SIAM/ASA J. Uncertainty Quantification
Vol. 12, No. 2, pp. 549–578

Differential Equation–Constrained Optimization with Stochasticity*

Qin Li†, Li Wang‡, and Yunan Yang§

Abstract. Most inverse problems from physical sciences are formulated as PDE-constrained optimization problems. This involves identifying unknown parameters in equations by optimizing the model to generate PDE solutions that closely match measured data. The formulation is powerful and widely used in many science and engineering fields. However, one crucial assumption is that the unknown parameter must be deterministic. In reality, however, many problems are stochastic in nature, and the unknown parameter is random. The challenge then becomes recovering the full distribution of this unknown random parameter. It is a much more complex task. In this paper, we examine this problem in a general setting. In particular, we conceptualize the PDE solver as a push-forward map that pushes the parameter distribution to the generated data distribution. In this way, the SDE-constrained optimization translates to minimizing the distance between the generated distribution and the measurement distribution. We then formulate a gradient flow equation to seek the ground-truth parameter probability distribution. This opens up a new paradigm for extending many techniques in PDE-constrained optimization to optimization for systems with stochasticity.

Key words. inverse problem, constrained optimization, Wasserstein gradient flow, particle method, push-forward map

MSC codes. 65M32, 49Q22, 65M75, 65K10

DOI. 10.1137/23M1571162

1. Introduction.

We study the problem of inferring the random parameters in a differential equation. In particular, we ask,

How do we recover the distribution of an unknown random parameter in a differential equation from that of measurements?

The problem comes from the fact that many differential equations are equipped with parameters that are random in nature. Even with a fixed set of boundary/initial conditions and measuring operators, the measurements are nevertheless random, with the randomness coming from different realizations of the parameter. Aligned with other inverse problems that

*Received by the editors May 8, 2023; accepted for publication (in revised form) March 4, 2024; published electronically June 7, 2024.

https://doi.org/10.1137/23M1571162

Funding: The first author was partially supported by ONR-N00014-21-1-214 and NSF-1750488. The second author was partially supported by NSF grant DMS-1846854. The third author was partially supported by ONR-N00014-24-1-2088 and also received support from Dr. Max Rössler, the Walter Haefner Foundation, and the ETH Zürich Foundation.

†Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53706 USA (qinli@math.wisc.edu).

‡Department of Mathematics, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA (wang8818@umn.edu).

§Department of Mathematics, Cornell University, Ithaca, NY 14853 USA (yunan.yang@cornell.edu).

infer unknown parameters from measurements, we now aim to recover the distribution of the parameters from the distribution of measurements. Therefore, the problem under study is the stochastic extension of the deterministic inverse problems, particularly the PDE-constrained optimization.

In the deterministic and finite-dimensional setting, many problems are formulated as

$$y = \mathcal{G}(u)\,, \quad \mathcal{G}: \mathcal{A} \subset \mathbb{R}^m \to \mathcal{R} \subset \mathbb{R}^n\,. \tag{1.1}$$

This is to feed a system, described by $\mathcal{G}$, with an m-dimensional parameter u, to produce the data $y \in \mathbb{R}^n$. The set $\mathcal{A}$ stands for all the admissible parameters, and $\mathcal{R}$ is the range of $\mathcal{G}$. The inverse problem is to revert the process. The model $\mathcal{G}$ is still given, but the goal is to infer the parameter u using the measured data y^*. Potentially, y^* contains measurement error.

Inverse problems present many challenges. To determine u uniquely from y^*, one requires $\mathcal{G}$ to be injective and y^* to lie within the range of $\mathcal{G}$. However, as elegantly summarized in [29], this is not typically the case in practice. Therefore, various techniques have been introduced to approximate the inference, and optimization is one of the most popular numerical strategies. Specifically, the goal is to find the configuration of u that best matches the data y^*. Denoting D as the metric or divergence applied to the data range $\mathcal{R}$, we search for u^* such that

$$u^* \in \operatorname{argmin} D(y^*, \mathcal{G}(u)) =: L(u)\,. \tag{1.2}$$

Although this minimization formulation solves the original inverse problem when y^* falls within the range of $\mathcal{G}$, it leaves the problem of "invertibility" unresolved. This is because this formulation may have many, or even infinitely many, minimum points. Furthermore, this formulation introduces an additional layer of difficulty regarding "achievability": even if the landscape of the loss function guarantees a unique minimizer, conventional optimization strategies may not be able to find it. Nonetheless, among the many optimization algorithms available, it is common to use gradient descent (GD) or other first-order optimization methods to search for the optimizer if a good initial guess is provided. In the continuous setting, the iteration number in GD is transformed to the time variable s, and the corresponding gradient flow is written as follows:

$$\frac{\mathrm{d}u}{\mathrm{d}s} = -\alpha \nabla_u L\,, \tag{1.3}$$

where α signifies the rate and can be adjusted according to the user's preferences. When u is a function, $\nabla_u L$ denotes the functional derivative of L with respect to u.

Many real-world problems can be formulated using (1.2), such as PDE-constrained optimization problems. In such problems, $\mathcal{G}$ is the map induced by the underlying PDE, which maps the PDE parameter to the measurements. For instance, consider f_i as the solution to a PDE characterized by the PDE operator $\mathcal{L}(u)$ that is parameterized by the coefficient u with the ith source term S_i, and let $\mathcal{M}_j$ denote its jth measurement operator, $1 \le i \le I$, $1 \le j \le J$. Then, the measurement is an $I \times J$ matrix with its ij-h entry as $(\mathcal{G}(u))_{ij} = \mathcal{M}_j(f_i)$. In this case, (1.2) can be naturally represented as a PDE-constrained optimization problem:

$$u^* \in \operatorname{argmin} \frac{1}{IJ} \sum_{ij} |\mathcal{M}_j(f_i) - d_{ij}|^2 \quad \text{subject to} \quad \mathcal{L}(u)[f_i] = S_i\,. \tag{1.4}$$

As with the more general formulation (1.2), the PDE-constrained optimization problem (1.4) may not always be solvable. However, if we attempt to solve it using GD or its flow formulation (1.3), the updating formula necessitates the computation of the gradient, which is typically done by solving both the forward and the adjoint equations: $\mathcal{L}(u)[f_i] = S_i$ and $\mathcal{L}^*(u)[g_j] = \psi_j$, with ψ_j determined by choice of the objective function. Over the years, this problem has attracted significant interest from various scientific communities, and many aspects of gradient computation have been investigated; see the book [14].

Building upon this framework, we are interested in a class of problems where the unknown parameter u is stochastic. Since u is known to be random a priori, we aim to infer its distribution, denoted by ρ_u. Due to the inherent randomness in u, the measurement is also random. By using (1.1), we can write the distribution of data by regarding $\mathcal{G}$ as a push-forward map:

$$\rho_y = \mathcal{G}_\sharp \rho_u \,.$$

Consequently, the inverse problem is to infer the ground-truth distribution of u, denoted by ρ_u^*, using the measured data distribution, denoted by ρ_y^*. Denoting D as the metric or divergence used to measure data discrepancy in the space of probability measures, we aim to find ρ_u^* such that its push-forward matches ρ_y^* as closely as possible. Similar to (1.4), $\mathcal{G}$ can be induced by a PDE, which leads to the following problem of *differential equation–constrained optimization with stochasticity*:

$$\rho_u^* \in \operatorname{argmin} D(\mathcal{G}_\sharp \rho_u, \rho_y^*) \,. \tag{1.5}$$

As with the PDE-constrained optimization, whether the optimizer is unique and whether a simple gradient-based method can achieve the optimizer are unknown. The answers to these questions are rather problem-specific, and this goal is beyond the scope of a single paper. Instead, we strive to make the following contributions:

1. We will provide a recipe for a first-order gradient-based solver for (1.5). Using different metrics to define the "gradient" for probability measures and different definitions of the distance/divergence function D, we can generate the corresponding gradient flows on this metric to move ρ_u along.
2. When D is the Kullback–Leibler (KL) divergence, and the underlying metric for the probability space is the 2-Wasserstein distance, we will provide a particle method to simulate the gradient flow equation. The updating formula for the particle is a combination of a forward and an adjoint solver pair.
3. In the linear case, we study the well-posedness in both underdetermined and overdetermined scenarios and draw a relation to their counterparts in the deterministic setting.

Our formulation appears to be closely related to several well-established research topics, which we will discuss in section 2, including their connections and differences. This will be followed by a gradient flow formulation using different metrics for various choices of distance or divergence D in (1.5). In section 3, we also discuss the associated particle method, which solves the Wasserstein gradient flow for the KL distance. Section 4 presents the available theoretical results when the underlying map $\mathcal{G}$ is linear. We have discovered that the theoretical results

correspond one-to-one to the over-/underdetermined linear system under the deterministic scenario. Finally, numerical evidence is presented in section 5 to support our findings, followed by the conclusion in section 6.

2. Comparisons with other subjects. Several other research fields in applied mathematics are closely related to the problem we propose to study in the introduction. The Bayesian inverse problem [29], for instance, is a field that uses probabilistic models to infer unknown parameters of a system from observed data. Another related field is density estimation [27] that focuses on estimating the probability density function of a random variable from a set of observations. In the following subsections, we will conduct a comprehensive review of these related fields, discuss how they are related to our problem, and point out some key differences between them and our proposed approach.

2.1. Bayesian inverse problem. We now draw the connection to the Bayesian inverse problem [21, 29] and the associated sampling problem [12, 7, 35, 16]. Again starting from (1.1), it considers the scenario where the measurement data is corrupted by noise. Most commonly, the observed data y is assumed to be the sum of the true data and a random noise term:

$$y = \mathcal{G}(u) + \eta\,, \quad u \in \mathcal{A} \subset \mathbb{R}^m\,,$$

where the additive noise is assumed to be Gaussian, i.e., $\eta \sim \mathcal{N}(0,\Gamma)$. Additionally, a prior distribution $\rho^{\text{prior}}(u)$ for the parameter u is given, often assumed to be Gaussian as well. Then the goal is to find the most efficient way to sample from the posterior given by Bayes' theorem:

$$\rho^{\text{post}}(u) \propto \mathbb{P}(y|u)\rho^{\text{prior}}(u)\,, \tag{2.1}$$

where $\mathbb{P}(y|u)$ is the likelihood function, whose concrete form is determined by the noise assumption for the data y. Since $\eta \sim \mathcal{N}(0,\Gamma)$, the likelihood function is given by

$$\mathbb{P}(y|u) \propto e^{-\frac{1}{2}|y-\mathcal{G}(u)|_\Gamma^2}\,.$$

Although the ultimate objective in both the Bayesian inverse problem and our framework (1.5) is to identify the distribution of the parameter u, the sources of randomness in our formulation are *fundamentally different* from those in the Bayesian framework. Specifically, in the Bayesian framework, uncertainty arises due to noise in the measurement process of obtaining y while the true data $\mathcal{G}(u)$ is assumed to be deterministic. This means that if the measurement is entirely precise and $\mathcal{G}$ is invertible, the posterior distribution would be a delta measure, meaning that u would be deterministic.

In contrast, in our case, the true data is a probability distribution by itself. Under a deterministic forward map $\mathcal{G}$, the randomness of the parameter is an inherent feature of the system being modeled, and even the precise "reading" y would nevertheless lead to a data distribution.

With the prior $\rho_0 = \mathcal{N}(m_0,\Sigma_0)$ and the noise assumption on the data, the posterior distribution is of the form $\rho(u) := \frac{1}{Z}e^{-V(u)}$ with $V(u) = \frac{1}{2}|y-\mathcal{G}(u)|_\Gamma^2 + \frac{1}{2}|u-m_0|_{\Sigma_0}^2$. One way to sample from the posterior (2.1) is to use the Langevin dynamics [24], where a set of particles $\{u^{(j)}\}$ are evolved by

$$\mathrm{d}u^{(j)}(t) = -\nabla V(u^{(j)})\mathrm{d}t + \sqrt{2}\mathrm{d}W(t)\,, \quad u^{(j)}(0) \sim \rho^{\text{prior}}(u)\,.$$

In the infinite time horizon, i.e., when $t \to \infty$, $\{u^{(j)}(t)\}$ will be samples from the posterior $\rho^{\text{post}}(u)$.

The Bayesian inverse problem also bares a variational formulation. The posterior $\rho(u)$ can be characterized as a distribution that minimizes the KL divergence [36], i.e.,

$$\rho^{\text{post}}(u) \in \arg\min_{\pi(u)} KL\left(\pi(u)|\mathbb{P}(y|u)\rho^{\text{prior}}(u)\right). \tag{2.2}$$

Therefore, $\rho^{\text{post}}(u)$ can be obtained by evolving the Wasserstein gradient flow of the divergence in (2.2) to equilibrium. The Wasserstein metric and its corresponding energy function, such as the KL divergence, can take different forms in the formulation [30]. It is apparent from both Bayesian formulation (2.2) and our formulation (1.5) that, although both attempt to match the target distribution, the target distribution and source distribution in each approach differ.

2.2. Density estimation. Another related field to our formulation is density estimation. Density estimation is a statistical technique used to estimate the probability density function of a random variable from observed samples [27, 26, 33]. It also finds great use in artificial intelligence such as generative modeling [1, 28, 17, 6]. One common approach to density estimation is to use kernel density estimation [8], which involves convolving a set of basis functions (typically kernels) with the observed samples to estimate the underlying density function. The choice of kernel function and bandwidth parameter can affect the accuracy and smoothness of the resulting estimate. Other methods for density estimation include histogram-based methods, parametric models (e.g., using a normal distribution), flow-matching type methods, and nonparametric models (e.g., using a mixture of distributions).

Density estimation is mostly studied for a parameterized distribution where the goal is to estimate the parameters of a specific distribution rather than the entire density function. This is commonly done using maximum likelihood estimation, a variational approach [34]. Given a set of observations $y_1, y_2, \ldots, y_N$, and a true but unknown probability density function $p(y)$, we seek to estimate the parameter θ in the density function $q(y;\theta)$ that minimizes the KL divergence between $q(y;\theta)$ and $p(y)$. That is,

$$\begin{aligned}\theta^* &= \arg\min_\theta \text{KL}\left(p(y)|q(y;\theta)\right) = \arg\min_\theta \mathbb{E}_{p(y)}\left[\log p(y) - \log q(y;\theta)\right] \\ &= \arg\max_\theta \mathbb{E}_{p(y)}\left[\log q(y)\right] \approx \arg\max_\theta \frac{1}{N}\sum_{i=1}^{N} \log q(y_i;\theta),\end{aligned}$$

where the last two terms correspond to the so-called maximum log-likelihood estimation.

In our framework (1.5), when the push-forward map $\mathcal{G} = I$, the identity map, then the problem reduces to a classical density estimation problem. Otherwise, we are performing a density estimation for u not from samples of u, but samples of $\rho_y = \mathcal{G}\sharp\rho_u$ where $u \sim \rho_u$. The problem in (1.5) combines both the density estimation aspect (from samples of y to the density of y) and the inversion part (from the density of y to the density of u).

3. Gradient flow formulation and particle methods. Denote $\mathcal{P}_2(\mathcal{A})$ the collection of all probability measures supported on $\mathcal{A}$, the admissible set, with finite second-order moments. We endow $\mathcal{P}_2(\mathcal{A})$ with a metric d_g. Being confined to the set $\mathcal{P}_2(\mathcal{A})$ equipped with a metric d_g,

the variational formulation (1.5) for a differential equation–constrained optimization problem becomes

$$\rho_u^* = \underset{\rho_u \in (\mathcal{P}_2(\mathcal{A}), d_g)}{\arg\min} \; E(\rho_u) := D(\rho_y, \rho_y^*) = D(\mathcal{G}_\sharp \rho_u, \rho_y^*)\,. \tag{3.1}$$

This is to find the optimal distribution $\rho_u^* \in \mathcal{P}_2(\mathcal{A})$, which, upon being pushed forward by $\mathcal{G}$, is the closest to the probability distribution of the data: ρ_y^*, measured by the data discrepancy D. The map $\mathcal{G}$ is then given by the deterministic forward operator that maps a fixed parameter configuration to the measurement.

Similar to the fact that the gradient flow (1.3) is used to solve the deterministic optimization problem (1.2), one can run GD type algorithms on the space of probability measures to solve a variational problem defined over the probability space such as (3.1). Since such an optimization problem is over an infinite-dimensional space, gradient-based algorithms are particularly attractive in terms of computational cost. Using gradient-based algorithms for (3.1) amounts to updating ρ_u based on the gradient direction of $E(\rho_u)$. The precise definition of the "gradient" here relies on the metric d_g that the underlying probability space $\mathcal{P}_2(\mathcal{A})$ is equipped with.

We want to address the fact that different pairs of (d_g, E) yield different gradient flow formulations to update ρ_u. We will focus on a few concrete examples in the following subsections.

3.1. The Wasserstein gradient flow strategy. First, we consider d_g as the quadratic Wasserstein metric (W_2). We first define W_2 using the Kantorovich formulation of the optimal transportation problem.

Definition 3.1 (Kantorovich formulation). *Let (M, d) be a metric space. The quadratic Wasserstein metric between two probability measures μ and ν defined on M with finite second-order moments is*

$$W_2(\mu,\nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{M\times M} \mathrm{d}(x,y)^2 \mathrm{d}\gamma(x,y) \right)^{1/2},$$

where $\Gamma(\mu,\nu)$ is the set of all coupling for μ and ν. A coupling γ is a joint probability measure on $M \times M$ whose marginal distributions are μ and ν, respectively. That is,

$$\int_M \gamma(x,y)\,\mathrm{d}y = \mu(x)\,, \quad \int_M \gamma(x,y)\,\mathrm{d}x = \nu(y)\,.$$

Throughout our paper, we set $\mathcal{A} \subset M = \mathbb{R}^m$ with d being the Euclidean distance. Equipped with the 2-Wasserstein metric, the Wasserstein gradient flow equation for $E(\rho_u)$ writes [2, 25] as follows:

$$\partial_t \rho_u = -\nabla_{W_2} E(\rho_u) = \nabla_u \cdot \left(\rho_u \nabla_u \frac{\delta E}{\delta \rho_u} \right). \tag{3.2}$$

This gradient flow is guaranteed to descend the energy. To see this, we multiply $\frac{\delta E}{\delta \rho_u}$ on both sides:

$$\frac{\mathrm{d}}{\mathrm{d}t} E(\rho_u) = \int \partial_t \rho_u \frac{\delta E}{\delta \rho_u} \mathrm{d}u = \int \nabla_u \cdot \rho_u \nabla_u \left(\frac{\delta E}{\delta \rho_u} \right) \frac{\delta E}{\delta \rho_u} \mathrm{d}u = -\int \rho_u(u) \left| \nabla_u \frac{\delta E}{\delta \rho_u} \right|^2 \mathrm{d}u \le 0\,.$$

Immediately, we see from the fact that the right-hand side is negative, $E(\rho_u)$ decays in time, which implies that the equilibrium ρ_u^∞ should satisfy

$$\nabla_u \frac{\delta E}{\delta \rho_u^\infty} = 0 \quad \text{on the support of } \rho_u^\infty . \tag{3.3}$$

For the given specific form of $E(\rho_u)$ in (3.1), we find an explicit formulation for $\frac{\delta E}{\delta \rho_u}$. This can be done through the standard technique from the calculus of variation. According to the definition of the Fréchet derivative, we perturb ρ_u by $\delta\rho_u$ where $\int \delta\rho_u \mathrm{d}u = 0$. Then we have

$$\lim_{\|\delta\rho_u\|_2 \to 0} [E(\rho_u + \delta\rho_u) - E(\rho_u)] = \int \frac{\delta E}{\delta \rho_u} \delta\rho_u \mathrm{d}u . \tag{3.4}$$

Since $y = \mathcal{G}(u)$, it follows that

$$\rho_y = \mathcal{G}_\sharp \rho_u \quad \text{and} \quad \delta\rho_y = \mathcal{G}_\sharp \delta\rho_u .$$

Substituting the definition $E(\rho_u) = D(\mathcal{G}_\sharp \rho_u, \rho_y^*)$, we find

$$\begin{aligned} &E(\rho_u + \delta\rho_u) - E(\rho_u) \\ &\quad = D(\rho_y + \delta\rho_y, \rho_y^*) - D(\rho_y, \rho_y^*) \\ &\quad = \int \frac{\delta D}{\delta \rho_y}(y) \delta\rho_y(y) \mathrm{d}y + \text{higher order terms} \\ &\quad = \int \frac{\delta D}{\delta \rho_y}(\mathcal{G}(u)) \delta\rho_u(u) \mathrm{d}u + \text{higher order terms}, \end{aligned} \tag{3.5}$$

where the last equality uses the definition of a push-forward map. That is, if $f = T_\sharp g$ by $y = T(x)$, then for any measurable function F and Ω in the support of f,

$$\int_{x \in T^{-1}(\Omega)} F(T(x)) g(x) \mathrm{d}x = \int_{y \in \Omega} F(y) f(y) \mathrm{d}y.$$

Comparing (3.4) with (3.5), we have

$$\frac{\delta E}{\delta \rho_u}(u) = \frac{\delta D}{\delta \rho_y} \circ \mathcal{G}(u) . \tag{3.6}$$

This means the Fréchet derivative of E on ρ_u is that of D on ρ_y composing with the push-forward map $\mathcal{G}$. As a concrete example, we consider D in (3.1) to be the KL divergence, namely,

$$D(\rho_y, \rho_y^*) = \int \rho_y \log \frac{\rho_y}{\rho_y^*} \mathrm{d}y \quad \text{and} \quad \frac{\delta D}{\delta \rho_y} = \log \rho_y - \log \rho_y^* + 1 . \tag{3.7}$$

Combining this with (3.2) and (3.6), we finalize the evolution equation:

$$\partial_t \rho_u = \nabla_u \cdot \left(\rho_u \nabla_u \left(\log \frac{\rho_y}{\rho_y^*} (\mathcal{G}(u)) \right) \right) . \tag{3.8}$$

Throughout the paper, it is the convention to use the quadratic Wasserstein metric (i.e., $d_g = W_2$) to equip $\mathcal{P}_2(\mathcal{A})$ if not specifically mentioned otherwise. Similarly, $D = \mathrm{KL}$ is used as the convention to define the cost functional $E(\rho_u)$.

3.2. Other possible metrics. The probability space can be metricized in various ways, and there exist several data misfit functions D to serve as the data discrepancy, used in place of the KL divergence. In this subsection, we provide a few examples to demonstrate the breadth and versatility of the proposed framework.

Wasserstein gradient flow of the Wasserstein-based objective function. One choice is to set D based on the Wasserstein metric with the cost function c while setting $d_g = W_2$. More precisely, let

$$D(\rho_y, \rho_y^*) := \inf_{\gamma \in \Gamma(\rho_y, \rho_y^*)} \int_{M\times M} \mathsf{c}(x,y)\mathrm{d}\gamma\,, \tag{3.9}$$

where $\Gamma(\rho_y, \rho_y^*)$ is the set of all coupling of ρ_y and ρ_y^*, and c is a cost function, e.g., $\mathsf{c}(x,y) = |x-y|^2$. Then by the Kantorovich duality [31], it has the following equivalent form:

$$D(\rho_y, \rho_y^*) = \sup_{(\phi,\psi)\in\Phi(\rho_y,\rho_y^*)} \int_M \phi(x)\rho_y(x)\mathrm{d}x + \int_M \psi(y)\rho_y^*(y)\mathrm{d}y\,, \tag{3.10}$$

where $\Phi(\rho_y, \rho_y^*)$ is the set of pairs (ϕ,ψ) such that $\phi(x)+\psi(y) \le \mathsf{c}(x,y)$ for all $(x,y)\in M\times M$. If we denote by (ϕ^*,ψ^*) the maximizing pairs, also referred to as the Kantorovich potentials, then we have [31]

$$\frac{\delta D}{\delta \rho_y} = \phi^*(x)\,. \tag{3.11}$$

Following (3.2) and (3.6), the corresponding Wasserstein gradient flow is

$$\partial_t \rho_u = \nabla_u \cdot \left(\rho_u \nabla_u \phi^*\left(\mathcal{G}(u)\right)\right)\,. \tag{3.12}$$

Hellinger gradient flow of the χ^2 divergence. While the Wasserstein metric is typically used for its simplicity in bridging particle systems and the underlying flows, one can also equip $\mathcal{P}_2(\mathcal{A})$ with other metrics (i.e., d_g in (3.1)) over the probability space. Another example is the Hellinger distance defined below.

Definition 3.2 (the Hellinger distance). *Consider two probability measures P and Q both defined on a measure space M that are absolutely continuous with respect to an auxiliary measure μ, i.e.,*

$$P(\mathrm{d}x) = p(x)\mu(\mathrm{d}x), \quad Q(\mathrm{d}x) = q(x)\mu(\mathrm{d}x)\,.$$

The Hellinger distance between P and Q is $H^2(P,Q) = \frac{1}{2}\int_M (\sqrt{p(x)} - \sqrt{q(x)})^2\mu(\mathrm{d}x)$.

Following [19], we consider the gradient flow when D is the chi-squared (χ^2) divergence and $E(\rho_u)$ is determined correspondingly. More specifically, we have

$$D(\rho_y, \rho_y^*) = \chi^2(\mathcal{G}_\sharp \rho_u, \rho_y^*) = \int \frac{(\mathcal{G}_\sharp \rho_u)^2}{\rho_y^*} \mathrm{d}y - 1\,. \tag{3.13}$$

Then the gradient flow of (3.13) with respect to the Hellinger distance can be derived via the so-called JKO scheme [15]. That is,

$$\partial_t \rho_u = \lim_{\varepsilon \to 0} \frac{\rho_u^\varepsilon - \rho_u}{\varepsilon}\,, \tag{3.14}$$

where $\rho_u^\varepsilon \in \arg\min_{\int \tilde{\rho}_u \mathrm{d}u = 1} \left\{ D(\mathcal{G}_\sharp \tilde{\rho}_u, \rho_y^*) + \frac{1}{2\varepsilon} H^2(\tilde{\rho}_u, \rho_u) \right\}$. Then the optimality condition yields the following relation:

$$\rho_u^\varepsilon = \rho_u - 4\varepsilon \rho_u \left[\frac{2\rho_y}{\rho_y^*}(\mathcal{G}(u)) - \lambda \right], \tag{3.15}$$

where λ is the Lagrangian multiplier to make sure that ρ_u^ε integrates to one. Hence, $\lambda = \int \frac{2\rho_y}{\rho_y^*}(\mathcal{G}(u))\rho_u \mathrm{d}u$. Plugging (3.15) into (3.14), we get

$$\partial_t \rho_u = 8\rho_u \left[\int \frac{\rho_y}{\rho_y^*}(\mathcal{G}(u))\rho_u \mathrm{d}u - \frac{\rho_y}{\rho_y^*}(\mathcal{G}(u)) \right].$$

Kernelized Wasserstein gradient flow. The so-called Stein variational gradient descent (SVGD) can be seen as the kernelized Wasserstein gradient flow of the KL divergence [20]. Through an equivalent reformulation of the same dynamics, SVGD can also be regarded as the kernelized Wasserstein gradient flow of the χ^2 divergence but with a different kernel [9]. In [11], SVGD is formulated as the true gradient flow of the KL divergence under the newly defined Stein geometry. That is, d_g is a metric based on the Stein geometry, while E is decided by setting D as the KL divergence in (3.1).

3.3. Particle method. One significant advantage of using the gradient flow formulation (3.2) with the underlying metric being W_2 is the ease of translating the produced PDE formulation, such as (3.2) and (3.8), to a particle formulation. This feature allows an easy implementation of numerical schemes. According to (3.8), each independent and identically distributed (i.i.d.) particle drawn from ρ_u should evolve by descending along its negative velocity field:

$$\frac{\mathrm{d}}{\mathrm{d}t} u(t) = -\nabla_u \left. \frac{\delta E}{\delta \rho_u} \right|_{\rho_u(t)} (u(t)) = -\nabla_u \mathcal{G}^\top|_{u(t)} \xi(y(u(t)))\,,$$

where we used the definition of

$$\xi(y(u(t))) := \nabla_y \left. \frac{\delta D}{\delta \rho_y} \right|_{\mathcal{G}_\sharp \rho_u(t)} (y(u(t)))\,. \tag{3.16}$$

When KL divergence is used, the expression for ξ can be made more explicit, and thus

$$\frac{\mathrm{d}}{\mathrm{d}t} u = -\nabla_u \left(\log \frac{\rho_y}{\rho_y^*}(\mathcal{G}(u)) \right) = -\left. \nabla_u \mathcal{G}^\top \right|_{u(t)} \nabla_y \log\left(\rho_y / \rho_y^*\right)(y(t))\,, \text{ where } y(t) = \mathcal{G}(u(t))\,, \tag{3.17}$$

where $\nabla_u \mathcal{G}|_{u_j(t)} \in \mathbb{R}^{n\times m}$ is a Jacobian matrix of $\mathcal{G}$. Multiplying the Jacobian $\frac{\mathrm{d}y}{\mathrm{d}u} = \nabla_u \mathcal{G}$ on both sides, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} y = -\nabla_u \mathcal{G}|_{u(t)} \nabla_u \mathcal{G}|_{u(t)}^{\top} \nabla_y \log\left(\rho_y/\rho_y^*\right), \quad \text{where} \quad y \sim \rho_y. \tag{3.18}$$

Note that in practice, these two formulas cannot be executed because ρ_u and ρ_y are unknown. Therefore, one needs to replace them with their numerical approximation. To be more precise, let u_i be a list of particles drawn from ρ_u; then we write ρ_u^N as an empirical distribution approximating ρ_u, i.e.,

$$\rho_u \approx \rho_u^N = \frac{1}{N}\sum_{j=1}^N \delta_{u_j}, \quad \text{and thus} \quad \rho_y^N = \mathcal{G}_\sharp \rho_u^N = \frac{1}{N}\sum_{j=1}^N \mathcal{G}_\sharp \delta_{u_j} = \frac{1}{N}\sum_{j=1}^N \delta_{y_j}. \tag{3.19}$$

In this ensemble version, all the particles u_j evolve according to

$$\begin{aligned}
\partial_t u_j &= -\nabla_u \left.\frac{\delta E}{\delta \rho_u}\right|_{\rho_u^N}(u_j) = -\nabla_u\left[\log \rho_y^N(\mathcal{G}(u_j)) - \log \rho_y^*(\mathcal{G}(u_j))\right] \\
&= \frac{\nabla_u \rho_y^*(\mathcal{G}(u_j))}{\rho_y^*(\mathcal{G}(u_j))} - \frac{\nabla_u \rho_y^N(\mathcal{G}(u_j))}{\rho_y^N(\mathcal{G}(u_j))} \\
&= \nabla_u \mathcal{G}^{\top}|_{u_j} \underbrace{\left(\frac{\nabla_y \rho_y^*(y_j)}{\rho_y^*(y_j)} - \frac{\nabla_y \rho_y^N(y_j)}{\rho_y^N(y_j)}\right)}_{\xi_j = \nabla_y \left.\frac{\delta D}{\delta \rho_y}\right|_{\rho_y^N}(y_j)},
\end{aligned} \tag{3.20}$$

where $y_j = \mathcal{G}(u_j) \in \mathbb{R}^n$ and they both evolve in time.

It is easy to see that when ρ_y is the ensemble distribution ρ_y^N defined in (3.19), ξ_j is not well defined. In particular, the singularity induced by the Dirac deltas can be numerically inaccessible. In simulations, one has to approximate ρ_y^N using probability density functions to implement (3.20). One option is to use kernel density estimation with an isotropic Gaussian kernel [8, 4]. That is,

$$\rho_y^N \approx \frac{1}{N}\sum_{j=1}^N \phi^\epsilon(y - y_j), \quad \phi^\epsilon(y) = \frac{1}{(2\pi\epsilon)^{n/2}} \exp\left(-\frac{y^2}{2\epsilon}\right).$$

This formulation leads to

$$\frac{\nabla_y \rho_y^N}{\rho_y^N}(y) = -\frac{1}{\epsilon}\frac{\sum_j (y - y_j) e^{-(y-y_j)^2/2\epsilon}}{\sum_j e^{-(y-y_j)^2/2\epsilon}} \quad \forall y,$$

and when plugged into (3.20), it gives the final particle method:

$$\partial_t u_j = \left(\nabla_u \mathcal{G}|_{u_j}\right)^{\top} \xi_j, \quad \text{with} \quad \xi_j = \frac{\nabla_y \rho_y^*(y_j)}{\rho_y^*(y_j)} + \frac{1}{\epsilon}\frac{\sum_i (y_j - y_i) e^{-(y_j - y_i)^2/2\epsilon}}{\sum_i e^{-(y_j - y_i)^2/2\epsilon}}. \tag{3.21}$$

If the observed data ρ_y^* is also an empirical distribution, we can apply kernel density estimation to obtain an approximated reference density.

3.4. Adjoint solver simplification. When $\mathcal{G}$ is explicitly given, $\nabla_u \mathcal{G}$ in (3.21) is rather immediate. However, in many situations, $\mathcal{G}$ is generated by the underlying differential equations, and the explicit calculation of $\mathcal{G}$ relies on PDE solvers. Computing the associated gradient would be even more complicated. In particular, recall $\mathcal{G}$ maps $\mathbb{R}^m$ to $\mathbb{R}^n$. The gradient is stored in a Jacobian matrix of size $n \times m$. As such, the preparation of the entire matrix directly calls for mn partial derivatives computations, each of which, in turn, calls for a PDE solver. The associated computational cost is prohibitive.

To simplify the computation, we note that in (3.21), the Jacobian matrix is applied to a vector as the source term for updating $u_j(t)$. So instead of preparing for the whole Jacobian matrix and then multiplying it on a vector, one can directly compute the matrix-vector product on the equation level based on the adjoint approach [23, sect. 3.3], as summarized below.

For a fixed parameter u, instead of dealing with the explicit map $y = \mathcal{G}(u)$, we consider the following implicit relation between u and y that encodes the PDE information:

$$g(y, u) = 0. \tag{3.22}$$

Here g is the PDE operator that maps u from parameter space and y from PDE solution space to the right-hand side of the PDE. We assume g is Fréchet differentiable in both arguments. Based on the first-order variation of (3.22) in both u and y, we have

$$\nabla_y g \, \nabla_u y + \nabla_u g = 0 \quad \Rightarrow \quad \nabla_y g \, \nabla_u \mathcal{G} = -\nabla_u g \, .$$

To compute $\nabla_u \mathcal{G}^\top \xi$, we can set

$$\nabla_y g^\top \lambda = \xi, \tag{3.23}$$

which immediately translates to

$$\nabla_u \mathcal{G}^\top \xi = -\nabla_u g^\top \lambda \, . \tag{3.24}$$

We should note that in most PDE settings, g maps the function space of y and u to a function space, and the notation of $\nabla_y g^\top \lambda = \xi$ really means the inner product taken on the function space of g and its dual, where λ is chosen from. This typically translates to the adjoint PDE solver. Such adjoint solution then gets integrated in (3.24) for the final functional gradient.

An illustrative example is to consider the acoustic wave equation on a spatial domain Ω and time interval $[0, T]$,

$$u(x)\partial_{tt} y(x,t) - \Delta y(x,t) = s(x,t), \quad y(x, t=0) = \partial_t y(x, t=0) = 0 \, , \tag{3.25}$$

where $y(x,t)$ is the wave equation solution, $s(x,t)$ is the given source term, and $u(x)$ is the squared slowness representing the medium property for the wave propagation. Without loss of generality, we consider $\Omega = \mathbb{R}^d$. For a fixed s, the solution y is determined by u, but in general, we do not have an explicit formulation for the forward map $\mathcal{G}$ such that $y = \mathcal{G}(u)$. Instead, the implicit relation (3.25) is used. We can write (3.25) as

$$g(u, y) = 0, \quad \text{where} \quad g(u, y) = u \, \partial_{tt} y - \Delta y - s \, .$$

Algorithm 3.1 Particle method for (3.8) with $g(u,y)=0$, where $y=\mathcal{G}(u)$ is an implicit forward map.

Input: $\rho_y^*(y)$, initial guess $\{u_j^0\}_{j=1}^N$, and step size Δt.
for Iteration $n=0,1,2,\dots,N_{\max}$ **do**
1. Set $y_j^n=\mathcal{G}(u_j^n)$, $j=1,\dots,N$.
2. Compute ξ_j according to (3.21).
3. Solve for λ_j from the adjoint equation (3.23) for every ξ_j.
4. Update u_j according to (3.24).

end for
Output: Final particle locations $\{u_j^{N_{\max}}\}_{j=0}^N$.

To compute the linear action of the adjoint Jacobian, $\xi\mapsto\nabla_u\mathcal{G}^\top\xi$, we need to solve the adjoint equation. To this end, we first easily obtain $\nabla_u g=\partial_{tt}y$. To compute $\nabla_y g^\top\lambda=\xi$, we realize that

$$\nabla_y g^\top\lambda=\nabla_y\langle u\partial_{tt}y-\Delta y-s\,,\lambda\rangle_{x,t}=\nabla_y\langle y\,,\partial_{tt}(u\lambda)-\Delta\lambda\rangle_{x,t}=u\partial_{tt}\lambda-\Delta\lambda\,,$$

and λ satisfies the zero final-time condition, i.e., $\lambda(x,t=T)=\partial_t\lambda(x,t=T)=0$. Here, $\langle\cdot,\cdot\rangle_{xt}$ represents integration in both x and t domains, and we perform two levels of integration by parts leading to the equation of

$$u(x)\partial_{tt}\lambda-\Delta\lambda=\xi\,,\quad\text{with zero final condition}\,. \tag{3.26}$$

Assembling the functional derivative according to (3.24), we have

$$\nabla_u\mathcal{G}^\top\xi=-\int_0^T\int_{\mathbb{R}^d}\partial_{tt}y(x,t)\lambda(x,t)\mathrm{d}x\mathrm{d}t\,.$$

Note that, as usual, y solves the forward wave equation, and λ solves the adjoint equation (3.26). For this particular problem, the wave equation PDE operator is self-adjoint, making the forward and adjoint equations have the same form, except for the different source terms and the boundary conditions in time.

Returning to (3.21), to update the values for u_j, one can first compute

$$\xi_j(t)=\nabla_y\left.\frac{\delta D}{\delta\rho_y}\right|_{\rho_y^N}(y_j(t))$$

and solve the adjoint equation (3.23) with respect to λ before finally assembling the directional derivative in (3.24). The algorithm is summarized in Algorithm 3.1.

3.5. Discussion on the particle method. In this subsection, we will examine the similarities and differences between the particle method and other similar systems, highlighting the shared features as well as the distinguishing characteristics.

The first system we compare (3.17) with is the Langevin dynamics, the continuous version of the classical Langevin Monte Carlo (LMC) algorithm [10] developed to perform Bayesian sampling:

$$\mathrm{d}y_t = -\nabla_y f(y_t)\mathrm{d}t + \mathrm{d}W_t\,,$$

where W_t denotes the Wiener process. It is a convention to denote by $\pi(t,y)$ the associated probability distribution along time t, and apply the Itô calculus to obtain the following Fokker–Planck equation:

$$\partial_t \pi - \nabla_y \cdot (\nabla_y f \pi + \nabla_y \pi) = 0\,,$$

which can be viewed as the Wasserstein gradient flow of energy E

$$\partial_t \pi - \nabla_y \cdot \left(\pi \nabla_y \frac{\delta E}{\delta \pi}\right) = 0\,, \quad \text{with} \quad E(\pi) = \mathrm{KL}(\pi|\pi^*)\,, \tag{3.27}$$

where $\pi^* \propto e^{-f}$ is the target distribution. This shows that one interpretation of the Fokker–Planck PDE is that the evolution represents a first-order descending scheme that pushes π to the target π^* in the long time horizon when the distance/divergence and the metric d_g are set to be KL and W_2, respectively. Algorithmically, this means that LMC is the first-order method to draw samples from a target distribution, justifying LMC's validity (asymptotic under some conditions).

While the original LMC requires samples from the Wiener process by adding Gaussian random variables in the updating formula, the corresponding gradient flow equation (3.27) admits a much more straightforward particle method. Namely, we directly let the particles descend in the negative gradient direction:

$$\frac{\mathrm{d}}{\mathrm{d}t} y = -\nabla_y \log\left(\pi/\pi^*\right)\,, \quad \text{where} \quad y \sim \pi\,. \tag{3.28}$$

As with (3.17), this method is not immediately feasible due to the lack of explicit form of π, so in practice, we set $\pi \sim \pi^N = \frac{1}{N}\sum_i \delta_{y_i}$ as the ensemble distribution and obtain the following updating formula for the particle method:

$$\frac{\mathrm{d}}{\mathrm{d}t} y_i = -\left(\nabla \log \pi^N(y_i) - \nabla \log \pi^*(y_i)\right)\,.$$

The same issue on the singularity of computing $\nabla \pi^N$ arises, and the blob method was constructed to mitigate such difficulties; see [5] for a reference.

We should note the strong similarity between the above formulation (3.28) and (3.18). The main difference between our approach and LMC sampling is that, in our framework, the sampler's motion is presented in the u domain, and when translated to the y domain, must be projected onto the space spanned by the $\nabla_u \mathcal{G}\, \nabla_u \mathcal{G}^\top$, whereas LMC sampling operates directly on y without projection. In some sense, the new formulation can be viewed as a projected gradient flow onto the tangent kernel space with the kernel spanned by columns of $\nabla_u \mathcal{G}$.

On the other hand, the connection to LMC also inspires the possibility of introducing stochasticity, particularly the Brownian motion, to avoid dealing with Dirac delta functions. Indeed, denoting $C(u(t)) = \nabla_u \mathcal{G}|_{u(t)} \nabla_u \mathcal{G}|_{u(t)}^\top$, and assuming that $\mathcal{G}$ is invertible, we set $B(y) = C(\mathcal{G}^{-1}(y))$. Then the evolution equation for ρ_y is explicit from (3.18) or (3.8):

$$\partial_t \rho_y = \nabla_y \cdot \left(\rho_y B(y) \nabla_y \log \left(\frac{\rho_y}{\rho_y^*} \right) \right) = \nabla_y \cdot (B \nabla_y \rho_y + \rho_y B \nabla_y f) ,$$

where we assume $\rho_y^* \propto e^{-f(y)}$. As a consequence, we have the following stochastic particle method:

$$\mathrm{d}y_t = -B(y(t)) \nabla_y f(y_t) \mathrm{d}t + \sqrt{2B(y(t))} \mathrm{d}W_t . \tag{3.29}$$

In the most simplified case, consider a linear dependence by letting $y = \mathcal{G}(u) = \mathsf{A}u$. Then we have $\nabla_u \mathcal{G} = \mathsf{A}$, and (3.29) becomes

$$\mathrm{d}y_t = -\mathsf{A}\mathsf{A}^\top \nabla_y f \mathrm{d}t + \sqrt{2\mathsf{A}\mathsf{A}^\top} \mathrm{d}W_t ,$$

a formulation that resembles the ensemble Kalman sampler developed in [12], where the matrix in front of $\nabla_y f$ is the data ensemble covariance matrix.

Finally, we draw the connection to the mirror descent method that the update formula (3.21) for u carries [3, 18, 32]. The mirror descent performs GD on the mirror variable, defined by taking the gradient of a convex function. For example, one can define a convex function ϕ on $\mathbb{R}^m \ni x$ and the mirror variable $z(x) = \nabla_x \phi(x)$. The mirror descent in the continuous-in-time limit represents

$$\frac{\mathrm{d}}{\mathrm{d}t} z = -\nabla f(x) , \quad \text{or equivalently} \quad \frac{\mathrm{d}}{\mathrm{d}t} x = -H_\phi^{-1} \nabla f(x) ,$$

where $H_\phi(x)$ is the Hessian term of ϕ evaluated at x. In our case shown in (3.18), if we view y as x and u as z, the descending formula in (3.18) writes as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} x = -\nabla_z \mathcal{G}|_{z(t)} \nabla_z \mathcal{G}^\top|_{z(t)} \nabla_x f(x) .$$

If $\nabla_z \mathcal{G}(z)$ is of full row rank for any fixed z, the matrix $\nabla_z \mathcal{G} \nabla_z \mathcal{G}^\top$ is naturally strictly positive definite. If one can view it as the Hessian of a convex function ϕ, meaning $H_\phi^{-1}(z) = \nabla_z \mathcal{G} \nabla_z \mathcal{G}^\top$, then our update formula (3.21) can be seen as a mirror descending procedure on u using the mirror function ϕ.

4. Well-posedness result for linear push-forward operators. Like deterministic PDE-constrained optimization problems, when the differential equation has parameters that have inherent randomness, the above formulated SDE-constrained optimization using the gradient flow structure and the associated particle method for implementing the gradient flow are rather generic: the dimensions m and n in (1.1) can be arbitrary in the execution of the algorithm. However, the performance of the GD algorithm and its capability of capturing the global minimum highly depend on the structure of the forward map $\mathcal{G}$. This makes providing a full-fledged convergence theory impossible.

Nevertheless, we can pinpoint certain properties when the push-forward map $\mathcal{G}$ is linear. Resonating the situation in the deterministic setting, we carry out the studies by separating the discussion into overdetermined and underdetermined scenarios. In each scenario, we will begin our discussion with the deterministic setup (1.2) with respect to the L^2 geometry and proceed with the stochastic setup (1.5), emphasizing the similarity.

First, we fix the notation. Since $\mathcal{G}$ is linear, we denote it as

$$y = \mathcal{G}(u) = \mathsf{A}u\,, \quad \text{with} \quad \mathsf{A} \in \mathbb{R}^{n\times m}$$

throughout this section. We also assume that A is full-rank, in the sense that $\operatorname{rank}(\mathsf{A}) = \min\{m,n\} =: r$, so there are no redundant rows/columns, and the size of the matrix determines with the system is under- or overdetermined. Additionally, we conduct the compact SVD for A and write

$$\mathsf{A} = \mathsf{V}\mathsf{S}\mathsf{U}^\top\,, \quad \text{with} \quad \mathsf{V} \in \mathbb{R}^{n\times r}\,, \quad \mathsf{S} \in \mathbb{R}^{r\times r}\,, \quad \mathsf{U} \in \mathbb{R}^{m\times r}\,. \tag{4.1}$$

To begin with, we realize that in this linear setting, the gradient flow equation (3.2) is much more explicit:

$$\partial_t \rho_u - \nabla_u \cdot \left(\rho_u \nabla_u \left(\frac{\delta E}{\delta \rho_u}\right)\right) = 0\,, \quad E(\rho_u) = D(\rho_y, \rho_y^*) = D(\mathsf{A}_\sharp \rho_u, \rho_y^*)\,. \tag{4.2}$$

All discussions below extend this analysis to both underdetermined and overdetermined scenarios. We use ρ to denote probability density and measure interchangeably.

4.1. Underdetermined scenario. The underdetermined scenario corresponds to the case when $n \le m$. That is, the number of the to-be-determined parameters is no fewer than the collected data, and matrix A is short-wide. We incorporate the fully determined matrix ($n = m$) in this regime as a special case. When $n < m$, in either deterministic or stochastic settings, we expect infinitely many solutions to exist.

4.1.1. Deterministic case. The optimization (1.2) in the linear case becomes

$$\min_u \|\mathsf{A}u - y^*\|^2\,. \tag{4.3}$$

When A is fully determined, the solution is unique, but when $n < m$, there are infinitely many possibilities to choose u so that the $y^* = \mathsf{A}u$ exactly. In practice, to select a unique parameter, one typical approach is to add a regularization term. This way, the selected parameter configuration not only minimizes (4.3) but also satisfies certain properties known a priori. One of the most classic examples is Tikhonov regularization: $\min_u \|\mathsf{A}u - y^*\|^2 + \|u - u_0\|^2$. This ensures that the error term $y - \mathsf{A}u$ is small and that the optimizer is relatively close to the suspected ground truth u_0. Here we will take a different point of view. Instead of adding a regularization, we run GD on the vanilla objective (4.3). As expected, the choice of the initial guess u_{i} plays the role of selecting a unique minimizer. In other words, the optimization method itself, which, in our case, is the GD algorithm, has an implicit regularization effect on the converging solution.

To this end, we first augment A with $\tilde{\mathsf{A}}$ to form a rank-m matrix and correspondingly define the augmented y^{ex}:

$$\mathsf{A}^{\mathrm{ex}} = \begin{bmatrix} \mathsf{A} \\ \tilde{\mathsf{A}} \end{bmatrix}, \quad y^{\mathrm{ex}} = \mathsf{A}^{\mathrm{ex}} u = \begin{bmatrix} \mathsf{A}u \\ \tilde{\mathsf{A}}u \end{bmatrix} = \begin{bmatrix} y \\ \tilde{y} \end{bmatrix}. \tag{4.4}$$

Note the definition of $\tilde{\mathsf{A}}$ is not unique, but it does not affect the upcoming calculation. The easiest choice is to set $\tilde{\mathsf{A}}^\top = \mathsf{U}^\perp$, where $\mathsf{U}^\perp$ is the orthogonal complement of U given in (4.1). As a result, the right singular vector set for $\tilde{\mathsf{A}}$ is $\mathsf{U}^\perp$. Suppose $u^* \in \{u : \mathsf{A}u = y^*\}$. Then the solution set can be written as

$$\mathcal{S} = \{u^* + \tilde{u} : \quad \mathsf{A}\tilde{u} = 0\} = \{u^* + \mathrm{span}\mathsf{U}^\perp\}. \tag{4.5}$$

The particular solution chosen from the solution set $\mathcal{S}$ is uniquely determined by the optimization process and the initial data. Suppose we perform GD on (4.3). Then in the continuous-time limit, it amounts to solving the following ODE:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\mathsf{A}^\top (\mathsf{A}u - y^*), \quad \text{with} \quad u(t=0) = u_{\mathrm{i}}. \tag{4.6}$$

The following result shows that the GD leads us to the solution that agrees with u_0 when projected onto $\mathsf{U}^\perp$ and agrees with u^* when projected onto U.

Proposition 4.1. *The equilibrium solution to (4.6), denoted by u_f, given the initial iterate u_i, can be written as*

$$u_f = \mathsf{U}\mathsf{U}^\top u^* + \mathsf{U}^\perp (\mathsf{U}^\perp)^\top u_i.$$

Moreover, $y_f^{\mathrm{ex}} = \mathsf{A}^{\mathrm{ex}} u_f$ as defined in (4.4) satisfies

$$y_f = y^* \quad \text{and} \quad \tilde{y}_f = \tilde{\mathsf{A}} u_i. \tag{4.7}$$

One aspect of this result is that the generated solution, when confined to the space spanned by A, entirely agrees with the given data u^*, and when confined to the augmented section, purely agrees with the generated data from the initial guess u_0. When $n = m$, $\mathsf{U}^\perp$ only contains the zero vector, so $u_{\mathrm{f}} \equiv u^*$. The proof of the proposition is rather standard and is put in Appendix A for completeness.

4.1.2. Stochastic case. The situation in the stochastic setting is an analogy to that in the deterministic setting. In the current underdetermined situation, the push-forward map $\mathcal{G} = \mathsf{A}$ has more degrees of freedom to pin in the parameter space than the data offers. Therefore, it is guaranteed that one can find infinitely many ρ_u that achieve the exact match:

$$\rho_y = \mathsf{A}_\sharp \rho_u = \rho_y^*.$$

As with the deterministic case, the particular solution in the solution set $\mathcal{S}$ we obtain is determined by the initial guess and the optimization algorithm in a combined manner. We expect the same to hold in the stochastic case. Furthermore, we use the same notation as in (4.4). The final conclusion is an analogy of Proposition 4.1.

Theorem 4.2. *Suppose* (4.2) *using the initial data* ρ_u^0 *has an equilibrium solution, and we denote it to be* ρ_u^∞ *and let* $\rho_{y^{ex}}^\infty$ *be the push-forward density of* ρ_u^∞ *under the map* A^{ex}*, i.e.,*

$$\rho_{y^{\text{ex}}}^\infty = \mathsf{A}_\sharp^{ex} \rho_u^\infty \,. \tag{4.8}$$

Then we can uniquely determine the marginal distributions of $\rho_{y^{\text{ex}}}^\infty$*, in the sense that*

- *the marginal distribution on* y *of* $\rho_{y^{\text{ex}}}^\infty$ *entirely recovers that of the data* ρ_y^**;*
- *the marginal distribution on* $\tilde{y}$ *of* $\rho_{y^{\text{ex}}}^\infty$ *is uniquely determined by that of* ρ_y^0*.*

A very natural corollary of Theorem 4.2, when A is fully determined, suggests the unique recovery of the ground truth.

Corollary 4.3. *When* $n = m$*, i.e., the matrix* A *is fully determined, under the same assumptions of Theorem* 4.2*, the equilibrium solution*

$$\rho_y^\infty = \mathsf{A}_\sharp \rho_u^\infty = \rho_y^*$$

is unique and independent of the initial distribution.

To prove Theorem 4.2, we realize that since the system is underdetermined, there are infinitely many solutions. The particular solution we get is the single distribution function that falls at the intersection of the solution set, denoted by $\mathcal{S}_\rho$ and defined in (4.10) and the gradient flow dynamics (4.2). The proof is similar to what we had in the deterministic setting: we first identify the class of functions on the gradient flow dynamics and then evaluate when and how they intersect with $\mathcal{S}_\rho$.

The following lemma first characterizes its solution by following the flow trajectory.

Lemma 4.4. *Suppose one starts the gradient flow equation* (4.2) *with the initial condition* $\rho_u(u,0) = \rho_u^0$*. Then the solution lives in the following set:*

$$\rho_u(t) \in \left\{ \rho_u^0 + h \,, \quad \textit{with} \quad \int h(u)\mathrm{d}u = 0 \,, \quad \tilde{\mathsf{A}}_\sharp h = 0 \right\}, \tag{4.9}$$

where $\tilde{\mathsf{A}}$ *is defined in* (4.4)*.*

Proof. Let ρ_u be the solution to (4.2) and ρ_u^0 be the initial distribution. Then $\rho_u(u,\tau)$ differs from ρ_u^0 by

$$\rho_u(u,\tau) - \rho_u^0(u) = h(u;\tau) := \int_0^\tau \nabla_u \cdot \left(\rho_u(t) \nabla_u \frac{\delta D}{\delta \rho_y(t)} (\mathsf{A}u) \right) \mathrm{d}t \,,$$

a function of u parameterized by τ. Essentially, to demonstrate that $\int h(u)\mathrm{d}u = 0$ and $\tilde{\mathsf{A}}_\sharp h = 0$, we only need to establish the validity of these two equations for any function $q(u)$ in the form of

$$q(u) = \nabla_u \cdot \left(\rho_u \nabla_u \frac{\delta D}{\delta \rho_y} (\mathsf{A}u) \right) .$$

The mean-zero property $\int q(u)\mathrm{d}u = 0$ is easy to show given that q has a divergence form. To show $\tilde{\mathsf{A}}_\sharp q = 0$, we recall that for any ψ that is integrable with respect to q upon being composed with $\tilde{\mathsf{A}}$, by performing integration by parts, we have

$$\int \psi(\tilde{\mathsf{A}}u)q(u)\mathrm{d}u = -\int \nabla_u \psi(\tilde{\mathsf{A}}u)\cdot\nabla_u\left(\frac{\delta D}{\delta \rho_y}(\mathsf{A}u)\right)\rho_u(u)\mathrm{d}u$$
$$= -\int \underbrace{\tilde{\mathsf{A}}^\top \nabla_y \psi(y)|_{y=\tilde{\mathsf{A}}u}}_{\xi_1}\cdot\underbrace{\mathsf{A}^\top\nabla_y\left(\frac{\delta D}{\delta\rho_y}(y)\right)|_{y=\mathsf{A}u}}_{\xi_2}\rho_u(u)\mathrm{d}u\,.$$

Note that the row spaces of A and $\tilde{\mathsf{A}}$ are U and $\mathsf{U}^\perp$, respectively, so ξ_1 and ξ_2 belong to these two perpendicular spaces. We then have $\int \psi(\tilde{\mathsf{A}}u)q(u)\mathrm{d}u = 0$, yielding $\tilde{\mathsf{A}}_\sharp q = 0$. Since at each infinitesimal time, q satisfies these two conditions, we prove (4.9). ∎

Next, we study the steady state of the gradient flow system (4.10).

Lemma 4.5. *When D is the KL divergence, and if we let ρ_u^* be a reference probability measure (i.e., $\mathsf{A}_\sharp\rho_u^* = \rho_y^*$), then all equilibria of (4.2) are in the following set:*

$$\mathcal{S}_\rho = \left\{\rho_u^* + g : \int g(u)\mathrm{d}u = 0\,, \mathsf{A}_\sharp g = 0\right\}. \tag{4.10}$$

Here, $\mathsf{A}_\sharp g = 0$ can also be interpreted as $\int \psi(\mathsf{A}u)g(u)\mathrm{d}u = 0$, which means that for all ψ, when composed with A, it is g-integrable to 0.

Proof. Recalling (4.2) and the discussion leading to (3.3), we have that the equilibrium set consists of the states ρ_u^∞ at which the Fréchet derivative is trivial on the support. This, combined with (3.6), gives

$$\nabla_u \frac{\delta E}{\delta \rho_u}\Big|_{\rho_u^\infty}(u) = \nabla_u \frac{\delta D}{\delta \rho_y}\Big|_{\mathsf{A}\sharp\rho_u^\infty}(\mathsf{A}u) = \mathsf{A}^\top\nabla_y \frac{\delta D}{\delta\rho_y}\Big|_{\mathsf{A}\sharp\rho_u^\infty}(\mathsf{A}u) = 0\,.$$

Since A is a flat matrix with full rank,

$$\mathsf{A}^\top\nabla_y\frac{\delta D}{\delta\rho_y}\Big|_{\mathsf{A}\sharp\rho_u^\infty}(\mathsf{A}u) = 0 \quad \text{on the support of } \rho_u^\infty\,. \tag{4.11}$$

In the case when D is the KL divergence, based on (3.7), we have that $\rho_y^\infty \propto \rho_y^*$. Since they both are probability measures, we can conclude that $\rho_y^\infty = \rho_y^*$. Since $\rho_y^\infty = \mathsf{A}_\sharp\rho_u^\infty$, this means that ρ_u^∞ is in $\mathcal{S}_\rho$ defined in (4.10). ∎

Remark 1. To show that $\rho_y^\infty = \rho_y^*$ from (4.11), D only needs to be strictly displacement convex with respect to ρ_y (see section 5.2.1 in [31] for its definition). Indeed, note that if we view $D(\rho_y^\infty, \rho_y^*)$ as a functional of ρ_y^∞, the equilibrium set of its Wasserstein gradient flow is (4.11). Then under the convexity condition mentioned above, the equilibrium set has one element [22], which is the unique minimizer of $D(\rho_y^\infty, \rho_y^*)$.

These two lemmas prepare us to investigate the specific solution by following (4.2) from an initial ρ_u^0. We are ready to prove the main theorem.

Proof of Theorem 4.2. From (4.9)–(4.10) we see that the equilibrium ρ_u^∞ admits the following expression:

$$\rho_u^\infty = \rho_u^0 + h = \rho_u^* + g\,,$$

where $\int h\mathrm{d}u = \int g\mathrm{d}u = 0$ and $\mathsf{A}_\sharp g = 0$, $\tilde{\mathsf{A}}_\sharp h = 0$. Therefore

$$\mathsf{A}_\sharp \rho_u^\infty = \mathsf{A}_\sharp \rho_u^*, \quad \tilde{\mathsf{A}}_\sharp \rho_u^\infty = \tilde{\mathsf{A}}_\sharp \rho_u^0.$$

That is, for any integrable test function ψ

$$\int \psi(\mathsf{A}u)\rho_u^\infty(u)\mathrm{d}u = \int \psi(\mathsf{A}u)\rho_u^*(u)\mathrm{d}u, \quad \int \psi(\tilde{\mathsf{A}}u)\rho_u^\infty(u)\mathrm{d}u = \int \psi(\tilde{\mathsf{A}}u)\rho_u^0(u)\mathrm{d}u. \tag{4.12}$$

From (4.8), we have that $\rho_{y^{\mathrm{ex}}}^\infty(\mathsf{A}^{\mathrm{ex}}u)\det(\mathsf{A}^{\mathrm{ex}}) = \rho_u^\infty(u)$. Using the notation in (4.4), the two equations above become

$$\int \psi(y)\rho_y^\infty(y)\mathrm{d}y = \underbrace{\int \psi(y)\rho_{y^{\mathrm{ex}}}^\infty(y^{\mathrm{ex}})\mathrm{d}y^{\mathrm{ex}} = \int \psi(y)\rho_{y^{\mathrm{ex}}}^*(y^{\mathrm{ex}})\mathrm{d}y^{\mathrm{ex}}}_{\text{from (4.12)}} = \int \psi(y)\rho_y^*(y)\mathrm{d}y,$$

where we used $\rho_{y^{\mathrm{ex}}}^* = \mathsf{A}_\sharp^{\mathrm{ex}}\rho_u^*$ and the definition of the marginal distributions $\rho_y^*(y) = \int \rho_{y^{\mathrm{ex}}}^*(y^{\mathrm{ex}} = [y, \tilde{y}])\mathrm{d}\tilde{y}$, and $\rho_y^\infty(y) = \int \rho_{y^{\mathrm{ex}}}^\infty(y^{\mathrm{ex}} = [y, \tilde{y}])\mathrm{d}\tilde{y}$ for the first and last equal signs. Similarly the second equation in (4.12) becomes

$$\int \psi(\tilde{y})\rho_{y^{\mathrm{ex}}}^\infty(y^{\mathrm{ex}})\mathrm{d}y^{\mathrm{ex}} = \int \psi(\tilde{y})\rho_{y^{\mathrm{ex}}}^0(y^{\mathrm{ex}})\mathrm{d}y^{\mathrm{ex}}, \quad \text{where } \rho_y^0 = \mathsf{A}_\sharp^{\mathrm{ex}}\rho_u^0.$$

Considering that ψ can be chosen as any function, we obtain all the moments, and thus the marginal distribution $\rho_{y^{\mathrm{ex}}}^\infty$ on y and $\tilde{y}$, finishing the proof. ∎

4.2. Overdetermined scenario. The overdetermined scenario corresponds to the case when $n > m$, so there are more data to fit than the to-be-determined parameters. It is often unlikely to find a solution that satisfies the equation exactly, but one can nevertheless find the best approximation that minimizes a chosen misfit function.

4.2.1. Deterministic case. As an example, we look for the configuration that provides the minimum misfit under the vector 2-norm. That is,

$$\min_u \frac{1}{2}\|\mathsf{A}u - y^*\|_2^2.$$

For a linear system like this, the minimizer is explicit:

$$u^* = (\mathsf{A}^\top\mathsf{A})^{-1}\mathsf{A}^\top y^* =: \mathsf{A}^\dagger y^*, \tag{4.13}$$

and hence, calling (4.1)

$$y = \mathsf{A}u^* = \mathsf{A}\mathsf{A}^\dagger y^* = \mathsf{V}\mathsf{V}^\top y^*, \quad \text{or equivalently} \quad y = y_{\mathsf{A}}^* = \mathrm{Proj}_{\mathsf{V}}\, y^*. \tag{4.14}$$

In other words, y is y^* projected onto the column space of V (also the column space of A), and thus $\mathsf{V}^\top y = \mathsf{V}^\top y^*$ and $\left(\mathsf{V}^\perp\right)^\top y = 0$.

4.2.2. Stochastic case. In the stochastic case, we expect to obtain a similar result that suggests ρ_y^∞ is a "projection" of ρ_y^* onto the column space of A. More specifically, we have the following theorem that characterizes the equilibrium ρ_y^∞ when ρ_y^* has a separable structure.

Theorem 4.6. *Let ρ_u^∞ be the equilibrium solution to (4.2). Consider D to be KL, i.e., $D(\rho_y, \rho_y^*) = \int \rho_y \log \frac{\rho_y}{\rho_y^*} \mathrm{d}y$, and ρ_y^* has the separable form*

$$\rho_y^*(y) = e^{f_1(y_{\mathsf{A}})} e^{f_2(y_{\mathsf{A}^\perp})},$$

where $y_{\mathsf{A}} = \mathsf{V}\mathsf{V}^\top y$, with V being the column space of A, and $y_{\mathsf{A}^\perp} = y - y_{\mathsf{A}}$ being the perpendicular component. Then

$$\rho_y^\infty(y) = \mathsf{A}\mathsf{A}_\sharp^\dagger \rho_y^* \propto e^{f_1(y_{\mathsf{A}})}.$$

This theorem is the counterpart of its deterministic version stated in (4.14), showing that we can identify partial information of ρ_y^∞—the part that is in the column space of A (or equivalently V). The perpendicular direction information is projected out.

Proof. To start off, we assume $\rho_y^\infty(y) \propto e^{g(y_{\mathsf{A}})}$, and we will show that g is the same as f_1 up to a constant addition. It is safe to assume $\rho_y^\infty(y)$ has no $y_{\mathsf{A}^\perp}$ component because it is pushed forward by A, so its support is confined to the range of A.

Recall the argument leading to (3.3), and combine it with the calculation (3.6). We have that the equilibrium of (4.2) in this case satisfies

$$\nabla_u \frac{\delta E}{\delta \rho_u}\bigg|_{\rho_u^\infty} = \nabla_u \frac{\delta D}{\delta \rho_y}\bigg|_{\rho_y^\infty}(\mathsf{A}u) = \mathsf{A}^\top \nabla_y \frac{\delta D}{\delta \rho_y}\bigg|_{\rho_y^\infty}(\mathsf{A}u) = 0 \;\Rightarrow\; \nabla_y \frac{\delta D}{\delta \rho_y}\bigg|_{\rho_y^\infty}(\mathsf{A}u) \in \mathsf{A}^\perp.$$

Considering $\nabla_y = \mathsf{V}\mathsf{V}^\top \nabla_{y_{\mathsf{A}}} + \mathsf{V}^\perp (\mathsf{V}^\perp)^\top \nabla_{y_{\mathsf{A}^\perp}}$, we should have, for all u,

$$\nabla_{y_{\mathsf{A}}} \frac{\delta D}{\delta \rho_y}\bigg|_{\rho_y^\infty}(\mathsf{A}u) = 0.$$

Noting $\nabla_{y_{\mathsf{A}}} \frac{\delta D}{\delta \rho_y}|_{\rho_y^\infty} = \nabla_{y_{\mathsf{A}}} \log \rho_y^\infty - \nabla_{y_{\mathsf{A}}} \log \rho_y^*$, we then have

$$\nabla_{y_{\mathsf{A}}} g = \nabla_{y_{\mathsf{A}}} f_1 \quad \text{on} \quad \mathrm{Range}(\mathsf{A}).$$

Thus, $g = f_1$ up to a constant, finishing the proof. ∎

4.3. Setting data discrepancy D to be W_2. Most results developed in this paper so far set D as the KL divergence. We switch gears and adopt (3.9) in this subsection, by setting D to be the quadratic Wasserstein metric W_2. In this case, the variational formulation (1.5) becomes

$$\min_{\rho_u \in \mathcal{P}(\mathcal{A})} W_2(\mathcal{G}_\sharp \rho_u, \rho_y^*), \tag{4.15}$$

with $\mathcal{P}(\mathcal{A})$ denoting all probability distributions over the parameter domain $\mathcal{A}$ as usual.

In the fully determined case where A is invertible, the minimizer will be uniquely given by ${\mathsf{A}^{-1}}_\sharp \rho_y^*$. In the underdetermined case, the same argument in section 4.1 still holds and no

uniqueness can be obtained. However, when A is overdetermined with full column rank and ρ_y^* is absolutely continuous with respect to the Lebesgue measure, we claim that the unique minimizer to (4.15) is $\mathsf{A}^\dagger{}_\sharp\rho_y^*$.

Theorem 4.7. *Consider D to be W_2, $\mathcal{A}=\mathbb{R}^m$, and set $\mathcal{G}=\mathsf{A}$, an overdetermined matrix with full column rank. Moreover, assume the reference data distribution ρ_y^* is absolutely continuous with respect to the Lebesgue measure, and is supported on $\mathcal{R}$, the closure of a bounded connected open set. Then we have that*

- *the variational problem (4.15) has a unique minimizer $\mathsf{A}_\sharp^\dagger\rho_y^*$;*
- *the Wasserstein gradient flow of (3.1) with D being the W_2 metric, as given in (3.12), has a unique equilibrium, which is also the unique minimizer $\rho_u^\infty=\mathsf{A}_\sharp^\dagger\rho_y^*$.*

Proof. First, we show that $\mathsf{A}_\sharp^\dagger\rho_y^*$ is the global minimizer. Define a set $\mathcal{S}=\{\rho:\rho=\mathsf{A}_\sharp\rho_u$ for $\rho_u\in\mathcal{P}(\mathcal{A})\}$ collecting all probability measures pushed by A. Then for any $\rho\in\mathcal{S}$, we have $\mathrm{supp}(\rho)\subset\mathrm{col}(\mathsf{A})$. Furthermore,

$$W_2^2(\rho,\rho_y^*)=\int_y |T_\rho(y)-y|^2\rho_y^*(y)\mathrm{d}y \tag{4.16}$$

$$\geq\int_y |\mathsf{A}\mathsf{A}^\dagger y-y|^2\rho_y^*(y)\mathrm{d}y \tag{4.17}$$

$$\geq W_2^2\left((\mathsf{A}\mathsf{A}^\dagger)_\sharp\rho_y^*,\,\rho_y^*\right). \tag{4.18}$$

In the derivation, (4.16) holds by definition where we denote by T_ρ the optimal transport map from ρ_y^* to ρ. The existence of the optimal map is guaranteed since ρ_y^* is absolutely continuous [25, Thm. 1.17]. Inequality (4.17) holds because of the pointwise inequality:

$$\operatorname*{arg\,min}_{x\in\mathrm{col}(\mathsf{A})}|x-y|^2=\mathsf{A}\mathsf{A}^\dagger y\quad\Rightarrow\quad|\mathsf{A}\mathsf{A}^\dagger y-y|^2\leq|T_\rho(y)-y|^2,$$

where we deployed the fact that $\mathrm{supp}(\rho)\subset\mathrm{col}(\mathsf{A})$ and thus $\mathrm{range}(T_\rho)\subset\mathrm{col}(\mathsf{A})$. Inequality (4.18) comes from the definition of the Wasserstein distance (note that we do not necessarily claim $\mathsf{A}\mathsf{A}^\dagger$ is the optimal map from ρ_y^* to $\left(\mathsf{A}\mathsf{A}^\dagger\right)_\sharp\rho_y^*$). Consequent to this derivation, we have

$$\mathsf{A}_\sharp\left(\mathsf{A}_\sharp^\dagger\rho_y^*\right)=\left(\mathsf{A}\mathsf{A}^\dagger\right)_\sharp\rho_y^*=\operatorname*{arg\,min}_{\rho=\mathsf{A}_\sharp\rho_u,\,\forall\rho_u\in\mathcal{P}(\mathcal{A})}W_2(\rho,\rho_y^*),$$

or equivalently

$$\mathsf{A}_\sharp^\dagger\rho_y=\operatorname*{arg\,min}_{\rho_u\in\mathcal{P}(\mathcal{A})}W_2(\mathsf{A}_\sharp\rho_u,\rho_y^*), \tag{4.19}$$

finishing the proof that $\mathsf{A}_\sharp^\dagger\rho_y^*$ is the global minimizer.

Next, we show that $\mathsf{A}_\sharp^\dagger\rho_y^*$ is also the unique equilibrium of the corresponding Wasserstein gradient flow equation (3.12). That is, the equilibrium of the gradient flow equation is the global minimizer of (3.1). At equilibrium, we have

$$\nabla_u\cdot(\rho_u^\infty\nabla_u\phi^*(\mathsf{A}u))=0\quad\text{on the support of }\rho_u^\infty,$$

where ϕ^* is the Kantorovich potential associated with $\rho_y^\infty = \mathsf{A}_\sharp \rho_u^\infty$; see (3.10)–(3.11). This implies that

$$\mathsf{A}^\top \nabla_y \phi^*(y) = 0 \quad \text{on the support of } \rho_y^\infty . \tag{4.20}$$

Based on (4.20), we deduce that

$$\nabla_y \phi^* : \mathrm{supp}(\rho_y^\infty) \subset \mathrm{col}(\mathsf{A}) \longrightarrow \mathrm{col}(\mathsf{A})^\perp,$$

where $\mathrm{col}(\mathsf{A})^\perp$ denotes the orthogonal complement of the linear subspace $\mathrm{col}(\mathsf{A})$.

On the other hand, one can express the Kantorovich potential with respect to ρ_y^*, denoted by ψ^*, using the c-transform:

$$\psi^*(x) = (\phi^*(y))^c := \min_{y \in \mathrm{col}(\mathsf{A})} \left\{ \frac{1}{2}\|x-y\|^2 - \phi^*(y) \right\}, \quad x \in \mathcal{R}.$$

For any $x \in \mathcal{R}$, we denote by y_x the minimizer of the above c-transform. Since the minimization problem is strictly convex, the first-order optimality condition is also sufficient:

$$x - y_x - \nabla_y \phi^*(y_x) \perp \mathrm{col}(\mathsf{A}). \tag{4.21}$$

Based on (4.20), we know $\nabla_y \phi^*(y_x) \in \mathrm{col}(\mathsf{A})^\perp$. Moreover, $y_x \in \mathrm{col}(\mathsf{A})$. Therefore, (4.21) is equivalent to the following orthogonal decomposition of x:

$$x = y_x + z, \quad z \in \mathrm{col}(\mathsf{A})^\perp .$$

Since orthogonal decomposition with respect to $\mathrm{col}(\mathsf{A})$ is unique, we obtain that $y_x = \mathsf{A}\mathsf{A}^\dagger x$ $\forall x \in \mathcal{R}$. Based on the optimal transportation theory [25, 31], $y_x = \mathsf{A}\mathsf{A}^\dagger x = T(x)$, where T is the optimal transport map from ρ_y^* to ρ_y^∞. That is,

$$\rho_y^\infty = (\mathsf{A}\mathsf{A}^\dagger)_\sharp \rho_y^*,$$

which implies that $\rho_u^\infty = \mathsf{A}_\sharp^\dagger \rho_y^*$. ∎

Equation (4.19) shows that we have a simple form for the minimizer of the variational problem (1.5) and the equilibrium of the gradient flow equation (3.1) if D is W_2 and A is tall-skinny with full column rank. Just as $\mathsf{A}^\dagger y^*$ provides the optimizer in the deterministic case (see (4.13)), the minimizer in the stochastic context is its vanilla extension $\mathsf{A}_\sharp^\dagger \rho_y^*$ under the W_2 case, also the only equilibrium of the Wasserstein gradient flow under conditions.

5. Numerical examples.

This section presents a few numerical inversion examples using the proposed particle method (3.20). Throughout the section, we use the KL divergence (3.7) as the objective function and the W_2 metric to determine the geometry. In all examples in this section, we set $\epsilon = 0.5$ as the hyperparameter in the density estimation step (3.21). The time step size to discretize the gradient dynamics (3.21) is chosen using the Armijo backtracking line search.

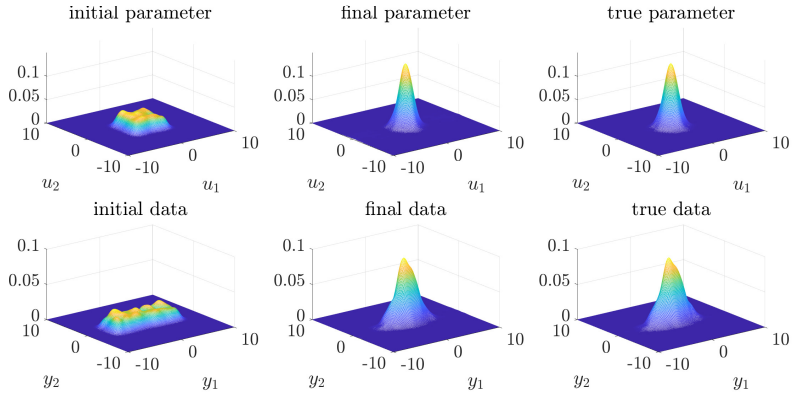


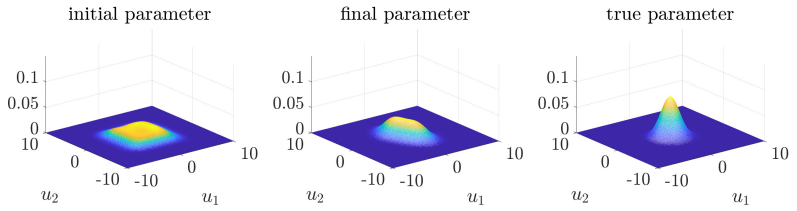
Figure 1. *Parameter and data distributions in the fully determined case with the map* $T(x) = \mathsf{A}x$, *where* $\mathsf{A} = \mathrm{diag}([2, 0.75])$.

5.1. Linear push-forward map. We first present examples with the linear push-forward map $y = \mathcal{G}(u) = \mathsf{A}u$, which we theoretically studied in section 4. Given the matrix A is fully, under-, or overdetermined, we show that there are different phenomena in inversion using the gradient flow approach (3.2).

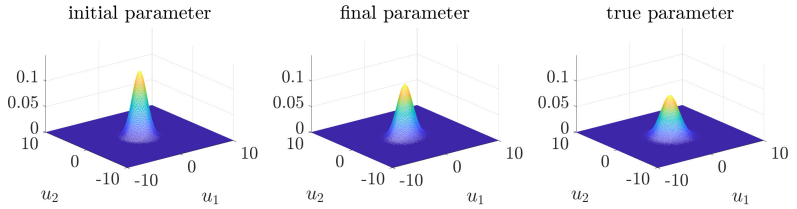
5.1.1. Fully determined case. First, we consider a fully determined case. We let $\mathsf{A} = \mathrm{diag}([2, 0.75])$, and the true parameter $u \sim \mathcal{N}(0, I)$. As a result, the reference data $y \sim \mathcal{N}(0, \mathsf{A}\mathsf{A}^\top)$, represented by an empirical distribution with 1000 i.i.d. particles. The initial guess for the parameter is the uniform distribution $\mathcal{U}[-3, 3]^2$, which is also represented by 1000 i.i.d. samples. We then follow the particle method (3.20) to implement the gradient flow equation (3.2). The convergence results after 30 iterations are shown in Figure 1. We have recovered both the parameter distribution and the data distribution well.

5.1.2. Underdetermined case. Next, we consider that $\mathsf{A} = [2, 0.75]$, which is underdetermined. The true parameter distribution is $\mathcal{N}(0, I)$ and the reference data distribution $y \sim \mathcal{N}(0, \mathsf{A}\mathsf{A}^\top = 4.5625)$, which we use 3000 i.i.d. samples to represent. Since A is underdetermined, based on our analysis in section 4, there are infinitely many solutions u, the same as the deterministic case. Since we use the gradient flow formulation (3.2), the algorithm can only find one of the solutions, which is determined by the initial condition. We then demonstrate this through numerical examples.

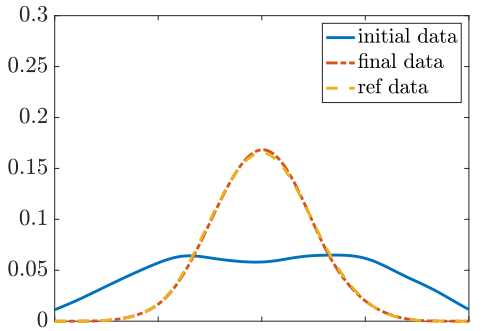
We consider two different initial distributions u^1 and u^2, as shown in Figure 2. We also use 3000 particles to implement the Wasserstein gradient flow of minimizing the KL divergence between the reference and the data distribution computed from the current iterate of the parameter distribution. Although neither of the gradient flows started from the initial distributions converges to the *true* parameter distribution from which we generate the data, their corresponding data distributions match the reference and successfully achieve data fitting. This again demonstrates the intrinsic nonuniqueness of the inverse problem when the linear push-forward map A is underdetermined.



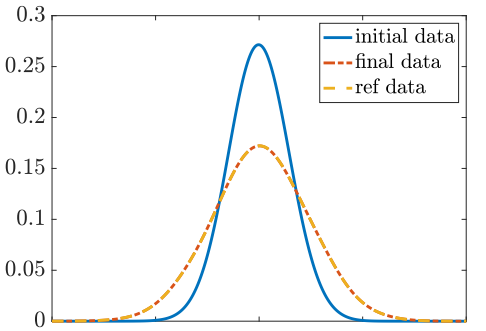
(a) Parameter distribution with initial guess u^1



(b) Parameter distribution with initial guess u^2



(c) Data with initial guess u^2



(d) Data with initial guess u_2

Figure 2. *Underdetermined case under two initial distributions, u^1 and u^2, where the map $\mathcal{G}(u) = \mathsf{A}u$, $\mathsf{A} = [2, 0.75]$. The reference data distribution is computed using the true parameter distribution.*

5.1.3. Overdetermined case. Next, we show the overdetermined case with $\mathsf{A} = [2, 1]^\top$. We set the true parameter distribution to be $\mathcal{N}(0, 1)$ and choose a reference data distribution that is polluted by random noise and thus is not in the range of the forward push-forward map. Inversion in this scenario is similar to the least-squares method in the deterministic case; see section 4. We use 3000 samples to represent the reference data distribution and 3000 particles to implement the gradient flow method (3.2).

In Figure 3, we plot the initial and final converged parameter distributions and the corresponding data distributions pushed forward by the forward map from those parameter distributions. We also show the true parameter distribution and the reference data distribution for comparison. The final data distribution from the recovered parameter distribution does not fit the reference data entirely. However, their marginal distributions along y_{A} (orthogonal projection of y over the column space of A) match exactly. This verifies our result in Theorem 4.6.

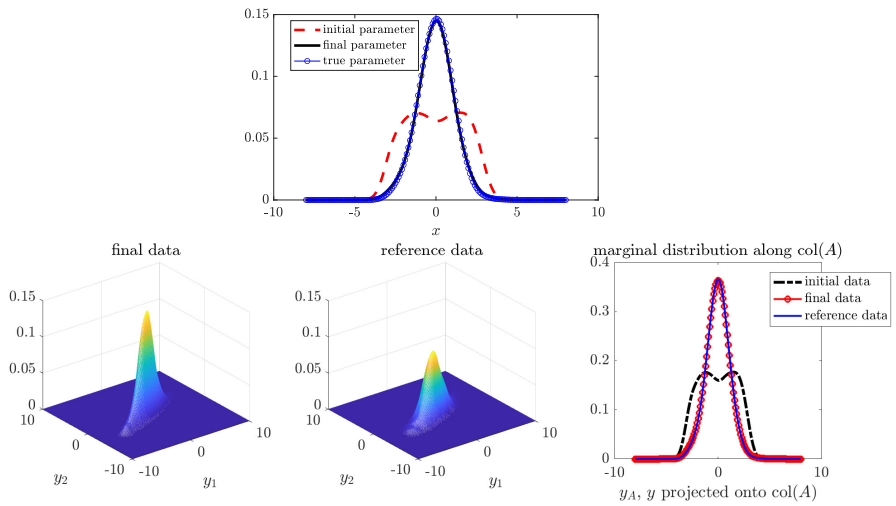


Figure 3. *Overdetermined case with the map $T(x) = \mathsf{A}x$, where $\mathsf{A} = [2,1]^\top$. Although the final recovered data distribution does not fit the reference data distribution entirely, their marginal distributions on y_A match very well, as proved in Theorem 4.6.*

5.2. An inverse problem example. Many inverse problems can be solved in our framework. Let us first consider a one-dimensional (1D) elliptic boundary value problem, a test case considered in [13, 12]:

$$-\big(\exp(u_1)p'(x)\big)' = 1, \quad x \in [0,1], \tag{5.1}$$
$$p(0) = 0, \quad p(1) = u_2.$$

Its analytic solution can be written as

$$p(x) = u_2 x + \exp(-u_1)\left(-\frac{x^2}{2} + \frac{x}{2}\right), \tag{5.2}$$

which shows the log stability of this particular setup (parameter change in response to the PDE solution change). In the setup of [12, eq. (4.4)], the authors consider the forward model mapping the two independent scalar coefficients u_1 and u_2 to the observed data $y_1 := p(x_1)$ and $y_2 := p(x_2)$, where $x_1 = 0.25$ and $x = 0.75$.

Following the same setup, considering u_1 and u_2 are scalar-valued random variables, our observations $[y_1, y_2]^\top$ are also random in nature. We represent ρ_y^* using $N = 5000$ samples in this test and run the gradient flow simulation using $M = 5000$ simulated particles. Two settings are considered.

(1) The true parameters $u_1 \sim \mathcal{N}(0, 0.5)$, and $u_2 \sim \mathcal{U}([0,2])$. The initial guess $u_1^{(0)}, u_2^{(0)}$ both follow $\mathcal{N}(0,2)$.

(2) $u_1 \sim \mathcal{N}(-1.5, 0.5)$, and $u_2 \sim \mathcal{U}([0,2])$. The initial guess $u_1^{(0)} \sim \mathcal{U}([-3,-1])$ and $u_2^{(0)} \sim \mathcal{U}([0,2])$.

Since the analytic solution (5.2) suggests stronger sensitivity of data on the negative values of u_1, we expect a better stability of the inverse problem in setting (2), given its ground truth

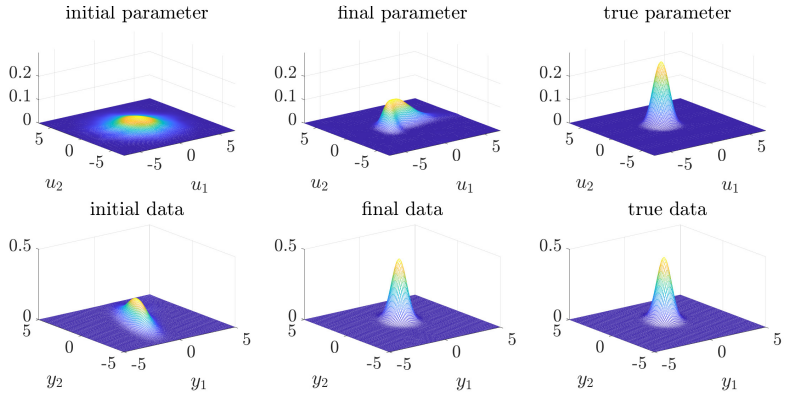


Figure 4. *Numerical inversion based on the* 1*D diffusion equation* (5.1) *with setting* (1).

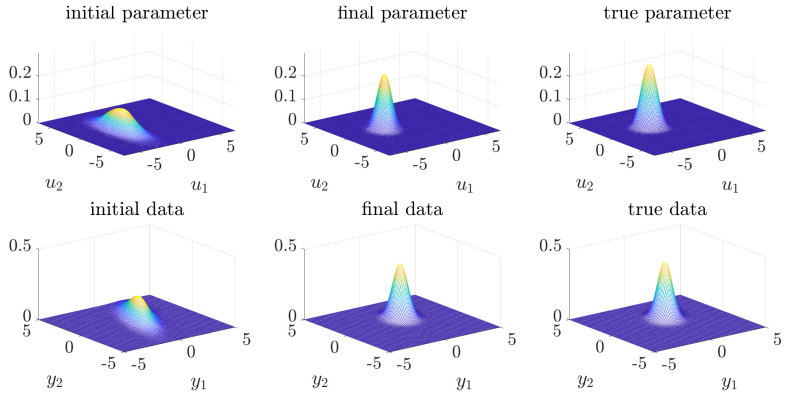


Figure 5. *Numerical inversion based on the* 1*D diffusion equation* (5.1) *with setting* (2).

taking on a negative value with high probability. This is indeed what we observe from the experiments; see Figures 4 and 5. In both cases, the data distributions are matched very well. The recovered parameter distribution, however, demonstrates different features: It is visually far away from the truth in setting (1) but is in much better agreement with the ground truth in setting (2).

The higher-dimensional version of (5.1) becomes

$$
\begin{aligned}
(5.3) \qquad -\nabla\cdot(a(\mathbf{x})\nabla p(\mathbf{x})) &= f(\mathbf{x}), && \mathbf{x}\in D,\\
p(\mathbf{x}) &= \sin(2x_1\pi)\cos(4x_2\pi), && \mathbf{x}=(x_1,x_2)\in\partial D.
\end{aligned}
$$

We consider $D=[0,1]^2$ and $a(\mathbf{x})=\exp(u_1\phi_1(\mathbf{x})+u_2\phi_2(\mathbf{x}))$ where $\phi_1(\mathbf{x})=\frac{10}{9+\pi^2}\cos(\pi x_1)$ and $\phi_2(\mathbf{x})=\frac{10}{9+2\pi^2}\cos(\pi(x_1+x_2))$, following a similar setup in [12, sect. 4.4]. Two observed empirical distributions $y_1=p(\mathbf{x}_1)$ and $y_2=p(\mathbf{x}_2)$ are used to perform the inversion where the receivers are located at $\mathbf{x}_1=(0.25,1)^\top$ and $\mathbf{x}_2=(1,0.5)^\top$. The true distributions are $u_1\sim\mathcal{N}(1,1)$ and $u_2\sim\mathcal{U}([0,1])$, while the initial distribution $u_1^{(0)},u_2^{(0)}\sim\mathcal{U}([-2.5,2.5])$. We use

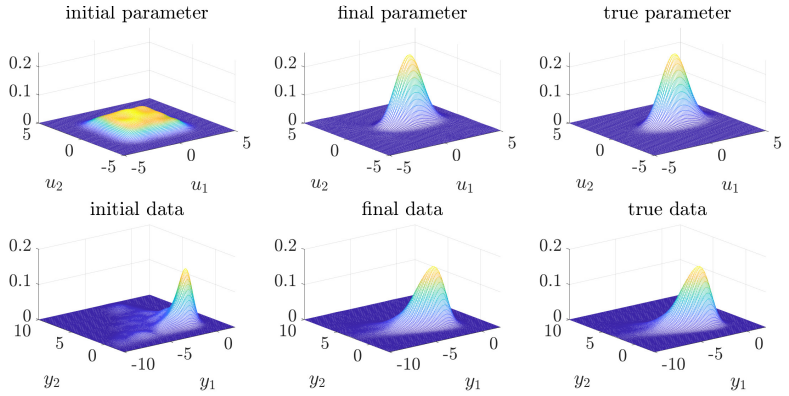
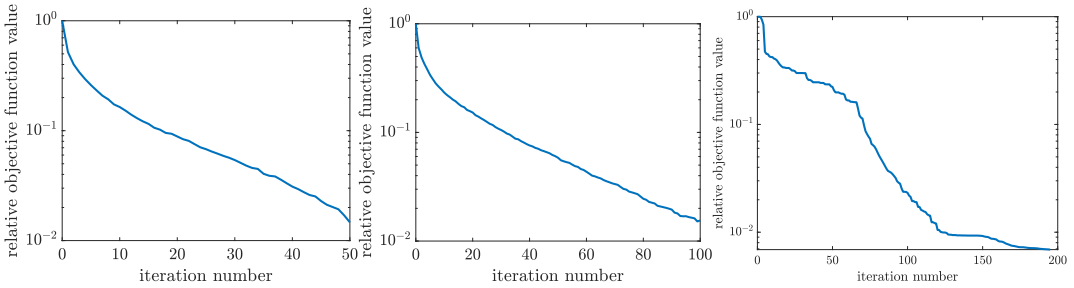


Figure 6. *Numerical inversion based on the 2D diffusion equation (5.3).*



(a) For Figure 4 (b) For Figure 5 (c) For Figure 6

Figure 7. *The objective function decay for the tests shown in Figures 4–6.*

$N = 5000$ particles for the observed empirical distribution and $M = 1000$ for the simulation. Both the parameter reconstruction and its generated data distribution are close to the truth. The smooth-basis parameterization of $a(\mathbf{x})$ significantly improves the well-posedness of the problem. The top row of Figure 6 shows the initial parameter distribution, the inverted parameter distribution, and the true parameter distribution that generates the data. The bottom row of Figure 6 illustrates the respective data distributions. The convergence history of the objective function (i.e., the KL divergence) for these three cases is plotted in Figure 7.

6. Conclusion and discussions. While most research in inverse problems focuses on deterministic unknowns, many real-world problems are inherently stochastic, introducing random variations in the parameters to be inferred. This necessitates a paradigm shift in research from optimizing a single, deterministic value to characterizing the full probability distribution that governs these parameters.

This paper presents a framework for conducting stochastic inverse problems. For linear push-forward maps, we also discuss the well-posedness theory both for the formulation and for the gradient flow algorithm. Particle-based solvers are designed to simulate the gradient flow for finding the optimal probability measure.

The proposed approach shares many traits with Bayesian inversion. While it is true that both formulations return probability distributions of the unknown parameter, there is a stark difference between the two: the sources of randomness are different. In Bayesian inference, it is assumed there is randomness in the prior knowledge (encoded in the prior distribution) and there is measurement error (encoded in the likelihood function). On the contrary, our formulation assumes the data is devoid of any noise, attributing randomness solely to the parameters. To incorporate the randomness in the measurement error, we need to reformulate the problem as a probability distribution over a metric space consisting of probability measures space, and seek for Law$\{\rho_u\}$, instead of ρ_u itself. This would be a much more intricate problem, and we leave it to future research.

The current paper is just the beginning of our research endeavor to explore stochastic inverse problems. The rich geometries of probability spaces present a unique opportunity for computational solver design. We expect different combinations of the data discrepancy and the metric deployed to measure probability distances would present different features of the problem.

Appendix A. Proof of Proposition 4.1.

Proof of Proposition 4.1. The proof is simple algebra. We note that the source term in (4.6) is always in the column space of $\mathsf{A}^\top$, hence the column space of U, which leads to the fact that

$$u(t) \in u_0 + \mathrm{span}\{\mathsf{U}\}\,. \tag{A.1}$$

Since the convergence of gradient descent is achieved when $u(t) \in \mathcal{S}$, we have

$$u_{\mathrm{f}} \in \{u_0 + \mathrm{span}\{\mathsf{U}\}\} \cap \{u^* + \mathrm{span}\{\mathsf{U}^\perp\}\}\,.$$

Let $\mathsf{U} = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix}$ and $\mathsf{U}^\perp = \begin{bmatrix} u_{n+1} & \dots & u_m \end{bmatrix}$, where the column vector $u_i \in \mathbb{R}^m$ and $\|u_i\|_2 = 1$, $1 \le i \le n$. Then we have that

$$u_{\mathrm{f}} = u_0 + \sum_{i=1}^{n} \lambda_i u_i = u^* - \sum_{i=n+1}^{m} \lambda_i u_i\,,$$

making $\lambda_i = u_i^\top (u^* - u_0)$, which finalizes to

$$u_{\mathrm{f}} = u_0 + \sum_{i=1}^{n} u_i^\top (u^* - u_0) u_i = u^* - \sum_{i=n+1}^{m} u_i^\top (u^* - u_0) u_i\,.$$

Equivalently, $u_{\mathrm{f}} = u_0 - \mathsf{U}\mathsf{U}^\top (u_0 - u^*) = u^* - \mathsf{U}^\perp (\mathsf{U}^\perp)^\top (u^* - u_0)$. Then the result follows from the following identity:

$$\mathsf{U}\mathsf{U}^\top + \mathsf{U}^\perp (\mathsf{U}^\perp)^\top = \mathsf{I}\,,$$

which follows from $\mathsf{U}^\top \mathsf{U}^\perp = 0$ and $(\mathsf{U}^\perp)^\top \mathsf{U} = 0$. Furthermore, given y and $\tilde{y}$ in (4.4), and the fact that the row spaces for A and $\tilde{\mathsf{A}}$ are U and $\mathsf{U}^\perp$, respectively, we have $y_{\mathrm{f}} = \mathsf{A} u_{\mathrm{f}} = y^*$ and $\tilde{y}_{\mathrm{f}} = \tilde{\mathsf{A}} u_0$. ∎

The proof is rather straightforward. We quickly comment on its geometric interpretation. The result essentially is looking for the intersection of two sets. One is defined by the trajectory (A.1), and the other is defined by the equilibrium (4.5). The intersection point is unique, with its U component determined by u^* and the $\mathsf{U}^\perp$ component determined by the initial guess u_0. Note also that although u^* is not unique, its projection to the column space of U is unique. Indeed, since $\mathsf{A}u^* = y^*$, we have $\mathsf{U}\mathsf{U}^\top u^* = \mathsf{U}\mathsf{S}^{-1}\mathsf{V}^\top y^*$, which is purely determined by A and the given y^*.

Acknowledgments. We thank Prof. Youssef Marzouk, Prof. Levon Nurbekyan, Prof. Kui Ren, and Prof. Andrew Stuart for all the valuable discussions. We also thank the anonymous referees for their time and helpful suggestions.

REFERENCES

[1] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden, *Stochastic Interpolants: A Unifying Framework for Flows and Diffusions*, preprint, arXiv:2303.08797, 2023.

[2] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*, Springer Science & Business Media, 2005.

[3] A. Beck and M. Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Oper. Res. Lett., 31 (2003), pp. 167–175.

[4] E. Calvello, S. Reich, and A. M. Stuart, *Ensemble Kalman Methods: A Mean Field Perspective*, preprint, arXiv:2209.11371, 2022.

[5] J. A. Carrillo, K. Craig, and F. S. Patacchini, *A blob method for diffusion*, Calc. Var. Partial Differential Equations, 58 (2019), 53.

[6] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. R. Zhang, *Sampling is as easy as learning the score: Theory for diffusion models with minimal data assumptions*, in The Eleventh International Conference on Learning Representations, 2023.

[7] Y. Chen, D. Z. Huang, J. Huang, S. Reich, and A. M. Stuart, *Gradient Flows for Sampling: Mean-field Models, Gaussian Approximations and Affine Invariance*, preprint, arXiv:2302.11024, 2023.

[8] Y.-C. Chen, *A tutorial on kernel density estimation and recent advances*, Biostatist. Epidemiol., 1 (2017), pp. 161–187.

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

[10] A. Dalalyan, *Further and stronger analogy between sampling and optimization: Langevin Monte Carlo and gradient descent*, in Conference on Learning Theory, PMLR 65, 2017, pp. 678–689.

[11] A. Duncan, N. Nüsken, and L. Szpruch, *On the geometry of Stein variational gradient descent*, J. Mach. Learn. Res., 24 (2023), pp. 1–39.

[12] A. Garbuno-Inigo, F. Hoffmann, W. Li, and A. M. Stuart, *Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler*, SIAM J. Appl. Dyn. Syst., 19 (2020), pp. 412–441, https://doi.org/10.1137/19M1251655.

[13] M. Herty and G. Visconti, *Kinetic methods for inverse problems*, Kinet. Relat. Models, 12 (2019), pp. 1109–1130.

[14] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Math. Model. Theory Appl. 23, Springer Science & Business Media, 2008.

[15] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the Fokker–Planck equation*, SIAM J. Math. Anal., 29 (1998), pp. 1–17, https://doi.org/10.1137/S0036141096303359.

[16] J. W. Kim and S. Reich, *On Forward-Backward Sde Approaches to Continuous-Time Minimum Variance Estimation*, preprint, arXiv:2304.12727, 2023.

[17] H. Lee, J. Lu, and Y. Tan, *Convergence of score-based generative modeling for general data distributions*, in International Conference on Algorithmic Learning Theory, PMLR, 2023, pp. 946–985.

[18] R. Li, M. Tao, S. S. Vempala, and A. Wibisono, *The mirror Langevin algorithm converges with vanishing bias*, in International Conference on Algorithmic Learning Theory, PMLR, 2022, pp. 718–742.
[19] M. Lindsey, J. Weare, and A. Zhang, *Ensemble Markov chain Monte Carlo with teleporting walkers*, SIAM/ASA J. Uncertain. Quantif., 10 (2022), pp. 860–885, https://doi.org/10.1137/21M1425062.
[20] Q. Liu, *Stein variational gradient descent as gradient flow*, in Advances in Neural Information Processing Systems 30, 2017, pp. 3118–3126.
[21] Y. M. Marzouk, H. N. Najm, and L. A. Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, J. Comput. Phys., 224 (2007), pp. 560–586.
[22] R. J. McCann, *A convexity principle for interacting gases*, Adv. Math., 128 (1997), pp. 153–179.
[23] L. Nurbekyan, W. Lei, and Y. Yang, *Efficient natural gradient descent methods for large-scale PDE-based optimization problems*, SIAM J. Sci. Comput., 45 (2023), pp. A1621–A1655.
[24] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed., Springer Texts Statist., Springer, 1999.
[25] F. Santambrogio, *Optimal Transport for Applied Mathematicians. Calculus of Variations, PDEs, and Modeling*, Progr. Nonlinear Differential Equations Appl. 87, Birkhäuser/Springer, Cham, 2015.
[26] S. J. Sheather, *Density estimation*, Statist. Sci., 19 (2004), pp. 588–597.
[27] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Mongr. Statist. Appl. Probab., CRC Press, 1986.
[28] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, in Advances in Neural Information Processing Systems 32, 2019, pp. 11895–11907.
[29] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559.
[30] N. G. Trillos and D. Sanz-Alonso, *The Bayesian update: Variational formulations and gradient flows*, Bayesian Anal., 15 (2020), pp. 29–56.
[31] C. Villani, *Topics in Optimal Transportation*, Grad. Stud. Math. 58, American Mathematical Society, Providence, RI, 2021.
[32] L. Wang and M. Yan, *Hessian informed mirror descent*, J. Sci. Comput., 92 (2022), 90.
[33] S. Wang and Y. Marzouk, *On Minimax Density Estimation via Measure transport*, preprint, arXiv:2207.10231, 2022.
[34] M. Welling and Y. W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 681–688.
[35] Y. M. Marzouk, S. Reich, and A. Teckentrup, *Data assimilation—Mathematical foundation and applications*, Oberwolfach Rep., 19 (2022), pp. 489–515.
[36] A. Zellner, *Optimal information processing and Bayes's theorem*, Amer. Statist., 42 (1988), pp. 278–280.