Faster Convergence of Riemannian Stochastic Gradient Descent with Increasing Batch Size

Kanata Oowada KANATA7527@GMAIL.COM
Hideaki Iiduka IIDUKA@CS.MEIJI.AC.JP
Meiji University, Japan

Editors: Hung-yi Lee and Tongliang Liu

Abstract

We theoretically analyzed the convergence behavior of Riemannian stochastic gradient descent (RSGD) and found that using an increasing batch size leads to faster convergence than using a constant batch size, not only with a constant learning rate but also with a decaying learning rate, such as cosine annealing decay and polynomial decay. The convergence rate improves from $O(T^{-1} + C)$ with a constant batch size to $O(T^{-1})$ with an increasing batch size, where T denotes the total number of iterations and C is a constant. Using principal component analysis and low-rank matrix completion, we investigated, both theoretically and numerically, how an increasing batch size affects computational time as quantified by stochastic first-order oracle (SFO) complexity. An increasing batch size was found to reduce the SFO complexity of RSGD. Furthermore, an increasing batch size was found to offer the advantages of both small and large constant batch sizes.

Keywords: Batch Size; Stochastic First-order Oracle Complexity; Learning Rate; Riemannian Optimization; Riemannian Stochastic Gradient Descent

1. Introduction

Stochastic gradient descent (SGD) (Robbins and Monro, 1951) is a basic algorithm, widely used in machine learning (Liu et al., 2021; He et al., 2016; Krizhevsky et al., 2012). Riemannian stochastic gradient descent (RSGD) was introduced in Bonnabel (2013). Riemannian optimization (Absil et al., 2008; Boumal, 2023), which addresses RSGD and its variants, has attracted attention due to its potential application in various tasks, including many machine learning tasks. For example, it has been used in principal components analysis (PCA) (Breloy et al., 2021; Liu and Boumal, 2020; Roy et al., 2018), low-rank matrix completion (Vandereycken, 2013; Boumal and Absil, 2015; Kasai and Mishra, 2016), convolutional neural networks (Wang et al., 2020; Huang et al., 2017), and graph neural networks (Zhu et al., 2020; Chami et al., 2019; Liu et al., 2019). It has also been used in applications of optimal transportation theory (Lin et al., 2020b; Weber and Sra, 2023). It is well established that the performance of Euclidean SGD strongly depends on both the batch size (BS) and learning rate (LR) settings (Goyal et al., 2017; Smith et al., 2018; Zhang et al., 2019; Lin et al., 2020a). In the context of Riemannian SGD, Ji et al. (2024); Bonnabel (2013); Kasai et al. (2019, 2018) addressed the use of a constant BS, and Sakai and Iiduka (2025); Han and Gao (2022) addressed the use of an adaptive BS, although the latter did not consider RSGD specifically. In parallel, various decaying LR strategies, including cosine annealing (Loshchilov and Hutter, 2017) and polynomial decay (Chen et al., 2018), have been devised, and their effectiveness has been demonstrated for the Euclidean case. Motivated by these

Study/Theorem	Batch Size	Learning Rate	Convergence Analysis
Bonnabel (2013)	–	s.a. LR	$\lim_{t\to\infty} \mathbb{E}[\mathrm{grad} f(x_t)] = 0$ a.s.
Zhang and Sra (2016)	–	$\eta_t = O((L_r + \sqrt{t})^{-1})$	$\mathbb{E}[f(x_T) - f^\star] = O(\sqrt{T}(C+T)^{-1})$
Tripuraneni et al. (2018)	Constant	$\eta_t = O(t^{-\alpha})$	$\mathbb{E}\|\Delta_T\| = O(T^{-\frac{1}{2}})$
Hosseini and Sra (2020)	Constant	$\eta = O(T^{-\frac{1}{2}})$	$\|G_T\|^2 = O(T^{-\frac{1}{2}})$
Durmus et al. (2021)	Constant	Constant	$\mathbb{E}[f(x_T) - f^\star] = O(C_1^T + C_2)$
Sakai and Iiduka (2024)	Constant	$\eta_t = O((t+1)^{-\frac{1}{2}})$	$\|G_T\|^2 = O(T^{-\frac{1}{2}} \log T)$
Sakai and Iiduka (2025)	Increasing	Constant	$\|G_T\|^2 = O(T^{-1})$
Theorem 5	Constant	Constant and Decay	$\|G_T\|^2 = O(T^{-1} + C)$
Theorem 6	Increasing	Constant and Decay	$\|G_T\|^2 = O(T^{-1})$
Theorem 7	Increasing	Warm-up	$\|G_T\|^2 = O((T - T_w)^{-1})$
Theorem 8	Constant	Warm-up	$\|G_T\|^2 = O((T - T_w)^{-1} + C)$

Table 1: Comparison of RSGD convergence analyses, where $\alpha \in (\frac{1}{2}, 1), C_1 \in (0,1), C_2$ and C are constants, and (x_t) is a sequence generated by RSGD. The usual stochastic approximation LR (s.a. LR) is defined as $\sum_{t=0}^{\infty} \eta_t = \infty, \sum_{t=0}^{\infty} \eta_t^2 < \infty$. $\mathbb{E}[\|\Delta_T\|] \coloneqq \mathbb{E}[\|R_{x^\star}^{-1}(x_T)\|]$. T is the total number of iterations, and T_w is the number of warm-up iterations (see Section 3.3). $\|G_T\|^2 \coloneqq \min_{t\in\{0,\cdots,T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2]$. 'Decay' includes LR schedules of the form $\eta_t = O(t^{-\frac{1}{2}})$. More detailed explanations of the convergence criteria are provided in Appendix F.

previous studies, we developed a novel convergence analysis of RSGD, proved that the convergence rate is improved with an increasing BS, and showed that the stochastic first-order oracle (SFO) complexity is reduced with an increasing BS. Our numerical results support our theoretical results.

1.1. Previous Results and Our Contributions

Constraint: Bonnabel (2013); Zhang and Sra (2016); Durmus et al. (2021); Sakai and Iiduka (2024) considered Hadamard manifolds or their submanifolds. Tripuraneni et al. (2018) treated Riemannian manifolds that satisfy the condition required to ensure a unique geodesic connecting any two points, including Hadamard manifolds. Sakai and Iiduka (2025) addressed embedded submanifolds of Euclidean space. In contrast, we considered general Riemannian manifolds that encompass all of the above conditions.

Learning rate: Bonnabel (2013) considered s.a. LRs, and Zhang and Sra (2016); Tripuraneni et al. (2018); Sakai and Iiduka (2024) considered diminishing LRs of the form $\eta_t = \frac{1}{t^a}$. Hosseini and Sra (2020); Durmus et al. (2021) considered constant LRs. Sakai and Iiduka (2025) used both constant and diminishing LRs. We considered constant, diminishing, cosine annealing, and polynomial decay LRs, as well as their warm-up versions.

Batch size: Because Bonnabel (2013) considered expected risk minimization using expectation instead of sample mean—as is customary in empirical risk minimization—the concept of BS is not generally applicable in the expected-risk-minimization setting. Zhang and Sra (2016); Tripuraneni et al. (2018); Hosseini and Sra (2020); Durmus et al. (2021); Sakai and Iiduka (2024) considered a constant BS. Sakai and Iiduka (2025) and this study considered

an increasing BS. Furthermore, we numerically and theoretically compared using a constant BS with using an increasing BS on the basis of SFO complexity. Our contributions are as follows: (i) we theoretically showed that using an increasing BS yields a more favorable SFO complexity of $O(\epsilon^{-2})$, whereas using a constant BS equal to the critical BS yields a rate of $O(\epsilon^{-4})$; (ii) we numerically observed that an increasing BS yields both a better optimal solution—in terms of attaining a smaller gradient norm—and a shorter computational time than either a small or a large constant BS (see Section 4.3 for more details).

Objective function: Bonnabel (2013) considered three times continuously differentiable functions, with both the gradient and Hessian uniformly bounded. Tripuraneni et al. (2018) treated functions that are twice continuously differential, subject to additional conditions on the Hessian. Other studies considered only once continuously differentiable functions. Zhang and Sra (2016) addressed convex and smooth functions, while Durmus et al. (2021) considered strongly convex and smooth functions. Hosseini and Sra (2020); Sakai and Iiduka (2024, 2025) treated more general classes of functions, specifically nonconvex functions with bounded gradients. We treated nonconvex functions without bounded gradients.

Convergence rate: A comparison of the previous and current RSGD convergence analyses is presented in Table 1; here we outline the key results. Tripuraneni et al. (2018) obtained a convergence rate of $O(T^{-\frac{1}{2}})$ (with a criterion other than the gradient norm) using an LR of $\eta = O(T^{-\frac{1}{2}})$; however, this rate decays to zero as $T \to \infty$, making it unsuitable for practical use. Durmus et al. (2021) attained exponential convergence under strong convexity and smoothness assumptions. Sakai and Iiduka (2025) attained a rate of $O(T^{-1})$ for several adaptive methods, including RSGD. Our work differs from Sakai and Iiduka (2025) in four key aspects : (i) we provide both theoretical and numerical evidence—based on SFO complexity—that using an increasing BS leads to better performance than using a constant BS; (ii) in addition to a constant LR and a diminishing LR, we addressed cosine annealing, polynomial decay, and warm-up LRs; (iii) Sakai and Iiduka (2024) focused on embedded submanifolds of Euclidean space unlike us; (iv) we do not assume bounded gradient norms, thereby expanding the range of applicable scenarios.

Contributions:

- We developed a convergence analysis of RSGD that incorporates an increasing BS, a cosine annealing LR, and a polynomial decay LR, under assumptions more general than those used in prior work. Our analysis shows that using an increasing BS improves the convergence rate of RSGD from $O(T^{-1} + C)$ to $O(T^{-1})$.

- We numerically and theoretically demonstrated—using PCA and low-rank matrix completion (LRMC) tasks—the advantage of an increasing BS in reducing computational time, as quantified by SFO complexity. Specifically, we observed that an increasing BS yields a better optimal solution (compared with a small constant BS) in terms of attaining a smaller gradient norm and a shorter computation time (compared with a large constant BS). These findings are consistent with our theoretical analysis, which shows that an increasing BS reduces the SFO complexity for achieving $\|G_T\|^2 \leq \epsilon^2$ from $O(\epsilon^{-4})$—with a constant BS set equal to the critical BS—to $O(\epsilon^{-2})$ in the experimental setting.

2. Preliminaries

Let $\mathcal{M}$ be a Riemannian manifold and $T_x\mathcal{M}$ denote the tangent space at $x \in \mathcal{M}$. On $T_x\mathcal{M}$, the inner product of $\mathcal{M}$ is denoted by $\langle \cdot, \cdot \rangle_x$ and induces the norm $\| \cdot \|_x$. For a smooth map $f\colon \mathcal{M} \to \mathbb{R}$, we can define gradient $\mathrm{grad} f$ as a unique vector field that satisfies $\forall v \in T_x\mathcal{M}\colon Df(x)[v] = \langle \mathrm{grad} f(x), v \rangle_x$. A Riemannian manifold is generally not equipped with vector addition, whereas in Euclidean space iterative methods are updated by addition. Iterative updates are instead performed using alternative operations—such as retraction.

Definition 1 (Retraction) *Let 0_x denote the zero element of $T_x\mathcal{M}$. A map $R\colon T\mathcal{M} \ni (x, v) \mapsto R_x(v) \in \mathcal{M}$ is called a retraction on $\mathcal{M}$ if it satisfies the two following conditions for all $x \in \mathcal{M}$. (I) $R_x(0_x) = x$; (II) with the canonical identification $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$, $DR_x(0_x) = \mathrm{id}_{T_x\mathcal{M}}$, where $\mathrm{id}_{T_x\mathcal{M}}\colon T_x\mathcal{M} \to T_x\mathcal{M}$ denotes the identity map.*

Iterative methods on Riemannian manifold are defined by $x_{t+1} = R_{x_t}(\eta_t d_t)$ generally. Exponential maps are often used as retractions. The following assumption plays a central role in Lemma 4.

Assumption 2 (Retraction Smoothness) *Let $f\colon \mathcal{M} \to \mathbb{R}$ be a smooth map. Then there exists $L_r > 0$, such that $^\forall x \in \mathcal{M}, ^\forall v \in T_x\mathcal{M}$, $f(R_x(v)) \leq f(x) + \langle \mathrm{grad} f(x), v \rangle_x + \frac{L_r}{2}\|v\|_x^2$.*

In the Euclidean space setting, L-smoothness implies a property similar to retraction smoothness. The property corresponding to L-smoothness in Euclidean space is defined for $f\colon \mathcal{M} \to \mathbb{R}$ as $^\exists L > 0, ^\forall x, y \in \mathcal{M}\colon \|\mathrm{grad} f(x) - \Gamma_y^x \mathrm{grad} f(y)\|_x \leq L_r d(x, y)$, where Γ is the parallel transport from $T_y\mathcal{M}$ to $T_x\mathcal{M}$, and $d(\cdot, \cdot)$ is the Riemannian distance. This condition is sufficient for Assumption 2 with $R \coloneqq \mathrm{Exp}$ (see Boumal (2023, Corollary 10.54)). This case is frequently used (e.g., Zhang and Sra (2016); Criscitiello and Boumal (2023); Kim and Yang (2022); Liu et al. (2017)). Other sufficient conditions for Assumption 2 were identified by Kasai et al. (2018, Lemma 3.5) and Sakai and Iiduka (2025, Proposition 3.2). Now, we consider the empirical risk minimization problem such that

$$\text{minimize } f(x) \coloneqq \frac{1}{N}\sum_{j=1}^{N} f_j(x) \quad \text{subject to } x \in \mathcal{M},$$

where each $f_j\colon \mathcal{M} \to \mathbb{R}$ is smooth and lower bounded. Therefore, f is also smooth and lower bounded. This assumption is often made for both Euclidean space and Riemannian space. The lower boundedness of f is essential for analyses using optimization theory because unbounded f may not have optimizers. We denote an optimal value of f by $f^\star$ and let N denote the size of the dataset. In many machine learning tasks, either the dimension of model parameters x or the size of dataset N is large. Hence, we use the following minibatch gradient to efficiently approximate the gradient of f: $\mathrm{grad} f_B(x) \coloneqq \frac{1}{b}\sum_{j=1}^{b} \mathrm{grad} f_{\xi_j}(x)$, where B represents a minibatch with size b, and $(\xi_j)_{j=1}^b$ is a sequence of $\{1, \cdots, N\}$-valued i.i.d. random variables distributed as Π. That is, the gradient of $f_B(x) = \frac{1}{b}\sum_{j=1}^{b} f_{\xi_j}(x)$ is used instead of $\mathrm{grad} f(x)$. The following assumption is reasonable given this situation.

Assumption 3 (Bounded Variance Estimator) *The stochastic gradient given by a distribution Π is an unbiased estimator of the full gradient and has bounded variance: (I) $^\forall x \in \mathcal{M}\colon \mathbb{E}_{\xi\sim\Pi}[\mathrm{grad} f_\xi(x)] = \mathrm{grad} f(x)$; (II) $^\exists \sigma > 0, ^\forall x \in \mathcal{M}\colon \mathbb{V}_{\xi\sim\Pi}(\mathrm{grad} f_\xi(x)) \leq \sigma^2$.*

For example, if f is lower bounded, satisfies Assumption 2, and Π is taken to be the uniform distribution over $\{1, \dots, N\}$, then Assumption 3 holds. RSGD is defined by $x_{t+1} = R_{x_t}(-\eta_t \mathrm{grad} f_{B_t}(x_t))$, and is a generalization of Euclidean SGD. Because $(\xi_{i,t})_{i,t}$ are i.i.d. samples, $\boldsymbol{\xi}_T \coloneqq (\xi_{1,T}, \cdots, \xi_{b_T,T})^\top$ is independent of $(x_t)_{t=0}^T$. Note that the BS $(b_t)_t$ may vary at each iteration. The detailed definitions of $\mathbb{E}_{\xi\sim\Pi}, \mathbb{V}_{\xi\sim\Pi}$, and $\mathbb{E}$ can be found in Appendix D.

3. Convergence Analysis

We begin by presenting a lemma that will be used in the convergence analysis of RSGD. Lemma 4 plays a central role is the following convergence analyses; its proof is provided in Appendix D.1.

Lemma 4 (Underlying Analysis) *Let $(x_t)_t$ be a sequence generated by RSGD and let $\eta_{\max} > 0$. Consider a positive-valued sequence $(\eta_t)_t$ such that $\eta_t \in [0, \eta_{\max}] \subset [0, \frac{2}{L_r})$. Then, under Assumptions 2 and 3, we obtain*

$$\min_{t\in\{0,\cdots,T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2] \leq \frac{2(f(x_0)-f^\star)}{2-L_r\eta_{\max}} \frac{1}{\sum_{t=0}^{T-1}\eta_t} + \frac{L_r\sigma^2}{2-L_r\eta_{\max}} \frac{\sum_{t=0}^{T-1}\eta_t^2 b_t^{-1}}{\sum_{t=0}^{T-1}\eta_t}.$$

3.1. Case (i): Constant BS; Constant or Decaying LR

In this case, we consider a BS $(b_t)_t$ and an LR $(\eta_t)_t$ such that $b_t = b$ and $\eta_{t+1} \leq \eta_t$. In particular, we present the following examples with constant or decaying LRs.

$$\text{Constant LR:} \quad \eta_t = \eta_{\max}, \tag{1}$$

$$\text{Diminishing LR:} \quad \eta_t = \frac{\eta_{\max}}{\sqrt{t+1}}, \tag{2}$$

$$\text{Cosine Annealing LR:} \quad \eta_t \coloneqq \eta_{\min} + \frac{\eta_{\max}-\eta_{\min}}{2}\left(1+\cos\frac{t}{T}\pi\right), \tag{3}$$

$$\text{Polynomial Decay LR:} \quad \eta_t \coloneqq \eta_{\min} + (\eta_{\max}-\eta_{\min})\left(1-\frac{t}{T}\right)^p, \tag{4}$$

where $\eta_{\max}$ and $\eta_{\min}$ are positive values satisfying $0 \leq \eta_{\min} \leq \eta_{\max} < \frac{2}{Lr}$. Note that $\eta_{\max}$ (resp. $\eta_{\min}$) becomes the maximum (resp. minimum) value of η_t; namely, $\eta_t \in [\eta_{\min}, \eta_{\max}] \subset [0, \frac{2}{L_r})$.

Theorem 5 *We consider LRs (1), (2), (3), and (4) and a constant BS $b_t = b > 0$ under the assumptions of Lemma 4. Then, we obtain*

$$\textit{Diminishing LR (2):} \quad \min_{t\in\{0,\cdots,T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2] \leq \frac{Q_1 + Q_2\sigma^2 b^{-1}\log T}{\sqrt{T}} = O\left(\frac{\log T}{\sqrt{T}}\right),$$

$$\textit{Otherwise (1), (3), (4):} \quad \min_{t\in\{0,\cdots,T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2] \leq \frac{\tilde{Q}_1}{T} + \frac{\tilde{Q}_2\sigma^2}{b} = O\left(\frac{1}{T}+\frac{1}{b}\right),$$

where $Q_1, Q_2, \tilde{Q}_1$, and $\tilde{Q}_2$ are constants that do not depend on T.

3.2. Case (ii): Increasing BS; Constant or Decaying LR

In this case, we consider a BS $(b_t)_t$ and an LR $(\eta_t)_t$ such that $b_t \leq b_{t+1}$ and $\eta_{t+1} \leq \eta_t$. We use the same examples as in **Case (i)** [(1), $\cdots$, (4)], again with constant or decaying LRs, and a BS that increases every $K \in \mathbb{N}$ steps, where $\mathbb{N}$ is the set of positive integers. We let T denote the total number of iterations and define $M := \lfloor \frac{T}{K} \rfloor$ to represent the number of times the BS is increased. The BS, which takes the form of $\gamma^m b_0$ or $(am + b_0)^p$ every K steps, is an example of an increasing BS. We can formalize the resulting BSs: for every $m \in \{0, \ldots, M-1\}, t \in S_m := [mK, (m+1)K) \cap \mathbb{N}_0$,

$$\text{Exponential Growth BS: } b_t := b_0 \gamma^{m \left\lceil \frac{t}{(m+1)K} \right\rceil}, \tag{5}$$

$$\text{Polynomial Growth BS: } b_t := \left(am \left\lceil \frac{t}{(m+1)K} \right\rceil + b_0 \right)^c, \tag{6}$$

where $\gamma, c > 1, a > 0$, and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

Theorem 6 *We consider BSs (5) and (6) together with LRs (1), (2), (3) and (4) under the assumptions stated in Lemma 4. Then, the following results hold for both constant and increasing BSs.*

$$\textit{Diminishing LR (2)}: \quad \min_{t \in \{0, \cdots, T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2] \leq \frac{Q_1 + Q_2 \sigma^2 b_0^{-1}}{\sqrt{T}} = O\left(\frac{1}{\sqrt{T}}\right),$$

$$\textit{Otherwise (1), (3), (4)}: \quad \min_{t \in \{0, \cdots, T-1\}} \mathbb{E}[\|\mathrm{grad} f(x_t)\|_{x_t}^2] \leq \frac{\tilde{Q}_1 + \tilde{Q}_2 \sigma^2 b_0^{-1}}{T} = O\left(\frac{1}{T}\right),$$

where $Q_1, Q_2, \tilde{Q}_1$, and $\tilde{Q}_2$ are constants that do not depend on T.

3.3. Case (iii): Increasing BS; Warm-up Decaying LR

In this case, we consider a BS $(b_t)_t$ and an LR $(\eta_t)_t$ such that $b_t \leq b_{t+1}$ and $\eta_t \leq \eta_{t+1}$ for $(t \leq T_w - 1)$ and $\eta_{t+1} \leq \eta_t$ for $(t \geq T_w)$. As examples of an increasing warm-up LR, we consider an exponential growth LR and a polynomial growth LR, both increasing every K' steps. We set K, as defined in **Case (ii)**, to be lK', where $l \in \mathbb{N}$ (thus $K > K'$). Namely, we consider a setting in which the BS is increased every l times the LR is increased. To formulate examples of an increasing LR, we define $M' := \lfloor \frac{T}{K'} \rfloor$ and formalize the LR: for every $m \in \{0, \ldots, M'-1\}, t \in S'_m := [mK', (m+1)K') \cap \mathbb{Z}$,

$$\text{Exponential Growth LR: } \eta_t := \eta_0 \delta^{m \left\lceil \frac{t}{(m+1)K'} \right\rceil}, \tag{7}$$

$$\text{Polynomial Growth LR: } \eta_t := \left(sm \left\lceil \frac{t}{(m+1)K'} \right\rceil + \eta_0 \right)^q, \tag{8}$$

where $s > 0$ and $q > 1$. Furthermore, we choose $\gamma, \delta > 1$, and $l \in \mathbb{N}$ such that $\delta^{2l} < \gamma$ holds. Additionally, we set $l_w \in \mathbb{N}$ such that $T \geq T_w := l_w K' \geq lK'$. The examples of an increasing BS used in this case are an exponential growth BS (5) and a polynomial growth BS (6). As examples of a warm-up LR, we use LRs that are increased using the exponential growth LR (7) and the polynomial growth LR (8) corresponding respectively to the exponential and polynomial growth BSs for the first T_w steps and then decreased using the constant LR (1), the diminishing LR (2), the cosine annealing LR (3), or the polynomial decay LR (4) for

the remaining $T - T_w$ steps. Note that $\eta_{\max} \coloneqq \eta_{T_w - 1}$. A more detailed version of Theorem 7 is provided in Appendix D.4.

Theorem 7 *We consider BSs (5) and (6) together with warm-up LRs (7) and (8) with decay parts given by (1), (2), (3), or (4) under the assumptions stated in Lemma 4. Then, the following results hold for both constant and increasing BSs: (I) Decay part [Diminishing (2)]:* $\|G_T\|^2 = O((\sqrt{T+1} - \sqrt{T_w+1})^{-1})$*; (II) Decay part [Otherwise (1), (3), (4)]:* $\|G_T\|^2 = O((T - T_w)^{-1})$.

3.4. Case (iv): Constant BS; Warm-up Decaying LR

We consider a BS $(b_t)_t$ and an LR $(\eta_t)_t$ such that $b_t = b$ and $\eta_t \leq \eta_{t+1}$ for $(t \leq T_w - 1)$ and $\eta_{t+1} \leq \eta_t$ for $(t \geq T_w)$. As examples of a warm-up LR, we use the exponential growth LR (7) and the polynomial growth LR (8) for the first T_w steps and then the constant LR (1), the diminishing LR (2), the cosine annealing LR (3), or the polynomial decay LR (4) for the remaining $T - T_w$ steps. The other conditions are the same as those in **Case (iii)**. A more detailed version of Theorem 8 is provided in Appendix D.5.

Theorem 8 *We consider a constant BS* $b_t = b > 0$ *and warm-up LRs (7) and (8) with decay parts given by (1), (2), (3), or (4) under the assumptions stated in Lemma 4. Then, we obtain the following results: (I) Decay part [Diminishing (2)]:* $\|G_T\|^2 = O(\log \frac{T}{T_w}(\sqrt{T+1} - \sqrt{T_w+1})^{-1})$*, (II) Decay part [Otherwise (1), (3), (4)]:* $\|G_T\|^2 = O((T - T_w)^{-1} + b^{-1})$.

4. Numerical Experiment

We experimentally evaluated the performance of RSGD for the two types of BSs and various types of LRs introduced in Section 3. The experiments were run on an iMac (Intel Core i5, 2017) running the macOS Ventura operating system (ver. 13.7.1). The algorithms were written in Python (3.12.7) using the NumPy (1.26.0) and Matplotlib (3.9.1) packages. The Python code is available at https://github.com/iiduka-researches/RSGD_acml2025.git. We set $p = 2.0$ in (4) and $\eta_{\min} \coloneqq 0$. In **Cases (i) and (ii)**, we used an initial LR $\eta_{\max}$ selected from $\{0.5, 0.1, 0.05, 0.01, 0.005\}$. In **Case (ii)**, we set $K = 1000, \gamma = 3.0$, and $a = c = 2.0$. All the plots of the objective function values presented in this section are provided in Appendices A.2 and E. Those for **Cases (iii) and (iv)** are provided in Appendix G.

4.1. Principal Component Analysis

We can formulate the PCA problem as an optimization problem on the Stiefel manifold (Kasai et al., 2019); for a given dataset $\{x_j\}_{j=1,\ldots,N} \subset \mathbb{R}^n$ and r $(\leq n)$, minimize$_{U \in \mathrm{St}(r,n)} f(U) \coloneqq \frac{1}{N}\sum_{j=1}^{N} \|x_j - UU^\top x_j\|^2$, where $\mathrm{St}(r,n) \coloneqq \{U \in \mathbb{R}^{r \times n} \mid U^\top U = I_n\}$. We set $r = 10$ and used the COIL100 (Nene et al., 1996) and MNIST (LeCun et al., 1998) datasets. The Columbia Object Image Library (COIL100) dataset contains 7200 color camera images of 100 objects (72 poses per object) taken from different angles. We resized the images to 32×32 pixels and transformed each one into a 1024 $(= 32^2)$ dimensional vector. Hence, we set $(N, n, r) = (7200, 1024, 10)$. The MNIST dataset contains $60,000$ 28×28-pixel grayscale images of handwritten digits 0 to 9. We transformed each image into a 784

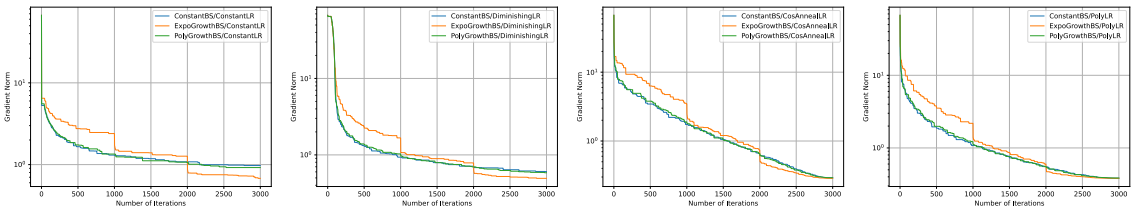

Figure 1: Norm of the gradient of the objective function versus the number of iterations for LRs (1), (2), (3), and (4) on COIL100 dataset (PCA).

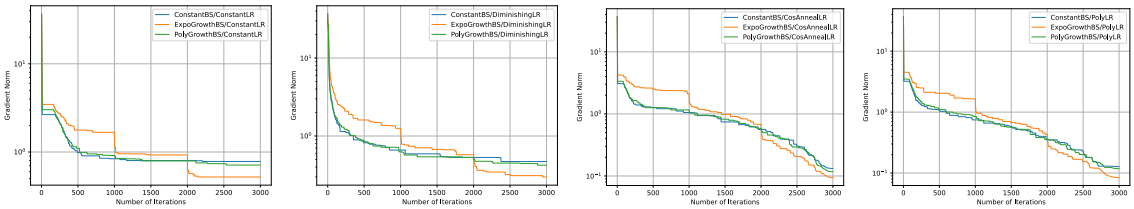

Figure 2: Norm of the gradient of the objective function versus the number of iterations for LRs (1), (2), (3), and (4) on MNIST dataset (PCA).

$(= 28^2)$ dimensional vector and normalized each pixel to the range $[0, 1]$. Hence, we set $(N, n, r) = (60000, 784, 10)$. Furthermore, we used a constant BS with $b_t := 2^{10}$, an exponential growth BS with an initial value $b_0 := 3^5$, and a polynomial growth BS with an initial value $b_0 := 30$.

4.2. Low-rank Matrix Completion

The LRMC problem involves completing an incomplete matrix $Z = (z_1, \ldots, z_N) \in \mathbb{R}^{n \times N}$; Ω denotes a set of indices for which we know the entries in Z. For $a \in \mathbb{R}^n$, we define $P_{\Omega_i}(a)$ such that the j-th element is a_j if $(i, j) \in \Omega$ and 0 otherwise. For $U \in \mathbb{R}^{n \times r}, z \in \mathbb{R}^n$, $q_j(U, z) := \mathrm{argmin}_{a \in \mathbb{R}^r} \|P_{\Omega_j}(Ua - z)\|$. We can now formulate the LRMC problem as the following optimization problem on the Grassmann manifold (Boumal and Absil, 2015):

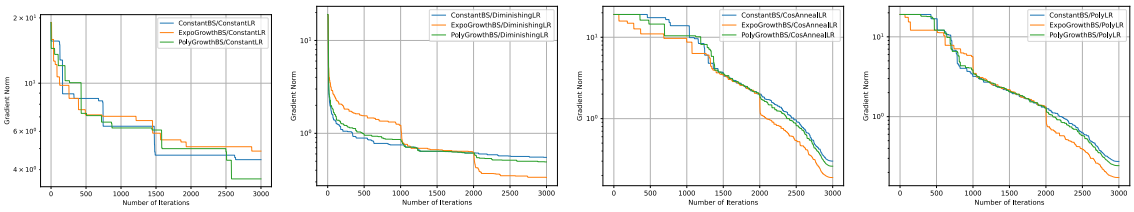

Figure 3: Norm of the gradient of the objective function versus the number of iterations for LRs (1), (2), (3), and (4) on MovieLens-1M dataset (LRMC).

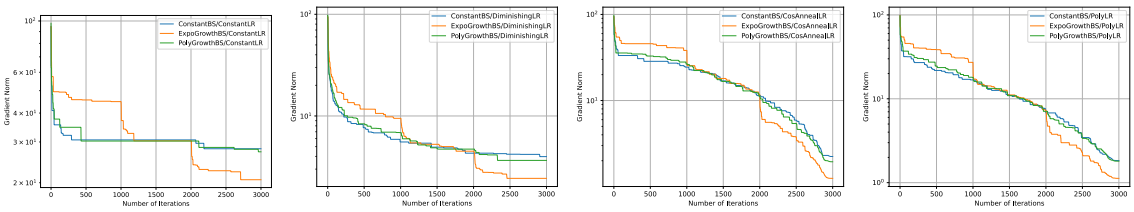

Figure 4: Norm of the gradient of the objective function versus the number of iterations for LRs (1), (2), (3), and (4) on Jester dataset (LRMC).

	COIL100 (PCA)	MNIST (PCA)	MovieLens-1M (LRMC)	Jester (LRMC)
Large constant BS	1411.37	2697.24	4482.41	5432.54
Full constant BS	1488.34	2719.66	12947.91	6816.28
Increasing BS	**468.53**	**745.13**	**1133.92**	**1695.85**

Table 2: Computational time [s] (CPU time) for large-constant, full-constant and increasing batch sizes.

$\text{minimize}_{U \in \mathrm{Gr}(r,n)} f(U) \coloneqq \frac{1}{N} \sum_{j=1}^{N} \|P_{\Omega_j}(U q_j(U, z_j) - z_j)\|^2$, where $\mathrm{Gr}(r,n) \coloneqq \mathrm{St}(r,n)/\mathrm{O}(r)$. We set $r = 10$ and used the MovieLens-1M (Harper and Konstan, 2016) and Jester datasets (Goldberg et al., 2001). The MovieLens-1M dataset contains $1,000,209$ ratings given by 6040 users on 3952 movies. Every rating lies in $[0, 5]$. We normalized each rating to the range $[0, 1]$. Hence, we set $(N, n, r) = (3952, 6040, 10)$. The Jester dataset contains ratings of 100 jokes from $24,983$ users. Every rating is bounded by the range $[-10, 10]$. Hence, we set $(N, n, r) = (24983, 100, 10)$. Furthermore, we used a constant BS $b_t \coloneqq 2^8$, an exponential growth BS with an initial value of $b_0 \coloneqq 3^4$, and a polynomial growth BS with an initial value of $b_0 \coloneqq 14$.

The performances in terms of the gradient norm of the objective function versus the number of iterations for LRs (1), (2), (3), and (4) on the COIL100, MNIST, MovieLens-1M, and Jester datasets are plotted in Figures 1, 2, 3, and 4, respectively. Achieving a small gradient norm was better with an increasing BS than with a constant BS. Among the increasing BSs, the exponential growth BS outperformed the polynomial BS. Because the magnitude of increase in the exponential BS is $O(\gamma^m)$ and that in the polynomial BS is $O(m)$, these numerical results indicate that a larger rate of increase in BS leads to better performance.

4.3. Comparison of Computational Time between Constant BS and Increasing BS versus SFO Complexity

What are the differences between using a large constant BS, a small constant BS, and an increasing BS? We numerically investigated this question on the basis of SFO complexity, defined as the number of stochastic gradient evaluations executed over T iterations (Agarwal and Bottou, 2015; Shallue et al., 2019; Sato and Iiduka, 2023). For a constant BS b, SFO complexity is represented by bT. For other BSs, it can be computed numerically, serves as

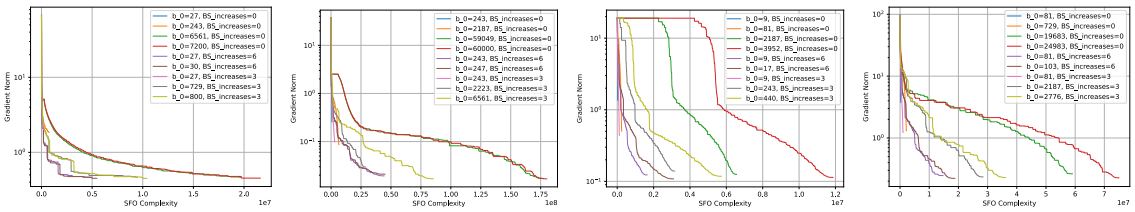

Figure 5: Norm of objective function gradient versus SFO complexity. Datasets used were COIL100 (PCA), MNIST (PCA), MovieLens-1M (LRMC), and Jester (LRMC) in order from left to right. A cosine annealing LR was used except for COIL100, for which a constant LR was used. For 'BS_increases= 0,' a constant BS $b = b_0$ was used. For 'BS_increases= 3' and 'BS_increases= 6,' the BS was increased 3 and 6 times, respectively, in accordance with the exponential growth BS.

a proxy for computational time. Figure 5 plots the gradient norm of the objective function versus SFO complexity for PCA and LRMC. Each curve corresponds to RSGD for 3000 steps under one of the following settings: constant BS, BS tripled every 1000 steps (three increases in total), or BS tripled every 500 steps (six increases in total). These settings follow the update formula for constant or exponential growth BS. Additional experimental details are provided in the caption of Figure 5. As shown in the figure, an increasing BS combines the advantages of both small and large BSs—namely shorter computational time and convergence to a solution with a smaller final gradient norm.

As shown in Figure 5, although both full and large constant BSs lead to optima achieving a small gradient norm, they require more SFO complexity. In contrast, while small constant BSs require less SFO complexity, they do not lead to optima achieving a small gradient norm. As shown in Table 2, the computational time (CPU time) of an increasing BS is shorter than that of both large and full constant BSs. Between small and large constant BSs, there is a trade-off between convergence to a solution with a smaller final gradient norm and a shorter computational time. Our results show that an increasing BS balances this trade-off effectively.

Why is increasing BS better? Our theoretical analyses provide an answer to this question. From Theorem 5, the SFO complexity of a constant BS with the critical BS for achieving $\|G_T\|^2 \leq \epsilon^2$ is $O(\epsilon^{-4})$. From Theorem 6, the SFO complexity of an exponential growth BS for achieving $\|G_T\|^2 \leq \epsilon^2$ is $O(\gamma^{\epsilon^{-2}})$, where $\gamma := 3$ in this experiment. Although $O(\gamma^{\epsilon^{-2}})$ is inferior to $O(\epsilon^{-4})$ in our strict theoretical setting—with fixed K (number of steps for each BS) and dynamic M (number of BS increases)—the assumptions in this experiment differ. Specifically, K depended on the value assigned to M, and our analysis can be applied to this experimental setting. Under these conditions, the SFO complexity of an exponential growth BS for achieving $\|G_T\|^2 \leq \epsilon^2$ is $O(\epsilon^{-2})$, which is superior to $O(\epsilon^{-4})$ and consistent with our experimental results. This provides our theoretical justification: increasing BS achieves lower SFO complexity than a constant BS. Summarizing the above, we obtain the following theorem. The derivation is provided in Appendix A.

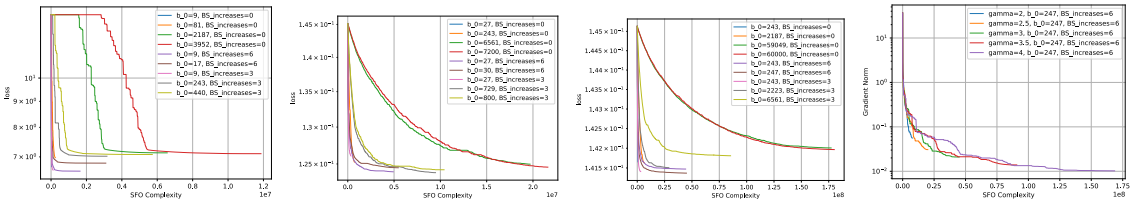

Figure 6: The plot on the far left depicts the objective function value (loss) for MNIST (PCA) under the same setting as in Figure 5. The plot in the middle left (resp. right) depicts the objective function value (loss) for our proposed optimization problem when $(N, n) = (7200, 1024)$ (resp. $(60000, 1024)$). The plot on the far right depicts the trade-off with respect to the setting of γ.

(b_0, M)	(27,0)	(243, 0)	(6561, 0)	(7200, 0)	(27, 3)	(729, 3)	(800, 3)	(27, 6)	(30, 6)
x (COIL100)	2.233	1.055	0.903	**0.902**	1.173	0.911	**0.902**	0.973	0.951
(b_0, M)	(243, 0)	(2187, 0)	(59049, 0)	(60000, 0)	(243, 3)	(2223, 3)	(6561, 3)	(243, 6)	(247, 6)
y (MNIST)	4.846	2.292	4.847	5.332	1.141	**-4.622**	2.134	4.247	-4.139

Table 3: Let U_{EVD} be the solution to PCA by eigenvalue decomposition, and let U_{RSGD} be the solution to PCA by RSGD. Then, the differences between the representations of the subspaces spanned by the principal components of U_{EVD} and U_{RSGD} are given by $l := \|U_{\mathrm{EVD}}U_{\mathrm{EVD}}^{\top} - U_{\mathrm{RSGD}}U_{\mathrm{RSGD}}^{\top}\|$. For comparison with the results for COIL100, we introduce x such that $l = 9.487 + x \times 10^{-3}$, and for MNIST, we introduce y such that $l = 0.4 + y \times 10^{-4}$. M is the number of BS increments.

Theorem 9 *Let $\epsilon > 0$, and consider an LR scheduler other than the diminishing LR one. Under the assumptions of Theorems 5 and 6, the SFO complexity for achieving $\|G_T\|^2 < \epsilon^2$ is $O(\epsilon^{-4})$ for a constant BS and $O(\epsilon^{-2})$ for an increasing BS.*

Remark 10 *Under the rigorous conditions of our theory, $M \to \infty$ implies that $T = MK + \mathrm{const.} \to \infty$ with fixed K. In contrast, the conditions of our experiment do not permit an increase in BS to achieve $T \to \infty$ because it would require $K \to \infty$ for T to diverge with fixed M, which means that the number of steps for the initial BS is infinite. However, as is done in many experimental setups, including ours, T is predetermined as a finite value before conducting the experiment. This setting is equivalent to using the number of steps as the stopping criterion. Under this practical setting, the conditions of our experiment are meaningful because $T < \infty$ implies $K < \infty$ with a fixed M. On the other hand, by considering the conditions of our proofs, our theoretical analysis can also be applied to calculate SFO complexities, even within our experimental setting. From these reasons, our theoretical analysis can support the numerical results. To enhance understanding, the derivation of the SFO complexity for an increasing BS can be found in Appendix A.*

Objective Function Values: Generalization performance is an important issue, and the objective function value serves as a criterion for measuring it. As shown in part of

Appendix A.2, although the exponential growth BS performed better than or equal to the constant BS with respect to the objective function values, the differences in the values were small. One possible reason for this is that the objective function may be flat around the optimal solution. To test this hypothesis, we conducted an experiment comparing the solution obtained by RSGD with that obtained by eigenvalue decomposition in PCA. When the data dimensionality is not large, PCA can obtain the exact solution by eigenvalue decomposition, and the difference from the approximate solution obtained by RSGD can be used as a measure of generalization. As shown in Table 3, the difference is small, which indicates that the objective functions of PCA with COIL100 and MNIST are flat around the optimal (exact) solution. Conversely, by considering optimization problems in which the objective functions do not have this property, the superiority of an increasing BS in terms of generalization performance can be revealed more clearly. In fact, this has already been demonstrated in LRMC with MovieLens-1M, as shown at the far left of Figure 6 (the corresponding results for the other datasets are provided in Appendix A.2). For further verification of this, we formulated a new optimization problem on the sphere S^{n-1}: $\mathrm{minimize}_{w\in\mathrm{S}^{n-1}} f(w) := \frac{1}{N}\sum_{j=1}^{N}\sqrt{|\langle x_j, w\rangle|}$, where $\{x_j\}_{j=1}^{N}$ is a dataset uniformly sampled from S^{n-1} (see Appendix C for more details). We set $(N, n) = (7200, 1024)$ and $(60000, 1024)$ and used the cosine annealing LR with an initial LR $\eta_{\max} = 0.01$. The BS was configured in exactly the same way as that for the COIL100 and MNIST datasets in Section 4.3. As in the two central graphs of Figure 6, an increasing BS achieved superior performance in terms of the objective function value for this proposed problem. Although the order of the differences (vertical axis) $O(10^{-2}) - O(10^{-1})$ may appear small, it is 10–100 times larger than that observed for PCA because the differences in the values shown in Table 3 are on the order of $O(10^{-4}) - O(10^{-3})$. These results further confirm the validity of our hypothesis for this problem. Given these results, we suggest that an increasing BS should also improve generalization performance. Turning to a different topic, the objective function of our proposed optimization problem has an unbounded gradient norm (see Appendix C), and the corresponding experiments numerically confirmed that our theoretical results are applicable even in this case. The graphs of the gradient norm for our proposed problem are provided in Appendix C.1.

Guidelines for setting γ, b_0 and M in the exponential growth BS : From Theorem 6, $\|G_T\|^2 = O(1+\frac{\gamma}{\gamma-1}) = O(1+b_0^{-1}) = O(M^{-1})$ and $\mathrm{SFO}_{\epsilon}^{\mathrm{incr}} = O(\gamma^M) = O((b_0^2-1)^{-1}b_0^3) = O(\gamma^M M^{-1})$ hold. These theoretical results suggest a trade-off: for smaller gradient norm, γ, b_0, and M should be assigned larger values while for smaller SFO complexity, γ, b_0, and M should be assigned smaller values. The far-right plot in Figure 6 illustrates the trade-off between the gradient norm and SFO complexity achieved experimentally on MNIST (PCA) using the cosine annealing LR with an initial LR $\eta_{\max} = 0.01$, where $(b_0, M) = (247, 6)$, while varying γ among $\{2.0, 2.5, 3.0, 3.5, 4.0\}$. For b_0, this trade-off can be observed by comparing the three entries under 'BS_increases= 3' or the two entries under 'BS_increases= 6' in Figure 5. For M, which represents the value 'BS_increases', this trade-off can be observed by comparing the purple entry with the pink entry in Figure 5. From these results, we were able to show, both theoretically and experimentally, that for each hyperparameter the trade-off between the gradient norm and SFO complexity holds. Figure 5 shows that an increasing BS with small b_0 and large M $(= 6)$ leads to multiple optima characterized by a small gradient norm and relatively low SFO complexity, indicating that well-balanced

configurations were achieved by taking into account the trade-offs inherent in b_0 and M individually. The relevant derivation is provided in Appendix B.

5. Conclusion

Our theoretical analysis, conducted with several learning rate schedules, including cosine annealing and polynomial decay, demonstrated that using an increasing batch size rather than a constant batch size improves the convergence rate of Riemannian stochastic gradient descent. This result is supported by our experimental results. Furthermore, an increasing batch size yields optima characterized by a smaller gradient norm within a shorter computational time owing to reductions in the stochastic first-order oracle complexity. These findings indicate that an increasing batch size combines the advantages of both small and large constant batch size. Due to the nature of the experimental tasks, we were unable to directly investigate the effect on generalization performance. However, evaluation via the objective function value suggests that an increasing batch size also enhances generalization performance. We believe our results, which clarify one aspect of the effectiveness of an increasing batch size for generalization performance, provide valuable insight into this topic.

Acknowledgments

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 24K14846 awarded to Hideaki Iiduka.

References

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, 2008.

Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 78–86, 2015.

Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

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

Nicolas Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebla and its Applications*, 475:200–239, 2015.

Arnaud Breloy, Sandeep Kumar, Ying Sun, and Daniel P. Palomar. Majorization-minimization on the Stiefel manifold with application to robust sparse PCA. *IEEE Transactions on Signal Processing*, 69:1507–1520, 2021.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

Christopher Criscitiello and Nicolas Boumal. Curvature and complexity: Better lower bounds for geodesically convex optimization. In *The Thirty Sixth Annual Conference on Learning Theory*, volume 195, pages 2969–3013, 2023.

Alain Durmus, Pablo Jiménez, Eric Moulines, and Salem Said. On Riemannian stochastic approximation schemes with fixed step-size. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 1018–1026, 2021.

Kenneth Y. Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. 2017. arXiv:1706.02677.

Andi Han and Junbin Gao. Improved variance reduction methods for Riemannian non-convex optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7610–7623, 2022.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Reshad Hosseini and Suvrit Sra. An alternative to EM for Gaussian mixture models: Batch and stochastic Riemannian optimization. *Mathematical Programming*, 181(1):187–223, 2020.

Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on Lie groups for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chunlin Ji, Yuhao Fu, and Ping He. Adaptive Riemannian stochastic gradient descent and reparameterization for Gaussian mixture model fitting. In *Proceedings of the 15th Asian Conference on Machine Learning*, volume 222, pages 534–549, 2024.

Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: A Riemannian manifold preconditioning approach. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48, pages 1012–1021, 2016.

Hiroyuki Kasai, Hiroyuki Sato, and Bamdev Mishra. Riemannian stochastic recursive gradient algorithm. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2516–2524, 2018.

Hiroyuki Kasai, Pratik Jawanpuria, and Bamdev Mishra. Riemannian adaptive stochastic gradient algorithms on matrix manifolds. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3262–3271, 2019.

Jungbin Kim and Insoon Yang. Nesterov acceleration for Riemannian optimization, 2022. arXiv:2202.02036.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Yann LeCun, Corinna Cortes, and C.J.C. Burges. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist*, 1998.

Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *8th International Conference on Learning Representations*, 2020a.

Tianyi Lin, Chenyou Fan, Nhat Ho, Marco Cuturi, and Michael Jordan. Projection robust Wasserstein distance and Riemannian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 9383–9397, 2020b.

Changshuo Liu and Nicolas Boumal. Simple algorithms for optimization on Riemannian manifolds with constraints. *Applied Mathematics & Optimization*, 82(3):949–981, 2020.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yuanyuan Liu, Fanhua Shang, James Cheng, Hong Cheng, and Licheng Jiao. Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations*, 2017.

Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia object image library (COIL-100). Technical report, Columbia University, 1996. CUCS-006-96.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Soumava Kumar Roy, Zakaria Mhammedi, and Mehrtash Harandi. Geometry aware constrained optimization techniques for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Hiroyuki Sakai and Hideaki Iiduka. Convergence of Riemannian stochastic gradient descent on hadamard manifold. *Pacific Journal of Optimization*, 20(4):743–767, 2024.

Hiroyuki Sakai and Hideaki Iiduka. A general framework of Riemannian adaptive optimization methods with a convergence analysis. *Transactions on Machine Learning Research*, 2025.

Naoki Sato and Hideaki Iiduka. Existence and estimation of critical batch size for training generative adversarial networks with two time-scale update rule. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 30080–30104, 2023.

Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20(112):1–49, 2019.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *6th International Conference on Learning Representations*, 2018.

Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I. Jordan. Averaging stochastic gradient descent on Riemannian manifolds. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 650–687, 2018.

Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X. Yu. Orthogonal convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Melanie Weber and Suvrit Sra. Riemannian optimization via Frank-Wolfe methods. *Mathematical Programming*, 199(1):525–556, 2023.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E. Dahl, Christopher J. Shallue, and Roger B. Grosse. Which algorithmic choices matter at which batch sizes? Insights from a noisy quadratic model. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *29th Annual Conference on Learning Theory*, volume 49, pages 1617–1638, 2016.

Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. Graph geometry interaction learning. *Advances in Neural Information Processing Systems*, 33:7548–7558, 2020.