Accelerated Natural Gradient Method for Parametric Manifold Optimization*

Chenyi Li[†]   Shuchen Zhu[‡]   Zhonglin Xie[§]   Zaiwen Wen[¶]

April 9, 2025

Abstract

Parametric manifold optimization problems frequently arise in various machine learning tasks, where state functions are defined on infinite-dimensional manifolds. We propose a unified accelerated natural gradient descent (ANGD) framework to address these problems. By incorporating a Hessian-driven damping term into the manifold update, we derive an accelerated Riemannian gradient (ARG) flow that mitigates oscillations. An equivalent first-order system is further presented for the ARG flow, enabling a unified discretization scheme that leads to the ANGD method. In our discrete update, our framework considers various advanced techniques, including least squares approximation of the update direction, projected momentum to accelerate convergence, and efficient approximation methods through the Kronecker product. It accommodates various metrics, including H^s, Fisher-Rao, and Wasserstein-2 metrics, providing a computationally efficient solution for large-scale parameter spaces. We establish a convergence rate for the ARG flow under geodesic convexity assumptions. Numerical experiments demonstrate that ANGD outperforms standard NGD, underscoring its effectiveness across diverse deep learning tasks.

Keywords— Natural gradient, Parametric manifold, Riemannian optimization, Accelerated flow, Machine learning

1 Introduction

We focus on the parametric optimization problem of the form

$$\min_{\theta \in \mathbb{R}^p} L(\rho_\theta), \tag{1}$$

where $\rho_{(\cdot)} : \Theta \to \mathcal{M}$ is a mapping from the parameter space $\Theta \subseteq \mathbb{R}^p$ to the state space $\mathcal{M}$, which inherently exhibits the structure of an infinite-dimensional Riemannian manifold. The loss functional is given as $L(\cdot) : \mathcal{M} \to \mathbb{R}$. Classical Riemannian optimization treats θ as residing on a specific manifold, while in parametric manifold optimization, ρ lies on the manifold $\mathcal{M}$ and is parameterized by θ. Typically, ρ_θ is produced by a deep neural network with input $x \in \mathbb{R}^d$ and parameters $\theta \in \mathbb{R}^p$. This framework encompasses a wide range of machine learning and data analysis problems, where $\mathcal{M}$ can be specialized to function spaces such as Sobolev spaces H^s or probability measures $\mathcal{P}(\mathbb{R}^p)$. Two classic examples of this problem are the physics-informed neural networks (PINNs) [22, 23] and variational Monte Carlo (VMC) methods [21] for solving Schrödinger equations.

[†]School of Mathematical Sciences, Peking University, China (lichenyi@stu.pku.edu.cn).

[‡]Center for Data Science, Peking University, China (shuchenzhu@stu.pku.edu.cn).

[§]Beijing International Center for Mathematical Research, Peking University, China (zlxie@pku.edu.cn).

[¶]Corresponding author. Beijing International Center for Mathematical Research, Peking University, China (wenzw@pku.edu.cn).

1.1 Literature Review

Gradient flow is helpful in the analysis and design of accelerated optimization algorithms. The continuous limit of the Nesterov's acceleration method [19] is known as the accelerated gradient flow [26]. A Hessian-driven damping term is incorporated into high-resolution ODE in [25]. A convergence analysis of accelerated ODE with Hessian-driven damping is provided in [5]. The extension of accelerated gradient flow on Riemannian manifolds has been extensively studied. Gradient flow on the manifold is introduced to analyze the Riemannian Nesterov accelerated gradient method in [1]. Accelerated information gradient flow [29] extends it to infinite-dimensional Riemannian manifolds. Minimization of the KL divergence using the Nesterov acceleration method from a continuous perspective is explored in [14]. The Wasserstein-2 gradient flow and its parametrization are studied in [12, 30].

Implementing the traditional gradient descent method for high-dimensional nonconvex problems (1) is a significant challenge. For PDE-based optimization problems, traditional gradient descent algorithms suffer from pathologies [28, 24]. Natural gradient methods are often used to enhance the performance of the gradient descent algorithm by incorporating local curvature information, leading to faster convergence [15, 32, 6]. The natural gradient method pulls back the curvature of θ to the manifold [2]. This adjustment allows for more efficient search directions, overcoming some of the limitations of traditional gradient descent algorithm, such as slow convergence or getting stuck in local minima. Several modifications of the natural gradient method have been explored. The adaptively regularized natural gradient method [31] introduces an adaptive damping term. General natural gradient descent methods for PDE-based problems are discussed in [20], which formulates least squares problems for general metrics. The energy natural gradient method [17] is proposed to accelerate the PINNs training from the perspective of an energy metric.

Recent advancements in natural gradient methods for probability distributions have largely focused on the Fisher-Rao and Wasserstein-2 metrics. The computation of the natural gradient direction for the Fisher-Rao metric is reformulated as a least squares problem in [8]. This approach is enhanced by incorporating the projected momentum [9] (also known as the Kaczmarz method) to refine the solution. In the realm of Wasserstein-2 natural gradient methods, a computationally efficient modification of the Wasserstein information matrix is introduced by [12]. Additionally, Arbel et al. estimate the natural gradient by solving a regularized mini-max problem [3], whereas the KL divergence is used to approximate the Wasserstein-2 natural gradient [18], thereby avoiding the intractable computation of the Wasserstein information matrix.

Various approximation methods have been proposed to reduce the computational cost of the natural gradient methods. KFAC [16, 10] approximates the layer-wise information matrix using the Kronecker decomposition, allowing its inverse to be computed efficiently as the product of the inverses of these smaller matrices. Sketch-based methods [33] reduce the computational cost of matrix multiplication and inversion by sampling rows or columns, enabling efficient approximations of natural gradient. The quasi-natural gradient method [11] integrates LBFGS with natural gradient descent method to improve efficiency in statistical learning problems. Layer-wise block-diagonal approximations [6] are used to approximate the information matrix.

1.2 Our Contributions

In this paper, we propose a new accelerated natural gradient method to solve (1). Our main contributions are as follows.

- We derive the natural gradient descent algorithm from a novel two-stage optimization process. The first stage identifies an update flow on the manifold, while the second projects this flow onto the parameter space for a discrete update scheme. This separation allows for distinct designs of the natural gradient method in the manifold and parameter space. Though taking the metric of the manifold into consideration [15], traditional natural gradient methods do not seek acceleration for the first stage. We introduce an accelerated Riemannian gradient (ARG) flow to address this. A Hessian-driven damping term, designed to mitigate oscillations and accelerate convergence, is incorporated into the analysis of manifold-accelerated gradient flows for the first time. This ODE generalizes different first-order acceleration methods with different choices of hyper-parameters. Theoretical analysis shows that the ARG flow achieves a convergence rate of $\mathcal{O}(t^{-2})$ under geodesic convexity, strictly generalizing Euclidean acceleration mechanisms while maintaining their characteristic decay properties.

- We present a novel algorithmic framework that unifies projection across different metric spaces in the second stage. This framework discretizes the update quantity of the ARG flow in the tangent space of the manifold and maps it to the parameter space, resulting in a unified accelerated natural gradient descent (ANGD) method. Consequently, the ANGD method achieves faster convergence compared to traditional natural gradient descent methods [20]. Our approach generalizes several established methods, including the least squares approximation [33], projected momentum [9], and KFAC [16]. These methods provide increased flexibility in selecting preconditioners and eliminate the need to estimate metric-specific information matrices, thus improving computational efficiency, particularly for metrics with intractable information matrices, such as the H^s metric (for $s < 0$) and the Wasserstein-2 metric. Numerical experiments demonstrate the substantial acceleration of the ANGD method over non-accelerated natural gradient methods.

1.3 Notation

We use $\langle \cdot, \cdot \rangle$ to denote the inner product in Euclidean space. The spatial and parameter gradients are represented by ∇ and ∂_θ, respectively. For a time-varying map and input, we abbreviate $\partial_t f_t(x)\big|_{x=x_t}$ as $\partial_t f_t(x_t)$. The divergence of a vector-valued function F is written as $\nabla \cdot F$. The symbol $\circ$ is used to indicate the composition of two maps. In integrals, we omit the differential notation dx when there is no risk of ambiguity. We utilize $x[i]$ to refer to the i-th coordinate of the vector x. Given a vector $v \in \mathbb{R}^n$, its center value is denoted by $\bar{v} = v - \sum_{i=1}^n v[i]/n$, where the subtraction is applied element-wise.

1.4 Organization

The rest of this paper is organized as follows. In Section 2, we introduce the ARG flow with Hessian drive damping. The general discretization scheme and approximation methods on the parameter space are discussed in Section 3. Four kinds of specific metrics are considered in Section 4. In Section 5, we give the convergence analysis for the ARG flow. Finally, we show the performance of the ANGD algorithms with different numerical experiments in Section 6.

2 A Continuous-time Model for Accelerating NGD

In this section, we derive an accelerated gradient flow featuring Hessian-driven damping on the manifold $\mathcal{M}$. The trajectory of this flow exhibits a faster convergence rate and improved convergence properties compared with gradient flow. This continuous-time model serves as an ideal template for developing accelerated natural gradient descent methods on manifolds. Before proceeding, we provide a brief review of some basic concepts related to the Riemannian metric.

2.1 Background on Riemannian metric

We begin by outlining the definition of Riemannian metrics on an (infinitely dimensional) Riemannian manifold. Let $\mathcal{M}$ be a set of functions that are defined on Ω, a region in $\mathbb{R}^d$. The tangent space of $\mathcal{M}$ at $\rho(x) \in \mathcal{M}$ is defined as

$$T_\rho \mathcal{M} = \{\partial_t \rho_t(x)|_{t=0} : \rho_t(x) : (-1, 1) \times \Omega \to \mathbb{R} \text{ is a smooth curve in } \mathcal{M} \text{ and } \rho_0 = \rho\}.$$

In this section, we omit the variable x of the ρ function when there is no risk of ambiguity. The cotangent space $T_\rho^* \mathcal{M}$ is the dual space of $T_\rho \mathcal{M}$, consisting of linear functionals acting on $T_\rho \mathcal{M}$ defined via the L^2 inner product. The metric tensor $\mathcal{G}(\rho) : T_\rho \mathcal{M} \to T_\rho^* \mathcal{M}$ is an invertible and Hermitian linear operator. It satisfies the following properties: 1) $\int \sigma_1 \mathcal{G}(\rho) \sigma_2 dx = \int \sigma_2 \mathcal{G}(\rho) \sigma_1 dx$ for all $\sigma_1, \sigma_2 \in T_\rho \mathcal{M}$; 2) $\int \sigma \mathcal{G}(\rho) \sigma dx \geq 0$ for all $\sigma \in T_\rho \mathcal{M}$, with equality holding if and only if $\sigma \equiv 0$. We then define the Riemannian metric on $T_\rho \mathcal{M}$ as

$$g_\rho(\sigma_1, \sigma_2) = \int \sigma_1 \mathcal{G}(\rho) \sigma_2 dx = \int \Phi_1 \mathcal{G}(\rho)^{-1} \Phi_2 dx, \ \forall \sigma_1, \sigma_2 \in T_\rho \mathcal{M},$$

where $\Phi_i = \mathcal{G}(\rho)\sigma_i$. Given a metric, we can define the Riemannian distance between two elements $\rho_0, \rho_1 \in \mathcal{M}$ as $\text{dist}(\rho_0, \rho_1)^2 = \inf \int_0^1 g_{\rho_t}(\partial_t \rho_t, \partial_t \rho_t) dt$, where the infimum is taken over all smooth curves $\rho_t : [0, 1] \to \mathcal{M}$ connecting ρ_0 and ρ_1.

Then we extend the concept of the Riemannian gradient from finite-dimensional Riemannian manifolds to the infinite-dimensional case following [7]. We introduce some basic notation about calculus variations. The L^2 first variation for a functional $L(\rho) : L^2 \to \mathbb{R}$ with respect to ρ is as $\frac{\delta L}{\delta \rho}$, which is defined as the function such that $\lim_{\epsilon \to 0} \frac{L(\rho + \epsilon \sigma) - L(\rho)}{\epsilon} = \int \frac{\delta L}{\delta \rho} \sigma$, holds for any $\sigma \in L^2$. For any given $\rho_0 \in L^2$, we denote $\left. \frac{\delta L}{\delta \rho} \right|_{\rho = \rho_0}$ by $\frac{\delta L}{\delta \rho_0}$ when there is no ambiguity. The first order variation with respect to a function $h(\rho)(x) : L^2 \times \mathbb{R}^d \to \mathbb{R}$ is given as $\frac{\delta h}{\delta \rho}(x, y) = \frac{\delta}{\delta \rho} \int h(y) \delta(x - y) dy$, where δ is the Dirac delta function. We denote the canonical action of a tangent field $h(\rho)(x)$ to a smooth functional $L(\rho)$ as $h \circ L(\rho) = \int \frac{\delta E}{\delta \rho}(x) \cdot h(\rho)(x) dx$.

Further, we define the directional variational derivative of a smooth linear mapping $\mathcal{A}(\rho) : T_\rho \mathcal{M} \to T_\rho^* \mathcal{M}$. For any $\sigma \in T_\rho \mathcal{M}$, the directional variational derivative $\frac{\partial \mathcal{A}(\rho)}{\partial \rho} \cdot \sigma : T_\rho \mathcal{M} \to T_\rho^* \mathcal{M}$ is defined as $\left[\frac{\partial \mathcal{A}(\rho)}{\partial \rho} \cdot \sigma \right] \tau = \lim_{\epsilon \to 0} \frac{\mathcal{A}(\rho + \epsilon \sigma)\tau - \mathcal{A}(\rho)\tau}{\epsilon}$, for any fixed $\tau \in T_\rho \mathcal{M}$. This definition can be naturally extended to smooth linear mappings $\mathcal{A}(\rho) : T_\rho^* \mathcal{M} \to T_\rho \mathcal{M}$. The Riemannian gradient of a smooth functional $L(\rho)$ on $\mathcal{M}$ is given as $\text{grad} L(\rho) = \mathcal{G}(\rho)^{-1} \frac{\delta L}{\delta \rho}$.

The objects ρ in this paper fall into two types, with their tangent and cotangent spaces exhibiting different properties.

1. ρ **as a PDE-based model.** In this case, we primarily consider $\rho \in H^s(\Omega)$ (with s being an integer), where Ω is a bounded open domain or the entire Euclidean space $\mathbb{R}^d$. The Sobolev space $\mathcal{M} = H^s(\Omega)$ is a Hilbert space, with the tangent space $T_\rho \mathcal{M} = H^s(\Omega)$ and the cotangent space $T_\rho^* \mathcal{M} = L^2(\Omega)$.

2. ρ **as a probability distribution.** Define the set of smooth probability densities on Ω as $\mathcal{M} = \mathcal{P}(\Omega) = \left\{ \rho \in \mathcal{C}^\infty(\Omega) : \int_\Omega \rho dx = 1, \rho \geq 0 \right\}$, where Ω is an open set in $\mathbb{R}^d$. Here, $\mathcal{C}^\infty(\Omega)$ denotes the set of smooth functions defined on Ω. In this case, it holds that $T_\rho \mathcal{M} = \left\{ \sigma \in \mathcal{C}^\infty(\Omega) : \int \sigma dx = 0 \right\}$, $T_\rho^* \mathcal{M} = \mathcal{C}^\infty(\Omega)/\mathbb{R}$. For this case, we mainly consider the Fisher-Rao and Wasserstein-2 metric.

2.2 Accelerated Gradient Flow in Riemannian Manifolds

We start with an optimization problem in finite-dimensional Euclidean space:

$$\min_{\varrho \in \mathbb{R}^p} \ell(\varrho), \tag{2}$$

where ℓ is a differentiable function. The gradient descent method updates ϱ by following the steepest update direction, with the continuous counterpart described by $\dot{\varrho}_t + \nabla \ell(\varrho_t) = 0$. Nesterov proposed an accelerated method (NAG) to improve the efficiency of gradient descent method for convex functions [19]. A second-order ODE has been proposed to elucidate the acceleration mechanism corresponding to NAG in [26] as $\ddot{\varrho}_t + \alpha_t \dot{\varrho}_t + \nabla \ell(\varrho_t) = 0$. A Hessian-driven damping term is introduced to the continuous Nesterov acceleration ODE by [4, 5], which takes the form:

$$\ddot{\varrho}_t + \alpha_t \dot{\varrho}_t + \beta_t \nabla^2 \ell(\varrho_t) \dot{\varrho}_t + \gamma_t \nabla \ell(\varrho_t) = 0, \tag{3}$$

where $\alpha_t, \beta_t, \gamma_t$ are non-negative functions only depending on t. The Hessian-driven damping term $\nabla^2 \ell(\varrho_t) \dot{\varrho}_t$ facilitates faster convergence and reduces oscillations.

We extend these concepts to Riemannian manifolds. The Riemannian gradient flow of the target functional L on Riemannian manifold $\mathcal{M}$ is defined as

$$\partial_t \rho_t = -\text{grad} L(\rho_t) = -\mathcal{G}(\rho)^{-1} \frac{\delta L}{\delta \rho}, \tag{4}$$

where $\rho_t : [t_0, \infty) \to \mathcal{M}$ is a smooth curve on the manifold. For the accelerated gradient flow on Riemannian manifolds, note that $\ddot{\varrho}_t = \frac{d}{dt} \frac{d}{dt} \varrho_t$ in the accelerated gradient flow (3) represents taking directional derivatives of $\frac{d}{dt} \varrho_t$ along $\frac{d}{dt} \varrho_t$. It holds for the damping term that $\nabla^2 f(\varrho_t) \dot{\varrho}_t = \frac{d}{dt} \nabla f(\varrho_t)$. Hence, by replacing the second-order term $\frac{d}{dt} \frac{d}{dt} \varrho_t$ and $\frac{d}{dt} \nabla f(\varrho_t)$ with the Levi-Civita connection and the Riemannian gradient as

4

$\nabla_{\partial_t \rho_t} \partial_t \rho_t$ and $\nabla_{\partial_t \rho_t} \operatorname{grad} L(\rho_t)$, we obtain the accelerated Riemannian gradient flow with Hessian-driven damping:

$$\nabla_{\partial_t \rho_t} \partial_t \rho_t + \alpha_t \partial_t \rho_t + \beta_t \nabla_{\partial_t \rho_t} \operatorname{grad} L(\rho_t) + \gamma_t \operatorname{grad} L(\rho_t) = 0. \tag{5}$$

Note that Levi-Civita connection in (5) is not straightforward to discrete with respect to time t. To derive the appropriate first order ODE formulation for numerical discretization, which only contains first order derivative to time and space, we apply the following transformation.

Proposition 1. *Define $\Phi_t = \Psi_t - \beta_t \frac{\delta L}{\delta \rho_t}$ and the Riemannian correction term*

$$R_t = \frac{1}{2} \frac{\delta}{\delta \rho} \left[\int \Phi_t \mathcal{G}(\rho)^{-1} \Phi_t \, dx \right]\bigg|_{\rho = \rho_t} + \frac{\beta_t}{2} \mathcal{G}(\rho_t) \left[\frac{\partial(\mathcal{G}(\rho_t)^{-1})}{\partial \rho_t} \cdot \mathcal{G}(\rho_t)^{-1} \Phi_t \right] \frac{\delta L}{\delta \rho_t}.$$

The second-order accelerated Riemannian gradient flow (5) is equivalent to

$$\begin{cases} \partial_t \rho_t - \mathcal{G}(\rho_t)^{-1} \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right) = 0, & (6) \\[2mm] \partial_t \Psi_t + \alpha_t \Psi_t + R_t + \left(\gamma_t - \dot{\beta}_t - \alpha_t \beta_t \right) \frac{\delta L}{\delta \rho_t} = 0, & (7) \end{cases}$$

with initial values ρ_0 and $\Psi_0 = \beta_0 \frac{\delta L}{\delta \rho_0}$.

The proof of this proposition can be found in Section 5.1.

By incorporating the specific metric $\mathcal{G}$ into equation (6) and (7), we can derive the accelerated flow corresponding to each concrete metric. We refer the reader to [20] and [29] for foundational information on the metrics.

Example 1 (L^2 ARG flow)**.** *Note that $L^2 = H^0$ is a special case of H^s space. Since in L^2 space, $\mathcal{G}(\rho_t) = Id$, the ARG flow for (ρ_t, Ψ_t) in (6) can be greatly simplified as:*

$$\begin{cases} \partial_t \rho_t = \Psi_t - \beta_t \frac{\delta L}{\delta \rho_t}, \\ \partial_t \Psi_t + \alpha_t \Psi_t - (\alpha_t \beta_t - \gamma_t + \dot{\beta}_t) \frac{\delta L}{\delta \rho_t} = 0. \end{cases} \tag{8}$$

Example 2 (H^s ($s \geq 0$) ARG flow)**.** *H^s ($s \geq 0$) ARG flow satisfies*

$$\begin{cases} \partial_t \rho_t - \left[\sum_{i=0}^{s} (-\Delta)^i \right]^{-1} \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right) = 0, \\ \partial_t \Psi_t + \alpha_t \Psi_t - (\alpha_t \beta_t - \gamma_t + \dot{\beta}_t) \frac{\delta L}{\delta \rho_t} = 0. \end{cases} \tag{9}$$

Example 3 (H^s ($s < 0$) ARG flow)**.** *H^s ($s < 0$) ARG flow satisfies*

$$\begin{cases} \partial_t \rho_t - \sum_{i=0}^{|s|} (-\Delta)^i \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right) = 0, \\ \partial_t \Psi_t + \alpha_t \Psi_t - (\alpha_t \beta_t - \gamma_t + \dot{\beta}_t) \frac{\delta L}{\delta \rho_t} = 0. \end{cases} \tag{10}$$

Example 4 (Fisher-Rao ARG flow)**.** *Fisher-Rao ARG flow satisfies*

$$\begin{cases} \partial_t \rho_t - \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} - \mathbb{E}_{\rho_t} \left[\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right] \right) \rho_t = 0, \\[2mm] \partial_t \Psi_t + \alpha_t \Psi_t + \frac{1}{2} \left(\Phi_t - \mathbb{E}_{\rho_t} [\Phi_t] \right) \left(\Psi_t - \mathbb{E}_{\rho_t} [\Psi_t] \right) - (\alpha_t \beta_t - \gamma_t + \dot{\beta}_t) \frac{\delta L}{\delta \rho_t} = 0. \end{cases} \tag{11}$$

Example 5 (Wasserstein ARG flow)**.** *Wasserstein ARG flow satisfies*

$$\begin{cases} \partial_t \rho_t + \nabla \cdot \left(\rho_t \nabla \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right) \right) = 0, \\ \partial_t \Psi_t + \alpha_t \Psi_t + \frac{1}{2} \|\nabla \Phi_t\|^2 + \frac{\beta_t}{2} w_t - (\alpha_t \beta_t - \gamma_t + \dot{\beta}_t) \frac{\delta L}{\delta \rho_t} = 0, \end{cases} \tag{12}$$

where w_t is a solution of $\nabla \cdot \left(h_t \nabla \frac{\delta L}{\delta \rho_t} - \rho_t \nabla w_t \right) = 0$ with $h_t = \nabla \cdot (\rho_t \nabla \Phi_t)$.

5

3 A Discretization Scheme on the Parameter Space

Along the trajectory of the ARG flow (5), L converges to its minimum. However, the discrete updates are performed in terms of the parameters $\theta \in \Theta$. Therefore, we need to map the iterations from the manifold back to the parameters. For some complicated metrics, we transform the original ODE of $\partial_t \rho_t(x)$ in (6) as follows

$$\partial_t \mathcal{A}(\rho_t)(x) = u_t(x), \tag{13}$$

where $\mathcal{A}$ is a metric-specific map with function output, $u_t(x)$ is obtained based on $\mathcal{A}$ and the original $\mathcal{G}(\rho_t)^{-1} \left(\Psi_t - \beta_t \frac{\delta L}{\delta \rho_t} \right)$ in (6). The purpose of this transformation is to adapt to various problem structures. For example, when dealing with a normalized probability density $\rho_t(x) = \psi_t(x) / \int \psi_t(x) dx$, the quantity $\partial_t \log \rho_t(x) = \partial_t \log \psi_t(x) - \mathbb{E}_{\rho_t}[\partial_t \log \psi_t]$ can be easily estimated, while $\partial_t \rho_t(x)$ is not directly accessible. Specific examples of (13) are provided in Table 1.

Metric	$\mathcal{G}(\rho_t)^{-1} \Phi_t$	$\mathcal{A}(\rho_t)$	$\mathcal{S}(\rho_t)$	$\mathcal{B}(\rho_t)$	u_t	q_t
Fisher-Rao	$\left(\Phi_t - \int \Phi_t \rho_t dx \right) \rho_t$	$\log \rho_t$	$\log \rho_t$	$\log \rho_t$	$-\left(\Phi_t - \int \Phi_t \rho_t dx \right)$	ρ_t
Wasserstein-2	$-\nabla \cdot (\rho_t \nabla \Phi_t)$	$\log \rho_t$	$\log \rho_t$	$\log \rho_t$	$-\langle \nabla \log \rho_t, \nabla \Phi_t \rangle - \Delta \Phi_t$	ρ_t
H^s ($s \in \mathbb{N}$)	$\left[\sum_{i=0}^s (-\Delta)^i \right]^{-1} \Phi_t$	ρ_t	$\left[\sum_{i=0}^s (-\Delta)^i \right] \rho_t$	$\mathbf{D}^s \rho_t$	$-\left[\sum_{i=0}^s (-\Delta)^i \right]^{-1} \Phi_t$	Leb.
H^s ($s \in \mathbb{Z}^-$)	$\left[\sum_{i=0}^{|s|} (-\Delta)^i \right] \Phi_t$	ρ_t	ρ_t	ρ_t	$-\left[\sum_{i=0}^s (-\Delta)^i \right] \Phi_t$	Leb.

Table 1: Examples of metric-specific transformed ARG flow (13). Leb. denotes the Lebesgue measure in Euclidean spaces.

3.1 A Unified Discretization Scheme

In this subsection, we discuss a unified discretization scheme for (13) and (7). To clarify the notation, we use t as a subscript for continuous-time variables, k as a subscript for discrete-time variables. The core procedure is as follows. We obtain the first-order system corresponding to $\partial_t \rho_t(x)$ in (13) and $\partial_t \Psi_t(x)$ in (7). Introducing parameters θ into the update of $\rho_t(x)$, it yields a parameterized ODE where $\rho_{\theta_t}(x)$ approximates $\rho_t(x)$. We denote the update direction for θ_k at the k-th iteration as d_k, which can be seen as an approximation of continuous parameter update $\partial_t \theta_t$ at time t_k. Denote the step size at the k-th iteration as h_k. The update rule for θ_k is given as

$$\theta_{k+1} = \theta_k + h_k d_k. \tag{14}$$

Sampling both sides of the parameterized ODE for $\rho_{\theta_t}(x)$ produces a linear system for the update direction d_k. Moreover, we derive the update rule for $\Psi_t(x)$ from a direct discretization, which is used to update the right-hand side of the linear system. We summarize the associated notations in Table 2.

We next discuss how to form a linear system to get d_k. By the chain rule, it yields

$$\partial_t \mathcal{A}(\rho_{\theta_t})(x) = (\partial_\theta \mathcal{A}(\rho_{\theta_t})(x))^\top \partial_t \theta_t, \tag{15}$$

where $\partial_\theta \mathcal{A}(\rho_{\theta_t})(x) = [\partial_{\theta[1]} \mathcal{A}(\rho_{\theta_t})(x), \cdots, \partial_{\theta[p]} \mathcal{A}(\rho_{\theta_t})(x)]^\top$ are p functions on the tangent space. Hence, we construct the following parameterized ODE:

$$(\partial_\theta \mathcal{A}(\rho_{\theta_t})(x))^\top \partial_t \theta_t = u_t(x), \quad \forall \, x \in \mathbb{R}^d. \tag{16}$$

6

	Continuous Time	Parametrized	Discrete Time
Parameters	not occurred	θ_t	θ_k
Update Direction	not occurred	$\partial_t \theta_t$	d_k
ρ-manifold	$\rho_t(x)$	$\rho_{\theta_t}(x)$	$\rho_{\theta_k}(x)$
Ψ-momentum	$\Psi_t(x)$	not used	$\Psi_k(x)$
Riemannian correction term	$R_t(x)$	not used	$R_k(x)$

Table 2: Different notations used for the discretized scheme.

Equation (16) shows that two functions are equal at the tangent space. We can only take a finite set of sample points $\{x_k^j\}_{j=1}^n$ to approximate these functions with matrices and vectors. Therefore, based on these samples, we build a linear system approximating (16) to find a suitable update direction d_k. It can also be seen as the discretization of $\partial_t \theta_t$ at the k-th iteration. The right-hand side of (16) becomes a real number for a specific sample. For the left-hand side, the function vector $\partial_\theta \mathcal{A}(\rho_{\theta_t})(x_k^j)$ turns into a vector in $\mathbb{R}^p$. Given the complexity of our networks, we can assume that the number of parameters p exceeds the sample size n. This leads to an underdetermined system as follows:

$$(\partial_\theta \mathcal{A}(\rho_{\theta_k})(x_k^j))^\top d_k = u_k(x_k^j), \quad \forall\, 1 \le j \le s. \tag{17}$$

For specific problems, the samples $\{x_k^j\}_{j=1}^n$ may be either fixed or changeable. For simplicity, we denote the following notations:

$$O_k = [\partial_\theta \mathcal{A}(\rho_{\theta_k})(x_k^1), ..., \partial_\theta \mathcal{A}(\rho_{\theta_k})(x_k^n)]^\top \in \mathbb{R}^{n \times p},$$

$$b_k = \left[u_k(x_k^1), ..., u_k(x_k^n)\right]^\top \in \mathbb{R}^{n \times 1}.$$

The system (17) can be rewritten as

$$O_k d_k = b_k. \tag{18}$$

The method of solving system (18) varies depending on the metric and the optimization problem. We summarize some general approaches in Section 3.2.

Now we focus on the update rule for $\Psi_k(x_k^j)$, which is used to update b_k. Since $\Psi_t(x)$ serves as an auxiliary variable in the ODEs (6) and (7), we approximate $\partial_t \Psi_t(x)$ at the k-th step using a forward difference over the time interval h_k at sample points as follows

$$\partial_t \Psi_t(x_k^j)\big|_{t=t_k} \approx \frac{\Psi_k(x_k^j) - \Psi_{k-1}(x_k^j)}{h_k}. \tag{19}$$

For terms in (7) which are not related to the time derivative, we simply use their values at t_k. This gives us the update for $\Psi_k(x_k^j)$ as follows:

$$\Psi_k(x_k^j) = \mu_k \Psi_{k-1}(x_k^j) - h_k \left(R_k(x_k^j) + \left(\gamma_k - \dot{\beta}_k - \alpha_k \beta_k\right) \frac{\delta L}{\delta \rho_{\theta_k}}(x_k^j) \right), \tag{20}$$

where $\mu_k = 1 - h_k \alpha_k$. The update scheme for $\Psi_k(x_k^j)$ may vary between different metrics, which will be thoroughly discussed when considering specific metrics.

In each iteration, $\Psi_k(x_k^j)$ is computed using (20), which in turn determines b_k. We then solve equation (18) for the update direction d_k, and update the parameters according to equation (14). The general method of solving (18) is discussed in Section 3.2. For some cases, b_k in (18) is hard to get. We discuss methods to deal with these cases in Section 3.3.

3.2 Direct Methods for Solving (18)

As an underdetermined system, directly solving the pseudo-inverse of O_k for (18) by singular value decomposition is time-consuming. Without loss of generality, suppose that O_k has full row rank. We propose three different algorithms to solve this system. We can multiply $O_k^\top$ on both sides of the system

(18). Since $O_k^\top O_k$ is a positive semidefinite matrix, it can be inverted by adding a damping term. The direction is given as

$$d_k = (O_k^\top O_k)^\dagger O_k^\top b_k. \tag{21}$$

From the singular value decomposition, the update (21) is also equivalent to

$$d_k = O_k^\top (O_k O_k^\top)^{-1} b_k. \tag{22}$$

However, as the norm of the solution in (21) increases with the number of samples, directly selection leads to significant errors, especially for small sample sizes. To address this issue, we can incorporate momentum by projecting the historical solution onto the solution set. The parameter update direction d_k is computed iteratively as:

$$d_k = O_k^\top (O_k O_k^\top)^{-1} b_k + \eta (I - O_k^\top (O_k O_k^\top)^{-1} O_k) d_{k-1}, \tag{23}$$

where $0 < \eta < 1$ is the decay rate, and $I - O_k^\top (O O_k^\top)^{-1} O_k$ is the projection to the null space of O_k. By projecting the momentum onto the subspace, we can effectively reduce the error in estimating the true solution.

3.3 Handling the Unavailability of b_k

In some cases, the target function L involves high-order derivatives. Computing $\frac{\delta L}{\delta \rho_{\theta_k}}(x_k^j)$ of $\Phi_k(x_k^j)$ in $u_k(x_k^j)$ requires these derivatives, which are often challenging to evaluate directly. Hence, the update (22) and (23) can not be used here since we can not get the value of b_k. To address the unavailability of b_k, we leverage an indirect approach. Although $u_k(x)$ cannot be computed directly, it can be approximated through a composition involving another quantity, $\mathcal{S}(\rho_{\theta_k}(x))$, enabling its subsequent use in computations. We define

$$S_k = \left[\partial_\theta \mathcal{S}(\rho_{\theta_k})(x_k^1), ..., \partial_\theta \mathcal{S}(\rho_{\theta_k})(x_k^n) \right]^\top \in \mathbb{R}^{n \times p}$$

and assume it has full row rank. Consequently, for any $d_k \in \mathbb{R}^{p \times 1}$, (18) holds if and only if the following equation holds

$$\frac{1}{n} S_k^\top O_k d_k = \frac{1}{n} S_k^\top b_k. \tag{24}$$

This equivalence enables us to reformulate the problem and solve (18). However, the non-symmetric form of the precondition matrix on the left-hand side may lead to numerical instability. To address this, we assume the existence of an equivalent transformation that reformulates the original integral into a semi-definite form. Assume our samples $\{x_k^j\}$ are generated from the distribution $q_k(x)$ (see Table 1 for more examples). Therefore, the summation over samples $S_k^\top O_k$ can be viewed as an expectation as

$$\frac{1}{n} S_k^\top O_k \approx \mathbb{E}_{x \sim q_k(x)} \left[\partial_\theta \mathcal{S}(\rho_{\theta_k})(x) \partial_\theta \mathcal{A}(\rho_{\theta_k})(x)^\top \right]. \tag{25}$$

Assume that through integral by parts, the right hand side of (25) can be reformulated into a semi-definite form using a function-valued map $\mathcal{B}(\rho_{\theta_k})$ as:

$$\mathbb{E}_{x \sim q_k(x)} \left[\partial_\theta \mathcal{S}(\rho_{\theta_k})(x) \partial_\theta \mathcal{A}(\rho_{\theta_k})(x)^\top \right] = \mathbb{E}_{x \sim q_k(x)} \left[\partial_\theta \mathcal{B}(\rho_{\theta_k})(x) \partial_\theta \mathcal{B}(\rho_{\theta_k})(x)^\top \right].$$

Denote $B_k = \left[\partial_\theta \mathcal{B}(\rho_{\theta_k})(x_k^1), ..., \partial_\theta \mathcal{B}(\rho_{\theta_k})(x_k^n) \right]^\top$ as the discretization of $\partial_\theta \mathcal{B}(\rho_{\theta_k})$ on samples. It yields the following approximation

$$\frac{1}{n} B_k^\top B_k \approx \mathbb{E}_{x \sim q_k(x)} \left[\partial_\theta \mathcal{B}(\rho_{\theta_k})(x) \partial_\theta \mathcal{B}(\rho_{\theta_k})(x)^\top \right].$$

Consequently, we can use $B_k^\top B_k$ to approximate $S_k^\top O_k$ in (24).

A similar approach is also used to estimate the right hand side of (24) as v_k. Further details of practical methods for certain metric can be found in Table 1 and the following sections. Hence, the remaining task is to compute the update direction

$$d_k = \left(\frac{1}{n} B_k^\top B_k \right)^\dagger v_k, \tag{26}$$

where various methods, such as KFAC, can be employed to efficiently handle the matrix-inverse-vector product.

Remark 1. *In most cases, we take $\mathcal{S} = \mathcal{A}$ directly with $B_k = O_k$. The update (26) can be seen as a least squares solution of the original problem (18).*

3.4 Approximation Algorithms

Directly calculating $(O_k^\top O_k)^\dagger \in \mathbb{R}^{p \times p}$ in (21) and $\left(B_k^\top B_k\right)^\dagger$ in (26) is complicated due to the huge matrix size. In this subsection, an approximation technique using the Kronecker decomposition is introduced to reduce the calculation overhead. We focus on the k-th iteration, and omit the subscript for simplicity.

Consider the state variable $\rho_\theta \in \mathcal{M}$ is parameterized by a feedforward network with K layers with a collection of the weight matrices for each layer $\theta = \left(\text{vec}(W_1)^\top, \cdots, \text{vec}(W_K)^\top\right)^\top \in \mathbb{R}^p$. We omit the bias term since it can be incorporated into the weight matrix. Let $\theta^{(l)} = \text{vec}(W_l) \in \mathbb{R}^{n_l n_{l-1}}$, $b_i = \partial_{\theta^{(l)}} \mathcal{A}(\rho_{\theta^{(l)}})(x^i)$ and $G_l = \frac{1}{n} \sum_{i=1}^n b_i b_i^\top$. The preconditioner in (26), (36), and (41) can be approximated as a block diagonal matrix $\text{Diag}\{G_1, G_2, \cdots, G_K\}$. For certain input x^i, we denote the input of the l-th layer by a_{l-1}^i, and $s_l^i = W_l a_{l-1}^i$. It follows that

$$\partial_{\theta^{(l)}} \mathcal{A}(\rho_{\theta^{(l)}})(x^i) = \text{vec}\left(\frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i} a_{l-1}^i{}^\top\right) = a_{l-1}^i \otimes \frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i}. \tag{27}$$

Assuming that a_{l-1} and $\frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})}{\partial s_l}$ are independent with respect to the sample distribution, we can approximate G_l by

$$\begin{aligned}
G_l &= \frac{1}{n} \sum_{i=1}^n (a_{l-1}^i (a_{l-1}^i)^\top) \otimes \left(\frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i}^\top \frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i}\right) \\
&\approx \underbrace{\left(\frac{1}{n} \sum_{i=1}^n a_{l-1}^i (a_{l-1}^i)^\top\right)}_{A_{l-1}} \otimes \underbrace{\left(\frac{1}{n} \sum_{i=1}^n \frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i}^\top \frac{\partial \mathcal{A}(\rho_{\theta^{(l)}})(x^i)}{\partial s_l^i}\right)}_{S_l}.
\end{aligned} \tag{28}$$

It yields $G_l^\dagger \approx (A_{l-1} \otimes S_l)^\dagger = A_{l-1}^\dagger \otimes S_l^\dagger$. This approximation efficiently reduces the computation overhead for the matrix inverse from $\mathcal{O}((n_l n_{l-1})^3)$ to $\mathcal{O}(n_l^3 + n_{l-1}^3)$.

4 Accelerated Natural Gradient Methods with Specific Metrics

4.1 L^2 Metric

The L^2 metric mainly serves for PDE-based optimization problems, where the functional mapping $\mathcal{A}(\rho)$ and metric tensor $\mathcal{G}(\rho)$ are both the identity. In this case, we focus on working with fixed sampled points, thus omitting the subscripts for the samples. We have

$$b_k = \left[\left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}}\right)(x^1), ..., \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}}\right)(x^n)\right]^\top.$$

From (8) and (20), the update rule for each $\Psi_k(x^j)$ is given as

$$\Psi_k(x^j) = \mu_k \Psi_{k-1}(x^j) - h_k \left(\gamma_k - \dot{\beta}_k - \alpha_k \beta_k\right) \frac{\delta L}{\delta \rho_{\theta_k}}(x^j). \tag{29}$$

In this case, b_k is not available since calculating $\frac{\delta L}{\delta \rho_{\theta_k}}(x^j)$ involves high order derivatives, which are difficult to obtain. Hence, we apply (26) with $B_k = O_k$ to get d_k for L^2 metric. For $v_k = O_k^\top b_k$ in (26), the following

equation holds

$$v_k = O_k^\top b_k = \underbrace{\sum_{j=1}^{n} \partial_\theta \rho_{\theta_k}(x^j) \cdot \Psi_k(x^j)}_{(A)} - \beta_k \underbrace{\sum_{j=1}^{n} \partial_\theta \rho_{\theta_k}(x^j) \cdot \frac{\delta L}{\delta \rho_{\theta_k}}(x^j)}_{(B)}. \tag{30}$$

We cannot obtain the (A) and (B) parts in equation (30) through direct multiplication. However, notice that by the chain rule, it holds

$$\partial_\theta L(\rho_{\theta_k}) = \int \partial_\theta \rho_{\theta_k}(x) \frac{\delta L}{\delta \rho_{\theta_k}}(x) dx \approx \frac{1}{n} \sum_{j=1}^{n} \partial_\theta \rho_{\theta_k}(x^j) \cdot \frac{\delta L}{\delta \rho_{\theta_k}}(x^j). \tag{31}$$

The approximation arises from the correlation between the continuous integral in the inner product and the discrete summation over the samples. From (31), we directly use $\partial_\theta L(\rho_{\theta_k})$ to calculate (B) in (30). The calculation of $\partial_\theta L(\rho_{\theta_k})$ can be done by automatic differentiation and is easy to get. For (A) in (30), we introduce an iterative approximation technique to establish an iterative update rule for v_k from v_{k-1} and the gradient towards θ of the target function L. From (29) and (30), it yields

$$v_k[i] = \sum_{j=1}^{n} \partial_{\theta[i]} \rho_{\theta_k}(x^j) \left(\mu_k \Psi_{k-1}(x^j) - \left(\mu_k \beta_k + h_k(\gamma_k - \dot{\beta}_k) \right) \frac{\delta L}{\delta \rho_{\theta_k}}(x^j) \right). \tag{32}$$

For the last line in (32), we can only approximate $\sum_{j=1}^{n} \partial_{\theta[i]} \rho_{\theta_{k-1}}(x^j) \Psi_{k-1}(x^j)$ from the last v_{k-1}. Assuming that the discrete step size is small enough on the manifold, ρ_{θ_k} and $\rho_{\theta_{k-1}}$ are not far away on the tangent space. Hence, we can approximate $\partial_{\theta_i} \rho_{\theta_{k-1}}$ with $\partial_{\theta_{[i]}} \rho_{\theta_k}$. We employ the following approximation

$$\sum_{j=1}^{n} \partial_{\theta[i]} \rho_{\theta_k}(x^j) \Psi_{k-1}(x^j) \approx \sum_{j=1}^{n} \partial_{\theta[i]} \rho_{\theta_{k-1}}(x^j) \Psi_{k-1}(x^j).$$

Further we can get the following equation:

$$\sum_{j=1}^{n} \partial_{\theta[i]} \rho_{\theta_{k-1}}(x^j) \Psi_{k-1}(x^j) = v_{k-1}[i] + \sum_{j=1}^{n} \beta_{k-1} \partial_{\theta[i]} \rho_{\theta_{k-1}}(x^j) \frac{\delta L}{\delta \rho_{\theta_{k-1}}}(x^j).$$

From this approximation, taking (31) into consideration, the following update rule holds for the vector w_k to approximate the original v_k:

$$w_k = \mu_k \left(w_{k-1} + n\beta_{k-1} \nabla_\theta L(\rho_{\theta_{k-1}}) \right) - n \left(\mu_k \beta_k + h_k(\gamma_k - \dot{\beta}_k) \right) \nabla_\theta L(\rho_{\theta_k}). \tag{33}$$

Finally, we can derive the accelerated L^2 gradient descent method as Algorithm 1.

Algorithm 1 Accelerated L^2 Natural Gradient Descent

Input: Initial parameters θ_0, step sizes h_k, decay rates $\{\alpha_k\}, \{\beta_k\}, \{\gamma_k\}$
Output: Updated parameters θ_T

1: **for** $k = 1, \ldots, T-1$ **do**
2:    Update w_k according to (33).
3:    Calculate $(O_k^\top O_k)^\dagger$ through samples.
4:    Calculate update direction $d_k = (O_k^\top O_k)^\dagger w_k$.
5:    Update model parameters $\theta_{k+1} = \theta_k + h_k d_k$.
6: **end for**

4.2 H^s Metric

The H^s metric is a generalization of the basic L^2 metric. According to Table 1, the quantities O_k and b_k in (18) are given as:

$$O_k = [\partial_\theta \rho_{\theta_k}(x^1), ..., \partial_\theta \rho_{\theta_k}(x^n)]^\top,$$

$$b_k = \left[((\mathbf{D}^s)^* \mathbf{D}^s)^{-1} \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^1), ..., ((\mathbf{D}^s)^* \mathbf{D}^s)^{-1} \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^n) \right]^\top,$$

where $\mathbf{D}^s \sigma$ is a vector consisting of all derivatives of σ up to order s. In this case, the term b_k is computationally intractable due to the existence of high-order (positive or negative) derivatives. To overcome this difficulty, we follow the approach outlined in Section 3.3 by selecting $\mathcal{S}(\rho)$ as given in Table 1. The treatment of the equivalent system (24) depends on whether s is positive or negative.

Case 1: $s > 0$ We first introduce the following definitions:

$$S_k = [\partial_\theta ((\mathbf{D}^s)^* \mathbf{D}^s) \rho_{\theta_k}(x^1), ..., \partial_\theta ((\mathbf{D}^s)^* \mathbf{D}^s) \rho_{\theta_k}(x^n)]^\top \in \mathbb{R}^{n \times p},$$

$$B_k^{\mathcal{D}} = [\partial_\theta \mathcal{D} \rho_{\theta_k}(x^1), ..., \partial_\theta \mathcal{D} \rho_{\theta_k}(x^n)]^\top \in \mathbb{R}^{n \times p}, \quad \mathcal{D} \in \mathbf{D}^s,$$

$$z_k = \left[\left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^1), ..., \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^n) \right]^\top \in \mathbb{R}^{p \times 1}.$$

Integration by parts then shows that:

$$
\begin{aligned}
\frac{1}{n} S_k^\top O_k &\approx \int \partial_\theta ((\mathbf{D}^s)^* \mathbf{D}^s) \rho_{\theta_k}(x) \partial_\theta \rho_{\theta_k}(x)^\top dx \\
&= \int \partial_\theta \mathbf{D}^s \rho_{\theta_k}(x) \partial_\theta \mathbf{D}^s \rho_{\theta_k}(x)^\top dx \approx \frac{1}{n} \sum_{\mathcal{D} \in \mathbf{D}^s} (B_k^{\mathcal{D}})^\top B_k^{\mathcal{D}}.
\end{aligned}
\tag{34}
$$

It also holds that

$$
\begin{aligned}
\frac{1}{n} S_k^\top b_k &\approx \int \partial_\theta ((\mathbf{D}^s)^* \mathbf{D}^s) \rho_{\theta_k}(x) ((\mathbf{D}^s)^* \mathbf{D}^s)^{-1} \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x) dx \\
&= \int \partial_\theta \rho_{\theta_k}(x) \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x) dx \approx \frac{1}{n} O_k^\top z_k.
\end{aligned}
\tag{35}
$$

The approximation is given through the connection between integral of functions and the summation over samples. Thus, the update direction d_k is given by:

$$d_k = \left(\frac{1}{n} \sum_{\mathcal{D} \in \mathbf{D}^s} (B_k^{\mathcal{D}})^\top B_k^{\mathcal{D}} \right)^\dagger \left(\frac{1}{n} O_k^\top z_k \right). \tag{36}$$

It is worth noting that the term $O_k^\top z_k$ coincides with v_k from the L^2 case. Hence, its update can be approximated by that of w_k.

Case 2: $s < 0$ Similar to (34) and (35), we employ integration by parts to redistribute the high-order derivatives to other terms, thereby alleviating the computational complexity of handling b_k. To be specific, we take $S_k = O_k$ and

$$B_k^{\mathcal{D}} = [\partial_\theta \mathcal{D} \rho_{\theta_k}(x^1), ..., \partial_\theta \mathcal{D} \rho_{\theta_k}(x^n)]^\top \in \mathbb{R}^{n \times p}, \quad \mathcal{D} \in \mathbf{D}^{|s|},$$

$$z_k^{\mathcal{D}} = \left[\mathcal{D} \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^1), ..., \mathcal{D} \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right) (x^n) \right]^\top \in \mathbb{R}^{p \times 1}, \quad \mathcal{D} \in \mathbf{D}^{|s|}.$$

It yields the approximation:

$$
\begin{aligned}
\frac{1}{n} S_k^\top b_k &\approx \int \partial_\theta \rho_{\theta_k}(x)((\mathbf{D}^s)^* \mathbf{D}^s) \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right)(x) dx \\
&= \int \partial_\theta \mathbf{D}^s \rho_{\theta_k}(x) \mathbf{D}^s \left(\Psi_k - \beta_k \frac{\delta L}{\delta \rho_{\theta_k}} \right)(x) dx \approx \frac{1}{n} \sum_{\mathcal{D} \in \mathbf{D}^s} (B_k^{\mathcal{D}})^\top z_k^{\mathcal{D}}.
\end{aligned}
\tag{37}
$$

Furthermore, analogous to (20), each $z_k^{\mathcal{D}}$ can be computed via the iterative update:

$$
\mathcal{D}\Psi_k(x^j) = \mu_k \mathcal{D}\Psi_{k-1}(x^j) + h_{k-1} \left(\gamma_k - \dot{\beta}_k - \alpha_k \beta_k \right) \mathcal{D} \frac{\delta L}{\delta \rho_{\theta_k}}(x^j), \quad k \geq 1.
\tag{38}
$$

4.3 Fisher-Rao metric

Beyond PDE-based models, we now consider ρ_θ as a parameterized probability density function. The ρ-trajectory in the Fisher-Rao ARG flow (11) can be reformulated as $\partial_t \log \rho_t - (\Phi_t - \mathbb{E}_{\rho_t}[\Phi_t]) = 0$, where $\Phi_t = \Psi_t - \beta_t \frac{\delta L}{\delta \rho_t}$. By defining

$$
O_k = [\partial_\theta \log(\rho_{\theta_k})(x_k^1), ..., \partial_\theta \log(\rho_{\theta_k})(x_k^n)]^\top \in \mathbb{R}^{n \times p},
$$

$$
b_k = \left[\overline{\Phi_k}(x_k^1), ..., \overline{\Phi_k}(x_k^n) \right]^\top \in \mathbb{R}^{n \times 1},
$$

we provide the explicit form of the linear system previously introduced in (18).

Next we discretize Ψ-trajectory for updating the cotangent variable. To estimate $\mathbb{E}_{\rho_k}[\Phi_k]$ at the k-th iteration, it is necessary to update samples $\{x_k^i\}_{i=1}^n \sim \rho_{\theta_k}$ and evaluate Φ_k at these points. However, a fundamental challenge arises: storing a function state variable Φ_k for each $x \in \mathbb{R}^d$ is intractable. To address this, we use the solution of the previous linear system, d_{k-1}, to estimate centered values of Φ_k at samples $\{x_k^i\}_{i=1}^n$ in the k-iteration as follows:

$$
\overline{\Phi}_{k-1}(x_k^i) \triangleq \left\langle \partial_\theta \log \rho_{\theta_{k-1}}(x_k^i), d_{k-1} \right\rangle.
$$

Specifically, we discretize the Ψ-trajectory as:

$$
\frac{\overline{\Psi}_k(x_k^i) - \overline{\Psi}_{k-1}(x_k^i)}{h_k} = -\alpha_k \overline{\Phi}_{k-1}(x_k^i) - \frac{1}{2} \overline{\Phi}_{k-1}(x_k^i) \overline{\Psi}_{k-1}(x_k^i) - (\gamma_k - \dot{\beta}_k) \overline{\frac{\delta L}{\delta \rho_{\theta_k}}}(x_k^i),
$$

where $\overline{\Psi}_{k-1}(x_k^i) = \overline{\Phi}_{k-1}(x_k^i) + \beta_{k-1} \overline{\frac{\delta L}{\delta \rho_{\theta_{k-1}}}}(x_k^i)$. Here we use the centralized cotangent variable over samples since centralization does not influence the iteration update. Consequently, the update for the cotangent variable is derived as:

$$
\begin{aligned}
\overline{\Phi}_k(x_k^i) =& (1 - h_k \alpha_k) \overline{\Phi}_{k-1}(x_k^i) - \frac{h_k}{2} \overline{\Phi}_{k-1}(x_k^i) \overline{\Psi}_{k-1}(x_k^i) \\
&+ (\beta_k - h_k \dot{\beta}_k - h_k \gamma_k) \overline{\frac{\delta L}{\delta \rho_{\theta_k}}}(x_k^i) - \beta_{k-1} \overline{\frac{\delta L}{\delta \rho_{\theta_{k-1}}}}(x_k^i).
\end{aligned}
\tag{40}
$$

Based on the above discussion, we are ready to present the ANGD method in Algorithm 2 for the Fisher-Rao metric.

4.4 Wasserstein-2 metric

The evolution of probability density can equivalently be interpreted as the movement of particles. From the perspective of the continuity equation, (12) explicitly defines the velocity of each particle $x_t \sim \rho_t$ as $\dot{x}_t = \nabla \Phi_t(x_t)$, where $\Phi_t = \Psi_t - \beta_t \frac{\delta L}{\delta \rho_t}$. Substituting this into the evolution of Ψ_t in (12) and taking the spatial gradient, we obtain the particle-wise velocity evolution:

$$
\frac{d}{dt} \left[\nabla \Psi_t(x_t) \right] = -\alpha_t \nabla \Psi_t + \beta_t \underbrace{\left(\nabla^2 \frac{\delta L}{\delta \rho_t} \nabla \Phi_t - \frac{1}{2} \nabla w_t \right)}_{W_{1,t}} + \eta_t \nabla \frac{\delta L}{\delta \rho_t}.
$$

Algorithm 2 Accelerated Fisher-Rao Natural Gradient Descent

Input: Initial model parameters θ_0, step sizes $\{h_k\}$, decay rates $\{\alpha_k\}$, $\{\beta_k\}$, $\{\gamma_k\}$.
Output: Updated model parameters θ_K.

1: Initialize $\Phi_0 = \mathbf{0} \in \mathbb{R}^n$.
2: Sample $\{x_0^i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \rho_{\theta_0}$.
3: **for** $k = 0, 1, \ldots, K-1$ **do**
4:     Estimate $\overline{\Phi}_{k-1}(x_k^i) = \left\langle \partial_\theta \log \rho_{\theta_{k-1}}(x_k^i), d_{k-1} \right\rangle$ for $1 \le i \le n$.
5:     Update the cotangent variable $b_k[i] \overset{\triangle}{=} \overline{\Phi}_k(x_k^i)$ according to (40) for $1 \le i \le n$.
6:     Compute the update direction d_k by solving $O_k d_k = b_k$.
7:     Update model parameters $\theta_{t+1} = \theta_k + h_k d_k$.
8:     Update samples x_{k+1}^i based on x_k^i via sampling methods for $1 \le i \le n$.
9: **end for**

Note that the evolution of $\log \rho_t$ given by (12) is $\partial_t \log \rho_t + \langle \nabla \log \rho_t, \nabla \Phi_t \rangle + \Delta \Phi_t = 0$, which also involves second-order derivatives of Φ_t (or Ψ_t). To capture these dynamics, we further examine the evolution of the spatial Hessian at time-varying samples:

$$\frac{d}{dt}\left[\nabla^2 \Psi_t(x_t)\right] = -\alpha_t \nabla^2 \Psi_t + \beta_t \underbrace{\left(\nabla^3 \frac{\delta L}{\delta \rho_t} \nabla \Phi_t - \frac{1}{2}\nabla^2 w_t\right)}_{W_{2,t}} - [\nabla^2 \Phi_t]^2 + \eta_t \nabla^2 \frac{\delta L}{\delta \rho_t}.$$

Thus the full particle-density evolution system is summarized as follows:

$$\begin{cases} \partial_t x_t - \nabla \Phi_t(x_t) = 0, \\ \partial_t \log \rho_t(x) + \langle \nabla \Phi_t(x), \nabla \log \rho_t(x) \rangle + \Delta \Phi_t(x) = 0, & \text{(fixed } x) \\ \dfrac{d}{dt}\left[\nabla \Psi_t(x_t)\right] + \alpha_t \nabla \Psi_t(x_t) - \beta_t W_{1,t} - \eta_t \nabla \dfrac{\delta L}{\delta \rho_t}(x_t) = 0, \\ \dfrac{d}{dt}\left[\nabla^2 \Psi_t(x_t)\right] + \alpha_t \nabla^2 \Psi_t(x_t) - \beta_t W_{2,t} + [\nabla^2 \Phi_t(x_t)]^2 - \eta_t \nabla^2 \dfrac{\delta L}{\delta \rho_t}(x_t) = 0, \end{cases}$$

where $\Phi_t = \Psi_t + \beta_t \frac{\delta L}{\delta \rho_t}$. Unlike the Fisher-Rao metric, which requires tracking updates of the cotangent variable at all spatial points in $\mathbb{R}^d$, the current framework for Wasserstein-2 metric only needs updates at sample points, making it suitable for practical applications. However, a computational challenge arises in evaluating ∇w_t and $\nabla^2 w_t$. The condition $\nabla \cdot \left(h_t \nabla \frac{\delta L}{\delta \rho_t} - \rho_t \nabla w_t\right) = 0$ does not implies $\rho_t \nabla w_t = h_t \nabla \frac{\delta L}{\delta \rho_t}$, as $\frac{h_t}{\rho_t} \nabla \frac{\delta L}{\delta \rho_t}$ is generally not curl-free and therefore can not guaranteed to be a gradient. To simplify, we approximate ∇w_t and $\nabla^2 w_t$ as zero, or set $\beta_t \equiv 0$ for computational efficiency.

Now we focus on the case where $\beta_t \equiv 0$, which implies $\Phi_t = \Psi_t$. The linear system (18) is specified by taking:

$$O_k = [\partial_\theta \log(\rho_{\theta_k})(x_k^1), ..., \partial_\theta \log(\rho_{\theta_k})(x_k^n)]^\top \in \mathbb{R}^{n \times p},$$

$$b_k = \left[\frac{\nabla \cdot (\rho_{\theta_k} \nabla \Phi_k)(x_k^1)}{\rho_{\theta_k}(x_k^1)}, ..., \frac{\nabla \cdot (\rho_{\theta_k} \nabla \Phi_k)(x_k^n)}{\rho_{\theta_k}(x_k^n)}\right]^\top \in \mathbb{R}^{n \times 1}.$$

Despite this simplification, additional challenges persist. Estimating b_k is computationally intractable due to the difficulty of directly computing or storing the spatial Hessian in most practical scenarios. This aligns with the case discussed in Section 3.3. To address this, we apply integration by parts, yielding the following identity:

$$\mathbb{E}_{x \sim \rho_{\theta_k}}\left[\frac{\nabla \cdot (\rho_{\theta_k} \nabla \Phi_k)(x)}{\rho_{\theta_k}(x)} \partial_\theta \log \rho_{\theta_k}(x)\right] = -\mathbb{E}_{x \sim \rho_{\theta_k}}\left[\partial_\theta \langle \nabla \log \rho_{\theta_k}(x), \nabla \Phi_k(x) \rangle\right],$$

where the right-hand side can be efficiently estimated by sampling and automatic differentiation. Thus we can estimate d_k by solving

$$\left(\frac{1}{n} O_k^\top O_k\right) d_k = \frac{1}{n} \sum_{i=1}^n \partial_\theta \langle \nabla \log \rho_{\theta_k}(x_k^i), \nabla \Phi_k(x_k^i) \rangle.$$

This approach requires only the computation of first-order derivatives, eliminating the need for second-order derivatives.

Synthesizing all the above insights, we can now formulate the ANGD method for Wasserstein-2 metric in Algorithm 3. To address discretization errors, we usually incorporate additional sampling steps (e.g., MCMC) for $x_k^i + h_k V_{k+1}^i$, ensuring that $x_{k+1}^i (1 \le i \le n)$ tend to follow $\rho_{\theta_{k+1}}$.

Algorithm 3 Accelerated Wasserstein-2 Natural Gradient Descent

Input: Initial parameters θ_0, step sizes $\{h_k\}$, decay rates $\{\alpha_k\}$, $\{\beta_k\}$, $\{\gamma_k\}$.
Output: Updated model parameters θ_K.

1: Initialize $V_0^i = \mathbf{0} \in \mathbb{R}^d$ for $1 \le i \le n$.
2: Sample $\{x_0^i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} \rho_{\theta_0}$.
3: **for** $k = 0, 1, \ldots, K-1$ **do**
4:    Update cotangent gradients $V_{k+1}^i = (1 - \alpha_k h_k) V_k^i - h_k \nabla \frac{\delta L}{\delta \rho_{\theta_k}}(x_k^i)$ for $1 \le i \le n$.
5:    Solve the parameter update direction d_k from

$$\left(\frac{1}{n} O_k^\top O_k \right) d_k = \frac{1}{n} \sum_{i=1}^n \partial_\theta \left\langle \nabla \log \rho_{\theta_k}(x_k^i), V_{k+1}^i \right\rangle.$$

6:    Update model parameters $\theta_{k+1} = \theta_k + h_k d_k$.
7:    Update samples x_{k+1}^i based on $x_k^i + h_k V_{k+1}^i$ for $1 \le i \le n$.
8: **end for**

---

5   Theoretical Analysis

In this section, we present a theoretical analysis of the ARG flow (5). First, we rigorously establish the equivalence between (5) and the system composed of coupled ODEs (6)–(7) through the proof of Proposition 1. Subsequently, we derive convergence guarantees for the ARG flow (5) under the geodesic convexity assumption.

5.1   Proof of Proposition 1

Proof. In the following proof, for notational convenience, we assume that any smooth tangent field defined on curve ρ_t possesses a local extension, which does not impact the final conclusion. We begin by computing $\nabla_{\partial_t \rho_t} \partial_t \rho_t$ using the Koszul formula. For any tangent field h_t along ρ_t, it holds:

$$g_{\rho_t} \left(\nabla_{\partial_t \rho_t} \partial_t \rho_t, h_t \right) = \underbrace{\frac{d}{dt} g_{\rho_t} \left(\partial_t \rho_t, h_t \right)}_{(A)} - \frac{1}{2} \underbrace{h_t \circ g_{\rho_t} \left(\partial_t \rho_t, \partial_t \rho_t \right)}_{(B)} \tag{41}$$
$$+ g_{\rho_t} \left(\partial_t \rho_t, [h_t, \partial_t \rho_t] \right),$$

where $[\cdot, \cdot]$ denotes Lie brackets. Define $\Phi_t = \mathcal{G}(\rho_t) \partial_t \rho_t \in T_{\rho_t}^* \mathcal{M}$, the components on the right side of (41) can be evaluated using calculus rules:

$$(A) = \frac{d}{dt} \int \Phi_t h_t dx = \int \partial_t \Phi_t h_t dx + \int \Phi_t \partial_t h_t dx. \tag{42}$$

Besides for component (B), it yields

$$(B) = \int \frac{\delta \left(g_{\rho_t} \left(\partial_t \rho_t, \partial_t \rho_t \right) \right)}{\delta \rho_t} h_t dx + 2 \int \mathcal{G}(\rho_t) \partial_t \rho_t(x) \int \frac{\delta}{\delta \rho_t} \partial_t \rho_t(x, y) h_t(y) dy dx. \tag{43}$$

Then we compute $[h_t, \partial_t \rho_t]$. For any smooth functional $E(\rho)$, it holds

$$[h_t, \partial_t \rho_t]E(\rho_t) = h_t \circ \frac{d}{dt}E(\rho_t) - \frac{d}{dt}\int \frac{\delta}{\delta \rho_t}E(\rho_t)h_t dx$$

$$= \int \frac{\delta E}{\delta \rho_t}(x)\left(\int \frac{\delta}{\delta \rho_t}\partial_t \rho_t(x,y)h_t(y)dy - \partial_t h_t(x)\right)dx,$$

where the last equality holds due to $\frac{\delta}{\delta \rho_t}\frac{d}{dt}E(\rho_t) = \frac{d}{dt}\frac{\delta}{\delta \rho_t}E(\rho_t)$. It implies:

$$[h_t, \partial_t \rho_t](x) = \int \frac{\delta}{\delta \rho_t}\partial_t \rho_t(x,y)h_t(y)dy - \partial_t h_t(x). \tag{44}$$

Substituting (42), (43), and (44) into (41), it gives:

$$g_{\rho_t}\left(\nabla_{\partial_t \rho_t}\partial_t \rho_t, h_t\right) = \int \partial_t \Phi_t h_t dx - \frac{1}{2}\int \frac{\delta\left(g_{\rho_t}\left(\partial_t \rho_t, \partial_t \rho_t\right)\right)}{\delta \rho_t}h_t dx. \tag{45}$$

Consequently, we have

$$\mathcal{G}(\rho_t)\nabla_{\partial_t \rho_t}\partial_t \rho_t = \partial_t \Phi_t - \frac{1}{2}\frac{\delta}{\delta \rho_t}\left[g_{\rho_t}\left(\partial_t \rho_t, \partial_t \rho_t\right)\right]. \tag{46}$$

Next we analyze the expression for $\nabla_{\partial_t \rho_t}\operatorname{grad} L$ using the Koszul formula again. For any smooth tangent field h_t along the curve ρ_t, it holds that:

$$2g_{\rho_t}(\nabla_{\partial_t \rho_t}\operatorname{grad} L, h_t)$$
$$= \underbrace{\partial_t \rho_t \circ g_{\rho_t}(\operatorname{grad} L, h_t)}_{(C)} + \underbrace{\operatorname{grad} L \circ g_{\rho_t}(\partial_t \rho_t, h_t)}_{(D)} - \underbrace{h_t \circ g_{\rho_t}(\operatorname{grad} L, \partial_t \rho_t)}_{(E)}$$
$$+ \underbrace{g_{\rho_t}([\partial_t \rho_t, \operatorname{grad} L], h_t)}_{(F)} + \underbrace{g_{\rho_t}([h_t, \partial_t \rho_t], \operatorname{grad} L)}_{(G)} + \underbrace{g_{\rho_t}([h_t, \operatorname{grad} L], \partial_t \rho_t)}_{(H)}. \tag{47}$$

Calculating the terms using the chain rule gives:

$$(C) - (E) + (G)$$
$$= \partial_t \rho_t \circ (h_t \circ \operatorname{grad} L) - h_t \circ (\partial_t \rho_t \circ \operatorname{grad} L) + [h_t, \partial_t \rho_t] \circ \operatorname{grad} L = 0, \tag{48}$$

and it holds for (D) that

$$(D) = \operatorname{grad} L \circ \int \partial_t \rho_t \mathcal{G}(\rho_t)h_t$$
$$= \iint \frac{\delta h_t}{\delta \rho_t}(x,y)\operatorname{grad} L(y)dy \cdot \mathcal{G}(\rho_t)\partial_t \rho_t(x)dx$$
$$+ \iint \frac{\delta h_t}{\delta \rho_t}(x,y)\operatorname{grad} L(y)dy \cdot \mathcal{G}(\rho_t)\partial_t \rho_t(x)dx \tag{49}$$
$$+ \int \partial_t \rho_t\left(\frac{\partial \mathcal{G}}{\partial \rho_t}\operatorname{grad} L\right)h_t dx \int \mathcal{G}(\rho_t)h_t(x)\int \frac{\delta}{\delta \rho_t}\partial_t \rho_t(x,y)\operatorname{grad} L(y)dydx.$$

For (F) and (H), we use a test functional $E(\rho)$:

$$[\partial_t \rho_t, \operatorname{grad} L] \circ E = \frac{d}{dt}\int \frac{\delta L}{\delta \rho_t}\operatorname{grad} L - \operatorname{grad} L \circ \partial_t E(\rho_t)$$
$$= \int \frac{\delta L}{\delta \rho_t}\partial_t \operatorname{grad} L(\rho_t) - \int \frac{\delta L}{\delta \rho_t}(x)\int \frac{\delta}{\delta \rho_t}\partial_t \rho_t(x,y)\operatorname{grad} L(y)dydx,$$

where the last equality hold due to the self-adjoint property of $\frac{\delta^2 L}{\delta \rho_t^2}$. This leads to:

$$[\partial_t \rho_t, \operatorname{grad} L](x) = \partial_t \operatorname{grad} L(x) - \int \frac{\delta}{\delta \rho_t}\partial_t \rho_t(x,y)\operatorname{grad} L(y)dy. \tag{50}$$

15

Similarly, we obtain:

$$[h_t, \operatorname{grad} L](x) = \int \frac{\delta}{\delta \rho_t} \operatorname{grad} L(x,y) h_t(y) dy - \int \frac{\delta}{\delta \rho_t} h_t(x,y) \operatorname{grad} L(y) dy. \tag{51}$$

Substituting (48), (49), (50) and (51) into (47) gives

$$2g_{\rho_t}(\nabla_{\partial_t \rho_t} \operatorname{grad} L, h_t) \overset{(48)}{=} (D) + (F) + (H)$$

$$= \int \left(\partial_t \mathcal{G}(\rho_t) \cdot \operatorname{grad} L + 2\mathcal{G}(\rho_t) \partial_t \operatorname{grad} L(\rho_t) \right) h_t$$

$$= \int \left(\partial_t \frac{\delta L}{\delta \rho_t} + \mathcal{G}(\rho_t) \partial_t \operatorname{grad} L(\rho_t) \right) h_t.$$

Thus, the conclusion follows:

$$\mathcal{G}(\rho_t) \nabla_{\partial_t \rho_t} \operatorname{grad} L = \frac{1}{2} \partial_t \frac{\delta L}{\delta \rho_t} + \frac{1}{2} \mathcal{G}(\rho_t) \partial_t \operatorname{grad} L(\rho_t). \tag{52}$$

With (46) and (52) substituted into (5), we arrive at:

$$\partial_t \Phi_t + \alpha_t \Phi_t - \frac{1}{2} \frac{\delta}{\delta \rho_t} \left(\int \partial_t \rho_t \mathcal{G}(\rho_t) \partial_t \rho_t \, dx \right)$$

$$+ \frac{\beta_t}{2} \partial_t \frac{\delta L}{\delta \rho_t} + \frac{\beta_t}{2} \underbrace{\mathcal{G}(\rho_t) \partial_t \operatorname{grad} L(\rho_t)}_{(I)} + \gamma_t \frac{\delta L}{\delta \rho_t} = 0. \tag{53}$$

To compute (I), the following expansion is applied:

$$(I) = \mathcal{G}(\rho_t) \partial_t \left(\mathcal{G}(\rho_t)^{-1} \frac{\delta L}{\delta \rho_t} \right) = \mathcal{G}(\rho_t) \partial_t \left(\mathcal{G}(\rho_t)^{-1} \right) \frac{\delta L}{\delta \rho_t} + \partial_t \frac{\delta L}{\delta \rho_t}$$

$$\overset{\partial_t \rho_t = \mathcal{G}(\rho_t)^{-1} \Phi_t}{=\!=\!=\!=\!=\!=\!=\!=\!=} \mathcal{G}(\rho_t) \left[\frac{\partial \left(\mathcal{G}(\rho_t)^{-1} \right)}{\partial \rho_t} \cdot \mathcal{G}(\rho_t)^{-1} \Phi_t \right] \frac{\delta L}{\delta \rho_t} + \partial_t \frac{\delta L}{\delta \rho_t}. \tag{54}$$

We ultimately establish the conclusion by making transformation $\Psi_t = \Phi_t + \beta_t \frac{\delta L}{\delta \rho_t}$, and employing $\frac{\delta}{\delta \rho} \left(\int \partial_t \rho_t \mathcal{G}(\rho) \partial_t \rho_t \, dx \right) \Big|_{\rho = \rho_t} = - \frac{\delta}{\delta \rho} \left(\int \Phi_t \mathcal{G}(\rho)^{-1} \Phi_t \, dx \right) \Big|_{\rho = \rho_t}$. This equality is proved in Section A.2 of [29]. $\qquad \square$

5.2 Analysis of ODE-flow

In this subsection, we aim to prove the convergence of the ODE trajectory (5) with $\alpha_t = \alpha/t$. We first provide a technical lemma that analyzes two quantities related to the metrics.

Lemma 1. Let $T_t = \log_{\rho_t} \rho^*$ denote the exponential map from ρ_t to ρ^*. For the H^s $(s \in \mathbb{Z})$ metric, Fisher-Rao metric, and Wasserstein-2 metric, the following inequality holds:

$$g(\partial_t \rho_t, \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t) \geq 0. \tag{55}$$

For the H^s $(s \in \mathbb{Z})$ metric and Fisher-Rao metric, we further have:

$$g(\operatorname{grad} F(\rho_t), \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t) \geq g(\operatorname{grad} F(\rho_t), \partial_t \rho_t) - \| \operatorname{grad} F(\rho_t) \|_g \cdot \| \partial_t \rho_t \|_g. \tag{56}$$

Proof. H^s *metrics.* The proof is straightforward, as the Riemannian manifold is flat. We have $T_t = \rho^* - \rho_t$, and $\nabla_{\partial_t \rho_t} T_t = \partial_t T_t = -\partial_t \rho_t$. This immediately gives the desired conclusion (55) and (56).

Fisher-Rao metric. According to Proposition 10 in [29], the geodesic connecting τ_0 and τ_1 for any $\tau_0, \tau_1 \in \mathcal{M}$ is given by

$$\tau_t(x) = \frac{1}{\sin^2 H} \left[\sin(Ht) \sqrt{\tau_1(x)} + \sin(H(1-t)) \sqrt{\tau_0(x)} \right]^2, \quad 0 \leq t \leq 1,$$

16

where $H = \cos^{-1}(\int \sqrt{\tau_0(x)\tau_1(x)}dx) \in [0, \frac{\pi}{2})$. This leads to the expression:

$$T_t = \frac{2H_t}{\sin(H_t)}\sqrt{\rho_t \rho^*} - \frac{2H_t \cos(H_t)}{\sin(H_t)}\rho_t, \tag{57}$$

where $H_t = \cos^{-1}(\int \sqrt{\rho_t(x)\rho^*(x)}dx) \in [0, \frac{\pi}{2})$. We denote $[f]_{\rho_t} = f - \mathbb{E}_{\rho_t}[f]$. It yields:

$$
\begin{aligned}
\partial_t H_t &= -\frac{1}{2\sin H_t}\int \sqrt{\frac{\rho^*}{\rho_t}}\partial_t \rho_t \, dx = -\frac{1}{2\sin H_t}\int \sqrt{\rho^* \rho_t}[\Phi_t]_{\rho_t} \, dx, \\
\partial_t T_t &= -\frac{\sin H_t - H_t \cos H_t}{\sin^3 H_t}\left(\int \sqrt{\rho^* \rho_t}[\Phi_t]_{\rho_t} dx\right)\left[\sqrt{\frac{\rho^*}{\rho_t}}\right]_{\rho_t}\rho_t \\
&\quad + \frac{H_t}{\sin H_t}\left(\sqrt{\frac{\rho^*}{\rho_t}}[\Phi_t]_{\rho_t} - \int \sqrt{\rho^* \rho_t}[\Phi_t]_{\rho_t} dx\right)\rho_t - \frac{2H_t \cos H_t}{\sin H_t}[\Phi_t]_{\rho_t}\rho_t.
\end{aligned}
\tag{58}
$$

Thus for any $f \in T^*_{\rho_t}\mathcal{M}$, we have:

$$
\begin{aligned}
\int \partial_t T_t[f]_{\rho_t} &= -\frac{\sin H_t - H_t \cos H_t}{\sin^3 H_t}\left(\int \left[\sqrt{\frac{\rho^*}{\rho_t}}\right]_{\rho_t}[\Phi_t]_{\rho_t}\rho_t \, dx\right)\int \left[\sqrt{\frac{\rho^*}{\rho_t}}\right]_{\rho_t}[f]_{\rho_t}\rho_t \, dx \\
&\quad + \frac{H_t}{\sin H_t}\int \sqrt{\rho^* \rho_t}[\Phi_t]_{\rho_t}[f]_{\rho_t} \, dx - \frac{2H_t \cos H_t}{\sin H_t}\int [\Phi_t]_{\rho_t}[f]_{\rho_t}\rho_t \, dx.
\end{aligned}
\tag{59}
$$

The Cauchy-Schwarz inequality implies that:

$$
\begin{aligned}
&\left(\int \left[\sqrt{\frac{\rho^*}{\rho_t}}\right]_{\rho_t}[\Phi_t]_{\rho_t}\rho_t dx\right)\left(\int \left[\sqrt{\frac{\rho^*}{\rho_t}}\right]_{\rho_t}[f]_{\rho_t}\rho_t dx\right) \\
&\leq \left(\int \left[\sqrt{\frac{\rho^*}{\rho_t}}\right]^2_{\rho_t}\rho_t dx\right)\left(\int [\Phi_t]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}}\left(\int [f]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}} \\
&= \sin^2 H_t \left(\int [\Phi_t]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}}\left(\int [f]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}}.
\end{aligned}
\tag{60}
$$

Since $\tan H_t \geq H_t$ for all $H_t \in [0, \frac{\pi}{2})$, substituting (60) into (59) yields:

$$
\begin{aligned}
\int \partial_t T_t[f]_{\rho_t} dx &= -\frac{\sin H_t - H_t \cos H_t}{\sin H_t}\left(\int [\Phi_t]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}}\left(\int [f]^2_{\rho_t}\rho_t dx\right)^{\frac{1}{2}} \\
&\quad + \frac{H_t}{\sin H_t}\int \sqrt{\rho^* \rho_t}[\Phi_t]_{\rho_t}[f]_{\rho_t} dx - \frac{2H_t \cos H_t}{\sin H_t}\int [\Phi_t]_{\rho_t}[f]_{\rho_t}\rho_t dx.
\end{aligned}
\tag{61}
$$

We now consider the expression in (55):

$$
\begin{aligned}
&g(\partial_t \rho_t, \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t) \\
&= \frac{d}{dt}g(\partial_t \rho_t, T_t) - g(\nabla_{\partial_t \rho_t}\partial_t \rho_t, T_t) + g(\partial_t \rho_t, \partial_t \rho_t) \\
&\stackrel{(46)}{=} \int (\partial_t \Phi_t T_t + [\Phi_t]_{\rho_t}\partial_t T_t)dx - \int \left(\partial_t \Phi_t - \frac{1}{2}\frac{\delta g(\partial_t \rho_t, \partial_t \rho_t)}{\delta \rho_t}\right)T_t dx + g(\partial_t \rho_t, \partial_t \rho_t) \\
&\stackrel{(61)}{\geq} -\frac{\sin H_t - H_t \cos H_t}{\sin H_t}\int [\Phi_t]^2_{\rho_t}\rho_t dx + \frac{H_t}{\sin H_t}\int \sqrt{\rho^* \rho_t}[\Phi_t]^2_{\rho_t} dx + \int [\Phi_t]^2_{\rho_t}\rho_t dx \\
&\quad - \frac{2H_t \cos H_t}{\sin H_t}\int [\Phi_t]^2_{\rho_t}\rho_t dx - \frac{1}{2}\int [\Phi_t]^2_{\rho_t}\left(\frac{2H_t}{\sin(H_t)}\sqrt{\rho_t \rho^*} - \frac{2H_t \cos(H_t)}{\sin(H_t)}\rho_t\right)dx \\
&= 0.
\end{aligned}
$$

17

A similar approach is applied to handle (56), which gives

$$
g(\operatorname{grad} F(\rho_t), \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t)
$$

$$
= \frac{d}{dt} g(\operatorname{grad} F(\rho_t), T_t) - g(\nabla_{\partial_t \rho_t} \operatorname{grad} F(\rho_t), T_t) + g(\operatorname{grad} F(\rho_t), \partial_t \rho_t)
$$

$$
\overset{(61)}{\geq} - \frac{\sin H_t - H_t \cos H_t}{\sin H_t} \left(\int [\Phi_t]_{\rho_t}^2 \rho_t dx \right)^{\frac{1}{2}} \left(\int \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t}^2 \rho_t dx \right)^{\frac{1}{2}}
$$

$$
+ \frac{H_t}{\sin H_t} \int \sqrt{\rho^* \rho_t} [\Phi_t]_{\rho_t} \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t} dx - \frac{2 H_t \cos H_t}{\sin H_t} \int [\Phi_t]_{\rho_t} \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t} \rho_t dx
$$

$$
- \frac{1}{2} \int \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t} [\Phi_t]_{\rho_t} \left(\frac{2 H_t \sqrt{\rho_t \rho^*}}{\sin H_t} - \frac{2 H_t \cos H_t}{\sin H_t} \rho_t \right) dx + \int [\Phi_t]_{\rho_t} \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t} \rho_t dx
$$

$$
\geq - \left(\int [\Phi_t]_{\rho_t}^2 \rho_t \right)^{\frac{1}{2}} \left(\int \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t}^2 \rho_t dx \right)^{\frac{1}{2}} + \int [\Phi_t]_{\rho_t} \left[\frac{\delta F}{\delta \rho_t} \right]_{\rho_t} \rho_t dx,
$$

where the last inequality uses Cauchy-Schwarz inequality. Thus, the conclusion (55) and (56) holds for Fisher-Rao metric.

Wasserstein-2 metric. Suppose that P_t is the optimal transport mapping from ρ_t to ρ^*. Hence, we have $P_{t\#}\rho_t = \rho^*$. By Brenier's Theorem [27], there exists a strictly convex function $c_t : \mathbb{R}^d \to \mathbb{R}$ such that $P_t = \nabla c_t$, and consequently, $\nabla P_t = \nabla^2 c_t$ is symmetric. Then according to the continuity equation, the exponential map is expressed as $T_t = -\nabla \cdot (\rho_t(P_t - id))$.

Next, we compute the time derivative of P_t. Noting that

$$
0 = \partial_t(P_t^{-1} \circ P_t) = \partial_t(P_t^{-1}) \circ P_t + \nabla(P_t^{-1}) \cdot \partial_t P_t, \tag{62}
$$

and $I_d = \nabla(P_t^{-1} \circ P_t) = \nabla(P_t^{-1}) \cdot \nabla P_t$, we can deduce

$$
\partial_t P_t(x) = -\nabla P_t(x) u_t(x), \tag{63}
$$

where $u_t = \partial_t(P_t^{-1}) \circ P_t$. Given that P_t is the optimal transport mapping, the distribution of $y_t = P_t^{-1}(y_0)$ is ρ_t for $y_0 \sim \rho^*$. Its velocity is given by $\partial_t y_t = u_t(y_t)$. By the continuity equation, we have

$$
\nabla \cdot (\rho_t u_t) = \nabla \cdot (\rho_t \nabla \Phi_t) = -\partial_t \rho_t. \tag{64}
$$

Based on the preceding discussion, we now proceed with the term $g(\partial_t \rho_t, \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t)$ as follows:

$$
\begin{aligned}
g(\partial_t \rho_t, \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t) &\overset{(46)}{=} \int \Phi_t \partial_t T_t dx + \int \frac{\delta g(\partial_t \rho_t, \partial_t \rho_t)}{\delta \rho_t} \frac{T_t}{2} dx + g(\partial_t \rho_t, \partial_t \rho_t) \\
&\overset{(b),(63)}{=} \int \langle \nabla \langle \nabla \Phi_t, P_t - x \rangle, \nabla \Phi_t \rangle \rho_t dx - \langle \nabla \Phi_t, \nabla P_t u_t \rangle \rho_t dx \\
&\quad + \frac{1}{2} \langle \nabla \| \nabla \Phi_t \|^2, P_t - x \rangle \rho_t dx + \int \| \nabla \Phi \|^2 \rho_t dx \\
&= \int \langle \nabla \Phi_t, \nabla P_t \nabla \Phi_t \rangle \rho_t dx - \int \langle \nabla \Phi_t, \nabla P_t u_t \rangle \rho_t dx,
\end{aligned} \tag{65}
$$

where $(a), (b)$ use integration by parts. The equations (63) and (64) imply that:

$$
\int \langle \nabla \Phi_t - u_t, \nabla P_t u_t \rangle \rho_t = - \int \langle \nabla \Phi_t - u_t, \partial_t \nabla c_t \rangle \rho_t = \int \nabla \cdot (\rho_t(\nabla \Phi_t - u_t)) \partial_t c_t = 0.
$$

Substituting it into (65) and using the semi-definiteness of $\nabla P_t = \nabla^2 c_t$, we obtain the following inequality:

$$
g(\partial_t \rho_t, \nabla_{\partial_t \rho_t} T_t + \partial_t \rho_t) = \int \langle \nabla \Phi_t - u_t, \nabla P_t(\nabla \Phi_t - u_t) \rangle \rho_t dx \geq 0.
$$

Thus, the proof is complete. $\qquad \square$

Before presenting the convergence theorem, we define $w_t = \gamma_t - \dot{\beta}_t - \beta_t/t$ and $\delta_t = t^2\left(\gamma_t + (\alpha - 3)\beta_t/(2t) - \dot{\beta}_t\right)$.

Theorem 1. *Assume that the target function L is geodesically convex towards ρ on the manifold $\mathcal{M}$, and L attains its minimum at ρ^*. Let $\rho : [t_0, +\infty) \to \mathcal{M}$ ($t_0 > 0$) be a solution trajectory of (5). Suppose that $\alpha_t = \alpha/t$ with $\alpha > 1$, and $\gamma_t > 0$. Then if $w_t > 0$ and $\dot{\delta}_t \leq 2tw_t(\alpha - 1)$ hold for the H^s ($s \in \mathbb{Z}$) metric and the Fisher-Rao metric, or $\beta_t \equiv 0$ holds for the Wasserstein-2 metric, we have*

$$L(\rho_t) - L(\rho^*) = \mathcal{O}\left(\frac{1}{t^2 w_t}\right) \quad as \quad t \to \infty. \tag{66}$$

Proof. For notational convenience, we use a dot over a variable to denote its derivative with respect to time. Consider the following Lyapunov function

$$E_t = \delta_t\left(L(\rho_t) - L(\rho^*)\right) + \frac{1}{2}g(v_t, v_t) + \frac{\alpha - 1}{2}g(\log_{\rho_t}(\rho^*), \log_{\rho_t}(\rho^*)), \tag{67}$$

where $v_t = -(\alpha - 1)\log_{\rho_t}(\rho^*) + 2t(\dot{\rho}_t + \beta_t \operatorname{grad} L(\rho_t))$. Since ρ_t is the trajectory of the ODE (5), we can calculate the derivative of v_t as

$$\nabla_{\dot{\rho}_t} v_t = (\alpha - 1)(-\nabla_{\dot{\rho}_t}\log_{\rho_t}(\rho^*) - \dot{\rho}_t) - (\alpha - 1)\dot{\rho}_t - 2tw_t \operatorname{grad} L(\rho_t). \tag{68}$$

It gives

$$\frac{d}{dt}E_t = \underbrace{\dot{\delta}_t(L(\rho_t) - L(\rho^*)) + 2tw_t(\alpha - 1)g(\log_{\rho_t}(\rho^*), \operatorname{grad} L(\rho_t))}_{(A)}$$
$$- 4t^2\beta_t\delta_t g(\operatorname{grad} L(\rho_t), \operatorname{grad} L(\rho_t)) - \underbrace{(\alpha - 1)^2 g(\log_{\rho_t}(\rho^*), -\nabla_{\dot{\rho}_t}\log_{\rho_t}(\rho^*) - \dot{\rho}_t)}_{(B)}$$
$$- 2t(\alpha - 1)g(\dot{\rho}_t, \dot{\rho}_t)^2 + \underbrace{2t(\alpha - 1)g(\dot{\rho}_t + \beta_t \operatorname{grad} L(\rho_t), -\nabla_{\dot{\rho}_t}\log_{\rho_t}(\rho^*) - \dot{\rho}_t)}_{(C)}.$$

Since the target function L is geodesically convex, we have for part (A):

$$(A) \leq \left(\dot{\delta}_t - 2tw_t(\alpha - 1)\right)(L(\rho_t) - L(\rho^*)) \leq 0.$$

For (B), we have the following equation

$$(B) = -\frac{(\alpha - 1)^2}{2}\frac{d}{dt}\operatorname{dist}(\rho_t, \rho^*)^2 + \frac{(\alpha - 1)^2}{2}\frac{d}{dt}\operatorname{dist}(\rho_t, \rho^*)^2 = 0.$$

One more term occurs different from the Euclidean case is (C). By Lemma 1, for H^s, Fisher-Rao and Wasserstein-2 metrics ($\beta_t = 0$), it yields

$$\frac{d}{dt}E_t \leq -4t^2\beta_t\delta_t\|\operatorname{grad} L(\rho_t)\|_g^2 - 2t(\alpha - 1)\|\dot{\rho}_t\|_g^2 + 4t(\alpha - 1)\beta_t\|\operatorname{grad} L(\rho_t)\|_g\|\dot{\rho}_t\|_g$$
$$= -2t(\alpha - 1)(\|\dot{\rho}_t\|_g - \beta_t^2\|\operatorname{grad} L(\rho_t)\|_g)^2$$
$$- 2t\beta_t(2t\delta_t - \beta_t(\alpha - 1))\|\operatorname{grad} L(\rho_t)\|_g^2 \leq 0.$$

The convergence rate (66) follows from the monotonic decreasing property of E_t. $\square$

6 Numerical Experiments

In this section, we give some numerical examples to show how machine learning problems can be fitted in form (1). Our method exhibits superior numerical performance on these examples.

6.1 The Burgers' Equation

This subsection addresses the Burgers' equation, which is known for its challenges associated with shock waves and discontinuities. The equation, supplemented with boundary data $h(x)$, is formalized as follows:

$$u_t + uu_x - \frac{0.01}{\pi} u_{xx} = 0, \quad x \in [-1, 1], \quad t \in [0, 1],$$
$$u(0, x) = h(x), \quad u(t, -1) = u(t, 1) = 0. \tag{69}$$

Let $\Omega = [-1, 1] \times [0, 1]$ and define $\partial \Omega_p = \{-1, 1\} \times [0, 1] \cup [-1, 1] \times \{0\}$. A neural network u_θ is employed to approximate the solution with six hidden layers with $(20, 50, 80, 80, 50, 20)$ neurons. The associated PDE and boundary loss for u_θ are:

$$L(u_\theta) = \|(u_\theta)_t + u_\theta(u_\theta)_x - \frac{0.01}{\pi}(u_\theta)_{xx}\|^2_{L^2(\Omega)} + \lambda \|u_\theta - g\|^2_{L^2(\partial\Omega)}, \tag{70}$$

where the function $g(x, t) = h(x)$ for $(x, t) \in [-1, 1] \times \{0\}$ and $g(x, t) = 0$ for $(x, t) \in \{-1, 1\} \times [0, 1]$ represents the initial and boundary conditions. Taking u_θ on the L^2 space, the training problem (70) can be fitted into (1).

In our investigation, we evaluate the efficacy of the ANGD method, comparing against the stochastic gradient descent (SGD) algorithm, Adam [13] and natural gradient method without acceleration (NGD). The most important difference between the ANGD method and the NGD method is whether or not acceleration is considered on the manifold. We employ a systematic grid search to identify hyper-parameters for several algorithms. For Adam, we vary the initial learning rate among $\{0.001, 0.005, 0.01\}$, the parameters for the momentum terms $\beta_1 \in \{0.9, 0.99\}$ and $\beta_2 \in \{0.99, 0.999\}$, and the weight decay from the set $\{0, 1e\text{-}4, 5e\text{-}5\}$. Similarly, for SGD, ANGD and NGD, the optimal configurations are determined by grid searching across the same ranges for the initial learning rate and weight decay. The ODE parameters α_k and β_k are initially set to the optimal values chosen from $\{0.01, 0.05, 0.1, 0.15\}$ and subsequently decay linearly.

We examine two distinct boundary conditions $h(x) = \sin(\pi x)$ and $h(x) = 1 - \cos(2\pi x)$, each presenting varying levels of training difficulty. The efficacy of ANGD is demonstrated in Figures 1 and 2. We consider the training loss and the testing loss versus iterations. ANGD demonstrates substantial convergence improvements over NGD, while surpassing both Adam and SGD. The lowest loss is also attained by the ANGD method. As a natural gradient method, ANGD consistently maintains a significantly lower testing loss compared to Adam and SGD after acceleration.

Figure 1: Numerical results for boundary condition $h(x) = \sin(\pi x)$

6.2 The Euler Equations

In this subsection, we solve the following conservative hyperbolic PDE:

$$\frac{\partial U}{\partial t} + \nabla \cdot F = 0. \tag{71}$$

20

Figure 2: Numerical results for boundary condition $h(x) = 1 - \cos(2\pi x)$

A notable example is the Euler equations, which represent a complex fluid dynamics problem frequently involving discontinuities. For the one-dimensional case of the Euler equations, the vectors U and F are defined as $U = (\rho, \rho u, E) \in \mathbb{R}^3$, $F = (\rho u, \rho u^2 + p, u(E + p)) \in \mathbb{R}^3$. In the context of an ideal gas, ρ symbolizes the density, u represents the velocity, p denotes the pressure, and $E = \frac{1}{2}\rho u^2 + \frac{p}{0.4}$ is the total energy. We employ a neural network $g_\theta = (\rho_\theta, u_\theta, p_\theta)$ with input (x, t) designed to simultaneously approximate ρ, u, and p. Through this parametrization, we can get the value of vectors U and F as U_θ and F_θ. The initial condition is the same as the Sod problem, which has been extensively studied. It is a 1D Riemann problem with the initial constant states in a tube with unit length formulated as

$$g(x, 0) = (\rho, u, p) = \begin{cases} (1, -2, 0.4) & \text{if } 0 \leq x \leq 0.5, \\ (1, 2, 0.4) & \text{if } 0.5 < x \leq 1. \end{cases}$$

We test the performance of the network at $t = 0.2s$. To quantify the performance of our model, the loss function on the area $\Omega = [0, 1] \times [0, 0.2]$ is given as

$$L(\rho_\theta, u_\theta, p_\theta) = \|(U_\theta)_t + \nabla \cdot F_\theta\|_{L^2(\Omega)}^2 + \lambda\|g_\theta - g(x, 0)\|^2. \tag{72}$$

Here we consider the manifold $L^2(\Omega)^{\otimes 3}$, allowing (72) to be fitted in the form of (1). We evaluate the efficacy of ANGD method using L^2 metric, comparing its performance against SGD, Adam, and non-accelerated NGD. The hyper-parameters are set as described in Section 6.1. Figure 3 illustrates the evolution of training and testing loss with respect to iterations. The ANGD method demonstrates a faster convergence rate in terms of training loss compared with the conventional method and the non-accelerated NGD, highlighting its efficiency in optimizing PINNs. Moreover, the ANGD method achieves a substantially lower testing loss, indicating generalization and improved alignment between the network's predictions and the ground truth.

6.3 Many-body quantum problem

We consider a many-body quantum system with N electrons $x = \{x_1, ..., x_N\} \in \mathbb{R}^{3N}$. The wavefunction $\psi_\theta : x \to \mathbb{R}$ describing the quantum state of the system is typically parameterized using neural networks, such as Ferminet [21]. The goal is to solve for the ground state energy and wavefunction, which is formulated as a variational problem:

$$\min_\theta \frac{\int_{x \in \mathbb{R}^{3N}} \psi_\theta(x)(\mathcal{H}\psi_\theta)(x)dx}{\int_{x \in \mathbb{R}^{3N}} \psi_\theta(x)^2 dx} = \int_{x \in \mathbb{R}^{3N}} \sqrt{\rho_\theta}(x)(\mathcal{H}\sqrt{\rho_\theta})(x)dx \stackrel{\triangle}{=} L(\rho_\theta), \tag{73}$$

where $\mathcal{H}$ is a Hamiltonian operator, and $\rho_\theta := \frac{\psi_\theta^2}{\int_{x \in \mathbb{R}^{3N}} \psi_\theta^2 dx}$ is a probability density. From this, we derive $\frac{\delta L}{\delta \rho_\theta} = \frac{\mathcal{H}\sqrt{\rho_\theta}}{\sqrt{\rho_\theta}} = \frac{\mathcal{H}\psi_\theta}{\psi_\theta}$ and $\partial_\theta \log \rho_\theta(x) = 2(\partial_\theta \psi_\theta(x) - \mathbb{E}_{\rho_\theta}[\partial_\theta \psi_\theta])$.

21

Figure 3: Numerical results for solving the Euler equations

Figure 4: Numerical results of VMC on the molecules Be, Li_2, H_{10}, N_2. We use "FR" and "W2" to denote the Fisher-Rao and Wasserstein-2 metrics, respectively, for brevity.

Variational Monte Carlo (VMC) methods utilize Markov Chain Monte Carlo (MCMC) sampling to estimate expectations with an unnormalized probability distribution. Using this approach, we conduct numerical experiments on a small atom (Be), and three molecules (Li_2, H_{10}, N_2), to verify the acceleration effects of our proposed ANGD algorithms on the Fisher-Rao metric using projected momentum discretization outlined in (23), and the Wasserstein-2 metric with KFAC discretization. Notably, the non-accelerated NGD-Fisher-Rao algorithm corresponds to the SPRING algorithm [9], while the non-accelerated NGD-Wasserstein-2 algorithm is essentially the WQMC algorithm [18], differing primarily in numerical stability techniques.

The experimental setup is as follows. The sample size n is set to 2000 for the Fisher-Rao metric and 4000 for the Wasserstein-2 metric. The initial learning rate h_0 is searched in $\{0.001, 0.005, 0.01, 0.05, 0.1\}$

for the baseline NGD algorithms, and in $\{\sqrt{0.001}, \sqrt{0.005}, \sqrt{0.01}, \sqrt{0.02}, \sqrt{0.05}\}$ for the ANGD algorithms. Following the approach in [9], all algorithms employ a linear decay schedule $h_k = \frac{h_0}{1+\epsilon k}$ with $\epsilon = 5\text{e-}5$ for ANGD and $\epsilon = 1\text{e-}4$ for NGD (doubled due to the square root). We also impose linear decay on α_k, β_k with $\alpha_0 \in \{0.1/h_0, 0.2/h_0, 0.5/h_0\}$, $\beta_0 \in \{0.0, 0.01, 0.05, 0.1, 0.15\}$ for ANGD methods, and set $\beta_k \equiv 0$ for ANGD-Wasserstein-2. Other hyper-parameters including clipping, sampling steps, and the Ferminet architecture follow [9].

Comparisons of the performance between the ANGD and NGD algorithms on the four benchmark particles are shown in Figure 4. We normalize the loss (energy) them by subtracting the physical lower bound (reported in Hartrees to four decimal places). For both the Fisher-Rao and Wasserstein-2 metrics, the ANGD methods demonstrate significantly faster convergence rates and attain lower final losses than the NGD methods. Remarkably, even the worst-performing ANGD variant outperforms the best NGD variant in terms of final loss, highlighting the significant benefits of incorporating acceleration into the optimization process.

7   Conclusion

In this paper, we introduce a novel ANGD framework for solving parametrized manifold optimization problems. An ARG flow is designed to characterize accelerated optimization on a manifold, incorporating Hessian-driven damping. An equivalent system of first-order ODEs for several metrics is proposed. We develop a discretization scheme to project the ODE flow onto the parameter space, leading to an efficient solution of the accelerated direction. The convergence analysis of ARG flow under convexity assumptions is also established. Numerical experiments show that ANGD accelerates the optimization process compared to other methods.

References

[1] Foivos Alimisis, Antonio Orvieto, Gary Becigneul, and Aurelien Lucchi. A Continuous-time Perspective for Modeling Acceleration in Riemannian Optimization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1297–1307. PMLR, 26–28 Aug 2020.

[2] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[3] Michael Arbel, Arthur Gretton, Wuchen Li, and Guido Montúfar. Kernelized Wasserstein natural gradient. In *International Conference on Learning Representations*, 2020.

[4] Hedy Attouch, Aïcha Balhag, Zaki Chbani, and Hassan Riahi. Fast convex optimization via inertial dynamics combining viscous and Hessian-driven damping with time rescaling. *Evolution Equations and Control Theory*, 11(2):487–514, 2022.

[5] Hedy Attouch, Zaki Chbani, Jalal Fadili, and Hassan Riahi. First-order optimization algorithms via inertial systems with Hessian driven damping. *Mathematical Programming*, 193(1):113–155, 2022.

[6] Achraf Bahamou, Donald Goldfarb, and Yi Ren. A Mini-Block Fisher Method for Deep Neural Networks. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 9191–9220, 25–27 Apr 2023.

[7] Leonardo Biliotti and Francesco Mercuri. Riemannian Hilbert Manifolds. In *Hermitian–Grassmannian Submanifolds*, pages 261–271, Singapore, 2017. Springer Singapore.

[8] Ao Chen and Markus Heyl. Empowering deep neural quantum states through efficient optimization. *Nature Physics*, 20(9):1476–1481, 2024.

[9] Gil Goldshlager, Nilin Abrahamsen, and Lin Lin. A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions. *Journal of Computational Physics*, 516, 8 2024.

[10] Roger Grosse and James Martens. A Kronecker-factored approximate Fisher matrix for convolution layers. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 573–582. PMLR, 20–22 Jun 2016.

[11] Xiaoyu He, Zibin Zheng, Yuren Zhou, and Chuan Chen. QNG: A Quasi-Natural Gradient Method for Large-Scale Statistical Learning. *SIAM Journal on Optimization*, 32(1):228–255, 2022.

[12] Yijie Jin, Shu Liu, Hao Wu, Xiaojing Ye, and Haomin Zhou. Parameterized Wasserstein gradient flow. *Journal of Computational Physics*, 524:113660, 2025.

[13] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Yi-An Ma, Niladri S. Chatterji, Xiang Cheng, Nicolas Flammarion, Peter L. Bartlett, and I Jordan, Michael. Is there an analog of Nesterov acceleration for gradient-based MCMC? *Bernoulli*, 27(3):1942–1992, aug 2021.

[15] James Martens. New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[16] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2408–2417. JMLR.org, 2015.

[17] Andreas Munk, Alexander Mead, and Frank Wood. Achieving high accuracy with pinns via energy natural gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, 2023.

[18] Kirill Neklyudov, Jannes Nys, Luca Thiede, Juan Felipe Carrasquilla Alvarez, qiang liu, Max Welling, and Alireza Makhzani. Wasserstein Quantum Monte Carlo: A Novel Approach for Solving the Quantum Many-Body Schrödinger Equation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[19] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[20] Levon Nurbekyan, Wanzhou Lei, and Yunan Yang. Efficient natural gradient descent methods for large-scale PDE-based optimization problems. *SIAM Journal on Scientific Computing*, 45(4):A1621–A1655, 2023.

[21] David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Physical review research*, 2(3):033429, 2020.

[22] 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. *Journal of Computational Physics*, 378:686–707, 2019.

[23] Maziar Raissi. Deep hidden physics models: deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(1):932–955, January 2018.

[24] Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42159–42191. PMLR, 21–27 Jul 2024.

[25] Bin Shi, Simon S. Du, Michael I. Jordan, and Weijie J. Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, 195(1-2), 2021.

[26] Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights. *Journal of Machine Learning Research*, 17(153):1–43, 2016.

[27] Cédric Villani. *Optimal Transport: Old and New*. Springer, Berlin Heidelberg, 2009.

[28] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

[29] Yifei Wang and Wuchen Li. Accelerated Information Gradient Flow. *Journal of Scientific Computing*, 90(1), jan 2022.

[30] Hao Wu, Shu Liu, Xiaojing Ye, and Haomin Zhou. Parameterized Wasserstein Hamiltonian Flow. *SIAM Journal on Numerical Analysis*, 63(1):360–395, 2025.

[31] Jiayuan Wu, Jiang Hu, Hongchao Zhang, and Zaiwen Wen. Convergence analysis of an adaptively regularized natural gradient method. *IEEE Transactions on Signal Processing*, 72:2527–2542, 2024.

[32] Minghan Yang, Dong Xu, Qiwen Cui, Zaiwen Wen, and Pengxiang Xu. An Efficient Fisher Matrix Approximation Method for Large-Scale Neural Network Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5391–5403, 2023.

[33] Minghan Yang, Dong Xu, Zaiwen Wen, Mengyun Chen, and Pengxiang Xu. Sketch-based empirical natural gradient methods for deep learning. *J. Sci. Comput.*, 92(3), September 2022.