# Elaborating on the Value of Flow Matching for Density Estimation


Maternus Herold [1,2,3] Faried Abu Zaid [1]

[1]appliedAI Institute for Europe [2]BMW Group [3]University of the Bundeswehr Munich


## Continuous Normalizing Flows

*Continuous-time Flows* (CNFs) transform a simple base distribution $p_0 = p_B$, usually a standard normal distribution, into a complex data distribution $p_D$. For each point in time $t \in [0, 1]$ the time-dependent vector field defines a distribution (probability path) and the goal is to find a vector field $f_\theta$ such that $p_1 = p_D$.

A CNF is constructed by parameterizing the time derivative of $x_t$ by a parametric function $f_\theta$, s.t.:

$$\frac{dx_t}{dt} = f_\theta(t, x_t). \tag{1}$$

Unlike "discrete" flows, continuous flows do not require limiting structural assumptions to yield a diffeomorphism [6]. To compute the forward and backward transformations, the system is integrated in time:

$$\phi_{t_1}(x_0) = x_{t_1} = x_0 + \int_{t=t_0}^{t_1} f_\theta(t, x_t)dt \tag{2}$$

$$\phi_{t_0}(x_1) = x_{t_0} = x_1 - \int_{t=t_0}^{t_1} f_\theta(t, x_t)dt \tag{3}$$

The above shows that CNFs have the same computational complexity in each direction, compared to their counterpart constructed from discrete transformations.

While CNFs are very flexible, they are also computationally expensive to train naively with maximum likelihood since the flow has to be integrated over time for each sample. This is especially problematic for large datasets which are needed for the precise estimation of complex high-dimensional distributions.

## CNFs via Flow Matching

Lipman et al. [4] train a CNF by regressing $f_\theta$ directly from an implicit definition of the target vector field that defines $p_t(x)$ where $p_0 = p_B$ and $p_1 = p_D$. They do so by defining a conditional vector field w.r.t. single samples from the data set.

If the target vector field $u_t$ would be known, it can be regressed directly via

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, p_t(x)} \| f_\theta(t, x) - u_t(x) \|^2 \tag{4}$$

Lipman et al. [4] show that one can define appropriate conditional target vector fields when conditioning on the outcome $x_1$:

$$p_t(x) = \int p_t(x \mid x_1) p_D(x_1) dx_1. \tag{5}$$

One possibility for $p_t(x \mid x_1)$ is a Gaussian probability path, leading to a specific form of $\phi_t(x \mid x_1)$:

$$p_t(x \mid x_1) = \mathcal{N}(x; \mu_t(x_1), \sigma_t(x_1)^2 I) \tag{6}$$

$$\phi_t(x \mid x_1) = \sigma_t(x_1)x + \mu_t(x_1) \tag{7}$$

with $\mu_0(x_1) = 0,\ \sigma_0(x_1) = 1$ and $\mu_1(x_1) = x_1,\ \sigma_1(x_1) = \sigma_{\min}$.

The authors show that the conditional flow matching loss obtains equivalent gradients as the flow matching loss eq. (4) w.r.t. the vector field $u_t(x)$.

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, p_t(x \mid x_1), p_D(x_1)} \| f_\theta(t, x) - u_t(x \mid x_1) \|^2 \tag{8}$$

## Neural Posterior Estimation

Bayesian inference is often not applicable to simulation-based models due to the intractable likelihood of the simulator. Simulation-Based Inference [2] (SBI) and especially Neural Posterior Inference [7] (NPE) mitigate this issue by learning the posterior directly using conditional Normalizing Flows.

Given a simulator $\mathcal{M}(\theta) = x$, where $\theta \sim \pi(\theta)$.

1. Generate $\{(\theta, x)_i\}_{i=1}^N$, where $(\theta, x)_i \sim p(\theta, x) = p(x \mid \theta)\pi(\theta)$ by
   1. $\theta \sim \pi(\theta)$
   2. $x \sim p(x \mid \theta)$ by evaluating $x = \mathcal{M}(\theta)$
2. Train a cond. NF $q_\omega(\theta \mid x) \approx p(\theta \mid x)$ minimizing the negative log-likelihood

$$\arg\min_{\omega \in \mathbb{R}^M} \frac{1}{N} \sum_{i=1}^{N} -\log q_\omega(\theta_i \mid x_i) \tag{9}$$

3. Obtain an amortized estimator by conditioning on an observation $x_o$: $q_\omega(\theta \mid x = x_o)$.

Running $\mathcal{M}$ is usually expensive and mapping $\mathcal{M} : \theta \mapsto x$ is highly non-linear. Therefore, $N$ tends to be small.

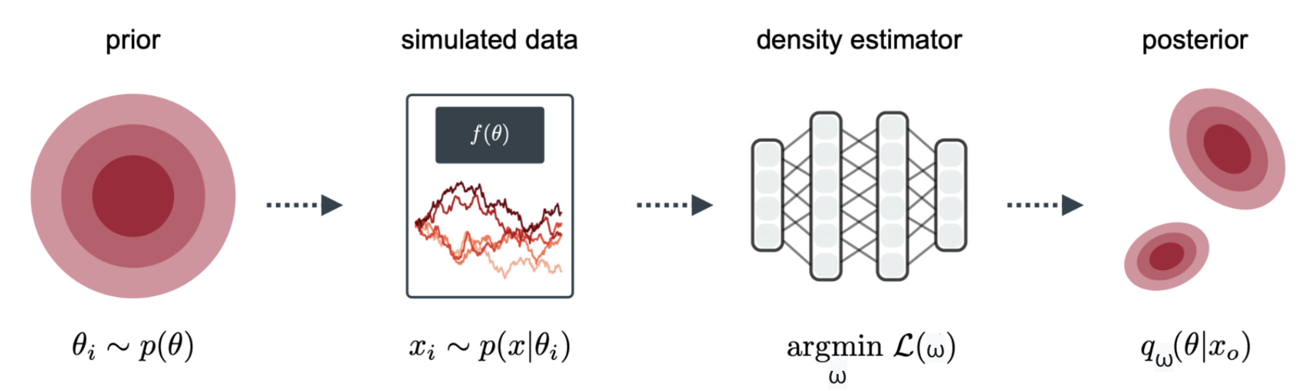


Figure 1. Schematic illustration of fitting a neural posterior estimator using samples from the joint distribution. Figure taken from [1].

## Flow Matching Posterior Esitmation

In SBI, sample efficiency, scalability, and expressivity of the density model are important. Flow matching improves on such aspects due to the efficient transport between source and target density and the higher flexibility of applicable transformations allowed by continuous normalizing flows.

| | NPE | NPSE | FMPE |
|---|---|---|---|
| Tractable posterior density | ✔ | | ✔ |
| Flexible network architecture | | ✔ | ✔ |
| Network passes for sampling | single | many | many |

Table 1. Information taken from [3].

Dax et al. [3] propose flow matching posterior estimation (FMPE) by adapting the flow matching loss, utilizing Bayes' rule change $\mathbb{E}_{p(x)p(\theta|x)}$ into $\mathbb{E}_{\pi(\theta)p(x|\theta)}$.

$$\mathcal{L}(\omega) = \mathbb{E}_{t, \theta_1, x, \theta_t} \| f_{\omega, x}(\theta_t, t) - u_t(\theta_t \mid \theta_1) \|^2 \tag{10}$$

where $t \sim p(t), \theta_1 \sim \pi(\theta), x \sim p(x \mid \theta_1), \theta_t \sim p_t(\theta \mid \theta_1)$.

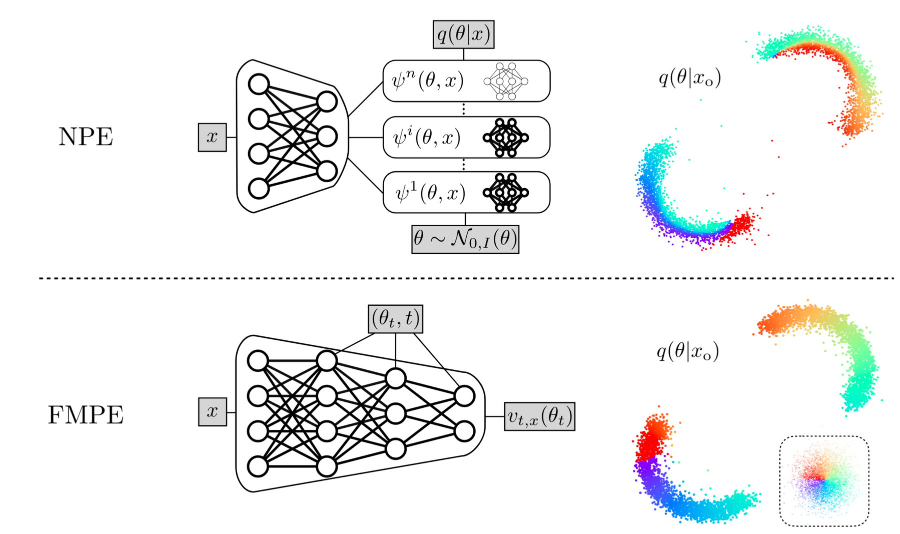


Figure 2. Highlighting the different structure of previous NPE approaches vs. FMPE. The color coding indicates the efficient transportaion of probability mass when using optimal-transport flow matching. Figure taken from [3].

## Performance of Flow Matching on Benchmarking Tasks

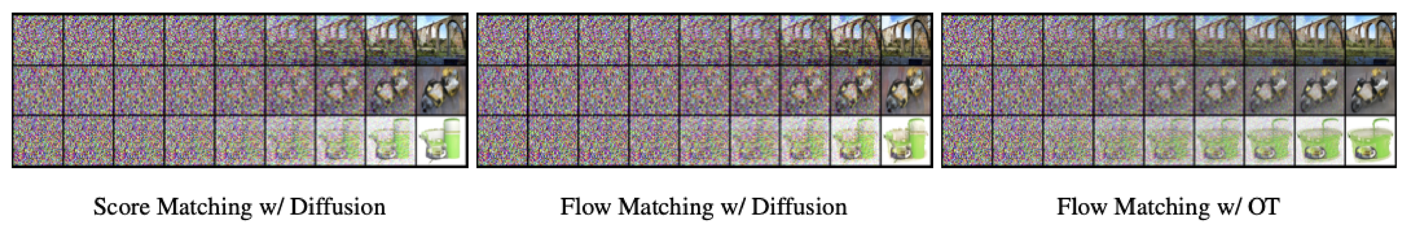


Figure 3. Sample paths from the same initial noise with models trained on ImageNet 64×64. The OT path reduces noise roughly linearly, while diffusion paths visibly remove noise only towards the end of the path. Note also the differences between the generated images. Figure from [4].

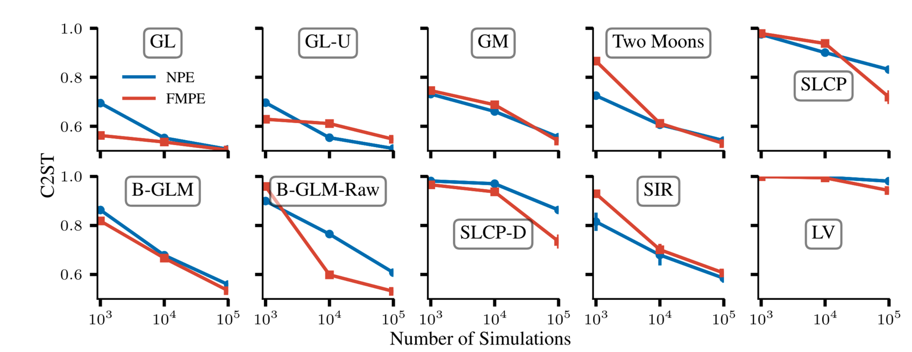


Figure 4. Comparissons of FMPE to popular NPE methods on SBI benchmarking tasks [5]. Figure taken from [3].

## References


[1] Jan Bölts.
*Advancing Methods and Applicability of Simulation-Based Inference in Neuroscience.*
PhD thesis, Universität Tübingen, July 2023.

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

[3] Maximilian Dax, Jonas Wildberger, Simon Buchholz, Stephen R. Green, Jakob H. Macke, and Bernhard Schölkopf.
Flow Matching for Scalable Simulation-Based Inference, May 2023.

[4] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le.
Flow Matching for Generative Modeling, February 2023.

[5] Jan-Matthis Lueckmann, Jan Boelts, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke.
Benchmarking Simulation-Based Inference, April 2021.

[6] George Papamakarios.
Neural Density Estimation and Likelihood-free Inference, October 2019.

[7] George Papamakarios and Iain Murray.
Fast $\epsilon$-free Inference of Simulation Models with Bayesian Conditional Density Estimation.
2016.





m.herold@appliedai-institute.de