swyft: Truncated Marginal Neural Ratio Estimation in Python

Benjamin Kurt Miller [1,2,3], **Alex Cole** [1], **Christoph Weniger** [1], **Francesco Nattino** [4], **Ou Ku** [4], **and Meiert W. Grootes** [4]

1 Gravitation Astroparticle Physics Amsterdam (GRAPPA), University of Amsterdam, Science Park 904, 1098 XH Amsterdam **2** Amsterdam Machine Learning Lab (AMLab), University of Amsterdam, Science Park 904, 1098 XH Amsterdam **3** AI4Science Lab, University of Amsterdam, Science Park 904, 1098 XH Amsterdam **4** Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands

DOI: 10.21105/joss.04205

Software

- Review
- Repository
- Archive

Editor: Pierre de Buyl

Reviewers:

- @mattpitkin
- @olgadoronina

Submitted: 18 January 2022
Published: 19 July 2022

Summary

Parametric stochastic numerical simulators are ubiquitous in science. They model observed phenomena by mapping a parametric representation of simulation conditions to a hypothetical observation–effectively sampling from a probability distribution over observational data known as the likelihood. Simulators are advantageous because they easily encode relevant scientific knowledge. *Simulation-based inference* (SBI) is a machine learning technique which applies a simulator, a fitted statistical surrogate model, and a set of prior beliefs to estimate a probabilistic description of the parameters which plausibly generated some observational data. This description of parameters is known as the posterior and it is the end-product of Bayesian inference.

Our package `swyft` implements a specific, simulation-efficient SBI method called *Truncated Marginal Neural Ratio Estimation* (TMNRE) (Miller et al., 2021); it estimates the likelihood-to-evidence ratio to approximate the posterior, as in Hermans et al. (2020). `swyft` (Miller et al., 2020) provides a collection of tools to simulate and store data, locally or in a distributed computing setting, and perform (marginalized) simulation-based Bayesian inference. It produces ready-to-publish plots that demonstrate the calibration of the posterior estimate along with the posterior itself.

Statement of Need

Estimating the posterior can be prohibitively expensive for complex data and slow simulators. Part of the reason is the sequential nature of likelihood-based Markov chain Monte-Carlo (Hastings, 1970; Metropolis et al., 1953). In contrast, SBI parallelizes simulation in most circumstances, thereby reducing the practical waiting time for results. In pursuit of further simulation efficiency, Miller et al. (2021) argue that fitting the joint posterior for all parameters is unnecessary when a marginal estimate of the posterior will suffice. Some SBI methods are amortized, whereby the statistical model is fit to estimate posteriors for all possible observations simultaneously. While amortization enables necessary posterior calibration checks, like expected coverage probability (Hermans et al., 2021; Miller et al., 2021), it is more efficient to fit the model on only a subset of the parameters that could have plausibly generated the observation.

`swyft` satisfies necessary requirements, like estimating the marginal posteriors of interest and enabling posterior calibration checks, while taking a lean approach to avoid all unnecessary simulation. In this pursuit, `swyft` truncates the prior to regions relevant for given observational data and reuses compatible existing simulations. `swyft` automates irksome matters like distributed computing and data storage with dask (Dask Development Team, 2016) and zarr (Miles et al., 2021) respectively. swyft is designed to:

1. Estimate arbitrary marginal posteriors, i.e., the posterior over parameters of interest, marginalizing over nuisance parameters.
2. Perform targeted inference by truncating the prior distribution with an indicator function estimated in a sequence of inferences.
3. Estimate the expected coverage probability of fully amortized SBI posteriors and locally amortized posteriors that are limited to truncated regions.
4. Seamlessly reuse simulations from previous analyses by drawing already-simulated data first via a flexible storage solution.
5. Integrate advanced distribution and storage tools to simplify application of complex simulators.

Although there is a rich ecosystem of SBI implementations, TMNRE did not naturally fit in an existing framework since it requires parallel estimation of marginal posteriors and a truncated prior. swyft does the parallel training of the ratio using another dimension in a PyTorch tensor and created a custom truncated prior data structure to overcome these challenges. swyft aims to meet the ever-increasing demand for efficient and testable Bayesian inference in fields like physics, cosmology, and astronomy by implementing TMNRE together with practical distributed computing and storage tools.

Existing research with swyft

The software package has enabled inference on dark matter substructure in strongly lensed galaxies (Coogan et al., 2020), estimated cosmological parameters from cosmic microwave background simulation data (Cole et al., 2021), and was cited in a white paper laying out a vision for astropartical physics research during the next decade (Batista et al., 2021). Ongoing work with swyft aims to reduce the response time to gravitational wave triggers from LIGO-Virgo by estimating the marginal posterior with unprecedented speed. There is an existing proof-of-concept by Delaunoy et al. (2020) although the swyft software package was not applied. Generally, speeding up gravitational wave inference using simulation-based inference is an active area of research (Chua & Vallisneri, 2020; Dax et al., 2021; Gabbard et al., 2022). In another work-in-progress, swyft helps to characterize the magnetohydrodynamics of binary neutron star mergers using multi-messenger gravitational and electrodynamic data where marginalization would be impossible with likelihood-based methods.

Related theoretical work

There is a long tradition of likelihood-free inference, also known as *Approximate Bayesian Computation* (ABC), going back to as early as the 1980s (Beaumont et al., 2009; Diggle & Gratton, 1984; Rubin, 1984; Tavaré et al., 1997; Toni et al., 2009). Traditional techniques use Monte-Carlo rejection sampling and are summarized by Sisson et al. (2018) and Karabatsos & Leisen (2018). We track the development of classifiers for the estimation of likelihood ratios to a few references. Cranmer et al. (2015) compared the ratio between the likelihood of a freely varying parameter and a fixed reference value for frequentist inference. Pham et al. (2014) estimated the ratio between likelihoods for Markov chain Monte-Carlo sampling. Thomas et al. (2016) and Gutmann et al. (2018) introduced the framework which allows for likelihood-to-evidence ratio estimation. Like swyft, Blum & François (2010) proposed to truncate the prior for sampling but do so within an ABC scheme.

Modern SBI is a quickly evolving field that has several techniques under development (Cranmer et al., 2020). Neural network-based methods are categorized according to the term they approximate in Bayes' formula. swyft is a method which approximates the likelihood-to-evidence ratio $\frac{p(x|\theta)}{p(x)}$ where θ are the parameters and x is the observational data. Works by Hermans et al. (2020), Durkan et al. (2020), and Rozet & Louppe (2021) are closely related to `swyft` as they also approximate the likelihood-to-evidence ratio. Like `swyft`, Rozet & Louppe (2021) estimate marginal posteriors, but unlike swyft, they attempt to amortize over all possible marginals with a single neural network. Other methods estimate the posterior directly (Durkan et al., 2020; Greenberg et al., 2019; Lueckmann et al., 2017; Papamakarios & Murray, 2016) or the likelihood itself (Lueckmann et al., 2019; Papamakarios et al., 2019).

Related software

`swyft` is unique because it implements TMNRE and a method for simulation reuse. It also offers sophisticated distributed simulation and storage tools coupled directly to the software. We briefly discuss the alternatives in the thriving ecosystem of SBI software packages.

`sbi` (Tejero-Cantero et al., 2020) features a selection of modern neural SBI algorithms. It is accompanied by a benchmark `sbibm` (Lueckmann et al., 2021) which tests those methods against a set of tractable toy problems. `pydelfi` (Alsing, 2019) estimates the likelihood of a learned summary statistic (Alsing et al., 2018, 2019)—`swyft` users should pay special attention to this repository since it can also project out nuisance parameters (Alsing & Wandelt, 2019). `carl` (Louppe et al., 2016) uses a classifier to estimate the likelihood ratio as Cranmer et al. (2015) did and `hypothesis` (Hermans, 2019) includes several toy simulators.

Non-neural implementations for SBI also exist. `elfi` (Lintusaari et al., 2018) implements BOLFI, an algorithm based on Gaussian processes (Gutmann et al., 2016). `pyabc` (Klinger et al., 2018) and `ABCpy` (Dutta et al., 2017) are two suites of ABC algorithms.

Description of software

`swyft` implements *Marginal Neural Ratio Estimation* (MNRE), a method which trains an amortized likelihood-to-evidence ratio estimator for any marginal posterior of interest. `swyft` makes it easy to estimate a set of marginals in parallel, e.g., for a corner plot. To use `swyft`, the operator must provide a quantification of prior beliefs, a python-callable or bash-scriptable simulator, and an observation-of-interest.

Performing TMNRE with `swyft`, by restricting simulation to a truncated prior region, is simple and demonstrated in the documentation. Constructing these truncated regions can be done manually or based on a previous inference. Routines are provided for all necessary plots and for calculating the expected coverage probability of a given likelihood-to-evidence ratio estimator. This calculation is essential as a sanity check to determine whether the approximate posterior is calibrated.

The machine learning aspects of `swyft` are implemented in PyTorch (Paszke et al., 2019) while the truncated prior is implemented within `numpy` (Harris et al., 2020). Storing previously simulated data for reuse in later analyses is accomplished with `zarr` (Miles et al., 2021) and parallelization of simulation is achieved with `dask` (Dask Development Team, 2016). `swyft` has other important dependencies, namely `scipy` (Virtanen et al., 2020), `seaborn` (Waskom, 2021), `matplotlib` (Hunter, 2007), `pandas` (McKinney, 2010; Reback et al., 2021), and `jupyter` (Kluyver et al., 2016).

Acknowledgements

The developers of `swyft` want to thank early adopters of the software Adam Coogan, Noemi Anau Montel, Kosio Karchev, Elias Dubbeldam, Uddipta Bhardwaj, and Ioannis Bousdoukos. We are grateful for the additional expertise offered by Patrick Forré and Samaya Nissanke.

Benjamin Kurt Miller is funded by the University of Amsterdam Faculty of Science (FNWI), Informatics Institute (IvI), and the Institute of Physics (IoP). This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864035 – UnDark). This work was also supported by the Netherlands eScience Center and SURF under grant number ETEC.2019.018. This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. We also would like to thank SURF for providing computational resources via the EINF-1194 grant.

References

Alsing, J. (2019). pydelfi: Density estimation likelihood-free inference. In *GitHub repository*. https://github.com/justinalsing/pydelfi; GitHub.

Alsing, J., Charnock, T., Feeney, S., & Wandelt, B. (2019). Fast likelihood-free cosmology with neural density estimators and active learning. *Monthly Notices of the Royal Astronomical Society*, *488*(3), 4440–4458. https://doi.org/10.1093/mnras/stz1960

Alsing, J., & Wandelt, B. (2019). Nuisance hardened data compression for fast likelihood-free inference. *Monthly Notices of the Royal Astronomical Society*, *488*(4), 5093–5103. https://doi.org/10.1093/mnras/stz1900

Alsing, J., Wandelt, B., & Feeney, S. (2018). 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. https://doi.org/10.1093/mnras/sty819

Batista, R. A., Amin, M., Barenboim, G., Bartolo, N., Baumann, D., Bauswein, A., Bellini, E., Benisty, D., Bertone, G., Blasi, P., & others. (2021). EuCAPT white paper: Opportunities and challenges for theoretical astroparticle physics in the next decade. *arXiv Preprint arXiv:2110.10074*.

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

Blum, M. G., & François, O. (2010). Non-linear regression models for approximate bayesian computation. *Statistics and Computing*, *20*(1), 63–73. https://doi.org/10.1007/s11222-009-9116-0

Chua, A. J., & Vallisneri, M. (2020). Learning bayesian posteriors with neural networks for gravitational-wave inference. *Physical Review Letters*, *124*(4), 041102. https://doi.org/10.1103/PhysRevLett.124.041102

Cole, A., Miller, B. K., Witte, S. J., Cai, M. X., Grootes, M. W., Nattino, F., & Weniger, C. (2021). Fast and credible likelihood-free cosmology with truncated marginal neural ratio estimation. *arXiv Preprint arXiv:2111.08030*.

Coogan, A., Karchev, K., & Weniger, C. (2020). Targeted likelihood-free inference of dark matter substructure in strongly-lensed galaxies. *Machine Learning and the Physical Sciences: Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS)*.

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proc. Natl. Acad. Sci. U. S. A.*

Cranmer, K., Pavez, J., & Louppe, G. (2015). Approximating likelihood ratios with calibrated discriminative classifiers. *arXiv Preprint arXiv:1506.02169*.

Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. https://dask.org

Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A., & Schölkopf, B. (2021). Real-time gravitational wave science with neural posterior estimation. *Physical Review Letters*, *127*(24), 241103. https://doi.org/10.1103/PhysRevLett.127.241103

Delaunoy, A., Wehenkel, A., Hinderer, T., Nissanke, S., Weniger, C., Williamson, A. R., & Louppe, G. (2020). Lightning-fast gravitational wave parameter inference through neural amortization. *Machine Learning and the Physical Sciences: Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS)*.

Diggle, P. J., & Gratton, R. J. (1984). Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, *46*(2), 193–212. https://doi.org/10.1111/j.2517-6161.1984.tb01290.x

Durkan, C., Murray, I., & Papamakarios, G. (2020). On contrastive learning for likelihood-free inference. *International Conference on Machine Learning*, 2771–2781.

Dutta, R., Schoengens, M., Onnela, J.-P., & Mira, A. (2017). ABCpy: A user-friendly, extensible, and parallel library for approximate bayesian computation. *Proceedings of the Platform for Advanced Scientific Computing Conference*, 8:1–8:9. https://doi.org/10.1145/3093172.3093233

Gabbard, H., Messenger, C., Heng, I. S., Tonolini, F., & Murray-Smith, R. (2022). Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy. *Nature Physics*, *18*(1), 112–117. https://doi.org/10.1038/s41567-021-01425-7

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

Gutmann, M. U., Corander, J., & others. (2016). Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*.

Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2018). Likelihood-free inference via classification. *Statistics and Computing*, *28*(2), 411–425. https://doi.org/10.1007/s11222-017-9738-6

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., R'ıo, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *57*(1), 97–109. https://doi.org/10.1093/biomet/57.1.97

Hermans, J. (2019). Hypothesis. In *GitHub repository*. https://github.com/montefiore-ai/hypothesis; GitHub.

Hermans, J., Begy, V., & Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. *International Conference on Machine Learning*, 4239–4248.

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

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Karabatsos, G., & Leisen, F. (2018). An approximate likelihood perspective on ABC methods. *Statistics Surveys*, *12*, 66–104. https://doi.org/10.1214/18-ss120

Klinger, E., Rickert, D., & Hasenauer, J. (2018). pyABC: Distributed, likelihood-free inference. *Bioinformatics*, *34*(20), 3591–3593. https://doi.org/10.1093/bioinformatics/bty361

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & team, J. development. (2016). Jupyter notebooks - a publishing format for reproducible computational workflows. In F. Loizides & B. Scmidt (Eds.), *Positioning and power* *in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. https://eprints.soton.ac.uk/403913/

Lintusaari, J., Vuollekoski, H., Kangasrääsiö, A., Skytén, K., Järvenpää, M., Marttinen, P., Gutmann, M. U., Vehtari, A., Corander, J., & Kaski, S. (2018). ELFI: Engine for likelihood-free inference. *The Journal of Machine Learning Research*, *19*(1), 643–649.

Louppe, G., Cranmer, K., & Pavez, J. (2016). carl: A likelihood-free inference toolbox. *Journal of Open Source Software*, *1*(1), 11. https://doi.org/10.21105/joss.00011

Lueckmann, J.-M., Bassetto, G., Karaletsos, T., & Macke, J. H. (2019). Likelihood-free inference with emulator networks. *Symposium on Advances in Approximate Bayesian Inference*, 32–53.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., & Macke, J. (2021). Benchmarking simulation-based inference. In A. Banerjee & K. Fukumizu (Eds.), *Proceedings of the 24th international conference on artificial intelligence and statistics* (Vol. 130, pp. 343–351). PMLR. http://proceedings.mlr.press/v130/lueckmann21a.html

Lueckmann, J.-M., Gonçalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., & Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1289–1299.

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, *21*(6), 1087–1092. https://doi.org/10.2172/4390578

Miles, A., jakirkham, Bussonnier, M., Moore, J., Fulton, A., Bourbeau, J., Onalan, T., Hamman, J., Patel, Z., Rocklin, M., & al., et. (2021). *Zarr-developers/zarr-python:* https://doi.org/10.5281/zenodo.5712786

Miller, B. K., Cole, A., Forré, P., Louppe, G., & Weniger, C. (2021). Truncated marginal neural ratio estimation. *Advances in Neural Information Processing Systems*, *34*.

Miller, B. K., Cole, A., Louppe, G., & Weniger, C. (2020). Simulation-efficient marginal posterior estimation with swyft: Stop wasting your precious time. *Machine Learning and the Physical Sciences: Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS)*.

Papamakarios, G., & Murray, I. (2016). Fast ϵ-free inference of simulation models with bayesian conditional density estimation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 29). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/6aca97005c68f1206823815f66102863-Paper.pdf

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pham, K. C., Nott, D. J., & Chaudhuri, S. (2014). A note on approximating ABC-MCMC using flexible classifiers. *Stat*, *3*(1), 218–227. https://doi.org/10.1002/sta4.56

Reback, J., jbrockmendel, McKinney, W., Bossche, J. V. den, Augspurger, T., Cloud, P., Hawkins, S., Roeschke, M., gfyoung, Sinhrks, & al., et. (2021). *Pandas-dev/pandas*. https://doi.org/10.5281/zenodo.3509134

Rozet, F., & Louppe, G. (2021). Arbitrary marginal neural ratio estimation for simulation-based inference. *Machine Learning and the Physical Sciences: Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS)*.

Rubin, D. B. (1984). Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician. *The Annals of Statistics*, *12*(4), 1151–1172. https://doi.org/10.1214/aos/1176346785

Sisson, S. A., Fan, Y., & Beaumont, M. (2018). *Handbook of approximate bayesian computation*. CRC Press.

Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring Coalescence Times From DNA Sequence Data. *Genetics*, *145*(2), 505–518. https://doi.org/10.1093/genetics/145.2.505

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., & Macke, J. H. (2020). Sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, *5*(52), 2505. https://doi.org/10.21105/joss.02505

Thomas, O., Dutta, R., Corander, J., Kaski, S., Gutmann, M. U., & others. (2016). Likelihood-free inference by ratio estimation. *Bayesian Analysis*. https://doi.org/10.1214/20-ba1238

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. R. Soc. Interface*, *6*(31), 187–202. https://doi.org/10.1098/rsif.2008.0172

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021