SIAM J. SCI. COMPUT.
Vol. 45, No. 4, pp. A1621–A1655

EFFICIENT NATURAL GRADIENT DESCENT METHODS FOR LARGE-SCALE PDE-BASED OPTIMIZATION PROBLEMS*

LEVON NURBEKYAN†, WANZHOU LEI‡, AND YUNAN YANG§

Abstract. We propose efficient numerical schemes for implementing the natural gradient descent (NGD) for a broad range of metric spaces with applications to PDE-based optimization problems. Our technique represents the natural gradient direction as a solution to a standard least-squares problem. Hence, instead of calculating, storing, or inverting the information matrix directly, we apply efficient methods from numerical linear algebra. We treat both scenarios where the Jacobian, i.e., the derivative of the state variable with respect to the parameter, is either explicitly known or implicitly given through constraints. We can thus reliably compute several natural NGDs for a large-scale parameter space. In particular, we are able to compute Wasserstein NGD in thousands of dimensions, which was believed to be out of reach. Finally, our numerical results shed light on the qualitative differences between the standard gradient descent and various NGD methods based on different metric spaces in nonconvex optimization problems.

Key words. natural gradient, constrained optimization, least-squares method, gradient flow, inverse problem

MSC codes. 65K10, 49M15, 49M41, 90C26, 49Q22

DOI. 10.1137/22M1477805

1. Introduction.

In this paper, we are interested in solving optimization problems of the form

$$\inf_{\theta} f(\rho(\theta)), \tag{1.1}$$

where f is the objective/loss function and $\rho(\theta)$ is the state variable parameterized by θ. We mainly consider $\rho(\theta)$ as a PDE-based forward model, and f is a suitable discrepancy measure between the output of the forward model and the data. Inverse problems, such as the full waveform inversion (FWI), are classical examples of (1.1). More recent examples are machine learning–based PDE solvers where $\rho(\theta)$ is a neural network with weights θ that approximates the solution to the PDE [42]. They are typical large-scale optimization problems either due to fine grid parameterization of the unknown parameter or large networks employed to approximate the solutions.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 14, 2022; accepted for publication (in revised form) February 27, 2023; published electronically July 10, 2023.

https://doi.org/10.1137/22M1477805

Funding: The first author was partially supported by the AFOSR MURI FA 9550 18-1-0502 grant. The second author was partially supported by the 2021 Summer Undergraduate Research Experience (SURE) at the Department of Mathematics, Courant Institute of Mathematical Sciences, New York University. The third author was partially supported by NSF grant DMS-1913129. This work was done in part while the third author was visiting the Simons Institute for the Theory of Computing in Fall 2021. The third author also acknowledges support from Dr. Max Rössler, the Walter Haefner Foundation, and the ETH Zürich Foundation.

†Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 USA (lnurbek@math.ucla.edu).

‡Harvard University, Cambridge, MA 02138 USA (wanzhoulei@g.harvard.edu).

§Department of Mathematics, Cornell University, Ithaca, NY 14853 USA (yy837@cornell.edu).

First-order methods, especially in neural network training, are workhorses of high-dimensional optimization tasks. One such approach is the gradient descent (GD) method, whose continuous analogue is the following gradient flow equation:

$$\dot{\theta} = -\partial_\theta f(\rho(\theta)).$$

Although reasonably effective and computationally efficient, GD might suffer from local minima trapping, slow convergence, and sensitivity to hyperparameters. Consequently, first-order methods and some of their (stochastic and deterministic) variants are not robust and require a significant hyperparameter tuning on a problem-by-problem basis [51]. Such performance is often explained by the lack of curvature information in the parameter updates. Many optimization algorithms have been developed to improve the convergence speed, such as Newton-type methods [48], quasi-Newton methods [37], and various acceleration techniques [36] including momentum-based methods [41].

Recently, there has been a revival of second-order methods in the machine learning community [48]. Significant developments include the AdaHessian [51] and NGD [1, 31]. Both techniques incorporate curvature information into the parameter update. AdaHessian preconditions the gradient with an adaptive diagonal approximation to the Hessian [51]. The diagonal approximation is estimated by an adaption of Hutchinson's trace estimator [17]. Consequently, one obtains an optimization method for (1.1) with a similar observed convergence rate to Newton's method with a computational cost comparable to first-order methods. AdaHessian shows state-of-the-art performance across a range of machine learning tasks and is observed to be more robust and less sensitive to hyperparameter choices compared to several stochastic first-order methods [51].

A different approach is the natural gradient descent (NGD) method [1, 2, 38, 23, 24, 30, 31, 45], which preconditions the gradient with the *information matrix* instead of the Hessian; see (1.2). NGD performs the steepest descent with respect to the ρ-space, the *natural* manifold where $\rho(\theta)$ resides, instead of the parameter θ-space [1, 2]. A Riemannian structure is imposed on the parameterized subset $\{\rho(\theta)\}$ and then pulled back into the θ-space. NGD is sometimes also regarded as a generalized Gauss–Newton method [44, 38, 31], which has a faster convergence rate than GD. In particular, NGD can be interpreted as an approximate Netwon method when the manifold metric and the objective function f are compatible [31]. Other properties of NGD include local invariance with respect to the reparameterization, robustness with respect to hyperparameter choices, ability to progress with large step-sizes, and enforcing a state-dependent positive semidefinite preconditioning matrix. Inspired by the success of NGD in machine learning, we aim to extend and apply it to PDE-based optimization problems, which are mostly formulated in proper functional spaces with rich flexibility in choosing the metric.

Mathematically, continuous-time NGD is the preconditioned gradient flow

$$\dot{\theta} = -G(\theta)^{-1}\partial_\theta f(\rho(\theta)), \tag{1.2}$$

where $G(\theta)$ is the pull-back of a (formal) Riemannian metric in the ρ-space. It is often referred to as an information matrix and will be discussed in detail in section 2. There are two options to discretize (1.2): explicit and implicit. An explicit Euler discretization of (1.2) is

$$\theta^{l+1} = \theta^l - \tau^l G(\theta^l)^{-1}\partial_\theta f(\rho(\theta^l)), \quad l = 0, 1, \ldots, \tag{1.3}$$

where $\tau^l > 0$ is the step-size or learning rate. An implicit Euler discretization of (1.2) gives rise to

$$\theta^{l+1} = \operatorname*{argmin}_{\theta} \left\{ f(\rho(\theta)) + \frac{\langle G(\theta^l)(\theta - \theta^l), (\theta - \theta^l) \rangle}{2\tau^l} \right\}, \tag{1.4}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. If we denote by d_ρ the divergence or distance generating $G(\theta)$, the second term in (1.4) is the leading-order Taylor expansion of $\frac{1}{2\tau} d_\rho(\rho(\theta), \rho(\theta^l))^2$ at θ^l. Thus, the solution of (1.4) agrees with

$$\theta^{l+1} = \operatorname*{argmin}_{\theta} \left\{ f(\rho(\theta)) + \frac{d_\rho(\rho(\theta), \rho(\theta^l))^2}{2\tau^l} \right\}, \tag{1.5}$$

up to the first order. Note that (1.5) captures the underlying idea of the NGD: taking advantage of the geometric structure to find a direction with a maximum descent in the ρ-space. In contrast, finding a maximum descent in the θ-space as done by the "standard" implicit GD is

$$\theta^{l+1} = \operatorname*{argmin}_{\theta} \left\{ f(\rho(\theta)) + \frac{d_\theta(\theta, \theta^l)^2}{2\tau^l} \right\}, \tag{1.6}$$

where d_θ is the chosen metric for the θ-space. In this work, we focus on different d_ρ and consider d_θ as the Euclidean distance for simplicity. Intuitively, one may interpret it as a shift from the parametric θ-space to the more "natural" ρ-space. Thus, the infinitesimal decrease in the value of f and the direction of motion for ρ on $\mathcal{M}$ at $\rho = \rho(\theta)$ are invariant under reparameterizations [31].

NGD has been proven to be advantageous in various problems in machine learning and statistical inference, such as blind source separation [3], reinforcement learning [39], and neural network training [44, 33, 38, 32, 21, 31, 45, 25]. Further applications include solution methods for high-dimensional Fokker–Planck equations [22, 28]. Despite its success in statistical inferences and machine learning, the NGD method is far from being a mainstream computational technique, especially in PDE-based applications. A major obstacle is its computational complexity. In (1.3), explicit discretization of NGD reduces to preconditioning the standard gradient by the inverse of an often dense information matrix. The numerical computation is often intractable.

Existing works in the literature focused on explicit formulae [49], fast matrix-vector products [44, 33, 38, 31], and factorization techniques [32] for natural gradients generated by the Fisher–Rao metric in the ρ-space, where ρ is the output of feed-forward neural networks. These methods exploit the structural compatibility of standard loss functions and the Fisher metric by interpreting the Fisher NGD as a generalized Gauss–Newton or Hessian-free optimization [31, sec. 9.2]. The computational aspects of feed-forward neural networks are also utilized since computations through the forward and backward passes are recycled. Thus, to the best of our knowledge, the neural network community focuses on the Hessian approximation aspect in the context of feed-forward neural network models rather than the geometric properties of the forward-model-space. For the Wasserstein NGD (WNGD), [21, 7] rely on implicit Euler discretization, but their methods still suffer from accuracy issues due to the high dimensionality of the parameter space [45, sec. 2]. A regularized WNGD was considered in [45]. Unfortunately, by design, the method blows up when the regularization parameter decreases to zero, so it cannot compute the original WNGD. In [52], compactly supported wavelets were used to diagonalize the information matrix,

which is limited to the periodic setting with strictly positive $\rho(\theta)$ and also certain smoothness assumptions for $\rho(\theta)$.

There are three main contributions in our work. First, we depart from the Hessian approximation framework and adopt a more general *geometric formalism* of the NGD. Our approach applies to a general metric for the state space, which can be independent of the choice of the objective function. As examples, we treat Euclidean, Wasserstein, Sobolev, and Fisher–Rao natural gradients in a single framework for an arbitrary loss function. We focus on the standard least-squares formulation of the NGD direction. Second, we streamline the general NGD computation and develop two approaches to whether the forward model $\theta \mapsto \rho(\theta)$ is explicit or implicit. When the Jacobian $\partial_\theta \rho$ is analytically available, we utilize the (column-pivoting) QR decomposition for which a low-rank approximation can be directly applied if necessary [16]. When $\partial_\theta \rho$ is only implicitly available through the optimization constraints, we employ iterative solution procedures such as the conjugate gradient method [34] and utilize the adjoint-state method [40]. This second approach shares the same flavor with the method of the fast matrix-vector product for the Fisher–Rao NGD for neural network training [44, 33, 38, 31], but it allows one to apply the general NGD to large-scale optimization problems (see subsection 4.3, for example). In particular, our method can perform high-dimensional Wasserstein NGD, which was believed to be out of reach in the literature [45, sec. 1]. Last but not least, we use a few representative examples to demonstrate that the choice of metric in NGD matters as it not only quantitatively affects the convergence rate but also qualitatively determines which basin of attraction the iterates converge to.

The rest of the paper is organized as follows. In section 2, we first present the general mathematical formulations of the natural gradient based on a given metric space $(\mathcal{M}, g)$ and how it contrasts with the standard gradient. We then discuss a few common natural gradient examples and how they can all be reduced to a standard L^2-based minimization problem on the continuous level. In section 3, we demonstrate our general computational approaches under a unified framework that applies to any NGD method. The strategies concentrate on two scenarios regarding whether the Jacobian $\partial_\theta \rho$ is explicitly given or not, followed by section 4, where we apply the proposed numerical strategies for NGD methods to optimization problems under these two scenarios. Conclusions and further discussions follow in section 5.

2. Mathematical formulations of NGD. We begin by discussing the NGD method in an abstract setting before focusing on the common examples.

Assume that ρ is in a Riemannian manifold $(\mathcal{M}, g)$, and θ is in an open set $\Theta \subset \mathbb{R}^p$. Furthermore, assume that the correspondence $\theta \in \Theta \mapsto \rho(\theta) \in \mathcal{M}$ is smooth so that there exist tangent vectors

$$\left\{ \partial^g_{\theta_1} \rho(\theta), \partial^g_{\theta_2} \rho(\theta), \dots, \partial^g_{\theta_p} \rho(\theta) \right\} \subset T_\rho \mathcal{M}. \tag{2.1}$$

The superscript g in ∂^g highlights the dependence of tangent vectors on the choice of the Riemannian structure $(\mathcal{M}, g)$. Furthermore, assume that $f : \mathcal{M} \mapsto \mathbb{R}$ is a smooth function and denote by $\partial^g_\rho f \in T_\rho \mathcal{M}$ its metric gradient; that is, for all smooth curves $t \mapsto \rho(t)$, we have

$$\frac{df(\rho(t))}{dt} = \left\langle \partial^g_\rho f(\rho(t)), \partial^g_t \rho(t) \right\rangle_{g(\rho(t))}.$$

Tangent vectors $\{\partial^g_{\theta_i} \rho\}_{i=1}^p$ incorporate fundamental information on how $\rho(\theta)$ traverses $\mathcal{M}$ when θ traverses Θ. Indeed, an infinitesimal motion of θ along the coordinate θ_i-axis in Θ induces an infinitesimal motion of ρ along $\partial^g_{\theta_i} \rho$ in $\mathcal{M}$. More generally, if

$$\frac{d\theta}{dt} = \dot{\theta} = \eta = (\eta_1, \dots, \eta_p)^\top,$$

then

$$\partial_t^g \rho(\theta) = \eta_1 \partial_{\theta_1}^g \rho + \cdots + \eta_p \partial_{\theta_p}^g \rho.$$

Consequently, we have that

$$\frac{df(\rho(\theta))}{dt} = \left\langle \partial_\rho^g f, \partial_t^g \rho(\theta) \right\rangle_{g(\rho(\theta))} = \left\langle \partial_\rho^g f, \sum_{i=1}^p \eta_i \partial_{\theta_i}^g \rho \right\rangle_{g(\rho(\theta))}.$$

Intuitively, to achieve the largest descent in the loss $f(\rho(\theta))$, we want to choose $\eta = (\eta_1, \dots, \eta_p)^\top$ such that $\partial_\rho^g f$ is as *negatively* correlated with $\sum_{i=1}^p \eta_i \partial_{\theta_i}^g \rho$ as possible in terms of the given metric g. Thus, the NGD direction corresponds to the best approximation of $-\partial_\rho^g f$ by $\{\partial_{\theta_i}^g \rho\}$ in $T_\rho \mathcal{M}$:

$$\eta^{nat} = \operatorname*{argmin}_\eta \left\| \partial_\rho^g f + \sum_{i=1}^p \eta_i \partial_{\theta_i}^g \rho \right\|_{g(\rho(\theta))}^2. \tag{2.2}$$

In other words, the NGD corresponds to the evolution of θ that attempts to follow the *manifold GD of f on $\mathcal{M}$* as closely as possible. Since $(T_\rho \mathcal{M}, g)$ is an inner-product space where g may depend on ρ, and ρ depends on θ, (2.2) implies that under the natural gradient flow, the direction of motion for ρ on $\mathcal{M}$ is given by the $g(\rho(\theta))$-orthogonal projection of $-\partial_\rho^g f$ onto $\operatorname{span}\{\partial_{\theta_1}^g \rho, \dots, \partial_{\theta_p}^g \rho\}$:

$$\partial_t^g \rho = \sum_{i=1}^p \eta_i^{nat}\ \partial_{\theta_i}^g \rho =: P \partial_\rho^g f. \tag{2.3}$$

Since $\operatorname{span}\{\partial_{\theta_1}^g \rho, \dots, \partial_{\theta_p}^g \rho\}$ is invariant under smooth changes of coordinates $\theta = \theta(\psi)$, we obtain that (2.3) is also invariant under such transformations. Additionally, the infinitesimal decay of the loss function is also invariant under smooth changes in the coordinates. Indeed,

$$\frac{df(\rho(\theta))}{dt} = -\|P \partial_\rho^g f\|_{g(\rho(\theta))}^2.$$

A critical benefit of these invariance properties is mitigating potential negative effects of a poor choice of parameterization by filtering them out (since the corresponding decrease in the loss function is parameter-invariant) and reaching $\operatorname{argmin}_{\rho \in \mathcal{M}} f(\rho)$ as quickly and as closely as possible. For the analysis of NGD based on this insight, we refer the reader to [31, 28] for more details.

Remark 2.1. When $\{\partial_{\theta_i}^g \rho\}$ are linearly dependent, the η^{nat} in (2.2) is not unique, and we pick the one with the minimal length for computational purposes; that is, we replace $G^{-1}(\theta)$ by the Moore–Penrose pseudoinverse $G(\theta)^\dagger$ in (1.2) and elsewhere. It is worth noting that this choice is crucial to guarantee convergence and generalization properties of the NGD method in some applications; see [54] for example. Alternatively, one may consider a damping variant of G; see subsection 3.5.1.

To compare the natural gradient with the standard gradient $\partial_\theta f(\rho(\theta))$, first note that

$$\frac{df(\rho(\theta))}{dt} = \left\langle \partial_\rho^g f, \sum_{i=1}^p \eta_i \partial_{\theta_i}^g \rho \right\rangle_{g(\rho(\theta))} = \sum_{i=1}^p \left\langle \partial_\rho^g f, \partial_{\theta_i}^g \rho \right\rangle_{g(\rho(\theta))} \eta_i = \partial_\theta f(\rho(\theta)) \cdot \eta.$$

Therefore, in a form similar to (2.2), the GD direction is the solution to

$$\eta^{std} = \operatorname*{argmin}_{\eta} \|\partial_\theta f(\rho(\theta)) + \eta\|^2.$$

In other words, GD is the steepest descent in the θ-space, whereas NGD is an approximation of the steepest descent in the ρ-space based on a given metric g. Furthermore, GD leads to

$$\partial_t^g \rho = \sum_{i=1}^p \eta_i^{std}\ \partial_{\theta_i}^g \rho = -\sum_{i=1}^p \left\langle \partial_\rho^g f, \partial_{\theta_i}^g \rho \right\rangle_{g(\rho(\theta))} \partial_{\theta_i}^g \rho,$$
$$\frac{df(\rho(\theta))}{dt} = -\|\partial_\theta f(\rho(\theta))\|_2^2 = -\sum_{i=1}^p \left| \left\langle \partial_\rho^g f, \partial_{\theta_i}^g \rho \right\rangle_{g(\rho(\theta))} \right|^2,$$

which are not necessarily invariant under coordinate transformations.

When $\{\partial_{\theta_i}^g \rho\}$ are linearly independent, we obtain that

$$\eta^{nat} = -G(\theta)^{-1} \partial_\theta f(\rho(\theta)) = G(\theta)^{-1} \eta^{std}, \tag{2.4}$$

where $G(\theta)$ is the *information matrix* whose (i,j)th entry is

$$G_{ij}(\theta) = \left\langle \partial_{\theta_i}^g \rho, \partial_{\theta_j}^g \rho \right\rangle_{g(\rho(\theta))}, \quad i,j = 1,\dots,p. \tag{2.5}$$

Thus, an NGD direction is a GD direction preconditioned by the inverse of the information matrix.

Since the information matrix $G(\theta)$ is often dense and can be ill-conditioned, direct application of (2.4) is prohibitively costly for high-dimensional parameter space, that is, large p. Our goal is to calculate η^{nat} via the least-squares formulation (2.2), circumventing the computational costs from assembling and inverting the dense matrix G directly.

2.1. L^2 natural gradient. In this subsection, we embed ρ in the metric space $(\mathcal{M}, g) = (L^2(\mathbb{R}^d), \langle\cdot,\cdot\rangle_{L^2(\mathbb{R}^d)})$. In this case, the tangent space $T_\rho\mathcal{M} = L^2(\mathbb{R}^d)$ for any $\rho \in \mathcal{M}$, and

$$\left\langle \zeta, \hat{\zeta} \right\rangle_{g(\rho)} = \int_{\mathbb{R}^d} \zeta(x)\hat{\zeta}(x)dx \quad \forall \zeta, \hat{\zeta} \in T_\rho\mathcal{M}.$$

The linear structure of $L^2(\mathbb{R}^d)$ is advantageous for developing differential calculus, and many finite-dimensional concepts generalize naturally. Indeed, the tangent vectors (2.1) for a smooth mapping $\theta \in \Theta \mapsto \rho(\theta,\cdot) \in L^2(\mathbb{R}^d)$ are $\{\zeta_1, \zeta_2, \dots, \zeta_p\}$ given by

$$\zeta_i(x) = \partial_{\theta_i}\rho(\theta, x), \quad i = 1,\dots,p. \tag{2.6}$$

The information matrix in (2.5) is given by

$$G_{ij}^{L^2}(\theta) = \int_{\mathbb{R}^d} \partial_{\theta_i}\rho(\theta,x)\partial_{\theta_j}\rho(\theta,x)dx, \quad i,j = 1,2,\dots,p.$$

Next, for $f : L^2(\mathbb{R}^d) \mapsto \mathbb{R}$, we obtain that the L^2-derivative at ρ is $\partial_\rho f(\rho) \in L^2(\mathbb{R}^d)$ such that

$$\lim_{t\to 0} \frac{f(\rho + t\zeta) - f(\rho)}{t} = \int_{\mathbb{R}^d} \partial_\rho f(\rho)(x)\ \zeta(x)dx \quad \forall \zeta \in L^2(\mathbb{R}^d). \tag{2.7}$$

Thus, $\partial_\rho f$ is the commonly known derivative in the sense of calculus of variations. Finally, for smooth $\rho:\Theta\to L^2(\mathbb{R}^d)$ and $f:L^2(\mathbb{R}^d)\to\mathbb{R}$, formula (2.2) leads to the L^2 natural gradient

$$\eta_{L^2}^{nat}=\operatorname*{argmin}_{\eta\in\mathbb{R}^p}\left\|\partial_\rho f+\sum_{i=1}^p\eta_i\zeta_i\right\|^2_{L^2(\mathbb{R}^d)}. \tag{2.8}$$

The L^2 metric is not a typical choice for the NGD. Nevertheless, this metric is important as a basis for computing more complex NGDs. Additionally, see section 2.6 for the connection between L^2-based NGD and the Gauss–Newton method.

2.2. H^s natural gradient. In this subsection, we assume that ρ is embedded in the L^2-based Sobolev space $H^s(\mathbb{R}^d)$ for $s\in\mathbb{Z}$ (we return to the L^2 case if $s=0$). The metric space $(\mathcal{M},g)=(H^s(\mathbb{R}^d),\langle\cdot,\cdot\rangle_{H^s(\mathbb{R}^d)})$. Since this is also a Hilbert space, $T_\rho\mathcal{M}=H^s(\mathbb{R}^d)$ for all $\rho\in\mathcal{M}$, and

$$\langle\zeta,\hat\zeta\rangle_{g(\rho)}=\langle\zeta,\hat\zeta\rangle_{H^s(\mathbb{R}^d)}=\begin{cases}\int_{\mathbb{R}^d}\mathbf{D}^s\zeta\cdot\mathbf{D}^s\hat\zeta\,dx, & s\ge0,\\ \int_{\mathbb{R}^d}\mathbf{D}^{-s}\chi\cdot\mathbf{D}^{-s}\hat\chi\,dx, & s<0,\end{cases}\quad\zeta,\hat\zeta\in T_\rho\mathcal{M},$$

where $\mathbf{D}^s$ is the linear operator whose output is the vector of all the partial derivatives up to order s for $s\ge0$. For $s<0$, we define $\chi=((\mathbf{D}^{-s})^*\mathbf{D}^{-s})^{-1}\zeta$ and $\hat\chi=((\mathbf{D}^{-s})^*\mathbf{D}^{-s})^{-1}\hat\zeta$. For example, $(\mathbf{D}^{-s})^*\mathbf{D}^{-s}=I-\triangle$ if $s=-1$ and $I-\triangle+\triangle^2$ if $s=-2$ [50]. Note that $\mathbf{D}^{-s}((\mathbf{D}^{-s})^*\mathbf{D}^{-s})^{-1}=((\mathbf{D}^{-s})^*)^\dagger$ for $s<0$, where † is the notation for pseudoinverse. Thus, we can rewrite

$$\langle\zeta,\hat\zeta\rangle_{H^s(\mathbb{R}^d)}=\langle\mathbf{D}^{-s}\chi,\mathbf{D}^{-s}\hat\chi\rangle_{L^2(\mathbb{R}^d)}=\left\langle((\mathbf{D}^{-s})^*)^\dagger\zeta,((\mathbf{D}^{-s})^*)^\dagger\hat\zeta\right\rangle_{L^2(\Omega)}\quad\forall\zeta,\hat\zeta\in T_\rho\mathcal{M}.$$

For a smooth $\rho:\Theta\to H^s(\mathbb{R}^d)$, the tangent vectors are still $\{\zeta_i\}$ in (2.6) but now are considered as elements of $H^s(\mathbb{R}^d)$. This means that the information matrix $G^{H^s}(\theta)$ defined in (2.5) is given by

$$\begin{aligned}G_{ij}^{H^s}(\theta)&=\langle\partial_{\theta_i}\rho,\partial_{\theta_j}\rho\rangle_{H^s(\mathbb{R}^d)}\\&=\begin{cases}\int_{\mathbb{R}^d}\mathbf{D}^s\partial_{\theta_i}\rho(\theta,x)\cdot\mathbf{D}^s\partial_{\theta_j}\rho(\theta,x)\,dx, & s\ge0,\\ \int_{\mathbb{R}^d}((\mathbf{D}^{-s})^*)^\dagger\partial_{\theta_i}\rho(\theta,x)\cdot((\mathbf{D}^{-s})^*)^\dagger\partial_{\theta_j}\rho(\theta,x)\,dx, & s<0,\end{cases}\end{aligned}$$

for $i,j=1,\dots,p$. Note that G^{H^s} is different from G^{L^2} due to the inner product.

Next, we calculate the H^s gradient of smooth $f:H^s(\mathbb{R}^d)\to\mathbb{R}$. For $s\ge0$, we have that

$$\begin{aligned}\lim_{t\to0}\frac{f(\rho+t\zeta)-f(\rho)}{t}=\langle\partial_\rho^{H^s}f,\zeta\rangle_{H^s(\mathbb{R}^d)}&=\int_{\mathbb{R}^d}\mathbf{D}^s\partial_\rho^{H^s}f\cdot\mathbf{D}^s\zeta\;dx\\&=\int_{\mathbb{R}^d}(\mathbf{D}^s)^*\mathbf{D}^s\;\partial_\rho^{H^s}f\,\zeta\;dx,\end{aligned}$$

and so from (2.7) we obtain

$$\partial_\rho^{H^s}f=((\mathbf{D}^s)^*\mathbf{D}^s)^{-1}\partial_\rho f,\quad s\ge0.$$

When $s<0$, under analogous assumptions with the case $s\ge0$, we have that

$$\begin{aligned}&\lim_{t\to0}\frac{f(\rho+t\zeta)-f(\rho)}{t}\\&\quad=\langle\partial_\rho^{H^s}f,\zeta\rangle_{H^s(\mathbb{R}^d)}=\int_{\mathbb{R}^d}((\mathbf{D}^{-s})^*)^\dagger\partial_\rho^{H^s}f\cdot((\mathbf{D}^{-s})^*)^\dagger\zeta dx\\&\quad=\int_{\mathbb{R}^d}(\mathbf{D}^{-s})^\dagger((\mathbf{D}^{-s})^*)^\dagger\partial_\rho^{H^s}f\cdot\zeta dx=\int_{\mathbb{R}^d}((\mathbf{D}^{-s})^*\mathbf{D}^{-s})^\dagger\partial_\rho^{H^s}f\;\zeta dx.\end{aligned}$$

Thus, from (2.7), we have

$$\partial_\rho^{H^s} f = (\mathbf{D}^{-s})^* \mathbf{D}^{-s} \partial_\rho f, \quad s < 0.$$

Finally, for smooth $\rho : \Theta \to H^s(\mathbb{R}^d)$ and $f : H^s(\mathbb{R}^d) \to \mathbb{R}$, (2.2) leads to the H^s natural gradient

$$\eta_{H^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| \partial_\rho^{H^s} f + \sum_{i=1}^p \eta_i \zeta_i \right\|_{H^s(\mathbb{R}^d)}^2. \tag{2.9}$$

For numerical implementation, we reduce this previous formulation into a least-squares problem in $L^2(\mathbb{R}^d)$. More specifically, for $s \geq 0$, (2.9) can be written as

$$\eta_{H^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| \mathbf{D}^s ((\mathbf{D}^s)^* \mathbf{D}^s)^{-1} \partial_\rho f + \sum_{i=1}^p \eta_i \, \mathbf{D}^s \zeta_i \right\|_{L^2(\mathbb{R}^d)}^2.$$

Furthermore, for $s < 0$ we have that (2.9) can be written as

$$\eta_{H^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| \mathbf{D}^{-s} \partial_\rho f + \sum_{i=1}^p \eta_i \, ((\mathbf{D}^{-s})^*)^\dagger \zeta_i \right\|_{L^2(\mathbb{R}^d)}^2.$$

Both cases share the same form (2.10) with $\mathbf{L} = \mathbf{D}^s$ for $s \geq 0$ and $\mathbf{L} = ((\mathbf{D}^{-s})^*)^\dagger$ for $s < 0$:

$$\eta_{H^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| (\mathbf{L}^*)^\dagger \partial_\rho f + \sum_{i=1}^p \eta_i \, \mathbf{L} \zeta_i \right\|_{L^2(\mathbb{R}^d)}^2. \tag{2.10}$$

2.3. $\dot{H}^s$ natural gradient. Next, we consider the NGD with respect to the Sobolev seminorm $\dot{H}^s$. For simplicity, we assume that ρ is supported in a smooth bounded domain $\Omega \subset \mathbb{R}^d$. For $s > 0$, we define the space $\dot{H}^s(\Omega) = \{\zeta \in H^s(\Omega) : \int_\Omega \zeta = 0\}$ with the inner product

$$\langle \zeta, \hat{\zeta} \rangle_{\dot{H}^s(\Omega)} = \langle \widetilde{\mathbf{D}}^s \zeta, \widetilde{\mathbf{D}}^s \hat{\zeta} \rangle_{L^2(\Omega)} = \int_\Omega \widetilde{\mathbf{D}}^s \zeta \cdot \widetilde{\mathbf{D}}^s \hat{\zeta} dx \quad \forall \zeta, \hat{\zeta} \in \dot{H}^s(\Omega),$$

where $\widetilde{\mathbf{D}}^s$ is the linear operator whose output is the vector of all partial derivatives of positive order up to s. To consider the $\dot{H}^s$ natural gradient flows, we embed ρ in $(\mathcal{M}, g)$, where

$$\mathcal{M} = \left\{ \rho \in H^s(\Omega) : \int_\Omega \rho = 1 \right\}, \quad T_\rho \mathcal{M} = \dot{H}^s(\Omega),$$
$$\langle \zeta, \hat{\zeta} \rangle_{g(\rho)} = \langle \zeta, \hat{\zeta} \rangle_{\dot{H}^s(\Omega)} \quad \forall \zeta, \hat{\zeta} \in T_\rho \mathcal{M}.$$

For a smooth $\rho : \Theta \to \mathcal{M}$, we still have that the tangent vectors are $\{\zeta_i\}$ as defined in (2.6). Since $\int_\Omega \rho(\theta, x) dx = 1$ for all $\theta \in \Theta$, we have that

$$\int_\Omega \zeta_i(x) dx = \int_\Omega \partial_{\theta_i} \rho(\theta, x) dx = \partial_{\theta_i} \int_\Omega \rho(\theta, x) dx = 0, \quad i = 1, \ldots, p,$$

and thus $\{\zeta_i\} \subset T_\rho \mathcal{M}$. The information matrix (2.5) for this case is $G^{\dot{H}^s}(\theta)$ given by

$$G_{ij}^{\dot{H}^s}(\theta) = \langle \partial_{\theta_i} \rho, \partial_{\theta_j} \rho \rangle_{\dot{H}^s(\Omega)} = \int_\Omega \widetilde{\mathbf{D}}^s \partial_{\theta_i} \rho(\theta, x) \cdot \widetilde{\mathbf{D}}^s \partial_{\theta_j} \rho(\theta, x) dx, \quad i, j = 1, \ldots, p.$$

On the other hand, for $f:\mathcal{M}\to\mathbb{R}$, we have that $\partial_\rho^{\dot{H}^s} f\in\dot{H}^s(\Omega)$ where $\forall\zeta\in T_\rho\mathcal{M}$,

$$\lim_{t\to 0}\frac{f(\rho+t\zeta)-f(\rho)}{t}=\langle\partial_\rho^{\dot{H}^s} f,\zeta\rangle_{\dot{H}^s(\Omega)}=\int_\Omega \widetilde{\mathbf{D}}^s\partial_\rho^{\dot{H}^s} f\cdot\widetilde{\mathbf{D}}^s\zeta dx=\int_\Omega(\widetilde{\mathbf{D}}^s)^*\widetilde{\mathbf{D}}^s\partial_\rho^{\dot{H}^s} f\,\zeta dx.$$

The adjoint $(\widetilde{\mathbf{D}}^s)^*$ is taken with respect to the $L^2(\Omega)$ inner product. Hence, based on (2.7),

$$\int_\Omega\left(\partial_\rho f-(\widetilde{\mathbf{D}}^s)^*\widetilde{\mathbf{D}}^s\partial_\rho^{\dot{H}^s} f\right)\zeta dx=0\quad\forall\zeta\in T_\rho\mathcal{M}. \tag{2.11}$$

Furthermore, denote by $\mathbf{1}$ the constant function that is equal to 1 on Ω. We then have that

$$T_\rho\mathcal{M}=\operatorname{span}\{\mathbf{1}\}^\perp=\ker(\widetilde{\mathbf{D}}^s)^\perp=\operatorname{Im}((\widetilde{\mathbf{D}}^s)^*),$$

where $^\perp$ is again taken with respect to the $L^2(\Omega)$ inner product. Hence, using the properties of adjoint operators, we obtain

$$\partial_\rho^{\dot{H}^s} f=\left((\widetilde{\mathbf{D}}^s)^*\widetilde{\mathbf{D}}^s\right)^\dagger\partial_\rho f,\quad s>0.$$

Next, we discuss the case $s<0$. As the dual space of $\dot{H}^{-s}(\Omega)$, the space $\dot{H}^s(\Omega)$ is equipped with the dual norm

$$\|\zeta\|_{\dot{H}^s(\Omega)}=\sup\left\{\langle\zeta,\phi\rangle:\ \|\phi\|_{\dot{H}^{-s}(\Omega)}\le 1\right\}.$$

Using the Poincaré inequality and the Riesz representation theorem, we obtain that for every $\zeta\in\operatorname{span}\{\mathbf{1}\}^\perp$, the map $\phi\mapsto\int_\Omega\zeta\phi$ is a continuous linear operator on $\dot{H}^{-s}(\Omega)$, and there exists a unique $\chi\in\dot{H}^{-s}(\Omega)$ such that

$$\int_\Omega\zeta\,\phi\,dx=\int_\Omega\widetilde{\mathbf{D}}^{-s}\chi\,\widetilde{\mathbf{D}}^{-s}\phi\,dx\quad\forall\phi\in\dot{H}^{-s}(\Omega).$$

Hence, $\zeta=(\widetilde{\mathbf{D}}^{-s})^*\widetilde{\mathbf{D}}^{-s}\chi$ together with the homogeneous Neumann boundary condition. Therefore,

$$\|\zeta\|_{\dot{H}^s(\Omega)}=\|\widetilde{\mathbf{D}}^{-s}\chi\|_{L^2}=\|\chi\|_{\dot{H}^{-s}(\Omega)}.$$

Using similar arguments for the $s>0$ case, we obtain that

$$\begin{aligned}\langle\zeta,\hat{\zeta}\rangle_{\dot{H}^s(\Omega)}&=\langle\widetilde{\mathbf{D}}^{-s}\chi,\widetilde{\mathbf{D}}^{-s}\hat{\chi}\rangle_{L^2(\Omega)}\\&=\left\langle((\widetilde{\mathbf{D}}^{-s})^*)^\dagger\zeta,((\widetilde{\mathbf{D}}^{-s})^*)^\dagger\hat{\zeta}\right\rangle_{L^2(\Omega)}\quad\forall\zeta,\hat{\zeta}\in\operatorname{span}\{\mathbf{1}\}^\perp.\end{aligned}$$

For more details on $\dot{H}^s(\Omega)$ where $s<0$, we refer the reader to [4, Lecture 13].

Next, we embed ρ in space $\mathcal{M}=\{\rho\in L^2(\Omega):\int_\Omega\rho=1\}$ with $T_\rho\mathcal{M}=\operatorname{span}\{\mathbf{1}\}^\perp$ and

$$\langle\zeta,\hat{\zeta}\rangle_{g(\rho)}=\left\langle((\widetilde{\mathbf{D}}^{-s})^*)^\dagger\zeta,((\widetilde{\mathbf{D}}^{-s})^*)^\dagger\hat{\zeta}\right\rangle_{L^2(\Omega)}\quad\forall\zeta,\hat{\zeta}\in T_\rho\mathcal{M}.$$

Furthermore, for a smooth function $f:\mathcal{M}\to\mathbb{R}$, we have that

$$\lim_{t\to 0}\frac{f(\rho+t\zeta)-f(\rho)}{t}=\langle\partial_\rho^{\dot{H}^s} f,\zeta\rangle_{\dot{H}^s(\Omega)}=\int_\Omega((\widetilde{\mathbf{D}}^{-s})^*\widetilde{\mathbf{D}}^{-s})^\dagger\partial_\rho^{\dot{H}^s} f\,\zeta dx.$$

Together with (2.7), we have

$$\int_\Omega \left(\partial_\rho f - ((\widetilde{\mathbf{D}}^{-s})^* \widetilde{\mathbf{D}}^{-s})^\dagger \partial_\rho^{\dot{H}^s} f\right) \zeta dx = 0 \quad \forall \zeta \in T_\rho \mathcal{M}.$$

After performing analysis similar to the $s > 0$ case, we obtain that

$$\partial_\rho^{\dot{H}^s} f = (\widetilde{\mathbf{D}}^{-s})^* \widetilde{\mathbf{D}}^{-s} \partial_\rho f, \quad s < 0.$$

Finally, for both $s > 0$ and $s < 0$ cases, (2.2) leads to the $\dot{H}^s$ natural gradient

$$\eta_{\dot{H}^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| \partial_\rho^{\dot{H}^s} f + \sum_{i=1}^p \eta_i \zeta_i \right\|^2_{\dot{H}^s(\mathbb{R}^d)} \tag{2.12}$$

for smooth $\rho : \Theta \to \mathcal{M}$ and $f : \mathcal{M} \to \mathbb{R}$. As before, we can rewrite (2.12) as a least-squares problem

$$\eta_{\dot{H}^s}^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| (\mathbf{L}^*)^\dagger \partial_\rho f + \sum_{i=1}^p \eta_i \mathbf{L} \zeta_i \right\|^2_{L^2(\Omega)}, \quad \mathbf{L} = \begin{cases} \widetilde{\mathbf{D}}^s, & s > 0, \\ ((\widetilde{\mathbf{D}}^{-s})^*)^\dagger, & s < 0. \end{cases} \tag{2.13}$$

Note that (2.13) shares the same form with (2.10).

H^s and $\dot{H}^s$ natural gradients proved extremely useful for obtaining fast algorithms for solving the optimal transportation problem and related problems [20, 19, 18]. The authors in these papers do not use the natural gradient descent formalism, but their methods are indeed Sobolev NGDs.

2.4. Fisher–Rao–Hellinger natural gradient. Here, we assume that ρ is a strictly positive probability density function. We embed ρ in $(\mathcal{M}, g) = (L^1(\mathbb{R}^d), g)$ where $T_\rho(\mathcal{M}) = L^2_{\rho^{-1}}(\mathbb{R}^d)$ and

$$\langle \zeta, \hat{\zeta} \rangle_{g(\rho)} = \int_{\mathbb{R}^d} \frac{\zeta(x)\hat{\zeta}(x)}{\rho(x)} dx \quad \forall \zeta, \hat{\zeta} \in T_\rho \mathcal{M}.$$

This Riemannian metric is called the Fisher–Rao metric, and the distance induced by this metric is the Hellinger distance: $d_H(\rho_1, \rho_2) \propto \|\sqrt{\rho_1} - \sqrt{\rho_2}\|_{L^2(\mathbb{R}^d)}$. Next, we will derive the natural gradient flow based on the Fisher–Rao metric, first introduced by Amari in [2].

For a smooth $\rho : \Theta \to \mathcal{M}$, we have that the tangent vectors are $\{\zeta_i\}$ in (2.6) but now considered as elements of $L^2_{\rho^{-1}}(\mathbb{R}^d)$. Therefore, the information matrix in (2.5) becomes $G^{FR}(\theta) \in \mathbb{R}^{p \times p}$ where

$$G^{FR}_{ij}(\theta) = \int_{\mathbb{R}^d} \frac{\partial_{\theta_i} \rho(\theta, x) \partial_{\theta_j} \rho(\theta, x)}{\rho(\theta, x)} dx, \quad i, j = 1, 2, \ldots, p.$$

As before, $G^{FR}(\theta)$ is in general different from $G^{L^2}(\theta)$, $G^{H^s}(\theta)$, and $G^{\dot{H}^s}(\theta)$.

Furthermore, for a smooth function $f : \mathcal{M} \to \mathbb{R}$, we have that

$$\lim_{t \to 0} \frac{f(\rho + t\zeta) - f(\rho)}{t} = \int_{\mathbb{R}^d} \frac{\partial_\rho^{FR} f \, \zeta}{\rho} dx,$$

and so from (2.7) we obtain

$$\partial_\rho^{FR} f = \rho \, \partial_\rho f.$$

Finally, for smooth $\rho:\Theta\to\mathcal{M}$ and $f:\mathcal{M}\to\mathbb{R}$, (2.2) leads to the Fisher–Rao natural gradient

$$\eta_{FR}^{nat}=\operatorname*{argmin}_{\eta\in\mathbb{R}^p}\left\|\partial_\rho^{FR}f+\sum_{i=1}^p\eta_i\zeta_i\right\|^2_{L^2_{\rho^{-1}}(\mathbb{R}^d)}. \tag{2.14}$$

The L^2 least-squares formulation is

$$\eta_{FR}^{nat}=\operatorname*{argmin}_{\eta\in\mathbb{R}^p}\left\|\frac{\partial_\rho^{FR}f}{\sqrt{\rho}}+\sum_{i=1}^p\eta_i\frac{\zeta_i}{\sqrt{\rho}}\right\|^2_{L^2(\mathbb{R}^d)}=\operatorname*{argmin}_{\eta\in\mathbb{R}^p}\left\|(\mathbf{L}^*)^\dagger\partial_\rho f+\sum_{i=1}^p\eta_i\,\mathbf{L}\zeta_i\right\|^2_{L^2(\mathbb{R}^d)}, \tag{2.15}$$

where $\mathbf{L}\zeta=\frac{1}{\sqrt{\rho}}\zeta$ and $(\mathbf{L}^*)^\dagger\partial_\rho f=\sqrt{\rho}\;\partial_\rho f$.

2.5. W_2 natural gradient. We first revisit the WNGD method [23]. Denoting by $\mathcal{P}(\mathbb{R}^d)$ the set of Borel probability measures on $\mathbb{R}^d$, we first introduce the Wasserstein metric on the space $\mathcal{P}(\mathbb{R}^d)$. Furthermore, for $\rho\in\mathcal{P}(\mathbb{R}^d)$ and a measurable function $f:\mathbb{R}^d\to\mathbb{R}^n$, we denote by $f_\sharp\rho\in\mathcal{P}(\mathbb{R}^n)$ the probability measure defined by

$$(f_\sharp\rho)(B)=\rho(f^{-1}(B))\quad\forall B\subset\mathbb{R}^n\text{ Borel}$$

and call it the pushforward of ρ under f. Next, for any $\rho_1,\rho_2\in\mathcal{P}(\mathbb{R}^d)$, we denote $\Gamma(\rho_1,\rho_2)$ as the set of all possible joint measures $\pi\in\mathcal{P}(\mathbb{R}^{2d})$ such that

$$\int_{\mathbb{R}^{2d}}(\phi(x)+\psi(y))\,d\pi(x,y)=\int_{\mathbb{R}^d}\phi(x)d\rho_1(x)+\int_{\mathbb{R}^d}\psi(y)d\rho_2(y)$$

for all $(\phi,\psi)\in L^1(\rho_1)\times L^1(\rho_2)$. The 2-Wasserstein distance is defined as

$$W_2(\rho_1,\rho_2)=\left(\inf_{\pi\in\Gamma(\rho_1,\rho_2)}\int_{\mathbb{R}^{2d}}|x-y|^2d\pi(x,y)\right)^{\frac{1}{2}}.$$

Denoting by $\mathcal{P}_2(\mathbb{R}^d)$ the set of Borel probability measures with finite second moments, we have that $(\mathcal{P}_2(\mathbb{R}^d),W_2)$ is a complete separable metric space; see more details in [46, Chapters 7] and [5, Chapters 7]. More intriguingly, one can build a Riemannian structure on $(\mathcal{P}_2(\mathbb{R}^d),W_2)$. Our discussion is formal, and we refer the reader to [46, Chapter 8] and [5, Chapter 8] for rigorous treatments.

In short, tangent vectors in $(\mathcal{P}_2(\mathbb{R}^d),W_2)$ are the infinitesimal spatial displacements of minimal kinetic energy. More specifically, for a given $\rho\in\mathcal{P}_2(\mathbb{R}^d)$, we define the tangent space, $T_\rho\mathcal{P}_2(\mathbb{R}^d)$, as a set of all maps $v\in L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)$ such that

$$\|v+w\|_{L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)}\geq\|v\|_{L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)}\quad\forall w\in L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)\quad\text{s.t.}\quad\nabla\cdot(w\rho)=0, \tag{2.16}$$

where $L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)$ denotes the ρ-weighted L^2 space. When $\rho=1$, it reduces to the standard L^2. The divergence equation above is understood in the sense of distributions; that is,

$$\int_{\mathbb{R}^d}\nabla\phi(x)\cdot w(x)\;\rho(x)dx=0\quad\forall\phi\in C_c^\infty(\mathbb{R}^d).$$

If we think of ρ as a fluid density, then an infinitesimal displacement $\frac{dx}{dt}=\dot{x}=v(x)$ leads to an infinitesimal density change given by the continuity equation

$$\frac{\partial\rho}{\partial t}=-\nabla\cdot(v\rho). \tag{2.17}$$

Therefore, for a given w such that $\nabla \cdot (w\rho) = 0$, we have that both $\dot{x} = v(x)$ and $\dot{x} = v(x) + w(x)$ lead to the same continuity equation (2.17). Therefore, the evolution of the density is insensitive to the divergence-free vector fields, and we project them out leaving only a unique vector field with the minimal kinetic energy. The kinetic energy of a vector field v is then defined as

$$\|v\|^2_{L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)} = \int_{\mathbb{R}^d} |v(x)|^2 \rho(x) dx.$$

For a given evolution $t \mapsto \rho(t, \cdot)$, such a "distilled" vector field v is unique and incorporates critical geometric information on the spatial evolution of ρ.

Next, we define a Riemannian metric by

$$\langle v, \hat{v} \rangle_{g(\rho)} = \int_{\mathbb{R}^d} v(x) \cdot \hat{v}(x)\ \rho(x) dx, \quad v, \hat{v} \in T_\rho \mathcal{P}_2(\mathbb{R}^d).$$

Furthermore, a mapping $\theta \in \Theta \mapsto \rho(\theta, \cdot) \in \mathcal{P}(\mathbb{R}^d)$ is differentiable if for every $\theta \in \Theta$, there exists a set of bases $\{v_i(\theta)\} \subset T_\rho \mathcal{P}_2(\mathbb{R}^d)$ such that

$$\lim_{t \to 0} \frac{W_2\left(\rho(\theta + t\eta), \left(I + t \sum_{i=1}^p \eta_i v_i(\theta)\right) \sharp \rho(\theta)\right)}{t} = 0 \quad \forall \eta \in \mathbb{R}^p, \tag{2.18}$$

where I is the identity map. Thus,

$$\left\{v_1, v_2, \ldots, v_p\right\} = \left\{\partial^W_{\theta_1} \rho, \partial^W_{\theta_2} \rho, \ldots, \partial^W_{\theta_p} \rho\right\} \tag{2.19}$$

are the tangent vectors in (2.1) for the W_2 metric. Thus, the information matrix in (2.5) becomes $G^W(\theta) \in \mathbb{R}^{p \times p}$, where

$$G^W_{ij}(\theta) = \int_{\mathbb{R}^d} v_i(x) \cdot v_j(x)\ \rho(x) dx, \quad i, j = 1, 2, \ldots, p.$$

For $f : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$, the Wasserstein gradient at ρ is then $\partial^W_\rho f(\rho) \in T_\rho \mathcal{P}_2(\mathbb{R}^d)$, such that

$$\lim_{t \to 0} \frac{f\left((I + tv)\sharp\rho\right) - f(\rho)}{t} = \int_{\mathbb{R}^d} \partial^W_\rho f(\rho)(x) \cdot v(x)\ \rho(x) dx \quad \forall v \in T_\rho \mathcal{P}(\mathbb{R}^d). \tag{2.20}$$

Thus, for a smooth $\rho : \Theta \to \mathcal{P}_2(\mathbb{R}^d)$ and $f : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$, the W_2 NGD direction for θ is given by

$$\eta^{nat}_{W_2} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\| \partial^W_\rho f + \sum_{i=1}^p \eta_i v_i \right\|^2_{L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)}. \tag{2.21}$$

As seen in (2.6), the L^2 derivatives and gradients are typically easier to calculate. Here, we discuss the relations between the L^2 and W_2 metrics that are useful for calculating the W_2 derivatives and gradients, i.e., $\{v_i\}$ and $\partial^W_\rho f$. We formulate the main conclusions in Proposition 2.2.

PROPOSITION 2.2. *Let* $\{\zeta_i\}$ *and* $\{v_i\}$ *follow* (2.6) *and* (2.19)*, respectively. The* $\partial_\rho f$ *and* $\{\zeta_i\}$ *in* (2.8) *relate to the* $\partial^W_\rho f$ *and* $\{v_i\}$ *in* (2.21) *as follows:*

$$\partial^W_\rho f = \nabla \partial_\rho f, \tag{2.22}$$

$$v_i(\theta) = \operatorname*{argmin}_v \left\{ \|v\|^2_{L^2_{\rho(\theta)}(\mathbb{R}^d;\mathbb{R}^d)} : -\nabla\cdot(\rho(\theta)v) = \zeta_i(\theta) \right\}, \quad i = 1,\dots,p. \tag{2.23}$$

Informal derivation. Given a vector field v and a small $t>0$, we have that $I+tv$ is a first-order approximation of the trajectory below where I is the identity function. Note that in Lagrangian coordinates, $\dot{x} = v(x)$. Thus, from the continuity equation (2.17), we have that

$$(I+tv)\sharp\rho = \rho - t\,\nabla\cdot(\rho v) + o(t). \tag{2.24}$$

Recall that $\zeta_i = \partial_{\theta_i}\rho$ and $v_i = \partial^W_{\theta_i}\rho$. Using this observation together with (2.6) and (2.18), we have

$$\rho(\theta+t\eta) = \rho(\theta) + t\sum_{i=1}^p \eta_i\zeta_i(\theta) + o(t),$$
$$\rho(\theta+t\eta) = \rho(\theta) - t\sum_{i=1}^p \eta_i\nabla\cdot(\rho(\theta)v_i(\theta)) + o(t)$$

for all $\eta\in\mathbb{R}^p$. By comparing the above two equations, we have

$$-\nabla\cdot(\rho(\theta)v_i(\theta)) = \zeta_i(\theta), \quad 1\le i\le p. \tag{2.25}$$

After taking (2.16) into account, we obtain (2.23).

Next, we establish a connection between $\partial_\rho f$ and $\partial^W_\rho f$. Combining (2.7), (2.20), and (2.24)–(2.25), we obtain

$$\begin{aligned}\int_{\mathbb{R}^d} \partial^W_\rho f(\rho)(x)\cdot v(x)\rho(x)dx &= -\int_{\mathbb{R}^d} \partial_\rho f(\rho)(x)\nabla\cdot(\rho(x)v(x))\,dx \\ &= \int_{\mathbb{R}^d} \nabla\partial_\rho f(\rho)(x)\cdot v(x)\ \rho(x)dx\end{aligned}$$

for all $v\in T_\rho\mathcal{P}_2(\mathbb{R}^d)$. Hence, we obtain (2.22). ∎

Similar to previous cases, we want to turn (2.21) into an unweighted L^2 formulation. Using results in Proposition 2.2, we know that the Wasserstein tangent vectors at ρ are velocity fields of minimal kinetic energy in $L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)$. We first perform a change of variables

$$\tilde{v}_i = \sqrt{\rho}\,v_i, \quad i=1,\dots,p,$$

where the set of $\{v_i\}$ follows (2.19). As a result, for each $i=1,\dots,p$, (2.23) reduces to

$$\tilde{v}_i(\theta) = \operatorname{argmin}\left\{ \|\tilde{v}\|^2_{L^2(\mathbb{R}^d;\mathbb{R}^d)} : \mathbf{B}\tilde{v} = \zeta_i(\theta) \right\}, \quad \text{where} \quad \mathbf{B}\tilde{v} = -\nabla\cdot\left(\sqrt{\rho(\theta)}\,\tilde{v}\right). \tag{2.26}$$

We then have $\tilde{v}_i = \mathbf{B}^\dagger\zeta_i$ for $i=1,\dots,p$. Denote the adjoint operator of $\mathbf{B}$ as $\mathbf{B}^*$. Note that $\mathbf{B}^*\eta = \sqrt{\rho}\nabla\eta$. Combining these observations with Proposition 2.2, formulation (2.21) becomes

(2.27)

$$\begin{aligned}\eta^{nat}_{W_2} &= \operatorname*{argmin}_{\eta\in\mathbb{R}^p} \left\| \sqrt{\rho}\nabla\partial_\rho f + \sum_{i=1}^p \eta_i\tilde{v}_i \right\|^2_{L^2(\mathbb{R}^d;\mathbb{R}^d)} = \operatorname*{argmin}_{\eta\in\mathbb{R}^p} \left\| \mathbf{B}^*\partial_\rho f + \sum_{i=1}^p \eta_i\mathbf{B}^\dagger\zeta_i \right\|^2_{L^2(\mathbb{R}^d;\mathbb{R}^d)} \\ &= \operatorname*{argmin}_{\eta\in\mathbb{R}^p} \left\| (\mathbf{L}^*)^\dagger\partial_\rho f + \sum_{i=1}^p \eta_i\mathbf{L}\zeta_i \right\|^2_{L^2(\mathbb{R}^d;\mathbb{R}^d)}, \quad \text{where } \mathbf{L} = \mathbf{B}^\dagger.\end{aligned}$$

We have reformulated the W_2 NGD as a standard L^2 minimization (2.27).

Remark 2.3. Note that the Wasserstein natural gradient is closely related to the $\dot{H}^{-1}$ natural gradient presented in subsection 2.3. Indeed, taking $s=-1$ in (2.13), we obtain that

$$\eta^{nat}_{\dot{H}^{-1}} = \underset{\eta\in\mathbb{R}^p}{\operatorname{argmin}} \left\| \nabla\partial_\rho f + \sum_{i=1}^{p} \eta_i (\nabla^*)^\dagger \zeta_i \right\|^2_{L^2(\Omega)},$$

which matches (2.27) except that the weighted divergence operator $\mathbf{B}$ defined in (2.26) is replaced with the unweighted divergence operator $-\nabla\cdot = \nabla^*$. When $\rho(\theta)\equiv 1$, these two operators coincide.

In principle, one may consider NGDs generated by the generalized operator

$$\mathbf{B}_k\tilde{v} = -\nabla\cdot\left(\rho(\theta)^k\tilde{v}\right), \quad \mathbf{L} = (\mathbf{B}_k)^\dagger,$$

where the case $k=0$ corresponds to the $\dot{H}^{-1}$ natural gradient and $k=1/2$ corresponds to the W_2 NGD. The term ρ^k is often referred to as mobility in gradient flow equations [26].

Remark 2.4. NGDs based upon the L^2 norm (2.8), the H^s norm (2.9), the $\dot{H}^s$ norm (2.12), the Fisher–Rao metric (2.14), and the W_2 metric (2.21) are similar in form but equipped with different underlying metric space $(\mathcal{M}, g)$ for ρ. All of them can be reduced to the same common form but with a different $\mathbf{L}$ operator; see (2.8), (2.10), (2.13), (2.15), and (2.27), respectively. As a result, we expect that they may perform differently in the optimization process as NGD methods, which we will see later from numerical examples in section 4.

2.6. Gauss–Newton algorithm as an L^2 natural gradient. Next, we give an example to show that the Gauss–Newton method, a popular optimization algorithm [37], can be seen as an NGD method. More discussions on this connection can be found in [31]. Assume that f measures the least-squares difference between the model $\rho(x;\theta)$ and the reference $\rho^*(x)$ distributions; that is,

$$f(\rho(\theta)) = \frac{1}{2}\int_\Omega |\rho(x;\theta)-\rho^*(x)|^2 dx, \tag{2.28}$$

where Ω is the spatial domain. Thus, the problem of finding the parameter θ becomes

$$\begin{aligned}&\inf_\theta f(\rho(\theta))\\ &\quad = \inf_\theta \frac{1}{2}\int_\Omega |\rho(x;\theta)-\rho^*(x)|^2 dx = \inf_\theta \frac{1}{2}\int_\Omega |r(x;\theta)|^2 dx, \quad r(x;\theta)=\rho(x;\theta)-\rho^*(x).\end{aligned}$$

We will denote $\rho(x;\theta)$ as $\rho(\theta)$ and $r(x;\theta)$ as $r(\theta)$.

The Gauss–Newton (GN) algorithm [37] is one popular computational method to solve this nonlinear least-squares problem. In the continuous limit, the algorithm reduces to the flow

$$\begin{aligned}\dot{\theta} = \eta^{GN} &= \underset{\eta\in\mathbb{R}^p}{\operatorname{argmin}} \left\| r(\theta) + \sum_{i=1}^{p} \partial_{\theta_i} r(\theta)\eta_i \right\|^2_{L^2(\Omega)}\\ &= \underset{\eta\in\mathbb{R}^p}{\operatorname{argmin}} \left\| \rho(\theta) - \rho^* + \sum_{i=1}^{p} \partial_{\theta_i}\rho(\theta)\eta_i \right\|^2_{L^2(\Omega)},\end{aligned} \tag{2.29}$$

where we choose a mininal-norm η if there are multiple solutions. The algorithm is based on a first-order approximation of the residual term $r(\theta+\eta) = r(\theta) + \sum_{i=1}^{p} \partial_{\theta_i} r(\theta)\eta_i + o(\eta)$.

A key observation is that (2.29) is precisely the L^2 natural gradient flow. Indeed, we have that

$$\lim_{t\to 0} \frac{f(\rho+t\zeta)-f(\rho)}{t} = \int_\Omega (\rho(\theta)-\rho^*)\,\zeta(x)dx,$$

and therefore $\partial_\rho f(\rho) = \rho(\theta)-\rho^*$. As a result, (2.8) reduces to (2.29) precisely.

The convergence rate of the GN method is between linear and quadratic based on various conditions [37]. Typically, the method is viewed as an alternative to Newton's method if one aims for faster convergence than GD but does not want to compute/store the whole Hessian.

Remark 2.5. The L^2 natural gradient flow perspective of interpreting the GN algorithm suggests that mature numerical techniques for the GN algorithm are also applicable to *general* NGD methods, including those we introduced earlier in section 2. For further connections between GN algorithms, Hessian-free optimization, and NGD, see discussions and references in [44, 38, 32, 31].

Remark 2.6. All natural gradient methods introduced in this section can be formulated as $\eta^{nat} = \operatorname{argmin}_{\eta\in\mathbb{R}^p} \|(\mathbf{L}^*)^\dagger \partial_\rho f + \sum_{i=1}^p \eta_i \mathbf{L}\zeta_i\|_{L^2}^2$, while different metric space for ρ gives rise to different operator $\mathbf{L}$. The computational complexity of approximating $\mathbf{L}$ and $(\mathbf{L}^*)^\dagger$ determines the cost of implementing a particular NGD method. In general, L^2, H^s, and $\dot{H}^s$ NGDs are easier to implement as $\mathbf{L}$ and $(\mathbf{L}^*)^\dagger$ do not depend on ρ and thus can be reused from iteration to iteration once computed. On the other hand, for Fisher–Rao and Wasserstein NGDs, $\mathbf{L}$ is ρ-dependent. If we have access to ρ directly, the Fisher–Rao information matrix only involves a diagonal scaling by $1/\rho$ compared to the L^2 information matrix. If we only have access to ρ through an empirical distribution, there are also very efficient methods of estimating G^{FR}; see [31]. In contrast, the WNGD is the most expensive among all examples discussed in section 2. Next, in section 3, we will see that there are still efficient numerical methods to mitigate the computational challenges.

3. General computational approach. In this section, we discuss our general strategy to calculate the NGD directions. As mentioned earlier, our approach is based on efficient least-squares solvers since the problem of finding the NGD direction can be formulated as (2.2). In particular, we will introduce strategies for cases when the tangent vector $\partial_\theta\rho$ cannot be obtained explicitly, which is the case for large-scale PDE-constrained optimization problems. We will first describe the general strategies and then explain how to apply these techniques to different types of natural gradients discussed in section 2. We will work in the discrete setting hereafter.

By slightly abusing the notation, we assume that $\rho:\Theta\to\mathbb{R}^k$ is a proper discretization of $\theta\mapsto\rho(\theta)$ while $\Theta\subseteq\mathbb{R}^p$. Similarly, let $f:\mathbb{R}^k\to\mathbb{R}$ be a suitable discretization of $\rho\mapsto f(\rho)$. Hence, the standard finite-dimensional gradient and Jacobian, $\partial_\rho f\in\mathbb{R}^k$ and $\partial_\theta\rho\in\mathbb{R}^{k\times p}$, are discretizations of their continuous counterparts discussed in subsection 2.1. In particular, we denote the Jacobian

$$Z = (\zeta_1\ \zeta_2\ \cdots\ \zeta_p) = \partial_\theta\rho, \quad \text{where } \zeta_j = \partial_{\theta_j}\rho. \tag{3.1}$$

Without loss of generality, we always assume $k>p$. That is, we have more data than parameters.

3.1. A unified framework. For numerical computation, our main proposal is to translate the general formula (2.2) and (2.4) for the NGD direction into a discrete least-squares formulation, given any Riemannian metric space $(\mathcal{M}, g)$.

Based on (2.8), the discrete L^2 natural gradient problem reduces to the least-squares problem

$$\eta^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \|\partial_\rho f + Z\eta\|_2^2.$$

As we have seen in section 2, besides L^2, the computation of the H^s, $\dot{H}^s$, Fisher–Rao, and WNGD directions can also be formulated as a least-squares problem,

(3.2)
$$\eta_L^{nat} = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\|(L^\top)^\dagger \partial_\rho f + LZ\eta\right\|_2^2 = \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\|(L^\top)^\dagger \partial_\rho f + Y\eta\right\|_2^2, \quad \text{where } Y = LZ,$$

for a matrix L representing the discretization of the continuous operator $\mathbf{L}$ for different metric spaces as discussed in section 2. We regard (3.2) as a unified framework since changing the metric space for the natural gradient only requires changing L while the other components remain fixed.

Note that one can compute the standard gradient $\partial_\theta f = \partial_\theta \rho^\top \partial_\rho f = Z^\top \partial_\rho f$ by the chain rule. From (3.2), we can also obtain the common formulation for the NGD as

$$\begin{aligned} \eta_L^{nat} &= -(Z^\top L^\top L Z)^{-1}(Z^\top L^\top (L^\top)^\dagger \partial_\rho f) = -(Y^\top Y)^{-1}(Z^\top \partial_\rho f) \\ &= -(Y^\top Y)^{-1}\partial_\theta f = -G_L^{-1}\partial_\theta f, \end{aligned} \tag{3.3}$$

where $G_L = Y^\top Y$ is the corresponding information matrix defined in (2.5).

Remark 3.1. The unified framework (3.2) is general and applies to cases beyond NGDs discussed in section 2. For ρ in a metric space $(\mathcal{M}, g)$ with a corresponding tangent space $T_\rho \mathcal{M}$, we have

$$\langle \zeta_1, \zeta_2 \rangle_{g(\rho)} \approx \vec{\zeta_1}^\top A_\rho^g \, \vec{\zeta_2} \quad \forall \zeta_1, \zeta_2 \in T_\rho \mathcal{M},$$

where $\vec{\zeta_1}$, $\vec{\zeta_2}$ denote the discretized ζ_1, ζ_2. A proper discretization that preserves the metric structure should yield a symmetric positive definite matrix A_ρ^g that admits decomposition $A_\rho^g = L^\top L$. As a result, the discretization of (2.4) turns into the same formula as (3.2):

$$\begin{aligned} \eta_L^{nat} &= -(Z^\top A_\rho^g Z)^{-1}(Z^\top \partial_\rho f) = -(Z^\top L^\top L Z)^{-1}(Z^\top L^\top (L^\top)^\dagger \partial_\rho f) \\ &= \operatorname*{argmin}_{\eta \in \mathbb{R}^p} \left\|(L^\top)^\dagger \partial_\rho f + Y\eta\right\|_2^2, \quad \text{where } Y = LZ. \end{aligned}$$

The concrete form of L will depend on the specific metric space $(\mathcal{M}, g)$.

Next, we will first assume that L is given and discuss how to compute η_L^{nat} provided whether the Jacobian Z is available or not; see subsections 3.2 and 3.3. Later in subsection 3.4, we will comment on obtaining the matrix L based on the natural gradient examples in section 2.

3.2. Z available. When Z is available, there are two main methods to compute η_L^{nat}.

One may follow (3.3) by first constructing the information matrix $G_L = Y^\top Y$ and then computing its inverse. This is a reasonable method when the number of parameters, i.e., p, is small, and G_L is invertible. However, if G_L is singular or has bad conditioning, it is *more advantageous* to compute η_L^{nat} following (3.2). Note that the condition number of G_L can be nearly the square of the condition number of L, making it more likely to suffer from numerical instabilities.

The second and also our recommended approach is to solve the least-squares problem (3.2). We may utilize the QR factorization to do so [14]. Assume that $Y = LZ$ has full column rank. Let $Y = QR$, where Q has orthonormal columns and R is an upper triangular square matrix. Thus,

$$\eta_L^{nat} = -Y^\dagger (L^\top)^\dagger \partial_\rho f = -R^{-1} Q^\top (L^\top)^\dagger \partial_\rho f. \tag{3.4}$$

The additional computational cost of evaluating η_L^{nat} after the QR decomposition is the backward substitution to evaluate R^{-1} instead of inverting R directly.

If the given model $\rho(\theta)$ allows us to write down how ρ depends on θ analytically, then the Jacobian $\partial_\theta \rho$ is readily available. In such cases, we can directly solve (3.2) using the QR decomposition to obtain the NGDs; see subsection 4.1 for a Gaussian mixture example.

We summarize the algorithm when the Jacobian Z and the matrices $L, (L^\top)^\dagger$ are available; see subsection 3.4 for how to obtain L and $(L^\top)^\dagger$ for the examples presented in section 2, and see Appendix B.2 for discussions of what to do when $Y = LZ$ is rank-deficient.

3.3. Z unavailable. Often, the model $\rho(\theta)$ is not available analytically, but the relationship between ρ and θ is given implicitly via solutions of a system, e.g., a PDE constraint,

$$h(\rho, \theta) = \mathbf{0}, \tag{3.5}$$

for some smooth $h : \mathbb{R}^k \times \mathbb{R}^p \to \mathbb{R}^k$ such that $\det(\partial_\rho h) \neq 0$. In such cases, the Jacobian $Z = \partial_\theta \rho$ in (3.1) is not readily available and has to be computed or implicitly evaluated.

3.3.1. The implicit function theorem and adjoint-state method. Based on the first-order variation of (3.5), the most direct option is to apply the implicit function theorem

$$\partial_\rho h \; \partial_\theta \rho = \partial_\rho h \; Z = -\partial_\theta h. \tag{3.6}$$

The above equation consists of p linear systems in k variables. If $\partial_\rho h$ has a simple format, or the size of θ is not too large, it could still be computationally feasible to first obtain $Z = \partial_\theta \rho$ by solving (3.6), and then follow strategies in subsection 3.2 to compute the NGD.

However, if p is large, a more efficient option is to use methods based on the so-called adjoint-state method [40]. Note that Z is the rate of change of the *full state* ρ with respect to θ. Thus, if we only need the rate of change of ρ *along a specific vector* $\xi \in \mathbb{R}^k$, we do not need the whole Z; instead, we need $\xi^\top Z$, which can be calculated by solving only one linear system for each ξ.

Indeed, for a given $\xi \in \mathbb{R}^k$, let us consider the *adjoint equation*

$$\lambda_\xi^\top \partial_\rho h = \xi^\top \quad \Longleftrightarrow \quad (\partial_\rho h)^\top \lambda_\xi = \xi. \tag{3.7}$$

Combining (3.6) and (3.7), we obtain that

$$Z^\top \xi = Z^\top \left(\partial_\rho h\right)^\top \lambda_\xi = -\partial_\theta h^\top \lambda_\xi. \tag{3.8}$$

The vector λ_ξ in (3.7) is called the *adjoint variable* corresponding to the given vector ξ.

Here is an important example where we do not need the full Z. If we choose $\xi = \partial_\rho f \in \mathbb{R}^k$, then (3.8) gives the standard gradient

$$\partial_\theta f(\rho(\theta)) = \partial_\theta \rho^\top \partial_\rho f = Z^\top \partial_\rho f = -\partial_\theta h^\top \ \lambda_\xi, \tag{3.9}$$

where λ_ξ is the solution to (3.7) with $\xi = \partial_\rho f \in \mathbb{R}^k$. This is a widely used method to efficiently evaluate the gradient of a large-scale optimization in solving PDE-constrained optimization problems originated from optimal control and computational inverse problems [40].

Next, we will explain in detail how to harness the power of the adjoint-state method to evaluate the general NGD directions through iterative methods.

3.3.2. Krylov subspace methods. Given an arbitrary vector $\eta \in \mathbb{R}^p$, we may evaluate

$$G_L \ \eta = Z^\top L^\top L Z \ \eta \tag{3.10}$$

through the adjoint-state method even if we cannot access the information matrix G_L since the Jacobian Z is unavailable directly. Let $\widehat{\rho} \in \mathbb{R}^k$ be an arbitrary vector, and consider the following constrained optimization problem [34]:

$$\min_\theta J(\rho(\theta)) = \rho^\top \widehat{\rho}, \quad \text{s.t. } h(\rho(\theta), \theta) = \mathbf{0}. \tag{3.11}$$

Note that this objective function $J(\rho(\theta))$ in (3.11) is different from the main objective function (1.1) but has the same constraint (3.5). A direct calculation reveals that the gradient of $J(\rho(\theta))$ with respect to the parameter θ is $Z^\top \widehat{\rho}$. Therefore, if we set $\widehat{\rho} = L^\top L Z \eta$, the gradient

$$\partial_\theta J(\rho(\theta)) = Z^\top \widehat{\rho} = Z^\top L^\top L Z \ \eta = G_L \ \eta,$$

which is exactly what we aim to compute in (3.10).

From the constraint $h(\rho(\theta), \theta) = \mathbf{0}$ and its first-order variation (3.6), we have

$$\partial_\rho h \ Z \ \eta + \partial_\theta h \ \eta = \mathbf{0}.$$

Thus, $Z \ \eta$ can be obtained as the solution to a linear system with respect to γ:

$$\partial_\rho h \ \gamma = -\partial_\theta h \ \eta. \tag{3.12}$$

Based on the adjoint-state method introduced in section 3.3.1, we can compute the gradient as

$$\partial_\theta J(\rho(\theta)) = -\partial_\theta h^\top \ \lambda,$$

where λ satisfies the adjoint equation below with a given γ that solves (3.12):

$$\partial_\rho h^\top \lambda = \partial_\rho J = \widehat{\rho} = L^\top L Z \eta = L^\top L \gamma. \tag{3.13}$$

To sum up, with a fixed θ and the corresponding $\rho(\theta)$, we have an efficient way to evaluate the *linear action* $\eta \mapsto G_L \eta$ for any given η by three steps; see Algorithm 3.2.

Algorithm 3.1. Compute the NGD direction given Z, L, $(L^\top)^\dagger$, and $\partial_\rho f$.

1: Compute $Y = LZ$.
2: Perform economy-size QR factorization: $[Q, R] = \mathtt{qr}(Y)$.
3: Compute the NGD direction $\eta_L^{nat} = -R^{-1}Q^\top (L^\top)^\dagger \partial_\rho f$.

Algorithm 3.2. Evaluate the linear action $\eta \mapsto G_L \eta$ given an arbitrary vector η.

1: Given the implicit constraint h, solve the linear system $\partial_\rho h\ \gamma = -\partial_\theta h\ \eta$ and obtain γ.
2: Given linear actions based on L and $L^\top$, solve the linear system $\partial_\rho h^\top \lambda = L^\top L\gamma$ and obtain λ.
3: Evaluate $-\partial_\theta h^\top\ \lambda$, which equals to $G_L\ \eta$.

Algorithm 3.3. Compute the NGD direction when Z is not explicitly available.

1: Given the constraint h, solve the linear system $(\partial_\rho h)^\top \lambda = \partial_\rho f$ and obtain λ.
2: Compute the parameter gradient $\partial_\theta f(\rho(\theta)) = \partial_\theta \rho^\top \partial_\rho f = -\partial_\theta h^\top\ \lambda$.
3: Obtain the linear action $\eta \mapsto G_L \eta$ following steps in Algorithm 3.2.
4: Use the conjugate gradient method to solve for η_L^{nat} where $G_L\, \eta_L^{nat} = -\partial_\theta f(\rho(\theta))$.

TABLE 1
The number of propagations among different optimization methods.

	GD	NGD	Newton's method
Forward propagation $\theta \mapsto \rho(\theta)$	1	1	1
Backward propagation $\xi \mapsto \partial_\theta \rho^\top \xi$	1	1	2
Linearized forward propagation $\omega \mapsto \partial_\theta \rho\, \omega$	0	1*	1

*For NGD, different choice of metric affects the complexity of the linearized forward solve.

Given the linear action $\eta \mapsto G_L \eta$, we need to solve the linear system

$$G_L\ \eta_L^{nat} = -\partial_\theta f(\rho(\theta)) \tag{3.14}$$

to find the NGD direction η_L^{nat}. As seen in (3.9), we can obtain the right-hand side $-\partial_\theta f(\rho(\theta))$ through the adjoint-state method. One may then solve for η_L^{nat} through iterative linear solvers based on the Krylov subspace methods [43], e.g., the conjugate gradient method. We summarize all the steps above in Algorithm 3.3.

One may use Algorithm 3.3 instead of Algorithm 3.1 when Z is available but the QR factorization of $Y = LZ$ is too costly, for instance, in some machine learning applications. Since "wall-clock" time can be highly affected by the implementation and the computer specification, in Table 1, we summarize the number of propagations per iteration among different methods [48]. For different NGDs, the cost of the linear action $\gamma \mapsto L^\top L\gamma$ varies, which we will discuss in subsection 3.4.

3.4. Computation for natural gradient examples in section 2. In subsections 3.2 and 3.3, we have shown how to compute the NGD direction η_L^{nat} given whether Z is easily available or not. Both strategies require the matrix L, which depends on the particular metric space for the natural gradient. Next, we specify the form of L based on cases discussed in section 2.

The L^2 case in subsection 2.1 corresponds to $L = I$, the $k \times k$ identity matrix, while the Fisher–Rao–Hellinger natural gradient discussed in subsection 2.4 corresponds to

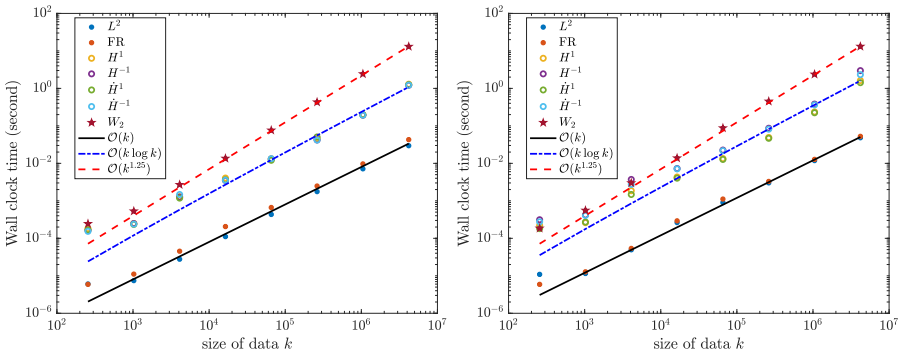
$L = \operatorname{diag}(1/\sqrt{\rho}) \in \mathbb{R}^{k\times k}$, which incurs $\mathcal{O}(k)$ more flops per iteration compared to the L^2 NGD method. For the H^s natural gradient discussed in subsection 2.2, L corresponds to proper discretization of $\mathbf{D}^s$ (for $s>0$) and $((\mathbf{D}^{-s})^*)^\dagger$ (for $s<0$). Next, we give a few concrete examples. When $s=1$, $\mathbf{L}=\mathbf{D}^1=[I,\nabla]^\top$ and $(\mathbf{L}^*)^\dagger = \mathbf{D}^1((\mathbf{D}^1)^*\mathbf{D}^1)^{-1} = [I,\nabla]^\top(I-\triangle)^{-1}$. When $s=-1$, $\mathbf{L}=((\widetilde{\mathbf{D}}^{-1})^*)^\dagger = [I,\nabla]^\top(I-\triangle)^{-1}$ while $(\mathbf{L}^*)^\dagger = [I,\nabla]^\top$. Similarly, for the $\dot{H}^s$ natural gradient discussed in subsection 2.3, L should correspond to proper discretization of $\widetilde{\mathbf{D}}^s$ (for $s>0$) and $((\widetilde{\mathbf{D}}^{-s})^*)^\dagger$ (for $s<0$). For instance, when $s=1$, $\mathbf{L}=\widetilde{\mathbf{D}}^1=\nabla$, and $(\mathbf{L}^*)^\dagger = \widetilde{\mathbf{D}}^1((\widetilde{\mathbf{D}}^1)^*\widetilde{\mathbf{D}}^1)^{-1} = \nabla(-\triangle)^{-1}$; when $s=-1$, $\mathbf{L}=((\widetilde{\mathbf{D}}^{-1})^*)^\dagger = \nabla(-\triangle)^{-1}$ while $(\mathbf{L}^*)^\dagger=\nabla$. The symmetry between the cases of $H^s/\dot{H}^s$ and the cases of $H^{-s}/\dot{H}^{-s}$ $\forall s>0$ comes from the fact that they are dual Sobolev spaces. The computation of the natural gradient based on the H^s and $\dot{H}^s$ metric can be efficiently computed. This is because there are fast algorithms for discretizing and computing the actions of the gradient and (inverse) Laplacian operators for periodic, Dirichlet, and zero-Neumann boundary conditions in $\mathbf{L}$ and $(\mathbf{L}^*)^\dagger$ [12, 55].

Based on the unweighted reformulation (2.27), computing the W_2 NGD discussed in subsection 2.5 requires the discretization of $\mathbf{L}=\mathbf{B}^\dagger$. We can first discretize the differential operator $\mathbf{B}$, denoted as B, and then compute $L=B^\dagger$, which can be used regardless of whether the Jacobian $Z=\partial_\theta\rho$ is explicitly given or implicitly provided through the constraint (3.5). As an example, we describe how to obtain the matrix L for the WNGD (2.27) in Appendix B.1 based on a finite-difference discretization of the differential operator. In Remark 2.3, we commented that when $\rho(x)$ is constant, WNGD reduces to $\dot{H}^{-1}$-based NGD. However, in general, the computation of the WNGD is more expensive than the $H^s/\dot{H}^s$ cases for two reasons. First, the information matrix G and the operator $\mathbf{L}$ for the WNGD are ρ-dependent, so in every iteration of the NGD method, one has to recompute them, which incurs extra complexity. Second, as mentioned above, the computation of $H^s/\dot{H}^s$ NGD can be done through fast Fourier or discrete cosine transforms (depending on the domain). It is, however, inapplicable to the Wasserstein case since it involves solving a *weighted* differential equation. In Appendix B.1, we use QR factorization to obtain $L=B^\dagger$ given B. We approximate B using the finite-difference method, so $B^\top$ is very sparse. Using a multifrontal multithreaded sparse QR factorization [9], it has much better complexity than the conventional $\mathcal{O}(k^3)$. We summarize the observed computational costs of obtaining L and $(L^\top)^\dagger$ for different NGD methods in Table 2. See also Figure 1a for the computational time comparison among different metrics.

After obtaining L and $(L^\top)^\dagger$, the QR factorization of $Y=LZ$ followed by computing the natural gradient direction η_L^{nat} based on (3.4) will incur $\mathcal{O}(kp^2)$ flops if the Jacobian Z is available; see Figure 1b for an observed computational time to obtain the NGD η among different metrics for a case where Z is analytically available (see section 4.1). When Z is not analytic, such as from PDE (section 4.3) or neural network models (section 4.2), we will see that the cost in computing NGDs among different methods is no longer dominated by the cost of computing L and $(L^\top)^\dagger$.

TABLE 2
Summary of the observed computational costs for linear actions L and $(L^\top)^\dagger$ in (3.2).

	L^2	Fisher–Rao	$H^s/\dot{H}^s$, $s>0$	$H^s/\dot{H}^s$, $s<0$	W_2
change over iteration	✘	✓	✘	✘	✓
computing $v\mapsto Lv$	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k\log k)$	$\mathcal{O}(k^{1.25})$
computing $v\mapsto (L^\top)^\dagger v$	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k\log k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$



(a) Evaluate L and $(L^\top)^\dagger$ linear actions (b) Compute NGD direction η

FIG. 1. *The observed wall clock time for evaluating $v \mapsto Lv$ and $v \mapsto (L^\top)^\dagger v$ linear actions (left) and for computing one NGD direction η with a fixed p (right) based on different metrics.*

3.5. Extensions and variants. In this section, we briefly comment on several practical variants of using the NGD method based on a particular choice of the data metric space.

3.5.1. A damped information matrix. If the discretized information matrix G_L is rank deficient or ill-conditioned, one may consider rank-revealing QR factorization; see Appendix B.2. As an alternative approach, a damped information matrix in the form $G_\lambda = \lambda I + G_L$ is often used for numerical stability and to avoid extreme updates, where λ is the damping parameter. One notable example is the Levenberg–Marquardt method as a damped Gauss–Newton method [44], while the latter is equivalent to the L^2 NGD in our framework; see subsection 2.6.

Since the fundamental difference between GD and NGD lies in how one measures the distance between the potential next iterate and the current iterate, the damped version corresponds to choosing the next iterate based on a mixed metric from the θ-domain and ρ-domain. Indeed, in the implicit form (1.5) and (1.6), the damped version can be written as

$$\theta^{l+1} = \operatorname*{argmin}_\theta \left\{ f(\rho(\theta)) + \frac{\lambda\, d_\theta(\theta, \theta^l)^2 + d_\rho(\rho(\theta), \rho(\theta^l))^2}{2\tau} \right\}. \tag{3.15}$$

When d_θ is the Euclidean metric on θ-domain, we obtain the identity matrix I in G_λ, but other choices of damping metric can also be considered.

Alternatively, one can use another ρ-space metric to regularize instead of any metric on the θ-space. For example, let d_{ρ_2} be the main natural gradient metric and d_{ρ_1} be the regularizing natural gradient metric. The next iterate obtained in the implicit Euler scheme is given by

$$\theta^{l+1} = \operatorname*{argmin}_\theta \left\{ f(\rho(\theta)) + \frac{\lambda\, d_{\rho_1}(\rho(\theta), \rho(\theta^l))^2 + d_{\rho_2}(\rho(\theta), \rho(\theta^l))^2}{2\tau} \right\}, \tag{3.16}$$

while the damping parameter λ determines the strength of regularization. We comment that the H^1 natural gradient can be seen as the $\dot{H}^1$ natural gradient damped by the L^2 natural gradient.

3.5.2. Mini-batch NGD. Similar to mini-batch GD, one can also use mini-batch NGD by computing the *natural* gradient of the objective function with respect to a subset of the data ρ. Consider a random sketching matrix $S \in \mathbb{R}^{k' \times k}$, $k' < k$.

Each row of S has at most one nonzero entry 1. Thus, $S\rho \in \mathbb{R}^{k'}$ is the mini-batch data. The objective function also becomes $f(S\rho(\theta))$.

The mini-batch NGD can find the next iterate θ^{l+1} implicitly through

$$\theta^{l+1} = \underset{\theta}{\operatorname{argmin}} \left\{ f(S\rho(\theta)) + \frac{d_\rho(S\rho(\theta), S\rho(\theta^l))^2}{2\tau} \right\},$$

where d_ρ is the ρ-space metric. It is equivalent to changing the data metric from $d_\rho(\cdot,\cdot)$ to a random pseudo metric $d_\rho(S\cdot, S\cdot)$. The information matrix and the NGD direction are

$$G = Z^\top S^\top L^\top LSZ, \qquad \eta = G^{-1}\partial_\theta f(S\rho(\theta)),$$

where L depends on $d_\rho(S\cdot, S\cdot)$ and Z is the Jacobian. Note that S changes over iterations.

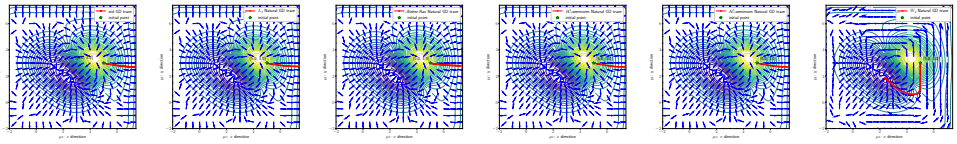
Also, we remark that $SZ \in \mathbb{R}^{k'\times p}$ can be seen as a random sketching of the Jacobian matrix Z. If Z is low-rank, the column space of $SZ \in \mathbb{R}^{k'\times p}$ can be a close approximation to the column space of Z, but SZ is much smaller in size. See Appendix B.4, where similar techniques from random linear algebra can help explore the column space of Z and further reduce the computational cost.

4. Numerical results. In this section, we present three optimization examples to illustrate the effectiveness of our computational strategies for NGD methods. We first present the parameter reconstruction of a Gaussian mixture model where the Jacobian $\partial_\theta\rho$ is analytically given. Our second example is to solve the 2D Poisson equation using physics-informed neural networks (PINN) [42], where the Jacobian $\partial_\theta\rho$ can be numerically obtained through automatic differentiation. We then present a large-scale waveform inversion, a PDE-constrained optimization problem where the Jacobian $\partial_\theta\rho$ is not explicitly given. Using our computational strategy proposed in subsection 3.3, we can efficiently implement the NGD method based on a general metric space. The first example shows that various (N)GD methods converge to different stationary points of a nonconvex objective function. The last two tests illustrate that different (N)GD methods have various convergence rates. Both phenomena are interesting as they indicate that one may achieve global convergence or faster convergence by choosing a proper metric space $(\mathcal{M}, g)$ that fits the problem.

4.1. Gaussian mixture model. Consider the Gaussian mixture model, which assumes that all the data points are generated from a mixture of a finite number of normal distributions with unknown parameters. Consider a probability density function $\rho(x;\theta) : \mathbb{R}^d \mapsto \mathbb{R}^+$, where

$$\rho(x;\theta) = w_1\mathcal{N}(x;\mu_1,\Sigma_1) + \cdots + w_i\mathcal{N}(x;\mu_i,\Sigma_i) + \cdots + w_k\mathcal{N}(x;\mu_k,\Sigma_k).$$

The ith Gaussian, denoted as $\mathcal{N}(x;\mu_i,\Sigma_i)$ with the mean vector $\mu_i \in \mathbb{R}^d$ and the covariance matrix $\Sigma_i \in \mathbb{R}^{d\times d}$, has a weight factor $w_i \geq 0$. Note that $\sum_i w_i = 1$. Here, θ could represent parameters such as $\{w_i\}$, $\{\mu_i\}$, and $\{\Sigma_i\}$. We formulate the inverse problem of finding the parameters as a data-fitting problem by minimizing the least-squares loss $f(\rho(\theta))$ on a compact domain Ω where the objective function follows (2.28). Here, ρ^* is the observed reference density function. Note that the dependence between the state variable ρ and the parameter θ is explicit here. Thus, we can compute the Jacobian $\partial_\theta\rho$ analytically, and the numerical scheme follows subsection 3.2.



(a) GD (b) L^2 NGD (c) FR NGD (d) H^1 NGD (e) H^{-1} NGD (f) W_2 NGD

FIG. 2. *Gaussian mixture example: Level sets, vector fields, and convergent paths using GD and different NGD methods to invert* μ_1. *All algorithms start from initial guess* $(5,3)$.

We consider reference $\rho^*(x) = 0.3\mathcal{N}(x;(1,3),0.6I) + 0.7\mathcal{N}(x;(3,2),0.6I)$ and the domain $\Omega = [-2.75, 7.25]^2$. We fix μ_2 and the weights to be incorrect and invert $\theta = \mu_1$. That is, $\rho(x;\theta) = 0.2\mathcal{N}(x;\theta,0.6I) + 0.8\mathcal{N}(x;(4,3),0.6I)$. Figure 2 shows the convergence paths of GD and L^2, Fisher–Rao, H^1, H^{-1}, W_2 NGD methods under the initial guess $(5,3)$, which is chosen since it belongs to different basins of attractions for different optimization methods. We choose the largest possible step size such that the objective function monotonically decays. They are 0.3, 0.04, 0.8, 0.2, 0.2, and 3 for methods in Figure 2 from left to right. WNGD converges to the global minimum, while all other methods converge to local minima by taking different convergence paths.

We aim to gain better a understanding of their different convergence behaviors. Given a fixed lth iterate, different algorithms find the $(l+1)$th iterate, but based on different "principles" nicely revealed in the proximal operators (1.5) and (1.6). Here, we use $\theta^{l+1}_{\text{std}}$, $\theta^{l+1}_{W_2}$, and $\theta^{l+1}_{L^2}$ to denote the next iterates based on GD, L^2 NGD, and WNGD, respectively. We then have

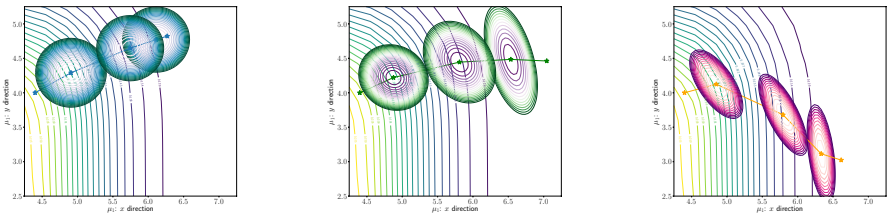
$$\begin{aligned}
\theta^{l+1}_{\text{std}} &= \theta^l + \operatorname*{argmin}_h \left\{ \nabla_\theta f^\top h + \frac{1}{2\tau} h^\top h \right\} \approx \operatorname*{argmin}_\theta \left\{ f(\rho(\theta)) + \frac{|\theta - \theta^l|^2}{2\tau} \right\}, \\
\theta^{l+1}_{L^2} &= \theta^l + \operatorname*{argmin}_h \left\{ \nabla_\theta f^\top h + \frac{1}{2\tau} h^\top \partial_\theta \rho^\top \partial_\theta \rho \, h \right\} \\
&\approx \operatorname*{argmin}_\theta \left\{ f(\rho(\theta)) + \frac{||\rho(\theta) - \rho(\theta^l)||_2^2}{2\tau} \right\}, \\
\theta^{l+1}_{W_2} &= \theta^l + \operatorname*{argmin}_h \left\{ \nabla_\theta f^\top h + \frac{1}{2\tau} h^\top (B^\dagger \partial_\theta \rho)^\top B^\dagger \partial_\theta \rho \; h \right\} \\
&\approx \operatorname*{argmin}_\theta \left\{ f(\rho(\theta)) + \frac{W_2^2(\rho(\theta), \rho(\theta^l))}{2\tau} \right\}.
\end{aligned}$$

The above equations show that, locally, different (N)GD methods solve different quadratic problems given the same step size τ. In Figure 3, we illustrate the level set of each quadratic problem for which the minimum is selected as the next iterate. The level set of the same objective function $f(\rho(\theta))$ is shown in the background. Our observation aligns with the example in [8, Figure 3].

4.2. Physics-informed neural networks. Physics-informed neural networks (PINN) is a variational approach to solve PDEs with the solution parameterized by neural networks [42]. Here, as an example, we use PINN to solve the 2D Poisson equation on the domain $\Omega = [-1,1]^2$,

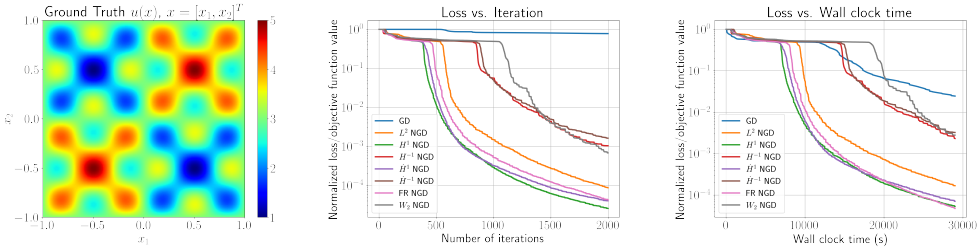
$$-\triangle u = \phi, \quad \text{with } u = \psi \text{ on } \partial\Omega,$$

where $\phi(x) = 2\pi^2 \sin(\pi x_1)\sin(\pi x_2) + 18\pi^2 \sin(3\pi x_1)\sin(3\pi x_2)$ and $\psi(x) = 3$, whose solution is $u(x) = \sin(\pi x_1)\sin(\pi x_2) + \sin(3\pi x_1)\sin(3\pi x_2) + 3$, $x = [x_1, x_2]^\top$. The training loss function is



(a) Standard gradient descent (b) L^2 natural gradient (c) W_2 natural gradient

FIG. 3. *The local quadratic models of GD, L^2 NGD, and W_2 NGD in the first several iterations.*



(a) True solution (b) Loss decay vs. iteration number (c) Loss decay vs. wall clock time

FIG. 4. (a) *PINN example true solution;* (b) *loss function value decay in terms of the number of iterations;* (c) *loss function value decay in terms of the wall clock time.*

$$f(\rho(\theta)) = \frac{\gamma}{N_1}\sum_{i=1}^{N_1} |\triangle\rho(x_i,\theta) + \phi(x_i)|^2 + \frac{2-\gamma}{N_2}\sum_{j=1}^{N_2} |\rho(x_j,\theta) - \psi(x_j)|^2,$$

where $\rho(x,\theta)$ is a feed-forward neural network of shape $(2, 20, 30, 20, 1)$ with the hyperbolic tangent `tanh` as the activation function. The parameters are the weights and biases, denoted by θ. We use $N_1 = 2304$ collocation points in the domain interior and $N_2 = 196$ points on $\partial\Omega$, both equally spaced. We set $\gamma = 0.01$ to balance the two terms in the loss function. For a weight matrix of size d_1-by-d_2, we initialize its entries i.i.d. following the normal distribution $\mathcal{N}(0, \frac{2}{d_1+d_2})$. All biases are initialized as zero, except the one in the last layer, which is set to be 3. We fix the random seed to ensure the same initialization for all optimization algorithms of interests.

We train PINN using GD and different NGDs based on metrics discussed in section 2. We use backtracking line search to select the step size (learning rate) in (N)GD algorithms. The true solution is shown in Figure 4a, while Figures 4b and 4c show the loss value decay with respect to the number of iterations and the wall-clock time, respectively. We can see that all NGD methods are faster than GD, while H^1 and $\dot{H}^1$-based NGDs yield the fastest convergence in both comparisons. Neural networks can suffer from slow convergence on the high-frequency parts of the residual due to its intrinsic low-frequency bias [53]. The $H^1/\dot{H}^1$-based NGDs enforce extra weights on the oscillatory components of the Jacobian, giving faster convergence than L^2 NGD. In contrast, $H^{-1}/\dot{H}^{-1}$ NGDs bias towards the smooth components of the Jacobian, which delay the convergence of high-frequency residuals and thus the overall convergence. As discussed in Remark 2.6, WNGD requires a ρ-dependent matrix L, which increases the wall clock time per iteration. Interestingly, when the loss value becomes small, WNGD has a faster decay rate than $H^{-1}/\dot{H}^{-1}$ NGDs despite being

asymptotically equivalent in spectral properties (see Remark 2.3), demonstrating the potential benefits of having a state-dependent information matrix $G(\theta)$.

4.3. Full waveform inversion. Finally, we present a full waveform inversion (FWI) example where the Jacobian is not explicitly given. As a PDE-constrained optimization, the dependence between the data and the parameter is implicitly given through the scalar wave equation

$$m(x)u_{tt}(x,t) + \triangle u(x,t) = s(x,t), \tag{4.1}$$

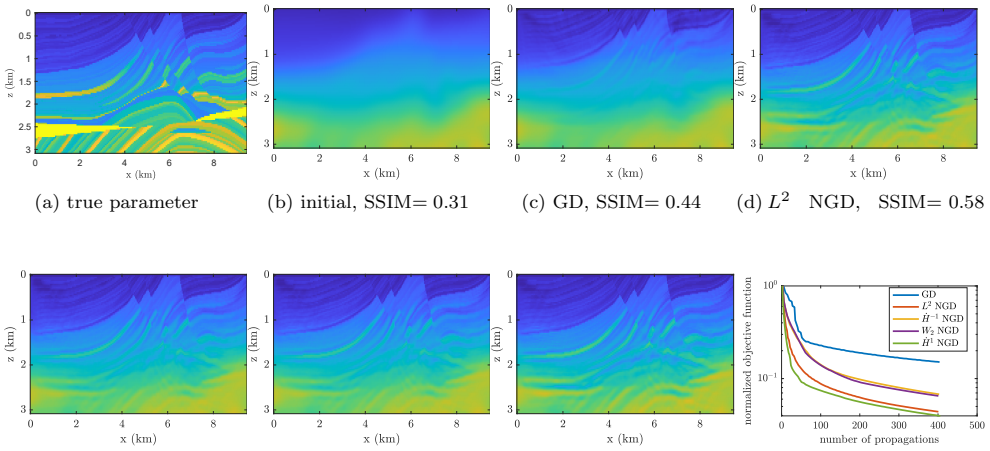
where $s(x,t)$ is the source term and (4.1) is equipped with the initial condition $u(x,0) = u_t(x,0) = 0$ and an absorbing boundary condition to mimic the unbounded domain.

After discretization, the unknown function $m(x)$ becomes a finite number of unknowns, which we denote by θ for consistency. Unlike the Gaussian mixture model, the size of θ in this example is large as $p = 36720$. We obtain the observed data $\rho_r = u(x_r,t)$ at a sequence of receivers $\{x_r\}$ for $r = 1,\dots,n_r$. The least-squares objective function is

$$f(\rho(\theta)) = \frac{1}{2}\sum_{i=1}^{n_s}\sum_{r=1}^{n_r} \|\rho^*_{i,r} - \rho_{i,r}(\theta)\|_2^2, \tag{4.2}$$

where ρ^* is the observed reference data, and i is the source term index to consider inversions with multiple sources $\{s_i(x,t)\}$ as the right-hand side in (4.1). In our test, $n_s = 21$ and $n_r = 306$.

The true parameter is presented in Figure 5a. We remark that minimizing (4.2) with the constraint (4.1) is a highly nonconvex problem [47]. We avoid dealing with the nonconvexity by choosing a good initial guess; see Figure 5b. One may also use other objective functions such as the Wasserstein metric to improve the optimization landscape [11]. We follow subsection 3.3 to carry out the implementation for various



(a) true parameter (b) initial, SSIM= 0.31 (c) GD, SSIM= 0.44 (d) L^2 NGD, SSIM= 0.58

(e) $\dot{H}^{-1}$ NGD, SSIM= 0.53 (f) W_2 NGD, SSIM= 0.53 (g) $\dot{H}^1$ NGD, SSIM= 0.61 (h) convergence history

FIG. 5. *FWI example:* (a) *ground truth;* (b) *initial guess;* (c)–(g) *inversion results using GD and NGDs based on the* L^2, $\dot{H}^{-1}$, W_2, *and* $\dot{H}^1$ *metrics after* 400 *PDE solves;* (h) *the history of the objective function decay versus the number of propagations/PDE solves. SSIM denotes the structural similarity index measure compared with* (a). *A bigger value means better similarity.*

NGD methods since the Jacobian $\partial_\theta \rho$ is not explicitly given, and the adjoint-state method has to be applied based on (4.1). The step-size is chosen based on backtracking linear search. We use the same criteria for all algorithms. The GD (see Figure 5c) converges slowly compared to the NGD methods, while $\dot{H}^1$, L^2, $\dot{H}^{-1}$, and W_2 NGDs are in descending order in terms of image resolution measured by both the objective function and the structural similarity index measure (SSIM); see Figures 5d–5h. The convergence history in Figure 5h shows the objective function decay with respect to the number of propagations (see Table 1). For FWI, each propagation corresponds to one wave equation (PDE) solve with different source terms. Note that wavefields are not naturally probability distributions. Thus, when we implement the W_2 natural gradient, we normalize the data to be probability densities following [10, 11]. As we have discussed in Remark 2.3, the W_2 and $\dot{H}^{-1}$ natural gradients are closely related, which are also reflected in this numerical example as the reconstructions in Figures 5e and 5f are very similar. All the tests shown in Figure 5 directly demonstrate that NGDs are typically faster than GD, and more importantly, the choice of the metric space $(\mathcal{M}, g)$ for NGD (see (2.2)) also has a direct impact on the convergence rate.

5. Conclusions. Inspired by the natural gradient descent (NGD) method in learning theory, we develop efficient computational techniques for PDE-based optimization problems for generic choices of the *natural* metric. NGD exploits the geometric properties of the state space, which is particularly appealing for PDE applications that have rich flexibility in choosing the metric spaces.

Handling the high-dimensional parameter space and state space are the two main computational challenges of NGD methods. Here, we propose numerical schemes to tackle the high-dimensional parameter space when the forward model, with a relatively low-dimensional state space, is discretized on a regular grid. Our approach relies on reformulating the problem of finding NGD directions as standard L^2-based least-squares problems on the continuous level. After discretization, the NGD directions can be efficiently computed by numerical linear algebra techniques. We discuss both explicit and implicit forward models by taking advantage of the adjoint-state method.

The second computational challenge of high-dimensional state space stands out for Sobolev and Wasserstein NGDs. In this work, we apply finite differences on regular grids for low-dimensional state space. On the one hand, when the state-space dimension is high, discretization on a regular grid suffers from the curse of dimensionality, and other parameterizations have to be considered. On the other hand, when the state variable is not given on a regular grid, there are other ways to discretize those differential operators, which require more careful attention. For example, generative models are pushforward mappings, representing probability measures in high-dimensional state spaces by point clouds (samples). Applying the Sobolev and Wasserstein NGDs to state variables in the form of empirical distributions will most likely require alternative discretization approaches for differential operators, such as graph- or neural network-based methods.

A very interesting question is what the best "natural" metric in NGD should be. Regarding this, we numerically investigated the convergence behaviors of GD and various NGD methods based on different metric spaces. The empirical results indicate that the choice of the metric space in an NGD not only can change the rate of convergence but also can influence the stationary point where the iterates converge, given a nonconvex optimization landscape. A rigorous understanding of the "best" metric choice for a given problem is an important research direction. For maximum likelihood estimation problems, the Fisher–Rao NGD is asymptotically

Fisher-efficient; Sobolev NGDs (e.g., H^1 and $\dot{H}^1$) are suitable for solving optimal transport and mean-field game problems [20, 18, 27, 29]; when the metric is induced by f and suitable conditions are met, the corresponding NGD is asymptotically Newton's method [30, 8, 31]. Despite these results, to the best of our knowledge, there is no general framework for a systematic derivation of the best natural gradient metric for a given problem.

It is reasonable to believe that as the topic matures, there will be an increasing necessity for efficient techniques for computing NGD directions for a diverse set of problems and metrics. Hence, in this paper, we choose to focus on a *generic computational framework* leveraging state-of-the-art optimization techniques. Nonetheless, the geometric formalism considered here could be beneficial for the theoretical understanding of the "best" metric choice. Indeed, as mentioned in [31, sec. 15], local approximation of the loss function cannot explain all global properties of NGD. The metric in the ρ-space, on the other hand, can impact the global properties of f. More specifically, it might convexify f [13, Appendix B] or make it Lipschitz, paving a way towards the analysis of the NGD as a first-order method in the ρ-space. We find this line of research an intriguing future direction.

Finally, the full potential of randomized linear algebra techniques remains to be explored. We discuss a mini-batch version of our algorithm in subsection 3.5.2 and several low-rank approximation techniques in Appendices B.2 to B.4. Nevertheless, the success of randomized linear algebra techniques for very high-dimensional problems warrants a more thorough investigation of the theoretical and computational aspects of these techniques adapted to our setting.

Appendix A. Symbols and notations. See Table 3 for all the notations in sections 1 to 3.

Appendix B. Algorithmic details regarding numerical implementation. This section presents more details on the numerical implementation of the NGD methods. In particular, we explain how to obtain the matrix L in (3.2) for the WNGD (2.27) in Appendix B.1. We have proposed in subsection 3.2 that the QR factorization could efficiently solve the least-squares problem (3.2). In Appendix B.2, we discuss how to handle rank deficiency in $Y = LZ$ through the QR factorization.

The main difficulties of computing NGD for large-scale problems include no direct access to the Jacobian Z (see subsection 3.3) and the computational cost of handling Z even if it is directly available. Here, we present two interesting ideas that may mitigate these challenges, although we have not thoroughly investigated them in the context of NGD methods. We discuss in Appendix B.3 one strategy based on randomized linear algebra if the Jacobian Z is unavailable. In Appendix B.4, we briefly comment on an idea to further reduce the computational complexity of the NGD methods by possibly obtaining a low-rank approximation of the Jacobian Z.

B.1. More discussions on computing the Wasserstein natural gradient. As explained in subsection 2.5, the Wasserstein tangent vectors at ρ are velocity fields of minimal kinetic energy in $L^2_\rho(\mathbb{R}^d;\mathbb{R}^d)$. After a change of variable, $\tilde{v}_i = \sqrt{\rho}\, v_i$ and $\tilde{v}_i$ satisfies (2.26). We will discuss next how to solve this minimization problem numerically.

Discretization of the divergence operator. To compute the Wasserstein natural gradient, the first step is to solve (2.26), which becomes (B.1) after discretization.

$$\min_y \|y\|_2^2 \quad \text{s.t. } By = \zeta_i, \quad i = 1, \dots, p. \tag{B.1}$$

TABLE 3
Table of notations in sections 1 *to* 3.

Section 1	
θ	the unknown parameter
ρ	the state variable that depends on θ
$f(\rho)$	the loss function that depends on ρ
$(\mathcal{M}, d_\rho)$, (Θ, d_θ)	the metric space of ρ and θ, respectively
Section 2	
$(\mathcal{M}, g)$	the space $\mathcal{M}$ endowed with a Riemannian metric g
$T_\rho\mathcal{M}$	the tangent space of $\mathcal{M}$
p	the dimension of the parameter, $\theta \in \Theta \subseteq \mathbb{R}^p$
$\partial^g_{\theta_i}\rho(\theta) \in T_\rho\mathcal{M}$	the tangent vector of $\rho(\theta)$ with respect to θ_i based on the Riemannian geometry $(\mathcal{M}, g)$, $1 \le i \le p$
$\partial^g_\rho f(\rho) \in T_\rho\mathcal{M}$	the metric gradient of $f(\rho)$ with respect to ρ based on the Riemannian geometry $(\mathcal{M}, g)$
η^{nat}, η^{std}	the natural and standard gradient directions for θ
$\partial_\theta f(\rho(\theta))$	the gradient of $f(\rho(\theta))$ with respect to θ
$P\partial^g_\rho f$	the $\langle\cdot,\cdot\rangle_{g(\rho)}$-orthogonal projection of $-\partial^g_\rho f$ onto $\operatorname{span}\{\partial^g_{\theta_1}\rho, \dots, \partial^g_{\theta_p}\rho\}$
$G(\theta)$	the information matrix $G_{ij}(\theta) = \langle \partial^g_{\theta_i}\rho, \partial^g_{\theta_j}\rho\rangle_{g(\rho(\theta))}, i, j = 1, \dots, p$
$\zeta, \hat{\zeta}$	tangent vectors on $T_\rho\mathcal{M}$
$\zeta_i = \partial_{\theta_i}\rho$, $i = 1, \dots, p$	tangent vectors on the Euclidean space $(L^2(\mathbb{R}^d), \langle\cdot,\cdot\rangle_{L^2(\mathbb{R}^d)})$
$\partial_\rho f$	the metric gradient of $f(\rho)$ in $(L^2(\mathbb{R}^d), \langle\cdot,\cdot\rangle_{L^2(\mathbb{R}^d)})$
$G^{L^2}, G^{H^s}, G^{\dot{H}^s}, G^{FR}, G^W$	the information matrices for different Riemannian metrics
$\mathbf{D}^s$	a differential operator that outputs a vector of all the partial derivatives up to order s where $s \ge 0$
A^*, $A^\dagger$	the adjoint and the pseudoinverse of the linear operator A
$\chi, \hat{\chi}$	the tangent vectors in H^{-s} mapped from $\zeta, \hat{\zeta}$ in H^s, $s < 0$
$\triangle$	the Laplacian operator
$\widetilde{\mathbf{D}}^s$	a differential operator that outputs a vector of all the partial derivatives of positive order up to s where $s > 0$
$\mathcal{P}_2(\mathbb{R}^d)$	the set of Borel probability measures of finite second moments
$f_\sharp\rho$	the pushforward distribution of ρ by f
$\Gamma(\rho_1, \rho_2)$	the set of all measure $\pi \in \mathcal{P}(\mathbb{R}^{2d})$ with ρ_1 and ρ_2 as marginals
$v, \hat{v}, w, \{v_i\}_{i=1}^p$	the tangent vectors in $T_\rho\mathcal{P}_2(\mathbb{R}^d) \subset L^2_\rho(\mathbb{R}^d; \mathbb{R}^d)$
$\{\tilde{v}_i\}_{i=1}^p$	the renormalized Wasserstein tangent vectors, $\tilde{v}_i = \sqrt{\rho}\, v_i$
$\mathbf{B}$	the differential operator defined by $\mathbf{B}\tilde{v} = -\nabla\cdot(\sqrt{\rho(\theta)}\,\tilde{v})$
$\mathbf{B}_k$	a generalized version of $\mathbf{B}$ given by $\mathbf{B}_k\tilde{v} = -\nabla\cdot(\rho(\theta)^k\tilde{v})$
$\mathbf{L}$	with different choice of $\mathbf{L}$, all natural gradient directions can be formulated as $\eta^{nat} = \operatorname{argmin}_{\eta\in\mathbb{R}^p} \|(\mathbf{L}^*)^\dagger\partial_\rho f + \sum_{i=1}^p \eta_i\, \mathbf{L}\zeta_i\|^2_{L^2(\mathbb{R}^d)}$
Section 3	
$\rho \in \mathbb{R}^k$	the discretized state variable
$\partial_\rho f$, $Z = \partial_\theta\rho$	the finite-dimensional gradient and Jacobian in Euclidean space
L	the discretization of the operator $\mathbf{L}$ for different metric spaces
$G_L = Y^\top Y$	the discretized information matrix, $Y = LZ$
η^{nat}_L	the natural gradient direction in a unified framework (3.2)
$h(\rho, \theta) = \mathbf{0}$	the implicit dependence of ρ on θ
λ_ξ, λ	the adjoint variable, solutions to the adjoint equation

If the domain Ω is a compact subset of $\mathbb{R}^d$ (in terms of numerical discretization), the divergence operator in (2.26) comes with a zero-flux boundary condition. That is, $\tilde{v} = 0$ on $\partial\Omega$. For simplicity, we describe the case $d = 2$ where Ω is a rectangular cuboid. All numerical examples we present earlier in this paper belong to this scenario.

First, we discretize the domain $[\mathfrak{a},\mathfrak{b}]\times[\mathfrak{c},\mathfrak{d}]$ with a uniform mesh with spacing Δx and Δy such that $x_0=\mathfrak{a}$, $x_{n_x}=\mathfrak{b}$, $y_0=\mathfrak{c}$, and $y_{n_y}=\mathfrak{d}$. The left-hand side of the linear constraint in (2.26) becomes a matrix

$$B=-\begin{bmatrix}A_xD & A_yD\end{bmatrix}$$

in (B.1) where $D=\operatorname{diag}(\sqrt{\vec{\rho}})$, $A_x=\frac{1}{2\Delta x}C_{n_x-1}\otimes I_{n_y-1}$ and $A_y=\frac{1}{2\Delta y}I_{n_x-1}\otimes C_{n_y-1}$. Here, $\vec{\rho}$ is a vector-format discretization of the function ρ while skipping the boundary points, $\otimes$ denotes the Kronecker product, $I_n\in\mathbb{R}^{n\times n}$ is the identify matrix, and $C_n\in\mathbb{R}^{n\times n}$ is the central difference matrix with the zero-Dirichlet boundary condition.

$$C_n=\begin{bmatrix}0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1\\ & & & -1 & 0\end{bmatrix}_{n\times n}. \tag{B.2}$$

One may also use a higher-order discretization for the divergence operator in (2.26). The discretization of the vector field $\tilde{v}=(\tilde{v}_x,\tilde{v}_y)^\top$ is $y=(y_1^\top,y_2^\top)^\top$ in (B.1), where y_1 and y_2 are respectively the vector-format of $\tilde{v}_x$ and $\tilde{v}_y$ while skipping the boundary points due to the zero-flux boundary condition. Note that B is full rank if ρ is strictly positive, and n_x, n_y are odd. We remark that B and y remain very similar structures if $\Omega\subset\mathbb{R}^d$ with $d>2$.

***Z* available.** If $Z=(\zeta_1\ \zeta_2\ \dots\ \zeta_p)$ is available, we can solve (2.26) directly. After discretization, these equations reduce to constrained minimum-norm problems (B.1), where B is the discretization of the differential operator $-\nabla\cdot(\sqrt{\rho}\,\bullet)$ evaluated at the current θ (and thus $\rho(\theta)$). The solution to (B.1) can be recovered via the pseudoinverse of B as

$$Y=B^\dagger Z,\quad \text{where}\quad Y=(\tilde{v}_1\,\tilde{v}_2\,\dots\,\tilde{v}_p)\quad\text{and}\quad Z=(\zeta_1\,\zeta_2\,\dots\,\zeta_p). \tag{B.3}$$

In our case, B is underdetermined, and we assume it to have full row ranks. We could perform the QR decomposition of $B^\top$ in the "economic" size:

$$B^\dagger=Q(R^\top)^{-1},\quad\text{where}\quad B^\top=QR. \tag{B.4}$$

Since $R^\top$ is lower diagonal, $\tilde{v}_i=Q(R^\top)^{-1}\zeta_i$ can be efficiently calculated via forward substitution. If p is not too large, and we have access to $\{\zeta_i\}$ directly, this is an efficient way to obtain $\{\tilde{v}_i\}$.

Once we obtain Y, we can compute the W_2 NGD direction since (2.21) reduces to

$$\eta_{W_2}^{nat}=\operatorname*{argmin}_{\eta\in\mathbb{R}^p}\left\|\sqrt{\rho}\,\partial_\rho^W f+\sum_{i=1}^p\eta_i\tilde{v}_i\right\|^2_{L^2(\mathbb{R}^d;\mathbb{R}^d)}=-Y^\dagger\left(\sqrt{\rho}\,\partial_\rho^W f\right), \tag{B.5}$$

where $\partial_\rho^W f$ is related to $\partial_\rho f$ based on (2.22), and $Y^\dagger$ is the pseudoinverse of Y, which one can obtain by QR factorization; see details in subsection 3.2.

We can also compute the W_2 information matrix based on $B^\dagger$ obtained via the QR factorization (B.4). That is,

$$G_{w_2}=Y^\top Y=Z^\top(BB^\top)^\dagger Z=Z^\top(B^\dagger)^\top B^\dagger Z.$$

Therefore, if Y has full column ranks, the common approach is to invert the information matrix G_{w_2} directly and obtain the NGD direction following (2.4) as

$$\eta_{W_2}^{nat} = -G_{w_2}^{-1}\, \partial_\theta f(\rho(\theta)).$$

Discretization of the Wasserstein Gradient $\partial_\rho^W f$. Based on (2.27), we need to discretize the weighted Wasserstein Gradient, $b \approx -\sqrt{\rho}\partial_\rho^W f = -\sqrt{\rho}\nabla\partial_\rho f$, such that the WNGD $\eta_{W_2}^{nat} = Y^\dagger b$ where $Y = B^\dagger Z$. We remark that the discretization of the gradient operator in $\sqrt{\rho}\nabla\partial_\rho f(\rho(\theta))$ needs to be the numerical adjoint with respect to the matrix $-B$, the discretization of the divergence operator. That is,

$$b \approx -\sqrt{\rho}\nabla\left(\partial_\rho f(\rho(\theta))\right) = (-B)^\top \partial_\rho f.$$

This requirement is to ensure that

$$\begin{aligned}\partial_{\theta_j} f(\rho(\theta)) \approx \partial_\rho f^\top \zeta_j &= \partial_\rho f^\top B y_j \\ &= \left(B^\top \partial_\rho f\right)^\top y_j = -b^\top y_j \approx \langle \sqrt{\rho}\nabla\partial_\rho f, \sqrt{\rho} v_j\rangle_{L^2(\mathbb{R}^d;\mathbb{R}^d)},\end{aligned}$$

which is the discrete version of

$$\begin{aligned}\lim_{t\to 0}\frac{f(\rho + t\zeta) - f(\rho)}{t} &= \int_{\mathbb{R}^d} \partial_\rho f(\rho)(x)\,\zeta(x)dx \\ &= \int_{\mathbb{R}^d} \sqrt{\rho}\nabla\partial_\rho f(\rho)(x)\cdot\tilde{v}(x)dx \quad \forall \zeta \in L^2(\mathbb{R}^d).\end{aligned}$$

The equation above is the main identity used in the proof for Proposition 2.2.

For example, if we use the central difference scheme for the divergence operator $-\nabla\cdot(\sqrt{\rho}\ \bullet)$, we also need to use central difference for the gradient operator ∇. Similarly, if one uses forward difference for $-B$, the backward difference should be employed for the gradient operator ∇.

B.2. Dealing with rank deficiency. Note that in (3.2), we need to solve a least-squares problem given the matrix $Y = LZ$ to find the NGD direction based upon a wide range of Riemannian metric spaces. For simplicity, we will consider the problem in its general form: finding the least-squares solution η to $Y\eta = b$, where $b = -(L^\top)^\dagger \partial_\rho f$ based on (3.2).

The standard QR approach only applies if Y has full column rank, i.e., $\operatorname{rank}(Y) = p$ while $Y \in \mathbb{R}^{k\times p}$. Otherwise, if $\operatorname{rank}(Y) = r < p$, we are facing a rank-deficient problem, and an alternative has to be applied. Even if Y is full rank, sometimes we may have a nearly rank-deficient problem when the singular values of Y, $\{\sigma_i\}$, $i = 1,\dots,p$, decay too quickly such that $\sigma_{r+1},\dots,\sigma_p \ll \sigma_r$. A conventional way to deal with such situations is via QR factorization with column pivoting.

In order to find and then eliminate unimportant directions of Y, essentially, we need a *rank-revealing* matrix decomposition of Y. While SVD (singular value decomposition) might be the most common choice, it is relatively expensive, which motivated various works on rank-revealing QR factorization as they take fewer flops (floating-point operations) than SVD. The column pivoted QR (CPQR) decomposition is one of the most popular rank-revealing matrix decompositions [16]. We remark that CPQR can be easily implemented in MATLAB and Python through the standard `qr` command, which is based upon LAPACK in both languages [6].

Applying CPQR to Y yields

$$YP = QR,$$

where P is the permutation matrix. Thus, the linear equation $Y\eta = b$ becomes

$$YPP^\top\eta = QRP^\top\eta = QR\eta_p = b, \quad \text{where } \eta_p = P^\top\eta.$$

Now, we denote by $\widetilde{Q}$ and $\widetilde{R}$ the truncated versions of Q and R, respectively, by keeping the first r columns of Q and the first r rows of R. We may solve the linear system below instead:

$$\widetilde{R}\eta_p = \widetilde{Q}^\top b.$$

The least-squares solution is no longer unique since we have truncated R due to the (nearly) rank deficiency of Y. By convention, one may pick the one with the minimum norm among all the least-squares solutions. Since $\|\eta\|_2 = \|\eta_p\|_2$ as P is a permutation matrix, this is equivalent to finding a minimum-norm solution to the above linear system. This can be done by an additional QR factorization. Let

$$\widetilde{R}^\top = Q_1 R_1,$$

where $Q_1 \in \mathbb{R}^{p\times r}$ has orthonormal columns and $R_1 \in \mathbb{R}^{r\times r}$ is invertible. As a result,

$$\eta_p = Q_1(R_1^\top)^{-1}\widetilde{Q}^\top b.$$

Finally, we may obtain the solution

$$\eta = P\eta_p = PQ_1(R_1^\top)^{-1}\widetilde{Q}^\top b.$$

Again, $(R_1^\top)^{-1}$ should be understood as forward substitution.

We may apply the same idea if B in (B.1) is (nearly) rank deficient, while we will keep its dominant r ranks. Note that B is short-wide. Applying CPQR to $B^\top$ yields

$$B^\top P = QR,$$

where P is the permutation matrix, Q has orthonormal columns, and R is a $p \times p$ square matrix. Thus, the constraint in (B.1) becomes

$$PP^\top By = PR^\top Q^\top y = \zeta_i.$$

Again, we denote by $\widetilde{Q}$ and $\widetilde{R}$ the truncated version of Q and R by keeping the first r columns of Q and the first r rows of R where $r \le p$. We may solve the linear system below instead:

$$\widetilde{R}^\top y_q = P^\top\zeta_i, \quad \text{where } y_q = \widetilde{Q}^\top y.$$

Since $\widetilde{R}^\top$ is tall-skinny, we may select the least-squares solution to the above system. We perform a QR decomposition in economic size for $\widetilde{R}^\top$ such that $\widetilde{R}^\top = Q_2R_2$. Therefore,

$$y_q = R_2^{-1}Q_2^\top P^\top\zeta_i$$

and eventually leads to

$$\tilde{v}_i = y = \widetilde{Q}y_q = \widetilde{Q}R_2^{-1}Q_2^\top P^\top\zeta_i.$$

Note that if $\widetilde{R} = R$ and $\widetilde{Q} = Q$, i.e., $r = p$, the solution above coincides with the one obtained from (B.3)–(B.4) since $R_2^{-1}Q_2^\top = (R^\top)^{-1}$.

To sum up, for a tall-skinny matrix Y, we compute the following by two QR factorizations while eliminating the unimportant directions during the process:

$$YP = \widetilde{Q} R_1^\top Q_1^\top,$$

where R_1 is a invertible square matrix while $\widetilde{Q}$ and Q_1 have orthonormal columns. Therefore,

$$Y^\dagger = P Q_1 (R_1^\top)^{-1} \widetilde{Q}^\top.$$

Finally, $\eta = Y^\dagger b = P Q_1 (R_1^\top)^{-1} \widetilde{Q}^\top b$. For a short-wide matrix B, we compute

$$B^\top P = \widetilde{Q} R_2^\top Q_2^\top,$$

where R_2 is invertible while $\widetilde{Q}$ and Q_2 have orthonormal columns. Consequently,

$$B^\dagger = \widetilde{Q} R_2^{-1} Q_2^\top P^\top.$$

Finally, $\tilde{v}_i = B^\dagger \zeta_i = \widetilde{Q} R_2^{-1} Q_2^\top P^\top \zeta_i$ for $i = 1, \dots, p$.

B.3. Z not available: The Hutchinson method. In this subsection, we present some ideas for approximating Z using Hutchinson's estimator [17, 35, 51], a powerful technique from randomized linear algebra. Let $\xi \in \mathbb{R}^k$ be a vector with i.i.d. random coordinates of mean 0 and variance 1. Such random vectors serve as a random basis. That is,

$$Z = \mathbb{E}\left[\xi \xi^\top Z\right].$$

Thus, if we have m such random vectors, $\xi_1, \xi_2, \dots, \xi_m$, then we can estimate

$$H_m(Z) = \frac{1}{m} \sum_{k=1}^{m} \xi_k \xi_k^\top Z.$$

Furthermore, by introducing the adjoint variables $\lambda_1, \lambda_2, \dots, \lambda_m$ such that

$$\lambda_k^\top \partial_\rho h = \xi_k^\top, \quad 1 \le k \le m, \tag{B.6}$$

and using (3.8), we obtain

$$H_m(Z) = -\frac{1}{m} \sum_{k=1}^{m} \xi_k \lambda_k^\top \partial_\theta h.$$

Hence, by replacing Z in (3.2) with its approximation $H_m(Z)$, we obtain an approximated NGD direction as

$$\eta_L^{nat} = \underset{\eta \in \mathbb{R}^p}{\operatorname{argmin}} \left\| (L^\top)^\dagger \partial_\rho f + L \; H_m(Z) \; \eta \right\|_2^2. \tag{B.7}$$

Once we obtain $H_m(Z)$, the above least-squares problem can be solved by QR factorization, similar to the framework presented in subsection 3.2 or Appendix B.2. However, we remark here that the convergence behavior of $H_m(Z) \xrightarrow{m \to \infty} Z$ depends on the spectral properties of Z.

B.4. Exploring the column space of Z implicitly. As discussed in Appendix B.3, one way to reduce the complexity of implementing the NGD method is to find a low-rank approximation to the Jacobian $Z = \partial_\theta \rho$. For any ζ, we have that $\zeta = \mathbb{E}[\langle \zeta, \xi \rangle \xi]$ given any random vector ξ whose covariance is the identity. Hence, by the law of large numbers, for m large enough, we have that

$$\mathbb{P}\left(\left\| \zeta - \hat{\zeta} \right\| > \epsilon \right) < \delta, \quad \text{where } \hat{\zeta} = \frac{1}{m} \sum_{k=1}^{m} \langle \zeta, \xi_k \rangle \xi_k, \tag{B.8}$$

where $\{\xi_1, \xi_2, \ldots, \xi_m\}$ are i.i.d. random vectors. Therefore,

$$\left\| L\left(\zeta_j - \hat{\zeta}_j \right) \right\| < \|L\| \epsilon, \quad 1 \le j \le p,$$

with high probability when m is large enough (depending on the spectral property of Z). Here, L is the important linear operator in the unified framework (3.2). In Appendix B.3, we approximate

$$Y = LZ \approx L H_m(Z),$$

which is to compute the approximation matrix $H_m(Z)$ directly. Next, we present another way to obtain an approximated Y whether or not Z is explicitly available.

If we can find such $\{\xi_k\}$ satisfying (B.8), our final approximation to each y_j in $Y = LZ = (y_1 \ldots y_j \ldots, y_p)$ could be written as

$$y_j = L\zeta_j \approx L\hat{\zeta}_j = \frac{1}{m} \sum_{k=1}^{m} \langle \zeta_j, \xi_k \rangle \, L\xi_k, \quad 1 \le j \le p. \tag{B.9}$$

Note that the inner product $\langle \zeta_j, \xi_k \rangle$ can be computed via the adjoint-state method if there is no direct access to $\{\zeta_j\}$; see section 3.3.1 for details. Therefore, to obtain an approximated Y, we only need to evaluate Lh_k and the inner products $\langle \zeta_j, \xi_k \rangle$ for each k and j, without directly accessing the Jacobian $Z = (\zeta_1 \ldots \zeta_p)$. A similar idea called randomized SVD could also apply here [15].

REFERENCES

[1] S.-I. Amari, *Differential-Geometrical Methods in Statistics*, Lecture Notes in Statistics 28, Springer, New York, 1985.

[2] S.-I. Amari, *Natural gradient works efficiently in learning*, Neural Comput., 10 (1998), pp. 251–276.

[3] S.-I. Amari and A. Cichocki, *Adaptive blind signal processing-neural network approaches*, Proc. IEEE, 86 (1998), pp. 2026–2048.

[4] L. Ambrosio, E. Brué, and D. Semola, *Lectures on Optimal Transport*, Springer, New York, 2021.

[5] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, 2nd ed., Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2008.

[6] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., Software Environ. Tools 9, SIAM, Philadelphia, 1999, https://doi.org/10.1137/1.9780898719604.

[7] M. Arbel, A. Gretton, W. Li, and G. Montufar, *Kernelized Wasserstein natural gradient*, in International Conference on Learning Representations, 2020.

[8] Y. Chen and W. Li, *Optimal transport natural gradient for statistical manifolds with continuous sample space*, Inform. Geom., 3 (2020), pp. 1–32.

[9] T. A. Davis, *Algorithm* 915*, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization*, ACM Trans. Math. Softw. (TOMS), 38 (2011), pp. 1–22.
[10] B. Engquist and Y. Yang, *Seismic inversion and the data normalization for optimal transport*, Methods Appl. Anal., 26 (2019), pp. 133–147.
[11] B. Engquist and Y. Yang, *Optimal transport based seismic inversion: Beyond cycle skipping*, Comm. Pure Appl. Math., 75 (2022), pp. 2201–2244.
[12] D. Fortunato and A. Townsend, *Fast Poisson solvers for spectral methods*, IMA J. Numer. Anal., 40 (2020), pp. 1994–2018.
[13] W. Gangbo and A. R. Mészáros, *Global well-posedness of master equations for deterministic displacement convex potential mean field games*, Comm. Pure Appl. Math., 75 (2022), pp. 2685–2801.
[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996.
[15] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.
[16] N. D. Heavner, *Building Rank-Revealing Factorizations with Randomization*, Ph.D. thesis, University of Colorado at Boulder, 2019.
[17] M. Hutchinson, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Commun. Stat. Simul. Comput., 19 (1990), pp. 433–450.
[18] M. Jacobs, W. Lee, and F. Léger, *The back-and-forth method for Wasserstein gradient flows*, ESAIM Control Optim. Calc. Var., 27 (2021), 28.
[19] M. Jacobs and F. Léger, *A fast approach to optimal transport: The back-and-forth method*, Numer. Math., 146 (2020), pp. 513–544.
[20] M. Jacobs, F. Léger, W. Li, and S. Osher, *Solving large-scale optimization problems with a convergence rate independent of grid size*, SIAM J. Numer. Anal., 57 (2019), pp. 1100–1123, https://doi.org/10.1137/18M118640X.
[21] W. Li, A. T. Lin, and G. Montúfar, *Affine natural proximal learning*, in Geometric Science of Information, F. Nielsen and F. Barbaresco, eds., Springer, Cham, 2019, pp. 705–714.
[22] W. Li, S. Liu, H. Zha, and H. Zhou, *Parametric Fokker-Planck equation*, in Geometric Science of Information, F. Nielsen and F. Barbaresco, eds., Springer, Cham, 2019, pp. 715–724.
[23] W. Li and G. Montúfar, *Natural gradient via optimal transport*, Inf. Geom., 1 (2018), pp. 181–214.
[24] W. Li and J. Zhao, *Wasserstein Information Matrix*, preprint, arXiv:1910.11248, 2019.
[25] A. T. Lin, W. Li, S. Osher, and G. Montúfar, *Wasserstein proximal of GANs*, in Geometric Science of Information, F. Nielsen and F. Barbaresco, eds., Springer, Cham, 2021, pp. 524–533.
[26] S. Lisini, D. Matthes, and G. Savaré, *Cahn-Hilliard and thin film equations with nonlinear mobility as gradient flows in weighted-Wasserstein metrics*, J. Differential Equations, 253 (2012), pp. 814–850.
[27] S. Liu, M. Jacobs, W. Li, L. Nurbekyan, and S. J. Osher, *Computational methods for first-order nonlocal mean field games with applications*, SIAM J. Numer. Anal., 59 (2021), pp. 2639–2668, https://doi.org/10.1137/20M1334668.
[28] S. Liu, W. Li, H. Zha, and H. Zhou, *Neural parametric Fokker–Planck equation*, SIAM J. Numer. Anal., 60 (2022), pp. 1385–1449, https://doi.org/10.1137/20M1344986.
[29] S. Liu and L. Nurbekyan, *Splitting methods for a class of non-potential mean field games*, J. Dyn. Games, 8 (2021), pp. 467–486.
[30] A. Mallasto, T. D. Haije, and A. Feragen, *A formalization of the natural gradient method for general similarity measures*, in Geometric Science of Information, F. Nielsen and F. Barbaresco, eds., Springer, Cham, 2019, pp. 599–607.
[31] J. Martens, *New insights and perspectives on the natural gradient method*, J. Mach. Learn. Res., 21 (2020), pp. 1–76.
[32] J. Martens and R. Grosse, *Optimizing neural networks with Kronecker-factored approximate curvature*, in International Conference on Machine Learning, PMLR, 2015, pp. 2408–2417.
[33] J. Martens and I. Sutskever, *Training deep and recurrent networks with Hessian-free optimization*, in Neural Networks: Tricks of the Trade, Springer, New York, 2012, pp. 479–535.
[34] L. Métivier, R. Brossier, J. Virieux, and S. Operto, *Full waveform inversion and the truncated Newton method*, SIAM J. Sci. Comput., 35 (2013), pp. B401–B437, https://doi.org/10.1137/120877854.
[35] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, *Hutch++: Optimal stochastic trace estimation*, in Symposium on Simplicity in Algorithms (SOSA), SIAM, Philadelphia, 2021, pp. 142–155, https://doi.org/10.1137/1.9781611976496.16.

[36] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547 (in Russian).
[37] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
[38] R. Pascanu and Y. Bengio, *Revisiting Natural Gradient for Deep Networks*, preprint, arXiv:1301.3584, 2013.
[39] J. Peters and S. Schaal, *Natural actor-critic*, Neurocomputing, 71 (2008), pp. 1180–1190.
[40] R.-E. Plessix, *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*, Geophys. J. Int., 167 (2006), pp. 495–503.
[41] N. Qian, *On the momentum term in gradient descent learning algorithms*, Neural Netw., 12 (1999), pp. 145–151.
[42] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707.
[43] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[44] N. N. Schraudolph, *Fast curvature matrix-vector products for second-order gradient descent*, Neural Comput., 14 (2002), pp. 1723–1738.
[45] Z. Shen, Z. Wang, A. Ribeiro, and H. Hassani, *Sinkhorn natural gradient for generative models*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 1646–1656.
[46] C. Villani, *Topics in Optimal Transportation*, Grad. Stud. Math. 58, AMS, Providence, RI, 2003.
[47] J. Virieux and S. Operto, *An overview of full-waveform inversion in exploration geophysics*, Geophysics, 74 (2009), pp. WCC1-WCC26.
[48] P. Xu, F. Roosta, and M. W. Mahoney, *Second-order optimization for non-convex machine learning: An empirical study*, in Proceedings of the 2020 SIAM International Conference on Data Mining, SIAM, Philadelphia, 2020, pp. 199–207, https://doi.org/10.1137/1.9781611976236.23.
[49] H. H. Yang and S.-i. Amari, *Complexity issues in natural gradient descent method for training multilayer perceptrons*, Neural Comput., 10 (1998), pp. 2137–2157.
[50] Y. Yang, A. Townsend, and D. Appelö, *Anderson acceleration based on the* $\mathcal{H}^{-s}$ *Sobolev norm for contractive and noncontractive fixed-point operators*, J. Comput. Appl. Math., 403 (2022), 113844.
[51] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. Mahoney, *ADAHESSIAN: An adaptive second order optimizer for machine learning*, in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 10665–10673.
[52] L. Ying, *Natural gradient for combined loss using wavelets*, J. Sci. Comput., 86 (2021), pp. 1–10.
[53] A. Yu, Y. Yang, and A. Townsend, *A Quadrature Perspective on Frequency Bias in Neural Network Training with Nonuniform Data*, preprint, arXiv:2205.14300, 2022.
[54] G. Zhang, J. Martens, and R. B. Grosse, *Fast convergence of natural gradient descent for over-parameterized neural networks*, in Advances in Neural Information Processing Systems 32, Curran Associates, Red Hook, NY, 2019.
[55] B. Zhu, J. Hu, Y. Lou, and Y. Yang, *Implicit Regularization Effects of the Sobolev Norms in Image Processing*, preprint, arXiv:2109.06255, 2021.