A generative flow model for conditional sampling via optimal transport

Jason Alfonso
University of the Philippines

Ricardo Baptista*
California Institute of Technology

Anupam Bhakta
Columbia University

Noam Gal
Baruch College

Alfin Hou
University of Edinburgh

Isa Lyubimova
Georgia Institute of Technology

Daniel Pocklington
Grinnell College

Josef Sajonz
University of Michigan

Giulio Trigila*
Baruch College

Ryan Tsai
Yale University

Abstract

Sampling conditional distributions is a fundamental task for Bayesian inference and density estimation. Generative models characterize conditionals by learning a transport map that pushes forward a reference (e.g., a standard Gaussian) to the target distribution. While these approaches can successfully describe many non-Gaussian problems, their performance is often limited by parametric bias and the reliability of gradient-based (adversarial) optimizers to learn the map. This work proposes a non-parametric generative model that adaptively maps reference samples to the target. The model uses block-triangular transport maps, whose components characterize conditionals of the target distribution. These maps arise from solving an optimal transport problem with a weighted L^2 cost function, thereby extending the data-driven approach in Trigila and Tabak [45] for conditional sampling. The proposed approach is demonstrated on a low-dimensional example and a parameter inference problem involving nonlinear ODEs.

1 Introduction

Characterizing the conditional distribution of parameters $X \in \mathbb{R}^d$ in a statistical model given an observation $y^* \in \mathbb{R}^m$ is the fundamental task of computational Bayesian inference. For many statistical models, approximating the posterior $\mu(x|y^*) \propto \mu(y^*|x)\mu(x)$, given the likelihood $\mu(y|x)$ and prior $\mu(x)$, requires sampling approaches, such as Markov-chain Monte Carlo (MCMC) [35]. While MCMC has many consistency guarantees, it is often difficult to produce uncorrelated samples for high-dimensional distributions with multi-modal behavior.

Generative modeling is a popular framework that avoids some of the drawbacks associated with MCMC by making use of transportation of measure [26, 37, 31, 22]. Broadly speaking, this approach finds a transport map T that pushes forward a reference distribution ρ that is easy to sample (e.g., a standard Gaussian) to the target distribution μ, which we denote as $T_\sharp \rho = \mu$. This map is often found by minimizing the KL divergence using the change-of-variables formula, a technique which first appeared in [43], or by minimizing Wasserstein distances as in [3]. After finding a transport map

*This paper results from a project in the Polymath Junior undergraduate research program. The project was mentored by RB and GT who designed the project. All the authors contributed to the code and performed the research. RB and GT wrote the paper. Corresponding authors: rsb@caltech.edu, giulio.trigila@baruch.cuny.edu.

Accepted to the Optimal Transport and Machine Learning Workshop at NeurIPS 2023.

T, one can generate i.i.d. samples in parallel from the target distribution by sampling $z^i \sim \rho$ and evaluating the map at these samples $T(z^i) \sim \mu$, thereby avoiding the use of Markov chain simulation.

In many inference problems, the likelihood model $\mu(y^*|x)$ is computationally expensive or intractable to evaluate (e.g., it involves marginalization over a set of high-dimensional latent variables) or the prior density is unavailable (e.g., it is only prescribed empirically by a collection of images). In these settings, evaluating the posterior density of $X|Y = y^*$ up to a normalizing constant, and hence variational inference, is not possible. Instead, likelihood-free (which is also known as simulation-based) inference [15] aims to sample the posterior distribution given only a collection of samples $(x^i, y^i) \sim \mu(x, y)$ drawn from the joint distribution[2]. To sample conditionals of the joint, [41, 4, 44] consider the class of transport maps with the lower block-triangular[3] structure

$$T(y, x) = \begin{bmatrix} T^{\mathcal{Y}}(y) \\ T^{\mathcal{X}}(y, x) \end{bmatrix}, \tag{1}$$

where $T^{\mathcal{Y}} \colon \mathbb{R}^m \to \mathbb{R}^m$ and $T^{\mathcal{X}} \colon \mathbb{R}^{m+d} \to \mathbb{R}^d$. In particular, Theorem 2.4 in [4] shows that if the reference density has the product form $\rho(y, x) = \mu(y)\rho(x)$ and $T_\sharp \rho(y, x) = \mu(y, x)$, then $T^{\mathcal{X}}(y^*, \cdot)_\sharp \rho(x) = \mu(x|T^{\mathcal{Y}}(y^*))$ for $\mu(y)$-a.e. y^*. Hence, the map $T^{\mathcal{X}}$ can be used to sample any conditional distribution for $X|Y = y^*$. Moreover, one can learn maps of the form in (1) given only samples from the joint distribution [26].

Most approaches for generative modeling (see related work below) find transport maps by imposing a parametric form for T and learning its parameters by the solution of a (possibly adversarial) optimization problem [44, 10]. In addition to the challenges of solving high-dimensional optimization, parametric approaches may introduce bias and can not be easily updated in an online data setting.

Our contribution: We propose a flow that is built from simple elementary maps (T_t) of the block-triangular form in (1) so that their composition $T = T_K \circ \cdots \circ T_2 \circ T_1$ pushes forward the reference $\rho(y, x)$ to the joint target distribution $\mu(y, x)$. In this work, we take a product reference distribution for ρ as in [44] and seek the map from $\rho(y, x) = \mu(y)\mu(x)$ to $\mu(y, x) = \mu(y)\mu(x|y)$. To preserve the marginal distribution for the observations $\mu(y)$, we can take the first map component as $T^{\mathcal{Y}} = \mathrm{Id}(y)$ so that the composition of the second map components $T_t^{\mathcal{X}}(y, \cdot)$ pushes forward the prior distribution $\mu(x)$ to the conditional $\mu(x|y)$ for each y. As compared to parametric approaches that have a fixed model capacity, our algorithm iteratively adapts the number of maps K in the composition to improve the approximation until the push-forward constraint is met.

The remainder of this article is organized as follows. Section 3 contains background on optimal transport (OT) maps. Section 4 shows how to learn a flow composed of block-triangular maps that are optimal for a weighted L^2 cost. Section 5 demonstrates this flow on a Bayesian inference problem.

2 Related work

Monte Carlo methods: A popular family of nonparametric statistical methods for sampling posterior distributions (often with intractable likelihood functions) given only joint samples is approximate Bayesian computation (ABC) [40]. To bypass the evaluation of the likelihood, ABC selects a distance function $d \colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}_+$ (e.g., the L^2 norm) and identifies parameters x^i, , whose synthetic observations $y^i \sim \mu(\cdot|x^i)$ are close to the true observation y^* up to a small tolerance $\epsilon > 0$, i.e., it rejects parameter samples x^i that do not satisfy $d(y^i, y^*) < \epsilon$. While ABC can correctly sample the posterior distribution as $\epsilon \to 0$ [6], the large distances between high-dimensional observations often cause ABC to reject many samples and produce poor approximations [29]. Given that many statistical models are often computationally expensive to simulate, this calls for strategies that don't waste any samples from the joint distribution $\mu(y, x)$.

Conditional generative models: Several generative approaches build maps for conditional sampling by directly seeking maps $T^{\mathcal{X}}$ parameterized by the conditioning variables y. These models include conditional normalizing flows [46, 51, 24], conditional generative adversarial net-

[2] Even if the likelihood and/or prior are intractable, it is often feasible to sample parameters $x^i \sim \mu(x)$ from the prior distribution and synthetic observations $y^i \sim \mu(\cdot|x^i)$ from the likelihood model.

[3] We can equivalently consider an upper-triangular structure with a reverse ordering for $T^{\mathcal{X}}$ and $T^{\mathcal{Y}}$.

works [27, 1, 23], and conditional diffusion models [7, 38]. These approaches all require a parameterization for the map or the score function in the case of diffusion models. A way to overcome a fixed parameterization was proposed in [42] where the first modern version of normalizing flows (NF) appeared. NFs build a map from the target to a Gaussian reference density in a gradual way by composing many elementary maps. Rather than finding the overall map at once, one deals with the more straightforward task of parameterizing simple elementary maps whose composition is supposed to reproduce the overall map. NFs were then popularized in computer vision [34], where the elementary maps were chosen to be a combination of relatively simple neural networks and affine transformations with tractable Jacobians in order to use likelihood-based training methods. Recently, many more choices of NFs have been proposed; see [31, 22] for reviews on this topic and [19, 30] for continuous-time variants. Despite their name, modern NF models select a small number of maps $K = \mathcal{O}(1)$ and jointly learn the composed transformation $T_K \circ \cdots \circ T_1$, thereby making NFs similar to seeking a map with a a fixed parametric capacity, rather than a flow.

Optimal transport: Among all maps that pushforward one measure to another, optimal transport (OT) select maps that minimize an integrated transportation cost of moving mass [48]. In recent years, an immense set of computational tools have been developed to find OT maps [32]. For instance, [16] showed that Sinkhorn's algorithm is an efficient procedure for computing transport plans between two empirical measures. The plan can then be used to estimate an approximate transport map [33]. Alternative approaches directly learn a map that can be evaluated at new inputs (that are not necessarily in the training dataset) by leveraging the analytical structure of the optimal Brenier map for the quadratic cost [8]. In particular, [25] parameterized the map T as the gradient of an input convex neural network [2]. The Brenier map T transports the samples in a single step and can be estimated by solving an adversarial optimization problem given only samples of the reference and target measures. This approach was extended in [44] for conditional sampling by imposing the block-triangular structure in (1) on T, thereby finding the conditional Brenier map [12]. The requirement to solve challenging min-max problems in these approaches, however, has inspired alternative methods to find the (conditional) Brenier map that are more stable in high dimensions [47]. In this work, we propose a flow-based approach based on OT that only requires the solution of minimization problems, such as those appearing in conditional normalizing flows.

3 Background on optimal transport

Given two measures ρ, μ defined on $\mathbb{R}^n$ that have densities[4], the Monge problem seeks a map $T\colon \mathbb{R}^n \to \mathbb{R}^n$ that satisfies $T_\sharp \rho = \mu$ and minimizes an integrated transportation cost given in terms of $c\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$. Here we will only consider strictly convex cost functions c, such as the quadratic cost $c(z, z') = \frac{1}{2}\|z - z'\|^2$. Then, the optimal transport map is the solution to the Monge problem

$$\min_T \left\{ \int c(z, T(z))\rho(z)\mathrm{d}z : T_\sharp \rho = \mu \right\}, \tag{2}$$

over all measurable functions with respect to ρ. To consider measures for which problem (2) does not admit a solution, it is common to work with the relaxation introduced by Kantorovich, which seeks a coupling, or transport plan, $\gamma\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ with marginals ρ and μ. The relaxation solves $\min_{\gamma \in \Pi(\rho,\mu)} \int c(z, z')\gamma(z, z')\mathrm{d}z\mathrm{d}z'$, where $\Pi(\rho, \mu)$ denotes all joint probability distributions that satisfy the constraints $\int \gamma(z, z')\mathrm{d}z' = \rho(z)$, and $\int \gamma(z, z')\mathrm{d}z = \mu(z')$. The Kantorovich problem is the continuous equivalent of a linear program and, as such, it admits a dual formulation that is useful for our purpose. The dual problem consists of solving the maximization problem

$$\max_{\varphi,\psi} \int \varphi(z)\rho(z)\mathrm{d}z + \int \psi(z')\mu(z')\mathrm{d}z', \tag{3}$$

among potential functions $\varphi\colon \mathbb{R}^n \to \mathbb{R}^n$ and $\psi\colon \mathbb{R}^n \to \mathbb{R}^n$ satisfying the constraint $\varphi(z) + \psi(z') \leq c(z, z')$ for all z, z'. It can be shown that the solution (φ, ψ) of (3) is given by the conjugate pair

$$\varphi(z) = \psi^c(z) \coloneqq \min_{z'} \{c(z, z') - \psi(z')\}$$
$$\psi(z') = \varphi^c(z') \coloneqq \min_{z} \{c(z, z') - \varphi(z)\},$$

[4]While this assumption can be relaxed, for ease of exposition we will assume densities exist in this work and denote them using the notation for the corresponding measures.

where f^c denotes the c-transform of f. One of the most important results of the dual Kantorovich problem is that, for sufficiently smooth ρ and μ, the solution of the dual problem is equivalent to the solution of the Monge optimal transport problem; in other words, when ρ and μ are sufficiently regular, the optimal plan γ is induced by a one-to-one map T. Moreover, one can recover the optimal transport map solving (2) from the solution of the dual problem for any cost function of the form $c(z, z') = h(z - z')$ with h strictly convex as

$$T(z) = z - (\nabla h)^{-1} \nabla \varphi(z). \tag{4}$$

We refer the reader to [39, Chapter 1.3] and [17, Chapter 2] for more details on the solution of the dual formulation for general costs.

Inspired by the form of the optimizer, [13, 18] show that the optimal potentials (and thus the optimal map by (4)) can be directly computed by maximizing the objective functional

$$\mathcal{J}(\varphi) = \int \varphi(z)\rho(z)\mathrm{d}z + \int \varphi^c(z')\mu(z')\mathrm{d}z'. \tag{5}$$

Moreover, the authors showed that the first variation, i.e., the functional derivative, of the objective $\mathcal{J}$ for the quadratic cost $h(z) = \frac{1}{2}\|z\|^2$ at φ_t can be explicitly computed as $\frac{\delta \mathcal{J}}{\delta \varphi}|_{\varphi_t} = \rho(z) - \mu(\nabla \varphi_t^{**}) \det \nabla^2 \varphi_t^{**}$, where φ_t^* denotes the convex conjugate of φ_t. This suggests that a natural way to solve (5) is via the gradient ascent iterations

$$\varphi_{t+1}(z) = \varphi_t(z) + \alpha \left. \frac{\delta \mathcal{J}}{\delta \varphi} \right|_{\varphi_t}, \tag{6}$$

where $\alpha > 0$ denotes a step-size parameter. Applying this iteration in practice, however, requires the functional form of the source and target densities as well as evaluating convex conjugates via the solution of separate optimization problems. The next section constructs a flow for which we can more easily evaluate the functional derivatives of the objective functional.

4 Conditional transport via data-driven flows

Given that the optimal map is the gradient of the optimal potential φ, one way to look at the gradient ascent iteration for the potentials is to take the gradient with respect to z on both sides of (6) in order to obtain the discrete-time evolution equation

$$z_{t+1} = z_t - \alpha (\nabla h)^{-1} \nabla_z \left. \frac{\delta \mathcal{J}}{\delta \varphi} \right|_{\varphi_t}, \tag{7}$$

starting from the identity map $z_0 = z$, or equivalently $\varphi_0(z) = \|z\|^2/2$ for the quadratic cost. In the limit of $t \to \infty$, the evolution in (7) defines a map $z_\infty(z)$ pushing forward ρ to μ. The challenge of considering this dynamic for φ is that computing the functional derivative is not straightforward due to the presence of convex conjugates in the definition for $\mathcal{J}$, as in (6).

A crucial observation made in Trigila and Tabak [45] shows that one can substitute (7) with

$$z_{t+1} = z_t - \alpha (\nabla h)^{-1} \nabla_z \left. \frac{\delta \mathcal{J}_t}{\delta \varphi} \right|_{\varphi = \text{const.}} \tag{8}$$

in terms of the time-dependent functional

$$\mathcal{J}_t(\varphi) = \int \varphi(z)\rho_t(z)\mathrm{d}z + \int \varphi^c(z')\mu(z)\mathrm{d}z', \tag{9}$$

where ρ_t is defined as the pushforward of ρ under the map $z_t(z)$. In this case, the functional derivative evaluated at a constant potential φ, that without loss of generality we take to be zero, was shown in [45] to be $\frac{\delta \mathcal{J}_t}{\delta \varphi}|_{\varphi=0} = \rho_t(z) - \mu(z)$. This computation avoids the use of convex conjugates as is in Section 3. A parametric approximation of this functional derivative will be presented in Section 4.2.

As $\alpha \to 0$, the iterations in (8) define a continuous-time flow gradually mapping ρ into μ. The flow evolves according to the dynamic $\dot{z} = -(\nabla h)^{-1} \nabla_z (\rho_t(z) - \mu(z))$, where the density ρ_t for $z(t)$ satisfies the continuity equation $\frac{\partial \rho_t}{\partial t} + \mathrm{div}(\rho_t \dot{z}) = 0$. Section 5 in [45] shows that for strictly convex cost functions, the squared L^2 norm between ρ_t and μ is strictly decreasing, which shows that $\rho_t \to \mu$ in L^2 as $t \to \infty$. An important direction of future work is to establish convergence rates of the flow in (8) under different metrics on probability spaces.

4.1 Block-triangular maps

With the quadratic cost c, the flow in (8) does not yield maps with a block-triangular structure in (1) whose blocks can be used for conditional sampling. To find a block-triangular transport map for $z = (y, x)$, we use a cost function that heavily penalizes mass movements in the y variable while making almost free movements in the x variable. An example is

$$c_\lambda(z, z') = \frac{1}{2}(\lambda\|y - y'\|^2 + \|x - x'\|^2), \tag{10}$$

with large positive λ. In this case, the optimal map in (4) has the form

$$T(z) = z - \nabla_\lambda \varphi(z), \tag{11}$$

where we define the rescaled gradient associated with the cost c_λ as $\nabla_\lambda \varphi(z) = (\partial_y \varphi(z)/\lambda, \partial_x \varphi(z))$. Hence, when $\lambda \to \infty$ the map in (11) converges to a block-triangular map of the form in (1) with $T^{\mathcal{Y}}(y) = \mathrm{Id}(y)$ and $T^{\mathcal{X}}(y, x) = x + \partial_x \varphi(y, x)$.

Remark 1. *The optimal transport map $x \mapsto T^{\mathcal{X}}(y, x)$ pushing forward $\rho(x)$ to $\mu(x|y)$ for each y with minimal quadratic cost $\int \|x - T^{\mathcal{X}}(y, x)\|^2 \rho(y, x) dx dy$ was coined in Carlier et al. [12] as the conditional Brenier map. Theorem 2.3 in Carlier et al. [12] shows that this map is monotone and unique among all functions written as the gradient of a convex potential with respect to the input x.*

Remark 2. *The cost function in (10) is related to the weighted L^2 cost function $\sum_{i=1}^n \lambda_i(\varepsilon)|z_i - z_i'|^2$ for $\lambda_i(\varepsilon) > 0$. For weights satisfying $\lambda_{i+1}(\varepsilon)/\lambda_i(\varepsilon) \to 0$ as $\varepsilon \to 0$ for all $i \in \{1, \dots, d-1\}$, Carlier et al. [11] showed that the optimal transport map with respect to this weighted cost converges to the strictly lower-triangular transport map known as the Knothe-Rosenblatt (KR) rearrangement [21, 36]. The KR map is uniquely defined given a variable ordering. For the purpose of conditional sampling, it is sufficient to consider block-triangular, rather than triangular, maps, as described in Section 1. The drawback is that the larger space of block-triangular maps T admits more transformations satisfying the push-forward condition $T_\sharp \rho = \mu$. The non-uniqueness can be resolved, however, by the regularization from the transport cost; see Remark 1.*

As in the previous section, we now derive a flow where each elementary map has a block-triangular structure of the form in (1). For the rescaled cost, the flow in (8) has the form $z_{t+1} = z_t - \alpha \nabla_\lambda(\rho_t(z_t) - \mu(z_t))$. Each update defines an elementary map T_t pushing forward ρ_t to ρ_{t+1}, which is exactly block-triangular as $\lambda \to \infty$. Moreover, each map is the sum of an identity and a perturbation given by the rescaled gradient of the maximum ascent direction for $\mathcal{J}_t$. We recall that the functional derivative of $\mathcal{J}_t$ is computed only using the current reference measure ρ_t and the target μ. The next section shows how to compute the functional gradient given only samples from ρ_t and μ.

4.2 Gradient approximation from samples

In this work, we follow Trigila and Tabak [45] and approximate the gradient in the span of a small set of features $F_j : \mathbb{R}^n \to \mathbb{R}$ where $n = d + m$ with coefficients $\beta_j \in \mathbb{R}$, i.e.,

$$\left.\frac{\delta \mathcal{J}_t}{\delta \varphi}\right|_{\varphi(z)=0} \approx \sum_j \beta_j F_j(z). \tag{12}$$

The features can include radial basis functions, polynomials, or neural networks; see [20] in the context of GANs. Here we choose F_j to be radial basis functions centered around a subset of random points. More details on the parameterization and center selection strategy is provided in Appendix A.

The approximation in (12) corresponds to the parameterization of a potential $\varphi(z) \approx \varphi_\beta(z) = \sum_j \beta_j F_j(z)$. Given a rich expansion for φ, one can hope to approximate the functional derivative sufficiently well. Nevertheless, a core advantage of the flow is that the elementary map at each step does not need to learn the full map pushing forward ρ_t to μ.

In an empirical setting, our goal is to estimate the potential functions φ_β given only i.i.d. samples $\{z_t^i\}_{i=1}^N \sim \rho_t$ and $\{(z')^i\}_{i=1}^M \sim \mu$. In practice, samples from the initial product reference $\rho_0(y, x) = \mu(y)\mu(x)$ can be generated by creating a tensor product set of the joint samples from $\mu(y, x)$. We use the samples to define a Monte Carlo approximation of the objective functional in (9). That is,

$$\widehat{\mathcal{J}}_t(\varphi_\beta) = \frac{1}{N}\sum_{i=1}^N \varphi_\beta(z_t^i) + \frac{1}{M}\sum_{i=1}^M \varphi_\beta^c((z')^i). \tag{13}$$

In this work, we find the coefficients β_j that maximize the objective in (13) via gradient ascent. To avoid selecting a step size, we choose the coefficients according to the one-step Newton scheme $\beta^* = (\nabla^2_\beta \widehat{\mathcal{J}}_t)^{-1} \nabla_\beta \widehat{\mathcal{J}}_t$, where the first and second derivatives are computed at $\beta = 0$. The Newton method captures the local curvature of $\mathcal{J}_t$ around the identity map and empirically results in faster convergence of the flow towards μ. Appendix B shows that the gradient and Hessian of (13) with respect to β can be easily computed as

$$\nabla_{\beta_j} \widehat{\mathcal{J}}_t \Big|_{\beta=0} = \frac{1}{N} \sum_{i=1}^{N} F_j(z_t^i) - \frac{1}{M} \sum_{i=1}^{M} F_j((z')^i),$$

$$\nabla^2_{\beta_j, \beta_k} \widehat{\mathcal{J}}_t \Big|_{\beta=0} = -\frac{1}{M} \sum_{i=1}^{M} \frac{1}{2} \langle \nabla F_j((z')^i), \nabla_\lambda F_k((z')^i) \rangle.$$

After computing the optimal coefficients β^*, we take the rescaled gradient of φ_{β^*} to obtain a discrete-time update for the parameterized version of (8). Each update defines one elementary map

$$z_{t+1} = T_t(z_t) \coloneqq z_t - \sum_j \beta_j^* \nabla_\lambda F_j(z_t). \tag{14}$$

We remind the reader that for large λ, the elementary map in (14) is mostly acting to update the x component of the z variable by penalizing transport of the y component. In practice, we implement the flow by *only* updating x and setting $y_{t+1} = y_t$.

We propose to update the samples $\{z_t\}_{i=1}^N$ until their values are no longer changing; specifically, we stop the procedure when the L^2 norm for the update of all points in (14) is below the threshold $\epsilon = 10^{-6}$. When the samples stop moving they are approximately equal in distribution to the target samples from μ. The composition of the resulting elementary maps in (14) defines a generative flow model pushing forward ρ to μ. Our complete procedure is provided in Algorithm 1.

Algorithm 1 Generative flow model for conditional sampling

1: **Input**: Joint samples $\{(y^i, x^i)\}_{i=1}^N \sim \mu(y, x)$, features (F_j), termination threshold ϵ
2: Split dataset to create reference $\rho(y, x) = \mu(y)\mu(x)$ and target $\mu(y, x)$ samples
3: Set $t = 0$ and $z_t^i = (y_t^i, x_t^i)$ to reference samples
4: **while** samples are still moving: $\sum_i \|z_{t+1}^i - z_t^i\|^2 > \epsilon$ **do**
5: Find coefficients β^* using one-step of Newton's method
6: Move points using the map in (14)
7: Increment counter $t \leftarrow t + 1$
8: **end while**

The resulting flow defines an overall block-triangular map that can be used to sample any conditional of the target measure. By preserving the block-triangular structure in each component, the composed map after running K steps of Algorithm 1 has the block-triangular form in (1) where the second component is given by $T^{\mathcal{X}}(y^*, x) \coloneqq T_K^{\mathcal{X}}(y^*, \cdot) \circ \cdots \circ T_1^{\mathcal{X}}(y^*, x)$ for any conditioning variable y^*. Theorem 2.4 in [4] shows that the map $x \mapsto T^{\mathcal{X}}(y^*, x)$ pushes forward $\mu(x)$ to $\mu(x|y^*)$. Thus, we can sample any conditional measure after learning the flow by pushing forward (new) prior samples through the composed map with a fixed argument for the y variable.

We conclude this section by presenting the core advantages of our algorithm. First, it does not require selecting a fixed number of elementary maps (i.e., flow layers) *a-priori*, which makes the approach non-parametric as compared to modern NFs [31]. The algorithm above proceeds until the difference between the reference and target measures is small according to the selected features, which can be chosen adaptively at each step. Second, the algorithm only uses minimization steps with respect to the parameters as compared to approaches that solve min-max problems. In fact, the complexity of each iteration is at most $\mathcal{O}(Mdp^2 + p^3)$ to update the coefficients, where p is the number of features, and $\mathcal{O}(Ndp)$ to move the sample points. Third, we don't evaluate the push-forward density through the map T for training, unlike approaches that use the change-of-variables formula to maximize the likelihood of the data. Hence, the algorithm does not require specific parameterizations that guarantee T is invertible and/or $\det \nabla T$ is tractable to evaluate, as in [50, 5]. Lastly, we don't require the functional form of the reference density, which permits us to construct maps that push-forward a general (possibly non-Gaussian) prior measure.

5 Numerical examples

In this section, we illustrate the flow on a two-dimensional example in Section 5.1, and a Bayesian inference problem of inferring four parameters of the Lotka–Volterra nonlinear ODE in Section 5.2.

5.1 2D banana distribution

Here we let the parameter and observation be $X \sim \mathcal{N}(0,1)$ and $Y = 0.5X^2 - 1 + \epsilon$ with $\epsilon \sim \mathcal{N}(0,1)$, respectively. The left panel of Figure 1 shows the un-normalized joint density $\mu(x,y)$ while the middle panel shows samples from $\mu(x,y)$ in red, and the product reference $\mu(x)\mu(y)$ in blue. In this example, we parameterize each elementary map using ten features given by radial basis functions with centers chosen at random points from the empirical measure for $\mu(x)\mu(y)$.

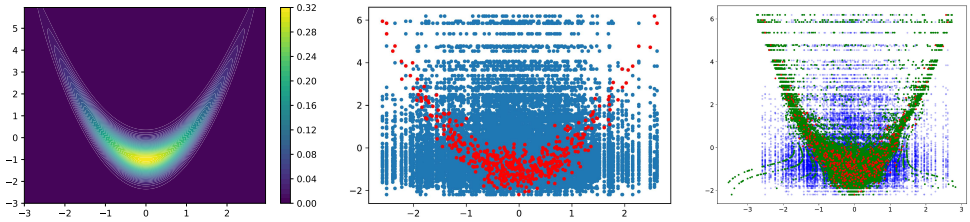

Figure 1: Left: Contours of the joint density $\mu(x,y)$. Middle: 500 i.i.d. samples from $\mu(x,y)$ in red and 10^4 samples from $\mu(x)\mu(y)$. Right: Samples from $\mu(x,y)$ in red, from $\mu(x)\mu(y)$ in blue, and from the pushforward of $\mu(x)\mu(y)$ through the flow in green. As expected, the pushforward samples overlap with the joint samples.

The right panel of Figure 1 plots the samples generated by pushing forward the product reference samples through the composed map T in green. At the end of the algorithm, we observe the push-forward condition $T_\sharp \rho = \mu$ is satisfied with the close match between the green and red samples. By Theorem 2.4 in [4], we can use the learned map to sample the conditional distribution $\mu(x|y^*)$ for any y^*. Figure 2 plots the approximate density (using a kernel density estimator) of 10^4 push-forward samples $T^{\mathcal{X}}(y^*, x^i)$ given $x^i \sim \mu(x)$ for the conditioning variable $y^* = 2$. In comparison to a conditional kernel density estimator from joint samples of $\mu(y,x)$, we observe close agreement between the mapped samples and the true multi-modal conditional density for $\mu(x|y^*)$.

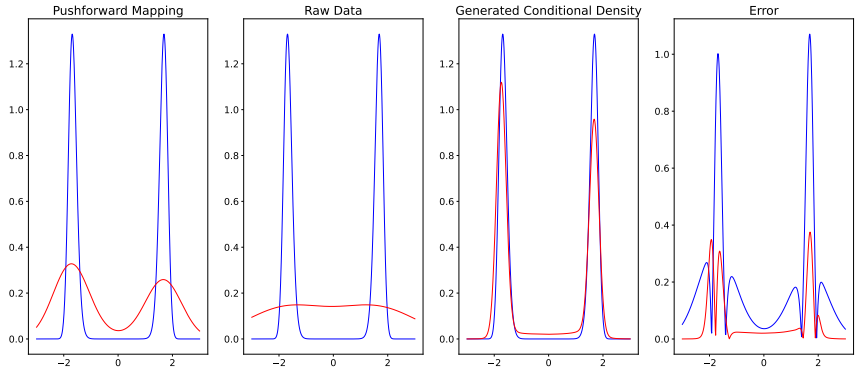

Figure 2: The blue lines in the first three panels represent the true conditional density for $\mu(x|y^* = 2)$. The red lines display a Nadaraya-Watson [28, 49] conditional kernel density estimator (CKDE) of the approximation to $\mu(x|y^* = 2)$ using samples obtained from: the pushforward of $\mu(x)\mu(y)$ (i.e., the green points in Figure 1) in the first panel, the joint distribution (i.e., the red points in Figure 1) in the second panel, and the push-forward of $\mu(x)$ via the learned flow $x \mapsto T(y^* = 2, x)$ in the third panel. The fourth (right) panel compares the error between the true conditional density and the CKDE of the first panel in blue and the CKDE of the third panel in red (i.e., the method proposed in this work). We observe that the flow provides a more accurate conditional approximation than kernel estimators given the same joint samples. In particular, the L_∞ error relative to the three conditional density estimates of the three panels above, from left to right, are 1.07, 1.19, and 0.37, respectively.

5.2 Lotka–Volterra dynamical system

Next, we apply Algorithm 1 to estimate static parameters in the Lotka–Volterra population model given noisy realizations of the states over time. The model describes the populations $p = (p_1, p_2) \in \mathbb{R}^2_+$ of prey and predator species, respectively. The populations $p(t)$ for times $t \in [0, T]$ solve the nonlinear coupled ODEs

$$\begin{aligned} \frac{dp_1(t)}{dt} &= \alpha p_1(t) - \beta p_1(t) p_2(t), \\ \frac{dp_2(t)}{dt} &= -\gamma p_2(t) - \delta p_1(t) p_2(t), \end{aligned} \tag{15}$$

with the initial conditional $p(0) = (30, 1)$, where $X = (\alpha, \beta, \gamma, \delta) \in \mathbb{R}^4$ are unknown parameters. The parameters are initially distributed according to a log-normal prior distribution given by $\log(X) \sim \mathcal{N}(\mu, 0.5 I_4)$ with $\mu = (-0.125, -0.125, -3, -3)$. We simulate the ODE for $T = 20$ time units and observe the state values every $\Delta t_{\text{obs}} = 2$ time units with independent and additive log-normal noise, i.e., $\log(Y_k) \sim \mathcal{N}(p(k\Delta t_{\text{obs}}), \sigma^2 I_2)$ for $k = 1, \ldots, 9$ with $\sigma^2 = 0.1$. Figure 3 displays the two states $p(t)$ for the parameter $x^* = (0.83, 0.041, 1.08, 0.04)$ and an observation $y^* \in \mathbb{R}^{18}$ drawn from the likelihood model $\mu(\cdot|x^*)$ in circles. The main reason for choosing this model is that the likelihood is known in closed form and hence the results can be compared to an MCMC sampling procedure, the gold standard for solving Bayesian inference problems. In this experiment, we learn the flow using $M = 1000$ samples from the joint distribution.

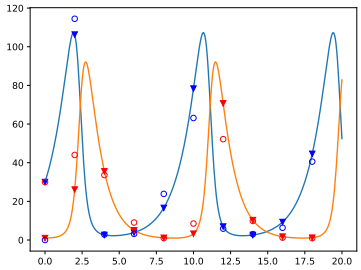

Figure 3: Species populations over time (solid lines) for the true parameter x^*. Circles are observations y^* from the true trajectory; triangles are before adding noise.

Figure 4 compares one and two-dimensional projections of the approximate posterior distribution for the parameters obtained with the flow (left) and with MCMC (right). The red vertical line represents the exact value of the parameters x^* used to generate the trajectory in Figure 3. We observe close agreement between the approximate samples, as well as similar mean squared errors between x^* (the red line) and the posterior means of 0.52 and 0.50 using the flow and MCMC, respectively. Additional numerical results with a comparison of the posterior predictive distributions of both methods is presented in Appendix C.

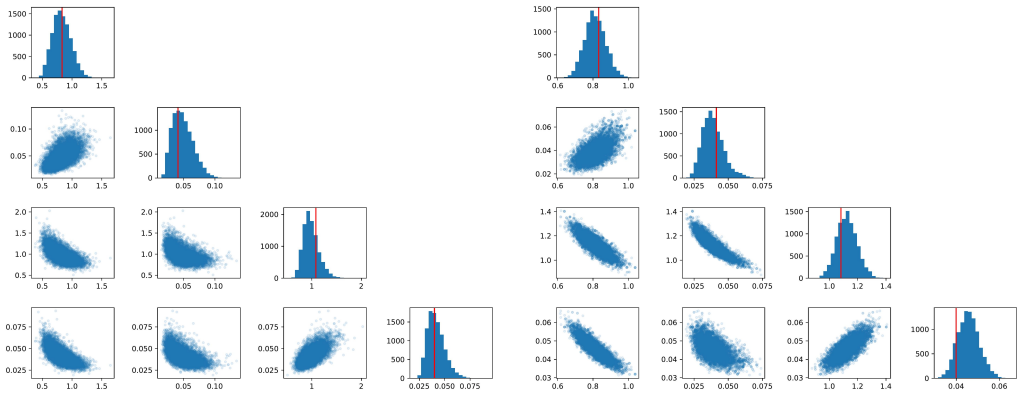

Figure 4: Left: approximate posterior samples generated by mapping 10^4 points from the prior using the map computed from the flow. Right: 10^4 MCMC samples. Both simulations are compared to the true parameters x^* (in red) that generated the observations y^* in Figure 3.

6 Conclusions and future work

This work presents a generative flow model for Bayesian inference where posterior samples are generated by pushing forward prior samples through a composition of maps. Finding the flow is entirely data-driven and it is based on the theory of optimal transport (OT) with a weighted L^2 cost function. This cost yields transport maps with a block-triangular structure, which is suitable for

conditional sampling. As compared to state-of-the-art OT approaches for conditional sampling that solve challenging min-max optimization problems, the flow is is constructed using a sequence of elementary maps that are found using only minimization and gradually push forward the prior to the posterior. Future work includes the possibility of enlarging the map feature space by means of projections into reproducing Kernel Hilbert spaces in a similar spirit to [14], as well as adapting the features to exploit low-dimensional structure between the reference and target distributions as in [9].

Acknowledgments and Disclosure of Funding

This work was done as part of the 2022 Polymath Junior summer undergraduate research program, supported by the NSF REU program under award DMS-2218374. RB gratefully acknowledges support from the US Department of Energy AEOLUS center (award DE-SC0019303), the Air Force Office of Scientific Research MURI on "Machine Learning and Physics-Based Modeling and Simulation" (award FA9550-20-1-0358), and a Department of Defense (DoD) Vannevar Bush Faculty Fellowship (award N00014-22-1-2790).

References

[1] J. Adler and O. Öktem. Deep Bayesian inversion. *arXiv preprint arXiv:1811.05910*, 2018.

[2] B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] R. Baptista, B. Hosseini, N. B. Kovachki, and Y. Marzouk. Conditional sampling with monotone GANs: from generative models to likelihood-free inference. *arXiv preprint arXiv:2006.06755*, 2023.

[5] R. Baptista, Y. Marzouk, and O. Zahm. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, 2023, To Appear.

[6] S. Barber, J. Voss, and M. Webster. The rate of convergence for approximate Bayesian computation. *Electronic Journal of Statistics*, 9(1):80 – 105, 2015.

[7] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

[8] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

[9] M. Brennan, D. Bigoni, O. Zahm, A. Spantini, and Y. Marzouk. Greedy inference with structure-exploiting lazy maps. *Advances in Neural Information Processing Systems*, 33:8330–8342, 2020.

[10] C. Bunne, A. Krause, and M. Cuturi. Supervised training of conditional monge maps. *Advances in Neural Information Processing Systems*, 35:6859–6872, 2022.

[11] G. Carlier, A. Galichon, and F. Santambrogio. From Knothe's transport to Brenier's map and a continuation method for optimal transport. *SIAM Journal on Mathematical Analysis*, 41(6): 2554–2576, 2010.

[12] G. Carlier, V. Chernozhukov, and A. Galichon. Vector quantile regression: An optimal transport approach. *Annals of Statistics*, 44(3):1165–1192, 2016.

[13] R. Chartrand, B. Wohlberg, K. Vixie, and E. Bollt. A gradient descent solution to the Monge-Kantorovich problem. *Applied Mathematical Sciences*, 3(22):1071–1080, 2009.

[14] S. Chewi, T. Le Gouic, C. Lu, T. Maunu, and P. Rigollet. SVGD as a kernelized Wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

[15] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[16] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] A. Figalli and F. Glaudo. *An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows*. EMS Press, 2021. doi: 10.4171/ETB/22.

[18] W. Gangbo. An elementary proof of the polar factorization of vector-valued functions. *Archive for rational mechanics and analysis*, 128:381–399, 1994.

[19] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[20] H. Gu, P. Birmpa, Y. Pantazis, L. Rey-Bellet, and M. A. Katsoulakis. Lipschitz regularized gradient flows and latent generative particles. *arXiv preprint arXiv:2210.17230*, 2022.

[21] H. Knothe. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4 (1):39–52, 1957.

[22] I. Kobyzev, S. J. Prince, and M. A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11): 3964–3979, 2020.

[23] S. Liu, X. Zhou, Y. Jiao, and J. Huang. Wasserstein generative learning of conditional distribution. *arXiv preprint arXiv:2112.10039*, 2021.

[24] J.-M. Lueckmann, G. Bassetto, T. Karaletsos, and J. H. Macke. Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference*, pages 32–53. PMLR, 2019.

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

[26] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification*, pages 1–41. Springer International Publishing, Cham, 2016. ISBN 978-3-319-11259-6. doi: 10.1007/978-3-319-11259-6_23-1.

[27] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[28] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1): 141–142, 1964.

[29] D. J. Nott, V. M.-H. Ong, Y. Fan, and S. Sisson. High-dimensional ABC. In *Handbook of Approximate Bayesian Computation*, pages 211–241. Chapman and Hall/CRC, 2018.

[30] D. Onken, S. W. Fung, X. Li, and L. Ruthotto. OT-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(10), pages 9223–9232, 2021.

[31] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[32] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[33] A.-A. Pooladian and J. Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021.

[34] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[35] C. P. Robert, G. Casella, and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[36] M. Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, 23(3):470–472, 1952.

[37] L. Ruthotto and E. Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, 44(2):e202100008, 2021.

[38] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[40] S. A. Sisson, Y. Fan, and M. Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[41] A. Spantini, R. Baptista, and Y. Marzouk. Coupling techniques for nonlinear ensemble filtering. *SIAM Review*, 64(4):921–953, 2022.

[42] E. G. Tabak and C. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[43] E. G. Tabak and E. Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[44] A. Taghvaei and B. Hosseini. An optimal transport formulation of Bayes' law for nonlinear filtering algorithms. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6608–6613. IEEE, 2022.

[45] G. Trigila and E. G. Tabak. Data-driven optimal transport. *Communications on Pure and Applied Mathematics*, 69(4):613–648, 2016.

[46] B. L. Trippe and R. E. Turner. Conditional density estimation with Bayesian normalising flows. In *Second workshop on Bayesian Deep Learning*, 2017.

[47] T. Uscidda and M. Cuturi. The Monge gap: A regularizer to learn all transport maps. *arXiv preprint arXiv:2302.04953*, 2023.

[48] C. Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[49] G. S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

[50] A. Wehenkel and G. Louppe. Unconstrained monotonic neural networks. *Advances in neural information processing systems*, 32, 2019.

[51] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

A Map parametrization and simulation details

In this section we discuss our parameterization for the elementary potential functions $\varphi_\beta(z) = \sum_j \beta_j F_j(z)$ that are used in the numerical experiments of Section 5.

In this work we selected the features (F_j) to be inverse multiquadric kernels or radially-symmetric kernels of the form

$$F(r) = r \operatorname{erf}\left(\frac{r}{\alpha}\right) + \frac{\alpha e^{-(r/\alpha)^2}}{\sqrt{\pi}} \tag{16}$$

where $\alpha \in \mathbb{R}_{>0}$ is the bandwidth and $r = \|z - z_c\|$ is the radius for some center point $z_c \in \mathbb{R}^n$. This choice aligns with the approach presented in the first modern version of normalizing flows [42], in which the features apply local expansions or contractions of the sample points around the centers z_c.

Once the functional form of the kernels is prescribed, the elementary potential function is completely defined by the choice for the bandwidths α and the centers z_c. In our numerical experiments, we selected the centers uniformly at random from the samples $z_t^i \sim \rho_t$ of the reference distribution and the samples $(z')^i \sim \mu$ of the joint (target) distribution. For problems with high-dimensional parameters and observations, such as the Lotka-Volterra example in Section 5.2, we found that adapting the random sampling for the centers improves the speed of convergence of Algorithm 1. In particular, we selected the observation location y_c of the center points $z_c = (y_c, x_c)$ to be near the particular observation of interest, y^*, more frequently. This choice refines the map pushing forward $\rho_t(y, x)$ to $\mu(y, x)$ around y^*, which is the map used to sample the target conditional $\mu(x|y^*)$.

We chose the bandwidth α for each feature according to the rule of thumb described in [45]. That is,

$$\alpha = \left(n_p \left(\frac{1}{\tilde{\rho}(z_c)} + \frac{1}{\tilde{\mu}(z_c)}\right)\right)^{1/d} \tag{17}$$

where $\tilde{\rho}$ and $\tilde{\mu}$ are kernel density estimates (KDE) of the reference and the target distributions, respectively. The rationale behind (17) is to have kernels with a larger bandwidth where there are fewer samples of the reference and the target distributions, and with a smaller bandwidth that can finely resolve the density in regions of the domain where the distributions are more concentrated. Given that the kernel estimator is only needed to compute the scalar bandwidth, they are not meant to be very accurate and updated at every step of the algorithm. In this work, the target density is time independent and hence its KDE is computed only once at the beginning of the procedure. The KDE of the reference distribution is instead updated after every 200 steps of the algorithm. The scalar $n_p \in \mathbb{R}_{>0}$ is a problem dependent parameter that can be either set to a fixed value (e.g., $n_p = 0.01$ in our experiments) or selected via cross-validation.

To reduce the impact of the specific bandwidth adopted in our procedure, we further multiplied the value of α in (17) by a time dependent constant $m(t) \in \mathbb{R}_{>0}$, which decreases as the algorithm advances. In particular, at the beginning of the experiment, the radius of influence of the kernels (i.e., features that result in local expansions or contractions) is set to be large in order to cover the entire domain containing the samples of $\mu(x)\mu(y)$. As the simulation advances, we gradually decrease the value of $m(t)$ to produce a more localized action of the elementary maps. In our experiments we chose

$$m(t) = 1 + \frac{m_0}{1 + e^{(t - t_{\max})/\sigma}}$$

with the parameters taken to be $m_0 = 10$, and $\sigma = t_{\max}/10$, where $t_{\max}$ is the maximum number of steps we allow the algorithm to complete.

Lastly, while the maps found using Algorithm 1 can be used to sample any conditional distribution, we suggest augmenting the reference samples when one is interested in the conditional distribution corresponding to one realization of the conditioning variable y^*. In particular, we include markers $\{(x^i, y^*)\}_{i=1}^N$ with $x^i \sim \mu(x)$ in the set of reference ρ_0 samples. The push-forward of these additional samples immediately provides samples from the desired conditional distribution $\mu(x|y^*)$.

B Derivation of the Jacobian and Hessian

In this section we derive expressions for the Jacobian and Hessian of the empirical objective functional $\widehat{\mathcal{J}}_t$ in (13) with respect to the parameters (β_j) of the elementary potential function φ_β. To compute these derivatives, we first derive an expression for the c-transform of the parametric potential function appearing inside the objective functional.

Proposition 1. *For the cost function $c_\lambda(z, z') = \frac{1}{2}(\lambda\|y - y'\|^2 + \|x - x'\|^2)$, let $\varphi^c_\beta(z') = \min_z\{c_\lambda(z, z') - \varphi_\beta(z)\}$ be the c-transform of the differentiable function $\varphi_\beta(z)\colon \mathbb{R}^n \to \mathbb{R}$ where $\varphi_\beta(z) = \sum_j \beta_j F_j(z)$. Then, φ^c_β has an second-order asymptotic expansion in β given by*

$$\varphi^c_\beta(z') = -\sum_j \beta_j F_j(z') - \frac{1}{2}\sum_{j,k} \beta_j\beta_k \langle \nabla F_j(z'), \nabla_\lambda F_k(z')\rangle + \mathcal{O}(\|\beta\|^3). \tag{18}$$

Proof. Let $\bar{z}$ be the optimal z that attains the minimum value for the c-transform, i.e., $\varphi^c_\beta(z') = c_\lambda(\bar{z}, z') - \varphi_\beta(\bar{z})$. For a differentiable function φ_β, the optimal value $\bar{z}$ satisfies $\nabla_z c_\lambda(\bar{z}, z') - \nabla_z\varphi_\beta(\bar{z}) = 0$. Thus, for the parametric expansion $\varphi_\beta(z) = \sum_j \beta_j F_j(z)$ we have the condition $\bar{z} = z' + \sum_j \beta_j \nabla_\lambda F_j(\bar{z})$, which defines an implicit function for $\bar{z}$ in terms of z'.

Substituting the expression for $\bar{z}$ in the c-transform gives us

$$\varphi^c_\beta(z') = \frac{1}{2}\left(\lambda\Big\|\sum_j \beta_j \nabla_y F_j(\bar{z})/\lambda\Big\|^2 + \Big\|\sum_j \beta_j \nabla_x F_j(\bar{z})\Big\|^2\right) - \sum_j \beta_j F_j(\bar{z}), \tag{19}$$

where $\bar{z}$ depends on z'. A first-order Taylor series expansion of each feature F_j in the first term of (19) around z' yields the following second-order asymptotic expansion in the coefficients β

$$\begin{aligned}
&\lambda\Big\|\sum_j \beta_j \nabla_y F_j(\bar{z})/\lambda\Big\|^2 + \Big\|\sum_j \beta_j \nabla_x F_j(\bar{z})\Big\|^2 \\
&= \sum_{j,k} \beta_j\beta_k \langle \nabla_y F_j(z'), \nabla_y F_k(z')/\lambda\rangle + \sum_{j,k} \beta_j\beta_k \langle \nabla_x F_j(z'), \nabla_x F_k(z')\rangle + \mathcal{O}(\|\beta\|^3).
\end{aligned}$$

Similarly, a first-order Taylor series expansion of each feature F_j in the second term of (19) around z' yields the asymptotic expansion

$$\sum_j \beta_j F_j(\bar{z}) = \sum_j \beta_j F_j(z') + \sum_{j,k} \beta_j \beta_k \langle \nabla F_j(z'), \nabla_\lambda F_k(z') \rangle + \mathcal{O}(\|\beta\|^3).$$

Substituting these expansions in (19), we arrive at the second-order expansion in (18) after collecting the quadratic terms in β and using the definition of the rescaled gradient. □

Using the result of Proposition 1, a second-order asymptotic expansion in β for the empirical objective functional is given by

$$\begin{aligned}\widehat{\mathcal{J}}_t(\varphi_\beta) =& \frac{1}{N} \sum_{i=1}^{N} \left(\sum_j \beta_j F_j(z_t^i) \right) - \\ & \frac{1}{M} \sum_{i=1}^{M} \left(\sum_j \beta_j F_j((z')^i) + \frac{1}{2} \sum_{j,k} \beta_j \beta_k \langle \nabla F_j((z')^i), \nabla_\lambda F_k((z')^i) \rangle + \mathcal{O}(\|\beta\|^3) \right).\end{aligned}$$

Computing the first and second derivatives of the functional above with respect to each coefficient and evaluating the result at $\beta = 0$ results in the Jacobian and Hessian presented in Section 4.

C Additional numerical results for the Lotka–Volterra model

For the flow and MCMC approximations, we obtain the 30 most significant posterior samples, which are closest to the empirical posterior mean in the Euclidean norm. For each parameter, we find the corresponding state trajectories by solving the ODEs in (15). The states obtained with the flow and with MCMC are compared in the left and right of Figure 5 respectively. The posterior predictive states give a visual representation of the uncertainty arising from estimating the true parameter x^* given noisy observations. As expected, the MCMC method displays lower uncertainty in the trajectories due to its use of the exact likelihood model. This is particularly noticeable for larger values of the populations p_1, p_2 where the effect of the noise on the sampled data (i.e., the difference between the circle and triangle markers in Figure 3) has a larger effect than during time intervals where p_1 and p_2 are nearly constant.

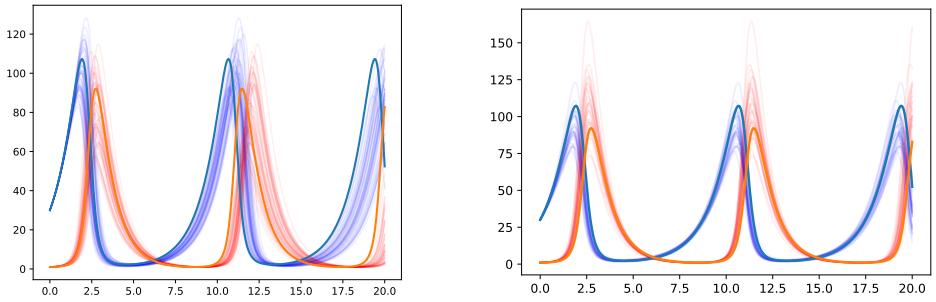

Figure 5: Populations as a function of time obtained by solving the ODE model with the parameter posterior samples displayed in Figure 4 from the flow model (left) and MCMC (right). The parameters were chosen to be the 30 closest values (in the Euclidean norm) to the empirical posterior mean.