HMC on standard SBI tasks showing that DNRE can often outperform the baselines. These experiments also show likelihood-free HMC to be competitive, which has not been previously shown. Our final contribution is the introduction of a novel design example for quadcopters.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

2 Preliminaries

We start with the derivation of the the likelihood-to-evidence ratio [7, 6]:

$$r(\mathbf{x}|\boldsymbol{\theta}) = \frac{\mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta})}{1 - \mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta})} = \frac{p(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{x})p(\boldsymbol{\theta})} = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}, \tag{1}$$

where $\mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta})$ is the optimal decision function that distinguishes between two independent samples, $\{\mathbf{x}, \boldsymbol{\theta}\} \sim p(\mathbf{x})p(\boldsymbol{\theta})$, and samples from the joint, $\{\mathbf{x}, \boldsymbol{\theta}\} \sim p(\mathbf{x}, \boldsymbol{\theta})$. To learn this classifier, Hermans et al. [6] train a neural network by sampling two independent $\boldsymbol{\theta}$'s from the prior, $\{\boldsymbol{\theta}, \boldsymbol{\theta}'\}$, and simulate $\mathbf{x}$ according to $p(\mathbf{x}|\boldsymbol{\theta})$. The parameterized classifier is then trained using a binary cross entropy loss whereby the samples from the joint, $(\mathbf{x}, \boldsymbol{\theta})$ are given the label 1 and the independent samples, $(\mathbf{x}, \boldsymbol{\theta}')$, are given the label 0. The output of the network is an estimate of the log ratio, $\log \hat{r}$.

Once learned, the likelihood-to-evidence ratio estimator is used to estimate the posterior by multiplying the ratio by the prior, $p(\boldsymbol{\theta}|\mathbf{x}) \approx \hat{r}(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. However, to estimate the likelihood ratio between two competing observations, $p(\mathbf{x}|\boldsymbol{\theta})/p(\mathbf{x}|\boldsymbol{\theta}')$, one needs to apply two forward passes through the network with two sets of parameters to get $r(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\theta}') = r(\mathbf{x}|\boldsymbol{\theta})/r(\mathbf{x}|\boldsymbol{\theta}')$. This form can then be used to perform likelihood-free MCMC by replacing the likelihood ratio inside the MH acceptance step.

3 DNRE: Direct Amortized Neural Likelihood Ratio Estimation

In this section we present our approach of Direct Amortized Neural Likelihood Ratio Estimation (DNRE), which directly parameterizes the classifier with the two parameter pairs, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, as $\mathrm{d}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}')$. As a result we introduce the new optimal classifier, d^* as follows:

$$\mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{p(\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}')}{p(\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}') + p(\mathbf{x}, \boldsymbol{\theta}')p(\boldsymbol{\theta})} = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x}|\boldsymbol{\theta}) + p(\mathbf{x}|\boldsymbol{\theta}')}, \tag{2}$$

which leads to the new direct amortized likelihood estimator:

$$r(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{\mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}')}{1 - \mathrm{d}^*(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}')} = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x}|\boldsymbol{\theta}')}. \tag{3}$$

An advantage of this approach is the additional information provided to the classifier for label $y = 0$. This is when the denominator, $p(\mathbf{x}|\boldsymbol{\theta}')$, must be greater than the numerator, $p(\mathbf{x}|\boldsymbol{\theta})$. We force this relationship to be the case by swapping $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ such that the parameter in the denominator becomes the one that generated the observation $\mathbf{x}$. This is compared to learning to distinguish between the joint distribution and independent samples which will result in a softer decision boundary for the classifier to learn. A key difference when using DNRE, compared to the original NRE is that DNRE takes three inputs $(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}')$, instead of the two. We ensure that the order is consistent while training using the binary cross entropy loss. In our case we assign the ordered triplet $(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\theta}')$ a label 1, and swap $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ for the label 0. We highlight that by explicitly incorporating both sets of parameters we aim to learn the optimal direct likelihood ratio estimator $r(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\theta}')$ in Equation (3).

3.1 Monte Carlo Posterior Approximation

Unlike the likelihood-to-evidence ratio proposed in Equation (1), DNRE requires Monte Carlo sampling to approximate the posterior. We achieve this by integrating out the $\boldsymbol{\theta}'$ in the denominator numerically, using M samples. This requires an inverse trick that we derive in the log-space:

$$\log p(\boldsymbol{\theta}|\mathbf{x}) \approx -\mathrm{logSumExp}\big\{-\log \hat{r}(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\theta}'_i))\big\} + \log M + \log p(\boldsymbol{\theta}).$$

For higher-dimensional $\boldsymbol{\theta}'$ this is computationally intractable, but for smaller dimensional problems we can use this form to compare posteriors in a manner consistent with previous approaches within the literature. In the longer version of this work, (anon.), we have shown this Monte Carlo approximation to be useful and accurate.

3.2 Likelihood-free Hamiltonian Monte Carlo

HMC is a gradient-based Markov chain Monte Carlo sampling scheme that augments the original parameter space with additional momentum parameters, $\mathbf{m}$, in order to sample using Hamiltonian dynamics [9, 10]. HMC defines the potential energy function as $U(\boldsymbol{\theta}) = -\log[p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})]$ and the kinetic energy function as $K(\mathbf{m}) = \mathbf{m}^\top\mathbf{m}/2$. The two requirements of running an HMC algorithms are access to $\nabla U(\boldsymbol{\theta})$ and access to the MH acceptance step of $\rho = \min(0, -U(\boldsymbol{\theta}^*) + U(\boldsymbol{\theta}) - K(\mathbf{m}^*) + K(\mathbf{m}))$, where $\boldsymbol{\theta}^*$ and $\mathbf{m}^*$ are the proposed parameter and momenta pair. For NRE and BNRE, ρ is derived from two passes through the estimator such that $-U(\boldsymbol{\theta}^*) + U(\boldsymbol{\theta}) = \log r(\mathbf{x}|\boldsymbol{\theta}^*) - \log r(\mathbf{x}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}^*) - \log p(\boldsymbol{\theta})$. For our new estimator DNRE, we only require one pass such that $-U(\boldsymbol{\theta}^*) + U(\boldsymbol{\theta}) = \log r(\mathbf{x}|\boldsymbol{\theta}^*, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}^*) - \log p(\boldsymbol{\theta})$. To estimate $\nabla U(\boldsymbol{\theta})$, Hermans et al. [6] use the chain rule to derive $\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) = -\nabla_{\boldsymbol{\theta}} r(\mathbf{x}|\boldsymbol{\theta})/r(\mathbf{x}|\boldsymbol{\theta})$. This estimate of the derivative can be numerically unstable in practice as it requires exponentiating the output of the estimator for the denominator, which we have observed can often be a small number. A more simple approach is to estimate $\nabla U(\boldsymbol{\theta})$ by treating the classifer as the approximate log ratio which allows us to separate the two terms. For NRE we can achieve the new approximated derivative of the log likelihood as:

$$\nabla_{\boldsymbol{\theta}} \log r(\mathbf{x}|\boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}) = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}). \tag{4}$$

We also derive the same approximation for our DNRE approach:

$$\nabla_{\boldsymbol{\theta}} \log r(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\theta}') \approx \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}') = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}). \tag{5}$$

We have found these new estimators to be more stable, as they avoid the exponential and division operations. For DNRE, we need to account for the $\boldsymbol{\theta}'$ when estimating the derivative. Therefore we choose to sample $\boldsymbol{\theta}'$ from its prior during our implementation. For estimators that closely match the true ratio, the contribution from $\boldsymbol{\theta}'$ to the derivative should tend to zero, following Equation (5).

4 SBI benchmark results

In this section we compare three amortized estimators: Neural Ratio Estimator (NRE) [6], Balanced Neural Ratio Estimator (BNRE) [11], and our new approach of Direct Neural Ratio Estimator (DNRE). The baseline approaches are implemented using the *LAMPE* Python package [12] and are applied to the SBI Benchmark examples of [5]. All estimators have an architecture of five layers of 64 units, using the Exponential Linear Unit non-linearity between layers. For all approaches we applied the same grid search over both the learning rate and standard deviation of the proposal distribution for the random-walk MH sampling scheme. For HMC, we extend the grid search to include desired acceptance rates ranging between 0.5 to 0.8 and the trajectory length. This uses the dual averaging scheme as introduced in [13] and as implemented in [14]. The best model for each task was selected according to the `C2ST` score applied for $1{,}000$ samples using the first observation of the benchmark. The `C2ST` score is a classifier-based test trained to distinguish between reference posterior samples and samples from the likelihood-free inference. The results in Table 1 are averaged across the 10 available observations in the benchmark. A score of 0.5 is optimal, whereas 1.0 would be the worst score. In addition to comparing across ratio estimators, the table compares both MH and HMC sampling schemes. For seven out of the ten tasks, the best performer comes from either DNRE MH, or DNRE HMC. Additionally, we note that different sampling schemes tend to outperform on different tasks and we would therefore recommend experimenting with both HMC and MH for any future use of likelihood ratio estimation.

5 Quadcopter design: Using DNRE to improve an existing design

An interesting use-case of SBI is to use our likelihood ratio estimator to improve on an existing cyberphysical system design.

Simulator. In particular, we run a custom flight dynamics pipeline which combines CAD software [15] with a computational flight simulation model [16, 17]. To use the pipeline, one can define an aircraft design and allow the simulator to determine performances such as its drag, lift, maximum flight time and hover time. We want to explore how we should parameterize a quadcopter such that we achieve a certain performance level. The observations, $\mathbf{x} \in \mathbb{R}^7$, consist

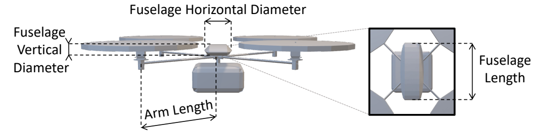

Figure 1: Schematic of the quadcopter design which highlights the key design parameters related to the fuselage and the arm length.

Approach	TM	GL	LV	SIR	SLCP
NRE MH	0.559 ± 0.026	0.533 ± 0.015	$\mathbf{0.995 \pm 0.004}$	$\mathbf{0.803 \pm 0.083}$	0.925 ± 0.039
NRE HMC	0.657 ± 0.026	0.531 ± 0.015	0.996 ± 0.005	$\mathbf{0.803 \pm 0.082}$	0.913 ± 0.042
BNRE MH	$\mathbf{0.544 \pm 0.033}$	0.59 ± 0.019	0.997 ± 0.002	0.944 ± 0.029	0.893 ± 0.038
BNRE HMC	0.562 ± 0.07	0.543 ± 0.019	0.997 ± 0.003	0.945 ± 0.029	0.891 ± 0.039
DNRE MH	0.587 ± 0.027	0.576 ± 0.034	0.998 ± 0.002	0.871 ± 0.064	$\mathbf{0.826 \pm 0.078}$
DNRE HMC	0.753 ± 0.14	$\mathbf{0.519 \pm 0.006}$	0.998 ± 0.001	0.891 ± 0.043	0.892 ± 0.098
	GLU	SLCP D	B GLM	GM	B GLM R
NRE MH	0.618 ± 0.025	0.982 ± 0.008	0.781 ± 0.046	0.751 ± 0.015	0.819 ± 0.035
NRE HMC	0.613 ± 0.025	0.995 ± 0.003	0.776 ± 0.044	0.752 ± 0.016	0.83 ± 0.044
BNRE MH	0.607 ± 0.021	0.984 ± 0.007	0.807 ± 0.039	0.755 ± 0.016	0.862 ± 0.047
BNRE HMC	0.598 ± 0.025	0.991 ± 0.009	0.788 ± 0.036	0.751 ± 0.015	0.86 ± 0.047
DNRE MH	$\mathbf{0.597 \pm 0.025}$	$\mathbf{0.98 \pm 0.011}$	0.813 ± 0.069	$\mathbf{0.747 \pm 0.015}$	0.777 ± 0.062
DNRE HMC	0.665 ± 0.162	0.989 ± 0.008	$\mathbf{0.738 \pm 0.038}$	$\mathbf{0.747 \pm 0.013}$	$\mathbf{0.77 \pm 0.109}$

Table 1: C2ST SBI benchmark results comparing NRE, BNRE, and DNRE with both Metropolis-Hastings (MH) and HMC sampling schemes.

of: the number of interferences; the mass; the maximum flight distance; the maximum hover time; the maximum lateral speed; the maximum control input at the maximum flight distance; and the maximum power at the maximum speed. The design parameters, $\boldsymbol{\theta} \in \mathbb{R}^{19}$, consist of: the arm length; four fuselage shape parameters; and seven 'x' and 'y' locations of electrical devices inside the fuselage. We highlight some of the key parameters in Figure 1.

Improving an Existing Design. Figure 2 shows how we can use likelihood-free inference with DNRE HMC to improve an existing design. Here, the current design has 60 structural interferences as highlighted by the seed design. Our objective is to condition on the design to have zero interferences, as well as other favorable design metrics, such as a hover time of 24 s and a maximum velocity of 33 m/s. We then initialize our HMC chain with this poor seed design and perform likelihood-free inference with DNRE HMC. We take 200 steps with a thinning of 8 and run each design through the simulator. We see the evolution of our seed design from having a 60 interferences (60) to only 4. This is shown in the Figure. Additionally, all other design metrics are closely followed, where there is low mean squared error between the design objectives and the actual performance.

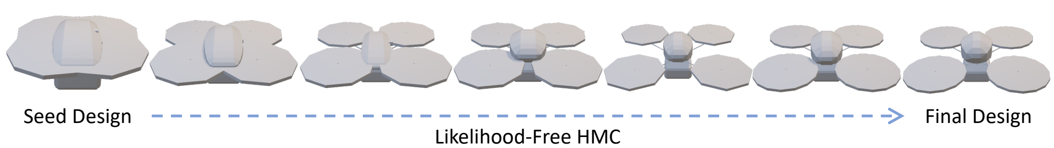

Figure 2: Sub-sampled quadcopter designs taken along the likelihood-free HMC chain using DNRE. The initial seed design on the far left has multiple structural interferences, including sensors that cut through the fuselage. As we move along the chain we see the design morph into our desired structure with very few interferences. This is achieved through increasing the arm length and changing the shape of the fuselage, as well as varying the placement of the interior sensing components.

6 Conclusion

The contribution of this paper is to demonstrate that directly learning to approximate the likelihood ratio between two pairs of parameters presents itself as a viable option for likelihood-free inference and often outperforms competing approaches on standard SBI benchmarks. As part of our contribution, we derive a new Monte Carlo estimator for the posterior distribution when using our DNRE approach. We also derive a simple likelihood gradient estimator that can be successfully used to perform HMC. We are therefore the first to compare random walk MH with HMC for likelihood ratio estimation approahces. We find that HMC is a viable MCMC approach and can outperform random-walk MH. Finally, we introduce a novel application of SBI for the design of a quadcopter and we release this data as part of the supplementary materials.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-20-C-0002. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

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

[2] George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with Bayesian conditional density estimation. *Advances in neural information processing systems*, 29, 2016.

[3] George Papamakarios, David Sterratt, and Iain Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

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

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

[6] Joeri Hermans, Volodimir Begy, and Gilles Louppe. Likelihood-free MCMC with amortized approximate ratio estimators. In *International conference on machine learning*, pages 4239–4248. PMLR, 2020.

[7] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*, 2016.

[8] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 17(1):1–31, 2022.

[9] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

[10] Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[11] Arnaud Delaunoy, Joeri Hermans, François Rozet, Antoine Wehenkel, and Gilles Louppe. Towards Reliable Simulation-Based Inference with Balanced Neural Ratio Estimation. *arXiv preprint arXiv:2208.13624*, 2022.

[12] François Rozet. LAMPE. https://github.com/francois-rozet/lampe, 2022.

[13] Matthew D Hoffman, Andrew Gelman, et al. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[14] Adam D Cobb, Atılım Güneş Baydin, Andrew Markham, and Stephen J Roberts. Introducing an Explicit Symplectic Integration Scheme for Riemannian Manifold Hamiltonian Monte Carlo. *arXiv preprint arXiv:1910.06243*, 2019.

[15] PTC. Creo CAD Software. `https://www.ptc.com/en/products/creo`, 2022.

[16] James D. Walker, F. Michael Heim, Bapiraju Surampudi, Pablo Bueno, Alexander Carpenter, Sidney Chocron, Jon Cutshall, Richard Lammons, Theodore Bapty, Brian Swenson, and Sydney Whittington. A Flight Dynamics Model for Exploring the Distributed Electrical eVTOL Cyber Physical Design Space. *Design Automation for CPS and IoT (DESTION)*, 2022.

[17] Theodore Bapty, Sydney Whittington, James D. Walker, Joseph Hite, Brian Swenson, Kathrine Owens, Fred Eisele, Jason Scott, and Robert Owens. Design Oracle for AI-Based CPS Design. *Design Automation for CPS and IoT (DESTION)*, 2022.