Conditional Flow Matching for Bayesian Posterior Inference

Percy S. Zhai* **So Won Jeong*** **Veronika Ročková**
University of Chicago, Booth School of Business

Abstract

We propose a generative multivariate posterior sampler via flow matching. It offers a simple training objective, and does not require access to likelihood evaluation. The method learns a dynamic, block-triangular velocity field in the joint space of data and parameters, which results in a deterministic transport map from a source distribution to the desired posterior. The inverse map, named vector rank, is accessible by reversibly integrating the velocity over time. It is advantageous to leverage the dynamic design: proper constraints on the velocity yield a monotone map, which leads to a conditional Brenier map, enabling a fast and simultaneous generation of Bayesian credible sets whose contours correspond to level sets of Monge-Kantorovich data depth. Our approach is computationally lighter compared to GAN-based and diffusion-based counterparts, and is capable of capturing complex posterior structures. Finally, frequentist theoretical guarantee on the consistency of the recovered posterior distribution, and of the corresponding Bayesian credible sets, is provided.

1 INTRODUCTION

Bayesian inference provides a principled framework of uncertainty quantification in statistical models, enabling estimation of the full posterior distribution. The primary bottleneck lies in computation, since exact posterior evaluation is often intractable. Traditional methods rely on approximate inference techniques such as MCMC or variational inference, which may suffer from slow convergence or restrictive assumptions. For instance, MCMC requires rerunning the entire chain for each new observation. Recent advances in generative modeling, particularly score-based diffusion models (Song et al., 2020; Ho and Salimans, 2022) and normalizing flows (Rezende and Mohamed, 2016), offer promising alternatives. However, notable limitations persist: diffusion models require iterative denoising steps and rely on approximating the score function, while normalizing flows impose invertibility constraints and exact likelihood evaluations.

This work develops a novel Bayesian posterior sampler based on *flow matching* (Lipman et al., 2023; Liu et al., 2023), a generative modeling technique that learns a deterministic velocity field transporting a simple source distribution to a complex target. Consider parameter space $\Theta \subset \mathbb{R}^d$ and data space $\mathcal{Y} \subset \mathbb{R}^n$. Observations $y \in \mathcal{Y}$ arise from parameters $\theta \in \Theta$, either via an explicit likelihood $L(y \mid \theta)$ or through an implicit simulator. We allow entries of y to be potentially dependent. Combining the data-generating process with a prior simulator $\pi(\theta)$, one can generate joint samples $\{y_i, \theta_i\}_{i=1}^N$, where the size of the synthetic joint pairs N can be large at a low computational cost. This setup enables conditional inference in a likelihood-free setting, also known as simulation-based inference (SBI).

For a fixed observation $y^* \in \mathcal{Y}$, the primary objective of Bayesian inference is to sample from the posterior $\pi(\theta \mid y^*)$. From the optimal transport point of view, this is equivalent to learning a transport map from a source distribution (e.g. uniform) to the target posterior. This perspective has been considered by multiple works in the literature. Wang and Ročková (2022) learns the posterior sampler by generative adversarial networks (GANs). A similar route is taken by Baptista et al. (2024) using the Monotone GAN (M-GAN) approach. More recently, Kim et al. (2025b) develops a conditional quantile learning method that recovers an optimal transport map on the parameter space Θ. Despite the considerably different approaches, the latter two both yield the conditional Brenier map (Carlier et al., 2017) under monotonicity assumptions. This paper marks an addition to the toolbox for poste-

*Equal contribution.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

rior sampling from another perspective, using the flow matching technique to achieve a dynamic extension. See Table 1 for a comparison of selected approaches, and Appendix B for a comprehensive literature review.

Our approach learns the posterior by matching the *joint distribution* of parameters and observations, rather than the marginal or conditional alone (Mirza and Osindero, 2014; Zhou et al., 2023). The theoretical cornerstone is the *block-triangular map*. We consider maps $T : \mathcal{Y} \times \Theta \to \mathcal{Y} \times \Theta$ that jointly transports random noise $y_0 \in \mathcal{Y}$ and $\theta_0 \in \Theta$ to the data denoted by $y_1 \in \mathcal{Y}$ and the parameter $\theta_1 \in \Theta$ at the following form,

$$T(y_0, \theta_0) = (F(y_0), G(F(y_0), \theta_0)), \tag{1}$$

where $F : \mathcal{Y} \to \mathcal{Y}$, and $G : \mathcal{Y} \times \Theta \to \Theta$. Such maps are called *block-triangular*, in a sense that the Jacobian matrix of the joint transport map has a lower block-triangular shape. A primary theoretical result established in Baptista et al. (2024) states that if a joint source distribution η is transported by such a map T to the joint distribution (i.e. the push-forward measure $T_\sharp \eta = \pi(y, \theta)$), then $G(y^*, \cdot)$ maps the Θ-marginal of the source distribution to the posterior distribution of θ. Due to this convenient result, the task of posterior sampling boils down to learning a joint map T that has a block-triangular structure. The M-GAN developed by Baptista et al. (2024) learns the parameter map $G(y^*, \cdot)$ through an adversarial scheme.

In this paper, we instead learn the block-triangular map T using the flow matching method. That is, for any $t \in [0, 1]$, we learn a joint velocity field in $\mathcal{Y} \times \Theta$ that results in a block-triangular transport map from $t = 0$ to $t = 1$. The dynamic nature of this method leads to a convenient training process. For example, we may choose any interpolation path to reduce computational complexity while obtaining a theoretical guarantee. When the map is monotone, both Baptista et al. (2024) and Kim et al. (2025b), as well as our method, eventually learn the conditional Brenier map, an optimal transport map from a source distribution to the posterior distribution. With this same goal in mind, Baptista et al. (2024) leverages adversarial learning scheme through GAN, and Kim et al. (2025b) relies on a variation of Quantile Neural Network (QNN), while our method leverages flow matching architecture to obtain the conditional map. See Table 1 for a detailed comparison. Recognizing the importance of the block-triangular maps, a followup work of the M-GAN method (Alfonso et al., 2023) formulated the problem of learning such a map by a discretized temporal mapping method. We highlight that our route reaching such maps is distinctive of previous works in that our methods rely on flow matching unlike GANs, and operate on continuous time space unlike Alfonso et al. (2023).

After finalizing our manuscript, we became aware of concurrent works by Kerrigan et al. (2024) and Chemseddine et al. (2024), which learn a conditional velocity field with the observation treated as a fixed input, aiming for optimal transport optimality. Our work, by contrast, learns a block-triangular velocity field on the joint space. Moreover, our proposed method aims for consistency of posterior distributional estimation, as opposed to optimal transport from source to target. A detailed comparison is provided in Appendix C.1.

Main Contributions

1. **Likelihood-free Posterior Sampler.** In Section 2, we adapt flow matching technique to the posterior sampling problem, with or without access to the explicit likelihood. With a similar accuracy in posterior sampling, our proposed method requires less computation time than MCMC and M-GAN. Our proposed method is a dynamic extension of block-triangular maps, with a straightforward access to the inverse maps.

2. **Inference through Credible Sets.** Our posterior sampling method enables us to implement posterior uncertainty quantification (Section 2.3). Under mild assumptions, we can draw Bayesian credible sets at multiple levels simultaneously without having to learn the maps repeatedly. The contour of these credible sets agree with the levels of Monge-Kantorovich data depth, providing a more sensible shape. Our method also allows an easy access to the inverse map, enabling a conditional rank function that plays a role similar to the p-value in comparing parameters to the posterior distribution.

3. **Consistency of Learned Posterior.** In Section 3, we provide theoretical guarantee on the asymptotic consistency of distributions recovered by flow matching if the velocity is learned by multilayer perceptron with ReLU activation functions, the first of its kind to the best of our knowledge. We establish that the recovered posterior distribution converges to the true posterior in 2-Wasserstein distance. As a corollary, the corresponding Bayesian credible sets are also consistent in Hausdorff distance.

2 METHODOLOGY

We develop a method based on the theory on block-triangular maps (1), but substantially different from M-GAN (Baptista et al., 2024). First, we extend the

Feature	Flow Matching (Ours)	GANs (e.g., Baptista et al. (2024))	Diffusion Models (e.g., Chung et al. (2023))	Quantile NN (e.g., Kim et al. (2025b))
Core Principle	Learns a velocity field (ODE) to transport noise to the joint distribution.	Learns a generator to fool a discriminator on joint samples.	Learns to reverse a forward noising process via score estimation.	Directly learns the conditional quantile function (Brenier map).
Training Objective	Regression Loss.	Adversarial Loss.	Score-Matching Loss.	Pinball Loss or MK objective
Likelihood-Free	✓	✓	∼	✓
Access to the Inverse Map	✓	✗	∼	✗
Uncertainty Quantification	✓	∼	✗	✓

✓: Supported Directly ∼: Possible with Constraints/Indirectly ✗: Not Directly Supported

Table 1: Comparison of selected generative approaches for posterior sampling.

block-triangular map to a dynamic domain. Instead of learning a one-off transport map, our proposed method learns a dynamic map over time $t \in [0,1]$. We then have a freedom on choosing the intermediate states (i.e., interpolation path). Second, unlike several existing posterior sampling methods that learns the posterior transport map directly, our proposed method is based on learning a transport map on the *joint space* $\mathcal{Y} \times \Theta$. This decoupling of marginal transport from conditional transport brings extra flexibility by alleviating representational burden on the neural network for learning the velocity field. A more detailed discussion in this regard is deferred to Appendix C.1. Our proposed method also comes with several byproducts, including the access to the data map F and the inverse maps F^{-1} and $[G(y^*, \cdot)]^{-1}$.

To recover the posterior distribution, it is only required that the final trained map T is block-triangular as in (1). By Theorem 3.4 in Baptista et al. (2024), the map $G(y^*, \cdot)$ transports the noise θ_0 to the posterior distribution $\theta_1 \sim \pi(\cdot \mid y^*)$. We aim to learn the joint map T via flow matching, which is characterized by a dynamic velocity field for $t \in [0,1]$ (the ending time is set to 1 hereafter, without loss of generality). The joint transport map from the source (y_0, θ_0) to the target (y_1, θ_1) is recovered by accumulating the velocity over time.

2.1 Learning the Joint via Flow Matching

For brevity, denote $x_t = (y_t, \theta_t)$ for any time point $t \in [0,1]$. Our goal is to learn a transport map from the source distribution $x_0 \sim p_0$ to the target joint distribution $x_1 \sim p_1 = \pi(y_1, \theta_1)$. This can be done by Flow Matching (FM, Lipman et al. (2023); Liu et al. (2023); Tong et al. (2023)), which learns a deterministic velocity field that governs the time-dependent flow of x_t. In general, we denote a *flow* starting in time t_0 by $\{x_{t_0,t}\}_{t\in[t_0,1]}$. In our problem, $t_0 = 0$, with the initial point $x_{0,0} = x_0$, and $x_{0,t} = x_t$. For simplicity, we shall denote the flow by $\{x_t\}_{t\in[0,1]}$ hereafter, unless otherwise specified. Each flow can be identified with a *velocity field*, which is the derivative of x_t with respect to time, $v_t(x_t) = \mathrm{d}x_t/\,\mathrm{d}t$. Each intermediate state, x_t, can thus be regarded as a collective result of the velocity $\{v_s\}_{s\in[0,t]}$, which also defines a dynamic push-forward map T_t applied from the origin x_0. In this case, T_1 corresponds to the final resulting map T in (1).

Denote the velocity field $\{v_t\}_{t\in[0,1]}$ as a function $v : \mathcal{X} \times [0,1] \to \mathbb{R}^{n+d}$. The FM loss function is defined as

$$\mathcal{L}_{\mathrm{FM}}(\hat{v}, v) = \int_0^1 \mathbb{E}_{x\sim p_t} \|\hat{v}_t(x) - v_t(x)\|^2 \,\mathrm{d}t. \tag{2}$$

However, this objective is generally intractable as we do not have an access to the unknown true velocity v_t. To make this objective more feasible, Lipman et al. (2023) proposes a *conditional* flow matching loss. The idea is to construct p_t implicitly as a mixture of conditionals

$$p_t(x) = \int p_t(x \mid z) q(z) \,\mathrm{d}z,$$

where z is an auxiliary conditioning variable (often taken as $z = (x_0, x_1)$) and $q(z)$ is a known joint distribution over pairs. A standard instantiation is the Gaussian interpolation path

$$x_t = \mu_t(z) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_t^2 I), \tag{3}$$

where $\mu_t(z) = (1-t)x_0 + tx_1$ defines a linear interpolation, and $\sigma_t > 0$ controls the spread of the conditional. The velocity field conditioned on the initial and ending states can be written explicitly as

$$v_t(x \mid x_0, x_1) = \frac{\sigma_t'}{\sigma_t}(x - \mu_t) + \mu_t',$$

where μ_t' and σ_t' denote the time derivative of μ_t and σ_t, respectively. This yields a tractable target for regression under the conditional model. In practice, we use the deterministic limit $\sigma_t \to 0$, yielding the straight-line path $x_t = (1-t)x_0 + tx_1$ with conditional velocity $v_t(x \mid x_0, x_1) = x_1 - x_0$.

The conditional flow matching loss is then written as

$$\mathcal{L}_{\mathrm{CFM}}(\hat{v}, v) = \int_0^1 \mathbb{E}_{q(z),p_t(x|z)} \|\hat{v}_t(x) - v_t(x|z)\|^2 \,\mathrm{d}t, \tag{4}$$

which is fully computable given samples from $q(z)$ and the interpolation path. Lipman et al. (2023) showed

that (4) is equivalent to the unconditional flow matching loss (2) up to a constant. In other words, the gradient with respect to neural network parameters φ for both loss functions (2) and (4) matches:

$$\nabla_\varphi \mathcal{L}_{\mathrm{FM}} = \nabla_\varphi \mathcal{L}_{\mathrm{CFM}}. \tag{5}$$

By learning the minimizer of $\mathcal{L}_{\mathrm{CFM}}$, we obtain the velocity field $\hat{v}_t(x)$ that also minimizes $\mathcal{L}_{\mathrm{FM}}$.

2.2 Dynamic Block-triangular Map

To apply the block-triangular structure on the dynamic joint maps learned by FM, an analytically and practically convenient setup is to constrain T_t to be block-triangular for all $t \in [0,1]$,

$$T_t(y_0, \theta_0) = (F_t(y_0), G_t(F_t(y_0), \theta_0)), \tag{6}$$

where for any $t \in [0,1]$, $F_t : \mathcal{Y} \to \mathcal{Y}$ maps y_0 to y_t, and $G_t : \mathcal{Y} \times \Theta \to \Theta$ maps the initial pair $x_0 = (y_0, \theta_0)$ to θ_t. Under these notations, the final maps F_1 and G_1 correspond to F and G respectively.

In fact, the maps T_t in (6) can be achieved by accumulating block-triangular velocity of the form

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = f_t(y_t), \quad \frac{\mathrm{d}\theta_t}{\mathrm{d}t} = g_t(y_t, \theta_t) \tag{7}$$

from time zero to time t. The correspondence between the univariate ODE $\mathrm{d}y_t/\,\mathrm{d}t = f_t(y_t)$ and its solution $y_t = F_t(y_0)$ is straightforward. It is therefore possible to write any y_t as a function of y_1: instead of pushing forward from $t = 0$ to $t = 1$, we can reversely push from $t = 1$ back to $t = 0$. Specifically, define

$$\tilde{F}_t(y_1) = y_t = F_t(y_0) = F_t(F^{-1}(y_1)). \tag{8}$$

Then the second half of (7) can be written as $\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = g_t(\tilde{F}_t(y_1), \theta_t)$, yielding a univariate ODE for θ_t. Here, θ_0 is a fixed value realization of a stochastic noise in the parameter space, and y_1 is treated as a constant. The solution of this ODE can be written as $\theta_t = G_t(y_1, \theta_0)$, which is equivalent to the form in (6). This essentially proves the following important lemma.

Lemma 1 *The dynamic map T_t is block-triangular in the form of* (6) *for all $t \in [0,1]$ if the velocity field can be expressed in the block-triangular form of* (7)*.*

This means that a block-triangular velocity design will always lead to block-triangular dynamic maps throughout the process, including the resulting map at $t = 1$. Based on (7) and (8), it is therefore obvious that given a fixed $y^* \in \mathcal{Y}$, the posterior can be simulated by pushing a random noise θ_0 through the learned velocity field,

$$G(y^*, \theta_0) = \theta_0 + \int_0^1 g_t(\tilde{F}_t(y^*), \theta_t)\,\mathrm{d}t. \tag{9}$$

If we obtain a velocity field accurately enough, by sampling several realizations of θ_0 from the source distribution and passing them through the map $G(y^*, \cdot)$, the resulting data points θ_1 should follow the posterior distribution $\pi(\cdot \mid y^*)$. (9) plays a central role in our proposed algorithm for posterior simulation.

A major advantage of this dynamic configuration is the automatic invertibility of the learned map. With a fixed observation y^*, the inverse of the conditional map $G(y^*, \cdot)$ can also be obtained by integrating (9) in reverse. Without additional monotonicity assumptions, these inverse maps are more intuitive than useful. We defer further discussions on inverse maps to Section 2.3. Our entire posterior learning framework is summarized in Algorithm 1.

Remark 1 *From the first sight, it may seem that the class of admissible joint flows is reduced. Indeed, the block-triangular map* (6) *is a smaller class compared to a fully general joint map. This restriction, however, does not affect the representation capability for the final joint distribution p_1. Any target joint law of (θ, y) can be represented by transporting through the marginal of y and then through the conditional law $\theta \mid y$. Rather than on the final joint distribution, the major restriction is essentially on the choice of intermediate flow path for $t \in (0,1)$, rather than the ability to represent the target distribution itself.*

Remark 2 *In Algorithm 1, line 6 employs linear interpolation for simplicity, but our framework is agnostic to the choice of conditional path. Any valid conditional path can be used, including the optimal transport path (Tong et al., 2023) or the widely used Gaussian path* (3)*.*

2.3 Monotonicity, Monge-Kantorovich Depth, and Inverse Maps

We can already construct a posterior sampler via the block-triangular map that recovers the true underlying posterior, $\pi(\cdot \mid y^*)$. From an optimal transport perspective, there exists multiple maps that push a source distribution on Θ to the true posterior. This lack of uniqueness does not affect the accuracy of posterior sampling. However, if we intend to make use of the recovered posterior distribution for inference tasks (for example, to create Bayesian credible sets with meaningful interpretation), then it is convenient to assume that the transport map $G(y^*, \cdot)$ is monotone, i.e. for any $\theta_0, \theta_0' \in \Theta$,

$$(G(y^*, \theta_0) - G(y^*, \theta_0'))^\top (\theta_0 - \theta_0') \geq 0.$$

See, e.g. Chernozhukov et al. (2017). We shall see that such monotonicity enables nested credible sets at a low

Algorithm 1 Posterior Sampling with Flow Matching

Input: Joint samples $\{x_1^{(i)}\}_{i=1}^N$ where $x_1 = (y_1, \theta_1)$, from the target distribution $p_1(x)$, a source distribution $p_0(x)$, and a neural velocity field $u_\varphi(x, t)$.
Parameters: Training steps S, learning rate η, batch size B
for $s = 1, \ldots, S$ **do**
  Sample minibatch $\{x_1^{(i)}\}_{i=1}^B \sim p_1(x)$
  Sample source points $\{x_0^{(i)}\}_{i=1}^B \sim p_0(x)$, and a random time $t^{(i)} \sim \text{Unif}[0,1]$ for each instance i.
  Form interpolation: $x_t^{(i)} = (1-t^{(i)})x_0^{(i)} + t^{(i)}x_1^{(i)}$
  Compute target conditional velocity:

$$v^{(i)} = x_1^{(i)} - x_0^{(i)}.$$

  Predict velocity: $\hat{v}^{(i)} = u_\varphi\big(x_t^{(i)}, t^{(i)}\big)$
  Compute loss: $\mathcal{L} = \frac{1}{B}\sum_{i=1}^B \|\hat{v}^{(i)} - v^{(i)}\|^2$
  Update parameters: $\varphi \leftarrow \varphi - \eta\nabla_\varphi\mathcal{L}$
end for
Sampling posterior given y^***:**
Draw $x_0 = (y_0, \theta_0) \sim p_0(x)$
Solve (9) or an equivalent ODE: $\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = u_\varphi(y^*, \theta_0, t)$ for $t \in [0,1]$ to obtain $\hat{G}_t(y^*, \cdot)$
Return θ_1 as posterior sample

computational cost. One method to enforce this in the flow matching setup is to constrain the Θ-coordinate of the velocity, $g_t(y_t, \theta_t)$, to the monotone functions in θ_t. One rigorous method to achieve this is to let

$$g_t(y, \theta) = \nabla_\theta \psi_t(y, \theta), \tag{10}$$

where $\psi_t(y, \theta)$ is convex in θ, and train the convex function ψ_t, for instance, with input convex neural network (ICNN, Amos et al. (2017)). This implementation represents a key aspect of our approach that is not explored in the concurrent work (Kerrigan et al., 2024).

Under the monotonicity assumption, let us specifically suppose that the source distribution of θ_0 is spherical uniform, i.e. $\theta_0 = r\phi$, with a unidimensional $r \sim \text{Unif}(0,1)$, and a $(d-1)$-dimensional ϕ following uniform distribution on the unit sphere $\mathcal{S}^{d-1}(1)$. Then the map $G(y^*, \cdot)$ can be regarded as a multivariate Monge-Kantorovich (MK) conditional vector quantile, also named Brenier map, denoted by $Q_{\theta|y^*} : \Theta \to \Theta$. In this case, the Bayesian τ-credible set can be established by passing a ball $S^d(\tau)$ through the learned transport map obtained from Algorithm 1,

$$\hat{C}_\tau(y^*) = \hat{G}(y^*, S^d(\tau)). \tag{11}$$

We can obtain multiple nested credible sets by passing balls of different radii through $\hat{Q}_{\theta|y^*}$, without resampling or relearning the map, significantly speeding up the computation process. We shall see that under some regularity conditions, these credible sets $\hat{C}_\tau(y^*)$ are consistent with the oracle sets,

$$C_\tau(y^*) = G(y^*, S^d(\tau)). \tag{12}$$

Interestingly, these oracle sets are meaningful in that each $C_\tau(y^*)$ is identical to the Monge-Kantorovich depth region with probability content τ (Chernozhukov et al., 2017), which is a multivariate analogue of quantiles.

Since this map is learned via flow matching, we automatically have access to the inverse map of $Q_{\theta|y^*}$, called Monge-Kantorovich conditional vector rank (Chernozhukov et al., 2017), denoted by

$$R_{\theta|y^*}(\theta_1) = [G(y^*, \cdot)]^{-1}(\theta_1). \tag{13}$$

The MK conditional vector rank can be decomposed into a MK conditional rank function $r_{\theta|y^*} : \Theta \to [0,1]$, with $r_{\theta|y^*}(\theta_1) = \|R_{\theta|y^*}(\theta_1)\|$, and a MK conditional sign function $u_{\theta|y^*}$, mapping any parameter $\theta_1 \in \Theta$ to $u_{\theta|y^*}(\theta_1) = R_{\theta|y^*}(\theta_1)/r_{\theta|y^*}(\theta_1) \in \mathcal{S}^{d-1}(1)$. Note that $r_{\theta|y^*}$ is a representation of the Monge-Kantorovich depth of the parameter in the posterior distribution $\pi(\cdot \mid y^*)$. For any $\theta_1, \theta_1' \in \Theta$, the MK depth of θ_1 is greater or equal to that of θ_1' if and only if $r_{\theta|y^*}(\theta_1) \le r_{\theta|y^*}(\theta_1')$. Moreover, $r_{\theta|y^*}$ plays a role similar to the p-value – when this quantity is close to one, it is unlikely that the corresponding θ is sampled from the posterior distribution.

Remark 3 *The representation* (10) *using ICNN comes with a caveat: the class of transport maps G representable by such a velocity g_t is in fact restricted, in that only expansive maps ($\nabla_\theta G \succeq I$) can be represented. This limitation can be practically mitigated by initializing the source distribution with a much smaller variance than the target. We provide reasoning of this claim and an alternative velocity formulation in Appendix C.2.*

3 THEORETICAL GUARANTEE

This section is aimed at establishing consistency results on the estimated posterior. By mapping the source distribution through the learned map $\hat{G}(y^*, \cdot)$, we obtain an estimated posterior $\hat{\pi}(\theta \mid y^*)$. We start by showing that it is close to the true posterior $\pi(\theta \mid y^*)$. Under regularity conditions, we establish an asymptotic consistency result on the posterior. With extra monotonicity assumptions, we establish the consistency of Bayesian credible sets based on the theoretical framework of Monge-Kantorovich quantiles. We defer the proofs of the results presented in this section to Appendix D.

3.1 Block Triangular Mapping Guarantee

We start by imposing the following mild technical assumptions on the true underlying velocity field, v_t, and its estimate by the algorithm, $\hat{v}_t$.

Assumption 1 (Smoothness of Flows) *For each $\xi \in \mathcal{Y} \times \Theta$ and $t_0 \in [0,1]$ there exist unique flows $\{\hat{x}_{t_0,t}\}_{t\in[t_0,1]}$ and $\{x_{t_0,t}\}_{t\in[t_0,1]}$ starting in $\hat{x}_{t_0,t_0} = \xi$ and $x_{t_0,t_0} = \xi$, such that their velocity fields are $\hat{v}_t(\xi)$ and $v_t(\xi)$ respectively. Moreover, both $\hat{x}_{t_0,t}$ and $x_{t_0,t}$ are continuously differentiable in ξ, t_0 and t.*

Assumption 2 (Spatial Lipschitzness) *The approximate velocity field $\hat{v}_t(x)$ is differentiable in both x and t. Also, for each $t \in (0,1)$ there exists a constant L_t such that $\hat{v}_t(x)$ is L_t-Lipschitz in x.*

Assumption 1 is required since the flow matching method relies on solving the ODE (7), while Assumption 2 imposes smoothness constraints on the estimated velocity field. From these two assumptions, it is already sufficient to provide a guarantee on the estimated joint distribution; see Theorem 1 of Benton et al. (2024).

We now extend this result from the joint sampler to the posterior sampler. The following theorem ensures proximity between the true posterior and its estimate from the sampler.

Theorem 1 (Posterior Sampler Accuracy) *Assume that the learned velocity field is constrained in the form of (7). Under Assumptions 1 and 2, when $\mathcal{L}_{FM}(\hat{v}, v) \le \varepsilon_N^2$, we have*

$$\mathbb{E}_{\pi_Y}\left[W_2^2(\hat{\pi}(\theta \mid Y), \pi(\theta \mid Y))\right] \le 2Le^{2L}\varepsilon_N^2,$$

where $L = \int_0^1 L_t \, \mathrm{d}t$, and the expectation is taken over the marginal distribution of Y, denoted by π_Y.

Intuitively, the quality of the posterior sampler relies on two aspects: the extent that the objective function is minimized, and the spatial-Lipschitz constant of the true underlying velocity field. The sampling error, measured by the 2-Wasserstein distance, grows exponentially with the Lipschitz constant L_t.

We note that L_t also captures the variability of the y-component of velocity with respect to θ, and the variability of the θ-component of velocity with respect to y. This inspires us to apply rescaling on the joint space $\mathcal{Y} \times \Theta$, such that each dimension has similar variance. In fact, our experiments also support this insight. With an extra scaling step applied on the joint space while training the neural network, the sampler yields better performance. See Appendix E.2 for further details.

3.2 Consistency of Posterior Sampler

An important factor in the upper bound in Theorem 1 is the control on the flow matching objective function, $\mathcal{L}_{\mathrm{FM}}$. We might be curious about the possibility to obtain $\mathcal{L}_{\mathrm{FM}} \to 0$ as we increase the generative sample size N. Theorem 2 provides an affirmative answer to this question, and establishes an asymptotic consistency result for the posterior sampler. In addition to the existing assumptions, it relies on the following regularity conditions.

Assumption 3 (Boundedness) *There exists an absolute constant $M > 0$ such that for all $t \in [0,1]$: (1) each entry of the true underlying velocity function v_t satisfies $\sup_{t\in[0,1]} |(v_t)_k|_\infty \le M$ in all dimensions $1 \le k \le d$, (2) the velocity learned by the feedforward neural network satisfies $\sup_{t\in[0,1]} |(\hat{v}_t)_k|_\infty \le 2M$ in all dimensions $1 \le k \le d$, and (3) the joint samples are created from compact domains, $\mathcal{Y} \subset [-M, M]^n$, $\Theta \subset [-M, M]^d$. Here, $|\cdot|_\infty$ denotes the supremum norm of a univariate function.*

Assumption 4 (Smoothness) *Assume that there exists $\beta \in \mathbb{N}_+$ such that the velocity field v lies in the Sobolev ball $\mathcal{W}^{\beta,\infty}([0,1] \times \mathcal{X})$.*

Assumptions 3 and 4 are originally imposed by Farrell et al. (2021) for the guarantee of functional estimation via the multilayer perceptron with ReLU activation. These boundedness assumptions are fairly standard in nonparametrics (see Farrell et al. (2021)), and the choice of M in Assumption 3 may be arbitrarily large; no properties of M are required apart from being finite. The Sobolev smoothness in assumption 4 is an increment from the differentiability in x and t in Assumption 1. This increment is natural, and the additional requirement is that the velocity field does not vary too drastically with x and t. Detailed discussion of Sobolev spaces can be found in Giné and Nickl (2021).

Theorem 2 (Consistency of Posterior) *Under the regularity conditions given in Assumptions 3 and 4, the velocity field $\hat{v}_t(x)$ learned by the multilayer perceptron with ReLU activation function achieves*

$$\mathcal{L}_{FM}(\hat{v}, v) \to 0 \quad a.s.$$

when $N \to \infty$. As a result, if Assumptions 1 and 2 hold in addition, we have

$$\sup_{y^* \in \mathcal{Y}} W_2(\hat{\pi}_{\theta|y^*}, \pi_{\theta|y^*}) \to 0$$

almost surely for the generated data $\{y_i, \theta_i\}_{i=1}^N$.

3.3 Inference Theory

With the monotonicity constraint applied on the velocity field $v_t(x)$, any intermediate map T_t is also mono-

tone in x. We can thus construct the following corollary of Theorem 2, verifying that the credible sets $\hat{C}_\tau(y^*)$ defined in (11) converge to the oracle sets (also the MK depth regions) $C_\tau(y^*)$, defined in (12). The proximity is measured by the Hausdorff distance,

$$d_H(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B}\|a-b\|, \sup_{b\in B}\inf_{a\in A}\|a-b\|\right\}.$$

Corollary 3 (Consistency of Credible Sets) *Assume that the true velocity field $v_t(x)$ is monotone in x, and all assumptions in Theorem 2 hold. Then almost surely for the generated data $\{y_i, \theta_i\}_{i=1}^N$, for any $\tau \in (0,1)$, as $N \to \infty$,*

$$\sup_{y^*\in\mathcal{Y}} d_H(\hat{C}_\tau(y^*), C_\tau(y^*)) \to 0.$$

Corollary 3 shows that the posterior credible sets are consistent to the Monge-Kantorovich depth regions. In terms of the inverse maps, consistency result on the MK conditional vector rank in (13) can be constructed as an equivalent of Corollary 3. Consider the estimated inverse map, $\hat{R}_{\theta|y^*} = [\hat{G}(y^*,\cdot)]^{-1}$. For any $\theta \in \Theta$, denote the estimated MK conditional rank function as $\hat{r}_{\theta|y^*}(\theta) = \|\hat{R}_{\theta|y^*}(\theta)\|$. The following result shows that this function acts like a p-value that compares θ with a posterior distribution.

Proposition 4 (MK Conditional Rank) *Under all assumptions in Corollary 3, with $\theta \sim \pi(\cdot \mid y^*)$, for any $\alpha \in (0,1)$ we have*

$$\pi_{\theta|y^*}(\hat{r}_{\theta|y^*}(\theta) > 1-\alpha) \to \alpha$$

uniformly in $y^* \in \mathcal{Y}$, almost surely for the generated data $\{y_i, \theta_i\}_{i=1}^N$ as $N \to \infty$.

It is worth pointing out that the theoretical analysis in this section is more general than the specific Algorithm 1 that we use in the experiment. For any flow matching algorithm that is based on a block-triangular velocity field in (7), under the technical assumptions, the results in this section should hold. More structural constraints can be imposed on the velocity field, e.g. various methods to ensure monotonicity, and we still have the consistency guarantee. For the asymptotic results, the convergence rate with respect to N might vary under different experiment setup and network structures. It might be interesting to explore the finite-sample rates of these consistency results, and we leave those for further study.

4 EXPERIMENTS

In this section, we empirically evaluate our approach from five complementary perspectives. Additional experiments and detailed setup are deferred to Appendix E.

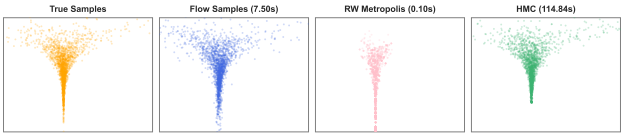


Figure 1: The traditional MCMC methods fail to search the entire region. The sampling time for each method is noted in the respective plot title. Our method requires one-time training (7.5 s); per sampling cost is nearly instantaneous and the model captures the underlying geometry.

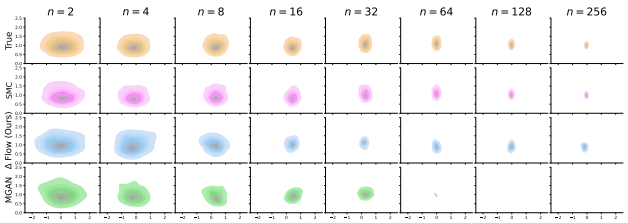


Figure 2: Posterior estimation for Gaussian conjugate model. Our model scales better than M-GAN with the increasing number of observations ($n \geq 64$).

Flexibility of Flow Matching. We first examine the consistency of joint samplers. Sampling difficulties arise when dealing with geometries exhibiting strong heteroskedasticity, such as *Neal's Funnel* (Neal, 2003):

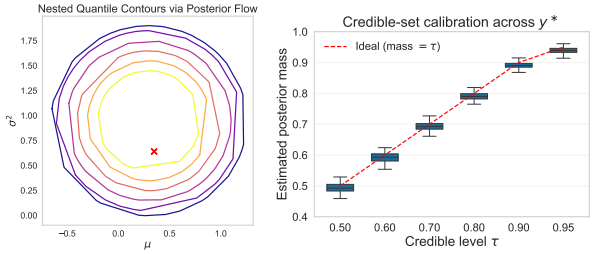
$$\nu \sim \mathcal{N}(0, 3^2), \quad x|\nu \sim \mathcal{N}(0, e^\nu). \tag{14}$$

Standard Monte Carlo Markov Chain (MCMC) methods like Hamiltonian Monte Carlo (HMC) often struggle with Neal's Funnel due to its extreme scale variation on x-axis; tight "necks" require small steps, while wide "mouths" favor large ones. A fixed step size leads to poor mixing and slow convergence. In contrast, generative models learn global transport maps that adapt across the geometry, enabling efficient sampling without region-specific tuning. The timing annotations in Figure 1 highlight the low per-sample cost of our generative approach—flow model requires one-time training (7.5s), and the sampling is nearly free; however, for MCMC methods, we need to run the entire chain every time we sample. We also point out that both MCMC methods fail to cover the entire data space where Random Walk Metropolis Hastings (RWM) fails to cover the "mouth" and Hamiltonian Monte Carlo (HMC) fails to cover the "neck".

Scalability with Data Dimension n. In the second experiment, we show how our model scales with an increasing data dimension n under simple Gaussian conjugate model (see Figure 2). Consider a one-dimensional Gaussian random variable X with parameter μ and σ^2, i.e. $X \mid \mu, \sigma^2 \sim N(\mu, \sigma^2)$. For the

Dataset	Type	Δ-Flow	G-Flow	M-GAN	R-ABC	S-ABC
Gaussian Linear	Training	719.4	512.55	4845.55		
	Inference	<0.01	<0.01	<0.01	0.25	9.85
Gaussian Mixture	Training	818.4	504.88	4653.37		
	Inference	<0.01	<0.01	<0.01	0.37	22.12
Bernoulli GLM	Training	849.7	997.81	4480.55		
	Inference	<0.01	<0.01	<0.01	4.88	20.07
Two Moons	Training	617.3	835.05	4792.71		
	Inference	<0.01	<0.01	<0.01	0.40	18.49
SLCP	Training	906.1	509.87	6088.87		
	Inference	<0.01	<0.01	<0.01	6.5	10.0

Table 2: Computation time (seconds) across SBI benchmarks. For generative methods (Δ-Flow, G-Flow, M-GAN), the table reports one-time training is costly but subsequent sampling is essentially free. For ABC methods, the reported time corresponds to generating posterior samples for each observation, which must be repeated for each inference task.



(a) The τ credible sets (b) Coverage

Figure 3: Panel (a) shows the credible sets with varying τ level ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$) for gaussian conjugate model with $n = 8$. Panel (b) shows the coverage rate of τ-level credible set across 100 simulations. The red dashed line denotes the ideal $y = x$.

conjugate relationship, we assume the prior distribution $\mu \mid \sigma^2 \sim N(\mu_0, \sigma^2/\kappa)$, and $\nu_0\sigma_0^2/\sigma^2 \sim \chi^2(\nu_0)$. For each μ_i, σ_i^2, we generate n many samples x_{ij}, $j \in [n]$, and we repeat the procedure N times. We are given $X \in \mathbb{R}^{N\times n}$ as observed samples and N pairs of parameters $\{\mu_i, \sigma_i^2\}_{i=1}^N$. Our model tracks the posterior distribution without overshrinkage, achieving performance comparable to that of Sequential Monte Carlo (SMC), which directly uses the likelihood. In contrast, the posterior estimated by M-GAN collapses starting from $n = 64$.

One-Time Training, Amortized Inference. We report computation times in Table 2, comparing our method (Δ-Flow) with guided flow (G-Flow) (Zheng et al., 2023), M-GAN (Baptista et al., 2024), rejection ABC (R-ABC), and Sequential Monte-Carlo ABC (S-ABC).

Flow matching methods (both Δ-Flow and guided flow) train substantially faster than adversarial approaches such as M-GAN. Their training time is longer than ABC, but this cost is amortized: once trained, generative models (Δ-Flow, G-Flow, and M-GAN) generate new posterior samples at negligible cost. In contrast, ABC methods incur repeated simulation expense for every new observation, as the inference procedure must be rerun from scratch. See Appendix E.6 for dataset descriptions and further details.

Recovery of Posterior. Section 3 established in theory the consistency of posterior; we now demonstrate this empirically. To quantitatively assess the ability of our sampler to recover the true posterior, we employ the classifier-based two-sample test (C2ST, Lueckmann et al. (2021); Lopez-Paz and Oquab (2016)), which trains a classifier to discriminate between samples from the true and estimated posterior distributions. A classification accuracy approaching 0.5 indicates ideal sampling performance.

Table 3 compares our method (Δ-Flow) against various baselines including rejection ABC (R-ABC), SMC-ABC (S-ABC) (Beaumont et al., 2009), M-GAN (Baptista et al., 2024), guided flow (G-Flow) (Zheng et al., 2023), Kerrigan et al. (2024) and Wildberger et al. (2023).

Our proposed method achieves competitive performance across all five SBI benchmarks. It consistently outperforms the conditional OT baseline of Kerrigan et al. (2024). This might be due to the additional flexibility brought by first estimating joint velocity field over $\mathcal{Y} \times \Theta$. See Appendix C.1 for more detailed discussion.

When compared to GAN-based posterior samplers (M-GAN), the results are mixed: Δ-Flow outperforms M-GAN on Gaussian Linear and SLCP whereas M-GAN performs better on Gaussian Mixture, Bernoulli GLM and SLCP. A key advantage of Δ-Flow is that its training objective is simple (ℓ_2 loss), avoiding adversarial optimization. As a result, Δ-Flow trains significantly faster than M-GAN (See Table 2).

Credible Set with Monotonicity. We revisit the Gaussian conjugate example to illustrate the effect of enforcing monotonicity in the block-triangular flow. As outlined in Section 2.3, monotonicity allows us to generate not only posterior samples but also well-defined Bayesian credible sets. To implement this, we use an input convex neural network (ICNN) to parameterize the monotone components of the map (see Appendix E.4).

Figure 3 summarizes the results: panel (a) illustrates the nested τ-level credible sets produced by the flow, while panel (b) evaluates their calibration by comparing the empirical posterior mass with the nominal level τ. These results demonstrate that the credible sets are not only visually well-behaved contours, but also achieve the expected nominal coverage under the posterior distribution.

Dataset	Δ-Flow	G-Flow	Kerrigan	Wildberger	M-GAN	R-ABC	S-ABC
Gaussian Linear	<u>0.71</u>	**0.69**	0.89	0.97	0.85	0.80	0.73
Gaussian Mixture	0.82	0.85	0.96	0.57	<u>0.73</u>	0.80	**0.65**
Bernoulli GLM	0.92	0.88	0.99	0.61	<u>0.84</u>	0.92	**0.80**
Two Moons	0.74	0.78	0.99	-	<u>0.67</u>	**0.64**	0.70
SLCP	<u>0.93</u>	**0.91**	0.97	0.96	0.98	0.97	0.98

Table 3: C2ST metrics of posterior samplers across SBI benchmark datasets (Lueckmann et al., 2021). Values closer to 0.5 indicate better performance. The best value in each row is bolded and the second best is underlined.

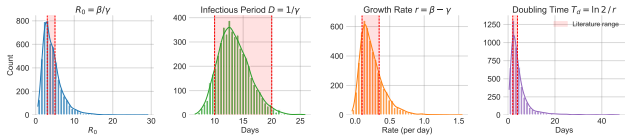


Figure 4: Posterior distributions of four key epidemiological quantities derived from the SIR model fitted to 200-day Illinois COVID-19 data.

Real Data Application: COVID-19 Example. As a real-world data analysis, we used the COVID-19 surveillance data from the Illinois Department of Public Health, which contains daily case, death, and testing counts for 200 days. From these observed quantities, we reconstruct susceptible (S), infected (I), and recovered (R) trajectories (AlQadi and Bani-Yaghoub, 2022), and fit a standard SIR model with β and γ being the transmission and removal rates, respectively. We impose a log-normal prior on (β, γ), sample its posterior using our proposed method, and construct the following quantities of interest: (1) basic reproduction number $R_0 = \beta/\gamma$, (2) days of infection $1/\gamma$, (3) early exponential growth rate $r = \beta - \gamma$, and (4) the doubling time $T_d = \log(2)/r$. The experiment setup and detailed analysis are deferred to Appendix E.5. The Maximum A Posteriori (MAP) estimates yield an infection period of 14.6 days, and basic reproduction number (R_0) around 3.5, both consistent with published estimates. Figure 4 displays the posterior distributions of these four epidemiologically meaningful functionals, with literature reference ranges overlaid for comparison.

Further analysis on the effect of prior on Susceptible-Infected-Recovered (SIR) Model in epidemiology can be found in Appendix E.3.

5 CONCLUDING REMARKS

This paper develops a simulation-based inference method to sample from a multivariate posterior distribution through flow matching. Based on the posterior sampler, our method also provides inferential tools to implement uncertainty quantification, including the construction of Bayesian credible sets and testing the extent of a certain parameter value deviating from the posterior. To apply these inferential tools, our method only requires training the map once. This saves computation time compared to MCMC or ABC, which usually require re-runs with new observations.

One limitation of our method is the reduced performance when the observation dimension (n) is very high. A possible, yet challenging, remedy is to replace the raw data with carefully chosen summary statistics and to train the flow matching model in this lower-dimensional latent space. Another open direction is the extension to high-dimensional parameter spaces, where incorporating sparsity or structural correlation assumptions may be necessary for tractable learning. Moreover, similar to other neural network-based approaches, our method is sensitive to hyperparameter tuning and requires careful implementation choices. In practice, this means that for each dataset, the model should be properly tuned to achieve its best performance. We leave these challenges for future work.

More broadly, our method shares a limitation that is common in push-forward generative models: difficulty with highly multimodal posteriors (Salmona et al., 2022) and extremely heavy-tailed distributions (Wiese et al., 2019; Tam and Dunson, 2025). Our Neal's Funnel experiment shows that moderate heavy-tailed behavior is recoverable, but robustness across all such regimes is not guaranteed. We also leave these challenges for future work to solve.

Acknowledgment

This research was supported in part through the computational resources and staff contributions provided for the Mercury high performance computing cluster at The University of Chicago Booth School of Business which is supported by the Office of the Dean. Veronika Ročková's work is partially supported by NSF/DMS 2515709.

References

Albergo, M. S. and Vanden-Eijnden, E. (2023). Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*.

Alfonso, J., Baptista, R., Bhakta, A., Gal, N., Hou, A., Lyubimova, V., Pocklington, D., Sajonz, J., Trigila, G., and Tsai, R. (2023). A generative flow model for conditional sampling via optimal transport. In *NeurIPS 2023 Workshop Optimal Transport and Machine Learning*.

AlQadi, H. and Bani-Yaghoub, M. (2022). Incorporating global dynamics to improve the accuracy of disease models: Example of a covid-19 sir model. *Plos one*, 17(4):e0265815.

Amos, B., Xu, L., and Kolter, J. Z. (2017). Input convex neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 146–155. PMLR.

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. (2018). Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*.

Baptista, R., Hosseini, B., Kovachki, N. B., and Marzouk, Y. M. (2024). Conditional sampling with monotone gans: From generative models to likelihood-free inference. *SIAM/ASA Journal on Uncertainty Quantification*, 12(3):868–900.

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

Ben-Hamu, H., Puny, O., Gat, I., Karrer, B., Singer, U., and Lipman, Y. (2024). D-flow: Differentiating through flows for controlled generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 3462–3483. PMLR.

Bendel, M., Ahmad, R., and Schniter, P. (2023). A regularized conditional gan for posterior sampling in image recovery problems. *Advances in neural information processing systems*, 36:68673–68684.

Benton, J., Deligiannidis, G., and Doucet, A. (2024). Error bounds for flow matching methods. *Transactions on Machine Learning Research*.

Carlier, G., Chernozhukov, V., and Galichon, A. (2017). Vector quantile regression beyond the specified case. *Journal of Multivariate Analysis*, 161:96–102.

Chemseddine, J., Hagemann, P., Steidl, G., and Wald, C. (2024). Conditional wasserstein distances with applications in bayesian ot flow matching. *arXiv preprint arXiv:2403.18705*.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chernozhukov, V., Galichon, A., Hallin, M., and Henry, M. (2017). Monge-kantorovich depth, quantiles, ranks and signs. *Annals of Statistics*, 45(1):223–256.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real nvp. In *International Conference on Learning Representations*.

Duan, L. L. (2023). Transport monte carlo: High-accuracy posterior approximation via random transport. *Journal of the American Statistical Association*, 118(543):1659–1670.

Farrell, M. H., Liang, T., and Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213.

Generale, A. P., Robertson, A. E., and Kalidindi, S. R. (2024). Conditional variable flow matching: Transforming conditional densities with amortized conditional optimal transport. *arXiv preprint arXiv:2411.08314*.

Giné, E. and Nickl, R. (2021). *Mathematical foundations of infinite-dimensional statistical models*. Cambridge university press.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. (2019). Neutralizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*.

Isobe, N., Koyama, M., Zhang, J., Hayashi, K., and Fukumizu, K. (2024). Extended flow matching: a method of conditional generation with generalized continuity equation. *arXiv preprint arXiv:2402.18839*.

Jiang, H., Wang, Y., and Yang, Y. (2025). Simulation-based inference via langevin dynamics with score matching. *arXiv preprint arXiv:2509.03853*.

Katzfuss, M. and Schäfer, F. (2024). Scalable bayesian transport maps for high-dimensional non-gaussian spatial fields. *Journal of the American Statistical Association*, 119(546):1409–1423.

Kerrigan, G., Migliorini, G., and Smyth, P. (2024). Dynamic conditional optimal transport through simulation-free flows. *Advances in Neural Information Processing Systems*, 37:93602–93642.

Kim, J., Kim, B. S., and Ye, J. C. (2025a). Flowdps: Flow-driven posterior sampling for inverse problems. *arXiv preprint arXiv:2503.08136*.

Kim, J., Zhai, P. S., and Ročková, V. (2025b). Deep generative quantile bayes. volume 258 of *Proceedings of Machine Learning Research*, pages 4141–4149. PMLR.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, K., Han, W., Wang, Y., and Yang, Y. (2025). Optimal transport-based generative models for bayesian posterior sampling. *arXiv preprint arXiv:2504.08214*.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Liu, X., Gong, C., et al. (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.

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

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). Benchmarking simulation-based inference. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 343–351. PMLR.

Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. (2020). Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pages 6672–6681. PMLR.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Muniz-Rodriguez, K., Chowell, G., Cheung, C.-H., Jia, D., Lai, P.-Y., Lee, Y., Liu, M., Ofori, S. K., Roosa, K. M., Simonsen, L., et al. (2020). Doubling time of the covid-19 epidemic by province, china. *Emerging infectious diseases*, 26(8):1912.

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3):705–767.

Oelsner, E. C., Sun, Y., Balte, P. P., Allen, N. B., Andrews, H., Carson, A., Cole, S. A., Coresh, J., Couper, D., Cushman, M., et al. (2024). Epidemiologic features of recovery from sars-cov-2 infection. *JAMA network open*, 7(6):e2417440–e2417440.

Pellis, L., Scarabel, F., Stage, H. B., Overton, C. E., Chappell, L. H., Fearon, E., Bennett, E., Lythgoe, K. A., House, T. A., Hall, I., et al. (2021). Challenges in control of covid-19: short doubling time and long delay to effect of interventions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1829).

Pokle, A., Muckley, M. J., Chen, R. T., and Karrer, B. (2024). Training-free linear image inverses via flows. *Transactions on Machine Learning Research*.

Polson, N. G. and Sokolov, V. (2024). Generative AI for Bayesian Computation. arXiv:2305.14972.

Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L., and Köthe, U. (2020). Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE transactions on neural networks and learning systems*, 33(4):1452–1466.

Rezende, D. J. and Mohamed, S. (2016). Variational inference with normalizing flows.

Salmona, A., De Bortoli, V., Delon, J., and Desolneux, A. (2022). Can push-forward generative models fit multimodal distributions? *Advances in Neural Information Processing Systems*, 35:10766–10779.

Sanche, S., Lin, Y. T., Xu, C., Romero-Severson, E., Hengartner, N., and Ke, R. (2020). High contagiousness and rapid spread of severe acute respiratory syndrome coronavirus 2. *Emerging infectious diseases*, 26(7):1470.

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

Tam, E. and Dunson, D. B. (2025). On the statistical capacity of deep generative models. *arXiv preprint arXiv:2501.07763*.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. (2023). Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*.

Wang, Y. and Ročková, V. (2022). Adversarial bayesian simulation. *arXiv preprint arXiv:2208.12113*.

Wiese, M., Knobloch, R., and Korn, R. (2019). Copula & marginal flows: Disentangling the marginal from its joint. *arXiv preprint arXiv:1907.03361*.

Wildberger, J., Dax, M., Buchholz, S., Green, S., Macke, J. H., and Schölkopf, B. (2023). Flow matching for scalable simulation-based inference. *Advances in Neural Information Processing Systems*, 36:16837–16864.

Zheng, Q., Le, M., Shaul, N., Lipman, Y., Grover, A., and Chen, R. T. (2023). Guided flows for generative modeling and decision making. *CoRR*.

Zhou, X., Jiao, Y., Liu, J., and Huang, J. (2023). A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, 118(543):1837–1848.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes. Appendix A is spared for further clarification in notations.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes. See Appendix 4.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes. See Appendix E.6

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes. All the assumptions are clearly outlined in Section 3.
   (b) Complete proofs of all theoretical results. Yes. See Appendix D
   (c) Clear explanations of any assumptions. Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes. Appendix E.6 includes the link to the anonymized code repository.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes. See Appendix E.6.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes. See Appendix E.6. We could not reveal the full statement on the usage of computing infrastructure, but specification have been disclosed.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Not Applicable
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable
   (d) Information about consent from data providers/curator. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Supplementary Materials

A MATHEMATICAL NOTATION

In this section we collect and define mathematical notation used throughout the paper.

We denote the Euclidean norm of a vector $x \in \mathbb{R}^d$ by $\|x\| = \sqrt{\sum_{i=1}^d x_i^2}$, and the Frobenius norm of a matrix $A \in \mathbb{R}^{d\times d}$ by $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$. For a function $f : \mathbb{R}^d \to \mathbb{R}$, we denote the gradient by $\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_d}\right)^\top$, and the Hessian by $\nabla^2 f(x) \in \mathbb{R}^{d\times d}$, with entries $\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$.

For a vector-valued function $v : \mathbb{R}^d \to \mathbb{R}^d$, the Jacobian matrix is denoted by $J_v(x) = \nabla v(x) \in \mathbb{R}^{d\times d}$, with entries $[J_v(x)]_{ij} = \frac{\partial v_i}{\partial x_j}$. The divergence of v is defined as the trace of the Jacobian

$$\nabla \cdot v(x) = \sum_{i=1}^d \frac{\partial v_i}{\partial x_i} = \operatorname{tr}(\nabla v(x)).$$

This quantity measures the net rate at which probability "flows out" of a point under the vector field v, and appears in the continuity equation that governs time-dependent probability densities.

For a measurable map $T : \mathbb{R}^d \to \mathbb{R}^d$ and a probability measure μ on $\mathbb{R}^d$, the pushforward measure $T_\sharp \mu$ is defined as the distribution of the random variable $T(X)$ when $X \sim \mu$. Formally, for any Borel set $A \subset \mathbb{R}^d$, we define

$$T_\sharp \mu(A) := \mu(T^{-1}(A)).$$

The Wasserstein-2 distance between two probability measures μ and ν on $\mathbb{R}^d$ is defined as

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, \mathrm{d}\gamma(x, y),$$

where $\Gamma(\mu, \nu)$ is the set of all couplings (joint distributions) with marginals μ and ν.

The Hausdorff distance between sets $A, B \subset \mathbb{R}^d$ is defined as

$$d_H(A, B) = \max\left\{ \sup_{a\in A} \inf_{b\in B} \|a - b\|, \sup_{b\in B} \inf_{a\in A} \|a - b\| \right\}.$$

B RELATED WORKS

Bayesian posterior sampling has been addressed through a variety of generative methods. Here we review relevant approaches categorized by underlying technique.

Normalizing Flows. A normalizing flow (NF) is a sequence of invertible transformations that maps a base distribution $p_0(x)$ (e.g., a simple Gaussian) to a more complex target distribution $p_1(x)$. The probability density function (PDF) is computed using the change of variables formula. Let $f : \mathbb{R}^d \to \mathbb{R}^d$ be an invertible transformation with Jacobian determinant $|\det J_f(x)|$, then the density transformation follows

$$p_1(x) = p_0(f^{-1}(x)) \left|\det J_{f^{-1}}(x)\right|,$$

where $J_f(x) = \frac{\partial f}{\partial x}$ is the Jacobian matrix of the transformation. One example is the real-valued non-volume preserving transformation (real NVP) (Dinh et al., 2017), which introduces block-triangular coupling layers to

ensure tractable computation of the Jacobian determinant. In each layer, the input is split into two subsets, where one subset remains unchanged while the other is transformed conditionally. Specifically, affine transformations of the form

$$x'_{1:d} = x_{1:d}, \quad x'_{d+1:D} = x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}),$$

yield a triangular Jacobian matrix whose log-determinant is a simple sum over outputs of $\mathbf{s}$, where $s(x_{1:d}) \in \mathbb{R}^{D-d}$ is a scale function, $t(x_{1:d}) \in \mathbb{R}^{D-d}$ is a translation function both parametrized by neural networks, and $\odot$ denotes an element-wise product. The model provides exact likelihood computation via the change of variables formula; however, training is constrained by architectural choices that ensure invertibility (e.g., affine coupling layers).

Normalizing flows have been applied directly to simulation-based inference. Ardizzone et al. (2018) use invertible neural networks to learn the posterior in an amortized fashion, while Radev et al. (2020) propose BayesFlow, which couples a summary network with a conditional normalizing flow to perform amortized Bayesian inference from simulated data. Both methods condition on static observations and require invertible architectures with tractable Jacobians, whereas our approach avoids invertibility constraints entirely by learning a velocity field via flow matching.

Continuous Normalizing Flow A Continuous Normalizing Flow (CNF) (Chen et al., 2018) generalizes NF to continuous-time dynamics by modeling the transformation as a solution to an ODE, $\frac{dx}{dt} = v_t(x_t, \omega)$, $(t \in [0, 1])$ where $v_t(x_t, \omega)$ is the velocity field with the learnable parameters $\omega \in \Omega$, for example, by neural networks. The model still relies on the modified version of the change of variable formulas.

Lemma 2 (Instantaneous Change of Variables (Chen et al., 2018)) *Let $x(t) \sim p_t$ be a finite continuous random variable, and $dx_t = v_t(x_t)dt$ be the velocity field. If v is uniformly Lipschitz continuous in $x(t)$ and continuous in t, then the change in log probability follows a differential equation*

$$\frac{\partial p_t\big(x(t)\big)}{\partial t} = -\mathrm{tr}\Big(\frac{\mathrm{d}v}{\mathrm{d}x(t)}\Big)$$

Diffusion Model Diffusion models (Song et al., 2020) add noise to data (Forward Process) and learn to reverse this process via score estimation (Reverse Process). While powerful, they require computationally expensive training and sampling, and rely on accurate score estimates. The likelihood computation is possible through integration of the score function. The SDE-based diffusion model has a connection to the probability flow ODE relying on the Fokker–Planck equation.

$$\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t,$$

where $f_t(x_t)$ is a drift function, g_t is the noise scaling, and B_t is a Brownian motion.

$$\mathrm{d}x_t = [f_t(x_t) - g_t^2 \nabla_x \log p_t(x_t)]\,\mathrm{d}t + g_t\,\mathrm{d}\tilde{B}_t$$

where $\nabla_x \log p_t(x_t)$ is the score function.

$$\begin{aligned}
\mathrm{d}x &= f(t,x)\,\mathrm{d}t + g(t)\,\mathrm{d}B_t \\
\Leftrightarrow \frac{\partial p_t(x)}{\partial t} &= -\nabla_x(f p_t) + \frac{1}{2}g^2\nabla_x^2 p_t \quad \text{(Fokker–Planck)} \\
&= -\nabla_x(f p_t) + \frac{1}{2}g^2\nabla_x(p_t \nabla_x \log p_t) \\
&= -\nabla_x\Big((f - \frac{1}{2}g^2\nabla_x \log p_t)p_t\Big) \\
\Rightarrow \mathrm{d}x &= \tilde{f}(x,t)\,\mathrm{d}t, \qquad \tilde{f}(x,t) = f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t(x)
\end{aligned}$$

B.1 Guidance vs. Conditional Sampling

Guidance techniques are employed to steer the sampling process of diffusion models toward specific desired outcomes. Two prominent methods are classifier guidance (Song et al., 2020) and classifier-free guidance (Ho and Salimans, 2022). It starts with the simple score decomposition of the form

$$\nabla \log p_t(x|y) = \nabla \log p_t(y|x) + \nabla \log p_t(x).$$

Introduced by Song et al. (2020), the classifier guidance utilizes an external classifier, $p(y|x)$, trained to predict the class label from a given image. Note that the model is still trained marginally $p(x)$ and the classifier signal only plays a role in the sampling procedure. During sampling, the gradients from this classifier are combined with the diffusion model's score estimates to guide the generation process toward images that are more likely to belong to the desired class. This approach effectively biases the model to produce outputs aligned with specific class labels.

The classifier-free guidance (CFG) by Ho and Salimans (2022) eliminates the need for an external classifier by training a single diffusion model capable of both conditional and unconditional generation. During training, the model learns to handle inputs with and without conditioning information (e.g., class labels) by introducing the Bernoulli variable indicating the conditionality. At inference, the model interpolates between these two modes by adjusting the guidance scale.

$$\tilde{v}_t(x|y) = (1 - w) \cdot v_t(x|\emptyset) + w \cdot v_t(x|y) \tag{15}$$

This interpolation (15) can be mathematically represented as a weighted combination of the conditional and unconditional score estimates.

Guided flow (Zheng et al., 2023) is the adaptation of classifier-free guidance (CFG) on flow matching from the diffusion setting. As in Ho and Salimans (2022), the score decomposition strategy is used for time dependent classifiers:

$$\tilde{v}_t(x|y) = v_t(x) + b_t \cdot \nabla \log p_\phi(y|x, t)$$

The primary advantage of guidance methods lies in their ability to control the fidelity of generated samples with respect to the conditioning information. However, we note that the goal of guidance is not exact conditional estimation; the guided distribution is a tempered approximation whose fidelity depends on the guidance scale.

B.2 Posterior Sampling via Generative Models

We now review generative approaches that directly target posterior distributions.

Generative Adversarial Network (GAN) based posterior sampling Wang and Ročková (2022) and Baptista et al. (2024) leverage matching of joint samples of data and parameters against pairs of marginal data and noise. Baptista et al. (2024) imposes additional constraints such as monotonicity and block-triangular structure, enhancing the stability of posterior estimation. Our method shares the block-triangular framework with Baptista et al. (2024) but replaces adversarial learning with a flow matching objective, resulting in simpler training and faster convergence (see Table 2). Bendel et al. (2023) build upon the conditional GAN framework (Mirza and Osindero, 2014) and propose a regularization scheme to enhance training stability and improve sample diversity, with a primary focus on linear inverse problems.

Diffusion based posterior sampling Diffusion-based posterior sampling, by Chung et al. (2023), incorporates classifier signals directly into training to handle general noisy inverse problems, avoiding explicit measurement consistency constraints.

$$\frac{dx}{dt} = -\frac{\beta(t)}{2}x - \beta(t)\left(\nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)\right) + \sqrt{\beta(t)}\,\mathrm{d}\bar{B}_t$$

The posterior mean estimation via Tweedie formula for the likelihood approximation during the diffusion posterior sampling:

$$\hat{x}_0 := \mathbb{E}[x_0|x_t] \approx \frac{1}{\sqrt{\bar{\alpha}(t)}}\left(x_t + (1 - \bar{\alpha}(t))s^*_\theta(x_t, t)\right)$$

Flow based posterior sampling Several recent works apply flow models to posterior estimation. Ben-Hamu et al. (2024) and Pokle et al. (2024) propose training-free methods for conditional generation in linear inverse problems, modifying the generative process at inference time to enforce data consistency rather than learning a posterior distribution directly. FlowDPS (Kim et al., 2025a) adapts the diffusion posterior sampling framework of Chung et al. (2023) to flow matching models, but continues to rely on score estimation via learned gradients of log densities. In contrast, our method departs from score-based training entirely, using a structured velocity decomposition to enable exact posterior recovery.

Wildberger et al. (2023) present a simulation-based framework using flow models for posterior estimation. They employ forward simulations to generate training data and then train a conditional flow to approximate the posterior, conditioning on static observations y. This setup is conceptually close to ours, but methodologically distinct: Wildberger et al. (2023) condition on the raw observation as a fixed input to the velocity network, whereas we learn a joint velocity field over $(\mathcal{Y} \times \Theta)$ in which both data and parameter components evolve over time. By decomposing the joint transport into marginal and conditional components, our approach distributes the geometric complexity across two cooperative sub-problems rather than placing the entire representational burden on a single conditional network.

B.3 Conditional Optimal Transport and Related Frameworks

A number of concurrent and recent works approach conditional sampling through the lens of optimal transport, and we discuss their relationship to our method here.

Kerrigan et al. (2024) generalize the Benamou–Brenier theorem to the conditional setting, learning a conditional velocity field that transports a source distribution to the target while keeping the conditioning variable y fixed (identity coupling in $\mathcal{Y}$). Their theoretical focus is on minimizing the conditional Wasserstein distance between source and target. Chemseddine et al. (2024) introduce a conditional Wasserstein distance $W_{p,Y}$ and derive an OT Bayesian Flow Matching algorithm that penalizes mass transport in the conditioning variable's space, arriving at a comparable velocity target in the Euclidean setting. Both methods learn a *conditional* velocity of the form $\frac{d\theta_t}{dt} = g_t(\theta_t; y)$, where y enters as a static input. We note that under linear interpolation in Euclidean space with matched data marginals, the target velocity fields of these approaches and ours coincide; the methods differ in what is held fixed during training and in the theoretical guarantees that follow.

Generale et al. (2024) address a different goal: forecasting dynamic systems where data arrives as a time series. They perform sequential Bayesian inference in which the source distribution for the flow at each step is the previous posterior approximation, not a fixed noise distribution. Isobe et al. (2024) propose an extension to conditional flow matching that addresses the problem of flowing between conditions (e.g., from $\pi(\theta \mid y_1)$ to $\pi(\theta \mid y_2)$) by integrating over a path in the conditional space, optimizing for smoothness of the mapping with respect to the conditioned variables. Both works address fundamentally different inference tasks from ours.

B.4 Transport-Based Methods for Bayesian Computation

Several works leverage transport maps to accelerate or replace traditional Bayesian computation, without necessarily using flow matching.

Makkuva et al. (2020) learn static OT maps between two fixed distributions using ICNN parameterizations of convex potentials, with the goal of recovering a global Monge map via direct optimization. In our approach, convexity is not required for posterior sampling itself but serves as an optional constraint for constructing credible sets (Section 2.3). Li et al. (2025) cast posterior learning directly as an OT problem and estimate a transport map pushing the prior onto the posterior; in simple settings the OT formulation yields linear maps that act as efficient posterior samplers. Our method shares the high-level goal of learning a deterministic prior-to-posterior map but obtains it implicitly through a flow matching objective rather than solving the OT problem directly.

Duan (2023) build randomized approximate transport maps to precondition MCMC (Transport Monte Carlo), improving asymptotic efficiency while still relying fundamentally on Monte Carlo chains. Similarly, Hoffman et al. (2019) train a neural transport map to "neutralize" the geometry of the posterior and then run HMC on the transformed space. In both cases, the transport map serves as an auxiliary preconditioner rather than a stand-alone sampler. Our method produces posterior samples directly by integrating the learned velocity field, without requiring Hamiltonian dynamics or accept–reject mechanics.

Katzfuss and Schäfer (2024) estimate a triangular Rosenblatt/Knothe transport map for high-dimensional spatial fields using Gaussian process regressions, maximin spatial ordering, and Bayesian regularization. Although both their method and ours involve triangular transport ideas, the approaches differ substantially: we do not assume spatial structure or impose GP priors, and we learn the transport implicitly via flow matching rather than through structured GP regressions.

Jiang et al. (2025) address simulation-based inference via score matching combined with Langevin dynamics, targeting settings with high-dimensional parameter spaces. Polson and Sokolov (2024) propose quantile-based deep generative samplers trained with pinball losses. Our method differs from both in using flow matching to learn time-dependent conditional flows, and in its ability to handle multivariate posteriors with joint velocity decomposition.

C TECHNICAL REMARKS

C.1 Comparison with Other Flow-based Methods

There are two notable concurrent works that also use flow matching to learn the posterior distribution. Chemseddine et al. (2024) minimizes the conditional Wasserstein distance to learn the velocity field leading to posterior distribution, while Kerrigan et al. (2024) generalized Benamou–Brenier optimization, leading to an equivalent optimal transport map toward the posterior.

Optimal Transport vs. Consistency. One key difference between these methods and ours is the main objective of learning the velocity field. Kerrigan et al. (2024) and Chemseddine et al. (2024) aim to minimize the conditional Wasserstein distance between the source distribution and the target posterior distribution. Our method, on the contrary, is mainly driven by the theoretical guarantee like Theorem 2, i.e., the consistency of the learned posterior distribution compared to the true underlying posterior, in terms of Wasserstein distance. The theoretical motivation is different.

The method of Kerrigan et al. (2024), for instance, is theoretically constrained on the specific geodesic path in the conditional Wasserstein space. Our method, on the other hand, allows for flexible interpolation schemes (linear, geodesic, or as in Albergo and Vanden-Eijnden (2023)), without affecting consistency guarantees of Section 3. The consistency results only rely on the accuracy of the learned joint velocity field, rather than the specific interpolation path.

Our method also allows for the possibility to enforce monotonicity of the transport map using ICNN, leading to the Monge-Kantorovich conditional vector quantile map (Chernozhukov et al., 2017). This allows us to generate nested Bayesian credible sets and compute conditional vector ranks, inferential tools that are difficult to obtain with standard generative models. On the contrary, enforcing such monotonicity in Kerrigan's framework would conflict with the objective of learning the true optimal transport path. Note that this should not be seen as a comparative advantage of our method per se, but rather the result of a different emphasis in method design. Our method trades off strict OT optimality to gain rigorous uncertainty quantification capabilities.

Learning the Posterior Directly vs. Learning the Joint First. Technically, the use of block-triangular structure is different for our method. Methods like Chemseddine et al. (2024), Wildberger et al. (2023), and Kerrigan et al. (2024) directly consider the posterior velocity field over Θ space alone, with observation y^* fixed. Our proposed method, on the other hand, learns a joint, block-triangle velocity field over $\mathcal{Y} \times \Theta$. The block-triangular map theory ensures that an accurately learned joint distribution will lead to an accurately learned posterior distribution.

Therefore, our approach allows for an extra layer of flexibility. An important aspect of this flexibility is the *decoupling of marginal transport from conditional transport*. Fixed-y^* methods use the form of velocity $\dot{\theta}_t = v_t(y^*, \theta_t)$, and learns the velocity as a function of the static, raw observation y^*. This places a heavy representational burden on the network for learning v_t. Our approach, on the contrary, considers both $\dot{y}_t = f_t(y_t)$, i.e. marginal transport, and $\dot{\theta}_t = g_t(y_t, \theta_t)$, which now depends on dynamic y_t. That is to say, the task of learning posterior velocity field is now decomposed into two sub-problems. By offloading the modeling of the data geometry to f_t, the conditional network g_t only needs to learn a simpler mapping: how parameters relate to the flow of data, rather than how they relate to static y^*. Theoretically, this allows the smoothness assumptions on f_t and g_t

to be more mild. By decomposing the joint transport into marginal and conditional components, we effectively distribute the geometric complexity of the target distribution across two simpler, cooperative functions, thus relieving representational burden.

Moreover, our proposed method allows the inference to be guided by the marginal transport f_t, rather than relying on the network to implicitly interpolate the geometry of y from static training data points. This may help in practice, especially when the conditioned observation y^* falls into a low-likelihood area. In our framework, the marginal velocity field f_t acts as a structural guide, effectively bridging the geometric gaps between training data points. This additional flexibility may explain improved empirical performance of our method over, e.g., Kerrigan et al. (2024), on several SBI benchmarks; see Table 3.

C.2 ICNN Representability Issue

In this subsection, we seek to elaborate the ICNN representability issue discussed in Remark 3. Under the scheme (10), our method learns the velocity field

$$\hat{g}_t(y_t, \theta_t) = \nabla_{\theta_t} \hat{\psi}_t(y_t, \theta_t)$$

by learning a convex function ψ_t with ICNN. It is indeed reducing representational capacity, in that not all monotone maps can be represented by a monotone velocity. Specifically, under linear interpolation path, the velocity can be written as

$$g_t(y, \theta \mid x_0, x_1) = G(y^*, \theta_0) - \theta_0.$$

Indeed, learning a velocity $\hat{g}_t(y, \theta)$ monotone in θ ensures that $\nabla_\theta g_t \succeq 0$, which implies $\nabla G \succeq I$. Therefore, those non-expanding monotone velocity fields cannot be represented. However, we can practically mitigate this limitation by initializing the source distribution with a much smaller variance than the target. This ensures the optimal transport map is naturally expansive ($\nabla G \succeq I$), bringing the problem within the representational capacity of the monotone velocity field.

In order to address the representational issue more fundamentally, we propose an alternative velocity formulation that allows for contractive maps. We can instead learn the velocity field

$$\hat{g}_t(y_t, \theta_t) = \nabla_{\theta_t} \hat{\psi}_t^*(y^*, \theta_t) - \theta_0,$$

where $\hat{\psi}_t^*$ is a convex function learned by ICNN. One technical issue is that flow matching requires θ_t as input of $\hat{g}_t$, instead of θ_0. To address this issue, we can replace it by $\theta_0 \approx \theta_t - t\hat{g}_t$ (under linear interpolation path). The velocity field is therefore formulated as

$$\hat{g}_t^*(y_t, \theta_t) = \frac{\nabla_{\theta_t} \hat{\psi}_t(y^*, \theta_t) - \theta_t}{1 - t}.$$

Under such formulation, it is possible for contracting monotone maps G to be represented by such a velocity g_t^*. Therefore this formulation avoids the loss of representation, and we can still learn the velocity by ICNN.

D PROOF OF THEORY IN SECTION 3

D.1 Proof of Theorem 1

Note that the true joint distribution p_1 can be recovered by passing p_0 through the map T as in (1), and the joint velocity field learned from the neural network leads to an estimated map $\hat{T}$ that transports p_0 to $\hat{p}_1$, an estimated joint distribution. Formally,

$$\hat{p}_1 = \hat{T}_\sharp p_0, \quad p_1 = T_\sharp p_0.$$

It is known that $\hat{p}_1$ is consistent to p_1 when the velocity is learned accurately enough in terms of the flow matching objective function $\mathcal{L}_{\text{FM}}$ given in (2). In fact, by Theorem 1 in Benton et al. (2024), under Assumptions 1 and 2, when $\mathcal{L}_{\text{FM}} \leq \varepsilon_N^2$, we have

$$W_2(\hat{p}_1, p_1) \leq \varepsilon_N \exp\left\{ \int_0^1 L_t \, \mathrm{d}t \right\}. \tag{16}$$

Recall that the estimated posterior $\hat{\pi}(\theta \mid y^*)$ is recovered by the partial map $\hat{G}(y^*, \cdot)$, which is a slice of the joint map $\hat{T}$. Also note that the true underlying map $G(y^*, \cdot)$ recovers the true posterior $\pi(\theta \mid y^*)$, a consequence of Lemma 1 and Theorem 2.4 in Baptista et al. (2024). Formally,

$$\hat{\pi}_{\theta|Y=y^*} = \hat{G}(y^*, \cdot)_\sharp p_0^\Theta, \quad \pi_{\theta|Y=y^*} = G(y^*, \cdot)_\sharp p_0^\Theta.$$

We shall now find the relation between $W_2(\hat{p}_1, p_1)$ and $W_2(\hat{\pi}_{\theta|Y}, \pi_{\theta|Y})$. Under the block-triangular map framework (1),

$$\begin{aligned} W_2^2(\hat{p}_1, p_1) &= W_2^2(\hat{T}_\sharp p_0, T_\sharp p_0) \\ &= \int \|\hat{T}(y_0, \theta_0) - T(y_0, \theta_0)\|^2 p_0(\mathrm{d}y_0, \mathrm{d}\theta_0) \\ &= \int \|\hat{F}(y_0) - F(y_0)\|^2 p_0^{\mathcal{Y}}(\mathrm{d}y) + \int \|\hat{G}(\hat{F}(y_0), \theta_0) - G(F(y_0), \theta_0)\|^2 p_0(\mathrm{d}y_0, \mathrm{d}\theta_0). \end{aligned}$$

Since both terms are nonnegative, the upper bound (16) implies that either term is upper bounded by the same rate. By Assumption 2, since $\hat{v}_t(x)$ is spatially Lipschitz, i.e. $\|\hat{v}_t(x) - \hat{v}_t(x')\| \leq L_t \|x - x'\|$, we obtain a similar Lipschitzness in the joint estimated map $\hat{T}$ by integrating over time. As a slice of $\hat{T}$, this property is also shared by $\hat{G}$,

$$\|\hat{G}(\hat{F}(y_0), \theta_0) - \hat{G}(F(y_0), \theta_0)\| \leq L \|\hat{F}(y_0) - F(y_0)\|.$$

By triangular inequality, we get

$$\|\hat{G}(F(y_0), \theta_0) - G(F(y_0), \theta_0)\| \leq \|\hat{G}(F(y_0), \theta_0) - \hat{G}(\hat{F}(y_0), \theta_0)\| + \|\hat{G}(\hat{F}(y_0), \theta_0) - G(F(y_0), \theta_0)\|.$$

Using $(a+b)^2 \leq 2a^2 + 2b^2$ and taking an expectation of the square over $p_0(y_0, \theta_0)$, we get

$$\begin{aligned} &\int \|\hat{G}(F(y_0), \theta_0) - G(F(y_0), \theta_0)\|^2 p_0(\mathrm{d}y_0, \mathrm{d}\theta_0) \\ &\leq 2W_2^2(\hat{p}_1, p_1) + (2L - 2) \int \|\hat{F}(y_0) - F(y_0)\|^2 p_0^{\mathcal{Y}}(\mathrm{d}y_0) \\ &\leq 2Le^{2L} \varepsilon_N^2. \end{aligned}$$

Therefore, we arrive at

$$\begin{aligned} \int W_2^2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) \pi_Y(\mathrm{d}y) &= \int \|\hat{G}(y, \theta_0) - G(y, \theta_0)\|^2 p_0^\Theta(\mathrm{d}\theta_0) \pi_Y(\mathrm{d}y) \\ &= \int \|\hat{G}(F(y_0), \theta_0) - G(F(y_0), \theta_0)\|^2 p_0(\mathrm{d}\theta_0, \mathrm{d}y_0) \\ &\leq 2Le^{2L} \varepsilon_N^2. \end{aligned}$$

This finishes the proof of Theorem 1.

D.2 Proof of Theorem 2

Although the way our method trains $\hat{v}_t(x)$ is through minimizing $\mathcal{L}_{\mathrm{CFM}}$ in (4) instead of $\mathcal{L}_{\mathrm{FM}}$ in (2), since they both have equal gradients with respect to the neural network parameters φ (see (5)). Indeed, Proposition 1 in Albergo and Vanden-Eijnden (2023) also verifies that

$$v = \arg\min_{\hat{v}} \mathcal{L}_{\mathrm{FM}}(\hat{v}, v) = \arg\min_{\hat{v}} \mathcal{L}_{\mathrm{CFM}}(\hat{v}, v).$$

It is interesting to point out that it is usually impossible for $\mathcal{L}_{\mathrm{CFM}}$ to converge to zero. Instead, it converges to a constant unrelated to $\hat{v}$,

$$\int_t \mathbb{E}_{p_t(x|z), q(z)} \|v_t(x \mid z)\|^2 \, \mathrm{d}t - \int_t \mathbb{E}_{p_t(x)} \|v_t(x)\|^2 \, \mathrm{d}t.$$

There is generally no guarantee on this quantity. Despite $\mathcal{L}_{\mathrm{FM}}$ being intractable, we can still apply the theoretical framework of Farrell et al. (2021). Note that the true underlying velocity v is the minimizer of the alternative

objective function $\mathcal{L}_{\text{CFM}}$, and the estimated velocity field $\hat{v}$ is obtained by empirically minimizing the empirical version of $\mathcal{L}_{\text{CFM}}$,

$$\hat{v} = \underset{\hat{v}\in\mathcal{V}_{\text{MLP}}}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} \|\hat{v}_{t_i}(x_i) - v_{t_i}(x_i \mid z_i)\|^2,$$

where $\mathcal{V}_{\text{MLP}}$ is the function class learnable by the multilayer perceptron, t_i is the i-th sampled time point, and $x_i = (y_i, \theta_i)$ is the i-th generated sample.

By Theorem 1 in Farrell et al. (2021), there exists a constant $C > 0$ unrelated to N such that for each dimension $1 \le k \le n + d$, with probability no less than $1 - \exp\{-N^{\frac{\beta}{\beta+n+d}} \log^8 N\}$,

$$\mathbb{E}_{t,x}(\hat{v}_{t,k}(x) - v_{t,k}(x))^2 \le C\left(N^{-\frac{\beta}{\beta+n+d}} \log^8 N + \frac{\log\log N}{N}\right).$$

Note that the expectation is taken over all variables of the functions $\hat{v}$ and v, i.e. $t \sim \text{Unif}(0,1)$ and $x \sim p_1$. Adding all dimensions up, it is exactly the flow matching objective function $\mathcal{L}_{\text{FM}}(\hat{v}, v)$. That is,

$$\mathcal{L}_{FM} \le C(n+d)\left(N^{-\frac{\beta}{\beta+n+d}} \log^8 N + \frac{\log\log N}{N}\right) := \varepsilon_N^2,$$

with probability no less than $1 - (n+d)\exp\{-N^{\frac{\beta}{\beta+n+d}} \log^8 N\}$. This upper bound provides a finite-sample rate for ε_N^2. Moreover, observe that

$$\sum_{N=1}^{\infty} \mathbb{P}(\mathcal{L}_{\text{FM}} > \varepsilon_N^2) \le (n+d)\sum_{N=1}^{\infty} \exp\{-N^{\frac{\beta}{\beta+n+d}} \log^8 N\} < \infty.$$

Then by the Borel-Cantelli's Lemma, we have $\mathcal{L}_{\text{FM}} < \varepsilon_N^2$ almost surely for the generated dataset. By Theorem 1, we have

$$\mathbb{E}_{\pi_Y} W_2^2(\hat{\pi}_{\theta|Y}, \pi_{\theta|Y}) = O\left(N^{-\frac{\beta}{\beta+n+d}} \log^8 N\right),$$

a rate that is polynomial to N. This immediately translates to

$$W_2(\hat{\pi}_{\theta|Y}, \pi_{\theta|Y}) \to_{\pi_Y} 0.$$

To prove uniform convergence, first note that $\mathcal{Y}$ is compact by Assumption 3. Second, from Assumptions 1 and 2 we can also show that $W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y})$ is equicontinuous in y. To see this, note that by triangular inequality, for any $y, y' \in \mathcal{Y}$,

$$|W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) - W_2(\hat{\pi}_{\theta|y'}, \pi_{\theta|y'})| \le W_2(\hat{\pi}_{\theta|y}, \hat{\pi}_{\theta|y'}) + W_2(\pi_{\theta|y}, \pi_{\theta|y'}).$$

By the Lipschitzness assumption in the velocity field, the overall map is also Lipschitz. The right hand side can be upper bounded by $C\|y - y'\|$ for some constants $C > 0$. Therefore, the 2-Wasserstein distance $W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y})$ is equicontinuous in y.

Since $\mathcal{Y}$, also the support of π_Y, is compact, every open ball in $\mathcal{Y}$ has strictly positive π_Y-measure. Now we can prove uniform convergence by contradiction. Assume $\sup_{y\in\mathcal{Y}} W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) > \delta$ for some $\delta > 0$, then there exists some $\bar{y} \in \mathcal{Y}$ such that $W_2(\hat{\pi}_{\theta|\bar{y}}, \pi_{\theta|\bar{y}}) > \delta$. By equicontinuity, for any y in the neighboring ball $B(\bar{y}, \epsilon)$ we have

$$W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) > \delta - C\epsilon.$$

If we pick $\epsilon = \delta/2C$, then we have $W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) > \delta/2$ within $y \in B(\bar{y}, \delta/2C)$. But then

$$\mathbb{E}_{\pi_Y} W_2^2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) \ge \int_{B(\bar{y},\delta/2C)} (\delta/2)^2 \pi_Y(\mathrm{d}y),$$

which is a positive constant unrelated to N. This contradicts the L^2 convergence result. Therefore we conclude that

$$\sup_{y\in\mathcal{Y}} W_2(\hat{\pi}_{\theta|y}, \pi_{\theta|y}) \to 0.$$

D.3 Proof of Corollary 3 and Proposition 4

Note that

$$W_2^2(\hat{\pi}_{\theta|y^*}, \pi_{\theta|y^*}) = \int \|\hat{G}(y^*, \theta_0) - G(y^*, \theta_0)\| p_0^{\Theta}(\mathrm{d}\theta_0).$$

By Theorem 2, this quantity converges to zero uniformly over $y^* \in \mathcal{Y}$. Since Θ is compact according to Assumption 3, and both maps G and $\hat{G}$ are Lipschitz on Θ as a result of Assumptions 1 and 2, by the same technique as in the proof of Theorem 2, we can derive the following uniform convergence guarantee that for any $y^* \in \mathcal{Y}$,

$$\sup_{\theta_0 \in \Theta} \|\hat{G}(y^*, \theta_0) - G(y^*, \theta_0)\| \to 0. \tag{17}$$

Recall the definition of Bayesian credible sets in (11) and oracle sets in (12). Consider any $\theta^* \in C_\tau(y^*)$. Due to the compactness of $S^d(\tau)$, there exists $\theta_0^* \in S^d(\tau)$ such that $\theta = G(y^*, \theta_0^*)$. In fact, we can identify that θ_0^* as the Monge-Kantorovich conditional vector rank $R_{\theta|y^*}(\theta^*)$. By transporting θ_0^* through the vector quantile map, we arrive at $\hat{\theta}^* := G(y^*, \theta_0^*)$. From the definition (11), $\hat{\theta}^*$ is in the credible set $\hat{C}_\tau(y^*)$. Combining this argument with (17), we conclude that for any $\theta^* \in C_\tau(y^*)$, there exists $\hat{\theta}^* \in \hat{C}_\tau(y^*)$ such that $\|\hat{\theta}^* - \theta^*\| \to 0$. This implies

$$\sup_{\theta \in C_\tau(y^*)} \inf_{\hat{\theta} \in \hat{C}_\tau(y^*)} \|\hat{\theta} - \theta\| \to 0.$$

On the other direction, we can use the same proof technique and conclude that for any $\hat{\theta}^* \in \hat{C}_\tau(y^*)$, there exists $\theta^* \in C_\tau(y^*)$ such that $\|\hat{\theta}^* - \theta^*\| \to 0$, and thus

$$\sup_{\hat{\theta} \in \hat{C}_\tau(y^*)} \inf_{\theta \in C_\tau(y^*)} \|\hat{\theta} - \theta\| \to 0.$$

Due to the uniform convergence guarantee in (17), the choice of $y^* \in \mathcal{Y}$ is arbitrary. The choice of $\tau \in (0, 1)$ is also arbitrary. Therefore,

$$\sup_{y^* \in \mathcal{Y}} d_H(\hat{C}_\tau(y^*), C_\tau(y^*)) \to 0.$$

This finishes the proof of Corollary 3.

For the proof of Proposition 4, observe that under the estimated posterior distribution recovered by $\hat{G}(y^*, \cdot)$, the event $\{\hat{r}_{\theta|y^*}(\theta) > 1 - \alpha\}$ is equivalent to $\{\theta_0 \in S^d(1) \setminus S^d(1 - \alpha)\}$. Therefore,

$$\hat{\pi}_{\theta|y^*}(\hat{r}_{\theta|y^*}(\theta) > 1 - \alpha) = p_0^{\Theta}(S^d(1)) - p_0^{\Theta}(S^d(1 - \alpha)) = \alpha.$$

The uniform convergence in Theorem 2 implies weak convergence $\hat{\pi}_{\theta|y^*} \Rightarrow \pi_{\theta|y^*}$. Recall that Θ is a compact domain, and the continuity of the maps are ensured by Assumptions 1 and 2. Denote

$$\hat{F}(\tau) = \hat{\pi}_{\theta|y^*}(\hat{r}_{\theta|y^*}(\theta) \le \tau), \quad F(\tau) = \pi_{\theta|y^*}(\hat{r}_{\theta|y^*}(\theta) \le \tau).$$

From the weak convergence and continuity, we get

$$|\hat{F}(1 - \alpha) - F(1 - \alpha)| \to 0.$$

This implies that $\pi_{\theta|y^*}(\hat{r}_{\theta|y^*}(\theta) > 1 - \alpha) \to \alpha$, finishing the proof of Proposition 4.

E ADDITIONAL EXPERIMENTS

To complement the evaluation in the main text, this section presents targeted experiments that probe two aspects of the proposed flow-matching posterior sampler: (1) monotonicity and recovery of the Bayesian credible set (Section E.4) and (2) the effect of scaling for recovering the true posterior (Section E.2).

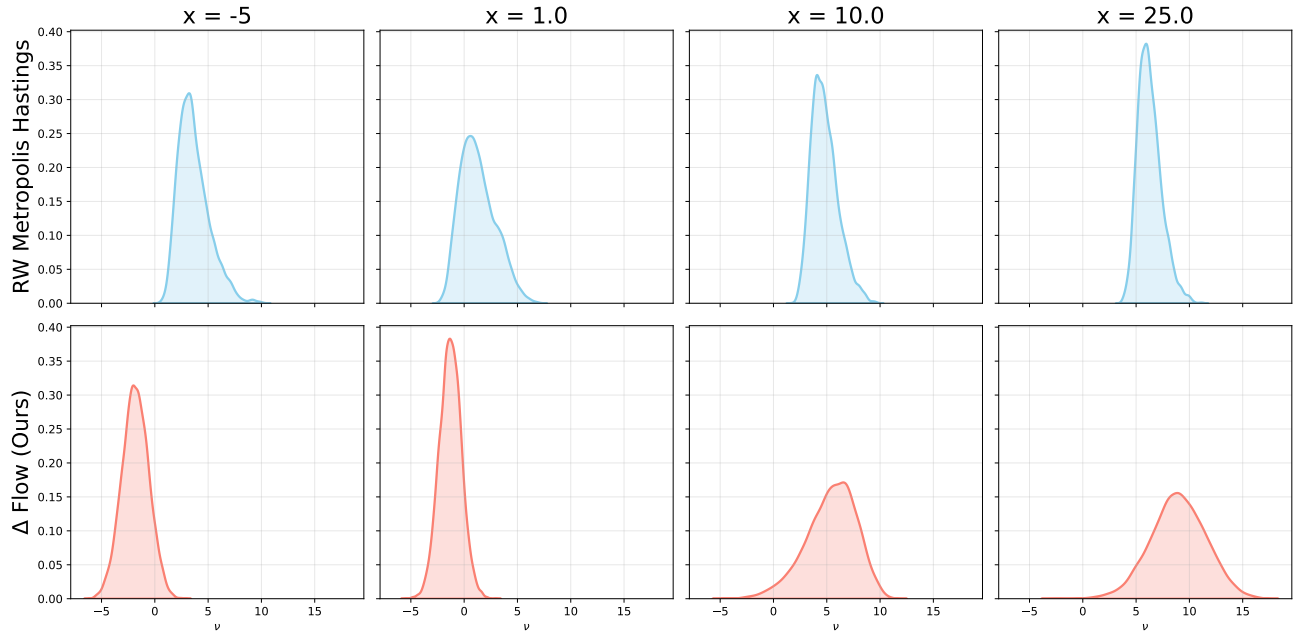


Figure 5: Posterior recovery of Neal's funnel (14) at different observation $x = [-5, 1, 10, 25]$, compared with standard Metropolis-Hastings (MH) sampler (first row). Overall, the learned posterior by flow (second row) matches with MH estimations across the practical range of x.

E.1 Posterior for Neal's Funnel

Here, we revisit the Neal's funnel (Section 4) for posterior sampling. As described earlier, Neal's funnel is characterized by (14). Accordingly, by rearranging (14), the posterior distribution is known up to normalizing constant:

$$p(\nu \mid x) \propto \exp\left[-\frac{1}{2}\left(\frac{\nu^2}{3^2} + e^{-\nu}x^2 + \frac{\nu}{2}\right)\right]. \tag{18}$$

The posterior distribution can be well estimated by Metropolis-Hastings (MH) with symmetric proposal proposal

$$\nu^* \sim q(\cdot|\nu^{(t-1)}) = \mathcal{N}(\nu^{(t-1)}, \sigma^2),$$

with corresponding acceptance probability being

$$\alpha = \min\left\{1, \exp\left(\ell(\nu^*; x) - \ell(\nu^{(t-1)}; x)\right)\right\},$$

where ℓ is the log posterior (unnormalized) (18). We evaluate the posterior distribution on different x values. Here, the problem is favorable for MH as unnormalized posterior distribution is log concave in ν given x (18). Overall, the learned posterior by flow is a faithful surrogate across the practical range of x. For $x = 25$, where ν is likely to be from a large value and the data is scarce, flow model expands the tail slightly more compared to MH, see Figure 5.

E.2 Effect of Proper Scaling

Recall Assumption 2, which states that the approximate flow $\hat{v}_t(x)$ is differentiable with respect to x and t, and for each t, there exists a constant L_t such that $\hat{v}_t(x)$ is L_t-Lipschitz in x. In other word, for every fixed t, there exists $L_t \in \mathbb{R}$,

$$\|v_t(x) - v_t(x')\| \leq L_t\|x - x'\| \quad \forall x, x' \in \mathbb{R}^{n+p}.$$

The posterior consistency relies on Theorem 1, which asserts that under Assumption 1 and 2, the expected squared 2-Wasserstein distance between the estimated and true posterior is bounded by $2\epsilon_N^2 L e^{2L}$, where $L = \int_0^1 L_t dt$. It shows that the sampling error exhibits an exponential growth with the Lipschitz constant L_t.

For $x_t = [y_t^\top, \theta_t^\top]^\top = [y_{1t}, ..., y_{nt}, \theta_{1t}, ..., \theta_{pt}]^\top$, we can denote the mean and covariance as $\mu_t = \mathbb{E}x_t$ and $\Sigma_t = \mathrm{Cov}(x_t) = \mathbb{E}[(x_t - \mu_t)(x_t - \mu_t)^\top]$. Because x_t follows the deterministic flow $\frac{\mathrm{d}}{\mathrm{d}t}x_t = v_t(x_t)$, we could write the time derivative of covariance Σ_t as

$$\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_t = \mathbb{E}\big[(\nabla_x v_t)(x_t)\Sigma_t + \Sigma_t(\nabla_x v_t)^\top(x_t)\big]. \tag{19}$$

Since Assumption 2 implies that the Lipschitz constant is a gradient in magnitude in that, for fixed t,

$$L_t = \sup_{x \neq x'} \frac{\|v_t(x) - v_t(x')\|}{\|x - x'\|} = \sup_x \|\nabla_x v_t(\cdot)\|_{\mathrm{op}}, \tag{20}$$

and thus $\|\nabla_x v_t(x)\|_{\mathrm{op}} \leq L_t$. Combining (19) with (20), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\Sigma_t \preceq 2L_t\Sigma_t \quad \Leftrightarrow \quad \Sigma_t \preceq \exp\Big(2\int_0^t L_s ds\Big)\Sigma_0, \quad t \in [0,1]$$

where $\Sigma_0 = I_{n+p}$ when we start from independent source distribution.

Any entries of $\nabla_x v_t$ numerically being large pushes the Lipschitz constant L_t up (20). For example, if j-th coordinate of x has a tiny scale, e.g. $\sigma(x^{(j)}) = 10^{-3}$, while other coordinates are order 1. A change of $\pm 10^{-3}$ along j-axis is negligible in Euclidean distance, $\|x - x'\|$. However, if the network output varies by even a modest amount in that direction, the quotient can blow up

$$\frac{|v_t^{(i)}(x) - v_t^{(i)}(x')|}{|x^{(j)} - x'^{(j)}|} \approx \frac{0.05}{10^{-3}} = 50,$$

meaning a single column of $\nabla_x v_t$ already contributes 50 to $\|\nabla_x v_t\|_{\mathrm{op}}$. Intuitively, a skinny coordinate makes gradients look artificially big and inflates the Lipschitz constant L_t. Rescaling the thin coordinate to unit variance removes the magnifying effect and lowers Lipschitz constant over t.

In optimization perspective, since neural velocity fields are trained by stochastic optimizer, if one coordinate has exceptionally large gradient it would not properly updating across the coordinates. Standardizing x_1 — or equivalently, reparameterizing y and θ so that their empirical covariance is close to the identity ($\widehat{\Sigma}_1 \approx I$) — restores balance and enables $\hat{v}_t$ to be trained effectively toward the target objective.

Inspired from this theoretical observation (20), we show that appropriate rescaling on the joint space $\mathcal{Y} \times \Theta$ such that each dimension possesses similar variance is important in learning the joint distribution, by effectively aiding in controlling the Lipschitz constant through an SIR example, where the range of two spaces ($\mathcal{Y}$ and Θ) are highly heterogeneous. See the differences in variance for the parameters and manual summary statistics space in Figure 6. The effect of proper standardization is shown in Figure 7 in terms of an accuracy of joint recovery.

E.3 Effect of Varying Priors on SIR Model

To examine the sensitivity of posterior inference to the choice of prior distribution, we conduct an additional experiment using the classical Susceptible-Infected-Recovered (SIR) epidemiological model. The SIR model describes the evolution of three compartments-susceptible, infected, and recovered, according to a nonlinear system of ordinary differential equations parameterized by a rate of contagion β and a mean recovery rate γ. Specifically, the model is defined with a system of ODEs,

$$\frac{\mathrm{d}}{\mathrm{d}t}[S, I, R] = \left[-\frac{\beta S}{N}I, \frac{\beta S}{N}I - \gamma I, \gamma I\right],$$

where S is the number of individuals susceptible to be infected, I is the number of individuals infected, R is the number of individuals recovered from the disease, and $S + I + R$, the total population, is fixed. For all experiments, we simulate trajectories over a fixed time horizon of 300 days using normalized initial conditions, with a fixed total population size.

We consider three distinct prior families over the parameters (β, γ), each commonly used in the epidemiological and simulation-based inference literature. The first is a log-normal prior (Figure 8), where $\log \beta$ follows a normal

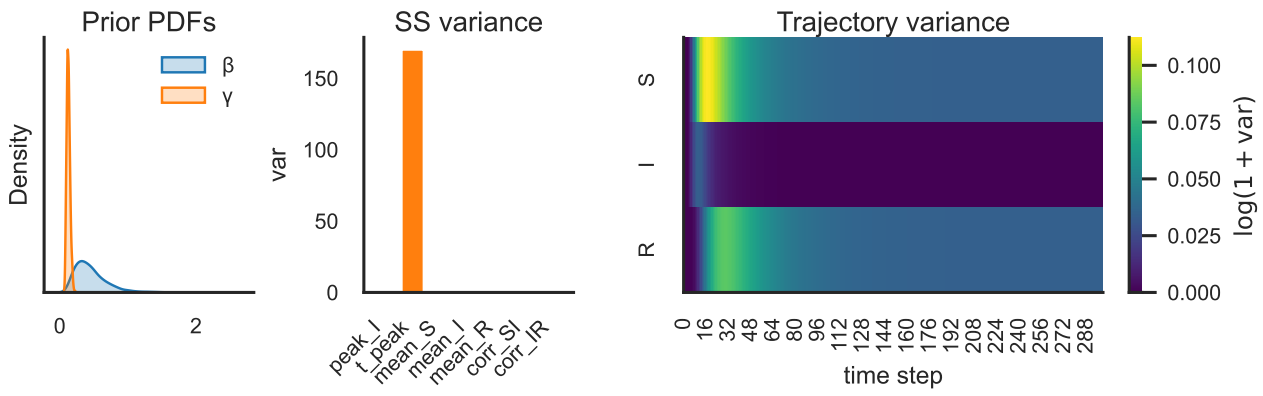
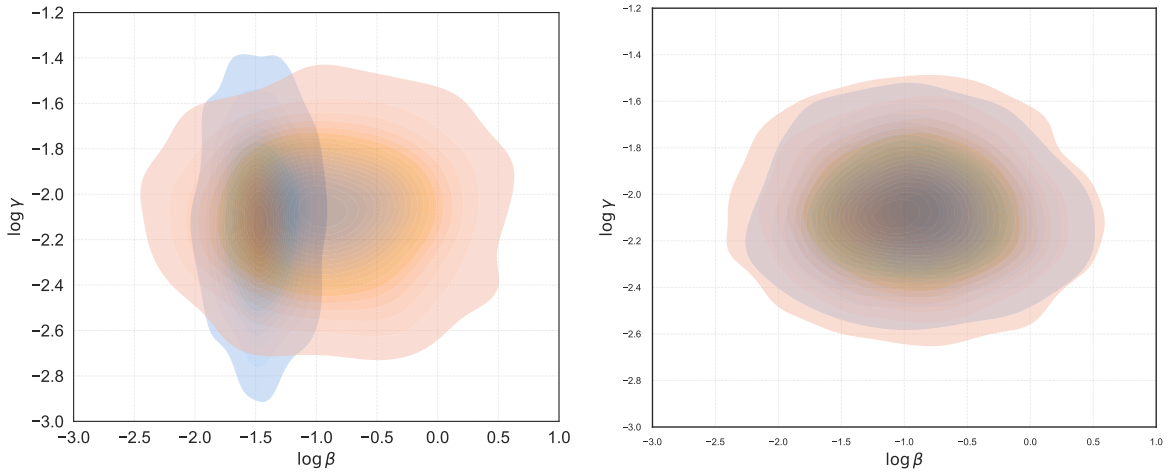


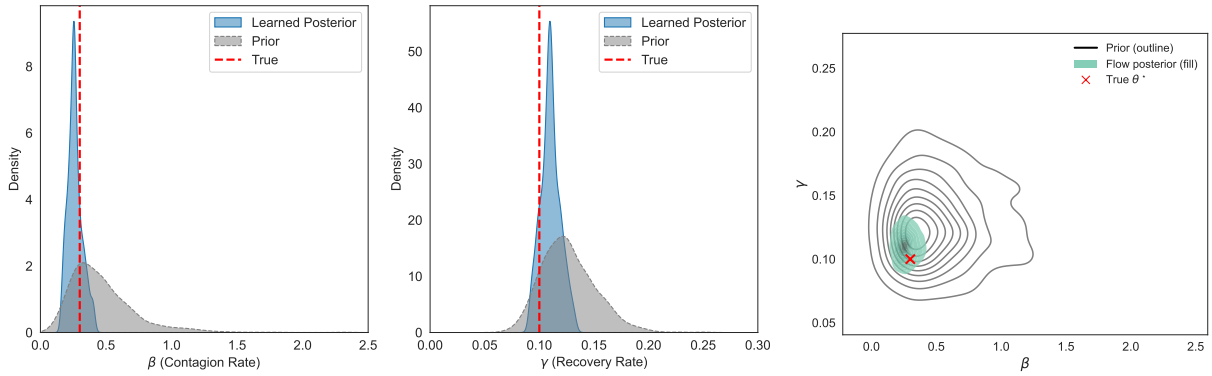
Figure 6: The prior for the infection rate β is much more dispersed than the prior for the recovery rate γ: the sample variance of β is 0.053 whereas γ is only 0.0007 (ratio $\approx 79:1$). Among the seven handcrafted summary statistics, dispersion differs by two orders of magnitude: the variance of the epidemic peak size (peak I) is 166, while the growth-slope and correlation summaries are all below 0.05. Point-wise trajectory variances $\mathrm{Var}(S_t, I_t, R_t)$ are shown on a $\log(1 + \mathrm{var})$ scale. The susceptible (S) and recovered (R) compartments start around 0.11 and decay steadily, whereas the infectious component (I) remains almost deterministic throughout, never exceeding 0.003 (below the first colour step). Together, the three panels illustrate that the statistical scales relevant to learning the joint flow range from 10^{-3} (I-variance) to 10^{2} (peak I variance).



(a) Without Standardization

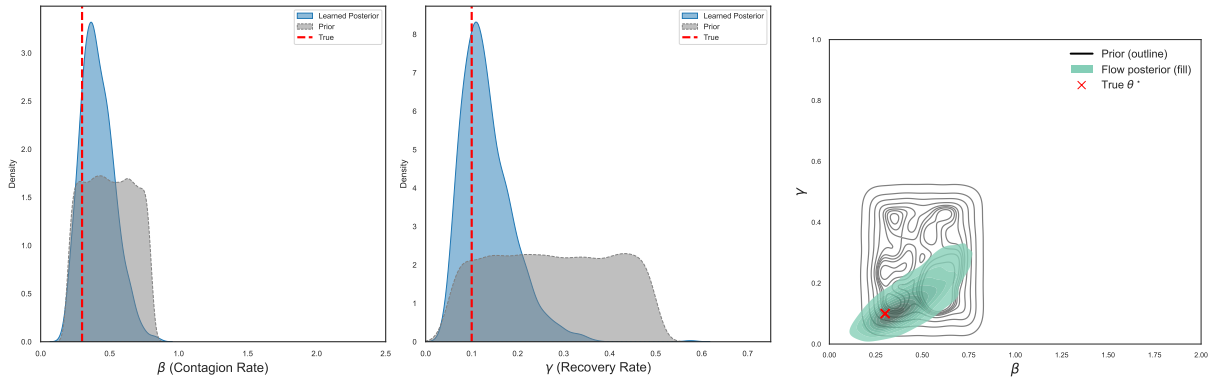
(b) With Standardization

Figure 7: The figures show the estimated (blue) and true prior distribution (orange). Our method hinges on the correct estimation of joint distribution $(y, \theta) \sim L(y|\theta)\pi(\theta)$, and if the joint estimation is not correct, we cannot ensure the quality of posterior estimation. Note the effect of standardization on correctly estimating the prior distribution. Without normalization (a), the model even fail to learn the prior distribution for the parameters.



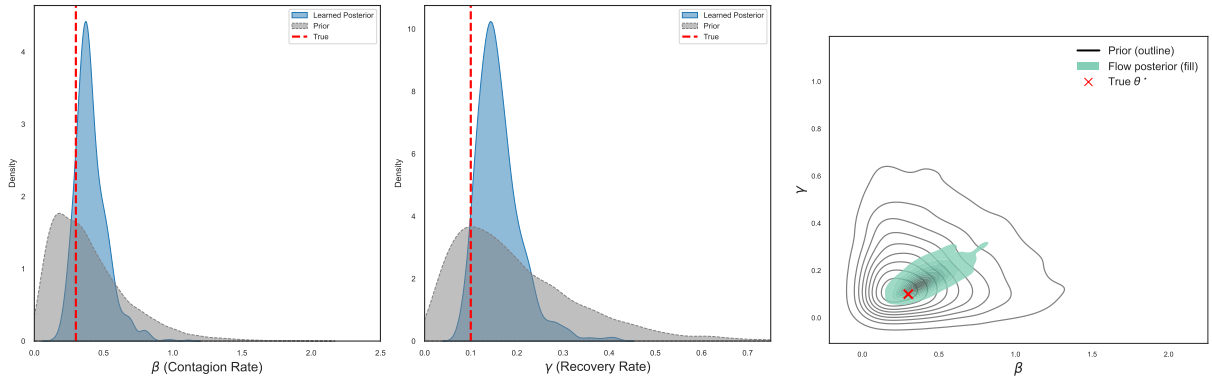
(a) Marginal Posterior distribution (b) Joint Posterior estimation

Figure 8: The log-normal prior for both parameters β and γ. The posterior (blue) for the parameters (β, γ) compared to their prior (grey) are given. The estimated posterior regions include the true parameter values (red dotted line or cross).



(a) Marginal Posterior distribution with Uniform prior (b) Joint Posterior estimation

Figure 9: Supplement to Figure 8. The log-normal prior has been replaced with uniform prior.



(a) Marginal Posterior distribution with Gamma prior (b) Joint Posterior estimation

Figure 10: Supplement to Figure 8. The log-normal prior has been replaced with gamma prior.

distribution with mean $\log 0.4$ and standard deviation 0.5, and $\log \gamma$ follows a normal distribution with mean $\log(1/8)$ and standard deviation 0.2. The second is a uniform prior (Figure 9), in which β is sampled uniformly between 0.2 and 0.8, and γ between 0.05 and 0.5, representing uninformative priors over a reasonable parameter range. The third is a gamma prior (Figure 10), where β is drawn from a Gamma distribution with shape 2.0 and scale 0.2 (mean 0.4), and γ from a Gamma distribution with shape 2.0 and scale 0.1 (mean 0.2), capturing skewed distributions with heavier right tails.

For each prior, we simulate 5,000 trajectories of the SIR system by sampling parameter values from the prior and integrating the system forward in time. We then compute a fixed set of handcrafted summary statistics for each trajectory, including peak infection level, time to peak, average number of the susceptible, infected and recovered, and cross-correlations. These summary statistics serve as inputs to a flow matching model, which is trained to learn a mapping from summary statistics to posterior parameters. The flow model architecture and other training hyperparameters are held fixed across prior settings to isolate the impact of the prior alone.

Results of this experiment indicate that the learned posterior distributions are substantially influenced by the choice of prior, even when the ground-truth parameters remain constant. When trained under the log-normal prior, the posterior concentrates more tightly around the true values ($\beta = 0.3, \gamma = 0.1$), with moderate skewness reflecting the asymmetric uncertainty encoded by the prior. Under the uniform prior, the resulting posterior is more diffuse, particularly in directions orthogonal to the dominant likelihood ridge, due to the lack of structural regularization from the prior. The gamma prior induces posteriors with visibly heavier right tails, especially in γ, consistent with the prior's long-tailed density and its interaction with the forward dynamics.

E.4 Monotonicity via Input Convex Neural Network (ICNN) (Amos et al., 2017)

Let $\psi : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}$ be a Partial Input Convex Neural Network (PICNN) (Amos et al., 2017) that is convex in $\theta \in \Theta$ and is unrestricted for the rest $[y^\top, t] \in \mathcal{Y} \times [0, 1]$. Define

$$g_t(y, \theta) := \nabla_\theta \psi_t(y, \theta). \tag{21}$$

Because the map $\theta \mapsto \psi.(\cdot, \theta)$ is C^1 and convex in θ, its gradient is monotone with respect to θ. Substituting the ordinary multilayer perceptron in g_t with the gradient of convex function (21) therefore imposes the required monotonicity with respect to θ by design, yet preserves full modeling flexibility in y and t.

For training, we need to generate x_0 by sampling y_0 from n-dimensional spherical uniform that is $y_0 = r \times \phi$, and sampling θ_0 from a d-dimensional spherical uniform, that is $\theta_0 = \tau \times \xi$, where $\phi \sim \text{Unif}(\mathcal{S}^{n-1}(1))$, $\xi \sim \text{Unif}(\mathcal{S}^{d-1}(1))$, and $r, \tau \sim \text{Unif}[0, 1]$. This step is not needed, for example, when we use isotropic gaussian as our source distribution, where partial coordinate still follows the isotropic gaussian.

For generating a credible set, as described in Section 2.3, given y^*, we can start from spherical uniform: we draw directions ξ from spherical uniform distribution on d-dimensional unit sphere and fix the radius τ, setting $\theta_0 = \tau \times \xi$. Evolving θ_0 under the joint velocity field (f_t, g_t) under fixed y^* transports the spherical shell $S^{d-1}(\tau)$ into the shell of the credible set $\partial \hat{C}_\tau(y^*) \subset \Theta$, which contains the parameter endpoints of all trajectories whose source norm equals τ. Monotonicity of g_t guarantees radial ordering: if $\tau_1 < \tau_2$ then every trajectory launched from the inner sphere remains inside the image of the outer sphere, hence $\hat{C}_{\tau_1} \subset \hat{C}_{\tau_2}$. As a consequence the family $\{\hat{C}_\tau(y^*)\}_{0 \le \tau \le 1}$ forms nested parameter regions whose posterior mass increases monotonically with τ. Choosing τ so that the spherical source contains probability $1 - \alpha$ yields $\hat{C}_\tau$ a $(1 - \alpha)$-Bayesian credible set.

We first verify that the learned velocity indeed satisfies the monotonicity condition by checking that the inner product mentioned in Section 2.3 is non-negative on a dense grid of (θ, θ') pairs. We then confirm the theoretical nesting property by inspecting credible sets obtained at successively larger radii: visualizations in parameter space show that sets corresponding to $\tau_1 < \tau_2$ never intersect improperly, thereby validating both the monotonicity enforcement via the PICNN gradient and the credibility interpretation of the spherical-launch construction (Figure 11).

We revisit SIR example with additional monotonicity enforced on g_t. Figure 12 shows both marginal and joint posterior distribution covers the true underlying parameter values ($(\beta, \gamma) = (0.3, 0.1)$). Like in gaussian conjugate experiment (Figure 11), the estimated credible sets are non-crossing across different τ values and display nested structure.

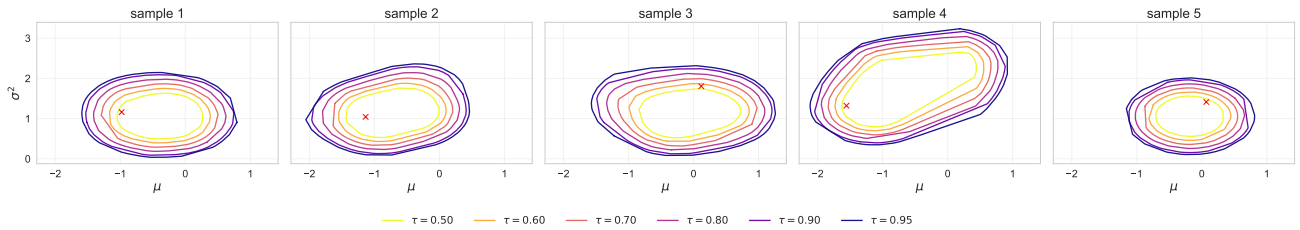


Figure 11: Bayesian credible set with increasing level of τ for Gaussian conjugate model described in Section 4. Here, $n = 4$. Each column represents different realization of observed X. Observe that there is nested structure as we vary the τ-level, and there is no crossing due to the monotonicity guaranteed by modeling through PICNN.

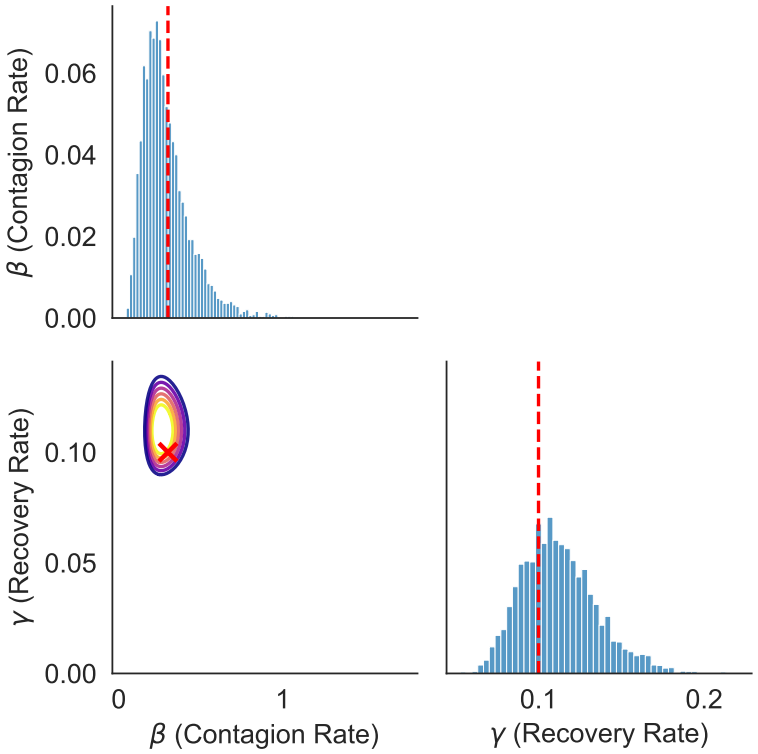


Figure 12: Estimated posterior distribution for β (the contagion rate) and γ (recovery rate). The corresponding $\tau = [0.5, 0.6, 0.7, 0.8, 0.9, 0.99]$ level credible sets are given as nested contours. The lightest contour corresponds to 0.5-level credible set and the darkest contour corresponds to 0.99-level credible set. The red vertical line on the histogram denotes the true parameter values ($(\beta, \gamma) = (0.3, 0.1)$).

	New cases (N_t)	Daily test	Death	S	I	R
Mean	1,375	25,286	4,950	32,336	18,372	111,496
SD	819	18,935	3,160	20,294	10,147	77,738
Min	0	0	0	14	0	0
Max	5,594	149,273	8,672	153,951	35,019	260,877

Table 4: Descriptive Statistics of the Illinois COVID-19 Dataset (March 10, 2020-September 25, 2020)

E.5 Real Data Example: COVID-19 data analysis

For our analysis we used publicly available daily COVID-19 surveillance data* from the Illinois Department of Public Health. The dataset spans March 10, 2020 to September 25, 2020 (200 days) and reports the date, total and new confirmed cases, total and new deaths, and the total number of individuals tested. From these quantities, we reconstructed the susceptible (S), infected (I), and recovered (R) compartments following the procedure in AlQadi and Bani-Yaghoub (2022).

The number of actively infected individuals on day t was defined as the rolling 14-day sum of newly confirmed infections, $I(t) = \sum_{k=t-13}^{t} N_k$. To estimate the susceptible population, we used both testing information and incubation dynamics. Assuming a 5-day incubation period, individuals who test positive on up to day $k+6$ are treated as susceptible on day k. Thus, $S(t)$ consists of those who test negative on day t together with individuals who will test positive at least six days later, reflecting cases still in incubation. Because the dataset does not record recoveries, we approximated $R(t)$ by assuming that 95–99% of confirmed infections eventually recover. Letting $D(t)$ denote cumulative deaths, the recovered compartment was estimated as $R(t) = p \cdot \big(\sum_{k=t-15}^{t} N_k\big) - D(t),\ p \in [0.95, 0.99]$. See the summary statistics of original data and reconstructed S,I,R value in Table 4.

In our analysis, we adopt log-normal priors on the transmission and removal rates,

$$\beta \sim \text{LogNormal}(\log 0.4, 0.5^2), \qquad \gamma \sim \text{LogNormal}\big(\log(1/8), 0.2^2\big).$$

Given a parameter draw (β, γ), we simulate the SIR trajectories via numerical solution of the ODE system and train our triangular flow model on pairs (β, γ, simulated data). Once the model is trained, we generate posterior samples conditioned on the observed 200-day Illinois COVID-19 data. To interpret the posterior we focus on epidemiologically meaningful functionals of (β, γ): (1) the basic reproduction number $R_0 = \beta/\gamma$, (2) the infectious period $D = 1/\gamma$, (3) the early exponential growth rate $r = \beta - \gamma$, and (4) the corresponding doubling time $T_{\text{d}} = \log 2/r$.

Results. The recovery rate γ implies an infectious period of approximately 14.6 days, consistent with published estimates of 10-20 days (Oelsner et al., 2024). The implied basic reproduction number $R_0 = \beta/\gamma \approx 3.5$ aligns with early-pandemic estimates for COVID-19 transmission in the United States, where R_0 values between 3 and 5 have been documented in the absence of interventions (Sanche et al., 2020). The MAP estimate of the early exponential growth rate $r = \beta - \gamma$ is 0.17 per day, corresponding to a doubling time is approximately 3.98 days. This is consistent with empirical estimates of 2-4 day doubling times reported during the early, unconstrained phase of COVID-19 spread (Muniz-Rodriguez et al., 2020; Pellis et al., 2021). See Figure 4 for the posterior distribution of each statistics and Table 5 for maximum a posteriori (MAP) estimate with 90% marginal credible bound. Although our estimates are not directly comparable to those in AlQadi and Bani-Yaghoub (2022), which rely on an extended SIR model with additional forcing terms, the overall magnitudes of the inferred parameters agree with the established epidemiological literature.

E.6 Experiment Details

Unless otherwise noted, every experiment is driven by the same *source distribution* – an isotropic Gaussian $\mathcal{N}(0, I_{n+d})$ in the joint space $\mathcal{Y} \times \Theta$. All code and exact configurations are available in our GitHub repository.

*https://www.kaggle.com/datasets/hadeelalqadi/uscovid-19data

	β	γ	R_0	$1/\gamma$ (days)	r	T_d (days)
MAP	0.24	0.07	3.54	14.58	0.17	3.98
LB	0.12	0.05	1.53	9.46	0.04	1.23
UB	0.64	0.11	9.25	18.75	0.56	14.26

Table 5: Maximum A Posteriori (MAP) estimates of parameters and key statistics with 90% marginal credible interval. Lower bound (LB) and Upper bound (UB) are given in the table.

Neal's Funnel Both the data marginal velocity f_t and the conditional velocity g_t are implemented as four-layer fully connected networks with hidden width 64 and ELU activations. We train for $20,000$ iterations using Adam (lr $= 0.001$) (Kingma, 2014).

Gaussian Conjugate The flow architecture mirrors that of the funnel experiment but with hidden width starting from 64 and increasing proportionally to the size of n; the lower dimensionality allows a lighter network without loss of accuracy.

SIR Denote the state trajectory by $y = (S_t, I_t, R_t)_{t=1}^{T}$ and let $\theta = (\beta, \gamma)$. The data marginal velocity f_t is a four-layer MLP with hidden width 64, while g_t is the gradient of a two-block *spectrally normalized* ResNet,

where each residual block has width 128 and SiLU activations. Input preprocessing follows the pipeline in Section 4 and Section E.2: (i) the raw time series is collapsed to the seven-dimensional summary vector $\big[\text{peak}_I,\ t_{\text{peak}},\ \overline{S},\ \overline{I},\ \overline{R},\ \text{corr}(S,I),\ \text{corr}(I,R)\big]$; (ii) every summary is standardized to zero mean and unit variance over the training batch; (iii) the parameter is log–transformed $(\log\beta, \log\gamma)$ and then standardized. Training uses AdamW (Loshchilov and Hutter, 2019) with weight-decay 10^{-4}, batch size 256, learning-rate 2×10^{-4}.

Coverage experiment In this experiment we assess the calibration of credible sets produced by our flow model under the Gaussian conjugate setup (see Section 4 for its exact setup). For each synthetic dataset y, we construct a candidate τ-level credible set $C_\tau(y)$ by sampling $n_{\text{set}} = 2000$ noise vectors uniformly within an inner τ-ball, pushing them through the learned flow conditioned on the data, y, and taking the convex hull of the resulting (μ, σ)-samples. This yields nested sets across $\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ as in Figure 11. To evaluate whether $C_\tau(y)$ captures the intended posterior mass, we independently draw $M = 2000$ posterior samples from the flow (conditioned on the same observation y) and compute the proportion that lies inside each convex hull.

We repeat this procedure for 100 independent draws of y. The resulting coverage proportions are aggregated across repetitions and visualized as boxplots in Figure 3, alongside the ideal diagonal $y = \tau$. The figure shows that the empirical mass closely tracks the nominal level τ.

Details for Table 2 and Table 3 For each task, we use the package `sbibm` (Lueckmann et al., 2021). We draw 10 ground-truth parameter and observation pairs from the prior and simulator, build 10k reference posterior samples per observation, and evaluate algorithms across simulation budgets from 1k to 100k. For rejection ABC and Sequential Monte-Carlo ABC (SMC-ABC) (Beaumont et al., 2009), we also use the `sbibm` implementation. Performance is reported primarily via classifier two-sample tests (C2ST) using an MLP with two hidden layers (width = 10×data dimension) and five-fold cross-validation; runtimes are also recorded (Table 2).

For guided flow, we follow Zheng et al. (2023) method with fixed the guidance strength $\omega = 1.5$.

We adopt the following five benchmark models for evaluation, all of which has analytical solution for posterior computation we could compare against.

1. **Gaussian Linear**
   - Prior: $\theta \sim \mathcal{N}(0, 0.1 \odot I)$, $\theta \in \mathbb{R}^{10}$
   - Likelihood: $\mathbf{x} \mid \theta \sim \mathcal{N}(\mathbf{m}_\theta = \theta, \mathbf{S} = 0.1 \odot I)$, $\mathbf{x} \in \mathbb{R}^{10}$
2. **Simple Likelihood and Complex Posterior (SLCP)**
   - Prior: $\theta \sim \text{Unif}(-3, 3)$, $\theta \in \mathbb{R}^5$

- Likelihood: $\mathbf{x} \mid \theta = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$, $\mathbf{x}_i \sim \mathcal{N}(\mathbf{m}_\theta, \mathbf{S}_\theta)$, where $\mathbf{m}_\theta = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}$, $\mathbf{S}_\theta = \begin{bmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{bmatrix}$, $s_1 = \theta_3^2$, $s_2 = \theta_4^2$, and $\rho = \tanh(\theta_5)$. $\mathbf{x} \in \mathbb{R}^8$

3. **Gaussian Mixture**
   - Prior: $\theta \sim \text{Unif}(-10, 10)$, $\theta \in \mathbb{R}^2$.
   - Likelihood: $\mathbf{x} \mid \theta \sim 0.5\mathcal{N}(\mathbf{x} \mid \mathbf{m}_\theta = \theta, \mathbf{S} = \mathbf{I}) + 0.5\mathcal{N}(\mathbf{x} \mid \mathbf{m}_\theta = \theta, \mathbf{S} = 0.01 \cdot \mathbf{I})$, $\mathbf{x} \in \mathcal{R}^2$.
4. **Bernoulli Generalized Linear Model (GLM)**
   - Prior: $\beta \sim \mathcal{N}(0, 2)$, $\mathbf{f} \sim \mathcal{N}\big(0, (\mathbf{F}^\top \mathbf{F})^{-1}\big)$, $\mathbf{F}_{i,i-2} = 1$, $\mathbf{F}_{i,i-1} = -2$, $\mathbf{F}_{i,i} = 1 + \sqrt{\frac{i-1}{9}}$, $\mathbf{F}_{i,j} = 0$ otherwise, $1 \leq i, j \leq 9$.
   - Likelihood: $\mathbf{x} \mid \theta = (\mathbf{x}_1, \ldots, \mathbf{x}_{10})$, $\mathbf{x}_1 = \Sigma_i^T z_i$, $\mathbf{x}_{2:10} = \frac{1}{x_1}\mathbf{V}\mathbf{z}$ $z_i \sim \text{Bern}\big(\eta(\mathbf{v}_i^\top \mathbf{f} + \beta)\big)$, $\eta(\cdot) = \exp(\cdot)/(1 + \exp(\cdot))$.
5. **Two Moons**
   - Prior: $\theta \sim \mathcal{U}(-1, 1)$, $\theta \in \mathbb{R}^2$
   - Likelihood: $\mathbf{x} \mid \theta = \begin{bmatrix} r\cos(\alpha) + 0.25 \\ r\sin(\alpha) \end{bmatrix} + \begin{bmatrix} -|\theta_1 + \theta_2|/\sqrt{2} \\ (-\theta_1 + \theta_2)/\sqrt{2} \end{bmatrix}$, where $\alpha \sim \mathcal{U}(-\pi/2, \pi/2)$, $r \sim \mathcal{N}(0.1, 0.01^2)$, $\mathbf{x} \in \mathbb{R}^2$

Computation The experiments in this study were conducted using a combination of personal and institutional computational resources. Preliminary analyses and prototyping were performed on a MacBook Pro with an Intel Core i7 processor and 16GB of RAM.

For larger-scale experiments, we used high-performance computing resources provided by the institution's research cluster, which includes access to multi-core CPUs with 128GB of RAM. We did not use any GPU for the experiments. While execution time varied by dataset and task, typical runs for clustering and evaluation completed within a few hours.