Flow Matching for Scalable Simulation-Based Inference

Jonas Wildberger [*1] **Maximilian Dax** [*1] **Simon Buchholz** [*1]
Stephen R. Green [2] **Jakob H. Macke** [1 3] **Bernhard Schölkopf** [1]

Abstract

Neural posterior estimation methods based on discrete normalizing flows have become established tools for simulation-based inference (SBI), but scaling them to high-dimensional problems can be challenging. Building on recent advances in generative modeling, we here present flow matching posterior estimation (FMPE), a technique for SBI using continuous normalizing flows. Like diffusion models, and in contrast to discrete flows, flow matching allows for unconstrained architectures, providing enhanced flexibility for complex data modalities. Flow matching, therefore, enables exact density evaluation, fast training, and seamless scalability to large architectures—making it ideal for SBI. To showcase the improved scalability of our approach, we apply it to a challenging astrophysics problem: for gravitational-wave inference, FMPE outperforms methods based on comparable discrete flows, reducing training time by 30% with substantially improved accuracy.

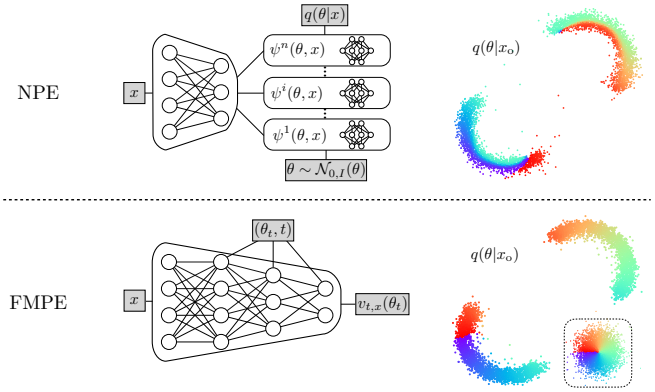

Figure 1. Comparison of network architectures (left) and flow trajectories (right). Discrete flows (NPE, top) require a specialized architecture for the density estimator. Continuous flows (FMPE, bottom) are based on a vector field parametrized with an unconstrained architecture. FMPE uses this additional flexibility to put an enhanced emphasis on the conditioning data x, which in the SBI context is typically high dimensional and in a complex domain. Further, the optimal transport path produces simple flow trajectories from the base distribution (inset) to the target.

1. Introduction

The ability to readily represent Bayesian posteriors of arbitrary complexity using neural networks would herald a revolution in scientific data analysis. Such networks could be trained using simulated data and used for amortized inference across observations—bringing tractable inference and speed to a myriad of scientific models. Thanks to innovative architectures such as normalizing flows (Rezende & Mohamed, 2015; Papamakarios et al., 2021), approaches to neural simulation-based inference (SBI) (Cranmer et al., 2020) have seen remarkable progress in recent years. Here, we show that modern approaches to deep generative modeling (particularly flow matching) deliver substantial improvements in simplicity, flexibility and scaling when adapted to SBI.

The Bayesian approach to data analysis is to compare observations to models via the posterior distribution $p(\theta|x)$. This gives our degree of belief that model parameters θ gave rise to an observation x, and is proportional to the model likelihood $p(x|\theta)$ times the prior $p(\theta)$. One is typically interested in representing the posterior in terms of a collection of samples, however obtaining these through standard likelihood-based algorithms can be challenging for intractable or expensive likelihoods. In such cases, SBI offers an alternative based instead on *data simulations* $x \sim p(x|\theta)$. Combined with deep generative modeling, SBI becomes a powerful paradigm for scientific inference (Cranmer et al., 2020). Neural posterior estimation (NPE) (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019), for instance, trains a conditional density estimator $q(\theta|x)$ to approximate the posterior, allowing for rapid sampling and density estimation for any x consistent with the

[*]Equal contribution [1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]University of Nottingham, Nottingham, United Kingdom [3]University of Tübingen, Tübingen, Germany. Correspondence to: Jonas Wildberger <wildberger.jonas@tuebingen.mpg.de>.

ICML 2023 Workshop on Machine Learning for Astrophysics, Honolulu, Hawaii, USA. PMLR 202, 2023.

training distribution.

The NPE density estimator $q(\theta|x)$ is commonly taken to be a (discrete) normalizing flow (Rezende & Mohamed, 2015; Papamakarios et al., 2021). Normalizing flows transform noise to samples through a discrete sequence of basic transforms. These have been carefully engineered to be invertible with simple Jacobian determinant, enabling efficient maximum likelihood training, while producing expressive $q(\theta|x)$. Although many such discrete flows are universal density approximators (Papamakarios et al., 2021), in practice, they can be challenging to scale to very large networks.

Recent studies (Sharrock et al., 2022; Geffner et al., 2022) propose neural posterior score estimation (NPSE), an approach that models the posterior distribution with score-matching (or diffusion) networks. These techniques were originally developed for generative modeling (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020), achieving state-of-the-art results in many domains, including image generation (Dhariwal & Nichol, 2021; Ho et al., 2022). Like normalizing flows, diffusion models transform noise into samples, but with trajectories parametrized by a *continuous* "time" parameter t. The trajectories solve a stochastic differential equation (Song et al., 2020) (SDE) defined in terms of a vector field v_t, which is trained to match the score of the intermediate distributions p_t. NPSE has several advantages compared to NPE, in particular the freedom to use unconstrained network architectures.

We here propose to use flow matching, another recent technique for generative modeling, for Bayesian inference, an approach we refer to as flow-matching posterior estimation (FMPE). Flow matching is also based on a vector field v_t and thereby also admits flexible network architectures (Fig. 1). For flow matching, however, v_t directly defines the velocity field of deterministic sample trajectories, which solve ordinary differential equations (ODEs). As a consequence, flow matching allows for additional freedom in designing non-diffusion paths such as optimal transport, and provides direct access to the density (Lipman et al., 2022). We apply FMPE to gravitational wave inference (see Section 3) and to a standard benchmark for SBI (see Appendix C).

2. Flow matching posterior estimation

In this section, we give a brief introduction to the flow matching technique (additional information in App. A) and discuss key differences when applying flow matching to simulation based inference instead of generative modelling.

2.1. Flow matching

Flow matching was recently introduced as an efficient approach to train continuous normalizing flows. Continuous flows (Chen et al., 2018) are a family $q_t(\theta|x)$ of distributions parametrized by "time" $t \in [0, 1]$, where $q_0(\theta|x) = q_0(\theta)$ is a fixed base distribution and $q_1(\theta|x) = q(\theta|x)$ the target distribution. They can be generated by a time-dependent vector field $v_{t,x}$ on the sample space describing the velocities of the sample trajectories. The advantage of continuous flows is that $v_{t,x}(\theta)$ can be simply specified by a neural network taking $\mathbb{R}^{n+m+1} \to \mathbb{R}^n$. In contrast, discrete normalizing flows are built using highly restricted bijections.

Continuous flows cannot be efficiently trained by maximizing the likelihood. An alternative training objective for continuous normalizing flows is provided by flow matching (Lipman et al., 2022). This directly regresses $v_{t,x}$ on a vector field $u_{t,x}$ that generates a target probability path $p_{t,x}$. It has the advantage that training does not require integration of ODEs, however it is not immediately clear how to choose $(u_{t,x}, p_{t,x})$, and how to make this objective tractable. The key insight of Lipman et al. (2022) is that, if the path is chosen on a *sample-conditional* basis,[1] then the training objective becomes extremely simple. Indeed, given a sample-conditional probability path $p_t(\theta|\theta_1)$ and a corresponding vector field $u_t(\theta|\theta_1)$, the sample-conditional loss is given by

$$\mathcal{L}_{\text{SCFM}} = \mathbb{E}_{\substack{t\sim\mathcal{U}[0,1],\, x\sim p(x),\\ \theta_1\sim p(\theta|x),\, \theta_t\sim p_t(\theta|\theta_1)}} \left\| v_{t,x}(\theta_t) - u_t(\theta_t|\theta_1) \right\|^2 . \tag{1}$$

Remarkably, minimization of this loss is equivalent to regressing $v_{t,x}(\theta)$ on the *marginal* vector field $u_{t,x}(\theta)$ that generates $p_t(\theta|x)$ (Lipman et al., 2022).

There is a lot of freedom in choosing a sample-conditional path $p_t(\theta|\theta_1)$, here we focus on the optimal transport path introduced by Lipman et al. (2022) where $p_t(\theta|\theta_1) = \mathcal{N}(t\theta_1, \sigma_t^2)$, with $\sigma_t = 1 - (1 - \sigma_{\min})t$ for a small constant $\sigma_{\min}$. The sample-conditional vector field then has the simple form $u_t(\theta|\theta_1) = \sigma_t^{-1}(\theta_1 - (1 - \sigma_{\min})\theta)$.

To apply flow matching to SBI we use Bayes' theorem to make the usual replacement $\mathbb{E}_{p(x)p(\theta|x)} \to \mathbb{E}_{p(\theta)p(x|\theta)}$ in the loss function (1), eliminating the intractable expectation values. This gives the FMPE loss

$$\mathcal{L}_{\text{FMPE}} = \mathbb{E}_{\substack{\theta_1\sim p(\theta), x\sim p(x|\theta_1),\\ t\sim p(t),\, \theta_t\sim p_t(\theta_t|\theta_1)}} \left\| v_{t,x}(\theta_t) - u_t(\theta_t|\theta_1) \right\|^2 , \tag{2}$$

which we minimize using empirical risk minimization over samples $(\theta, x) \sim p(\theta)p(x|\theta)$, i.e., training data is generated by sampling θ from the prior, and then simulating data x corresponding to θ. This is similar to NPE training, but replaces the log likelihood maximization with the sample-conditional flow matching objective. Note that in this expression we also sample $t \sim p(t)$, $t \in [0, 1]$ (see Sec. 2.3), which generalizes the uniform distribution in (6). This provides additional freedom to improve learning in our experiments.

[1] We refer to conditioning on θ_1 as *sample*-conditioning to distinguish from conditioning on x.

2.2. Network architecture

Generative diffusion or flow matching models typically operate on complicated and high dimensional data in the θ space (e.g., images with millions of pixels). One typically uses U-Net (Ronneberger et al., 2015) like architectures, as they provide a natural mapping from θ to a vector field $v(\theta)$ of the same dimension. The dependence on t and an (optional) conditioning vector x is then added on top of this architecture.

For SBI, the data x is often associated with a complicated domain, such as image or time series data, whereas parameters θ are typically low dimensional. In this context, it is therefore useful to build the architecture starting as a mapping from x to $v(x)$ and then add conditioning on θ and t. In practice, one can therefore use any established feature extraction architecture for data in the domain of x, and adjust the dimension of the feature vector to $n = \dim(\theta)$. In our experiments, we found that the (t, θ)-conditioning is best achieved using gated linear units (Dauphin et al., 2017) to the hidden layers of the network (see also Fig. 1); these are also commonly used for conditioning discrete flows on x.

2.3. Re-scaling the time prior

The time prior $\mathcal{U}[0, 1]$ in (6) distributes the training capacity uniformly across t. We observed that this is not always optimal in practice, as the complexity of the vector field may depend on t. For FMPE we therefore sample t in (2) from a power-law distribution $p_\alpha(t) \propto t^{1/(1+\alpha)}$, $t \in [0, 1]$, introducing an additional hyperparameter α. This includes the uniform distribution for $\alpha = 0$, but for $\alpha > 0$, assigns greater importance to the vector field for larger values of t.

3. Gravitational-wave inference

3.1. Background

Gravitational waves (GWs) are ripples of spacetime predicted by Einstein and produced by cosmic events such as the mergers of binary black holes (BBHs). GWs propagate across the universe to Earth, where the LIGO-Virgo-KAGRA observatories measure faint time-series signals embedded in noise. To-date, roughly 90 detections of merging black holes and neutron stars have been made (Abbott et al., 2021c), all of which have been characterized using Bayesian inference to compare against theoretical models.[2] These have yielded insights into the origin and evolution of black holes (Abbott et al., 2021a), fundamental properties of matter and gravity (Abbott et al., 2018; 2021b), and even the expansion rate of the universe (Abbott et al., 2017).

Under reasonable assumptions on detector noise, the GW likelihood is tractable,[3] and inference is typically performed using tools (Veitch et al., 2015; Ashton et al., 2019; Romero-Shaw et al., 2020; Speagle, 2020) based on Markov chain Monte Carlo (Metropolis et al., 1953; Hastings, 1970) or nested sampling (Skilling, 2006) algorithms. This can take from hours to months, depending on the nature of the event and the complexity of the signal model, with a typical analysis requiring up to $\sim 10^8$ likelihood evaluations. The ever-increasing rate of detections means that these analysis times risk becoming a bottleneck. SBI offers a promising solution for this challenge that has been actively studied in the literature (Cuoco et al., 2020; Gabbard et al., 2022; Green et al., 2020; Delaunoy et al., 2020; Green & Gair, 2021; Dax et al., 2021; 2022; Chatterjee et al., 2022; Dax et al., 2023). A fully amortized NPE-based method called DINGO recently achieved accuracies comparable to stochastic samplers with inference times of less than a minute per event (Dax et al., 2021). However, DINGO uses group-equivariant NPE (Dax et al., 2021; 2022) (GNPE), an NPE extension that integrates known conditional symmetries. GNPE, therefore, does not provide a tractable density, which is problematic when verifying and correcting inference results using importance sampling (Dax et al., 2023).

3.2. Experiments

We here apply FMPE to GW inference. As a baseline, we train an NPE network with the settings described in (Dax et al., 2021) with a few minor changes (see Appendix B).[4] This uses an embedding network (Radev et al., 2020) to compress x to a 128-dimensional feature vector, which is then used to condition a neural spline flow (Durkan et al., 2019). The embedding network consists of a learnable linear layer initialized with principal components of GW simulations followed by a series of dense residual blocks (He et al., 2015). This architecture is a powerful feature extractor for GW measurements (Dax et al., 2021). As pointed out in Section 2.2, it is straightforward to reuse such architectures for FMPE, with the following three modifications: (1) we provide the conditioning on (t, θ) to the network via gated linear units in each hidden layer; (2) we change the dimension of the final feature vector to the dimension of θ so that the network parameterizes the conditional vector field $(t, x, \theta) \to v_{t,x}(\theta)$; (3) we increase the number and width of the hidden layers to use the capacity freed up by removing

[2] BBH parameters $\theta \in \mathbb{R}^{15}$ include black-hole masses, spins, and the spacetime location and orientation of the system (see Tab. 2 in the Appendix). We represent x in frequency domain; for two LIGO detectors and complex $f \in [20, 512]$ Hz, $\Delta f = 0.125$ Hz, we have $x \in \mathbb{R}^{15744}$.

[3] Noise is assumed to be stationary and Gaussian, so for frequency-domain data, the GW likelihood $p(x|\theta) = \mathcal{N}(h(\theta)|S_\mathrm{n})(x)$. Here $h(\theta)$ is a theoretical signal model based on Einstein's theory of general relativity, and S_n is the power spectral density of the detector noise.

[4] Our implementation builds on the public DINGO code from https://github.com/dingo-gw/dingo.

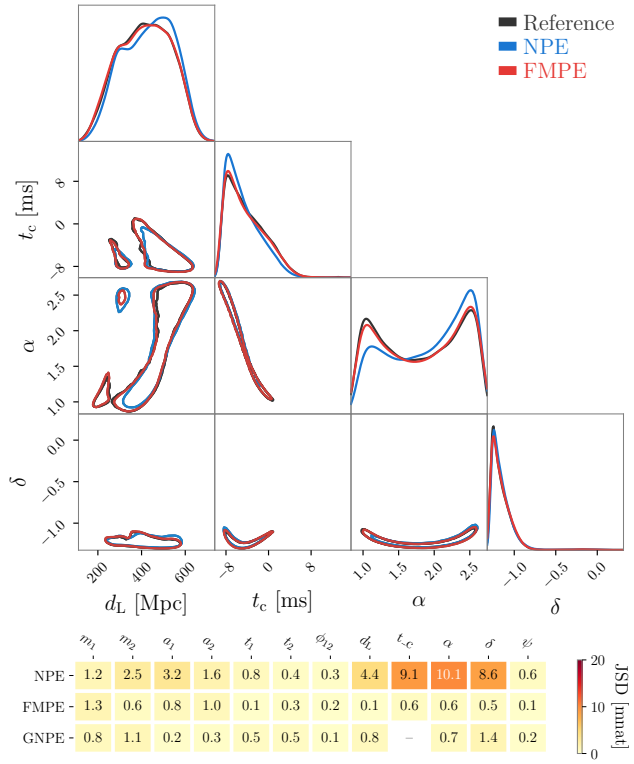

Figure 2. Results for GW150914 (Abbott et al., 2016). Top: Corner plot showing 1D marginals on the diagonal and 2D 50% credible regions. We display four GW parameters (distance d_L, time of arrival t_c, and sky coordinates α, δ); these represent the least accurate NPE parameters. Bottom: Deviation between inferred posteriors and the reference, quantified by the Jensen-Shannon divergence (JSD). The FMPE posterior matches the reference more accurately than NPE, and performs similarly to symmetry-enhanced GNPE. (We do not display GNPE results on the top due to different data conditioning settings in available networks.)

the discrete normalizing flow.

We train the NPE and FMPE networks with $5 \cdot 10^6$ simulations for 400 epochs using a batch size of 4096 on an A100 GPU. The FMPE network ($1.9 \cdot 10^8$ learnable parameters, training takes $\approx$ 2 days) is larger than the NPE network ($1.3 \cdot 10^8$ learnable parameters, training takes $\approx$ 3 days), but trains substantially faster. We evaluate both networks on GW150914 (Abbott et al., 2016), the first detected GW. We generate a reference posterior using the method described in (Dax et al., 2023). Fig. 2 compares the inferred posterior distributions qualitatively and quantitatively in terms of the Jensen-Shannon divergence (JSD) to the reference.[5]

FMPE substantially outperforms NPE in terms of accuracy, with a mean JSD of 0.5 mnat (NPE: 3.6 mnat), and max JSD < 2.0 mnat, an indistinguishability criterion for GW posteriors (Romero-Shaw et al., 2020). We believe that this is related to the network structure as follows. The NPE network allocates roughly two thirds of its parameters to the discrete normalizing flow and only one third to the embedding network (i.e., the feature extractor for x). Since FMPE parameterizes a much simpler vector field, it can devote its network capacity to the interpretation of the high-dimensional $x \in \mathbb{R}^{15744}$, and thereby scales much better to larger networks and achieve much higher accuracy. Remarkably, FMPE accuracy is even comparable to GNPE, which leverages physical symmetries to simplify data and has been validated in a variety of settings (Dax et al., 2021; 2022; 2023; Wildberger et al., 2023).

Finally, we find that the Bayesian evidences inferred with NPE ($\log p(x) = -7667.958 \pm 0.006$) and FMPE ($\log p(x) = -7667.969 \pm 0.005$) are consistent within their statistical uncertainties. A correct evidence is only obtained in importance sampling when the inferred posterior $q(\theta|x)$ covers the entire posterior $p(\theta|x)$ (Dax et al., 2023), indicating that FMPE induces mass-covering posteriors.

4. Conclusions

We introduced flow matching posterior estimation, a new simulation-based inference technique based on continuous normalizing flows. In contrast to existing neural posterior estimation methods, it does not rely on restricted density estimation architectures such as discrete normalizing flows, and instead parametrizes a distribution in terms of a conditional vector field. This enables more flexible network architectures and seamless scaling (like score matching), while enabling flexible path specification and direct access to the posterior density.

On the challenging task of gravitational-wave inference, FMPE substantially outperformed comparable discrete flows, producing samples on par with a method that explicitly leverages symmetries to simplify training. Additionally, flow matching latent spaces are more naturally structured than those of discrete flows, particularly when using paths such as optimal transport. Looking forward, it would be interesting to exploit such structure in designing learning algorithms. This performance and flexibilty underscores the capability of continuous normalizing flows to efficiently solve inverse problems.

[5] We omit the three parameters $\phi_c, \phi_{JL}, \theta_{JN}$ in the evaluation as we use phase marginalization in importance sampling and the reference therefore uses a different basis for these parameters (Dax et al., 2023). For GNPE we report the results from (Dax et al., 2021), which are generated with slightly different data conditioning. Therefore, we do not display the GNPE results in the corner plot, and the JSDs serve only as a rough comparison. The JSD for the t_c parameter is not reported in (Dax et al., 2021) due to a t_c marginalized reference.

References

Abbott, B. et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.*, 116(6): 061102, 2016. doi: 10.1103/PhysRevLett.116.061102.

Abbott, B. P. et al. A gravitational-wave standard siren measurement of the Hubble constant. *Nature*, 551(7678): 85–88, 2017. doi: 10.1038/nature24471.

Abbott, B. P. et al. GW170817: Measurements of neutron star radii and equation of state. *Phys. Rev. Lett.*, 121(16): 161101, 2018. doi: 10.1103/PhysRevLett.121.161101.

Abbott, R. et al. Population Properties of Compact Objects from the Second LIGO-Virgo Gravitational-Wave Transient Catalog. *Astrophys. J. Lett.*, 913(1):L7, 2021a. doi: 10.3847/2041-8213/abe949.

Abbott, R. et al. Tests of general relativity with binary black holes from the second LIGO-Virgo gravitational-wave transient catalog. *Phys. Rev. D*, 103(12):122002, 2021b. doi: 10.1103/PhysRevD.103.122002.

Abbott, R. et al. GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run. *arXiv preprint arXiv:2111.03606*, 11 2021c.

Ashton, G. et al. BILBY: A user-friendly Bayesian inference library for gravitational-wave astronomy. *Astrophys. J. Suppl.*, 241(2):27, 2019. doi: 10.3847/1538-4365/ab06fc.

Bohé, A., Hannam, M., Husa, S., Ohme, F., Pürrer, M., and Schmidt, P. PhenomPv2 – technical notes for the LAL implementation. *LIGO Technical Document, LIGO-T1500602-v4*, 2016. URL https://dcc.ligo.org/LIGO-T1500602/public.

Chatterjee, C., Wen, L., Beveridge, D., Diakogiannis, F., and Vinsen, K. Rapid localization of gravitational wave sources from compact binary coalescences using deep learning. *arXiv preprint arXiv:2207.14522*, 7 2022.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6572–6583, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *Proc. Nat. Acad. Sci.*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Cuoco, E., Powell, J., Cavaglià, M., Ackley, K., Bejger, M., Chatterjee, C., Coughlin, M., Coughlin, S., Easter, P., Essick, R., et al. Enhancing gravitational-wave science with machine learning. *Machine Learning: Science and Technology*, 2(1):011002, 5 2020. doi: 10.1088/2632-2153/abb93a.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *International conference on machine learning*, pp. 933–941. PMLR, 2017.

Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A., and Schölkopf, B. Real-Time Gravitational Wave Science with Neural Posterior Estimation. *Phys. Rev. Lett.*, 127(24):241103, 2021. doi: 10.1103/PhysRevLett.127.241103.

Dax, M., Green, S. R., Gair, J., Deistler, M., Schölkopf, B., and Macke, J. H. Group equivariant neural posterior estimation. In *International Conference on Learning Representations*, 11 2022.

Dax, M., Green, S. R., Gair, J., Pürrer, M., Wildberger, J., Macke, J. H., Buonanno, A., and Schölkopf, B. Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference. *Phys. Rev. Lett.*, 130(17): 171403, 2023. doi: 10.1103/PhysRevLett.130.171403.

Delaunoy, A., Wehenkel, A., Hinderer, T., Nissanke, S., Weniger, C., Williamson, A. R., and Louppe, G. Lightning-Fast Gravitational Wave Parameter Inference through Neural Amortization. In *Third Workshop on Machine Learning and the Physical Sciences*, 10 2020.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Dormand, J. and Prince, P. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. ISSN 0377-0427. doi: https://doi.org/10.1016/0771-050X(80)90013-3. URL https://www.sciencedirect.com/science/article/pii/0771050X80900133.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Advances in Neural Information Processing Systems*, pp. 7509–7520, 2019.

Farr, B., Ochsner, E., Farr, W. M., and O'Shaughnessy, R. A more effective coordinate system for parameter estimation of precessing compact binaries from gravitational waves. *Phys. Rev. D*, 90(2):024018, 2014. doi: 10.1103/PhysRevD.90.024018.

Friedman, J. H. On multivariate goodness–of–fit and two–sample testing. *Statistical Problems in Particle Physics, Astrophysics, and Cosmology*, 1:311, 2003.

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

Geffner, T., Papamakarios, G., and Mnih, A. Score modeling for simulation-based inference. *arXiv preprint arXiv:2209.14249*, 2022.

Green, S. R. and Gair, J. Complete parameter inference for GW150914 using deep learning. *Mach. Learn. Sci. Tech.*, 2(3):03LT01, 2021. doi: 10.1088/2632-2153/abfaed.

Green, S. R., Simpson, C., and Gair, J. Gravitational-wave parameter estimation with autoregressive neural network flows. *Phys. Rev. D*, 102(10):104057, 2020. doi: 10.1103/PhysRevD.102.104057.

Greenberg, D., Nonnenmacher, M., and Macke, J. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pp. 2404–2414. PMLR, 2019.

Hannam, M., Schmidt, P., Bohé, A., Haegel, L., Husa, S., Ohme, F., Pratten, G., and Pürrer, M. Simple model of complete precessing black-hole-binary gravitational waveforms. *Phys. Rev. Lett.*, 113: 151101, Oct 2014. doi: 10.1103/PhysRevLett.113.151101. URL https://link.aps.org/doi/10.1103/PhysRevLett.113.151101.

Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57 (1):97–109, 04 1970. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97. URL https://doi.org/10.1093/biomet/57.1.97.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47:1–47:33, 2022. URL http://jmlr.org/papers/v23/21-0635.html.

Khan, S., Husa, S., Hannam, M., Ohme, F., Pürrer, M., Forteza, X. J., and Bohé, A. Frequency-domain gravitational waves from nonprecessing black-hole binaries. II. A phenomenological model for the advanced detector era. *Phys. Rev.*, D93(4):044007, 2016. doi: 10.1103/PhysRevD.93.044007.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *CoRR*, abs/2210.02747, 2022. doi: 10.48550/arXiv.2210.02747. URL https://doi.org/10.48550/arXiv.2210.02747.

Lopez-Paz, D. and Oquab, M. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

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

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 343–351. PMLR, 2021.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Papamakarios, G. and Murray, I. Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Advances in neural information processing systems*, 2016.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL http://jmlr.org/papers/v22/19-1028.html.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. Bayesflow: Learning complex stochastic models with invertible neural networks, 2020.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Romero-Shaw, I. M. et al. Bayesian inference for compact binary coalescences with bilby: validation and application to the first LIGO–Virgo gravitational-wave transient catalogue. *Mon. Not. Roy. Astron. Soc.*, 499(3):3295–3319, 2020. doi: 10.1093/mnras/staa2850.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Sharrock, L., Simons, J., Liu, S., and Beaumont, M. Sequential neural score estimation: Likelihood-free inference with conditional score based diffusion models. *arXiv preprint arXiv:2210.04872*, 2022.

Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833 – 859, 2006. doi: 10.1214/06-BA127. URL https://doi.org/10.1214/06-BA127.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Speagle, J. S. dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493 (3):3132–3158, Feb 2020. ISSN 1365-2966. doi: 10.1093/mnras/staa278. URL http://dx.doi.org/10.1093/mnras/staa278.

Veitch, J., Raymond, V., Farr, B., Farr, W., Graff, P., Vitale, S., et al. Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Phys. Rev.*, D91(4): 042003, 2015. doi: 10.1103/PhysRevD.91.042003.

Wildberger, J., Dax, M., Green, S. R., Gair, J., Pürrer, M., Macke, J. H., Buonanno, A., and Schölkopf, B. Adapting to noise distribution shifts in flow-based gravitational-wave inference. *Phys. Rev. D*, 107(8):084046, 2023. doi: 10.1103/PhysRevD.107.084046.

A. Background

In this section we give a slightly extended version of Section 2.

Normalizing flows. A normalizing flow (Rezende & Mohamed, 2015; Papamakarios et al., 2021) defines a probability distribution $q(\theta|x)$ over parameters $\theta \in \mathbb{R}^n$ in terms of an invertible mapping $\psi_x : \mathbb{R}^n \to \mathbb{R}^n$ from a simple base distribution $q_0(\theta)$,

$$q(\theta|x) = (\psi_x)_* q_0(\theta) = q_0(\psi_x^{-1}(\theta)) \det \left| \frac{\partial \psi_x^{-1}(\theta)}{\partial \theta} \right| , \tag{3}$$

where $(\cdot)_*$ denotes the pushforward operator, and for generality we have conditioned on additional context $x \in \mathbb{R}^m$. Unless otherwise specified, a normalizing flow refers to a *discrete* flow, where ψ_x is given by a composition of simpler mappings with triangular Jacobians, interspersed with shuffling of the θ. This construction results in expressive $q(\theta|x)$ and also efficient density evaluation (Papamakarios et al., 2021).

Continuous normalizing flows. A continuous flow (Chen et al., 2018) also maps from base to target distribution, but is parametrized by a continuous "time" $t \in [0, 1]$, where $q_0(\theta|x) = q_0(\theta)$ and $q_1(\theta|x) = q(\theta|x)$. For each t, the flow is defined by a vector field $v_{t,x}$ on the sample space. This corresponds to the velocity of the sample trajectories,

$$\frac{d}{dt}\psi_{t,x}(\theta) = v_{t,x}(\psi_{t,x}(\theta)), \qquad \psi_{0,x}(\theta) = \theta. \tag{4}$$

We obtain the trajectories $\theta_t \equiv \psi_{t,x}(\theta)$ by integrating this ODE. The final density is given by

$$q(\theta|x) = (\psi_{1,x})_* q_0(\theta) = q_0(\theta) \exp\left(- \int_0^1 \operatorname{div} v_{t,x}(\theta_t)\, \mathrm{d}t \right) , \tag{5}$$

which is obtained by solving the transport equation $\partial_t q_t + \operatorname{div}(q_t v_{t,x}) = 0$.

The advantage of the continuous flow is that $v_{t,x}(\theta)$ can be simply specified by a neural network taking $\mathbb{R}^{n+m+1} \to \mathbb{R}^n$, in which case (4) is referred to as a *neural ODE* (Chen et al., 2018). Since the density is tractable via (5), it is in principle possible to train the flow by maximizing the (log-)likelihood. However, this is often not feasible in practice, since both sampling and density estimation require many network passes to numerically solve the ODE (4).

Flow matching. An alternative training objective for continuous normalizing flows is provided by flow matching (Lipman et al., 2022). This objective allows us to directly regress $v_{t,x}$ on a vector field $u_{t,x}$ that generates a target probability path $p_{t,x}$. Then training does not require integration of ODEs, however it is not immediately clear how to construct a suitable path $(u_{t,x}, p_{t,x})$. The key insight of (Lipman et al., 2022) is that, if the path is chosen on a *sample-conditional* basis, then the training objective becomes extremely simple. Indeed, given a sample-conditional probability path $p_t(\theta|\theta_1)$ and a corresponding vector field $u_t(\theta|\theta_1)$, we specify the sample-conditional flow matching loss as

$$\mathcal{L}_{\text{SCFM}} = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x\sim p(x),\, \theta_1\sim p(\theta|x),\, \theta_t\sim p_t(\theta|\theta_1)} \left\| v_{t,x}(\theta_t) - u_t(\theta_t|\theta_1) \right\|^2 . \tag{6}$$

Remarkably, minimization of this loss is equivalent to regressing $v_{t,x}(\theta)$ on the *marginal* vector field $u_{t,x}(\theta)$ that generates $p_t(\theta|x)$ (Lipman et al., 2022). Note that in this expression, the x-dependence of $v_{t,x}(\theta)$ is picked up via the expectation value, with the sample-conditional vector field independent of x.

There exists considerable freedom in choosing a sample-conditional path. Ref. (Lipman et al., 2022) introduces the family of Gaussian paths

$$p_t(\theta|\theta_1) = \mathcal{N}(\theta|\mu_t(\theta_1), \sigma_t(\theta_1)^2 I_n), \tag{7}$$

where the time-dependent means $\mu_t(\theta_1)$ and standard deviations $\sigma_t(\theta_1)$ can be freely specified (subject to boundary conditions[6]). We focus on the optimal transport paths introduced by Lipman et al. (2022). They are defined by $\mu_t(\theta_1) = t\theta_1$ and $\sigma_t(\theta_1) = 1 - (1 - \sigma_{\min})t$. The sample-conditional vector field then has the simple form

$$u_t(\theta|\theta_1) = \frac{\theta_1 - (1 - \sigma_{\min})\theta}{1 - (1 - \sigma_{\min})t}. \tag{8}$$

[6]The sample-conditional probability path should be chosen to be concentrated around θ_1 at $t = 1$ (within a small region of size $\sigma_{\min}$) and to be the base distribution at $t = 0$.

hyperparameter	values
residual blocks	$2048, 4096 \times 3, 2048 \times 3, 1024 \times 6, 512 \times 8, 256 \times 10, 128 \times 5, 64 \times 3, 32 \times 3, 16 \times 3$
residual blocks (t, θ) embedding	$16, 32, 64, 128, 256$
batch size	4096
learning rate	5.e-4
α (for time prior)	1
residual blocks	$2048 \times 2, 1024 \times 4, 512 \times 4, 256 \times 4, 128 \times 4, 64 \times 3, 32 \times 3, 16 \times 3$
residual blocks (t, θ) embedding	$16, 32, 64, 128, 256$
batch size	4096
learning rate	5.e-4
α (for time prior)	1

Table 1. Hyperparameters for the FMPE models used in the main text (top) and in the ablation study (bottom, see Fig. 3). The network is composed of a sequence of residual blocks, each consisting of two fully-connected hidden layers, with a linear layer between each pair of blocks. The ablation network is the same as the embedding network that feeds into the NPE normalizing flow.

Neural posterior estimation (NPE). NPE is an SBI method that directly fits a density estimator $q(\theta|x)$ (usually a normalizing flow) to the posterior $p(\theta|x)$ (Papamakarios & Murray, 2016; Lueckmann et al., 2017; Greenberg et al., 2019). NPE trains with the maximum likelihood objective $\mathcal{L}_{\text{NPE}} = -\mathbb{E}_{p(\theta)p(x|\theta)} \log q(\theta|x)$, using Bayes' theorem to simplify the expectation value with $\mathbb{E}_{p(x)p(\theta|x)} \to \mathbb{E}_{p(\theta)p(x|\theta)}$. During training, $\mathcal{L}_{\text{NPE}}$ is estimated based on an empirical distribution consisting of samples $(\theta, x) \sim p(\theta)p(x|\theta)$. Once trained, NPE can perform inference for every new observation using $q(\theta|x)$, thereby *amortizing* the computational cost of simulation and training across all observations. NPE further provides exact density evaluations of $q(\theta|x)$. Both of these properties are crucial for the physics application in section 3, so we aim to retain these properties with FMPE.

B. Gravitational-wave inference

We here provide the missing details and additional results for the gravitational wave inference problem analyzed in Section 3.

B.1. Network architecture and hyperparameters

Compared to NPE with normalizing flows, FMPE allows for generally simpler architectures, since the output of the network is simply a vector field. This also holds for NPSE (model also defined by a vector) and NRE (defined by a scalar). Our FMPE architecture builds on the embedding network developed in (Dax et al., 2021), however we extend the network capacity by adding more residual blocks (Tab. 1, top panel). For the (t, θ)-conditioning we use gated linear units applied to each residual block, as described in Section 2.2. We also use a small residual network to embed (t, θ) before applying the gated linear units.

In this Appendix we also perform an ablation study, using the *same* embedding network as the NPE network (Tab. 1, bottom panel). For this configuration, we additionally study the effect of conditioning on (t, θ) starting from different layers of the main residual network.

B.2. Data settings

We use the data settings described in (Dax et al., 2021), with a few minor modifications. In particular, we use the waveform model IMRPhenomPv2 (Hannam et al., 2014; Khan et al., 2016; Bohé et al., 2016) and the prior displayed in Tab. 2. Compared to (Dax et al., 2021), we reduce the frequency range from $[20, 1024]$ Hz to $[20, 512]$ Hz to reduce the computational load for data preprocessing. We also omit the conditioning on the detector noise power spectral density (PSD) introduced in (Dax et al., 2021) as we evaluate on a single GW event. Preliminary tests show that the performance with PSD conditioning is similar to the results reported in this paper. All changes to the data settings have been applied to FMPE and the NPE baselines alike to enable a fair comparison.

Description	Parameter	Prior
component masses	m_1, m_2	$[10, 120]$ $M_\odot$, $m_1 \geq m_2$
chirp mass	$M_c = (m_1 m_2)^{\frac{3}{5}}/(m_1 + m_2)^{\frac{1}{5}}$	$[20, 120]$ $M_\odot$ (constraint)
mass ratio	$q = m_2/m_1$	[0.125, 1.0] (constraint)
spin magnitudes	a_1, a_2	$[0, 0.99]$
spin angles	$\theta_1, \theta_2, \phi_{12}, \phi_{JL}$	standard as in (Farr et al., 2014)
time of coalescence	t_c	$[-0.03, 0.03]$ s
luminosity distance	d_L	$[100, 1000]$ Mpc
reference phase	ϕ_c	$[0, 2\pi]$
inclination	θ_{JN}	$[0, \pi]$ uniform in sine
polarization	ψ	$[0, \pi]$
sky position	α, β	uniform over sky

Table 2. Priors for the astrophysical binary black hole parameters. Priors are uniform over the specified range unless indicated otherwise. Our models infer the mass parameters in the basis (M_c, q) and marginalize over the phase parameter ϕ_c.

B.3. Additional results

Tab. B.3 displays the inference times for FMPE and NPE. NPE requires only a single network pass to produce samples and (log-)probabilities, whereas many forwards passes are needed for FMPE to solve the ODE with a specific level of accuracy. A significant portion of the additional time required for calculating (log-)probabilities in conjunction with the samples is spent on computing the divergence of the vector field, see Eq. (5). Fig. 3 presents a comparison of the FMPE performance

Figure 3. Jensen-Shannon divergence between inferred posteriors and the reference posteriors for GW150914 (Abbott et al., 2016). We compare two FMPE models with the same architecture as the NPE embedding network, see Tab. 1 bottom panel. For the model in the first row, the GLU conditioning of (θ, t) is only applied before the final 128-dim blocks. The model in the middle row is given the context after the very first 2048 block.

using networks of the same hidden dimensions as the NPE embedding network (Tab. 1 bottom panel). This comparison includes an ablation study on the timing of the (t, θ) GLU-conditioning. In the top-row network, the (t, θ) conditioning is applied only after the 256-dimensional blocks. In contrast, the middle-row network receives (t, θ) immediately after the initial residual block. With FMPE we can achieve performance comparable to NPE, while having only $\approx 1/3$ of the network size (most of the NPE network parameters are in the flow). This suggests that parameterizing the target distribution in terms of a vector field requires less learning capacity, compared to directly learning its density. Delaying the (t, θ) conditioning until the final layers impairs performance. However, the number of FLOPs at inference is considerably reduced, as the context embedding can be cached and a network pass only involves the few layers with the (t, θ) conditioning. Consequently, there's a trade-off between accuracy and inference speed, which we will explore in a greater scope in future work.

C. SBI benchmark

We further evaluate FMPE on ten tasks included in the benchmark presented in (Lueckmann et al., 2021), ranging from simple Gaussian toy models to more challenging SBI problems from epidemiology and ecology, with varying dimensions for parameters ($\dim(\theta) \in [2, 10]$) and observations ($\dim(x) \in [2, 100]$). For each task, we train three separate FMPE models with simulation budgets $N \in \{10^3, 10^4, 10^5\}$. We use a simple network architecture consisting of fully connected residual

	Network Passes	Inference Time (per batch)
FMPE (sample only)	248	26s
FMPE (sample and log probs)	350	352s
NPE (sample and log probs)	1	1.5s

Table 3. Inference times per batch for FMPE and NPE on a single Nvidia A100 GPU, using the training batch size of 4096. We solve the ODE for FMPE using the `dopri5` discretization (Dormand & Prince, 1980) with absolute and relative tolerances of 1e-7. For FMPE, generation of the (log-)probabilities additionally requires the computation of the divergence, see equation (5). This needs additional memory and therefore limits the maximum batch size that can be used at inference.

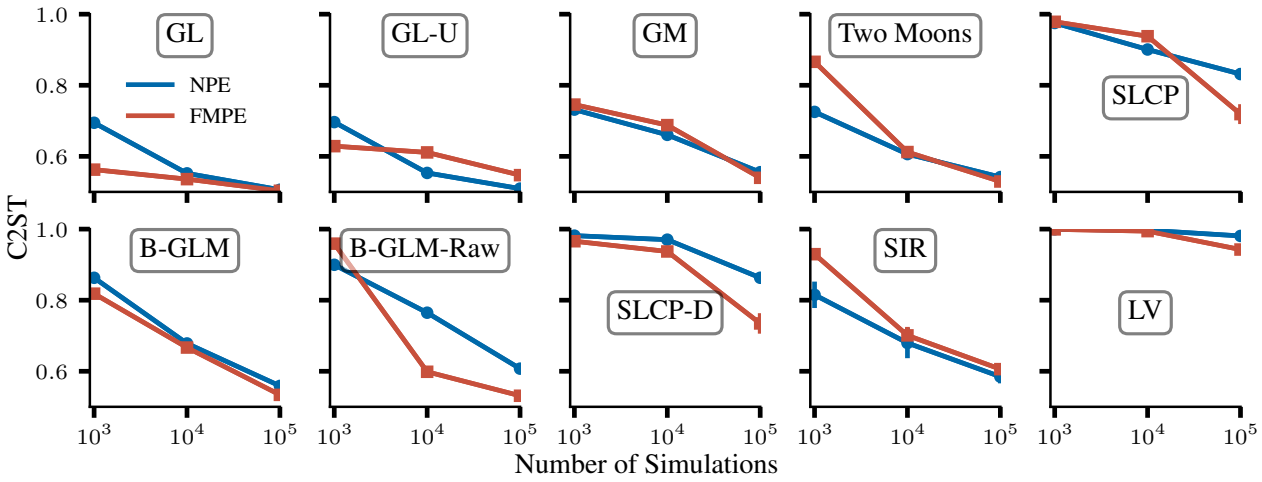

Figure 4. Comparison of FMPE with NPE, a standard SBI method, across 10 benchmark tasks (Lueckmann et al., 2021).

blocks (He et al., 2015) to parameterize the conditional vector field. For the two tasks with $\dim(x) = 100$ (B-GLM-Raw, SLCP-D), we condition on (t, θ) via gated linear units as described in Section 2.2. For the remaining tasks with $\dim(x) \leq 10$ we concatenate (t, θ, x) instead. We reserve 5% of the simulations for validation.

For each task and simulation budget, we evaluate the model with the lowest validation loss by comparing $q(\theta|x)$ to the reference posteriors $p(\theta|x)$ provided in (Lueckmann et al., 2021) for ten different observations x in terms of the C2ST score (Friedman, 2003; Lopez-Paz & Oquab, 2016). This performance metric is computed by training a classifier to discriminate inferred samples $\theta \sim q(\theta|x)$ from reference samples $\theta \sim p(\theta|x)$. The C2ST score is then the test accuracy of this classifier, ranging from 0.5 (best) to 1.0. We observe that FMPE exhibits comparable performance to an NPE baseline model for most tasks and outperforms on several (Fig. 4). As NPE is one of the highest ranking methods for many tasks in the benchmark, these results show that FMPE indeed performs competitively with other existing SBI methods.

As NPE and FMPE both directly target the posterior with a density estimator (in contrast to most other SBI methods), observed differences can be primarily attributed to their different approaches for density estimation. Interestingly, a great performance improvement of FMPE over NPE is observed for SLCP with a large simulation budget ($N = 10^5$). The SLCP task is specifically designed to have a simple likelihood but a complex posterior, and the FMPE performance underscores the enhanced flexibility of the FMPE density estimator.