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. We show that FMPE achieves competitive performance on an established SBI benchmark, and then demonstrate its improved scalability on a challenging scientific problem: for gravitational-wave inference, FMPE outperforms methods based on comparable discrete flows, reducing training time by 30% with substantially improved accuracy. Our work underscores the potential of FMPE to enhance performance in challenging inference scenarios, thereby paving the way for more advanced scientific applications.

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 sam-

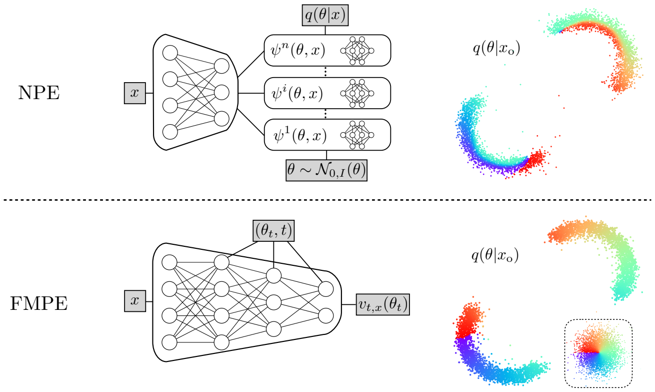


Figure 1. Comparison of network architectures (left) and flow trajectories (right). Discrete flows (NPE, top) require a specialized architecture for the density estimator. Continuous flows (FMPE, bottom) are based on a vector field parametrized with an unconstrained architecture. FMPE uses this additional 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.

[*] 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>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling.

pling and density estimation for any x.

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 evaluate FMPE on a standard SBI benchmark and parameter inference of gravitational waves (see Section 3).

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. In the supplemental material, we additionally investigate mass coverage (App. D).

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.

While continuous flows cannot be efficiently trained by maximizing the likelihood, an alternative training objective 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 $p_{t,x}$. Then, training does not require integration of ODEs, however it is not 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 the training objective becomes extremely simple, if the path is chosen on a *sample-conditional* basis.[1] 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).

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

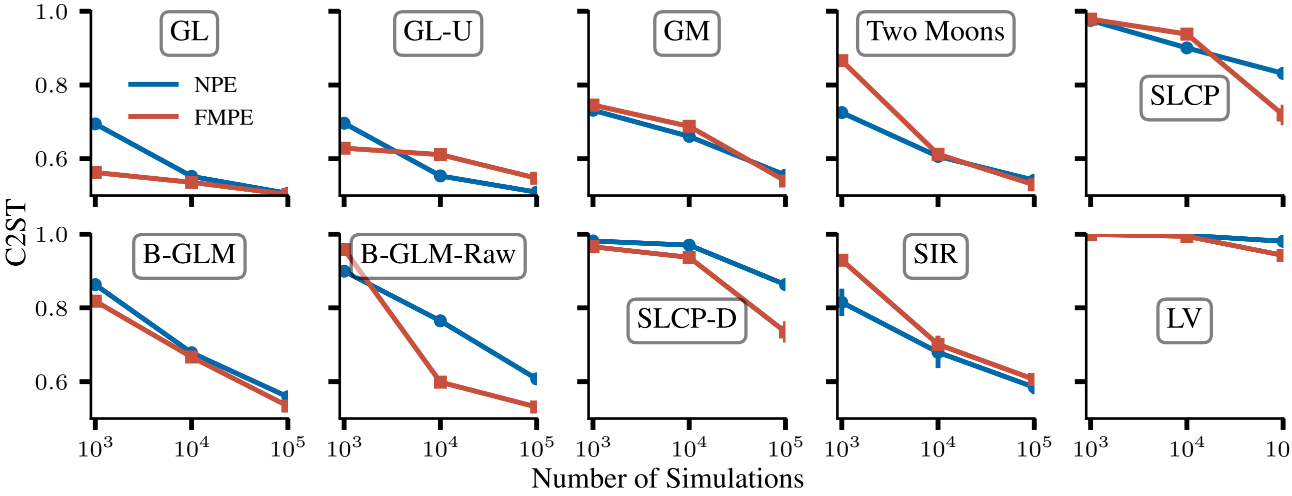


Figure 2. Comparison of FMPE with NPE, a standard SBI method, across 10 benchmark tasks (Lueckmann et al., 2021).

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 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).

2.3. Re-scaling the time prior

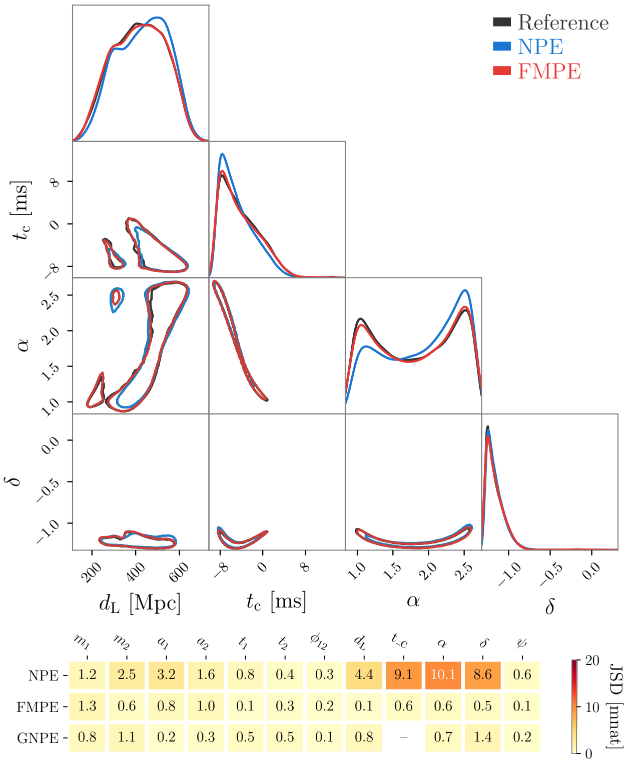
The time prior $\mathcal{U}[0, 1]$ in (6) distributes the training capacity uniformly across t. We observed that this is not always optimal in practice, as the complexity of the vector field may depend on t. For FMPE we therefore sample t in (2) from a power-law distribution $p_\alpha(t) \propto t^{1/(1+\alpha)}$, $t \in [0, 1]$, 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. We empirically found this to improve learning for distributions with sharp bounds (e.g., Two Moons in Section 3.1).

3. Experiments

3.1. SBI Benchmark

We 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 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. 2). 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. Notably, a great performance improvement of FMPE with GLU-conditioning over NPE is observed for B-GLM-Raw and SLCP-D with large simulation budgets



	m_1	m_2	a_1	a_2	t_1	t_2	ϕ_{12}	d_L	t_c	α	δ	ψ
NPE	1.2	2.5	3.2	1.6	0.8	0.4	0.3	4.4	9.1	10.1	8.6	0.6
FMPE	1.3	0.6	0.8	1.0	0.1	0.3	0.2	0.1	0.6	0.6	0.5	0.1
GNPE	0.8	1.1	0.2	0.3	0.5	0.5	0.1	0.8	–	0.7	1.4	0.2

Figure 3. 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 (Dax et al., 2022). (We do not display GNPE results on the top due to different data conditioning settings in available networks.)

($N = 10^4, 10^5$). This underscores the benefit of FMPE to adopt new network architectures according to the structure of the task at hand. For a comparison with the standard architecture, see App. C.

3.2. Gravitational-wave inference

Gravitational waves (GWs) are ripples of spacetime predicted by Einstein and produced by cataclysmic 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. These are analyzed using Bayesian inference to draw conclusions about BBH parameters including masses, spins and location. For further details see App. B.

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 App. B).[2] 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. 3 compares the inferred posterior distributions qualitatively and quantitatively in terms of the Jensen-Shannon divergence (JSD) to the reference.

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).

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.

We evaluated FMPE on a set of 10 benchmark tasks and found competitive or better performance compared to other simulation-based inference methods. 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

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

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.

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.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

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.

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.

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.

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.

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

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.

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.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5):1–19, 2019.

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.

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.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

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. Preliminaries

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 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 $(u_{t,x}, p_{t,x})$. The key insight of (Lipman et al., 2022) is that, if the path is chosen on a *sample-conditional* basis,[3] 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[4]). For our experiments, we focus on the optimal transport paths defined by $\mu_t(\theta_1) = t\theta_1$ and $\sigma_t(\theta_1) = 1 - (1 - \sigma_{\text{min}})t$ (also introduced in (Lipman et al., 2022)). The sample-conditional vector field then has the simple form

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

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

[4] The sample-conditional probability path should be chosen to be concentrated around θ_1 at $t = 1$ (within a small region of size σ_{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. 4). 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 3.2, 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 3.2.

B.1. Network architecture and hyperparameters

Our network architecture extends the NPE network with settings described in (Dax et al., 2021). 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 and use a small residual network to embed (t, θ) before applying the gated linear units; (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 the discrete normalizing flow (Tab. 1, top panel).

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

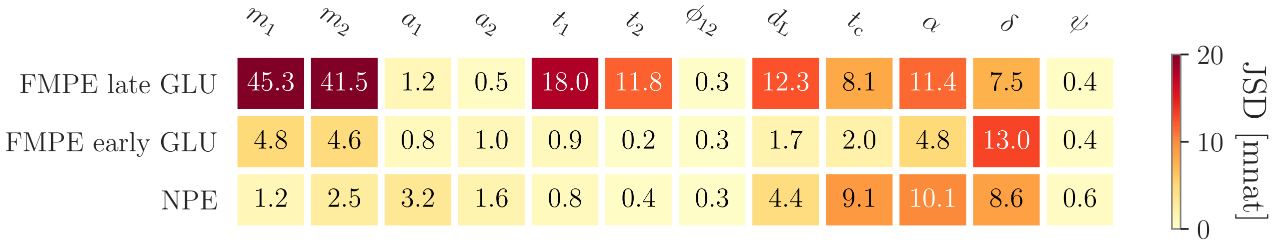
Description	Parameter	Prior
component masses	m_1, m_2	$[10, 120]$ $\mathrm{M}_\odot$, $m_1 \geq m_2$
chirp mass	$M_c = (m_1 m_2)^{\frac{3}{5}}/(m_1+m_2)^{\frac{1}{5}}$	$[20, 120]$ $\mathrm{M}_\odot$ (constraint)
mass ratio	$q = m_2/m_1$	[0.125, 1.0] (constraint)
spin magnitudes	a_1, a_2	$[0, 0.99]$
spin 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. We represent x in frequency domain; for two LIGO detectors and complex $f \in [20, 512]$ Hz, $\Delta f = 0.125$ Hz, we have $x \in \mathbb{R}^{15744}$.

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.

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. 4 presents a comparison of the FMPE performance 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.



	m_1	m_2	a_1	a_2	t_1	t_2	ϕ_{12}	d_L	t_c	α	δ	ψ
FMPE late GLU	45.3	41.5	1.2	0.5	18.0	11.8	0.3	12.3	8.1	11.4	7.5	0.4
FMPE early GLU	4.8	4.6	0.8	1.0	0.9	0.2	0.3	1.7	2.0	4.8	13.0	0.4
NPE	1.2	2.5	3.2	1.6	0.8	0.4	0.3	4.4	9.1	10.1	8.6	0.6

Figure 4. 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.

	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.

Table 4. Sweep values for the hyperparamters for the SBI benchmark. We split the configurations according to simulation budgets, e.g. for 1000 simulations, we only swept over smaller values for network size and batch size. The network architecture has a diamond shape, with increasing layer width from smallest to largest and then decreasing to the output dimension. Each block consists of two fully-connected residual layers.

hyperparameter	sweep values
hidden dimensions	2^n for $n \in \{4, \ldots, 10\}$
number of blocks	$10, \ldots, 18$
batch size	2^n for $n \in \{2, \ldots, 9\}$
learning rate	1.e-3, 5.e-4, 2.e-4, 1.e-4
α (for time prior)	-0.25, -0.5, 0, 1, 4

C. SBI Benchmark

In this section, we collect missing details and additional results for the analysis of the SBI benchmark in Section 3.1.

C.1. Network architecture and hyperparameters

For each task and simulation budget in the benchmark, we perform a mild hyperparameter optimization. We sweep over the batch size and learning rate (which is particularly important as the simulation budgets differ by orders of magnitudes), the network size and the α parameter for the time prior defined in Section 2.3 (see Tab. 4 for the specific values). We reserve 5% of the simulation budget for validation and choose the model with the best validation loss across all configurations.

C.2. Additional results

We here provide various additional results for the SBI benchmark. First, we compare the performance of FMPE and NPE when using the Maximum Mean Discrepancy metric (MMD). The results can be found in Fig. 5. FMPE shows superior performance to NPE for most tasks and simulation budgets. Compared to the C2ST scores in Fig. 2 the improvement shown by FMPE in MMD is more substantial.

Fig. 6 compares the FMPE results with the optimal transport path from the main text with a comparable score matching model using the Variance Preserving diffusion path (Song et al., 2020). The score matching results were obtained using the same batch size, network size and learning rate as the FMPE network, while optimizing for $\beta_{\text{min}} \in \{0.1, 1, 4\}$ and $\beta_{\text{max}} \in \{4, 7, 10\}$. FMPE with the optimal transport path clearly outperforms the score-based model on almost all configurations.

Finally we compare FMPE using the architecture proposed in Section 2.2 with (t, θ)-conditioning via gated linear units to FMPE with a naive architecture operating directly on the concatenated (t, θ, x) vector, see Fig. 7. For the two displayed tasks the context dimension $\dim(x) = 100$ is much larger than the parameter dimension $\dim(\theta) \in \{5, 10\}$, and there is a clear performance gain in using the GLU conditioning. Our interpretation is that the low dimensionality of (t, θ) means that it is not well-learned by the network when simply concatenated with x.

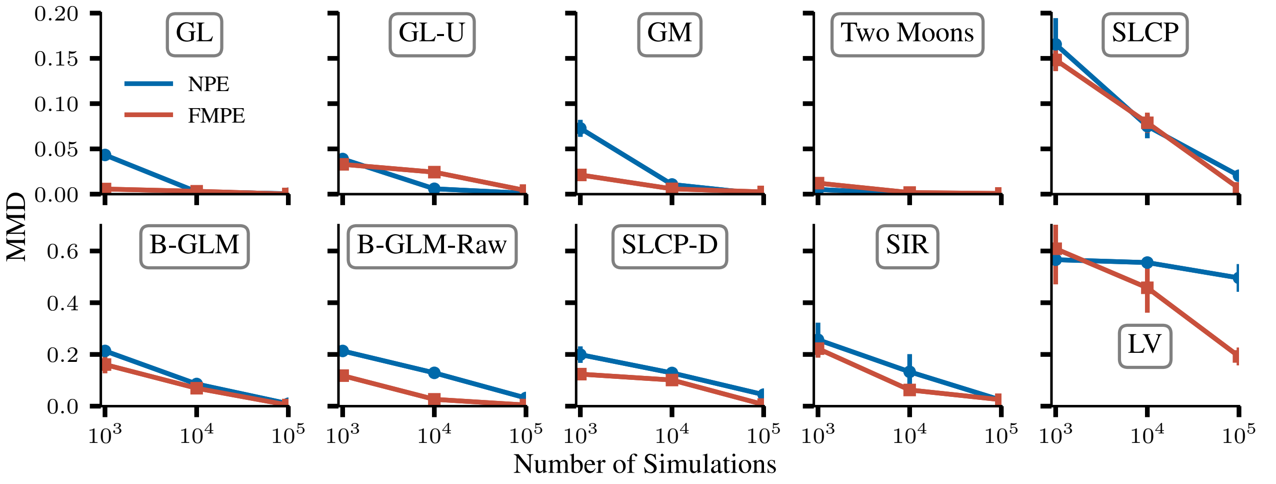


Figure 5. Comparison of FMPE and NPE performance across 10 SBI benchmarking tasks (Lueckmann et al., 2021). We here quantify the deviation in terms of the Maximum Mean Discrepancy (MMD) as an alternative metric to the C2ST score used in Fig. 2. MMD can be sensitive to its hyperparameters (Lueckmann et al., 2021), so we use the C2ST score as a primary performance metric.

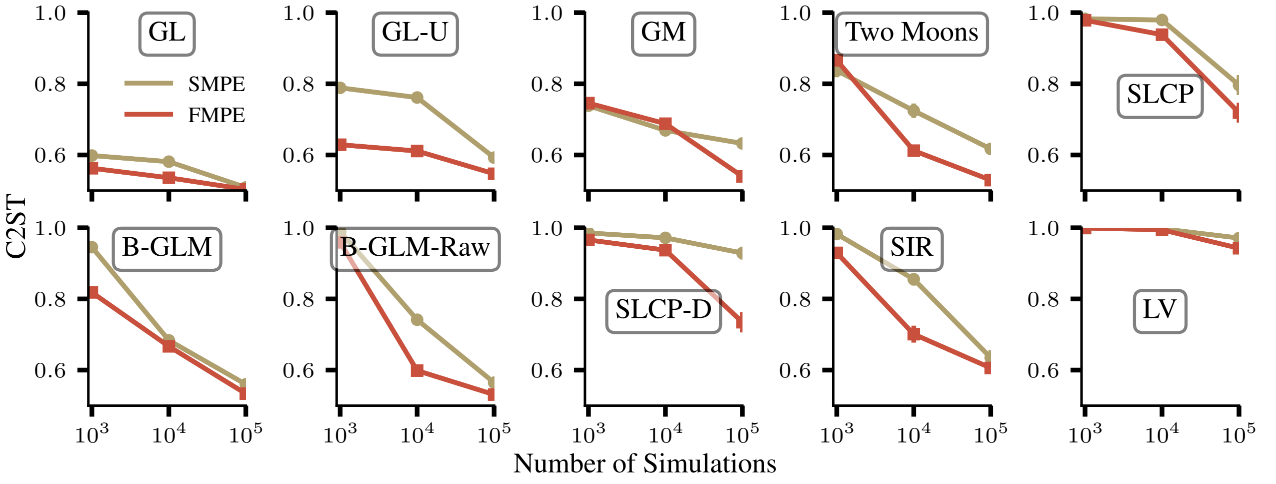


Figure 6. Comparison of FMPE with the optimal transport path (as used throughout the main paper) with comparable models trained with a Variance Preserving diffusion path (Song et al., 2020) by regressing on the score (SMPE). Note that the SMPE baseline shown here is not directly comparable to NPSE (Sharrock et al., 2022; Geffner et al., 2022), as this method uses Langevin steps, which reduces the dependence of the results on the vector field for small t (at the cost of a tractable density).

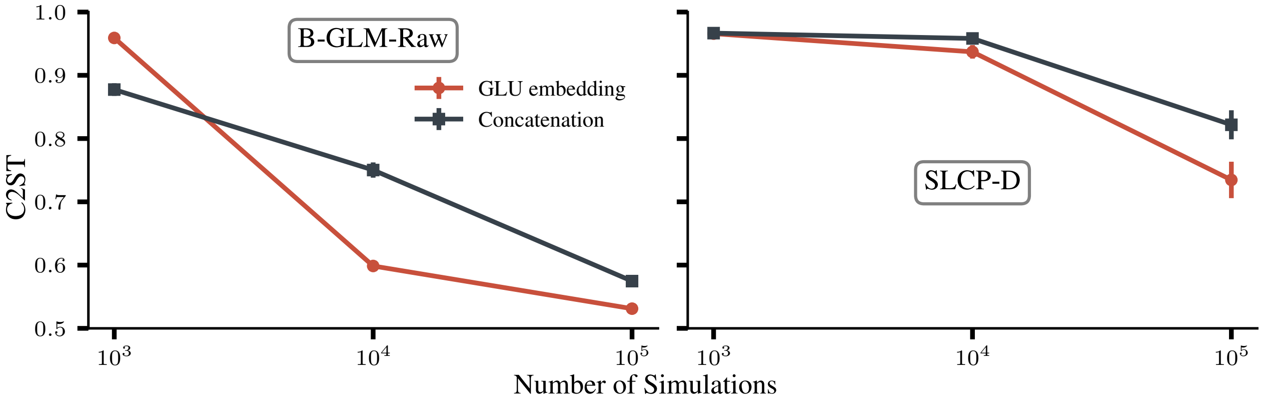


Figure 7. Comparison of the architecture proposed in Section 2.2 with gated linear units for the (t, θ)-conditioning (red) and a naive architecture based on a simple concatenation of (t, θ, x) (black). FMPE with the proposed architecture performs substantially better.

D. Mass coverage of FMPE

In this section we investigate the mass coverage of FMPE from a theoretical viewpoint. We here focus on our main results and delegate the technical arguments to App. E and F.

As we show in our experiments, trained FMPE models $q(\theta|x)$ can achieve excellent results in approximating the true posterior $p(\theta|x)$. However, it is not generally possible to achieve *exact* agreement due to limitations in training budget and network capacity. It is therefore important to understand how inaccuracies manifest. Whereas sample quality is the main criterion for generative modeling, for scientific applications one is often interested in the overall shape of the distribution. In particular, an important question is whether $q(\theta|x)$ is *mass-covering*, i.e., whether it contains the full support of $p(\theta|x)$. This minimizes the risk to falsely rule out possible explanations of the data. It also allows us to use importance sampling if the likelihood $p(x|\theta)$ of the forward model can be evaluated, which can be used for precise estimation of the posterior (Müller et al., 2019; Dax et al., 2023).

Consider first the mass-covering property for NPE. NPE directly minimizes the forward KL divergence $\mathrm{D}_{\mathrm{KL}}(p(\theta|x)||q(\theta|x))$, and thereby provides probability-mass covering results. Therefore, even if NPE is not accurately trained, the estimate $q(\theta|x)$ should cover the entire support of the posterior $p(\theta|x)$ and the failure to do so can be observed in the validation loss. As an illustration in an unconditional setting, we observe that a unimodal Gaussian q fitted to a bimodal target distribution p captures both modes when using the forward KL divergence $\mathrm{D}_{\mathrm{KL}}(p||q)$, but only a single mode when using the backwards direction $\mathrm{D}_{\mathrm{KL}}(q||p)$ (Fig. 8).

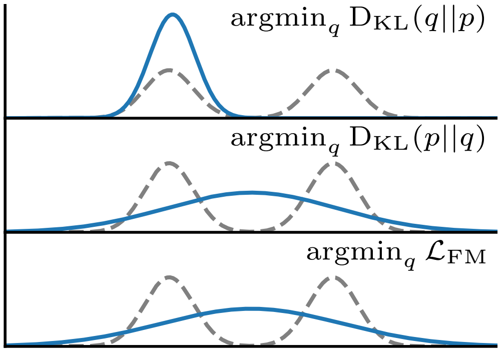


Figure 8. A Gaussian (blue) fitted to a bimodal distribution (gray) with various objectives.

As a motivation we now consider a similar settings for FMPE. We fit a Gaussian (i.e., restricting distributions to be Gaussian) flow-matching model $q(\theta) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ to the same bimodal target, in this case, parametrizing the vector field as

$$v_t(\theta) = \frac{(\sigma_t^2 + (t\hat{\sigma})^2 - \sigma_t)\theta_t + t\hat{\mu} \cdot \sigma_t}{t \cdot (\sigma_t^2 + (t\hat{\sigma})^2)} \tag{9}$$

(see App. E), we also obtain a mass-covering distribution when fitting the learnable parameters $(\hat{\mu}, \hat{\sigma})$ via (6). This provides some indication that the flow matching objective induces mass-covering behavior, and leads us to investigate the more general question of whether the mean squared error between vector fields u_t and v_t bounds the forward KL divergence. Indeed, the former agrees up to constant with the sample-conditional loss (6) (see Sec. A).

We denote the flows of u_t, v_t, by ϕ_t, ψ_t, respectively, and we set $q_t = (\psi_t)_* q_0$, $p_t = (\phi_t)_* q_0$. The precise question then is whether we can bound $\mathrm{D}_{\mathrm{KL}}(p_1||q_1)$ by $\mathrm{MSE}_p(u, v)^\alpha$ for some positive power α. It was already observed in (Albergo et al.,

2023) that this is not true in general, and we provide a simple example to that effect in Lemma F.1 in App. F. Indeed, it was found in (Albergo et al., 2023) that to bound the forward KL divergence we also need to control the Fisher divergence, $\int p_t(\mathrm{d}\theta)(\nabla \ln p_t(\theta) - \nabla q_t(\theta))^2$.

Here we show instead that a bound can be obtained under sufficiently strong regularity assumptions on p_0, u_t, and v_t.

Theorem D.1. *Let $p_0 = q_0$ and assume u_t and v_t are two vector fields whose flows satisfy $p_1 = (\phi_1)_* p_0$ and $q_1 = (\psi_1)_* q_0$. Assume that p_0 is square integrable and satisfies $|\nabla \ln p_0(\theta)| \leq c(1 + |\theta|)$ and u_t and v_t have bounded second derivatives. Then there is a constant $C > 0$ such that (for $\mathrm{MSE}_p(u, v) < 1$))*

$$\mathrm{D}_{\mathrm{KL}}(p_1 || q_1) \leq C\, \mathrm{MSE}_p(u, v)^{\frac{1}{2}}. \tag{10}$$

The proof of this result can be found in App. F. While the regularity assumptions are not guaranteed to hold in practice when v_t is parametrized by a neural net, the theorem nevertheless gives some indication that the flow-matching objective encourages mass coverage. In Section 3.1 and 3.2, this is complemented with extensive empirical evidence that flow matching indeed provides mass-covering estimates.

We remark that it was shown in (Song et al., 2021) that the KL divergence of SDE solutions can be bounded by the MSE of the estimated score function. Thus, the smoothing effect of the noise ensures mass coverage, an aspect that was further studied using the Fokker-Planck equation in (Albergo et al., 2023). For flow matching, imposing the regularity assumption plays a similar role.

E. Gaussian flow

We here derive the form of a vector field $v_t(\theta)$ that restricts the resulting continuous flow to a one dimensional Gaussian with mean $\hat{\mu}$ variance $\hat{\sigma}^2$. With the optimal transport path $\mu_t(\theta) = t\theta_1$, $\sigma_t(\theta) = 1 - (1 - \sigma_{\text{min}})t \equiv \sigma_t$ from (Lipman et al., 2022), the sample-conditional probability path (7) reads

$$p_t(\theta|\theta_1) = \mathcal{N}[t\theta_1, \sigma_t^2](\theta). \tag{11}$$

We set our target distribution

$$q_1(\theta_1) = \mathcal{N}[\hat{\mu}, \hat{\sigma}^2](\theta_1). \tag{12}$$

To derive the marginal probability path and the marginal vector field we need two identities for the convolution $*$ of Gaussian densities. Recall that the convolution of two function is defined by $f * g(x) = \int f(x - y) g(y)\, \mathrm{d}y$. We define the function

$$g_{\mu,\sigma^2}(\theta) = \theta \cdot \mathcal{N}\left[\mu, \sigma^2\right](\theta). \tag{13}$$

Then the following holds

$$\mathcal{N}[\mu_1, \sigma_1^2] * \mathcal{N}[\mu_2, \sigma_2^2] = \mathcal{N}[\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2] \tag{14}$$

$$g_{0,\sigma_1^2} * \mathcal{N}[\mu_2, \sigma_2^2] = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\left(g_{\mu_2,\sigma_1^2+\sigma_2^2} - \mu_2 \mathcal{N}[\mu_2, \sigma_1^2 + \sigma_2^2]\right) \tag{15}$$

Marginal probability paths

Marginalizing over θ_1 in (11) with (12), we find

$$\begin{aligned} p_t(\theta) &= \int p_t(\theta|\theta_1) q(\theta_1)\, d\theta_1 \\ &= \int \mathcal{N}\left[t\theta_1, \sigma_t^2\right](\theta)\, \mathcal{N}\left[\hat{\mu}, \hat{\sigma}^2\right](\theta_1) d\theta_1 \\ &= \int \mathcal{N}\left[0, \sigma_t^2\right](\theta - t\theta_1)\, \mathcal{N}(t\hat{\mu}, (t\hat{\sigma})^2)(t\theta_1) \cdot t\, d\theta_1 \\ &= \int \mathcal{N}\left[0, \sigma_t^2\right](\theta - \theta_1^t)\, \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right](\theta_1^t)\, d\theta_1^t \\ &= \mathcal{N}\left[t\hat{\mu}, \sigma_t^2 + (t\hat{\sigma})^2\right](\theta) \end{aligned} \tag{16}$$

where we defined $\theta_1^t = t\theta_1$ and used (14).

Marginal vector field

We now calculate the marginalized vector field $u_t(\theta)$ based on equation (8) in (Lipman et al., 2022). Using the sample-conditional vector field (8) and the distributions (11) and (12) we find

$$
\begin{aligned}
u_t(\theta) &= \int u_t(\theta|\theta_1) \frac{p_t(\theta|\theta_1) q(\theta_1)}{p_t(\theta)} \, d\theta_1 \\
&= \frac{1}{p_t(\theta)} \int \frac{(\theta_1 - (1-\sigma_{\min})\theta)}{\sigma_t} \cdot \mathcal{N}\left[t\theta_1, \sigma_t^2\right](\theta) \cdot \mathcal{N}\left[\hat{\mu}, \hat{\sigma}^2\right](\theta_1) \, d\theta_1 \\
&= \frac{1}{p_t(\theta)} \int \frac{(\theta_1 - (1-\sigma_{\min})\theta)}{\sigma_t} \cdot \mathcal{N}\left[0, \sigma_t^2\right](\theta - t\theta_1) \cdot \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right](t\theta_1) \cdot t \, d\theta_1 \\
&= \frac{1}{p_t(\theta)} \int \frac{(\theta_1' - (1-\sigma_{\min})t \cdot \theta)}{\sigma_t \cdot t} \cdot \mathcal{N}\left[0, \sigma_t^2\right](\theta - \theta_1') \cdot \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right](\theta_1') \cdot d\theta_1' \\
&= \frac{1}{p_t(\theta)} \int \frac{(-\theta_1'' + (1 - (1-\sigma_{\min})t) \cdot \theta)}{\sigma_t \cdot t} \cdot \mathcal{N}\left[0, \sigma_t^2\right](\theta_1'') \cdot \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right](\theta - \theta_1'') \cdot d\theta_1'' \\
&= \frac{1}{p_t(\theta)} \int \frac{(-\theta_1'' + \sigma_t \cdot \theta)}{\sigma_t \cdot t} \cdot \mathcal{N}\left[0, \sigma_t^2\right](\theta_1'') \cdot \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right](\theta - \theta_1'') \cdot d\theta_1''
\end{aligned}
\tag{17}
$$

where we used the change of variables $\theta_1' = t\theta_1$ and $\theta_1'' = \theta - \theta_1'$. Now we evaluate this expression using (13), then the identities (14) and (15) and the marginal probability (16)

$$
\begin{aligned}
u_t(\theta) &= \frac{-1}{p_t(\theta) \cdot \sigma_t \cdot t} \left(g_{0,\sigma_t^2} * \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right] \right)(\theta) + \frac{\theta}{p_t(\theta) \cdot t} \left(\mathcal{N}\left[0, \sigma_t^2\right] * \mathcal{N}\left[t\hat{\mu}, (t\hat{\sigma})^2\right] \right)(\theta) \\
&= \frac{-1}{p_t(\theta) \cdot \sigma_t \cdot t} \frac{(\theta - t\hat{\mu}) \cdot \sigma_t^2}{\sigma_t^2 + (t\hat{\sigma})^2} \cdot \mathcal{N}\left[t\hat{\mu}, (\sigma_t^2 + (t\hat{\sigma})^2)\right](\theta) + \frac{\theta}{p_t(\theta) \cdot t} \mathcal{N}\left[t\hat{\mu}, (\sigma_t^2 + (t\hat{\sigma})^2)\right](\theta) \\
&= \frac{(\sigma_t^2 + (t\hat{\sigma})^2)\theta - (\theta - t\hat{\mu}) \cdot \sigma_t}{p_t(\theta) \cdot t \cdot (\sigma_t^2 + (t\hat{\sigma})^2)} \cdot p_t(\theta) \\
&= \frac{(\sigma_t^2 + (t\hat{\sigma})^2 - \sigma_t)\theta + t\hat{\mu} \cdot \sigma_t}{t \cdot (\sigma_t^2 + (t\hat{\sigma})^2)}.
\end{aligned}
\tag{18}
$$

By choosing a vector field v_t of the form (18) with learnable parameters $\hat{\mu}, \hat{\sigma}^2$, we can thus define a continuous flow that is restricted to a one dimensional Gaussian.

F. Mass covering properties of continuous flows

In this section, we provide the technical arguments for Section D and also address the mass covering properties of continuous normalizing flows trained using mean squared error more broadly. We give two counterexamples and then prove Theorem D.1. We first introduce some notation. We always assume that the data is distributed according to $p_1(\theta)$. In addition, there is a known and simple base distribution p_0 and we assume that there is a vector field $u_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ that connects p_0 and p_1 in the following sense. We denote by ϕ_t the flow generated by u_t, i.e., ϕ_t satisfies

$$
\partial_t \phi_t(\theta) = u_t(\phi_t(\theta)). \tag{19}
$$

Then we assume that $(\phi_1)_* p_0 = p_1$ and we also define the interpolations $p_t = (\phi_t)_* p_0$.

We do not have access to the ground truth distributions p_t and the vector field u_t but we try to learn a vector field v_t approximating u_t. We denote its flow by ψ_t and we define $q_t = (\psi_t)_* q_0$ and $q_0 = p_0$. We are interested in the mass covering properties of the learned approximation q_1 of p_1. In particular, we want to relate the KL-divergence $\mathrm{D}_{\mathrm{KL}}(p_1||q_1)$ to the mean squared error,

$$
\mathrm{MSE}_p(u, v) = \int_0^1 \mathrm{d}t \int p_t(\mathrm{d}\theta)(u_t(\theta) - v_t(\theta))^2, \tag{20}
$$

of the generating vector fields. The first observation is that without any regularity assumptions on v_t it is impossible to obtain any bound on the KL-divergence in terms of the mean squared error.

Lemma F.1. *For every $\varepsilon > 0$ there are vector field u_t and v_t and a base distribution $p_0 = q_0$ such that*

$$\mathrm{MSE}_p(u, v) < \varepsilon \textit{ and } \mathrm{D}_{\mathrm{KL}}(p_1 || q_1) = \infty. \tag{21}$$

In addition we can construct u_t and v_t such that the support of p_1 is larger than the support of q_1.

Proof. We consider the uniform distribution $p_0 = q_0 \sim \mathcal{U}([-1, 1])$ and the vector fields

$$u_t(\theta) = 0 \tag{22}$$

and

$$v_t(\theta) = \begin{cases} \varepsilon & \text{for } 0 \leq \theta < \varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

As before, let ϕ_t denote the flow of the vector field u_t and similarly ψ_t denote the flow of v_t. Clearly $\phi_t(\theta) = \theta$. On the other hand

$$\psi_t(\theta) = \begin{cases} \min(\theta + \varepsilon t, \varepsilon) & \text{if } 0 \leq \theta < \varepsilon, \\ \theta & \text{otherwise.} \end{cases} \tag{24}$$

In particular

$$\psi_1(\theta) = \begin{cases} \varepsilon & \text{if } 0 \leq \theta < \varepsilon, \\ \theta & \text{otherwise.} \end{cases} \tag{25}$$

This implies that $p_1 = (\phi_1)_* p_0 \sim \mathcal{U}([-1, 1])$. On the other hand $q_1 = (\psi_1)_* q_0$ has support in $[-1, 0] \cup [\varepsilon, 1]$. In particular, the distribution of q_1 is not mass covering with respect to p_1 and $\mathrm{D}_{\mathrm{KL}}(p_1 || q_1) = \infty$. Finally, we observe that the MSE can be arbitrarily small

$$\mathrm{MSE}_p(u, v) = \int_0^1 \mathrm{d}t \int p_t(\mathrm{d}\theta) |u_t(\theta) - v_t(\theta)|^2 = \int_0^1 \int_0^\varepsilon \frac{1}{2} \varepsilon^2 = \frac{\varepsilon^3}{2}. \tag{26}$$

Here we used that the density of $p_t(\mathrm{d}\theta)$ is $1/2$ for $-1 \leq \theta \leq 1$. □

We see that an arbitrary small MSE-loss cannot ensure that the probability distribution is mass covering and the KL-divergence is finite. On a high level this can be explained by the fact that for vector fields v_t that are not Lipschitz continuous the flow is not necessarily continuous, and we can generate holes in the distribution. Note that we chose p_0 to be a uniform distribution for simplicity, but the result extends to any smooth distribution, in particular the result does not rely on the discontinuity of p_0.

Next, we investigate the mass covering property for Lipschitz continuous flows. When the flows u_t and v_t are Lipschitz continuous (in θ) this ensures that the flows ψ_1 and ϕ_1 are continuous in x and it is not possible to create holes in the distribution as shown above for non-continuous vector fields. We show a weaker bound in this setting.

Lemma F.2. *For every $0 \leq \delta \leq 1$ there is a base distribution $p_0 = q_0$ and the are Lipschitz-continuous vector fields u_t and v_t such that $\mathrm{MSE}_p(u, v) = \delta$ and*

$$\mathrm{D}_{\mathrm{KL}}(p_1 || q_1) \geq \frac{1}{2} \mathrm{MSE}_p(u, v)^{1/3}. \tag{27}$$

Proof. We consider p_0, q_0 and u_t as in Lemma F.1, and we define

$$v_t(\theta) = \begin{cases} 2\theta & \text{for } 0 \leq \theta < \varepsilon, \\ 2\varepsilon - \theta & \text{for } \varepsilon \leq \theta < 2\varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

Then we can calculate for $0 \leq \theta \leq e^{-2}\varepsilon$ that

$$\psi_t(\theta) = \theta e^{2t}. \tag{29}$$

Similarly we obtain for $\varepsilon \leq \theta \leq 2\varepsilon$ (solving the ODE $f' = 2f$)

$$\psi_t(\theta) = 2\varepsilon - (2\varepsilon - \theta)e^{-2t}. \tag{30}$$

We find

$$\psi_1(0) = 0,\ \psi_1(e^{-2}\varepsilon) = \varepsilon,\ \psi_1(\varepsilon) = 2 - \varepsilon e^{-2};\ \psi_2(2\varepsilon) = 2\varepsilon. \tag{31}$$

Next we find for the densities of q_1 that

$$q_1(\psi_1(\theta)) = q_0(\theta)|\psi_1'(\theta)|^{-1} = \frac{1}{2}\begin{cases} e^{-2} & \text{for } 0 \leq \theta \leq e^{-2}\varepsilon, \\ e^{2} & \text{for } \varepsilon \leq \theta \leq 2\varepsilon. \end{cases} \tag{32}$$

Together with (31) this implies that the density of q_1 is given by

$$q_1(\theta) = \frac{1}{2}\begin{cases} e^{-2} & \text{for } 0 \leq \theta \leq \varepsilon, \\ e^{2} & \text{for } 2\varepsilon - \varepsilon e^{-2} \leq \theta \leq 2\varepsilon. \end{cases} \tag{33}$$

Note that $p_1(\theta) = 1/2$ for $-1 \leq \theta \leq 1$ and therefore

$$\int_0^\varepsilon \ln \frac{p_1(\theta)}{q_1(\theta)} p_1(\mathrm{d}\theta) = \int_0^\varepsilon \ln(e^2)\frac{1}{2}\mathrm{d}\theta = \varepsilon, \tag{34}$$

and

$$\int_{2\varepsilon-\varepsilon e^{-2}}^{2\varepsilon} \ln \frac{p_1(\theta)}{q_1(\theta)} p_1(\mathrm{d}\theta) = \int_{2\varepsilon-\varepsilon e^{-2}}^{2\varepsilon} \ln(e^{-2})\frac{1}{2}\mathrm{d}\theta = -\varepsilon e^{-2}. \tag{35}$$

Moreover we note

$$\int_\varepsilon^{2\varepsilon-\varepsilon e^{-2}} q_1(\mathrm{d}\varepsilon) = \int_{e^{-2}\varepsilon}^{\varepsilon} q_0(\mathrm{d}\varepsilon) = \frac{1}{2}\varepsilon(1 - e^{-2}) = \int_\varepsilon^{2\varepsilon-\varepsilon e^{-2}} p_1(\mathrm{d}\varepsilon), \tag{36}$$

which implies (by positivity of the KL-divergence) that

$$\int_\varepsilon^{2\varepsilon-\varepsilon e^{-2}} \ln\left(\frac{p_1(\theta)}{q_1(\theta)}\right) p_1(\mathrm{d}\theta) \geq 0. \tag{37}$$

We infer using also $p_1(\theta) = q_1(\theta) = 1/2$ for $\theta \in [-1, 0] \cap [2\varepsilon, 1]$ that

$$\mathrm{D_{KL}}(p_1||q_1) = \int \ln\left(\frac{p_1(\theta)}{q_1(\theta)}\right) p_1(\mathrm{d}\theta) \geq \varepsilon(1 - e^{-2}). \tag{38}$$

On the other hand we can bound

$$\int_0^1 \mathrm{d}t \int p_t(\mathrm{d}\theta)|v_t(\theta) - u_t(\theta)|^2 = \frac{1}{2}\int_0^1 \mathrm{d}t \int_0^{2\varepsilon} |u_t(\theta)|^2 = \int_0^\varepsilon s^2\,\mathrm{d}s = \frac{\varepsilon^3}{3}. \tag{39}$$

We conclude that

$$\mathrm{D_{KL}}(p_1||q_1) \geq \frac{1}{2}\left(\mathrm{MSE}_p(u, v)\right)^{1/3}. \tag{40}$$

In particular, it is not possible to bound the KL-divergence by the MSE even when the vector fields are Lipschitz continuous. □

Let us put this into context. It was already shown in (Albergo et al., 2023) that we can, in general, not bound the forward KL-divergence by the mean squared error and our Lemmas F.1 and F.2 are concrete examples. On the other hand, when considering SDEs the KL-divergence can be bounded by the mean squared error of the drift terms as shown in (Song et al., 2021). Indeed, in (Albergo et al., 2023) the favorable smoothing effect was carefully investigated.

Here we show that we can alternatively obtain an upper bound on the KL-divergence when assuming that u_t, v_t, and p_0 satisfy additional regularity assumptions. This allows us to recover the mass covering property from bounds on the means squared error for sufficiently smooth vector fields. The scaling is nevertheless still weaker than for SDEs.

We now state our assumptions. We denote the gradient with respect to θ by $\nabla = \nabla_\mu$ and second derivatives by $\nabla^2 = \nabla^2_{\mu\nu}$. When applying the chain rule, we leave the indices implicit. We denote by $|\cdot|$ the Frobenius norm $|A| = \left(\sum_{ij} A_{ij}^2\right)^{1/2}$ of a matrix. The Frobenius norm is submultiplicative, i.e., $|AB| \leq |A| \cdot |B|$ and directly generalizes to higher order tensors.

Assumption F.3. We assume that

$$|\nabla u_t| \leq L,\ |\nabla v_t| \leq L,\ |\nabla^2 u_t| \leq L',\ |\nabla^2 v_t| \leq L'. \tag{41}$$

We require one further assumption on p_0.

Assumption F.4. There is a constant C_1 such that

$$|\nabla \ln p_0(\theta)| \leq C_1(1 + |\theta|). \tag{42}$$

We also assume that

$$\mathbb{E}_{p_0} |\theta|^2 < C_2 < \infty. \tag{43}$$

Note that (42) holds, e.g., if p_0 follows a Gaussian distribution but also for smooth distribution with slower decay at ∞. If we assume that $|\nabla \ln p_0(\theta)|$ is bounded the proof below simplifies slightly. This is, e.g., the case if $p_0(\theta) \sim e^{-|\theta|}$ as $|\theta| \to \infty$.

We need some additional notation. It is convenient to introduce $\phi_t^s = \phi_t \circ (\phi_s)^{-1}$, i.e., the flow from time s to t (in particular $\phi_t^0 = \phi_t$) and similarly for ψ. We can now restate and prove Theorem D.1.

Theorem F.5. *Let $p_0 = q_0$ and assume u_t and v_t are two vector fields whose flows satisfy $p_1 = (\phi_1)_* p_0$ and $q_1 = (\psi_1)_* q_0$. Assume that p_0 satisfies Assumption F.4 and u_t and v_t satisfy Assumption F.3. Then there is a constant $C > 0$ depending on L, L', C_1, C_2, and d such that (for $\mathrm{MSE}_p(u, v) < 1$)*

$$\mathrm{D}_{\mathrm{KL}}(p_1||q_1) \leq C\, \mathrm{MSE}_p(u, v)^{\frac{1}{2}}. \tag{44}$$

Remark F.6. We do not claim that our results are optimal, it might be possible to find similar bounds for the forward KL-divergence with weaker assumptions. However, we emphasize that Lemma F.2 shows that the result of the theorem is not true without the assumption on the second derivative of v_t and u_t.

Proof. We want to control $\mathrm{D}_{\mathrm{KL}}(p_1||q_1)$. It can be shown that (see equation above (25) in (Song et al., 2021) or Lemma 2.19 in (Albergo et al., 2023))

$$\partial_t \mathrm{D}_{\mathrm{KL}}(p_t||q_t) = -\int p_t(\mathrm{d}\theta)(u_t(\theta) - v_t(\theta)) \cdot (\nabla \ln p_t(\theta) - \nabla \ln q_t(\theta)). \tag{45}$$

Using Cauchy-Schwarz we can bound this by

$$\partial_t \mathrm{D}_{\mathrm{KL}}(p_t||q_t) \leq \left(\int p_t(\mathrm{d}\theta)|u_t(\theta) - v_t(\theta)|^2\right)^{\frac{1}{2}} \left(\int p_t(\mathrm{d}\theta)|\nabla \ln p_t(\theta) - \nabla \ln q_t(\theta)|^2\right)^{\frac{1}{2}}. \tag{46}$$

We use the relation (see (5))

$$\ln(p_t(\phi_t(\theta_0)) = \ln(p_0(\theta_0)) - \int_0^t (\mathrm{div}\, u_s)(\phi_s(\theta_0))\mathrm{d}s, \tag{47}$$

which can be equivalently rewritten (setting $\theta = \phi_t\theta_0$) as

$$\ln(p_t(\theta)) = \ln(p_0(\phi_0^t\theta)) - \int_0^t (\operatorname{div} u_s)(\phi_s^t\theta)\mathrm{d}s. \tag{48}$$

We use the following relation for $\nabla\phi_s^t$

$$\nabla\phi_s^t(\theta) = \exp\left(\int_t^s \mathrm{d}\tau\, (\nabla u_\tau)(\phi_\tau^t(\theta))\right). \tag{49}$$

This relation is standard and can be directly deduced from the following ODE for $\nabla\phi_s^t$

$$\partial_s\nabla\phi_s^t(\theta) = \nabla\partial_s\phi_s^t(\theta) = \nabla(u_s(\phi_s^t(\theta))) = \big((\nabla u_s)(\phi_s^t(\theta))\big)\cdot\nabla\phi_s^t(\theta). \tag{50}$$

We can conclude that for $0 \le s, t \le 1$ the bound

$$|\nabla\phi_s^t(\theta)| \le e^L \tag{51}$$

holds. We find

$$\begin{aligned} |\nabla\ln(p_t(\theta))| &= \left|\nabla\ln(p_0)(\phi_0^t\theta)\cdot\nabla\phi_0^t(\theta) - \int_0^t (\nabla\operatorname{div} u_s)(\phi_s^t\theta)\cdot\nabla\phi_s^t(\theta)\mathrm{d}s\right| \\ &\le |\nabla\ln(p_0)(\phi_0^t\theta)|e^L + L'e^L, \end{aligned} \tag{52}$$

and a similar bound holds for q_t. In words, we have shown that the score of p_t at θ can be bounded by the score of p_0 of theta transported along the vector field u_t minus a correction which quantifies the change of score along the path. We now bound using the definition $p_t = (\phi_t)_*p_0$ and the assumption (42)

$$\begin{aligned} \int p_t(\mathrm{d}\theta)|\nabla\ln p_0(\phi_0^t(\theta))|^2 &= \int p_0(\mathrm{d}\theta_0)|\nabla\ln p_0(\phi_0^t\phi_t(\theta_0))|^2 = \mathbb{E}_{p_0}\,|\nabla\ln p_0(\theta_0)|^2 \\ &\le \mathbb{E}_{p_0}(C_1(1+|\theta_0|)^2) \le 2C_1^2(1+\mathbb{E}_{p_0}\,|\theta_0|^2) \le 2C_1^2(1+C_2^2). \end{aligned} \tag{53}$$

Similarly we obtain using $q_0 = p_0$

$$\int p_t(\mathrm{d}\theta)|\nabla\ln q_0(\psi_0^t\theta)|^2 = \int p_0(\mathrm{d}\theta_0)|\nabla\ln q_0(\psi_0^t\phi_t\theta_0)|^2. \tag{54}$$

In words, to control the score of q integrated with respect to p_t we need to control the distortion we obtain when moving forward with u and backwards with v. We investigate $\psi_0^t\phi_t(\theta_0)$. We find

$$\partial_h\psi_t^{t+h}\phi_{t+h}^t(\theta)|_{h=0} = u_t(\theta) - v_t(\theta). \tag{55}$$

This implies

$$\partial_t(\psi_0^t\phi_t)(\theta_0) = \partial_h(\psi_0^t\psi_t^{t+h}\phi_{t+h}^t\phi_t)(\theta_0)|_{h=0} = (\nabla\psi_0^t)(\phi_t(\theta_0))\cdot((u_t-v_t)(\phi_t(\theta_0)))\,. \tag{56}$$

Using (51) we conclude that

$$\begin{aligned} |\psi_0^t\phi_t(\theta_0) - \theta_0| &\le \left|\int_0^t \partial_s\psi_0^s\phi_s(\theta_0)\,\mathrm{d}s\right| \le \int_0^t |(\nabla\psi_0^s)(\phi_s(\theta_0))|\cdot|u_s - v_s|(\phi_s(\theta_0))\,\mathrm{d}s \\ &\le e^L\int_0^t |u_s - v_s|(\phi_s(\theta_0))\,\mathrm{d}s. \end{aligned} \tag{57}$$

We use this and the assumption (42) to continue to estimate (54) as follows

$$
\begin{aligned}
\int p_t(\mathrm{d}\theta)|\nabla \ln q_0(\psi_0^t\theta)|^2 &= \int p_0(\mathrm{d}\theta_0)|\nabla \ln q_0(\psi_0^t\phi_t(\theta_0))|^2 \\
&\leq C_1^2 \int p_0(\mathrm{d}\theta_0)(1+|\psi_0^t\phi_t(\theta_0)|)^2 \\
&\leq C_1^2 \int p_0(\mathrm{d}\theta_0)(1+|\psi_0^t\phi_t(\theta_0)-\theta_0|+|\theta_0|)^2 \\
&\leq 3C_1^2 + 3C_1^2 \int p_0(\mathrm{d}\theta_0)\left(|\psi_0^t\phi_t(\theta_0)-\theta_0|^2+|\theta_0|^2\right) \\
&\leq 3C_1^2(1+\mathbb{E}_{p_0}|\theta_0|^2) + 3C_1^2 e^{2L}\int p_0(\mathrm{d}\theta_0)\left(\int_0^t \mathrm{d}s\,|u_s-v_s|(\phi_s(\theta_0))\right)^2 .
\end{aligned} \tag{58}
$$

Here we used $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$ in the second to last step. We bound the remaining integral using Cauchy-Schwarz as follows

$$
\begin{aligned}
\int p_0(\mathrm{d}\theta_0)\left(\int_0^t |u_s-v_s|(\phi_s(\theta_0))\right)^2 &\leq \int p_0(\mathrm{d}\theta_0)\left(\int_0^t \mathrm{d}s\,|u_s-v_s|^2(\phi_s(\theta_0))\right)\left(\int_0^t \mathrm{d}s\,1^2\right) \\
&\leq t\int_0^t \mathrm{d}s \int p_0(\mathrm{d}\theta_0)|u_s-v_s|^2(\phi_s(\theta_0)) \\
&= t\int_0^t \mathrm{d}s \int p_s(\mathrm{d}\theta_s)|u_s-v_s|^2(\theta_s) \\
&\leq \int_0^1 \mathrm{d}s \int p_s(\mathrm{d}\theta_s)|u_s-v_s|^2(\theta_s) = \mathrm{MSE}_p(u,v).
\end{aligned} \tag{59}
$$

The last displays together imply

$$
\int p_t(\mathrm{d}\theta)|\nabla \ln q_0(\psi_0^t\theta)|^2 \leq 3C_1^2\left(1+\mathbb{E}_{p_0}|\theta_0|^2 + e^{2L}\,\mathrm{MSE}_p(u,v)\right). \tag{60}
$$

Now we have all the necessary ingredients to bound the derivative of the KL-divergence. We control the second integral in (46) using (52) (and again $(\sum_{i=1}^4 a_i)^2 \leq 4\sum a_i^2$) as follows,

$$
\begin{aligned}
&\int p_t(\mathrm{d}\theta)|\nabla \ln p_t(\theta) - \nabla \ln q_t(\theta)|^2 \\
&\qquad \leq 2\cdot 2^2 \cdot L'^2 e^{2L} + 4e^{2L}\int p_t(\mathrm{d}\theta)\left(|\nabla \ln q_0(\psi_0^t)\theta)|^2 + |\nabla \ln p_0(\phi_0^t)\theta)|^2\right).
\end{aligned} \tag{61}
$$

Using (53) and (60) we finally obtain

$$
\begin{aligned}
\int p_t(\mathrm{d}\theta)|\nabla \ln p_t(\theta) - \nabla \ln q_t(\theta)|^2 &\leq 8\cdot L'^2 e^{2L} + C_1^2 e^{2L}\left(20(1+C_2^2) + 12\,\mathrm{MSE}_p(u,v)\right) \\
&\leq C(1+\mathrm{MSE}_p(u,v))
\end{aligned} \tag{62}
$$

for some constant $C > 0$. Finally, we obtain

$$
\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(p_1||q_1) &= \int_0^1 \mathrm{d}t\,\partial_t \mathrm{D}_{\mathrm{KL}}(p_t||q_t) \\
&\leq (C(1+\mathrm{MSE}_p(u,v)))^{\frac{1}{2}}\int_0^1 \mathrm{d}t\left(\int p_t(\mathrm{d}\theta)|u_t(\theta)-v_t(\theta)|^2\right)^{\frac{1}{2}} \\
&\leq (C(1+\mathrm{MSE}_p(u,v)))^{\frac{1}{2}}\left(\int_0^1 \mathrm{d}t\int p_t(\mathrm{d}\theta)|u_t(\theta)-v_t(\theta)|^2\right)^{\frac{1}{2}} \\
&\leq (C(1+\mathrm{MSE}_p(u,v)))^{\frac{1}{2}}\,\mathrm{MSE}_p(u,v)^{\frac{1}{2}}.
\end{aligned} \tag{63}
$$

□