Finite-Time Analysis of Stochastic Nonconvex Nonsmooth Optimization on the Riemannian Manifolds

Emre Sahinoglu
Northeastern University
sahinoglu.m@northeastern.edu

Youbang Sun
Tsinghua University
ybsun@mail.tsinghua.edu.cn

Shahin Shahrampour
Northeastern University
s.shahrampour@northeastern.edu

Abstract

This work addresses the finite-time analysis of nonsmooth nonconvex stochastic optimization under Riemannian manifold constraints. We adapt the notion of Goldstein stationarity to the Riemannian setting as a performance metric for nonsmooth optimization on manifolds. We then propose a Riemannian Online to NonConvex (RO2NC) algorithm, for which we establish the sample complexity of $O(\epsilon^{-3}\delta^{-1})$ in finding (δ, ϵ)-stationary points. This result is the first-ever finite-time guarantee for fully nonsmooth, nonconvex optimization on manifolds and matches the optimal complexity in the Euclidean setting. When gradient information is unavailable, we develop a zeroth order version of RO2NC algorithm (ZO-RO2NC), for which we establish the same sample complexity. The numerical results support the theory and demonstrate the practical effectiveness of the algorithms.

1 Introduction

Gradient-based iterative optimization algorithms are frequently used as numerical solvers for machine learning problems, many of which deal with search spaces with manifold structures. This includes deep learning [61], natural language processing [34], principal component analysis (PCA) [19], dictionary learning [11, 56], Gaussian mixture models [29] and low-rank matrix completion [27]. Riemannian optimization methods [1, 5, 54] offer a toolkit to solve optimization problems with manifold constraints and have attracted great interest due to their wide range of applications. However, unfortunately, conventional Riemannian optimization algorithms require access to the derivative of a *smooth* function, often falling short in highly nonsmooth, nonconvex problems such as training neural networks.

In this paper, we consider stochastic Riemannian optimization of an objective function $f : \mathcal{M} \to \mathbb{R}$,

$$\min_{x\in\mathcal{M}} \Big\{ f(x) := \mathbb{E}_\nu[F(x,\nu)] \Big\}, \tag{1}$$

where F is a stochastic *nonsmooth* objective function, defined on a d-dimensional complete manifold $\mathcal{M}$ that can be embedded in the Euclidean space, and ν corresponds to random data samples. Many important problems can be formulated as nonsmooth Riemannian optimization over smooth manifolds, including sparse PCA, compressed modes in physics, unsupervised feature selection, and robust low-rank matrix completion [10, 30]. This problem setting is also commonly encountered in deep neural networks, where optimization algorithms must be able to cope with deep nonlinear layers with ReLU activations [2, 20, 21, 55, 69].

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Table 1: Convergence rates of various algorithms over nonsmooth, nonconvex objectives: Composite objectives are in the form of $f(x) + h(x)$, where f is smooth and h is possibly nonsmooth and convex. Fully nonsmooth objectives can be written as $f(x) = E_\nu[F(x, \nu)]$, where F is nonsmooth and nonconvex. Our work is the first to provide the finite-time analysis of the *fully* nonsmooth setting. † and ‡ indicate that the objective is defined on Stiefel manifold or a compact manifold, respectively.

REFERENCE	METHOD	SETTING	OBJECTIVE	CONVERGENCE
[23]	SUBGRADIENT	DETERMINISTIC	NONSMOOTH	ASYMPTOTIC
[10]	PROXIMAL GRADIENT	DETERMINISTIC	COMPOSITE †	$O(\epsilon^{-2})$
[59]	PROX. GRAD. - SPIDER	STOCHASTIC	COMPOSITE †	$O(\epsilon^{-3})$
[44]	R-ADMM	DETERMINISTIC	COMPOSITE ‡	$O(\epsilon^{-4})$
[16]	AUG. LAGRANGIAN	DETERMINISTIC	COMPOSITE ‡	$O(\epsilon^{-3})$
[16]	AUG. LAGRANGIAN	STOCHASTIC	COMPOSITE ‡	$\tilde{O}(\epsilon^{-3.5})$
[50]	SMOOTHING	DETERMINISTIC	COMPOSITE ‡	$O(\epsilon^{-3})$
[50]	SMOOTHING	STOCHASTIC	COMPOSITE ‡	$O(\epsilon^{-5})$
OUR WORK	SUBGRADIENT	STOCHASTIC	NONSMOOTH	$O(\delta^{-1}\epsilon^{-3})$

There exists a scant literature on the theory of nonsmooth Riemannian optimization, mainly focusing on the asymptotic results and leaving the *finite-time analysis* unexplored. In fact, even in the Euclidean setup, finite-time results for the nonsmooth, nonconvex setting have been studied only recently in the last few years. This is perhaps not surprising; the finite-time analysis of smooth, nonconvex optimization is usually carried out by finding a stationary point x such that $\|\nabla f(x)\| \leq \epsilon$. However, for nonsmooth, nonconvex functions, finding such an ϵ-stationary point is in fact intractable. Instead, the notion of Goldstein stationarity has recently been analyzed by [66] to provide non-asymptotic results. Goldstein stationarity does not directly evaluate $\|\nabla f(x)\|$ and instead considers the convex hull of subgradients of points in the δ-neighborhood of x.

To address nonsmooth problems on *Riemannian* manifolds, this stationarity criterion can be adapted to the Riemannian setting using the Riemannian δ-subdifferential definition of [32]. However, the finite-time analysis of finding such (δ, ϵ)-stationary points on Riemannian manifolds is a tantalizing challenge that has remained elusive.

Our Contributions. In this paper, we address the *finite-time* analysis of *nonsmooth, nonconvex stochastic Riemannian* optimization:

- We adapt the notion of Goldstein stationarity [45, 66] to the Riemannian setting, providing a metric to evaluate the finite-time performance of nonsmooth, nonconvex optimization on manifolds. We then propose a Riemannian Online to NonConvex (RO2NC) algorithm that can be analyzed using regret bounds in online optimization. Our proposed algorithm can utilize retraction as a computationally efficient alternative for exponential mapping.
- We theoretically prove that RO2NC achieves the sample complexity of $O(\delta^{-1}\epsilon^{-3})$ in finding (δ, ϵ)-stationary points. We first establish this result using parallel transports (Theorem 3.1) and then extend it to projections (Theorem 3.3). This rate is the *first ever finite-time result for the fully nonsmooth, nonconvex optimization on manifolds* and matches the optimal sample complexity in the Euclidean setting [13]. It is derived under mild technical assumptions applicable to standard manifolds (e.g., Stiefel, Hadamard, Grassmann) without the need for restrictive assumptions on the objective function (e.g., weak convexity), which is assumed to be Lipschitz continuous.
- Extending our results to the *zeroth order* oracle setting, we show that ZO-RO2NC can achieve the same rate (Theorem 4.2) using a gradient estimator relying solely on stochastic function value queries. Our proposed gradient estimator samples vectors on the tangent space instead of the manifold, resulting in more computational efficiency without costly manifold-specific computations. We show that the gradient estimator satisfies certain distance bounds to the Goldstein subdifferential set, which is sufficient to achieve the Goldstein stationarity criterion. The main merit of this result is streamlining the zeroth order analysis without recourse to more challenging alternative estimators that require the calculation of volume and surface of manifolds, which could be computationally demanding.

1.1 Highlights of the Technical Analysis

RO2NC Algorithm Design. Developed by [13], O2NC is an optimal algorithm for finding Goldstein stationary points of nonsmooth, nonconvex objectives in the Euclidean space. O2NC works based on the interplay of two algorithms. At each epoch, an action Δ_t is generated via an online learning algorithm and is used to update the variable $x_{t+1} = x_t + \Delta_t$. Then, a gradient g_t is evaluated at a random point $w_t = x_t + s\Delta_t$ with $s \sim \texttt{unif}[0, 1]$ and is given to the online learning algorithm as a feedback to update Δ_t. (i) One difficulty in the Riemannian setting is that $x_{t+1} = x_t + \Delta_t$ does not ensure feasibility. We must keep the iterates $\{x_t\}$ on the manifold $\mathcal{M}$, which requires the use of retractions in the updates, i.e., $x_{t+1} = \mathrm{Retr}_{x_t}(\Delta_t)$. (ii) Also, since the action and feedback of the online learning algorithm belong to different tangent spaces, we must transport these vectors via parallel transports and/or projections. (i) and (ii) together introduce a technical challenge different from the Euclidean setting: we can no longer rely on the classical regret analysis in online optimization to streamline the intra-epoch analysis. Instead, we must creatively choose the right benchmark action at each epoch using projections and/or parallel transports to judiciously analyze the error terms caused by the manifold geometry.

Adapting Goldstein Stationarity to Riemannian Setting. In the Euclidean setting, the Goldstein subdifferential set is defined as a convex hull of Euclidean differential sets at different points. In the Riemannian setting, subdifferential sets at different points possibly belong to different tangent spaces, so it is required to transport those sets to a common tangent space. We choose the parallel transport operation in the definition of Riemannian Goldstein subdifferential set as it preserves the length of the vectors unlike the projection operation. Upper bounding the minimum norm element in the Goldstein differential set requires transportation of gradients at different points, where the distortion caused by parallel transport along the closed loops is analyzed with the curvature of the manifold (c.f. Lemma 2.9). Using a similar approach, we derive an upper bound on the distance between zeroth order gradient estimates and Goldstein subdifferential set by quantifying the effect of parallel transport along geodesic triangles (c.f. Lemma 4.1).

Zeroth Order Riemannian Gradient Estimator. In the Euclidean setting, the analysis of zeroth order gradient estimator for nonsmooth objectives is carried out through smoothing. The key is that the gradient of the smoothed objective f_δ is estimated by a stochastic gradient estimator g_δ, such that $\mathbb{E}[g_\delta(x)] = \nabla f_\delta(x)$ [45, Lemma E.1]. Extending this approach to the Riemannian setting introduces a great challenge. The smoothed objective f_δ necessitates the volume calculation of a manifold set, which is computationally expensive. To make the estimation practical, we define an objective h_δ based on efficiently sampling a vector in the tangent space, such that $\mathbb{E}[g_\delta(x)] = \mathrm{grad} h_\delta(x)$, where g_δ is the Riemannian gradient estimator. However, the inclusion property $\mathbb{E}[g_\delta(x)] \in \partial_\delta f(x)$ of the Euclidean setting does not hold anymore in the Riemannian setting. We address this technical challenge by deriving an upper bound on the distance between these two terms as a function of the curvature and δ (c.f. Lemma A.5). We further compute a bound on the Lipschitz constant of h_δ (Lemma A.3) and show that the $O(\delta^{-1}\epsilon^{-3})$ convergence rate is preserved.

1.2 Literature Review

Nonsmooth, Nonconvex Techniques in the Euclidean Setting. Given that calculating an ϵ-stationary point for nonsmooth, nonconvex functions is intractable in general, various assumptions or conditions have been proposed in the literature (e.g., weak convexity [8, 14]). In recent years, the finite-time convergence analysis of the Goldstein stationarity, proposed by [66], has gained much interest. Their algorithm is guaranteed to reach a (δ, ϵ)-stationary point with the complexity of $O(\epsilon^{-4}\delta^{-1})$. [13] improved this rate to $O(\epsilon^{-3}\delta^{-1})$ with the introduction of an online to nonconvex conversion method. The method was then extended to various settings, such as decentralized optimization [51].

For some optimization applications, gradient evaluations are not possible, and only zeroth order information is available. To solve these problems, [40] proposed a gradient estimator, which is calculated at point x and requires evaluation of $f(x + tv)$. In nonsmooth, nonconvex optimization, the convergence rate for finding (δ, ϵ)-stationary points with gradient-free methods was analyzed by [45]. Later [37] proved that the optimal dimensional dependence of $O(d)$ in the zeroth order setting could be achieved with O2NC.

Manifold Optimization. Due to the unique properties of manifolds, many Euclidean optimization algorithms have been adapted to the Riemannian setting. Utilizing manifold operations, these

methods are able to match the convergence rates of their Euclidean counterparts. These Riemannian algorithms include gradient descent methods [3, 65], projection-free methods [63, 64], and accelerated methods [36, 48]. The convergence results also cover various settings, such as min-max [35, 49, 68], variance reduction [54], online optimization [6, 46, 52, 53, 62], and decentralized optimization [7, 57]. However, the majority of the studies here focus on *smooth* and sometimes geodesically convex objective functions, leaving the nonsmooth optimization relatively underexplored.

For the gradient-free optimization setting, zeroth order methods have been adapted from Euclidean to the Riemannian setting [42, 43, 60]. The Riemannian zeroth order methods have also been extended to different settings, such as online Riemannian algorithms [46, 52, 62] and accelerated algorithms [26].

Nonsmooth, Nonconvex Optimization on Manifolds. The optimization task becomes more challenging when the objective function is nonsmooth and nonconvex over the Riemannian manifold. Some techniques have been developed and analyzed in nonsmooth, nonconvex optimization, such as (i) subgradient-oriented methods, (ii) proximal point methods, and (iii) operator splitting methods. In subgradient-oriented methods [4, 17, 18, 23, 25, 32], at iteration t a direction g_t in the tangent space of the manifold is calculated from the current or previous subgradient evaluations. The next iterate x_{t+1} is then calculated based on manifold operations, such as retractions or exponential mappings. For proximal point algorithms [9, 15, 28, 33], the algorithm solves a sub-problem in each iteration. The objectives for the sub-problems are formulated by an approximation of the original function combined with an additional penalty term. However, in some cases, solving the sub-problems is just as difficult as the original problem, making these methods difficult to implement. The operator-splitting methods [39] divide the original problem into several sub-problems that are easy to solve using techniques such as the alternating direction method of multipliers (ADMM). These methods either lack convergence guarantees [38] or need further technical conditions [67].

2 Preliminaries

In this section, we first provide a brief introduction on manifolds and useful notations/definitions. Then, we introduce Goldstein stationarity and state our technical assumptions.

2.1 Background on Manifold Optimization

We consider the optimization task in equation 1 defined on $\mathcal{M}$, which is an embedded submanifold in the Euclidean space. We denote by $T_x\mathcal{M}$ the tangent space of the manifold at point x. We denote the sphere (and ball) with radius r on $T_x\mathcal{M}$ by $\mathbb{S}_{T_x\mathcal{M}}(r)$ (and $\mathbb{B}_{T_x\mathcal{M}}(r)$), respectively. The set of pairs (x, ξ_x) where $\xi_x \in T_x\mathcal{M}$ is referred to as the tangent bundle, denoted by $T\mathcal{M}$. Similarly, the set of pairs (x, s_x) such that $\langle s_x, \xi_x\rangle = 0$ for every $\xi_x \in T_x\mathcal{M}$ is called the normal bundle, denoted by $N\mathcal{M}$. Let $\mathfrak{X}(\mathcal{M})$ denote space of vector fields on $\mathcal{M}$ and $\nabla : T\mathcal{M} \times T\mathcal{M} \to T\mathcal{M}, (\xi, \eta) \to \nabla_\xi \eta \in T\mathcal{M}$ denote the Levi-Civita connection on manifold $\mathcal{M}$. We adapt the Euclidean metric to the embedded manifold and use $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ to denote the inner product and norm, respectively.

Geodesics on the manifolds are generalizations of lines in Euclidean space, curves with constant speed that are locally distance-minimizing. Consequently, we can define the distance between two points on the manifold as the length of geodesic γ, $\mathrm{dist}(x, y) := \inf_\gamma \int_0^t \|\gamma'(t)\| dt$, where $\gamma(0) = x$ and $\gamma(1) = y$. With the help of geodesics, we next introduce the *exponential mapping* and *retraction* operations. From an optimization perspective, both exponential mapping and retraction are used to traverse manifold $\mathcal{M}$. The exponential mapping on a Riemannian manifold defines a geodesic on the manifold $\gamma(t) = \mathrm{Exp}_x(tv)$ and the distance between x and $\mathrm{Exp}_x(v)$ is $\mathrm{dist}(x, \mathrm{Exp}_x(v)) = \|v\|$. Exponential mappings are hard to compute in general; as mitigation, retractions $\mathrm{Retr}_x(\cdot) : T_x\mathcal{M} \to \mathcal{M}$ are introduced as first-order approximations of exponential mappings and are easier to compute. Based on the definition of distance, we denote by $B(x, \delta) := \{y \in \mathcal{M} \ : \mathrm{dist}(x, y) \le \delta\}$ a ball centered at x with radius δ [32].

Given that the tangent space $T_x\mathcal{M}$ is defined with respect to point x, vectors defined in different tangent spaces are not directly comparable. To this end, we can use the notion of *parallel transport*, which is a linear, isometric mapping from one tangent space to another, defining a way to transport the local geometry along a curve. We denote parallel transport along the minimizing geodesic by $P^g_{x,y} : T_x\mathcal{M} \to T_y\mathcal{M}$, which preserves the inner products such that $\langle u, v\rangle = \langle P^g_{x,y}(u), P^g_{x,y}(v)\rangle$

for $u, v \in T_x\mathcal{M}$. As a computationally efficient alternative to parallel transports, we can use vector transports [1], a specific case of which is projection. For $x, y \in \mathcal{M}$ the orthogonal projection onto tangent space can be defined as $\mathrm{Proj}_{T_y\mathcal{M}} : T_x\mathcal{M} \to T_y\mathcal{M}$, mapping $\xi \in T_x\mathcal{M}$ to $\mathrm{Proj}_{T_y\mathcal{M}}(\xi) \in T_y\mathcal{M}$ such that $\langle \mathrm{Proj}_{T_y\mathcal{M}}(\xi), \xi - \mathrm{Proj}_{T_y\mathcal{M}}(\xi)\rangle = 0$. While more efficient than parallel transports, vector transports (and projections) are not necessarily isometric, requiring more delicate analysis.

2.2 Goldstein Stationarity on Riemannian Manifolds

Using concepts defined in Riemannian geometry, we next provide the following notions for optimization on manifolds. As a result of Rademacher's theorem and local equivalence of the Riemannian distance with Euclidean distance in a chart, every Lipschitz function defined on a Riemannian manifold $\mathcal{M}$ is differentiable almost everywhere with respect to the Lebesgue measure on $\mathcal{M}$ [32].

Definition 2.1. We define the Riemannian subdifferential of f at x, denoted by $\partial f(x) := \mathrm{conv}\{\lim_{l\to\infty} \mathrm{grad} f(x_l) : x_l \to x, x_l \in \Omega_f\} \subset T_x\mathcal{M}$, where $\Omega_f : \{x \in \mathcal{M} : f \text{ is differentiable at } x\}$ and conv denotes the convex hull operator.

Here, $\mathrm{grad} f(x)$ denotes the Riemannian gradient, defined as the unique tangent vector that satisfies $df(x)[\xi] = \langle \mathrm{grad} f(x), \xi\rangle$ for all $\xi \in T_x\mathcal{M}$. Although the sequence $\{\mathrm{grad} f(x_l)\}_l$ lies in different tangent spaces $\{T_{x_l}\mathcal{M}\}_l$, the limit $\lim_{l\to\infty} \mathrm{grad} f(x_l)$ can still be defined in a weak sense. Specifically, it is characterized as the vector satisfying $\lim_{l\to\infty}\langle \mathrm{grad} f(x_l), X(x_l)\rangle \to \langle \mathrm{grad} f(x), X(x)\rangle$ for any smooth vector field X on $\mathcal{M}$ [23]. For a smooth function f defined on an embedded submanifold in the Euclidean space $\mathrm{grad} f(x) = \mathrm{Proj}_{T_x\mathcal{M}}(\nabla f(x))$. This also holds for the subdifferential sets and the Riemannian subdifferential set is the projection of the Euclidean subdifferential set onto the tangent space at x.

Definition 2.1 extends the notion of the Clarke subdifferential to the Riemannian setting, following prior work such as [23, 31]. Among the various subdifferential concepts, the Clarke subdifferential is particularly useful due to its inclusivity and well-behaved analytical properties, which facilitate the study of convergence in nonsmooth optimization. We next provide the definition for δ-subdifferential in the neighborhood of point x.

Definition 2.2. Let f be a Lipschitz continuous function on $\mathcal{M}$. δ-subdifferential of $x \in \mathcal{M}$ is defined as $\partial_\delta f(x) := \mathrm{cl}\ \mathrm{conv}\{P^g_{y,x}(\partial f(y)) : y \in \mathrm{cl}\ B(x,\delta)\}$, where cl denotes the closure.

Different from the Euclidean δ-subdifferential set, the Riemannian δ-subdifferential requires a transport operation $P^g_{y,x}$ since the tangent spaces are not identical. Next, we introduce the Riemannian version of Goldstein stationarity.

Definition 2.3. Given a Lipschitz continuous function $f : \mathcal{M} \to \mathbb{R}$, a point $x \in \mathcal{M}$ and $\delta > 0$, denote $\left\|\mathrm{grad} f(x)\right\|_\delta := \min\{\|g\| : g \in \partial_\delta f(x)\}$. A point x is called a (δ, ϵ)-stationary point of $f(\cdot)$ if $\left\|\mathrm{grad} f(x)\right\|_\delta \le \epsilon$.

This definition generalizes the Euclidean Goldstein stationarity to the Riemannian setting. The main difference is that parallel transport operations and Riemannian subdifferentials are used in δ-subdifferential set due to Riemannian geometry.

Let X and Y be vector fields on $\mathcal{M}$. Since $\mathcal{M} \subset \mathbb{R}^n$ is an embedded submanifold of Euclidean space equipped with Euclidean metric, X and Y can be extended in $\mathbb{R}^n$. Applying ambient covariant derivative $\tilde{\nabla}$ and decomposing it into tangential and normal components gives $\tilde{\nabla}_X Y = (\tilde{\nabla}_X Y)^T + (\tilde{\nabla}_X Y)^\perp$. The *second fundamental form* [43] is $\mathrm{II} : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \Gamma(NM)$ given by $\mathrm{II}(X, Y) = (\tilde{\nabla}_X Y)^\perp$ where $\Gamma(N\mathcal{M})$ denotes the space of smooth normal vector fields.

2.3 Technical Assumptions

We now introduce a series of assumptions about the problem setting, all of which are considered standard in the relevant literature and are needed for our analytical results. We have the following assumption on the manifold and its second fundamental form.

Assumption 2.4. We assume that the manifold $\mathcal{M}$ is an embedded submanifold of the Euclidean space and that the norm of the second fundamental form is bounded by C for all unit vectors $\xi, \eta \in T\mathcal{M}$, i.e., for all $\left\|\xi\right\| = \left\|\eta\right\| = 1$ we have $\left\|\mathrm{II}(\xi, \eta)\right\| \le C$. This implies that the curvature

tensor $R(X,Y)Z$ is bounded by a constant K_c, i.e., $\|R(X,Y)Z\| \leq K_c\|X\|\|Y\|\|Z\|$, and the sectional curvature is bounded by a constant K_s for any vector fields $X, Y, Z \in \mathfrak{X}(\mathcal{M})$.

The above assumption on the bounded second fundamental form connects the extrinsic and intrinsic geometries. As discussed in [42], it is a stronger condition than bounded sectional curvature, but it is required to measure the extrinsic difference on vectors, caused by transport operations. For more discussion, see Appendix C. In Riemannian optimization, variable updates may involve exponential mappings and retractions. Although exponential mapping has theoretically favorable properties, retractions are preferred in practice due to their computational efficiency. We have the following assumptions on retraction curves.

Assumption 2.5. For the retraction curve $\mathrm{Retr}_x(t\xi)$ with $\xi \in T_x\mathcal{M}$, we assume that

$$\|\frac{d}{dt}\mathrm{Retr}_x(t\xi)\| \leq \|\xi\| \qquad \text{and} \qquad \|\frac{d^2}{dt^2}\mathrm{Retr}_x(t\xi)\| \leq C'\|\xi\|^2,$$

where C' is a constant depending on the manifold properties.

Since retractions are approximations of exponential mappings, the above assumption can be thought as the cost of applying retractions instead of exponential mappings. It is presented in a simplified form for analytical clarity; however, the proposed implementation can be readily extended to more general retraction-based settings, such as projection-based or smooth first-order retractions (see Appendix B). Assumption 2.5 is satisfied by various matrix manifolds and commonly used retraction curves. In particular, for exponential mappings on any manifold $\mathcal{M}$, the first condition holds with equality, while the second condition holds with $C' = C$, where C is defined in Assumption 2.4. On the Stiefel manifold $St(p,n) = \{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}$, for polar decomposition retraction $\mathrm{Retr}_X(t\xi) = (X + t\xi)(I_p + t^2\xi^\top\xi)^{-\frac{1}{2}}$ Assumption 2.5 holds with $C' = 1$ (see Appendix B).

Next, we provide the following standard assumptions on the objective function, applicable to a diverse range of commonly used objectives.

Assumption 2.6. We assume that the objective function has the form $f(x) = \mathbb{E}_\nu[F(x,\nu)]$, where ν denotes the random index. The stochastic component of the function $F(\cdot,\nu) : \mathcal{M} \to \mathbb{R}$ is $L(\nu)$-Lipschitz for any ν, i.e., it holds that

$$|F(x,\nu) - F(y,\nu)| \leq L(\nu)\mathrm{dist}(x,y),$$

for any $x, y \in \mathcal{M}$. $L(\nu)$ has a bounded second moment such that $\mathbb{E}_\nu[L(\nu)^2] \leq L^2$. We also assume that f is lower bounded on $\mathcal{M}$, i.e., $\inf_{x\in\mathcal{M}} f(x) > -\infty$.

Assumption 2.7. We assume that the stochastic Riemannian oracle returns unbiased estimates, i.e.,

$$\mathbb{E}_\nu[\mathrm{grad}F(x,\nu)] = \mathrm{grad}f(x).$$

Furthermore, we assume that the second moment of the stochastic gradient is bounded such that $\mathbb{E}_\nu\left[\|\mathrm{grad}F(x,\nu)\|^2\right] \leq G^2$, and we denote its variance by σ^2.

Assumption 2.7 is standard in stochastic optimization with first order oracles [24]. With the definitions and assumptions provided above, we introduce the following lemma, which extends the result of [43, Theorem 4.1] on geodesic paths to broken geodesic paths.

Lemma 2.8. *Suppose $\mathcal{M}$ is an embedded submanifold of the Euclidean space with a second fundamental form bounded by C. Let $\gamma : [0,t] \to \mathcal{M}$ be a broken geodesic (a piecewise smooth curve with a finite number of curve segments, each of which is a geodesic) and $v \in T_{\gamma(0)}\mathcal{M}$. Then, we have*

$$\|\mathcal{P}^\gamma_{0,t}(v) - \mathrm{Proj}_{T_{\gamma(t)}\mathcal{M}}(v)\| \leq C\|v\|\mathrm{length}(\gamma),$$

where the parallel transport is computed along the path γ.

This result allows us to measure the distortion between an intrinsic parallel transport operation and an extrinsic projection operation on the manifold. We note that for the Euclidean case, the second fundamental form is 0. For the extension of Euclidean algorithms to the Riemannian setting, Lemma 2.8 characterizes the extra terms raised by the Riemannian geometry.

For a sequence of points $S_t = \{x_s\}_{s=0}^t$, let us define the parallel transport operator over S_t as $\mathcal{P}^s_{S_t} := P^g_{x_{t-1},x_t} \circ P^g_{x_{t-2},x_{t-1}} \circ ... \circ P^g_{x_0,x_1}$ and its inverse operator as $(\mathcal{P}^s_{S_t})^{-1}$. In general, $\|\mathrm{grad}f(y)\|_\delta$ is difficult to compute directly. The following lemma provides an analyzable upper bound on $\|\mathrm{grad}f(y)\|_\delta$, which will be used in our analysis.

Lemma 2.9. *Let Assumption 2.4 hold. For a sequence of points* $\{x_t\}_{t=0}^{T-1}$ *which satisfies* $\operatorname{dist}(x_t, x_{t+1}) \leq D \quad \forall t \in \{0, 1, \ldots, T-1\}$*, a set of gradient vectors* $\nabla_t \coloneqq \operatorname{grad} f(w_t) \in T_{w_t}\mathcal{M}$ *with* $\|\nabla_t\| \leq L$*,* $\operatorname{dist}(x_{t+1}, w_t) \leq D$*, and a point* y *that satisfies* $\operatorname{dist}(w_t, y) \leq \delta := DT, \quad \forall t \in \{0, 1, \ldots, T-1\}$ *, we have that*

$$\|\operatorname{grad} f(y)\|_\delta \leq \Big\|\frac{1}{T}\sum_{t=0}^{T-1} (\mathcal{P}^s_{S_{t+1}})^{-1} \circ P^g_{w_t, x_{t+1}}(\nabla_t)\Big\| + 3L\delta C.$$

The summands all live in $T_{x_0}\mathcal{M}$, so they can be summed. In the following sections, we design our algorithms and provide analytical bounds for the right-hand side of Lemma 2.9.

3 First Order Setting

In this section, we develop algorithms to find (δ, ϵ)-stationary points of nonsmooth nonconvex stochastic objectives and provide theoretical guarantees for their finite-time convergence rates. Our algorithms adapt the O2NC conversion of [13] to the Riemannian setting, and instead of directly implementing online Riemannian optimization, our specific formulation of the problem allows us to utilize techniques of Euclidean online algorithms with transport operations between tangent spaces.

3.1 The Structure of RO2NC versus O2NC

To understand O2NC, let us observe that for any algorithm with the update rule $x_{t+1} = x_t + \Delta_t$, one can write $f(x_{t+1}) = f(x_t) + \langle \Delta_t, \nabla_t \rangle$ where $\nabla_t = \int_0^1 \nabla f(x_t + s\Delta_t) ds$. For Euclidean problems, the O2NC approach proposed by [13] designed an algorithm to overcome the nonconvexity by observing that

$$f(x_T) - f(x_0) = \sum_{t=0}^{T-1} \langle g_t, \Delta_t - u \rangle + \sum_{t=0}^{T-1} \langle \nabla_t - g_t, \Delta_t \rangle + \sum_{t=0}^{T-1} \langle g_t, u \rangle,$$

where Δ_t can be generated via an online learning algorithm and g_t are the stochastic gradients provided to the online learning algorithm. The above equation holds for any u in hindsight, so the first summation corresponds to the regret of the online algorithm. For the last term we have freedom to select a suitable u. Specifically, the selection of $u = -D \sum_{t=0}^{T-1} \mathbb{E}[g_t] / \| \sum_{t=0}^{T-1} \mathbb{E}[g_t] \|$ with a suitable parameter D facilitates the Goldstein stationarity analysis. To have an efficient Riemannian adaptation for the O2NC algorithm, we encounter some major challenges:

(i) Since the iterates are constrained to be on the manifold, we must use retractions in the updates, so we have $x_{t+1} = \operatorname{Retr}_{x_t}(\Delta_t)$. Also, for the gradient evaluation part, we use the update $w_t = \operatorname{Retr}_{x_t}(s\Delta_t)$ for a randomly selected $s \sim \texttt{unif}[0, 1]$. In the Euclidean setting, two consecutive iterates are connected with a line segment as $x_{t+1} = x_t + \Delta_t$, parameterized to have zero acceleration. However, for manifolds with general retraction curves, we need to handle additional terms that contain the velocity and acceleration of the retraction curves.

(ii) Unlike the Euclidean case where we can directly add and subtract vectors, in the Riemannian setting different variables belong to various tangent spaces. Performing calculations using these variables necessitates operations such as parallel transport or projection, making the analysis more challenging compared to its Euclidean counterpart. This affects *both* algorithm design and technical analysis. For the design of RO2NC, we first consider the parallel transport in Section 3.2 and then consider a more efficient version of our algorithm with projections in Section 3.3. As for the analysis we cannot simply use the benchmark u as in [13], and we must cleverly design an optimal u, elaborated in the next section.

3.2 RO2NC with Parallel Transport

In this section, we consider the RO2NC algorithm where the actions Δ_t are updated via parallel transport operations. The updates are similar in vein to online gradient descent and allow us to rework the Euclidean regret analysis with the use of parallel transports. The algorithm is formally stated in Algorithm 1 and works based on an inner loop indexed by t (iteration) and an outer loop indexed by k (epoch). In this section, except in the algorithm outline, we omit k for notational convenience.

Algorithm 1 Riemannian Online to NonConvex (RO2NC)

Input: $K \in \mathbb{N}$, $T \in \mathbb{N}$, initial point $x_{0,T} \in \mathcal{M}$, clipping parameter D, step size $\eta = D/G\sqrt{T}$
for $k = 1$ **to** K **do**
  Initialize $x_{k,0} = x_{k-1,T}$, $\Delta_{k,0} = 0$
  for $t = 0$ **to** $T - 1$ **do**
    $x_{k,t+1} = \mathrm{Retr}_{x_{k,t}}(\Delta_{k,t})$
    $s_{k,t} \sim \mathtt{unif}[0, 1]$
    $w_{k,t} = \mathrm{Retr}_{x_{k,t}}(s_{k,t}\Delta_{k,t})$
    get gradient $g_{k,t} = \mathrm{grad}\, F(w_{k,t}, \nu_{k,t})$, where $\nu_{k,t}$ is a random index (based on data)
    if Using Parallel Transport **then**
      $g'_{k,t} = P^g_{w_{k,t},x_{k,t+1}}(g_{k,t}) \in T_{x_{k,t+1}}\mathcal{M}$
      set $\Delta_{k,t+1} = P^g_{x_{k,t},x_{k,t+1}}(\Delta_{k,t}) - \eta g'_{k,t}$
    end if
    if Using Projection **then**
      set $\Delta_{k,t+1} = \mathrm{Proj}_{T_{x_{k,t+1}}\mathcal{M}}(\Delta_{k,t} - \eta g_{k,t})$
    end if
    clip $\Delta_{k,t+1}$ on the convex set $\mathbb{B}_{T_{x_{k,t+1}}\mathcal{M}}(D)$
  end for
  Set $\bar{w}_k$ to $w_{k,\lfloor \frac{T}{2} \rfloor}$.
end for
Sample $w_{out} \sim \mathtt{unif}\{\bar{w}_1, ..., \bar{w}_K\}$
Output: w_{out}

At the t-th iteration in each epoch, we parallel transport both Δ_t and g_t to update $\Delta_{t+1} = \mathrm{clip}(P^g_{x_t,x_{t+1}}(\Delta_t) - \eta g'_t)$, where g'_t is the parallel transported version of g_t. We have the following convergence result for RO2NC.

Theorem 3.1. *Let $\delta, \epsilon \in (0, 1)$ and suppose that Assumptions 2.4,2.5,2.6,2.7 hold. Running Algorithm 1 with parallel transports for $N = KT$ rounds $T = O(\delta N)^{\frac{2}{3}}$ and $D = \delta/T$ gives an output that satisfies the following inequality*

$$\mathbb{E}[\left\|\mathrm{grad} f(w_{out})\right\|_\delta] \leq C_1(\delta N)^{-\frac{1}{3}} + 3\delta L C_2,$$

where constants C_1, C_2 are given in the Appendix E.

Remark 3.2. To find a (δ, ϵ)-stationary point, we can choose $\delta' = \min\{\delta, \frac{\epsilon}{6LC_2}\} \leq \delta$ to get a (δ', ϵ)-stationary point. Then, choosing $N = O(\delta^{-1}\epsilon^{-3})$ is sufficient for $\mathbb{E}[\left\|\mathrm{grad} f(w_{out})\right\|_\delta] \leq \epsilon$.

Theorem 3.1 indicates that for nonsmooth nonconvex optimization on Riemannian manifolds, RO2NC has the same complexity as its Euclidean counterpart [13]. The distortion caused by the curved geometry can be controlled with a suitable selection of parameters. Similar to the Euclidean setting, for smooth objectives, an ϵ-stationary point can be found in $O(\epsilon^{-4})$ iterations by selecting $\delta = O(\epsilon)$. A key innovation in the analysis of Theorem 3.1 is the choice of

$$u_t = \mathcal{P}^s_{S_t}\Big(- D\frac{\sum_{\tau=0}^{T-1}(\mathcal{P}^s_{S_{\tau+1}})^{-1} \circ P^g_{w_\tau,x_{\tau+1}}(\nabla_\tau)}{\left\|\sum_{\tau=0}^{T-1}(\mathcal{P}^s_{S_{\tau+1}})^{-1} \circ P^g_{w_\tau,x_{\tau+1}}(\nabla_\tau)\right\|}\Big),$$

where $\nabla_t = \mathbb{E}[g_t]$. The main idea behind this selection is to transport the gradient vectors along the path $\{x_T, ..., x_1, x_0\}$ to create a base vector u_0 and then choose $u_t = \mathcal{P}^s_{S_t}(u_0)$. Although the best actions for Algorithm 1 are time-dependent, they can still be analyzed with parallel transports.

3.3 RO2NC with Projection

We now address the case where parallel transport operations are costly and RO2NC use projections instead. While more efficient, the introduction of projections bring forward technical challenges as they lack the isometry property (unlike parallel transports). In this case, at the t-th iteration in each epoch, we calculate $\Delta_t - \eta g_t$ in the ambient space and project it to the tangent space of $\mathcal{M}$ at iterate x_{t+1} to obtain Δ_{t+1}. The convergence of Algorithm 1 with projection operations is presented in the following theorem.

Theorem 3.3. *Let $\delta, \epsilon \in (0,1)$ and suppose that Assumptions 2.4,2.5,2.6,2.7 hold. Running Algorithm 1 with projections for $N = KT$ rounds with $T = O(\delta N)^{\frac{2}{3}}$ and $D = \delta/T$ gives an output that satisfies the following inequality*

$$\mathbb{E}[\left\|\operatorname{grad} f(w_{out})\right\|_\delta] \leq C_3(\delta N)^{-\frac{1}{3}} + C_4\delta^{\frac{1}{3}}N^{-\frac{2}{3}} + \delta LC,$$

where constants C_3, C_4 are given in Appendix E.

Remark 3.4. Theorem 3.3 implies that $N = O(\delta^{-1}\epsilon^{-3})$ and $\delta = O(\epsilon)$ is sufficient to have $\mathbb{E}[\left\|\operatorname{grad} f(w_{out})\right\|_\delta] \leq \epsilon$, since $\delta < 1$ and $\delta^{\frac{1}{3}}N^{-\frac{2}{3}} < (\delta N)^{-\frac{1}{3}}$ order-wise. So, we can follow the same argument in Remark 3.2.

Theorem 3.3 shows that the same complexity can be achieved using projections instead of parallel transport operations, greatly improving the efficiency of RO2NC from an implementation perspective. While the implementation of the algorithm relies solely on projection operations, the analysis makes use of parallel transport to upper bound the term $\left\|\operatorname{grad} f(w_{out})\right\|_\delta$ in our theorem.

The projection-based analysis allows us to first calculate $u = -D\sum_{t=0}^{T-1}\nabla_t/\|\sum_{t=0}^{T-1}\nabla_t\|$ directly in the ambient space and then project it back to the tangent space to get $u_t = \operatorname{Proj}_{T_{x_t}\mathcal{M}}(u)$, which again highlights a key novelty in our analysis.

4 Zeroth Order Setting

In this section, we consider the case where gradient queries are unavailable and only noisy function evaluations can be obtained. In the context of online learning, this is analogous to a system with two-point bandit feedback. One common approach to address the nonsmooth problems in this setting is to derive a gradient estimator g_δ based on function values and use that in the gradient-based algorithms as a replacement of $\operatorname{grad} F$.

For zeroth order optimization in the Euclidean setting [37], the gradient estimator is constructed with $F(x \pm \delta u, \nu)$, where u is uniformly sampled from a unit sphere. In the Riemannian setting, $x + \delta u$ is replaced with $\operatorname{Exp}_x(\delta u)$, and u is sampled uniformly from a sphere in $T_x\mathcal{M}$. Then, the Riemannian gradient estimator is given as follows,

$$g_\delta(x) = \frac{d}{2\delta}(F(\operatorname{Exp}_x(\delta u), \nu) - F(\operatorname{Exp}_x(-\delta u), \nu))u, \tag{2}$$

where d is the dimension of $T_x\mathcal{M}$. Two major problems arise from the Riemannian formulation: (i) In the Euclidean setting, we have $\nabla f_\delta(x) = \mathbb{E}_u[\nabla f(x + \delta u)] \in \partial_\delta f(x)$, which implies that the expectation of the gradient estimator is included in the Goldstein subdifferential set [45]. However, in the Riemannian setting, stating the relation between $\mathbb{E}_u[g_\delta(x)]$ and $\partial_\delta f(x)$ is a challenge in itself. (ii) Also, the geometric relation $\partial_v f_\rho(x) \subseteq \partial_{v+\rho} f(x)$ used in the Euclidean zeroth order analysis [37] does not hold in the Riemannian setting due to distortion caused by the manifold geometry.

We first define $h_\delta(x) := \int f \circ \operatorname{Exp}_x(u) dp_x(u)$, where p_x is a uniform measure over $\mathbb{B}_{T_x\mathcal{M}}(\delta) \subset T_x\mathcal{M}$. By analyzing the relationship between $\operatorname{grad} h_\delta(x)$ and $\partial_\delta f(x)$ in Lemma A.5, we introduce the following lemma to address the above two challenges.

Lemma 4.1. *Consider a point y and a set of points $\{x_t\}_{t=0}^{T-1}$ which satisfy $\operatorname{dist}(y, x_t) \leq \frac{\delta}{2}$. The gradient estimator g_δ satisfies $\mathbb{E}_u[g_\delta(x_t)] \in \partial_\delta f(x_t) + \mathbb{B}_{T_x\mathcal{M}}(\frac{1}{3}K_s L\delta^2)$ and $P^g_{x_t,y}(\partial_{\frac{\delta}{2}} f(x_t)) \subset \partial_\delta f(y) + \mathbb{B}_{T_y\mathcal{M}}(2CL\delta)$, where $+$ denotes the Minkowski sum of two sets, C denotes the bound on the second fundamental form (Assumption 2.4) and K_s bounds the sectional curvature of the manifold.*

With the help of Lemmas 4.1 and A.5 we bound $\left\|\operatorname{grad} f(w_{out})\right\|_\delta$ in terms of $\left\|\operatorname{grad} h_{\frac{\delta}{2}}(w_{out})\right\|_{\frac{\delta}{2}}$, and we then employ that inequality for the following theorem, providing the finite-time convergence guarantee using the zeroth order gradient estimator.

Theorem 4.2. *Let $\delta, \epsilon \in (0,1)$ and suppose that Assumptions 2.4,2.5,2.6 hold. Running Algorithm 1 for $N = KT$ rounds with $T = O(\delta N)^{\frac{2}{3}}$ and $D = \delta/T$ using the zeroth order gradient estimator in equation 2 gives an output that satisfies*

$$\mathbb{E}[\left\|\operatorname{grad} f(w_{out})\right\|_\delta] \leq C_5(\delta N)^{-\frac{1}{3}} + \delta LC_6,$$

where C_5 and C_6 are given in Appendix E.

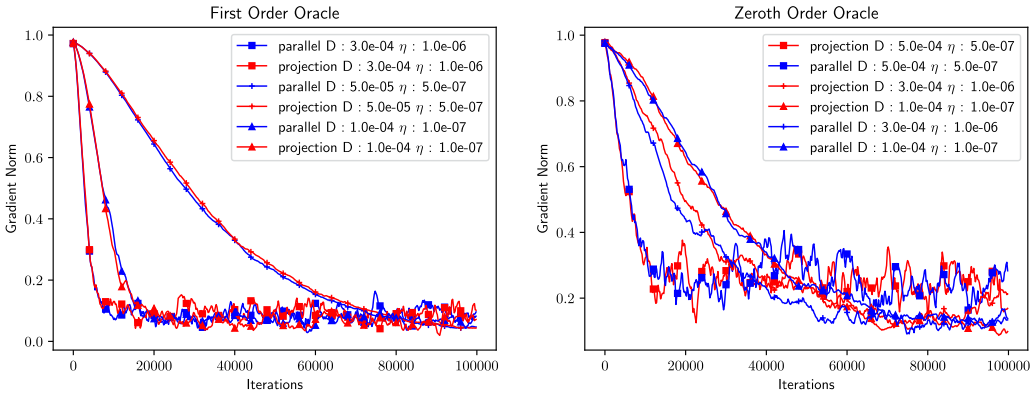

Figure 1: Evaluation of gradient norms in both settings; D: clipping parameter, η: step size.

The theorem considers Algorithm 1 with parallel transports, but a similar result can be obtained with projections. In our analysis, we show that although extra terms arise due to the introduction of zero order gradient estimator and the approximation to Goldstein subdifferential sets, these additional terms can be controlled by a careful adjustment of δ. Consequently, the overall iteration complexity with respect to (δ,ϵ) can be maintained in the zeroth order setting.

5 Numerical Experiments

We provide the following numerical experiments to validate our results.

Model and Setup. We consider the sparse principal component problem defined on the Euclidean unit sphere $\mathbb{S}^{n-1}$ in $\mathbb{R}^n$. The parallel transport operations have closed-form solutions on spheres. The objective function can be written as $\min_{x\in\mathbb{S}^{n-1}}\{-x^\top Ax + \mu\|x\|_1\}$, where $A = \mathbb{E}[\nu\nu^\top]$ and $\nu \sim \mathcal{N}(0, A)$ is sampled from a multivariate Gaussian distribution. For RO2NC, we choose our retraction curves to be $\mathrm{Retr}_x(v) = (x+v)/\|x+v\|$, and we use exponential mappings for calculating the gradient estimator in ZO-RO2NC.

Evaluation and Results. Since the direct evaluation of the Goldstein subdifferential set and $\|\mathrm{grad} f(w_{out})\|_\delta$ requires calculation over a convex hull, which is highly impractical, we instead evaluate $\|\frac{1}{T}\sum_{t=0}^{T-1}(\mathcal{P}^s_{S_{t+1}})^{-1}\circ P^g_{w_t,x_{t+1}}(\mathrm{grad} f(w_t))\|$ as an upper bound.

In our experiments for RO2NC, we illustrate the decay of the gradient norms. We run Algorithm 1 using both parallel transport and projection operations for $K = 500$ epochs, each consisting of $T = 200$ iterations. Convergence of the algorithms depends on the selection of parameters D and η. For both settings, η is chosen orders of magnitude smaller than D, and the plot is reported in Fig. 1. We can see that the performance of the projection approach is comparable with that of parallel transport approach. Optimization with the zeroth order oracle is more sensitive to the selection of parameters and the convergence is slower than the first order setting, but with a suitable choice of parameters ZO-RO2NC indeed converges.

6 Conclusion, Limitations, and Future Work

We addressed the finite-time analysis of nonsmooth, nonconvex stochastic Riemannian optimization. We proposed the RO2NC algorithm, which is guaranteed to find the Riemannian extension of (δ, ϵ)-Goldstein stationary points with $O(\delta^{-1}\epsilon^{-3})$ sample complexity. When gradient information is unavailable, we introduced ZO-RO2NC with a zeroth order gradient estimator, which also achieves optimal sample complexity. While our stationarity condition is defined via parallel transport of the Clarke subdifferential, alternative notions, such as different transport maps or subdifferentials, may also be considered. Furthermore, a deeper analysis of curvature effects and their role in optimality remains an open direction. Finally, designing adaptive algorithms that can exploit curvature information more effectively can be an interesting problem for future research.

Acknowledgments

The authors gratefully acknowledge the support of NSF ECCS-2240788 Award as well as Northeastern TIER 1 Program for this research.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[2] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR, 2016.

[3] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[4] Pierre B Borckmans, S Easter Selvan, Nicolas Boumal, and P-A Absil. A riemannian subgradient algorithm for economic dispatch with valve-point effect. *Journal of computational and applied mathematics*, 255:848–866, 2014.

[5] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[6] Hengchao Chen and Qiang Sun. Decentralized online riemannian optimization with dynamic environments. *arXiv preprint arXiv:2410.05128*, 2024.

[7] Shixiang Chen, Alfredo Garcia, Mingyi Hong, and Shahin Shahrampour. Decentralized riemannian gradient descent on the stiefel manifold. In *International Conference on Machine Learning*, pages 1594–1605. PMLR, 2021.

[8] Shixiang Chen, Alfredo Garcia, and Shahin Shahrampour. On distributed nonconvex optimization: Projected subgradient method for weakly convex problems in networks. *IEEE Transactions on Automatic Control*, 67(2):662–675, 2021.

[9] Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang. Proximal gradient method for nonsmooth optimization over the stiefel manifold. *SIAM Journal on Optimization*, 30(1):210–239, 2020.

[10] Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang. Nonsmooth optimization over the stiefel manifold and beyond: Proximal gradient method and recent variants. *SIAM Review*, 66(2):319–352, 2024.

[11] Anoop Cherian and Suvrit Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE transactions on neural networks and learning systems*, 28(12):2859–2871, 2016.

[12] Christopher Criscitiello and Nicolas Boumal. An accelerated first-order method for non-convex optimization on manifolds. *Foundations of Computational Mathematics*, 23(4):1433–1509, 2023.

[13] Ashok Cutkosky, Harsh Mehta, and Francesco Orabona. Optimal stochastic non-smooth non-convex optimization through online-to-non-convex conversion. In *International Conference on Machine Learning*, pages 6643–6670. PMLR, 2023.

[14] Damek Davis and Dmitriy Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[15] Glaydston de Carvalho Bento, João Xavier da Cruz Neto, and Paulo Roberto Oliveira. A new approach to the proximal point method: convergence on general riemannian manifolds. *Journal of Optimization Theory and Applications*, 168:743–755, 2016.

[16] Kangkang Deng, Jiang Hu, and Zaiwen Wen. Oracle complexity of augmented lagrangian methods for nonsmooth manifold optimization. *arXiv preprint arXiv:2404.05121*, 2024.

[17] Kangkang Deng, Zheng Peng, and Weihe Wu. Single-loop $\mathcal{O}(\epsilon^{-3})$ stochastic smoothing algorithms for nonsmooth riemannian optimization. *arXiv preprint arXiv:2505.09485*, 2025.

[18] Gunther Dirr, Uwe Helmke, and Christian Lageman. Nonsmooth riemannian optimization with applications to sphere packing and grasping. In *Lagrangian and Hamiltonian Methods for Nonlinear Control 2006: Proceedings from the 3rd IFAC Workshop, Nagoya, Japan, July 2006*, pages 29–45. Springer, 2007.

[19] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[20] Yanhong Fei, Yingjie Liu, Chentao Jia, Zhengyu Li, Xian Wei, and Mingsong Chen. A survey of geometric optimization for deep learning: from euclidean space to riemannian manifold. *ACM Computing Surveys*, 57(5):1–37, 2025.

[21] Yanhong Fei, Yingjie Liu, Xian Wei, and Mingsong Chen. O-vit: Orthogonal vision transformer. *arXiv preprint arXiv:2201.12133*, 2022.

[22] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[23] P Grohs and S Hosseini. ε-subgradient algorithms for locally lipschitz functions on riemannian manifolds. *Advances in Computational Mathematics*, 42(2):333–360, 2016.

[24] Philipp Grohs, Martin Holler, and Andreas Weinmann. *Handbook of Variational Methods for Nonlinear Geometric Data*. Springer, 2020.

[25] Philipp Grohs and Seyedehsomayeh Hosseini. Nonsmooth trust region algorithms for locally lipschitz functions on riemannian manifolds. *IMA Journal of Numerical Analysis*, 36(3):1167–1192, 2016.

[26] Chang He, Zhaoye Pan, Xiao Wang, and Bo Jiang. Riemannian accelerated zeroth-order algorithm: Improved robustness and lower query complexity. In *International Conference on Machine Learning*, pages 17972–18009. PMLR, 2024.

[27] Gennadij Heidel and Volker Schulz. A riemannian trust-region method for low-rank tensor completion. *Numerical Linear Algebra with Applications*, 25(6):e2175, 2018.

[28] Najmeh Hoseini Monjezi, Soghra Nobakhtian, and Mohamad Reza Pouryayevali. A proximal bundle algorithm for nonsmooth optimization on riemannian manifolds. *IMA Journal of Numerical Analysis*, 43(1):293–325, 2023.

[29] Reshad Hosseini and Suvrit Sra. Matrix manifold optimization for gaussian mixtures. *Advances in neural information processing systems*, 28, 2015.

[30] Seyedehsomayeh Hosseini, Boris Sholimovich Mordukhovich, and André Uschmajew. *Nonsmooth optimization and its applications*. Springer, 2019.

[31] Seyedehsomayeh Hosseini and MR Pouryayevali. Generalized gradients and characterization of epi-lipschitz sets in riemannian manifolds. *Nonlinear Analysis: Theory, Methods & Applications*, 74(12):3884–3895, 2011.

[32] Seyedehsomayeh Hosseini and André Uschmajew. A riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM Journal on Optimization*, 27(1):173–189, 2017.

[33] Wen Huang and Ke Wei. Riemannian proximal gradient methods. *Mathematical Programming*, 194(1):371–413, 2022.

[34] Pratik Jawanpuria, Mayank Meghwanshi, and Bamdev Mishra. Geometry-aware domain adaptation for unsupervised alignment of word embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3052–3058, 2020.

[35] Michael Jordan, Tianyi Lin, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. First-order algorithms for min-max optimization in geodesic metric spaces. *Advances in Neural Information Processing Systems*, 35:6557–6574, 2022.

[36] Jungbin Kim and Insoon Yang. Accelerated gradient methods for geodesically convex optimization: Tractable algorithms and convergence analysis. In *International Conference on Machine Learning*, pages 11255–11282. PMLR, 2022.

[37] Guy Kornowski and Ohad Shamir. An algorithm with optimal dimension-dependence for zero-order nonsmooth nonconvex stochastic optimization. *Journal of Machine Learning Research*, 25(122):1–14, 2024.

[38] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein. Madmm: a generic algorithm for non-smooth optimization on manifolds. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 680–696. Springer, 2016.

[39] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58:431–449, 2014.

[40] Jeffrey Larson, Matt Menickelly, and Stefan M Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[41] John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

[42] Jiaxiang Li, Krishnakumar Balasubramanian, and Shiqian Ma. Stochastic zeroth-order riemannian derivative estimation and optimization. *Mathematics of Operations Research*, 48(2):1183–1211, 2023.

[43] Jiaxiang Li, Krishnakumar Balasubramanian, and Shiqian Ma. Zeroth-order riemannian averaging stochastic approximation algorithms. *SIAM Journal on Optimization*, 34(4):3314–3341, 2024.

[44] Jiaxiang Li, Shiqian Ma, and Tejes Srivastava. A riemannian admm. *arXiv preprint arXiv:2211.02163*, 2022.

[45] Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35:26160–26175, 2022.

[46] Alejandro I Maass, Chris Manzie, Dragan Nesic, Jonathan H Manton, and Iman Shames. Tracking and regret bounds for online zeroth-order euclidean and riemannian optimization. *SIAM Journal on Optimization*, 32(2):445–469, 2022.

[47] Oren Mangoubi and Aaron Smith. Rapid mixing of geodesic walks on manifolds with positive curvature. *The Annals of Applied Probability*, 28(4):2501–2543, 2018.

[48] David Martínez-Rubio. Global riemannian acceleration in hyperbolic and spherical spaces. In *International Conference on Algorithmic Learning Theory*, pages 768–826. PMLR, 2022.

[49] David Martínez-Rubio and Sebastian Pokutta. Accelerated riemannian optimization: Handling constraints with a prox to bound geometric penalties. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 359–393. PMLR, 2023.

[50] Zheng Peng, Weihe Wu, Jiang Hu, and Kangkang Deng. Riemannian smoothing gradient type algorithms for nonsmooth optimization problem on compact riemannian submanifold embedded in euclidean space. *Applied Mathematics & Optimization*, 88(3):85, 2023.

[51] Emre Sahinoglu and Shahin Shahrampour. An online optimization perspective on first-order and zero-order decentralized nonsmooth nonconvex stochastic optimization. In *41st International Conference on Machine Learning*, volume 235, pages 43043–43059. PMLR, 2024.

[52] Emre Sahinoglu and Shahin Shahrampour. Decentralized online riemannian optimization beyond hadamard manifolds. *arXiv preprint arXiv:2509.07779*, 2025.

[53] Emre Sahinoglu and Shahin Shahrampour. Online optimization on hadamard manifolds: Curvature independent regret bounds on horospherically convex objectives. *arXiv preprint arXiv:2509.11236*, 2025.

[54] Hiroyuki Sato, Hiroyuki Kasai, and Bamdev Mishra. Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport. *SIAM Journal on Optimization*, 29(2):1444–1472, 2019.

[55] Suvrit Sra and Reshad Hosseini. Geometric optimization in machine learning. *Algorithmic Advances in Riemannian Geometry and Applications: For Machine Learning, Computer Vision, Statistics, and Optimization*, pages 73–91, 2016.

[56] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. *IEEE Transactions on Information Theory*, 63(2):885–914, 2016.

[57] Youbang Sun, Shixiang Chen, Alfredo Garcia, and Shahin Shahrampour. Retraction-free decentralized non-convex optimization with orthogonal constraints. *arXiv preprint arXiv:2405.11590*, 2024.

[58] Yue Sun, Nicolas Flammarion, and Maryam Fazel. Escaping from saddle points on riemannian manifolds. *Advances in Neural Information Processing Systems*, 32, 2019.

[59] Bokun Wang, Shiqian Ma, and Lingzhou Xue. Riemannian stochastic proximal gradient methods for nonsmooth optimization over the stiefel manifold. *Journal of machine learning research*, 23(106):1–33, 2022.

[60] Hongye Wang, Zhaoye Pan, Chang He, Jiaxiang Li, and Bo Jiang. Federated learning on riemannian manifolds: A gradient-free projection-based approach. *arXiv preprint arXiv:2507.22855*, 2025.

[61] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11505–11515, 2020.

[62] Xi Wang, Zhipeng Tu, Yiguang Hong, Yingyi Wu, and Guodong Shi. Online optimization over riemannian manifolds. *Journal of Machine Learning Research*, 24(84):1–67, 2023.

[63] Melanie Weber and Suvrit Sra. Projection-free nonconvex stochastic optimization on riemannian manifolds. *IMA Journal of Numerical Analysis*, 42(4):3241–3271, 2022.

[64] Melanie Weber and Suvrit Sra. Riemannian optimization via frank-wolfe methods. *Mathematical Programming*, 199(1):525–556, 2023.

[65] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *Conference on learning theory*, pages 1617–1638. PMLR, 2016.

[66] Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Suvrit Sra, and Ali Jadbabaie. Complexity of finding stationary points of nonconvex nonsmooth functions. In *International Conference on Machine Learning*, pages 11173–11182. PMLR, 2020.

[67] Junyu Zhang, Shiqian Ma, and Shuzhong Zhang. Primal-dual optimization algorithms over riemannian manifolds: an iteration complexity analysis. *Mathematical Programming*, 184(1):445–490, 2020.

[68] Peiyuan Zhang, Jingzhao Zhang, and Suvrit Sra. Sion's minimax theorem in geodesic metric spaces and a riemannian extragradient algorithm. *SIAM Journal on Optimization*, 33(4):2885–2908, 2023.

[69] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main contributions in this work are referenced at a high level in the abstract, expanded upon in the introduction, and developed in detail across Sections 3 and 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed limitations and future work in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have stated our theoretical assumptions in Section 2.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This is mostly a theory work, and experiments are for the proof of concept. We have provided the choice of all hyperparameters for reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The anonymized code for the numerical experiments is available at the following https://github.com/emreesahinoglu/RO2NC-RiemannianNonsmooth.git

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The same justification as Q4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Error bars do not apply to the convergence of our algorithms. We do not make comparisons.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The experiments are toy examples. They can be run on a laptop.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read the link and made sure the paper conforms with all components of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is a theory work at the interface of online and Riemannian optimization. We do not anticipate any negative societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There are no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We did not use LLMs for the derivation of results.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.