Greedy inference with structure-exploiting lazy maps

Michael C. Brennan ∗
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
mcbrenn@mit.edu

Daniele Bigoni ∗
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
dabi@mit.edu

Olivier Zahm
Université Grenoble Alpes, INRIA, CNRS, LJK
38000 Grenoble, France
olivier.zahm@inria.fr

Alessio Spantini
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
alessio.spantini@gmail.com

Youssef Marzouk
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
ymarz@mit.edu

Abstract

We propose a framework for solving high-dimensional Bayesian inference problems using structure-exploiting low-dimensional transport maps or flows. These maps are confined to a low-dimensional subspace (hence, lazy), and the subspace is identified by minimizing an upper bound on the Kullback–Leibler divergence (hence, structured). Our framework provides a principled way of identifying and exploiting low-dimensional structure in an inference problem. It focuses the expressiveness of a transport map along the directions of most significant discrepancy from the posterior, and can be used to build deep compositions of lazy maps, where low-dimensional projections of the parameters are iteratively transformed to match the posterior. We prove weak convergence of the generated sequence of distributions to the posterior, and we demonstrate the benefits of the framework on challenging inference problems in machine learning and differential equations, using inverse autoregressive flows and polynomial maps as examples of the underlying density estimators.

1 Introduction

Inference in the Bayesian setting typically requires the computation of integrals \( \int f \, d\pi \) over an intractable posterior distribution whose density \( \pi \) is known up to a normalizing constant. One approach to this problem is to construct a deterministic nonlinear transformation, i.e., a transport map \( [57] \), that induces a coupling of \( \pi \) with a tractable distribution \( \rho \) (e.g., a standard Gaussian). Formally, we seek a map \( T \) that pushes forward \( \rho \) to \( \pi \), written as \( T_*\rho = \pi \), such that the change of variables \( \int f \, d\pi = \int f \circ T \, d\rho \) makes integration tractable.

Many constructions for such maps have been developed in recent years. Normalizing flows (see \[34\] [42] [46] [54] and references therein) build transport maps via a deep composition of functions

∗These authors contributed equally to this work.

2In this paper, we only consider distributions that are absolutely continuous with respect to the Lebesgue measure on \( \mathbb{R}^d \), and thus will use the notation \( \pi \) to denote both the distribution and its associated density.
parameterized by neural networks, with certain ansatzes to enable efficient computation. Many recently proposed autoregressive flows (for example [17, 20, 27, 31, 43]) compose triangular maps, which allow for efficient evaluation of Jacobian determinants. In general, triangular maps [9, 33, 47] are sufficiently general to couple any absolutely continuous pair of distributions \((\rho, \pi)\), and their numerical approximations have been investigated in [29, 38, 40, 52]. The flow map of a neural ordinary differential equation [13, 21, 23] can also be seen as an infinite-layer limit of a normalizing flow. Alternatively, Stein variational methods [18, 35, 36] provide a nonparametric way of constructing \(T\) as a composition of functions lying in a chosen RKHS.

In general, it can be difficult to represent expressible maps in high dimensions. For example, triangular maps on \(\mathbb{R}^d\) must describe \(d\)-variate functions and thus immediately encounter the curse of dimensionality. Similarly, kernel-based methods lose expressiveness in high dimensions [12, 18]. Flow-based methods often increase expressiveness by adding layers, but this is typically performed in an ad hoc or unstructured way, which also requires tuning.

Here we propose a framework for inference that creates target-informed architectures around any class of transport maps or normalizing flows. In particular, our framework uses rigorous a priori error bounds to discover and exploit low-dimensional structure in a given target distribution. It also provides a methodology for efficiently solving high-dimensional inference problems via greedily constructed compositions of structured low-dimensional maps.

The impact of our approach rests on two observations. First, the coordinate basis in which one expresses a transport map (i.e., \(T(x)\) versus \(UT(x)\), where \(U\) is a rotation on \(\mathbb{R}^d\)) can strongly affect the training behavior and final performance of the method. Our framework identifies an ordered basis that best reveals a certain low-dimensional structure in the problem. Expressing the transport map in this basis focuses the expressiveness of the underlying transport class and allows for principled dimension reduction. This basis is identified by minimizing an upper bound on the Kullback–Leibler (KL) divergence between \(\pi\) and its approximation, which follows from logarithmic Sobolev inequalities (see [59]) relating the KL divergence to gradients of the target density.

Second, in the spirit of normalizing flows, we seek to increase the expressiveness of a transport map using repeated compositions. Rather than specifying the length of the flow before training, we increase the length of the flow sequentially. For each layer, we apply the framework above to a residual distribution that captures the deviation between the target distribution and its current approximation. We prove weak convergence of this greedy approach to the target distribution under reasonable assumptions. This sequential framework enables efficient layer-wise training of high-dimensional maps, which especially helps control the curse of dimensionality in certain transport classes. As we shall demonstrate empirically, the greedy composition approach can further improve accuracy at the end of training, compared to baseline methods.

Since Markov chain Monte Carlo (MCMC) methods are also a workhorse of inference, it is useful to contrast them with the variational methods discussed above. In general, these two classes of methods have different computational patterns. In variational inference, one might spend considerable effort to construct the approximate posterior, but afterwards enjoys cheap access to samples and normalized evaluations of the (approximate) target density. How well the approximation matches the true posterior depends on the expressiveness of the approximation class and on one’s ability to optimize within this class. MCMC, in contrast, requires continual computational effort (even after tuning), but (asymptotically) generates samples from the exact posterior. Yet there is a line of work that uses transport to improve the performance of MCMC methods [26, 44]—such that even if one desires exact samples, constructing a transport map can be beneficial. We will demonstrate this link in our numerical experiments.

**Preliminaries.** We will consider target distributions with densities \(\pi\) on \(\mathbb{R}^d\) that are differentiable almost everywhere and that can be evaluated up to a normalizing constant. Such a target will often be the posterior of a Bayesian inference problem, e.g., \(\pi(x) := p(x|y) \propto L_y(x)p_0(x)\), where \(L_y(x) := p(y|x)\) is the likelihood function and \(p_0\) is the prior. We denote the standard Gaussian density on \(\mathbb{R}^d\) as \(\rho\). We will consider maps \(T : \mathbb{R}^d \to \mathbb{R}^d\) that are diffeomorphisms [2] and with some abuse of

---

[1] In general \(T\) does not need to be a diffeomorphism, but only a particular invertible map; see Appendix B for more details. The distributions we will consider in this paper, however, fulfill the necessary conditions for \(T\) to be differentiable almost everywhere.
notation, we will write the pushforward density of $\rho$ under $T$ as $T_\#\rho(x) := \rho \circ T^{-1}(x) |\nabla T^{-1}(x)|$. We will frequently also use the notion of a pullback distribution or density, written as $T^\#_\pi := (T^{-1})^\# \pi$.

In [2], we show how to build a single map in the low-dimensional “lazy” format described above, and describe the class of posterior distributions that admit such structure. In §3 we develop a greedy algorithm for building deep compositions of lazy maps, which effectively decomposes any inference problem into a series of lower-dimensional problems. §4 presents numerical experiments highlighting the benefits of the lazy framework. While our numerical experiments employ inverse autoregressive flows [31] and polynomial transport maps [29][30] as the underlying transport classes, we emphasize that the lazy framework is applicable to any class of transport.

2 Lazy maps

Given a unitary matrix $U \in \mathbb{R}^{d \times d}$ and an integer $r \leq d$, let $\mathcal{T}_r(U)$ be the set that contains all the maps $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ of the form

$$T(z) = U \left[ \tau(z_1, \ldots, z_r) - z \right] U^\top z_\bot \quad (1)$$

for some diffeomorphism $\tau : \mathbb{R}^r \rightarrow \mathbb{R}^d$. Here $U_r \in \mathbb{R}^{d \times r}$ and $U_\bot \in \mathbb{R}^{d \times (d-r)}$ are the matrices containing respectively the $r$ first and the $d-r$ last columns of $U$, and $z = (z_1, \ldots, z_d)$. Any map $T \in \mathcal{T}_r(U)$ is called a lazy map with rank bounded by $r$, as it is nonlinear only with respect to the first $r$ input variables $z_1, \ldots, z_r$ and the nonlinearity is contained in the low-dimensional subspace range($U_r$). The next proposition gives a characterization of all the densities $T_\# \rho$ when $T \in \mathcal{T}_r(U)$.

**Proposition 1** (Characterization of lazy maps). Let $U \in \mathbb{R}^{d \times d}$ be a unitary matrix and let $r \leq d$. Then for any lazy map $T \in \mathcal{T}_r(U)$, there exists a strictly positive function $f : \mathbb{R}^r \rightarrow \mathbb{R}_{>0}$ such that

$$T_\# \rho(x) = f(U_r^\top x) \rho(x), \quad (2)$$

for all $x \in \mathbb{R}^d$ where $\rho$ is the density of the standard normal distribution. Conversely, any probability density function of the form $f(U_r^\top x) \rho(x)$ admits a representation as in (2) for some $T \in \mathcal{T}_r(U)$.

The proof is given in Appendix A.1. By Proposition 1 any posterior density $\pi(x) \propto \mathcal{L}_y(x) \pi_0(x)$ with standard Gaussian prior $\pi_0 = \rho$ and with likelihood function given by $\mathcal{L}_y(x) \propto f(U_r^\top x)$ can be written exactly as $\pi = T_\# \rho$ for some lazy map $T \in \mathcal{T}_r(U)$. In particular, posteriors of generalized linear models naturally fall into this class; see Appendix A.1 for more details. Following [59], Section 2.1], the solution $T_\star \in \mathcal{T}_r(U)$ to

$$\mathcal{D}_{\text{KL}}(\pi || T_\# \rho) = \min_{T \in \mathcal{T}_r(U)} \mathcal{D}_{\text{KL}}(\pi || T_\# \rho),$$

is such that $T_\# \rho(x) = f^*(U_r^\top x) \rho(x)$, where $f^*$ is the conditional expectation

$$f^*(x_r) = E \left[ \frac{\pi(X)}{\rho(X)} | U_r^\top X = x_r \right]$$

with $X \sim \rho$.

Now that we know the optimal lazy map in $\mathcal{T}_r(U)$, it remains to find a suitable matrix $U$ and rank $r$. In Appendix A.2 we show that

$$\mathcal{D}_{\text{KL}}(\pi || T_\# \rho) = \mathcal{D}_{\text{KL}}(\pi || \rho) - \mathcal{D}_{\text{KL}}((U_r^\top)^\# \pi || \rho_r), \quad (3)$$

where $\rho_r$ is the density of the standard normal distribution on $\mathbb{R}^r$ and $(U_r^\top)^\# \pi$ is the density of $U_r^\top X$ with $X \sim \pi$. Thus, for fixed $r$, minimizing $\mathcal{D}_{\text{KL}}(\pi || T_\# \rho)$ over $U$ is the same as finding the most non-Gaussian marginal $(U_r^\top)^\# \pi$. Such an optimal $U$ can be difficult to find in practice. The next proposition instead gives a computable bound on $\mathcal{D}_{\text{KL}}(\pi || T_\# \rho)$, which we will use to construct a $U$ suitable for our algorithm. The proof is given in Appendix A.3

**Proposition 2.** Let $(\lambda_i, u_i) \in \mathbb{R}_{>0} \times \mathbb{R}^d$ be the $i$-th eigenpair of the eigenvalue problem $H u_i = \lambda_i u_i$ where $H = \int (\nabla \log \frac{\pi}{\rho}) (\nabla \log \frac{\pi}{\rho})^\top d\pi$. Let $U = [u_1, \ldots, u_d] \in \mathbb{R}^{d \times d}$ be the matrix containing the eigenvectors of $H$. Then for any $r \leq d$ we have

$$\mathcal{D}_{\text{KL}}(\pi || T_\# \rho) \leq \frac{1}{2} (\lambda_{r+1} + \ldots + \lambda_d). \quad (4)$$
Algorithm 1 Construction of a lazy map.

1: procedure LAZYMAP(π, ρ, ε, rmax)
2: Compute $H = \int (\nabla \log \frac{π}{ρ}(z)) (\nabla \log \frac{π}{ρ}(z))^\top \, d\pi$
3: Solve the eigenvalue problem $Hu = \lambda_i u_i$
4: Let $r = r_{\max} \land \min\{r \leq d : \frac{1}{2} \sum_{i>r} \lambda_i \leq \varepsilon\}$ and assemble $U = [u_1, \ldots, u_d]$.
5: Find $T$ by solving $\min_{T \in T_r(U)} D_{KL}(T\rho || π)$
6: return lazy map $T$

3 Deeply lazy maps

The restriction $r \leq r_{\max}$ in Algorithm 1 helps control the computational cost of constructing the lazy map, but unless a problem admits sufficient lazy structure, $T_\rho$ may not adequately approximate the posterior. To extend the numerical benefits of the lazy framework to general problems, we consider the “deeply lazy” map $Σ_\ell$, a composition of ℓ lazy maps:

$$Σ_\ell = T_1 \circ \ldots \circ T_\ell, \quad T_k \in T_r(U^k),$$
Algorithm 2 Construction of a deeply lazy map

1: procedure LAYERSOFFLAZYMAPS(π, ρ, ε, r, ℓmax)
2:   Set π0 = π and ℓ = 0
3:   while ℓ ≤ ℓmax and \( \frac{1}{2} \text{Tr}(H_ℓ) \geq ε \) do
4:     ℓ ← ℓ + 1
5:     Compute \( T_ℓ = \text{LAZYMAP}(π_{ℓ−1}, ρ, 0, r) \) \( \triangleright \) Algorithm [I]
6:     Update \( \mathcal{ξ}_ℓ = \mathcal{ξ}_{ℓ−1} ∘ T_ℓ \)
7:     Compute \( π_ℓ = (\mathcal{ξ}_ℓ)^{♯}π \)
8:     Compute \( H_ℓ = \int (\nabla \log \frac{π_ℓ}{ρ})(\nabla \log \frac{π_ℓ}{ρ})^\top dπ_ℓ \)
9:   end while
10: return \( \mathcal{ξ}_ℓ = T_1 ∘ \cdots ∘ T_ℓ \)
11: end procedure

where each \( T_ℓ \) is a lazy map associated with a different unitary matrix \( U^k ∈ \mathbb{R}^{d×d} \). For simplicity we consider the case where each lazy layer \( T_k \) has the same rank \( r \), though it is trivial to allow the ranks to vary from layer to layer. In general, the composition of lazy maps is not itself a lazy map. For example, there exists \( U^1 \neq U^2 \) such that \( \mathcal{ξ}_2 = T_1 ∘ T_2 \) can depend nonlinearly on each input variable and so \( \mathcal{ξ}_2 \) cannot be written as in \( (I) \).

The diagnostic matrix \( H \) allows us to build deeply lazy maps in a greedy way. After \( ℓ − 1 \) iterations, the composition of maps \( \mathcal{ξ}_{ℓ−1} = T_1 ∘ \cdots ∘ T_{ℓ−1} \) has been constructed. We seek a unitary matrix \( U^ℓ ∈ \mathbb{R}^{d×d} \) and a lazy map \( T_ℓ ∈ T_r(U^ℓ) \) such that \( (\mathcal{ξ}_{ℓ−1} ∘ T_ℓ)^{♯}ρ \) best improves over \( (\mathcal{ξ}_{ℓ−1})^{♯}ρ \) as an approximation to the posterior. To this end, we define the residual distribution

\[
π_{ℓ−1} = (\mathcal{ξ}_{ℓ−1})^{♯}π,
\]

i.e., the pullback of \( π \) through the current transport map \( \mathcal{ξ}_{ℓ−1} \). Note that \( D_{\text{KL}}(π \vert \vert (\mathcal{ξ}_{ℓ−1} ∘ T_ℓ)^{♯}ρ) = D_{\text{KL}}(π_{ℓ−1} \Vert (T_ℓ)^{♯}ρ) \). We thus build \( T_ℓ \) using Algorithm [I] replacing the posterior \( π \) by the residual distribution \( π_{ℓ−1} \). We then update the transport map to be \( \mathcal{ξ}_ℓ = \mathcal{ξ}_{ℓ−1} ∘ T_ℓ \) and the residual density \( π_ℓ = (\mathcal{ξ}_ℓ)^{♯}π \).

We note that applying Proposition [2] to \( π_ℓ \) with \( r = 0 \) yields

\[
D_{\text{KL}}(π \vert \vert (\mathcal{ξ}_ℓ)^{♯}ρ) = D_{\text{KL}}(π_ℓ \Vert \rho) \leq \frac{1}{2}(λ_1 + \cdots + λ_d) = \frac{1}{2} \text{Tr}(H_ℓ),
\]

where we define the diagnostic matrix at iteration \( ℓ \) as,

\[
H_ℓ = \int \left( \nabla \log \frac{π_ℓ}{ρ} \right) \left( \nabla \log \frac{π_ℓ}{ρ} \right)^\top dπ_ℓ.
\]

Our framework thus naturally exposes the error bound \( \frac{1}{2} \text{Tr}(H_ℓ) \) on the forward KL divergence, which is of independent interest and applicable to any flow-based method. We refer to this bound as the *trace diagnostic*.

This bound can also be used as a stopping criterion for the greedy algorithm; one can continue adding layers until the bound falls below some desired threshold. This construction is summarized in Algorithm [I] and details on its numerical implementation are given in Appendix [F].

The next proposition gives a sufficient condition on \( U^ℓ \) to guarantee the convergence of our greedy algorithm. The proof is given in Appendix [A.4]

**Proposition 3.** Let \( U^1, U^2, \ldots \) be a sequence of unitary matrices. For any \( ℓ ≥ 1 \), we let \( T_ℓ ∈ T_r(U^ℓ) \) be a lazy map that minimizes \( D_{\text{KL}}(π_{ℓ−1} ∥ (T_ℓ)^{♯}ρ) \), where \( π_{ℓ−1} = (T_1 ∘ \cdots ∘ T_{ℓ−1})^{♯}π \). If there exists \( 0 < t ≤ 1 \) such that for any \( ℓ ≥ 1 \)

\[
D_{\text{KL}}((U^ℓ_r)^{♯}π_{ℓ−1} ∥ ρ_r) ≥ t \sup_{U ∈ \mathbb{R}^{d×d} \text{ s.t. } UU^\top = I_d} D_{\text{KL}}((U^ℓ_r)^{♯}π_{ℓ−1} ∥ ρ_r),
\]

then \( (T_1 ∘ \cdots ∘ T_ℓ)^{♯}ρ \) converges weakly to \( π \).
We present numerical demonstrations of the lazy framework as follows. We first illustrate Algorithm 2 (Algorithms 1 and 2) in several challenging inference problems. We consider Bayesian logistic regression and a Bayesian neural network, and compare the performance of a baseline transport map to lazy maps using the same underlying transport class. We measure performance improvements to lazy maps using the same underlying transport class. We measure performance improvements in four ways: (1) the final ELBO achieved by the transport maps after training; (2 and 3): the final trace diagnostics \( \frac{1}{2} \text{Tr}(H^T_t) \) and \( \frac{1}{2} \text{Tr}(H_t) \), which bound the error \( D_{KL}(\pi||\langle \Sigma_t \rangle r \rho) \); and (4) the variance diagnostic \( \frac{1}{2} V(\log \rho/\langle \Sigma_t \rangle r \rho) \), which is an asymptotic approximation of \( D_{KL}(\langle \Sigma_t \rangle r \rho||\pi) \) as \( \langle \Sigma_t \rangle r \rho \to \pi \) (see [40]). Finally, we highlight the advantages of greedily training lazy maps in a nonlinear problem defined by a high-dimensional elliptic partial differential equation (PDE), often used for testing high-dimensional inference methods [4, 16, 53]. Here, the lazy framework is needed to make variational inference tractable by controlling the total number of map parameters. We also illustrate the utility of such flows in preconditioning Markov chain Monte Carlo (MCMC) samplers [20, 44], or equivalently as a way of de-biasing the variational approximation on these three problems.

Numerical examples are implemented in both the TransportMaps framework [7] and using the TensorFlow probability library [19]. The PDE considered in [4, 44] is discretized and solved using the FEniCS [37] and dolfin-adjoint [22] packages.

Let us comment on the condition (5). Recall that the unitary matrix \( U \) that maximizes \( D_{KL}(\langle U^T \rangle r \pi_{l-1}||\rho_r) \) is optimal; see (3). By (5), the case \( t = 1 \) means that \( U^t \) is optimal at each iteration. This corresponds to an ideal greedy algorithm. The case \( 0 < t < 1 \) allows suboptimal choices for \( U^t \) without losing the convergence property of the algorithm. Such a greedy algorithm converges even with a potentially crude selection of \( U^t \) that corresponds to a \( t \) close to zero. This also is why an approximation to \( H_\ell \) is expected to be sufficient; see Section 4. We emphasize that condition (5) must apply simultaneously to all layers for a given \( 0 < t < 1 \). Following [55], one could relax this condition by replacing \( t \) with a sequence \( (t_\ell) \) that goes to zero sufficiently slowly. This development is left for future work. Finally, note that Proposition 5 does not require any constraints on \( r \), so we have convergence even with \( r = 1 \), where each layer only acts on a single direction at a time.
4.1 Illustrative toy example

We first apply the algorithm on the standard problem of approximating the rotated banana distribution $Q \pi_{X_1, X_2}$ defined by $X_1 \sim N(0, 0.5 \times 1)$ and $X_2 | X_1 \sim N(X_1^2, 0.2)$, and where $Q$ is a random rotation. We restrict ourselves to using a composition of rank-1 lazy maps. We consider degree 3 polynomial maps as the underlying transport class. We use Gauss quadrature rules of order 10 for the discretization of the KL divergence and the approximation of $H_B^m$ ($m = 121$ in Algorithm 3 and 5). Figure 1b shows the target distribution $\pi := \pi_{X_1, X_2}$. Figure 1a shows the convergence of the algorithm both in terms of the trace diagnostic $\frac{1}{2} \text{Tr}(H_B^m)$ and in terms of the variance diagnostic. After two iterations the algorithm has explored all directions of $\mathbb{R}^2$, leading to a fast improvement. The convergence stagnates once the trade-off between the complexity of the underlying transport class and the accuracy of the quadrature has been saturated. Figures 1c-1e show the progressive Gaussianization of the residual distributions $\Sigma^{-1}_m \pi$ for different iterations $\ell$.

4.2 Bayesian logistic regression

Next we consider a full rank Bayesian logistic regression problem using 605 observations. Here we compare a baseline IAF; $U$-IAF defined as before; and a 3-layer lazy map trained via the greedy Algorithm 2 denoted G3-IAF. In G3-IAF, each layer has rank $r = 200$. Results are summarized in Table 1 and again we see improvements in each of the performance metrics compared to the baseline IAF. Recall that the basis $U$ relates to a bound on the inclusive KL direction, while the objective function for map training within a layer optimizes the exclusive KL direction. Empirically we see benefits in metrics relating to both directions. Interestingly, we observe that $U$-IAF achieves the greatest ELBO while G3-IAF achieves the lowest trace diagnostics. This suggests that using a larger number of lazy layers tends to lead to improvements to the inclusive KL divergence. Also, though we chose to use the same number of training iterations in each case, we observe that training of the lazy maps converges more quickly; see Appendix G.1 for addition details.

Figure 2: Leading eigenvalues of the diagnostic matrices $H_B^m$ for the G3-IAF map applied to the full rank logistic regression problem. The spectrum flattens and falls as the approximation to the posterior improves.
We consider the problem of estimating the diffusion coefficient \( \kappa \) of an elliptic PDE from sparse observations of the field \( u(x) \) solving
\[
\begin{cases}
\nabla \cdot (e^{\kappa(x)} \nabla u(x)) = 0, & \text{for } x \in \mathcal{D} := [0, 1]^2, \\
u(x) = 0 \text{ for } x_1 = 0, & u(x) = 1 \text{ for } x_1 = 1, \quad \frac{\partial u(x)}{\partial n} = 0 \text{ for } x_2 \in \{0, 1\}.
\end{cases}
\]

As discussed in the introduction, a powerful use case for transport maps is the ability to precondition an MCMC method as described in [26, 43, 44], i.e., using the computed map to improve the posterior geometry. Applying Hamiltonian Monte Carlo [41] to the full rank Bayesian logistic regression problem (in particular, sampling the pullback \( \mathbb{Z}_T^* \pi \) where \( \mathbb{Z}_T \) is the learned U-IAF map), we achieve worst, best, and average component-wise effective sample sizes of 0.39\%, 1.8\%, and 0.99\%, compared to 0.056\%, 0.12\%, and 0.065\% without a transport map (sampling the target \( \pi \) directly). Note that applying \( \mathbb{Z}_T \) to MCMC samples from the pullback yields asymptotically exact samples from \( \pi \). Three leapfrog steps were used in the HMC proposal, and the step sizes were chosen adaptively during the burn-in period of the chains to obtain acceptance rates between 70\% and 90\% [3, 5, 6].

### 4.3 Bayesian neural network

We now consider a Bayesian neural network, also in [18, 36], trained on the UCI yacht hydrodynamics data set [2]. Our inference problem is 581-dimensional, given a network input dimension of 6, one hidden layer of dimension 20, and an output layer of dimension 1. We use sigmoid activations in the input and hidden layer, and a linear output layer. Model parameters are endowed with independent Gaussian priors with zero mean and variance 100. Further details are in Appendix G.2.

Here we consider affine maps as the underlying class of transport. This yields Gaussian approximations to the posterior distribution in both the lazy and baseline cases. We compare a baseline affine map and G3-affine, denoting a 3-layer lazy map where each layer has rank \( r = 200 \). The diagnostic matrices \( H^B \) are computed using 581 standard normal samples. We note improvements in each of the performance metrics using the lazy framework, summarized in Table 1. We also note a 64\% decrease in the number of trained flow parameters in G3-affine, relative to the baseline case (from 338142 to 120600).

Similarly to §4.2, we compare the performance of HMC applied with and without transport map preconditioning. We achieve worst, best, and average component-wise ESS of 0.073\%, 1.2\%, and 0.56\% using the learned G3-Affine map, compared to 0.047\%, 0.14\%, and 0.06\% without a transport map. Here five leapfrog steps were used in the HMC proposal, and the step sizes in each case were picked adaptively as before.

### 4.4 High-dimensional elliptic PDE inverse problem

We consider the problem of estimating the diffusion coefficient \( e^{\kappa(x)} \) of an elliptic PDE from sparse observations of the field \( u(x) \) solving

\[
\begin{cases}
\nabla \cdot (e^{\kappa(x)} \nabla u(x)) = 0, & \text{for } x \in \mathcal{D} := [0, 1]^2, \\
u(x) = 0 \text{ for } x_1 = 0, & u(x) = 1 \text{ for } x_1 = 1, \quad \frac{\partial u(x)}{\partial n} = 0 \text{ for } x_2 \in \{0, 1\}.
\end{cases}
\]
We have presented a framework for creating target-informed architectures for transport-based variational inference. Our approach uses a rigorous error bound to identify low-dimensional structure in the target distribution and focus the expressiveness of the transport map or flow on an important subspace. We also introduce and analyze a greedy algorithm for building deep compositions of low-dimensional maps that can iteratively approximate general high-dimensional target distributions. Empirically, these methods improve the accuracy of inference, accelerate training, and control the complexity of flows to improve tractability. Ongoing work will consider constructive tests for further varieties of underlying structure in inference problems, and their implications on the structure of flows.

5 Conclusions
Broader Impact

Who may benefit from this research? We believe users and developers of approximate inference methods will benefit from our work. Our framework works as an “outer wrapper” that can improve the effectiveness of any flow-based variational inference method by guiding its structure. We hope to make expressive flow-based variational inference more tractable, efficient, and broadly applicable, particularly in high dimensions, by developing automated tests for low-dimensional structure and flexible ways to exploit it. The trace diagnostic developed in our work rigorously assesses the quality of transport/flow-based inference, and may be of independent interest.

Who may be put at disadvantage from this research? We don’t believe anyone is put at disadvantage due to this research.

What are the consequences of failure of the system? We specifically point out that one contribution of this work is identifying when a poor posterior approximation has occurred. A potential failure mode of our framework would be inaccurate estimation of the diagnostic matrix $H$ or its spectrum, suggesting that the approximate posterior is more accurate than it truly is. However, computing the eigenvalues or trace of a symmetric matrix, even one estimated from samples, is a well studied problem. And numerical software guards against poor eigenvalue estimation or at least warns if this occurs. We believe the theoretical underpinnings of this work make it robust to undetected failure.

Does the task/method leverage biases in the data? We don’t believe our method leverages data bias. As a method for variational inference, our goal is to accurately approximate a posterior distribution. It is very possible to encode biases for/against a particular result in a Bayesian inference problem, but that occurs at the level of modeling (choosing the prior, defining the likelihood) and collecting data, not at the level of approximating the posterior.

Acknowledgments and Disclosure of Funding

This work was supported in part by the US Department of Energy, Office of Advanced Scientific Computing Research, AEOLUS (Advances in Experimental Design, Optimal Control, and Learning for Uncertain Complex Systems) project. The authors also gratefully acknowledge support from the Inria associate team UNQUESTIONABLE.

References


