The Journal of Open Source Software

sbi reloaded: a toolkit for simulation-based inference workflows

Jan Boelts[1,2,3*¶], **Michael Deistler**[1,2*¶], **Manuel Gloeckler**[1,2], **Álvaro Tejero-Cantero**[3,4], **Jan-Matthis Lueckmann**[5], **Guy Moss**[1,2], **Peter Steinbach**[6], **Thomas Moreau**[7], **Fabio Muratore**[8], **Julia Linhart**[7], **Conor Durkan**[9], **Julius Vetter**[1,2], **Benjamin Kurt Miller**[10], **Maternus Herold**[3,11,12], **Abolfazl Ziaeemehr**[13], **Matthijs Pals**[1,2], **Theo Gruner**[14], **Sebastian Bischoff**[1,2,15], **Nastya Krouglova**[16,17], **Richard Gao**[1,2], **Janne K Lappalainen**[1,2], **Bálint Mucsányi**[1,2,18], **Felix Pei**[19], **Auguste Schulz**[1,2], **Zinovia Stefanidi**[1,2], **Pedro Rodrigues**[20], **Cornelius Schröder**[1,2], **Faried Abu Zaid**[3], **Jonas Beck**[2,21], **Jaivardhan Kapoor**[1,2], **David S. Greenberg**[22,23], **Pedro J. Gonçalves**[17,24], and **Jakob H. Macke**[1,2,25¶]

1 Machine Learning in Science, University of Tübingen **2** Tübingen AI Center **3** TransferLab, appliedAI Institute for Europe **4** ML Colab, Cluster ML in Science, University of Tübingen **5** Google Research **6** Helmholtz-Zentrum Dresden-Rossendorf **7** Université Paris-Saclay, INRIA, CEA, Palaiseau, France **8** Robert Bosch GmbH **9** School of Informatics, University of Edinburgh **10** University of Amsterdam **11** Research and Innovation Center, BMW Group **12** Institute for Applied Mathematics and Scientific Computing, University of the Bundeswehr Munich, Germany **13** Aix Marseille, INSERM, INS, France **14** TU Darmstadt, hessian.AI, Germany **15** University Hospital Tübingen and M3 Research Center **16** Faculty of Science, B-3000, KU Leuven, Belgium **17** VIB-Neuroelectronics Research Flanders (NERF) and imec, Belgium **18** Methods of Machine Learning, University of Tübingen **19** Neuroscience Institute, Carnegie Mellon University **20** Université Grenoble Alpes, INRIA, CNRS, Grenoble INP, LJK, France **21** Hertie Institute for AI in Brain Health, University of Tübingen **22** Institute of Coastal Systems - Analysis and Modeling **23** Helmholtz AI **24** Departments of Computer Science Electrical Engineering, KU Leuven, Belgium **25** Department Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.07754

Software

- Review
- Repository
- Archive

Editor: Sébastien Boisgérault

Reviewers:

- @arnauqb
- @francois-rozet

Submitted: 18 October 2024
Published: 26 March 2025

Abstract

Scientists and engineers use simulators to model empirically observed phenomena. However, tuning the parameters of a simulator to ensure its outputs match observed data presents a significant challenge. Simulation-based inference (SBI) addresses this by enabling Bayesian inference for simulators, identifying parameters that match observed data and align with prior knowledge. Unlike traditional Bayesian inference, SBI only needs access to simulations from the model and does not require evaluations of the likelihood-function. In addition, SBI algorithms do not require gradients through the simulator, allow for massive parallelization of simulations, and can perform inference for different observations without further simulations or training, thereby amortizing inference. Over the past years, we have developed, maintained, and extended `sbi`, a PyTorch-based package that implements Bayesian SBI algorithms based on neural networks. The `sbi` toolkit implements a wide range of inference methods, neural network architectures, sampling methods, and diagnostic tools. In addition, it provides well-tested default settings but also offers flexibility to fully customize every step of the simulation-based inference workflow. Taken together, the `sbi` toolkit enables scientists and engineers to apply state-of-the-art SBI methods to black-box simulators, opening up new possibilities for aligning simulations with empirically observed data.

Statement of need

Bayesian inference is a principled approach for determining parameters consistent with empirical observations: Given a prior over parameters, a forward-model (defining the likelihood), and observations, it returns a posterior distribution. The posterior distribution captures the entire space of parameters that are compatible with the observations and the prior and it quantifies parameter uncertainty. When the forward-model is given by a stochastic simulator, Bayesian inference can be challenging: (1) the forward-model can be slow to evaluate, making algorithms that rely on sequential evaluations of the likelihood (such as Markov-Chain Monte-Carlo, MCMC) impractical, (2) the simulator can be non-differentiable, prohibiting the use of gradient-based MCMC or variational inference (VI) methods, and (3) likelihood-evaluations can be intractable, meaning that we can only generate samples from the model, but not evaluate their likelihoods.

Recently, simulation-based inference (SBI) algorithms based on neural networks have been developed to overcome these limitations (Hermans et al., 2020; Papamakarios et al., 2019; Papamakarios & Murray, 2016). Unlike classical methods from Approximate Bayesian Computation (ABC, Sisson et al. (2018)), these methods use neural networks to learn the relationship between parameters and simulation outputs. Neural SBI algorithms (1) allow for massive parallelization of simulations (in contrast to sequential evaluations in MCMC methods), (2) do not require gradients through the simulator, and (3) do not require evaluations of the likelihood, but only samples from the simulator. Finally, many of these algorithms allow for *amortized* inference, that is, after a large upfront cost of simulating data for the training phase, they can return the posterior distribution for any observation without requiring further simulations or retraining.

To aid in the effective application of these algorithms to a wide range of problems, we developed the `sbi` toolkit. `sbi` implements a variety of state-of-the-art SBI algorithms, offering both high-level interfaces, extensive documentation and tutorials for practitioners, as well as low-level interfaces for experienced users and SBI researchers (giving full control over simulations, the training loop, and the sampling procedure). Since the original release of the `sbi` package (Tejero-Cantero et al., 2020), the community of contributors has expanded significantly, resulting in a large number of improvements that have made `sbi` more flexible, performant, and reliable. `sbi` now supports a wider range of amortized and sequential inference methods, neural network architectures (including normalizing flows, flow- and score-matching, and various embedding network architectures), samplers (including MCMC, variational inference, importance sampling, and rejection sampling), diagnostic tools, visualization tools, and a comprehensive set of tutorials on how to use these features.

The `sbi` package is already used extensively by the machine learning research community (Boelts et al., 2022; Deistler, Gonçalves, et al., 2022; Dirmeier et al., 2023; Dyer et al., 2022b; Gloeckler et al., 2023, 2022, 2024; Hermans et al., 2022; Linhart et al., 2024; Muratore et al., 2022; Spurio Mancini et al., 2023; Wiqvist et al., 2021) but has also fostered the application of SBI in various fields of research (Avecilla et al., 2022; Bernaerts et al., 2023; Boelts et al., 2023; Bondarenko et al., 2023; Confavreux et al., 2023; Deistler, Macke, et al., 2022; Dingeldein et al., 2023; Dyer et al., 2022a; Gao et al., 2024; Groschner et al., 2022; Hahn & Melchior, 2022; Hashemi et al., 2023; Jin et al., 2023; Lemos et al., 2024; Lowet et al., 2023; Mishra-Sharma & Cranmer, 2022; Myers-Joseph et al., 2024; Rößler et al., 2023; Wang et al., 2024).

Description

`sbi` is a flexible and extensive toolkit for running simulation-based Bayesian inference workflows. `sbi` supports any kind of (offline) simulator and prior, a wide range of inference methods, neural networks, and samplers, as well as diagnostic methods and analysis tools (Figure 1).

The Journal of Open Source Software

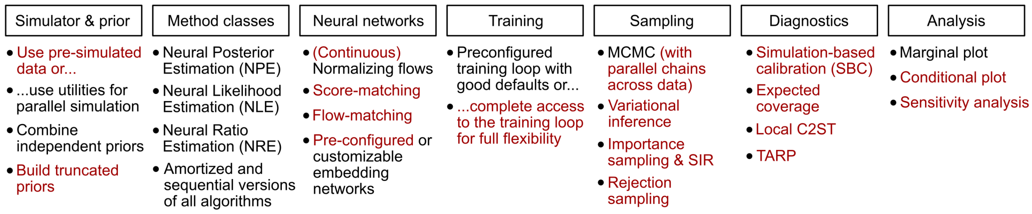

Figure 1: Features of the `sbi` package. Components that were added since the initial release described in Tejero-Cantero et al. (2020) are marked in red.

A significant challenge in making SBI algorithms accessible to a broader community lies in accommodating diverse and complex simulators, as well as varying degrees of flexibility in each step of the inference process. To address this, `sbi` provides pre-configured defaults for all inference methods, but also allows full customization of every step in the process (including simulation, training, sampling, diagnostics and analysis).

Simulator & prior: The `sbi` toolkit requires only simulation parameters and simulated data as input, without needing direct access to the simulator itself. However, if the simulator can be provided as a Python callable, `sbi` can optionally parallelize running the simulations from a given prior using `joblib` (Varoquaux, 2008). Additionally, `sbi` can automatically handle failed simulations or missing values, it supports both discrete and continuous parameters and observations (or mixtures thereof) and it provides utilities to flexibly define priors.

Methods: `sbi` implements a wide range of neural network-based SBI algorithms, among them Neural Posterior Estimation (NPE) with various conditional estimators, Neural Likelihood Estimation (NLE), and Neural Ratio Estimation (NRE). Each of these methods can be run either in an *amortized* mode, where the neural network is trained once on a set of pre-existing simulation results and then performs inference on *any* observation without further simulations or retraining, or in a *sequential* mode where inference is focused on one observation to improve simulation efficiency with active learning, running simulations with parameters likely to have resulted in the observation.

Neural networks and training: `sbi` implements a wide variety of state-of-the-art conditional density estimators for NPE and NLE, including normalizing flows (Greenberg et al., 2019; Papamakarios et al., 2021) (via `nflows` (Durkan et al., 2019) and `zuko` (Rozet, 2023)), diffusion models (Geffner et al., 2023; Simons et al., 2023; Song et al., 2021), mixture density networks (Bishop, 1994), and flow matching (Lipman et al., 2023; Wildberger et al., 2023) (via `zuko`), as well as ensembles of any of these networks. `sbi` also implements a large set of embedding networks that can automatically learn summary statistics of (potentially) high-dimensional simulation outputs (including multilayer perceptrons, convolutional networks, and permutation-invariant networks). The neural networks can be trained with a pre-configured training loop with established default values, but `sbi` also allows full access over the training loop when desired.

Sampling: For NLE and NRE, `sbi` implements a large range of samplers, including MCMC (with chains vectorized across observations), variational inference, rejection sampling, or importance sampling, as well as wrappers to use MCMC samplers from Pyro and PyMC (Abril-Pla et al., 2023; Bingham et al., 2019). `sbi` can perform inference for single observations or for multiple *i.i.d.* observations, and can use importance sampling to correct for potential inaccuracies in the posterior if the likelihood is available.

Diagnostics and analysis: The `sbi` toolkit also implements a large set of diagnostic tools, such as simulation-based calibration (SBC) (Talts et al., 2018), expected coverage (Deistler, Gonçalves, et al., 2022; Hermans et al., 2022), local C2ST (Linhart et al., 2024), and TARP (Lemos et al., 2023). Additionally, `sbi` offers visualization tools for the posterior, including marginal and conditional corner plots to visualize high-dimensional distributions, calibration plots, and wrappers for Arviz (Kumar et al., 2019) diagnostic plots.

With sbi, our goal is to advance scientific discovery and computational engineering by making Bayesian inference accessible to a broad range of models, including those with inaccessible likelihoods, and to a broader range of users, including both machine learning researchers and domain practitioners. We have created an open architecture and embraced community-driven development practices to encourage collaboration with other machine learning researchers and applied scientists to join us in this long-term vision.

Related software

Simulation-based inference methods implemented in the sbi package require only access to simulated data, which can also be generated offline in other programming languages or frameworks. This sets sbi apart from toolboxes for traditional Bayesian inference, such as MCMC-based methods (Abril-Pla et al., 2023; Bingham et al., 2019; Gelman et al., 2015), which rely on likelihood evaluations, and from probabilistic programming languages (e.g., Pyro (Bingham et al., 2019), NumPyro (Phan et al., 2019), Stan (Gelman et al., 2015), or Turing.jl (Ge et al., 2018)), which typically require the simulator to be differentiable and implemented within their respective frameworks (Quera-Bofarull et al., 2023).

Since the original release of the sbi package, several other packages that implement neural network-based SBI algorithms have emerged. The lampe (Rozet et al., 2021) package offers neural posterior and neural ratio estimation, primarily targeting SBI researchers with a low-level API and full flexibility over the training loop. Its development has stopped in favor of the sbi project in July 2024. The BayesFlow package (Radev et al., 2023) focuses on a set of amortized SBI algorithms based on posterior and likelihood estimation that have been developed in the respective research labs (Radev et al., 2020). The swyft package (undark-lab, 2023) specializes in algorithms based on neural ratio estimation. The sbijax package (Dirmeier et al., 2024) implements a set of inference methods in JAX.

Author contributions

This work represents a collaborative effort with contributions from a large and diverse team. Author contributions are categorized as follows: Jan Boelts and Michael Deistler are the current maintainers and lead developers of the sbi package and contributed equally to this work. Manuel Gloeckler, Álvaro Tejero-Cantero, Jan-Matthis Lueckmann, and Guy Moss have made substantial and sustained core contributions to the codebase and project direction. Peter Steinbach, Thomas Moreau, Fabio Muratore, Julia Linhart, and Conor Durkan have made major contributions to specific features or aspects of the package. All other authors listed have contributed to the sbi package through code, documentation, or discussions.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF, projects "Simalesam", FKZ 01IS21055 A-B and "DeepHumanVision", FKZ: 031L0197B, and the Tübingen AI Center FKZ: 01IS18039A), the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, PN 390727645) and SFB1233 (PN 276693517), SFB 1089 (PN 227953431), SPP 2041 (PN 34721065), SPP 2041 "Computational Connectomics", SPP 2298-2 (PN 543917411), SFB 1233 "Robust Vision", and Germany's Excellence Strategy EXC-Number 2064/1/Project number 390727645, the "Certification and Foundations of Safe Machine Learning Systems in Healthcare" project funded by the Carl Zeiss Foundation, the Else Kröner Fresenius Stiftung (Project "ClinbrAIn"), and the European Union (ERC, "DeepCoMechTome", ref. 101089288). CD was supported by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and

Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh. BKM is part of the ELLIS PhD program, receiving travel support from the ELISE mobility program which has received funding from the European Union's Horizon 2020 research and innovation programme under ELISE grant agreement No 951847. DSG is supported by Helmholtz AI. JL is a recipient of the Pierre-Aguilar Scholarship and thankful for the funding of the Capital Fund Management (CFM). ANK is supported by an FWO grant (G097022N). TG was supported by "Third Wave of AI", funded by the Excellence Program of the Hessian Ministry of Higher Education, Science, Research and Art. TM and PLCR were supported from a national grant managed by the French National Research Agency (Agence Nationale de la Recherche) attributed to the ExaDoST project of the NumPEx PEPR program, under the reference ANR-22-EXNU-0004. PS is supported by the Helmholtz Association Initiative and Networking Fund through the Helmholtz AI platform grant. MD, MG, GM, JV, MP, SB, JKL, AS, ZS, JB are members of the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

References

Abril-Pla, O., Andreani, V., Carroll, C., Dong, L., Fonnesbeck, C. J., Kochurov, M., Kumar, R., Lao, J., Luhmann, C. C., Martin, O. A., & others. (2023). PyMC: A modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, *9*, e1516. https://doi.org/10.7717/peerj-cs.1516

Avecilla, G., Chuong, J. N., Li, F., Sherlock, G., Gresham, D., & Ram, Y. (2022). Neural networks enable efficient and accurate simulation-based inference of evolutionary parameters from adaptation dynamics. *PLoS Biology*, *20*(5), e3001633. https://doi.org/10.1371/journal.pbio.3001633

Bernaerts, Y., Deistler, M., Gonçalves, P. J., Beck, J., Stimberg, M., Scala, F., Tolias, A. S., Macke, J., Kobak, D., & Berens, P. (2023). Combined statistical-mechanistic modeling links ion channel genes to physiology of cortical neuron types. *bioRxiv*, 2023–2003. https://doi.org/10.1101/2023.03.02.530774

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., & Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, *20*, 28:1–28:6. https://doi.org/10.48550/arXiv.1810.09538

Bishop, C. M. (1994). Mixture density networks. *Technical Report. Aston University, Birmingham*.

Boelts, J., Harth, P., Gao, R., Udvary, D., Yáñez, F., Baum, D., Hege, H.-C., Oberlaender, M., & Macke, J. H. (2023). Simulation-based inference for efficient identification of generative models in computational connectomics. *PLOS Computational Biology*, *19*(9), e1011406. https://doi.org/10.1101/2023.01.31.526269

Boelts, J., Lueckmann, J.-M., Gao, R., & Macke, J. H. (2022). Flexible and efficient simulation-based inference for models of decision-making. *Elife*, *11*, e77220. https://doi.org/10.7554/eLife.77220

Bondarenko, V., Nikolaev, M., Kromm, D., Belousov, R., Wolny, A., Blotenburg, M., Zeller, P., Rezakhani, S., Hugger, J., Uhlmann, V., & others. (2023). Embryo-uterine interaction coordinates mouse embryogenesis during implantation. *The EMBO Journal*, *42*(17), e113280. https://doi.org/10.15252/embj.2022113280

Confavreux, B., Ramesh, P., Gonçalves, P. J., Macke, J. H., & Vogels, T. (2023). Meta-learning families of plasticity rules in recurrent spiking networks using simulation-based inference. *Advances in Neural Information Processing Systems*, *36*, 13545–13558.

Deistler, M., Gonçalves, P. J., & Macke, J. H. (2022). Truncated proposals for scalable and hassle-free simulation-based inference. In A. H. Oh, A. Agarwal, D. Belgrave, & K. Cho (Eds.), *Advances in neural information processing systems*. https://doi.org/10.48550/arXiv.2210.04815

Deistler, M., Macke, J. H., & Gonçalves, P. J. (2022). Energy-efficient network activity from disparate circuit parameters. *Proceedings of the National Academy of Sciences*, *119*(44), e2207632119. https://doi.org/10.1073/pnas.2207632119

Dingeldein, L., Cossio, P., & Covino, R. (2023). Simulation-based inference of single-molecule force spectroscopy. *Machine Learning: Science and Technology*, *4*(2), 025009. https://doi.org/10.1016/j.bpj.2022.11.920

Dirmeier, S., Albert, C., & Perez-Cruz, F. (2023). Simulation-based inference using surjective sequential neural likelihood estimation. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2308.01054

Dirmeier, S., Ulzega, S., Mira, A., & Albert, C. (2024). *Simulation-based inference with the python package sbijax*. https://arxiv.org/abs/2409.19435

Durkan, C., Bekasov, A., Papamakarios, G., & Murray, I. (2019). Nflows: Normalizing flows in PyTorch. In *GitHub repository*. https://github.com/bayesiains/nflows; GitHub.

Dyer, J., Cannon, P., Farmer, J. D., & Schmon, S. (2022a). Black-box bayesian inference for economic agent-based models. *arXiv Preprint arXiv:2202.00625*. https://doi.org/10.48550/arXiv.2202.00625

Dyer, J., Cannon, P., Farmer, J. D., & Schmon, S. M. (2022b). Calibrating agent-based models to microdata with graph neural networks. *ICML 2022 Workshop AI for Agent-Based Modelling*. https://doi.org/10.48550/arXiv.2206.07570

Gao, R., Deistler, M., Schulz, A., Gonçalves, P. J., & Macke, J. H. (2024). Deep inverse modeling reveals dynamic-dependent invariances in neural circuit mechanisms. *bioRxiv*, 2024–2008. https://doi.org/10.1101/2024.08.21.608969

Ge, H., Xu, K., & Ghahramani, Z. (2018). Turing: A language for flexible probabilistic inference. *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, 1682–1690. http://proceedings.mlr.press/v84/ge18b.html

Geffner, T., Papamakarios, G., & Mnih, A. (2023). Compositional score modeling for simulation-based inference. *International Conference on Machine Learning*, 11098–11116. https://doi.org/10.48550/arXiv.2209.14249

Gelman, A., Lee, D., & Guo, J. (2015). Stan: A probabilistic programming language for bayesian inference and optimization. *Journal of Educational and Behavioral Statistics*, *40*(5), 530–543. https://doi.org/10.3102/1076998615606113

Gloeckler, M., Deistler, M., & Macke, J. H. (2023). Adversarial robustness of amortized bayesian inference. *International Conference on Machine Learning*, 11493–11524. https://doi.org/10.48550/arXiv.2305.14984

Gloeckler, M., Deistler, M., & Macke, J. H. (2022). Variational methods for simulation-based inference. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2203.04176

Gloeckler, M., Deistler, M., Weilbach, C. D., Wood, F., & Macke, J. H. (2024). All-in-one simulation-based inference. *Forty-First International Conference on Machine Learning*. https://doi.org/10.48550/arXiv.2404.09636

Greenberg, D., Nonnenmacher, M., & Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. *International Conference on Machine Learning*, 2404–2414. https://doi.org/10.48550/arXiv.1905.07488

Groschner, L. N., Malis, J. G., Zuidinga, B., & Borst, A. (2022). A biophysical account of multiplication by a single neuron. *Nature*, *603*(7899), 119–123. https://doi.org/10.1038/s41586-022-04428-3

Hahn, C., & Melchior, P. (2022). Accelerated bayesian SED modeling using amortized neural posterior estimation. *The Astrophysical Journal*, *938*(1), 11. https://doi.org/10.3847/1538-4357/ac7b84

Hashemi, M., Vattikonda, A. N., Jha, J., Sip, V., Woodman, M. M., Bartolomei, F., & Jirsa, V. K. (2023). Amortized bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators. *Neural Networks*, *163*, 178–194. https://doi.org/10.1016/j.neunet.2023.03.040

Hermans, J., Begy, V., & Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. *International Conference on Machine Learning*, 4239–4248. https://doi.org/10.48550/arXiv.1903.04057

Hermans, J., Delaunoy, A., Rozet, F., Wehenkel, A., & Louppe, G. (2022). A crisis in simulation-based inference? Beware, your posterior approximations can be unfaithful. *Transactions on Machine Learning Research*.

Jin, H., Verma, P., Jiang, F., Nagarajan, S. S., & Raj, A. (2023). Bayesian inference of a spectral graph model for brain oscillations. *NeuroImage*, *279*, 120278. https://doi.org/10.1016/j.neuroimage.2023.120278

Kumar, R., Carroll, C., Hartikainen, A., & Martin, O. (2019). ArviZ a unified library for exploratory analysis of bayesian models in python. *Journal of Open Source Software*, *4*(33), 1143. https://doi.org/10.21105/joss.01143

Lemos, P., Coogan, A., Hezaveh, Y., & Perreault-Levasseur, L. (2023). Sampling-based accuracy testing of posterior estimators for general inference. *International Conference on Machine Learning*, 19256–19273. https://doi.org/10.48550/arXiv.2302.03026

Lemos, P., Parker, L., Hahn, C., Ho, S., Eickenberg, M., Hou, J., Massara, E., Modi, C., Dizgah, A. M., Blancard, B. R.-S., & others. (2024). Field-level simulation-based inference of galaxy clustering with convolutional neural networks. *Physical Review D*, *109*(8), 083536. https://doi.org/10.1103/physrevd.109.083536

Linhart, J., Gramfort, A., & Rodrigues, P. (2024). L-c2st: Local diagnostics for posterior approximations in simulation-based inference. *Advances in Neural Information Processing Systems*, *36*. https://doi.org/10.48550/arXiv.2306.03580

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., & Le, M. (2023). Flow matching for generative modeling. *The Eleventh International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2210.02747

Lowet, E., Sheehan, D. J., Chialva, U., Pena, R. D. O., Mount, R. A., Xiao, S., Zhou, S. L., Tseng, H., Gritton, H., Shroff, S., & others. (2023). Theta and gamma rhythmic coding through two spike output modes in the hippocampus during spatial navigation. *Cell Reports*, *42*(8). https://doi.org/10.1016/j.celrep.2023.112906

Mishra-Sharma, S., & Cranmer, K. (2022). Neural simulation-based inference approach for characterizing the galactic center γ-ray excess. *Physical Review D*, *105*(6), 063017. https://doi.org/10.1103/PhysRevD.105.063017

Muratore, F., Gruner, T., Wiese, F., Belousov, B., Gienger, M., & Peters, J. (2022). Neural posterior domain randomization. *Conference on Robot Learning*, 1532–1542.

Myers-Joseph, D., Wilmes, K. A., Fernandez-Otero, M., Clopath, C., & Khan, A. G. (2024). Disinhibition by VIP interneurons is orthogonal to cross-modal attentional modulation in primary visual cortex. *Neuron*, *112*(4), 628–645. https://doi.org/10.1016/j.neuron.2023.11.006

Papamakarios, G., & Murray, I. (2016). Fast ε-free inference of simulation models with bayesian conditional density estimation. *Advances in Neural Information Processing Systems*, *29*. https://doi.org/10.48550/arXiv.1605.06376

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, *22*(57), 1–64. https://doi.org/10.48550/arXiv.1912.02762

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. https://doi.org/10.48550/arXiv.1805.07226

Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv Preprint arXiv:1912.11554*. https://doi.org/10.48550/arXiv.1912.11554

Quera-Bofarull, A., Dyer, J., Calinescu, A., Farmer, J. D., & Wooldridge, M. (2023). BlackBIRDS: Black-box inference foR differentiable simulators. *Journal of Open Source Software*, *8*(89), 5776. https://doi.org/10.21105/joss.05776

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., & Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(4), 1452–1466. https://doi.org/10.1109/tnnls.2020.3042395

Radev, S. T., Schmitt, M., Schumacher, L., Elsemüller, L., Pratz, V., Schälte, Y., Köthe, U., & Bürkner, P.-C. (2023). BayesFlow: Amortized Bayesian workflows with neural networks. *Journal of Open Source Software*, *8*(89), 5702. https://doi.org/10.21105/joss.05702

Rößler, N., Jungenitz, T., Sigler, A., Bird, A., Mittag, M., Rhee, J. S., Deller, T., Cuntz, H., Brose, N., Schwarzacher, S. W., & others. (2023). Skewed distribution of spines is independent of presynaptic transmitter release and synaptic plasticity, and emerges early during adult neurogenesis. *Open Biology*, *13*(8), 230063. https://doi.org/10.1098/rsob.230063

Rozet, F. (2023). Zuko - normalizing flows in PyTorch. In *GitHub repository*. https://github.com/probabilists/zuko; GitHub.

Rozet, F., Delaunoy, A., Miller, B., & others. (2021). LAMPE: Likelihood-free amortized posterior estimation. *Statistical Software*.

Simons, J., Sharrock, L., Liu, S., & Beaumont, M. (2023). Neural score estimation: Likelihood-free inference with conditional score based diffusion models. *Fifth Symposium on Advances in Approximate Bayesian Inference*. https://doi.org/10.48550/arXiv.2210.04872

Sisson, S. A., Y., F., & A., B. M. (2018). Overview of ABC. In *Handbook of approximate bayesian computation*. CRC Press, Taylor & Francis Group. https://doi.org/10.1201/9781315117195

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2011.13456

Spurio Mancini, A., Docherty, M., Price, M., & McEwen, J. (2023). Bayesian model comparison for simulation-based inference. *RAS Techniques and Instruments*, *2*(1), 710–722. https://doi.org/10.1093/rasti/rzad051

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating bayesian inference algorithms with simulation-based calibration. *arXiv Preprint arXiv:1804.06788*. https://doi.org/10.48550/arXiv.1804.06788

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

undark-lab. (2023). *Swyft: A system for scientific simulation-based inference at scale*. https://github.com/undark-lab/swyft.

Varoquaux, G. (2008). Joblib. In *GitHub repository*. https://github.com/joblib/joblib; GitHub.

Wang, X., Kelly, R. P., Jenner, A. L., Warne, D. J., & Drovandi, C. (2024). *A comprehensive guide to simulation-based inference in computational biology*. https://doi.org/10.2139/ssrn.4982890

Wildberger, J. B., Dax, M., Buchholz, S., Green, S. R., Macke, J. H., & Schölkopf, B. (2023). Flow matching for scalable simulation-based inference. *Thirty-Seventh Conference on Neural Information Processing Systems*. https://doi.org/10.48550/arXiv.2305.17161

Wiqvist, S., Frellsen, J., & Picchini, U. (2021). Sequential neural posterior and likelihood approximation. *arXiv Preprint arXiv:2102.06522*. https://doi.org/10.48550/arXiv.2102.06522