Distributed Retraction-Free and Communication-Efficient Optimization on the Stiefel Manifold

Anonymous Authors[1]

Abstract

Optimization problems on the Stiefel manifold, ranging from principal component analysis to enhancing neural network robustness, are ubiquitous in machine learning. The Landing algorithm avoids computationally expensive retraction operations, making it highly competitive for large-scale problems. This paper extends this method to distributed settings, introducing **EF-Landing**, the first retraction-free and communication-efficient algorithm for distributed stochastic optimization on the Stiefel manifold. By incorporating communication compression and error feedback, EF-Landing ensures convergence and constraint feasibility while significantly reducing communication overhead. We provide sharp convergence guarantees, demonstrating that EF-Landing achieves the same asymptotic linear speedup convergence rate as existing methods without communication compression. Furthermore, our analysis is highly versatile, applying to both deterministic and stochastic settings and encompassing algorithms based on gradient descent or momentum gradient descent. We also generalize EF-Landing to operate on block-wise Stiefel manifolds, enabling greater flexibility for structured constraints. Extensive numerical experiments validate our theoretical results.

1. Introduction

The Stiefel Manifold, defined as the set of matrices with orthonormal columns, plays a crucial role in enhancing the dissimilarity between learned features in machine learning. Many classical problems inherently require orthogonal constraints, such as Principal Component Analysis (PCA) (Hotelling, 1933) and Canonical Correlation Analysis (CCA) (Hotelling, 1936). Furthermore, recent research has demonstrated that incorporating additional orthogonal constraints in deep learning problems often improves the robustness of neural networks (Arjovsky et al., 2016; Wang et al., 2020; Bansal et al., 2018). These insights underscore the importance of optimization on the Stiefel Manifold in machine learning applications. Traditional methods typically involve a single computing node. However, state-of-the-art performance in modern tasks is often achieved using extremely large training datasets, necessitating efficient distributed algorithms for stochastic optimization on the Stiefel Manifold across multiple computing nodes.

This paper considers the following optimization problem across N collaborative nodes:

$$\min_{X\in\mathbb{R}^{n\times p}} f(X) = \frac{1}{N}\sum_{i=1}^{N}\left[f_i(X) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i} F(X;\xi_i)\right], \quad (1a)$$

$$\text{s.t.} \quad X^\top X = I_p, \quad (1b)$$

where $n \geq p$, $f_i(X)$ represents the objective function for each node i, and the random variable ξ_i corresponds to the local data maintained by node i, following a local distribution $\mathcal{D}_i$. The constraint (1b) can also be expressed as $X \in \mathrm{St}(p,n)$ in which $\mathrm{St}(p,n) := \{X \in \mathbb{R}^{n\times p} | X^\top X = I_p\}$ is referred to as Stiefel Manifold. It is straightforward to verify that the Stiefel Manifold constraint (1b) is non-convex.

Numerous approaches can be directly extended to solve the distributed problem (1). One line of research employs Riemannian gradient methods to iteratively move towards the desired solution, incorporating retraction operations to ensure feasibility on the Stiefel manifold (Edelman et al., 1998; Absil et al., 2009; Absil & Malick, 2012). However, retraction operations are computationally expensive, often requiring matrix inversion, matrix exponential calculations, or QR decomposition. To address this bottleneck, another line of research introduced the retraction-free Landing method (Ablin & Peyré, 2022), which relies solely on matrix multiplication and is particularly efficient on GPUs. In the Landing algorithm, iterates do not remain on the Stiefel manifold but gradually "land" (i.e., converge) onto it. This retraction-free property makes the method highly practical for large-scale optimization problems. For these reasons, this paper focuses on developing and analyzing

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

distributed Landing algorithms to solve problem (1).

In distributed optimization on the Stiefel Manifold, each worker transmits gradient matrices to a central server to update the model parameters. Given the substantial size of these gradient matrices, communicating them at every iteration incurs significant overhead, which hinders algorithmic efficiency and scalability. To address this issue, this paper explores communication compression techniques (Alistarh et al., 2017; Richtarik et al., 2021; Stich et al., 2018; Huang et al., 2022) to reduce overhead. Instead of transmitting full gradient or model matrices, these strategies communicate compressed matrices with significantly smaller sizes at each iteration. While communication compression has demonstrated both theoretical guarantees and empirical success in unconstrained distributed optimization, *no existing algorithms*, to the best of our knowledge, have been developed for distributed stochastic optimization on Stiefel Manifolds. Several key questions arise when developing algorithms:

Q1. Which components of the algorithm can be compressed to ensure convergence? Specifically, should we compress the Euclidean gradient, Riemannian gradient, or constraint penalty gradient?

Q2. Is the error compensation strategy necessary for optimization on the Stiefel Manifold to maintain both optimality and feasibility?

This paper addresses these questions and introduces the first retraction-free and communication-efficient algorithm for distributed stochastic optimization on the Stiefel Manifold. Specifically, we make the following contributions:

- **EF-Landing algorithm.** We identify that compressing the Euclidean gradient of each $f_i(x)$ ensures both optimality and feasibility. Furthermore, we demonstrate that error feedback mechanisms are essential for convergence, even in single-node settings. Building on these insights, we propose the EF-Landing algorithm for distributed stochastic optimization problem (1) on the Stiefel Manifold with communication compression.
- **Sharp convergence guarantees.** We provide convergence guarantees and establish convergence rates for EF-Landing. Our algorithm achieves the same asymptotic linear speedup convergence rate as the vanilla Landing algorithm without any communication compression. Furthermore, our results are highly versatile. By selecting appropriate hyper-parameters, our analysis applies to both deterministic and stochastic settings, encompassing a wide range of algorithms based on gradient descent or momentum gradient descent. Notably, our established convergence rates also recover results for communication compression in unconstrained distributed optimization. Our analysis is built upon a novel merit function bound, which serves as a foundational tool for algorithms utilizing lossy gradient estimates and may be of independent interest.
- **Generalization to block-wise Stiefel manifolds.** We extend our framework to a generalized setting where matrix variables are partitioned into blocks with block-wise orthogonal constraints. This enables greater flexibility for large-scale optimization with structured constraints. We develop communication-efficient algorithms for this setting and provide convergence analysis, showing that the proposed methods retain the same theoretical guarantees as their non-block counterparts.

2. Related Work

Optimization on manifolds. Riemannian methods are classical approaches for solving optimization problems on manifolds. The foundational principles of Riemannian gradient descent with retraction were established in (Edelman et al., 1998; Absil et al., 2009; Absil & Malick, 2012). Building on gradient-based techniques, various algorithms and convergence results have been developed, including first-order methods (Zhang & Sra, 2016; Boumal et al., 2019), second-order methods (Absil et al., 2007; Qi et al., 2010), and accelerated methods (Liu et al., 2017; Ahn & Sra, 2020; Alimisis et al., 2021). Stochastic optimization on manifolds has also been explored (Bonnabel, 2013; Tripuraneni et al., 2018), with further refinements presented in (Zhang et al., 2016; Zhou et al., 2019). On the other hand, retraction-free approaches primarily include methods based on penalty functions and augmented Lagrangian formulations (Xiao et al., 2021; 2022; Gao et al., 2019), as well as techniques that reformulate manifold constraints (Lezcano-Casado & Martínez-Rubio, 2019; Liu et al., 2024). Additionally, the Landing method, designed specifically for the Stiefel manifold, has been proposed and analyzed in (Ablin & Peyré, 2022; Ablin et al., 2024; Vary et al., 2024).

Distributed optimization. Distributed optimization has been extensively studied, with Distributed Gradient Descent (DGD) (Tsitsiklis et al., 1986) serving as the representative method. To address the communication bottleneck of central parameter servers, three main improvement strategies have been explored: decentralized communication and its enhancements to mitigate data heterogeneity (Lopes & Sayed, 2008; Shi et al., 2015; Nedić et al., 2017); lazy communication (McMahan et al., 2017; Stich, 2019a); and communication compression (Alistarh et al., 2017; Richtarik et al., 2021). Research on distributed optimization on manifolds remains relatively limited (Chen et al., 2021; Wang & Liu, 2022; Sun et al., 2024; Hu & Deng, 2024; Qu et al., 2024; Zhao et al., 2025), including work such as (Sun et al., 2024), which incorporated the Landing algorithm with gradient tracking techniques, and (Hu & Deng, 2024), which applied communication compression.

Communication compression. Communication compression significantly reduces the amount of information exchanged during distributed optimization. These techniques mainly fall into two categories: sparsification methods, such as Rand-K and Top-K (Stich, 2019b; Wangni et al., 2018), and quantization methods, like Sign-SGD and TurnGrad (Alistarh et al., 2017; Wen et al., 2017). To further mitigate information distortion, error feedback techniques have been widely adopted (Richtarik et al., 2021; Seide et al., 2014; Stich & Karimireddy, 2021), with (Fatkhullin et al., 2023) showing that incorporating momentum enhances the effectiveness of error feedback. Optimal compelxity with communicaiton compression has been examined in (Huang et al., 2022; He et al., 2024).

3. Preliminary

Notations. Given a matrix $X \in \mathbb{R}^{m\times n}$, let $\|X\|_F$ denote the Frobenius norm, with the corresponding inner product defined as $\langle X, Y\rangle := \mathrm{Tr}(X^\top Y)$. For a square matrix $M \in \mathbb{R}^{n\times n}$, the symmetric and skew-symmetric components are given by $\mathrm{sym}(M) := \frac{1}{2}(M + M^\top)$ and $\mathrm{skew}(M) := \frac{1}{2}(M - M^\top)$, respectively.

3.1. Landing Method

Descent direction. The Landing method (Ablin & Peyré, 2022; Ablin et al., 2024) utilizes both the Riemannian gradient and a penalty term to enforce constraints while eliminating the need for retraction. The descent direction is

$$\begin{aligned}\Lambda(X) =& \,\mathrm{grad}f(X) + \lambda\nabla\mathcal{N}(X), \qquad (2)\\ \text{where}\quad & \mathrm{grad}f(X) := \mathrm{skew}\Big(\nabla f(X)X^\top\Big)X,\\ & \nabla\mathcal{N}(X) := X\Big(X^\top X - I_p\Big).\end{aligned}$$

The first term, $\mathrm{grad}f(X)$, represents the Riemannian gradient on the Stiefel manifold $\mathrm{St}(p,n)$ when X satisfies the manifold constraints. The second term, $\nabla\mathcal{N}(X)$, denotes the gradient of the penalty term $\mathcal{N}(X) = \frac{1}{4}\|X^\top X - I_p\|^2$, where the penalty parameter $\lambda > 0$ controls its weight.

Landing. With descent direction (2), Landing method is:

$$X^{k+1} = X^k - \gamma\Lambda(X^k), \qquad (3)$$

where γ is the step size. Rather than enforcing an exact constraint at each step, the Landing method only requires the iterations to remain within the neighborhood of $\mathrm{St}(p,n)$, i.e., the safe region defined as follows (Ablin et al., 2024):

Definition 3.1 (Safe Region)**.** Given some $\epsilon \in (0, 3/4)$, we define the safe region of a Stiefel manifold as

$$\mathrm{St}(p,n)^\epsilon := \Big\{X \in \mathbb{R}^{n\times p}\Big|\|X^\top X - I_p\| \le \epsilon\Big\}.$$

The intuition behind the Landing algorithm is that $\mathrm{grad}f(X)$ drives the iteration towards the optimal solution on the Stiefel manifold, while $\nabla\mathcal{N}(X)$ steers the iteration towards the Stiefel manifold constraint. The orthogonality between these two terms ensures both optimality and feasibility as long as the algorithm converges.

3.2. Compressor and Error Feedback

Contractive compressor. This paper examines communication compression using contractive compressors, with Top-K and Rand-K being commonly used examples (Wangni et al., 2018; Stich, 2019b).

Assumption 3.2 (Contractive Compressor)**.** A compressor $\mathcal{C}$ is defined as a contractive compressor if it satisfies

$$\mathbb{E}_{\mathcal{C}}\Big[\|\mathcal{C}(X) - X\|_F^2\Big] \le (1-\alpha)\|X\|_F^2, \quad \forall X \in \mathbb{R}^{n\times p}$$

where $\alpha \in (0, 1]$ is the contractive factor. The expectation is taken over the randomness of the compression operator $\mathcal{C}$.

Error feedback. Consider the unconstrained problem:

$$\min_{X\in\mathbb{R}^{n\times p}} \quad f(X). \qquad (4)$$

The error feedback method (Richtarik et al., 2021) to solve the above unconstrained optimization problem is

$$X^{k+1} = X^k - \gamma Y^k, \qquad (5a)$$
$$Y^{k+1} = Y^k + \mathcal{C}(\nabla f(X^{k+1}) - Y^k), \qquad (5b)$$

where Y^k is the compressed approximation of the gradient $\nabla f(X^k)$, and $\mathcal{C}(\cdot)$ satisfies Assumption 3.2. The intuition behind error feedback is straightforward:

$$\begin{aligned}&\mathbb{E}_{\mathcal{C}}\|Y^{k+1} - \nabla f(X^{k+1})\|^2\\ \overset{(5b)}{=}\,&\mathbb{E}_{\mathcal{C}}\|Y^k - \nabla f(X^{k+1}) - \mathcal{C}(Y^k - \nabla f(X^{k+1}))\|^2\\ \le\,& (1-\alpha)\|Y^k - \nabla f(X^{k+1})\|^2.\end{aligned}$$

Suppose $\nabla f(X^k) \to \nabla f(X^\star)$ as $k \to \infty$, where $X^\star$ is a stationary solution to problem (4). The above inequality implies that Y^k converges to $\nabla f(X^\star)$. Combined with (5a), this guarantees that $X^k \to X^\star$ under compression, thereby validating the effectiveness of the error feedback method.

4. EF-Landing Algorithm

In this section, we introduce **EF-Landing**, the first distributed Landing method incorporating Error Feedback.

4.1. Compression on Euclidean Gradient

In the error feedback method for solving the unconstrained optimization problem (4), compressing the gradient $\nabla f(X)$ is sufficient to ensure convergence. Given that the Landing method can be expressed in a descent form (3), a natural question arises: should we mimic recursion (5) and compress $\Lambda(X)$ for EF-Landing?

We find that compressing $\Lambda(X)$ does not necessarily guarantee convergence. Note that the key factor ensuring the effectiveness of the Landing method (2)–(3) is the **orthogonality** property between the Riemannian gradient $\mathrm{grad}f(X)$ and the penalty gradient $\nabla\mathcal{N}(X)$, given by

$$\langle \mathrm{grad}f(X), \lambda\nabla\mathcal{N}(X)\rangle = 0, \quad \forall X \in \mathbb{R}^{n\times p}. \tag{6}$$

This orthogonality ensures that if $\|\Lambda(X)\|_F^2 \leq \delta$ for any $\delta > 0$, then both $\|\mathrm{grad}f(X)\|_F^2 \leq \delta$ and $\|\lambda\nabla\mathcal{N}(X)\|_F^2 \leq \delta$ hold, guaranteeing optimality while preserving feasibility on the Stiefel manifold. However, it remains unclear whether orthogonality (6) is preserved after compressing $\Lambda(X)$.

To address this issue, we propose to compress the Euclidean gradient $\nabla f(X)$ in EF-Landing. Let $\boldsymbol{g}$ be the compressed approximation of the Euclidean gradient $\nabla f(X)$, we let

$$\tilde{\Lambda}(X;\boldsymbol{g}) = \mathrm{grad}(\boldsymbol{g}) + \lambda\nabla\mathcal{N}(X), \tag{7}$$
$$\text{where} \quad \mathrm{grad}(\boldsymbol{g}) := \mathrm{skew}\Big(\boldsymbol{g}X^\top\Big)X,$$
$$\nabla\mathcal{N}(X) := X\Big(X^\top X - I_p\Big).$$

The proposition below ensures the orthogonality between $\mathrm{grad}(\boldsymbol{g})$ and $\nabla\mathcal{N}(X)$ to hold after compressing $\nabla f(X)$:

Proposition 4.1. *For any gradient estimate $\boldsymbol{g} \in \mathbb{R}^{n\times p}$, the orthogonality between $\mathrm{grad}(\boldsymbol{g})$ and $\nabla\mathcal{N}(X)$ preserves:*

$$\langle \mathrm{skew}(\boldsymbol{g}X^\top)X, \lambda X(X^\top X - I_p)\rangle = 0.$$

4.2. Error Feedback

Vanilla compression may lead to non-convergence. Various approaches exist for compressing the Euclidean gradient $\nabla f(X)$. The most straightforward approach is to directly compress $\nabla f(X)$, i.e., $\boldsymbol{g} = \mathcal{C}(\nabla f(X))$, where $\mathcal{C}(\cdot)$ is the contractive compressor satisfying Assumption 3.2. However, we observe that such vanilla compression may lead to non-convergence, even in the case of the single-node Landing (2)–(3). The main intuition is that the tangent space of the Stiefel manifold and the compressed gradient may become orthogonal, causing the iteration to stagnate.

Proposition 4.2. *There exists an L-smooth objective function $f : \mathbb{R}^{n\times p} \to \mathbb{R}$, a contractive compressor satisfying Assumption 3.2, and an initial point $X^0 \in \mathbb{R}^{n\times p}$ such that the following update scheme:*

$$\boldsymbol{g} = \mathcal{C}(\nabla f(X)),$$
$$\tilde{\Lambda}(X;\boldsymbol{g}) = \mathrm{grad}(\boldsymbol{g}) + \lambda\nabla\mathcal{N}(X),$$
$$X \leftarrow X - \gamma\tilde{\Lambda}(X;\boldsymbol{g})$$

results in the algorithm getting stagnant and failing to converge to the stationary solution. (See Proof in Appendix B.)

Error feedback corrects non-convergence. Error feedback can correct the non-convergence encountered with vanilla compression. Inspired by momentum error feedback (Fatkhullin et al., 2023), we employ the following updates within each node i:

$$\boldsymbol{v}_i^{k+1} = (1-\eta)\boldsymbol{v}_i^k + \eta\nabla F(X^{k+1};\xi_i^{k+1}), \tag{8a}$$
$$\boldsymbol{c}_i^k = \mathcal{C}(\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k), \tag{8b}$$
$$\boldsymbol{g}_i^{k+1} = \boldsymbol{g}_i^k + \boldsymbol{c}_i^k, \tag{8c}$$

where η is the momentum rate. Each node sends $\boldsymbol{c}_i^k$ to the master, and the master updates the Euclidean gradient:

$$\boldsymbol{g}^{k+1} = \boldsymbol{g}^k + \frac{1}{N}\sum_{i=1}^N \boldsymbol{c}_i^k. \tag{9}$$

The compressed gradient $\boldsymbol{g}^k$ provides a better approximation than the vanilla compressed gradient $\mathcal{C}(\nabla F(X^{k+1};\xi_i^{k+1}))$. Once $\boldsymbol{g}^k$ is obtained through the error feedback process (8)–(9), the master node will compute the descent direction $\tilde{\Lambda}(X;\boldsymbol{g})$ as outlined in (7).

4.3. EF-Landing Algorithm

EF-Landing consists of two main components: Euclidean gradient compression and error feedback, as discussed in the previous subsections. Furthermore, before the master node computes the Landing descent direction (7), we apply a clipping operation to $\boldsymbol{g}^k$ to prevent the integrated gradient estimate from pushing the iterations out of the safe region $\mathrm{St}(p,n)^\epsilon$ (see Definition 3.1). Given a gradient bound L' (specified later in Section 5.1), we clip $\boldsymbol{g}^k$ as follows:

$$\tilde{\boldsymbol{g}}^k = \min\left\{1, \frac{L'}{\|\boldsymbol{g}^k\|_F}\right\}\boldsymbol{g}^k. \tag{10}$$

Once clipped, $\boldsymbol{g}^k$ is ready for the computation of the Landing descent direction (7) and the iteration (3). The complete EF-Landing algorithm is presented in Algorithm 1.

The EF-Landing algorithm is highly versatile. When the gradient oracle $\nabla F(X;\xi_i) \equiv \nabla f_i(X)$, the problem reduces to deterministic optimization. The algorithm also applies when the compressor $\mathcal{C}(\cdot)$ is the identity mapping, i.e., distributed optimization without communication compression. Additionally, it holds when the momentum rate $\eta = 1$, corresponding to iterations without momentum. Notably, when all three of these conditions are satisfied simultaneously, EF-Landing reduces to the basic Landing algorithm, with exact convergence results as established in (Ablin et al., 2024).

5. Convergence Analysis

This section begins by presenting the necessary conditions, and then proceeds to establish the convergence guarantees and rates for EF-Landing.

Algorithm 1 EF-Landing

```
Require: starting point $X^0 \in \mathbb{R}^{n\times p}$; gradient bound $L'$;
    step size $\gamma > 0$; compressor $\mathcal{C}$; momentum $\eta \in (0, 1]$;
Each node initializes $\boldsymbol{v}_i^0 = \nabla F(X^0; \xi_i^0)$, $\boldsymbol{g}_i^0 = \mathcal{C}(\boldsymbol{v}_i^0)$
    for $i = 1, \dots, N$; Master initializes $\boldsymbol{g}^0 = \frac{1}{N}\sum_{i=1}^N \boldsymbol{g}_i^0$.
for $k = 0, 1, \dots, K-1$ do
    Master clips the gradient via (10);
    Master computes $\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)$ using (7);
    Master computes $X^{k+1} = X^k - \gamma\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)$ and
    broadcasts $X^{k+1}$ to all nodes.
    for all nodes $i = 1, \dots, N$ in parallel do
        Compute momentum $\boldsymbol{v}_i^{k+1}$ via (8a);
        Compress $\boldsymbol{c}_i^k$ via (8b) and send it to the master;
        Update local state $\boldsymbol{g}_i^{k+1}$ via (8c).
    end for
    Master updates $\boldsymbol{g}^{k+1}$ via (9).
end for
```

5.1. Assumptions

The following assumptions are standard for optimization on the Stiefel manifold.

Assumption 5.1 (Global Gradient Bound)**.** There exists constant L' such that the following inequality holds

$$\|\nabla F(X;\xi_i)\|_F \le L', \quad \forall X \in \mathrm{St}(p,n)^\epsilon, \forall \xi \sim \mathcal{D}_i$$

Remark. The assumption of a gradient bound L' is mild, as the search domain $\mathrm{St}(p,n)^\epsilon$ is bounded, and we assume that no highly anomalous data will cause gradient explosion.

Assumption 5.2 (Lipschitz Smoothness)**.** For each node i, $f_i(X) : \mathbb{R}^{n\times p} \to \mathbb{R}$ is differentiable and there exists a smooth constant L_i such that for any $X, Y \in \mathrm{St}(p,n)^\epsilon$,

$$\|\nabla f_i(X) - \nabla f_i(Y)\|_F \le L_i\|X - Y\|_F.$$

Remark. It is straightforward to show that the smoothness constant L of the global function $f(X)$ satisfies $L \le \max_{i=1,\dots,N}\{L_i\}$. Additionally, we define $\tilde{L} := \sqrt{\frac{1}{N}\sum_{i=1}^N L_i^2}$ as the averaged smoothness constant.

Assumption 5.3 (Lower Bounded Merit Function)**.** The objective function $f(X)$ is lower bounded by $f^* := \inf_{X\in\mathrm{St}(p,n)^\epsilon} f(X) > -\infty$.

Assumption 5.4 (Unbiasedness and Bounded Variance)**.** There exists a $\sigma \ge 0$ such that for each node i, for any $X \in \mathrm{St}(p,n)^\epsilon$,

$$\begin{aligned}\mathbb{E}_{\xi_i\sim\mathcal{D}_i}[\nabla F(X;\xi_i) - \nabla f_i(X)] &= 0,\\ \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left[\|\nabla F(X;\xi_i) - \nabla f_i(X)\|_F^2\right] &\le \sigma^2.\end{aligned}$$

5.2. Supporting Lemmas

This section extends the results for the single-node Landing to support the convergence analysis of EF-Landing.

Safe step size. The main objective of the Landing algorithm is to select a sufficiently small step size that ensures the iterations remain within $\mathrm{St}(p,n)^\epsilon$. In our analysis, we require only two assumptions—Assumptions 5.1 and 5.2—to determine a uniform, non-vanishing step size that is valid throughout the entire iteration process.

Lemma 5.5 (Uniform Safe Step Size)**.** *Consider the update* $\tilde{X} = X - \gamma\tilde{\Lambda}(X;\boldsymbol{g})$*, where* $\tilde{\Lambda}(X;\boldsymbol{g})$ *is defined in* (7) *and* γ *is a step size. For any* $X \in \mathrm{St}(p,n)^\epsilon$, $\boldsymbol{g} \in \mathbb{R}^{n\times p}$*, if* $\|\boldsymbol{g}\|_F \le L'$ *and we choose*

$$\gamma \le \gamma_s := \min\left\{\frac{\lambda(1-\epsilon)\epsilon}{(1+\epsilon)^2(L')^2 + \lambda^2(1+\epsilon)\epsilon^2},\right.$$
$$\left.\sqrt{\frac{\epsilon}{2(1+\epsilon)^2(L')^2}}, \frac{1}{2\lambda}\right\},$$

then the next iteration $\tilde{X}$ *remains in* $\mathrm{St}(p,n)^\epsilon$.

Merit function. We introduce a merit function to facilitate convergence analysis of EF-Landing:

$$m(X) = f(X) - h(X) + \mu\mathcal{N}(X), \tag{11}$$
$$\text{where } \; h(x) = \frac{1}{2}\langle \mathrm{sym}(X^\top\nabla f(X)), X^\top X - I_p\rangle.$$

The constant μ is a hyperparameter, and its value will be specified in Lemma 5.6. The merit function $m(X)$ is smooth with a smoothness factor L_m when restricted to the bounded safe region $\mathrm{St}(p,n)^\epsilon$. An upper bound for L_m is provided by (Ablin et al., 2024). Furthermore, under Assumption 5.3, since both $h(X)$ and $\mathcal{N}(X)$ are bounded when $X \in \mathrm{St}(p,n)^\epsilon$, the merit function $m(X)$ is also lower bounded, with an infimum denoted as m^*. In this work, we establish a lower bound for $\langle\nabla m(X), \tilde{\Lambda}(X;\boldsymbol{g})\rangle$, which plays a critical role in analyzing the influence of compression error, as it explicitly separates the difference term $\|\boldsymbol{g} - \nabla f(X)\|_F^2$.

Lemma 5.6 (Merit Function Bound)**.** *For all* $X \in \mathrm{St}(p,n)^\epsilon$ *and* $\boldsymbol{g} \in \mathbb{R}^{n\times p}$*, the inner product between the descent direction* (7) *and the gradient of merit function* (11) *can be lower bounded as*

$$\begin{aligned}\langle\tilde{\Lambda}(X;\boldsymbol{g}),\nabla m(X)\rangle \ge\; & \frac{1}{4}\|\mathrm{grad} f(X)\|_F^2 + \lambda\mu\mathcal{N}(X)\\ & + \frac{1}{4}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2 - \|\boldsymbol{g} - \nabla f(X)\|_F^2,\end{aligned}$$

for the choice of $\mu \ge \frac{2}{3-4\epsilon}\big(L(1-\epsilon) + 3\sqrt{1+\epsilon}L' + 2\hat{L}^2\frac{1+\epsilon}{\lambda}\big)$*, where* L *and* L' *are defined in Assumptions 5.2 and 5.1,* $\hat{L} = \max\{L, L'\}$ *and* $\epsilon < 3/4$.

5.3. Main Convergence Theorem

Given an arbitrary momentum coefficient $\eta > 0$, we define a Lyapunov function as follows:

$$\mathcal{L}^k := m(X^k) - m^\star + \frac{c_1\gamma}{\theta}\tilde{G}^k + \frac{2c_1\gamma\eta\beta}{\theta}\tilde{P}^k + \frac{c_2\gamma}{\eta}P^k \tag{12}$$

$$\text{where} \quad \tilde{G}^k := \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2,$$

$$\tilde{P}^k := \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{v}_i^k - \nabla f_i(X^k)\|_F^2,$$

$$P^k := \|\boldsymbol{v}^k - \nabla f(X^k)\|_F^2,$$

where $\boldsymbol{v}^k := \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}_i^k$, and $m^\star$ denotes the infimum of the merit function in (11). Constants β, c_1, c_2 will be determined later. Using this Lyapunov function, we derive the main convergence theorem as follows.

Theorem 5.7 (Main Convergence Theorem)**.** *Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold, for an arbitrary momentum coefficient $\eta \in (0, 1]$, by running Algorithm 1 for K iterations, with proper learning rate γ (see Appendix E), we have*

$$\frac{1}{4K}\sum_{k=0}^{K-1}\mathbb{E}[\|\mathrm{grad}f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=0}^{K-1}\mathbb{E}[\mathcal{N}(X^k)]$$
$$\le \frac{\mathcal{L}^0}{\gamma K} + \frac{c_1\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{2c_1\eta^3\beta\sigma^2}{\theta} + \frac{c_2\eta\sigma^2}{N},$$

In the inequality, $\mathcal{L}^k$ is defined in (12), θ and β are two scalars defined as $\theta := 1 - \sqrt{1-\alpha}$ and $\beta := \frac{1-\alpha}{1-\sqrt{1-\alpha}}$. c_1, c_2 are two constants differently specified in three cases:

- *When the momentum coefficient $\eta = 1$ and the variance of gradients $\sigma^2 = 0$, the momentum and stochastic error is 0, and c_1, c_2 are specified as $c_1 = 1, c_2 = 0$.*
- *When the contractive factor $\alpha = 1$, the compression error is 0, and c_1, c_2 are specified as $c_1 = 0, c_2 = 1$.*
- *Otherwise, we specify $c_1 = c_2 = 2$.*

Remark. The convergence theorem above is highly versatile. By appropriately selecting hyperparameters such as the momentum coefficient η, gradient variance σ, and compressor factor α, our analysis applies to both deterministic and stochastic settings, covering algorithms based on gradient descent and momentum gradient descent, with or without communication compression.

5.4. Convergence Rate in Deterministic Scenario

In this subsection, we set $\sigma^2 = 0$ in Assumption 5.4. Since momentum provides no significant advantage in deterministic settings, we further set $\eta = 1$ to completely eliminate both momentum and stochastic error. Under these conditions, the constants in Theorem 5.7 simplify to $c_1 = 1$ and $c_2 = 0$, allowing the theorem to naturally recover various convergence results in deterministic scenarios.

Theorem 5.8 (EF-Landing in Deterministic Scenarios)**.** *Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold, if $\sigma^2 = 0$, and $\eta = 1$, by running Algorithm 1 for K iterations, with proper constant learning rate γ (see Appendix F), we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\mathrm{grad}f(X^k)\|_F^2] \le \frac{4(m(X^0) - m^\star)}{K\gamma} + \frac{4\mathbb{E}[\tilde{G}^0]}{K\theta},$$

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \le \frac{2(m(X^0) - m^\star)}{K\gamma\lambda\mu} + \frac{2\mathbb{E}[\tilde{G}^0]}{K\theta\lambda\mu},$$

where θ and β are defined in Theorem 5.7.

By additionally setting $\alpha = 1$ and $\tilde{G}^0 = 0$, meaning no compression is applied to the Landing method, the theorem simplifies to the standard Landing convergence theorem:

Corollary 5.9 (Vanilla Landing in Deterministic Scenarios)**.** *Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold, if $\sigma = 0$, $\eta = 1$ and $\alpha = 1$, by running Algorithm 1 for K iterations, with proper constant learning rate γ (see Appendix F), we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\mathrm{grad}f(X^k)\|_F^2 \le \frac{4(m(X^0) - m^*)}{K\gamma},$$

$$\frac{1}{K}\sum_{k=1}^{K-1}\mathcal{N}(X^k) \le \frac{2(m(X^0) - m^*)}{K\gamma\lambda\mu}.$$

Remark. Our established rate for vanilla Landing without compression is $\mathcal{O}(1/K)$, matching the state-of-the-art result in (Ablin et al., 2024; Vary et al., 2024), confirming the sharpness of our analysis. Moreover, by comparing Theorem 5.8 and Corollary 5.9, we observe that Landing, with or without communication compression, achieves the same $\mathcal{O}(1/K)$ rate, indicating that compression does not degrade the convergence order. However, more aggressive compression (i.e., $\theta \to 0$) leads to slower convergence for EF-Landing.

5.5. Convergence Rate in Stochastic Scenario

In this section, we consider the case where the variance $\sigma^2 > 0$ (Assumption 5.4). Notably, our assumption differs from that of (Ablin et al., 2024), which directly assumes a bounded variance for $\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)$. The momentum technique plays a crucial role in stochastic settings.

We consider the case with communication compression, i.e., $\alpha < 1$, and analyze the most general version of the EF-Landing algorithm. In this setting, the constants in Theorem

5.7 are specified as $c_1 = c_2 = 2$. The following theorem presents the convergence result of EF-Landing in stochastic scenarios, demonstrating a linear speedup.

Theorem 5.10 (EF-Landing in Stochastic Scenarios)**.** *Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold, if we choose step size γ as in Lemma G.1, by running Algorithm 1 for K iterations, with proper learning rate γ and momentum coefficient η (see Appendix H), we have*

$$\frac{1}{K}\sum_{k=1}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] \leq 4\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big),$$

$$\frac{1}{K}\sum_{k=1}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \leq \frac{2}{\lambda\mu}\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big),$$

where the $\mathcal{O}(\frac{1}{\sqrt{NK}})$ term is specified as

$$\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big) = \frac{C_1^\gamma \mathcal{L}^0}{K} + 4\Big(\frac{2\sigma^2 C_2^\gamma \mathcal{L}^0}{NK}\Big)^{\frac{1}{2}}$$
$$+4\Big(\frac{2\sigma^2(1-\alpha)}{\theta}\Big)^{\frac{1}{3}}\Big(\frac{C_2^\gamma \mathcal{L}^0}{K}\Big)^{\frac{2}{3}} + 4\Big(\frac{4\beta\sigma^2}{\theta}\Big)^{\frac{1}{4}}\Big(\frac{C_2^\gamma \mathcal{L}^0}{K}\Big)^{\frac{3}{4}},$$

The constant $\mathcal{L}^0$ is defined in (12), while θ and β are specified in Theorem 5.7. The constants C_1^γ and C_2^γ are independent of N and K, with their definitions provided in Appendix H.

Remark. This theorem establishes that EF-Landing achieves an asymptotic linear speedup convergence rate. The impact of the communication compressor appears only in higher-order terms and does not affect the dominant linear speedup rate. Notably, this rate aligns with the momentum error feedback method (Fatkhullin et al., 2023), which is designed for unconstrained minimization problems. This suggests that, with careful algorithmic design, the Stiefel manifold constraint does not pose fundamental challenges to communication compression.

To illustrate the versatility of our convergence analysis, we now examine the stochastic Landing algorithm without compression. In this case, the constants in Theorem 5.7 are specified as $c_1 = 0$ and $c_2 = 1$, leading to the result:

Theorem 5.11 (Stochastic Landing Convergence)**.** *Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold, if $\alpha = 1$, with proper learning rate γ and momentum coefficient η (see Appendix I), by running Algorithm 1 for K iterations, we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] \leq \frac{4C_1^\gamma \mathcal{L}^0}{K} + 4\sigma\sqrt{\frac{C_2 \mathcal{L}^0}{NK}},$$

$$\frac{1}{K}\sum_{k=1}^{K-1}\mathcal{N}(X^k) \leq \frac{2C_1^\gamma \mathcal{L}^0}{K\lambda\mu} + \frac{2\sigma}{\lambda\mu}\sqrt{\frac{C_2^\gamma \mathcal{L}^0}{NK}},$$

where $\mathcal{L}^0$ is defined in (12), θ and β are defined in Theorem 5.7, C_1^γ, C_2^γ are two constants defined in Appendix I.

Remark. Our analysis demonstrates that the vanilla Landing method achieves a convergence rate of $\mathcal{O}(1/\sqrt{NK})$ in stochastic scenarios without compression. Furthermore, by comparing Theorem 5.10 and Theorem 5.11, we note that the Landing method attains the same $\mathcal{O}(1/\sqrt{NK})$ rate regardless of whether communication compression is applied. This indicates that the use of compression does not adversely affect the convergence order.

6. Optimization on Block-wise Manifolds

Practical problems typically require variables to satisfy orthogonal constraints in a block-wise manner, while the remaining variables remain unconstrained, e.g. orthogonal CNN with orthogonally regularized convolutional layers and unconstrained MLP layers (Wang et al., 2020). The optimization problem can thus be formulated as follows:

$$\begin{aligned} \min_{X_1,\dots,X_J;x} \quad & f(X_1,\dots,X_J;x) \\ \text{s.t.} \quad & X_j^\top X_j = I_{p_j}, \quad j = 1,\dots,J, \end{aligned} \tag{13}$$

where $X_j \in \mathbb{R}^{n_j \times p_j}$ represents a constrained variable block on $\mathrm{St}(p_j, n_j)$, and $x \in \mathbb{R}^{n_0}$ contains the remaining unconstrained variables. We define $\mathfrak{X} := (X_1,\dots,X_J;x)$, a composite data type, whose domain is given by $\mathfrak{R} := (\mathbb{R}^{n_1\times p_1},\dots,\mathbb{R}^{n_J\times p_J};\mathbb{R}^{n_0})$. Consequently, f is well-defined as a mapping from $\mathfrak{R}$ to $\mathbb{R}$, with its gradient as

$$\nabla f(\mathfrak{X}) := \Big(\frac{\partial f}{\partial X_1},\dots,\frac{\partial f}{\partial X_J};\frac{\partial f}{\partial x}\Big) \in \mathfrak{R}.$$

The inner product on $\mathfrak{R}$ is computed as

$$\langle \mathfrak{X}, \mathfrak{Y}\rangle := \sum_{j=1}^{J}\langle X_j, Y_j\rangle + \langle x, y\rangle_2, \quad \mathfrak{X}, \mathfrak{Y} \in \mathfrak{R}.$$

The norm on $\mathfrak{R}$ is defined as $\|\mathfrak{X}\| := \sqrt{\langle \mathfrak{X}, \mathfrak{X}\rangle}$, allowing Assumptions 3.2, 5.1, 5.2, 5.3, and 5.4 to be properly formulated for problem (13).

By applying the Landing descent direction to each orthogonally constrained block and vanilla gradient descent to the free variable block, the descent direction of the block-wise constrained problem (13) can be expressed as:

$$\begin{aligned} \tilde{\Lambda}(\mathfrak{X};\mathfrak{g}) := \Big(&\mathrm{skew}(\boldsymbol{g}_1 X_1^\top)X_1 + \lambda X_1(X_1^\top X_1 - I_{p_1}), \\ &\dots \\ &\mathrm{skew}(\boldsymbol{g}_J X_J^\top)X_J + \lambda X_J(X_J^\top X_J - I_{p_J}); \\ &\boldsymbol{g}_0\Big) \in \mathfrak{R}, \end{aligned} \tag{14}$$

where $\boldsymbol{g}_1,\dots,\boldsymbol{g}_J,\boldsymbol{g}_0$ can be $\partial f/\partial X_1,\dots,\partial f/\partial X_J$, $\partial f/\partial x$ or their compressed approximations, as indicated by (8)–(9). Using (14), the main update step follows:

$$\mathfrak{X}^{k+1} = \mathfrak{X}^k - \gamma\tilde{\Lambda}(\mathfrak{X};\mathfrak{g}). \tag{15}$$

Therefore, it is straightforward to extend the EF-Landing Algorithm 1 to solve the optimization problem (13) on the block-wise Stiefel manifolds. The implementation details are provided in Appendix J. Moreover, Appendix K presents the convergence results of EF-Landing in stochastic scenario with block-wise constraints.

7. Numerical Experiments

To validate the performance of EF-Landing, we provide experiments on two groups of problems: the distributed online PCA for deterministic scenario and deep learning using VGG16 (Simonyan, 2014) neural network architecture with orthogonal constraints applied to the convolutional layers for stochastic scenario. We compare EF-Landing with other algorithms for optimization on Stiefel manifold, including vanilla Landing, QR retraction and the Euclidean gradient descent with added ℓ_2 squared penalty norm. All tests were implemented in PyTorch and performed using a single GPU. Further experiments can be found in Appendix L.

7.1. Distributed Online PCA

The problem can be expressed as

$$\min_{X\in\mathbb{R}^{n\times p}} -\frac{1}{2N}\sum_{i=1}^{N}\|A_iX\|_F^2, \quad \text{s.t.} \quad X\in \mathrm{St}(p,n),$$

where $A_i \in \mathbb{R}^{l\times n}$ is the synthetically generated local data matrix for node i, with l being the number of samples for each node. We set $l = 5000$, $n = 5000$, $N = 4$ and $p = 1000$. For EF-Landing algorthm, we use two common compressors: Top-K (Wangni et al., 2018) with compression retention ratio 0.1 and QSGD (Alistarh et al., 2017) with quantization level $s = 16$. Using different algorithms mentioned above, we get the results as in Figure 1.

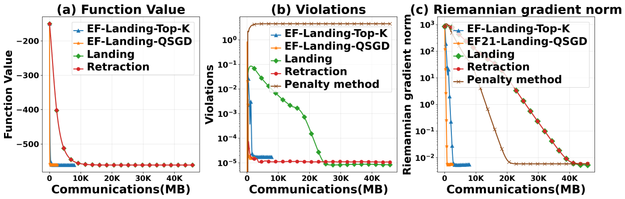

Figure 1: Performance of EF-Landing and other algorithms on distributed online PCA, with the principle component dimension $p = 1000$.

Figure 1 shows that EF-Landing reaches the same accuracy as vanilla Landing algorithm and other algorithms. Furthermore, due to communication compression technique, EF-Landing significantly reduces the communication overhead. Noticing that the function value of penalty method is different from others, it is not shown in the figure.

7.2. Neural Network with Orthogonal Constraints

Another group of experiments are conducted on VGG16. We apply orthogonal constraints to the convolutional layers by reshaping the convolutional kernels to size $n_{\text{out}} \times n_{\text{in}}n_xn_y$, where n_{in} and n_{out} are the number of input and output channels, and n_x, n_y are the filter dimensions (Ablin et al., 2024). In case that the reshaped matrix becomes wide instead of tall, we enforce orthogonality on its transpose. Problem formulation (13) is suitable for our settings.

We test the performance of EF-Landing on the CIFAR-10 dataset, using Top-K and Rand-K compressors with a compression retention ratio of 0.2. We compare the results of EF-Landing with vanilla Landing and QR retraction methods in Figure 2. For each algorithm, the network is trained for 80 epochs, and the learning rate is reduced to $\frac{1}{10}$ of its original value after the 50th epoch.

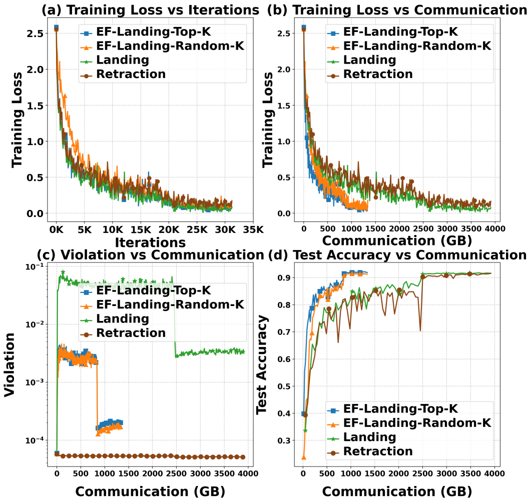

Figure 2: Performance of EF-Landing and other algorithms on deep learning with VGG16 on CIFAR-10.

As shown in Figure 2, EF-Landing and other algorithms reach similar accuracy in terms of iterations, while EF-Landing significantly reduces the communication overhead. Thanks to the well-designed selection of momentum and step size, EF-Landing outperforms vanilla Landing in controlling orthogonality violation.

8. Conclusion

In this paper, we propose EF-Landing algorithm for solving distributed optimization problems on Stiefel manifold, which is both computationally and communicationally efficient. EF-Landing encompasses various scenarios with sharp convergence guarantees and linear speedup rate. The good compatibility with block-wise problems extends the practicality of EF-Landing even further. Extensive numerical experiments validate our theoretical results.

Impact Statement

This paper centers on the theoretical analysis of machine learning algorithm convergence. We do not foresee any significant societal consequences arising from our work, none of which we believe need to be explicitly emphasized.

Bibliography

Ablin, P. and Peyré, G. Fast and accurate optimization on the orthogonal manifold without retraction. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 5636–5657. PMLR, May 2022.

Ablin, P., Vary, S., Gao, B., and Absil, P.-A. Infeasible Deterministic, Stochastic, and Variance-Reduction Algorithms for Optimization under Orthogonality Constraints, October 2024. arXiv:2303.16510 [stat].

Absil, P.-A. and Malick, J. Projection-like Retractions on Matrix Manifolds. *SIAM Journal on Optimization*, 22(1): 135–158, January 2012.

Absil, P.-A., Baker, C., and Gallivan, K. Trust-Region Methods on Riemannian Manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, July 2007.

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, April 2009.

Ahn, K. and Sra, S. From Nesterov's Estimate Sequence to Riemannian Acceleration. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 84–118. PMLR, July 2020.

Alimisis, F., Orvieto, A., Becigneul, G., and Lucchi, A. Momentum Improves Optimization on Riemannian Manifolds. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 1351–1359. PMLR, March 2021.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Arjovsky, M., Shah, A., and Bengio, Y. Unitary Evolution Recurrent Neural Networks. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1120–1128. PMLR, June 2016.

Bansal, N., Chen, X., and Wang, Z. Can We Gain More from Orthogonality Regularizations in Training Deep Networks? In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Boissin, T., Mamalet, F., Fel, T., Picard, A. M., Massena, T., and Serrurier, M. An adaptive orthogonal convolution scheme for efficient and flexible cnn architectures. *arXiv preprint arXiv:2501.07930*, 2025.

Bonnabel, S. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, September 2013. Conference Name: IEEE Transactions on Automatic Control.

Boumal, N., Absil, P.-A., and Cartis, C. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 39(1):1–33, January 2019.

Chen, S., Garcia, A., Hong, M., and Shahrampour, S. Decentralized Riemannian Gradient Descent on the Stiefel Manifold. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 1594–1605. PMLR, July 2021.

Edelman, A., Arias, T. A., and Smith, S. T. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, June 1998.

Fatkhullin, I., Tyurin, A., and Richtarik, P. Momentum Provably Improves Error Feedback! *Advances in Neural Information Processing Systems*, 36:76444–76495, December 2023.

Gao, B., Liu, X., and Yuan, Y.-x. Parallelizable Algorithms for Optimization Problems with Orthogonality Constraints. *SIAM Journal on Scientific Computing*, June 2019.

He, Y., Huang, X., and Yuan, K. Unbiased compression saves communication in distributed optimization: when and how much? *Advances in Neural Information Processing Systems*, 36, 2024.

Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

Hotelling, H. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

Hu, J. and Deng, K. Improving the communication in decentralized manifold optimization through single-step consensus and compression, July 2024. arXiv:2407.08904 [math].

Huang, X., Chen, Y., Yin, W., and Yuan, K. Lower Bounds and Nearly Optimal Algorithms in Distributed Learning with Communication Compression. *Advances in Neural Information Processing Systems*, 35:18955–18969, December 2022.

Lezcano-Casado, M. and Martínez-Rubio, D. Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3794–3803. PMLR, May 2019.

Liu, X., Xiao, N., and Yuan, Y.-x. A Penalty-Free Infeasible Approach for a Class of Nonsmooth Optimization Problems Over the Stiefel Manifold. *Journal of Scientific Computing*, 99(2):30, March 2024.

Liu, Y., Shang, F., Cheng, J., Cheng, H., and Jiao, L. Accelerated First-order Methods for Geodesically Convex Optimization on Riemannian Manifolds. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Lopes, C. G. and Sayed, A. H. Diffusion Least-Mean Squares Over Adaptive Networks: Formulation and Performance Analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, July 2008. Conference Name: IEEE Transactions on Signal Processing.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, April 2017.

Nedić, A., Olshevsky, A., and Shi, W. Achieving Geometric Convergence for Distributed Optimization Over Time-Varying Graphs. *SIAM Journal on Optimization*, December 2017.

Qi, C., Gallivan, K. A., and Absil, P.-A. Riemannian BFGS Algorithm with Applications. In Diehl, M., Glineur, F., Jarlebring, E., and Michiels, W. (eds.), *Recent Advances in Optimization and its Applications in Engineering*, pp. 183–192, Berlin, Heidelberg, 2010. Springer.

Qu, W., Chen, H., Xiu, X., and Liu, W. Distributed sparsity constrained optimization over the Stiefel manifold. *Neurocomputing*, 602:128267, October 2024.

Richtarik, P., Sokolov, I., and Fatkhullin, I. EF21: A New, Simpler, Theoretically Better, and Practically Faster Error Feedback. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4384–4396. Curran Associates, Inc., 2021.

Saxe, A., McClelland, J., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Represenatations 2014*. International Conference on Learning Represenatations 2014, 2014.

Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Interspeech 2014*, pp. 1058–1062. ISCA, September 2014.

Shi, W., Ling, Q., Wu, G., and Yin, W. EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization. *SIAM Journal on Optimization*, May 2015.

Simonyan, K. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Singla, S. and Feizi, S. Skew orthogonal convolutions. In *International Conference on Machine Learning*, pp. 9756–9766. PMLR, 2021.

Stich, S. U. Local sgd converges fast and communicates little. In *ICLR 2019-International Conference on Learning Representations*, 2019a.

Stich, S. U. Local SGD Converges Fast and Communicates Little. In *ICLR 2019 - International Conference on Learning Representations*, May 2019b.

Stich, S. U. and Karimireddy, S. P. The Error-Feedback Framework: Better Rates for SGD with Delayed Gradients and Compressed Communication, June 2021. arXiv:1909.05350 [cs].

Stich, S. U., Cordonnier, J.-B., and Jaggi, M. Sparsified SGD with Memory. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Sun, Y., Chen, S., Garcia, A., and Shahrampour, S. Retraction-Free Decentralized Non-convex Optimization with Orthogonal Constraints, December 2024. arXiv:2405.11590 [cs].

Tripuraneni, N., Flammarion, N., Bach, F., and Jordan, M. I. Averaging Stochastic Gradient Descent on Riemannian Manifolds. In *Proceedings of the 31st Conference On Learning Theory*, pp. 650–687. PMLR, July 2018.

Tsitsiklis, J., Bertsekas, D., and Athans, M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, September 1986. Conference Name: IEEE Transactions on Automatic Control.

Vary, S., Ablin, P., Gao, B., and Absil, P.-A. Optimization without Retraction on the Random Generalized Stiefel Manifold, November 2024. arXiv:2405.01702 [cs].

Wang, J., Chen, Y., Chakraborty, R., and Yu, S. X. Orthogonal Convolutional Neural Networks. pp. 11505–11515, 2020.

Wang, L. and Liu, X. Decentralized Optimization Over the Stiefel Manifold by an Approximate Augmented Lagrangian Function. *IEEE Transactions on Signal Processing*, 70:3029–3041, 2022. Conference Name: IEEE Transactions on Signal Processing.

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient Sparsification for Communication-Efficient Distributed Optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Wen, W., Xu, C., Yan, F., Wu, C., Wang, Y., Chen, Y., and Li, H. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xiao, N., Liu, X., and Yuan, Y.-x. Exact Penalty Function for $\ell_{2,1}$ Norm Minimization over the Stiefel Manifold. *SIAM Journal on Optimization*, December 2021.

Xiao, N., Liu, X., and Yuan, Y.-x. A class of smooth exact penalty function methods for optimization problems with orthogonality constraints. *Optimization Methods and Software*, 37(4):1205–1241, July 2022.

Yu, T., Li, J., Cai, Y., and Li, P. Constructing orthogonal convolutions in an explicit manner. In *International Conference on Learning Representations*, 2022.

Zhang, H. and Sra, S. First-order Methods for Geodesically Convex Optimization. In *Conference on Learning Theory*, pp. 1617–1638. PMLR, June 2016.

Zhang, H., J. Reddi, S., and Sra, S. Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Zhao, J., Wang, X., and Lei, J. Distributed Riemannian Stochastic Gradient Tracking Algorithm on the Stiefel Manifold, January 2025. arXiv:2405.16900 [math].

Zhou, P., Yuan, X.-T., and Feng, J. Faster First-Order Methods for Stochastic Non-Convex Optimization on Riemannian Manifolds. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 138–147. PMLR, April 2019.

A. Proof of Proposition 4.1

Proof. For any $\boldsymbol{g} \in \mathbb{R}^{n\times p}$, we have

$$\begin{aligned}&\langle \mathrm{skew}(\boldsymbol{g}X^\top)X, \lambda X(X^\top X - I_p)\rangle = \mathrm{Tr}\big(X^\top \mathrm{skew}(\boldsymbol{g}X^\top)^\top \cdot \lambda X(X^\top X - I_p)\big)\\ =&\mathrm{Tr}\big(\mathrm{skew}(\boldsymbol{g}X^\top)^\top \cdot \lambda X(X^\top X - I_p)X^\top\big) = \langle \mathrm{skew}(\boldsymbol{g}X^\top), \lambda X(X^\top X - I_p)X^\top\rangle,\end{aligned}$$

where $\mathrm{skew}(\boldsymbol{g}X^\top)$ is a skew-symmetric matrix and $\lambda X(X^\top X - I_p)X^\top$ is a symmetric one. For any skew-symmetric matrix X and symmetric matrix Y,

$$\langle X, Y\rangle = \langle X^\top, Y^\top\rangle = -\langle X, Y\rangle = 0,$$

which leads to the conclusion. □

B. Toy Example Showing the Necessity of Error Feedback

Here is a toy example illustrating the phenomenon of stagnation. Let $X \in \mathbb{R}^{2\times 1}$ and the Stiefel manifold $\{X \in \mathbb{R}^{2\times 1} | X^\top X = 1\}$ is now a unit circle. We only consider a single node with gradient $\nabla f(X) = (2,1)^\top$. Top-1 compressor will compress the gradient into $\mathcal{C}(\nabla f(X)) = (2,0)^\top$. If the iteration point X happens to be $(1,0)^\top$, the quasi-Riemannian gradient will be $\mathbf{0}$ since $\mathcal{C}(\nabla f(X)) = (2,0)^\top$ happens to be orthogonal to the tangent space of Stiefel manifold at point $X = (1,0)^\top$, i.e. $\mathrm{span}\{(0,1)^\top\}$. The iteration point X is on $\mathrm{St}(1,2)$, so the gradient of penalty term $\lambda\mathcal{N}(X)$ is also $\mathbf{0}$. Therefore, the descent direction $\tilde{\Lambda}(X;\mathcal{C}(\nabla f(X)))$ is $\mathbf{0}$ but the iteration haven't achieved a true stationary point yet. Furthermore, greedy compressor Top-1 will permanently output the same compressed gradient $(2,0)^\top$ as long as the iteration point doesn't move. This causes the iteration stagnant.

However, when using error feedback, for the same situation, we first make a stagnant step to $X' = X = (1,0)^\top$. But since the error feedback technique compresses the difference of two consecutive gradients, the next gradient after compression will be $\mathcal{C}(\nabla f(X)) + \mathcal{C}(\nabla f(X') - \mathcal{C}(\nabla f(X))) = (2,0)^\top + (0,1)^\top = (2,1)^\top$, which is no longer orthogonal to the tangent space of X' on Stiefel manifold because the direction parallel with the tangent space comes into effect. Hence, the iteration escapes the stagnation point. Figure 3 illustrates the process.

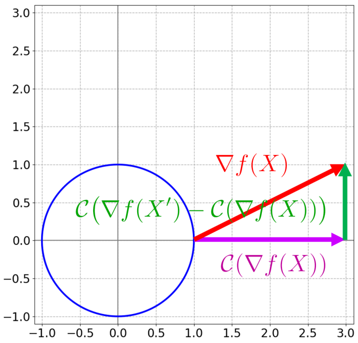
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.0 −0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
$\nabla f(X)$
$\mathcal{C}(\nabla f(X')) - \mathcal{C}(\nabla f(X)))$
$\mathcal{C}(\nabla f(X))$

Figure 3: Diagram for the toy example in Appendix B.

The heuristic reason explaining the phenomenon of stagnation is: Greedy compressors choose the directions with largest magnitude in unconstrained Euclidean space, but these directions are not necessarily directions with largest magnitude in feasible subspaces of Stiefel manifold. In extreme cases, when the two sets of directions do not intersect at all and the greedy compressors cannot give other feasible directions for the current iteration point, the iteration falls into stagnation. Error feedback, however, tends to select directions that were discarded in the previous iteration, ensuring that the feasible directions of Stiefel manifold could always be selected within a few steps. Hence, error feedback technique overcomes the phenomenon of stagnation. In fact, for any deterministic optimization problem on Stiefel manifold, when using greedy compressors, stagnation occurs with high probability. Thus this is a general problem that is eventually solved by error feedback technique.

C. Proof of Lemma 5.5

It is important to note that points within $\mathrm{St}(p,n)^\epsilon$ have bounded singular values (Ablin et al., 2024).

Lemma C.1 (Singular Values). *If $X \in \mathrm{St}(p,n)^\epsilon$, we have $\sqrt{1-\epsilon} \le \sigma \le \sqrt{1+\epsilon}$, for any singular value σ of X.*

And the proof of Lemma 5.5 is also based on (Ablin et al., 2024).

Proof. (Ablin et al., 2024) proved that for any given $X \in \mathrm{St}(p,n)^\epsilon$ and $\boldsymbol{g} \in \mathbb{R}^{n\times p}$, there exists an upper bound of step size $\gamma^*(X;\boldsymbol{g})$. If step size $\gamma(X;\boldsymbol{g}) \le \gamma^*(X;\boldsymbol{g})$, the next iteration remains in $\mathrm{St}(p,n)^\epsilon$. Further, if $\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F = a$, the upper-bound of step size $\gamma^*(X;\boldsymbol{g})$ will not disappear but can be lower-bounded as

$$\gamma^*(X;\boldsymbol{g}) \ge \min\left\{\frac{\lambda(1-\epsilon)\epsilon}{a^2+\lambda^2(1+\epsilon)\epsilon^2}, \sqrt{\frac{\epsilon}{2a^2}}, \frac{1}{2\lambda}\right\}.$$

Let's take one step further, if $\|\boldsymbol{g}\|_F \le L'$, we have:

$$\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2 \le (1+\epsilon)\|\mathrm{skew}(\boldsymbol{g}X^\top)\|_F^2 \le (1+\epsilon)\|\boldsymbol{g}X^\top\|_F^2 \le (1+\epsilon)^2\|\boldsymbol{g}\|_F^2 \le (1+\epsilon)^2(L')^2, \tag{16}$$

where the first and the third equations are from Lemma C.1 and the second is due to the orthogonality between symmetric and skew-symmetric matrices. Therefore, if we choose a uniform step size

$$\gamma \le \gamma_s := \min\left\{\frac{\lambda(1-\epsilon)\epsilon}{(1+\epsilon)^2(L')^2+\lambda^2(1+\epsilon)\epsilon^2}, \sqrt{\frac{\epsilon}{2(1+\epsilon)^2(L')^2}}, \frac{1}{2\lambda}\right\},$$

then for any given $X, \boldsymbol{g}$, the next iteration will remain in the safe region.

□

D. Proof of Lemma 5.6

Lemma D.1 (Gradient of merit function). *$m(X)$ is the merit function defined in (11), and its gradient can be expressed as*

$$\nabla m(X) = \nabla f(X) - \frac{1}{2}\mathcal{J}_X(\Phi)^*[X^\top X - I_p] - X\mathrm{sym}(X^\top \nabla f(X)) + \mu\nabla\mathcal{N}(X), \tag{17}$$

where $\Phi(X) = \mathrm{sym}(\nabla f(X)^\top X) : \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$, $\mathcal{J}_X(\Phi)$ denotes its derivative at X, and $\mathcal{J}_X(\Phi)^*[X^\top X - I_p]$ denotes the adjoint of the Jacobian in the sense of the Frobenius inner product of $\Phi(X) = \mathrm{sym}(X^\top\nabla f(X))$ in X evaluated in the direction $X^\top X - I_p$. Further, let $\mathrm{vec}(\cdot) : \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ denote the vectorization operation, and let $H_X \in \mathbb{R}^{np\times np}$ denote the matrix representation of the Hessian of f at X, we have

$$\mathrm{vec}(\mathcal{J}_X(\Phi)^*[X^\top X - I_p]) = H_X\mathrm{vec}(\nabla\mathcal{N}(X)) + \mathrm{vec}(\nabla f(X)(X^\top X - I_p)). \tag{18}$$

The main part of proof for this lemma can be found at (Ablin et al., 2024).

Now we are ready to prove lemma 5.6.

Proof. **STEP 1.** We will bound all the terms from the inner product $\langle\tilde{\Lambda}(X;\boldsymbol{g}), \nabla m(X)\rangle$ separately by considering the four terms in (17) respectively.

The inner product between the first term of (17) and the descent direction is

$$\begin{aligned}\langle\tilde{\Lambda}(X;\boldsymbol{g}), \nabla f(X)\rangle =& \langle\mathrm{skew}(\boldsymbol{g}X^\top)X + \lambda X(X^\top X - I_p), \nabla f(X)\rangle \\ =& \langle\mathrm{skew}(\boldsymbol{g}X^\top), \nabla f(X)X^\top\rangle + \lambda\langle X^\top X - I_p, X^\top\nabla f(X)\rangle \\ =& \langle\mathrm{skew}(\boldsymbol{g}X^\top), \mathrm{skew}(\nabla f(X)X^\top)\rangle + \lambda\langle X^\top X - I_p, \mathrm{sym}(X^\top\nabla f(X))\rangle.\end{aligned} \tag{19}$$

The first inner product term can be lower bounded as

$$\begin{aligned}&\langle \mathrm{skew}(\boldsymbol{g}X^\top), \mathrm{skew}(\nabla f(X)X^\top)\rangle\\ =&\frac{1}{2}\|\mathrm{skew}(\boldsymbol{g}X^\top)\|_F^2+\frac{1}{2}\|\mathrm{skew}(\nabla f(X)X^\top)\|_F^2-\frac{1}{2}\|\mathrm{skew}((\boldsymbol{g}-\nabla f(X))X^\top)\|_F^2\\ \geq&\frac{1}{2}\|\mathrm{skew}(\boldsymbol{g}X^\top)\|_F^2+\frac{1}{2}\|\mathrm{skew}(\nabla f(X)X^\top)\|_F^2-\frac{\sigma_1^2}{2}\|\boldsymbol{g}-\nabla f(X)\|_F^2, \qquad (20)\end{aligned}$$

where the last inequality is due to the orthogonality between symmetric and skew-symmetric parts of one matrix, and σ_1 is the maximum singular value of X.

For the inner product between the second term of (17) and the descent direction,

$$\begin{aligned}\langle \tilde{\Lambda}(X;\boldsymbol{g}), -\frac{1}{2}\mathcal{J}_X(\Phi)^*[X^\top X-I_p]\rangle =&-\frac{1}{2}\lambda\langle H_X\mathrm{vec}(\nabla\mathcal{N}(X)),\mathrm{vec}(\nabla\mathcal{N}(X))\rangle \qquad (21)\\ &-\frac{1}{2}\langle H_X\mathrm{vec}(\nabla\mathcal{N}(X)),\mathrm{vec}(\mathrm{skew}(\boldsymbol{g}X^\top)X)\rangle \qquad (22)\\ &-\frac{1}{2}\langle \nabla f(X)^\top\mathrm{skew}(\boldsymbol{g}X^\top)X, X^\top X-I_p\rangle \qquad (23)\\ &-\frac{1}{2}\lambda\langle \mathrm{sym}(X^\top\nabla f(X)), (X^\top X-I_p)^2\rangle. \qquad (24)\end{aligned}$$

In the third term of the inner product, the lossy gradient estimate $\boldsymbol{g}$ has no effect:

$$\begin{aligned}\langle \tilde{\Lambda}(X;\boldsymbol{g}), -X\mathrm{sym}(X^\top\nabla f(X))\rangle =&\langle \mathrm{skew}(\boldsymbol{g}X^\top)X+\lambda\nabla\mathcal{N}(X), -X\mathrm{sym}(X^\top\nabla f(X))\rangle \qquad (25)\\ =&-\lambda\langle \mathrm{sym}(X^\top\nabla f(X)), X^\top X(X^\top X-I_p)\rangle. \qquad (26)\end{aligned}$$

The last term is also not affected by the lossy gradient estimate:

$$\langle \tilde{\Lambda}(X;\boldsymbol{g}), \mu\nabla\mathcal{N}(X)\rangle = \lambda\mu\|\nabla\mathcal{N}(X)\|_F^2. \qquad (27)$$

Adding all the four terms and applying the lower bound in the first term gives

$$\begin{aligned}\langle \tilde{\Lambda}(X;\boldsymbol{g}), \nabla m(X)\rangle \geq&\frac{1}{2}\|\mathrm{skew}(\boldsymbol{g}X^\top)\|_F^2+\frac{1}{2}\|\mathrm{skew}(\nabla f(X)X^\top)\|_F^2-\frac{\sigma_1^2}{2}\|\boldsymbol{g}-\nabla f(X)\|_F^2 \qquad (28)\\ &+\lambda\langle(\mu I_{np}-\frac{1}{2}H_X)\mathrm{vec}(\nabla\mathcal{N}(X)),\mathrm{vec}(\nabla\mathcal{N}(X))\rangle \qquad (29)\\ &-\frac{3}{2}\lambda\langle(X^\top X-I_p)^2,\mathrm{sym}(X^\top\nabla f(X))\rangle \qquad (30)\\ &-\frac{1}{2}\langle H_X\mathrm{vec}(\nabla\mathcal{N}(X)),\mathrm{vec}(\mathrm{skew}(\boldsymbol{g}X^\top)X)\rangle \qquad (31)\\ &-\frac{1}{2}\langle \nabla f(X)^\top\mathrm{skew}(\boldsymbol{g}X^\top)X, X^\top X-I_p\rangle, \qquad (32)\end{aligned}$$

where the first line (28) comes from the lower bound (20); the second line (29) is a combination of (21) and (27); the third line (30) comes from the second term of (19), (24) and (26); the fourth and the fifth terms are the rest terms (22) and (23).

STEP 2. we should separately bound the new five terms. Setting aside the first term, we first analyze the second term (29):

$$\begin{aligned}\lambda\langle(\mu I_{np}-\frac{1}{2}H_X)\mathrm{vec}(\nabla\mathcal{N}(X)),\mathrm{vec}(\nabla\mathcal{N}(X))\rangle \geq&\lambda(\mu-\frac{L}{2})\|\nabla\mathcal{N}(X)\|_F^2\\ \geq&4\lambda(\mu-\frac{L}{2})\sigma_p^2\mathcal{N}(X), \qquad (33)\end{aligned}$$

where L is the Lipschitz constant of $\nabla f(X)$ over $\mathrm{St}(p,n)^\epsilon$ defined by Assumption 5.2, and σ_p is the minimum singular value of X; the first inequality comes from the smoothness of $f(X)$ and the second by the property of singular values of X.

The third term (30) can be lower bounded using Cauchy-Schwarz inequality as

$$\begin{aligned}-\frac{3}{2}\lambda\langle (X^\top X - I_p)^2, \mathrm{sym}(X^\top \nabla f(X))\rangle \geq & -6\lambda\mathcal{N}(X)\|\mathrm{sym}(X^\top\nabla f(X))\|_F \\ \geq & -6\lambda\sigma_1 L'\mathcal{N}(X), \end{aligned} \tag{34}$$

where the second inequality is again due to the orthogonality between symmetric and skew-symmetric matrices and L' is such that $\|\nabla f(X)\|_F \leq L'$ for all $X \in \mathrm{St}(p,n)^\epsilon$ defined by 5.1.

We further use Cauchy-Schwarz inequality to bound the fourth and the fifth terms. The fourth term (31) is lower bounded as

$$\begin{aligned}-\frac{1}{2}\langle H_X \mathrm{vec}(\nabla\mathcal{N}(X)), \mathrm{vec}(\mathrm{skew}(\boldsymbol{g}X^\top)X)\rangle \geq & -\frac{L}{2}\|X(X^\top X - I_p)\|_F\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F \\ \geq & -L\sigma_1\sqrt{\mathcal{N}(X)}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F. \end{aligned} \tag{35}$$

The fifth term is lower bounded as

$$\begin{aligned}-\frac{1}{2}\langle \nabla f(X)^\top \mathrm{skew}(\boldsymbol{g}X^\top)X, X^\top X - I_p\rangle \geq & -\frac{1}{2}\|\nabla f(X)\|_F\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F\|X^\top X - I_p\|_F \\ \geq & -L'\sqrt{\mathcal{N}(X)}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F. \end{aligned} \tag{36}$$

Putting the fourth and the fifth terms together, we have

$$\begin{aligned} & -\frac{1}{2}\langle H_X \mathrm{vec}(\nabla\mathcal{N}(X)), \mathrm{vec}(\mathrm{skew}(\boldsymbol{g}X^\top)X)\rangle - \frac{1}{2}\langle \nabla f(X)^\top \mathrm{skew}(\boldsymbol{g}X^\top)X, X^\top X - I_p\rangle \\ \geq & -(L' + L\sigma_1)\sqrt{\mathcal{N}(X)}\|\mathrm{skew}(gX^\top)X\|_F \\ \geq & -\frac{1}{2}(L' + L\sigma_1)(b\mathcal{N}(X) + b^{-1}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2), \end{aligned} \tag{37}$$

where in the last inequality we use the average inequality $\sqrt{xy} \leq \frac{1}{2}(x+y)$ with $x = b\mathcal{N}(X)$ and $y = b^{-1}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2$ for an arbitrary $b > 0$ which will be specified later.

Now for the first term (28), by using the property of singular values of X again, we have

$$\|\mathrm{skew}(\boldsymbol{g}X^\top)\|_F \geq \sigma_1^{-1}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F, \quad \|\mathrm{skew}(\nabla f(X)X^\top)\|_F \geq \sigma_1^{-1}\|\mathrm{grad}f(X)\|_F. \tag{38}$$

Adding all lower bounds (33), (34), (37) and (38) together, we have a total lower bound expressed as

$$\begin{aligned}\langle\tilde\Lambda(X;\boldsymbol{g}), \nabla m(X)\rangle \geq & \frac{1}{2\sigma_1^2}\|\mathrm{grad}f(X)\|_F^2 + \|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2\Big(\frac{1}{2\sigma_1^2} - \frac{1}{2}\frac{(L' + L\sigma_1)}{b}\Big) \\ & -\frac{\sigma_1^2}{2}\|\boldsymbol{g} - \nabla f(X)\|_F^2 + \mathcal{N}(X)\Big(4\lambda(\mu - \frac{L}{2})\sigma_p^2 - 6\lambda\sigma_1 L' - \frac{1}{2}(L' + L\sigma_1)b\Big). \end{aligned} \tag{39}$$

Finally, we choose b so that the coefficient of $\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2$ equals to $1/4$, namely $b = \frac{2\sigma_1^2(L' + L\sigma_1)}{2 - \sigma_1^2}$, and noticing $\sqrt{1-\epsilon} \leq \sigma_p \leq \sigma_1 \leq \sqrt{1+\epsilon} \leq \sqrt{2}$, we have

$$\begin{aligned}\langle\tilde\Lambda(X;\boldsymbol{g}), \nabla m(X)\rangle \geq & \frac{1}{4}\|\mathrm{grad}f(X)\|_F^2 + \frac{1}{4}\|\mathrm{skew}(\boldsymbol{g}X^\top)X\|_F^2 - \|\boldsymbol{g} - \nabla f(X)\|_F^2 \\ & + \mathcal{N}(X)\Big(4\lambda(\mu - \frac{L}{2})\sigma_p^2 - 6\lambda\sigma_1 L' - \sigma_1^2\frac{(L' + L\sigma_1)^2}{2 - \sigma_1^2}\Big). \end{aligned} \tag{40}$$

Denoting $\hat L = \max\{L', L\}$, the last term of (40) can be simplified as

$$\begin{aligned} & \mathcal{N}(X)(4\lambda(\mu - \frac{L}{2})\sigma_p^2 - 6\lambda\sigma_1 L' - \sigma_1^2\frac{(L' + L\sigma_1)^2}{2 - \sigma_1^2} \\ \geq & \mathcal{N}(X)(4\lambda(\mu - \frac{L}{2})(1-\epsilon) - 6\lambda\sqrt{1+\epsilon}L' - \hat L^2(1+\epsilon)\frac{(2\sqrt{1+\epsilon})^2}{2 - (1-\epsilon)}) \\ \geq & \mathcal{N}(X)(4\lambda(\mu - \frac{L}{2})(1-\epsilon) - 6\lambda\sqrt{1+\epsilon}L' - 4\hat L^2(1+\epsilon)) \\ \geq & \lambda\mu\mathcal{N}(X), \end{aligned} \tag{41}$$

where in the last inequality, we choose μ as

$$\mu \geq \frac{2}{3-4\epsilon}\Big(L(1-\epsilon) + 3\sqrt{1+\epsilon}L' + 2\hat{L}^2\frac{1+\epsilon}{\lambda}\Big), \tag{42}$$

which leads to the conclusion. □

E. Proof of Theorem 5.7

We introduce some notation for convenience.

- Auxiliary variables: momentum vector of the master$\boldsymbol{v}^k := \frac{1}{N}\sum_{i=1}^N \boldsymbol{v}_i^k$; random variable of the master $\boldsymbol{\xi}^k := (\xi_1^k, \ldots, \xi_N^k)$; stochastic gradient of the master $\nabla F(X^k; \boldsymbol{\xi}^k) := \frac{1}{N}\sum_{i=1}^N \nabla F(X^k; \xi_i^k)$.
- Momentum and stochastic error: error of each node $P_i^k := \|\boldsymbol{v}_i^k - \nabla f_i(X^k)\|_F^2$; averaged error $\tilde{P}^k := \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{v}_i^k - \nabla f_i(X^k)\|_F^2$; error of the master $P^k := \|\boldsymbol{v}^k - \nabla f(X^k)\|_F^2$.
- Compression error: error of each node $G_i^k := \|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2$; averaged error $\tilde{G}^k := \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2$; error of the master $G^k := \|\boldsymbol{g}^k - \boldsymbol{v}^k\|_F^2$.
- Filtrations: filtration for the conditional expectation of stochastic gradient $\mathcal{F}_P^k := \{\boldsymbol{\xi}^0, X^1, \boldsymbol{\xi}^1, X^2, \ldots, \boldsymbol{\xi}^{k-1}, X^k\}$; filtration for the conditional expectation of compressor $\mathcal{F}_C^k := \{\boldsymbol{\xi}^0, X^1, \boldsymbol{\xi}^1, X^2, \ldots, \boldsymbol{\xi}^{k-1}, X^k, \boldsymbol{\xi}^k\}$.

Next, for arbitrary momentum factor η, The following two lemmas provide the iterative formats of two kinds of errors.

Lemma E.1. *The iteration of momentum and stochastic error satisfies*

$$\mathbb{E}[P_i^{k+1}] \leq (1-\eta)\mathbb{E}[P_i^k] + (1-\eta)^2(1+\frac{1}{\eta})L_i^2\mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + \eta^2\sigma^2; \tag{43}$$

$$\mathbb{E}[\tilde{P}^{k+1}] \leq (1-\eta)\mathbb{E}[\tilde{P}^k] + (1-\eta)^2(1+\frac{1}{\eta})\tilde{L}^2\mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + \eta^2\sigma^2; \tag{44}$$

$$\mathbb{E}[P^{k+1}] \leq (1-\eta)\mathbb{E}[P^k] + (1-\eta)^2(1+\frac{1}{\eta})L^2\mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + \frac{\eta^2\sigma^2}{N}. \tag{45}$$

Proof. For the first inequality (43),

$$\begin{aligned}
\mathbb{E}[P_i^{k+1}] =& \mathbb{E}\big[\|\boldsymbol{v}_i^{k+1} - \nabla f_i(X^{k+1})\|_F^2\big] \\
=& \mathbb{E}\big[\|(1-\eta)\boldsymbol{v}_i^k + \eta\nabla F(X^{k+1}; \xi_i^{k+1}) - \nabla f_i(X^{k+1})\|_F^2\big] \\
=& \mathbb{E}\Big[\big\|(1-\eta)\big(\boldsymbol{v}_i^k - \nabla f_i(X^{k+1})\big) + \eta\big(\nabla F(X^{k+1}; \xi_i^{k+1}) - \nabla f_i(X^{k+1})\big)\big\|_F^2\Big] \\
=& \mathbb{E}\Big[\mathbb{E}_{\xi_i^{k+1}}\Big[\big\|(1-\eta)\big(\boldsymbol{v}_i^k - \nabla f_i(X^{k+1})\big) + \eta\big(\nabla F(X^{k+1}; \xi_i^{k+1}) - \nabla f_i(X^{k+1})\big)\big\|_F^2\Big|\mathcal{F}_P^{k+1}\Big]\Big] \\
\leq& \mathbb{E}\big[(1-\eta)^2\|\boldsymbol{v}_i^k - \nabla f_i(X^{k+1})\|_F^2\big] + \eta^2\sigma^2 \\
\leq& (1-\eta)^2(1+\eta)\mathbb{E}\big[\|\boldsymbol{v}_i^k - \nabla f_i(X^k)\|_F^2\big] + (1-\eta)^2(1+\frac{1}{\eta})\mathbb{E}\big[\|\nabla f_i(X^{k+1}) - \nabla f_i(X^k)\|_F^2\big] + \eta^2\sigma^2 \\
\leq& (1-\eta)\mathbb{E}[P_i^k] + (1-\eta)^2(1+\frac{1}{\eta})L_i^2\mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + \eta^2\sigma^2,
\end{aligned} \tag{46}$$

where the second equation is by the update rule of $\boldsymbol{v}_i^{k+1}$, the first inequality is due to Assumption 5.4, the second inequality is the consequence of Young's inequality and the last is by the smoothness of each local function.

By simply summing all inequalities in terms of node from 1 to N, we get the second inequality (44). Noticing that $\nabla F(X^k; \boldsymbol{\xi}^k) := \frac{1}{N}\sum_{i=1}^N \nabla F(X^k; \xi_i^k)$ and according to the independency of each entry of $\boldsymbol{\xi}$, we have $\mathbb{E}_{\xi^{k+1}}\big[\|\nabla F(X^{k+1}; \boldsymbol{\xi}^{k+1}) - \nabla f(X^{k+1})\|_F^2\big|\mathcal{F}_P^{k+1}\big] \leq \sigma^2/N$. Similarly, we can derive the third inequality (45).

□

Lemma E.2. *The iteration of compression error satisfies*

$$\mathbb{E}[G_i^{k+1}] \leq (1-\theta)\mathbb{E}[G_i^k] + 2\beta\eta^2 L_i^2 \mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + 2\beta\eta^2\mathbb{E}[P_i^k] + (1-\alpha)\eta^2\sigma^2; \tag{47}$$

$$\mathbb{E}[\tilde{G}^{k+1}] \leq (1-\theta)\mathbb{E}[\tilde{G}^k] + 2\beta\eta^2 \tilde{L}^2 \mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + 2\beta\eta^2\mathbb{E}[\tilde{P}^k] + (1-\alpha)\eta^2\sigma^2, \tag{48}$$

where θ and β are two scalars related to the contractive factor α, i.e. $\theta := 1 - \sqrt{1-\alpha}$ and $\beta := \frac{1-\alpha}{1-\sqrt{1-\alpha}}$.

Proof. For the first inequality (47),

$$\begin{aligned}
\mathbb{E}[G_i^{k+1}] =& \mathbb{E}\big[\|\boldsymbol{g}_i^{k+1} - \boldsymbol{v}_i^{k+1}\|_F^2\big] = \mathbb{E}\big[\|\boldsymbol{g}_i^k + \boldsymbol{c}_i^k - \boldsymbol{v}_i^{k+1}\|_F^2\big] \\
=& \mathbb{E}\big[\|(\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k) - \mathcal{C}(\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k)\|_F^2\big] = \mathbb{E}\Big[\mathbb{E}_{\mathcal{C}}\big[\|(\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k) - \mathcal{C}(\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k)\|_F^2 \big| \mathcal{F}_C^{k+1}\big]\Big] \\
\leq& (1-\alpha)\mathbb{E}\big[\|\boldsymbol{v}_i^{k+1} - \boldsymbol{g}_i^k\|_F^2\big] = (1-\alpha)\mathbb{E}\big[\|(1-\eta)\boldsymbol{v}_i^k + \eta\nabla F(X^{k+1};\xi_i^{k+1}) - \boldsymbol{g}_i^k\|_F^2\big] \\
=& (1-\alpha)\mathbb{E}\Big[\Big\|\boldsymbol{v}_i^k - \boldsymbol{g}_i^k + \eta\big(\nabla f_i(X^{k+1}) - \boldsymbol{v}_i^k\big) + \eta\big(\nabla F(X^{k+1};\xi_i^{k+1}) - \nabla f(X^{k+1})\big\|_F^2\Big] \\
=& (1-\alpha)\mathbb{E}\Big[\mathbb{E}_{\xi_i^{k+1}}\Big[\Big\|\boldsymbol{v}_i^k - \boldsymbol{g}_i^k + \eta\big(\nabla f_i(X^{k+1}) - \boldsymbol{v}_i^k\big) + \eta\big(\nabla F(X^{k+1};\xi_i^{k+1}) - \nabla f(X^{k+1})\big\|_F^2 \Big| \mathcal{F}_P^{k+1}\Big]\Big] \\
\leq& (1-\alpha)\mathbb{E}\Big[\Big\|\boldsymbol{v}_i^k - \boldsymbol{g}_i^k + \eta\big(\nabla f_i(X^{k+1}) - \boldsymbol{v}_i^k\big)\Big\|_F^2\Big] + (1-\alpha)\eta^2\sigma^2 \\
\leq& (1-\alpha)(1+\rho)\mathbb{E}\big[\|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2\big] + (1-\alpha)(1+\frac{1}{\rho})\eta^2\mathbb{E}\big[\|\nabla f_i(X^{k+1}) - \boldsymbol{v}_i^k\|_F^2\big] + (1-\alpha)\eta^2\sigma^2,
\end{aligned} \tag{49}$$

where the first inequality is due the property of contractive compressors (Assumption 3.2), the second inequality is by the bounded variance (Assumption 5.4) and the third is the consequence of Young's inequality with parameter ρ. We should choose a proper ρ satisfying $(1-\alpha)(1+\rho) < 1$ and $(1-\alpha)(1+1/\rho) < +\infty$. (Richtarik et al., 2021) provided a to some degree optimal choice $\rho = \frac{1}{\sqrt{1-\alpha}} - 1$. Denoting $1-\theta = (1-\alpha)(1+\rho) = \sqrt{1-\alpha}$, and $\beta = (1-\alpha)(1+1/\rho) = \frac{1-\alpha}{1-\sqrt{1-\alpha}}$, we have

$$\begin{aligned}
\mathbb{E}[G_i^{k+1}] \leq& (1-\theta)\mathbb{E}\big[\|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2\big] + \beta\eta^2\mathbb{E}\big[\|\nabla f_i(X^{k+1}) - \boldsymbol{v}_i^k\|_F^2\big] + (1-\alpha)\eta^2\sigma^2 \\
\leq& (1-\theta)\mathbb{E}\big[\|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2\big] + 2\beta\eta^2\mathbb{E}[\|\nabla f_i(X^{k+1}) - \nabla f_i(X^k)\|_F^2] \\
&+ 2\beta\eta^2\mathbb{E}\big[\|\nabla f_i(X^k) - \boldsymbol{v}_i^k\|_F^2\big] + (1-\alpha)\eta^2\sigma^2 \\
\leq& (1-\theta)\mathbb{E}[G_i^k] + 2\beta\eta^2 L_i^2\mathbb{E}\big[\|X^{k+1} - X^k\|_F^2\big] + 2\beta\eta^2\mathbb{E}[P_i^k] + (1-\alpha)\eta^2\sigma^2.
\end{aligned} \tag{50}$$

The second inequality (48) follows the same process as lemma E.1. Lacking the unbiasedness assumption of the compressor, we have no similar inequality for master, which is fortunately unnecessary. □

Another lemma helps us split $\|X^{k+1} - X^k\|_F^2$ term to two orthogonal component used for convergence analysis.

Lemma E.3.

$$\|X^{k+1} - X^k\|_F^2 = \gamma^2\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2 \leq \gamma^2\Big(\big\|\mathrm{skew}\big(\tilde{\boldsymbol{g}}^k(X^k)^\top\big)\big\|_F^2 + 4\lambda^2(1+\epsilon)\mathcal{N}(X^k)\Big) \tag{51}$$

Proof. This lemma can be easily proved by the orthogonality between two terms of $\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)$ and the sigular values bound of X^k (Lemma C.1). □

Now we are able to prove the main theorem.

Proof. **STEP 1. Dividing different Errors.** Using smoothness of $m(X^k)$ mentioned in Section 3.1 and Lemma 5.6, we have:

$$\begin{aligned}
m(X^{k+1}) \leq & m(X^k) + \langle \nabla m(X^k), X^{k+1} - X^k \rangle + \frac{L_m}{2} \|X^{k+1} - X^k\|_F^2 \\
= & m(X^k) - \gamma \langle \nabla m(X^k), \tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k) \rangle + \frac{L_m \gamma^2}{2} \|\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)\|_F^2 \\
\leq & m(X^k) - \frac{\gamma}{4} \|\mathrm{grad} f(X^k)\|_F^2 - \frac{\gamma}{4} \left\|\mathrm{skew}\left(\tilde{\boldsymbol{g}}^k (X^k)^\top\right) X^k\right\|_F^2 + \gamma \|\tilde{\boldsymbol{g}}^k - \nabla f(X^k)\|_F^2 \\
& - \gamma \lambda \mu \mathcal{N}(X^k) + \frac{L_m \gamma^2}{2} \|\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)\|_F^2.
\end{aligned} \tag{52}$$

Denote $\mathbb{S} := \left\{ X \in \mathbb{R}^{n \times p} \middle| \|X\|_F \leq L \right\}$, then we have $\tilde{\boldsymbol{g}}^k = \mathrm{proj}_{\mathbb{S}}(\boldsymbol{g}^k)$, and $\nabla f(X^k) = \mathrm{proj}_{\mathbb{S}}(\nabla f(X^k))$. Therefore, using the property of projections,

$$\|\tilde{\boldsymbol{g}}^k - \nabla f(X^k)\|_F^2 \leq \|\boldsymbol{g}^k - \nabla f(X^k)\|_F^2 \tag{53}$$

Next, we need to separete the momentum and stochastic error as well as compression error. We first introduce an inequality $\|\boldsymbol{g}^k - \nabla f(X^k)\|_F^2 \leq c_1 \|\boldsymbol{g}^k - \boldsymbol{v}^k\|_F^2 + c_2 \|\boldsymbol{v}^k - \nabla f(X^k)\|_F^2$, where two undetermined constants are specified in three cases:

- When the momentum rate $\eta = 1$ and the variance of gradients $\sigma^2 = 0$, the momentum and stochastic error will be 0, so c_1, c_2 are specified as $c_1 = 1, c_2 = 0$, which is trivial.

- When the contractive factor $\alpha = 1$, the compression error will be 0, so c_1, c_2 are specified as $c_1 = 0, c_2 = 1$, which is trivial too.

- Otherwise, we specify $c_1 = c_2 = 2$, which is the consequence of mean value inequality.

Therefore,

$$\begin{aligned}
\|\boldsymbol{g}^k - \nabla f(X^k)\|_F^2 \leq & c_1 \|\boldsymbol{g}^k - \boldsymbol{v}^k\|_F^2 + c_2 \|\boldsymbol{v}^k - \nabla f(X^k)\|_F^2 \\
\leq & \frac{c_1}{N} \sum_{i=1}^N \|\boldsymbol{g}_i^k - \boldsymbol{v}_i^k\|_F^2 + c_2 \|\boldsymbol{v}^k - \nabla f(X^k)\|_F^2 = c_1 \tilde{G}^k + c_2 P^k,
\end{aligned} \tag{54}$$

where the second inequality is due to Jensen's inequality.

For convenience, we denote $S^k = \|\mathrm{skew}(\tilde{\boldsymbol{g}}^k (X^k)^\top) X^k\|_F^2$, whose influence will be eliminated later by choosing a proper step size. Substituting (54) into (52) after using (53) , we have:

$$m(X^{k+1}) \leq m(X^k) - \frac{\gamma}{4} \|\mathrm{grad} f(X^k)\|_F^2 - \frac{\gamma}{4} S^k + \gamma c_1 \tilde{G}^k + \gamma c_2 P^k - \gamma \lambda \mu \mathcal{N}(X^k) + \frac{L_m \gamma^2}{2} \|\tilde{\Lambda}(X^k; \tilde{\boldsymbol{g}}^k)\|_F^2 \tag{55}$$

STEP 2. Constructing Lyapunov Function. Adding $\frac{c_1 \gamma}{\theta} G^{k+1}$ to both sides of 55, taking expectation and using lemma

E.2, we have:

$$
\begin{aligned}
&\mathbb{E}[m(X^{k+1})] + \frac{c_1\gamma}{\theta}\mathbb{E}[\tilde{G}^{k+1}] \\
\leq&\mathbb{E}[m(X^k)] - \frac{\gamma}{4}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] - \frac{\gamma}{4}\mathbb{E}[S^k] \\
&+ \frac{c_1\gamma}{\theta}\Big(\mathbb{E}[\tilde{G}^k] + 2\beta\eta^2\tilde{L}^2\mathbb{E}[\|X^{k+1} - X^k\|_F^2] + 2\beta\eta^2\mathbb{E}[\tilde{P}^k] + (1-\alpha)\eta^2\sigma^2\Big) \\
&+ c_2\gamma\mathbb{E}[P^k] - \gamma\lambda\mu\mathbb{E}[\mathcal{N}(X^k)] + \frac{L_m\gamma^2}{2}\mathbb{E}[\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2] \\
=&\mathbb{E}[m(X^k)] + \frac{c_1\gamma}{\theta}\mathbb{E}[\tilde{G}^k] \\
&- \frac{\gamma}{4}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] - \frac{\gamma}{4}\mathbb{E}[S^k] - \gamma\lambda\mu\mathbb{E}[\mathcal{N}(X^k)] \\
&+ \Big(\frac{2c_1\gamma^3\eta^2\beta\tilde{L}^2}{\theta} + \frac{L_m\gamma^2}{2}\Big)\mathbb{E}[\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2] \\
&+ \frac{2c_1\gamma\eta^2\beta}{\theta}\mathbb{E}[\tilde{P}^k] + c_2\gamma\mathbb{E}[P^k] + \frac{c_1\gamma\eta^2(1-\alpha)\sigma^2}{\theta}.
\end{aligned}
\tag{56}
$$

Further adding $\frac{2c_1\gamma\eta\beta}{\theta}\tilde{P}^{k+1}$, $\frac{c_2\gamma}{\eta}P^{k+1}$ to both sides, subtracting lower bound m^* by Assumption 5.3, and using lemma E.1, we have the following collated inequality

$$
\begin{aligned}
&\mathbb{E}[m(X^{k+1})] - m^* + \frac{c_1\gamma}{\theta}\mathbb{E}[\tilde{G}^{k+1}] + \frac{2c_1\gamma\eta\beta}{\theta}\mathbb{E}[\tilde{P}^{k+1}] + \frac{c_2\gamma}{\eta}\mathbb{E}[P^{k+1}] \\
\leq&\mathbb{E}[m(X^k)] - m^* + \frac{c_1\gamma}{\theta}\mathbb{E}[\tilde{G}^k] + \frac{2c_1\gamma\eta\beta}{\theta}\mathbb{E}[\tilde{P}^k] + \frac{c_2\gamma}{\eta}\mathbb{E}[P^k] \\
&- \frac{\gamma}{4}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] - \frac{\gamma}{4}\mathbb{E}[S^k] - \gamma\lambda\mu\mathbb{E}[\mathcal{N}(X^k)] \\
&+ \Big(\frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3)\gamma^3 + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta)\gamma^3 + \frac{L_m\gamma^2}{2}\Big)\mathbb{E}[\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2] \\
&+ \frac{c_1\gamma\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{2c_1\gamma\eta^3\beta\sigma^2}{\theta} + \frac{c_2\gamma\eta\sigma^2}{N},
\end{aligned}
\tag{57}
$$

which corresponds to the Lyapunov function we defined as (12).

STEP 3. Choosing Step Size. Next, we split $\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2$ using lemma E.3, which splits $\mathbb{E}[\|\tilde{\Lambda}(X^k;\tilde{\boldsymbol{g}}^k)\|_F^2]$ into $\mathbb{E}[S^k]$ and $\mathbb{E}[\mathcal{N}(X^k)]$. To eliminate the influence of S^k and to make the coefficient before $\mathcal{N}(X^k)$ negative enough, we are looking for a sufficiently small step size to make the coefficients before $\mathbb{E}[S^k]$ and $\mathbb{E}[\mathcal{N}(X^k)]$ satisfy

$$
-\frac{\gamma}{4} + \frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3)\gamma^3 + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta)\gamma^3 + \frac{L_m\gamma^2}{2} \leq 0 \tag{58}
$$

and

$$
-\gamma\lambda\mu + 4\lambda^2(1+\epsilon)\Big(\frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3)\gamma^3 + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta)\gamma^3 + \frac{L_m\gamma^2}{2}\Big) \leq -\frac{\gamma\lambda\mu}{2}. \tag{59}
$$

It's reducible to a quadratic inequality problem. suppose $a, b > 0$, if we choose $0 \leq x \leq (\sqrt{a} + b)^{-1}$, the inequality $ax^2 + bx \leq 1$ holds. So it's enough to choose step size γ as

$$
\gamma = \min\{\gamma_s, \gamma_1, \gamma_2\},
$$

where γ_s is the uniform safe step size defined by Lemma 5.5, and γ_1, γ_2 are defined as

$$
\begin{aligned}
\gamma_1 :=& \big(2\sqrt{a} + 2L_m\big)^{-1} \\
\gamma_2 :=& \Big(2\sqrt{\frac{2\lambda^2(1+\epsilon)a}{\mu}} + \frac{4\lambda^2(1+\epsilon)L_m}{\mu}\Big)^{-1},
\end{aligned}
$$

where

$$a = \frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3) + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta).$$

With a small enough step size γ, we have

$$\mathbb{E}[\mathcal{L}^{k+1}] \le \mathbb{E}[\mathcal{L}^k] - \frac{\gamma}{4}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] - \frac{\gamma\lambda\mu}{2}\mathbb{E}[\mathcal{N}(X^k)] + \frac{c_1\gamma\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{2c_1\gamma\eta^3\beta\sigma^2}{\theta} + \frac{c_2\gamma\eta\sigma^2}{N} \tag{60}$$

STEP 4. Summing Up. Summing up all inequalities from $k=0$ to $K-1$ and dividing by K, we have

$$\frac{1}{4K}\sum_{k=0}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=0}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \le \frac{\mathcal{L}^0}{\gamma K} + \frac{c_1\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{2c_1\eta^3\beta\sigma^2}{\theta} + \frac{c_2\eta\sigma^2}{N}, \tag{61}$$

□

F. Details of Theorem 5.8 and Corollary 5.9

1. Theorem 5.8. When $\sigma = 0$, $\eta = 1$, $c_1 = 1$ and $c_2 = 0$, the momentum and stochastic error $\tilde{P}^k = 0$, so the Lyapunov function defined in (12) can be simplified as

$$\mathcal{L}^k = m(X^k) - m^* + \frac{\gamma}{\theta}\tilde{G}^k. \tag{62}$$

According to the rule of choosing step size in Appendix E, the step size should be chosen as

$$\gamma = \min\{\gamma_s, \gamma_1, \gamma_2\},$$

where γ_s is the uniform safe step size defined by Lemma 5.5, and γ_1, γ_2 are defined as

$$\gamma_1 := \left(2\sqrt{a} + 2L_m\right)^{-1}$$
$$\gamma_2 := \left(2\sqrt{\frac{2\lambda^2(1+\epsilon)a}{\mu}} + \frac{4\lambda^2(1+\epsilon)L_m}{\mu}\right)^{-1},$$

where

$$a = \frac{2\beta\tilde{L}^2}{\theta}.$$

2. Theorem 5.9. When $\alpha = 1$, the compression error $\tilde{G}^k$ will be 0, so the Lyapunov function can be further simplified as $\mathcal{L}^k = m(X^k) - m^*$. And the choice of the step size can also be simplified as

$$\gamma = \min\{\gamma_s, \frac{1}{2L_m}, \frac{\mu}{4\lambda^2(1+\epsilon)L_m}\}$$

.

G. An adequate Selection of Step Size with momentum

Before deriving conclusions about convergence with momentum, we give a lemma about the step size. This lemma gives an adequate but not necessary choice for step size, for the convenience of analyzing the influence of momentum.

Lemma G.1 (An adequate Selection of Step Size with momentum)**.** *For the step size defined in Theorem 5.7, fix momentum rate $\eta \in (0,1)$. If we further choose $\gamma = \min\{\gamma_s, \frac{1}{6\tilde{L}}\sqrt{\frac{\theta}{2c_1\beta}}, \frac{\eta}{6L\sqrt{2c_2}}, \frac{1}{6L_m}, \frac{1}{12\tilde{L}}\sqrt{\frac{\mu\theta}{c_1\lambda^2(1+\epsilon)\beta}}, \frac{\eta}{12L}\sqrt{\frac{\mu}{c_2\lambda^2(1+\epsilon)}}, \frac{\mu}{12\lambda^2(1+\epsilon)L_m}\}$, then the step size satisfies $\gamma \le \min\{\gamma_s, \gamma_1, \gamma_2\}$ defined in Theorem 5.7.*

Proof. The rule of choosing step size in Appendix E is

$$\gamma = \min\{\gamma_s, \gamma_1, \gamma_2\},$$

where γ_s is the uniform safe step size defined by Lemma 5.5, and γ_1, γ_2 are defined as

$$\begin{aligned} \gamma_1 &:= \left(2\sqrt{a} + 2L_m\right)^{-1} \\ \gamma_2 &:= \left(2\sqrt{\frac{2\lambda^2(1+\epsilon)a}{\mu}} + \frac{4\lambda^2(1+\epsilon)L_m}{\mu}\right)^{-1}, \end{aligned}$$

where

$$a = \frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3) + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta).$$

We can bound $\sqrt{a}$ as

$$\sqrt{a} = \sqrt{\frac{2c_1\beta\tilde{L}^2}{\theta}(1-\eta+\eta^3) + \frac{c_2L^2}{\eta^2}(1-\eta)^2(1+\eta)} \le \sqrt{\frac{2c_1\beta\tilde{L}^2}{\theta} + \frac{2c_2L^2}{\eta^2}} \le \tilde{L}\sqrt{\frac{2c_1\beta}{\theta}} + \frac{L}{\eta}\sqrt{2c_2}, \tag{63}$$

where the last inequality is due to $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$, $x, y \ge 0$.

Therefore,

$$\gamma_1 = (2\sqrt{a} + 2L_m)^{-1} \ge \left(2\tilde{L}\sqrt{\frac{2c_1\beta}{\theta}} + 2\frac{L}{\eta}\sqrt{2c_2} + 2L_m\right)^{-1}. \tag{64}$$

For $(x+y+z)^{-1}$, $x, y, z > 0$, if we denote $M = \max\{x, y, z\}$, then $\frac{1}{M} = \min\{\frac{1}{x}, \frac{1}{y}, \frac{1}{z}\}$. So $(x+y+z)^{-1} \ge \frac{1}{3M} = \frac{1}{3}\min\{\frac{1}{x}, \frac{1}{y}, \frac{1}{z}\}$. Hence, if we choose $\gamma \le \min\{\frac{1}{6\tilde{L}}\sqrt{\frac{\theta}{2c_1\beta}}, \frac{\eta}{6L\sqrt{2c_2}}, \frac{1}{6L_m}\}$, then it satisfies $\gamma \le \gamma_1$.

The same reasoning leads to

$$\gamma_2 = \left(2\sqrt{\frac{2\lambda^2(1+\epsilon)a}{\mu}} + \frac{4\lambda^2(1+\epsilon)L_m}{\mu}\right)^{-1} \ge \left(4\tilde{L}\sqrt{\frac{c_1\lambda^2(1+\epsilon)\beta}{\mu\theta}} + 4\frac{L}{\eta}\sqrt{\frac{c_2\lambda^2(1+\epsilon)}{\mu}} + \frac{4\lambda^2(1+\epsilon)L_m}{\mu}\right)^{-1}. \tag{65}$$

Choosing $\gamma \le \min\{\frac{1}{12\tilde{L}}\sqrt{\frac{\mu\theta}{c_1\lambda^2(1+\epsilon)\beta}}, \frac{\eta}{12L}\sqrt{\frac{\mu}{c_2\lambda^2(1+\epsilon)}}, \frac{\mu}{12\lambda^2(1+\epsilon)L_m}\}$ satisfies $\gamma \le \gamma_2$. And the conclusion follows.

□

H. Proof of Theorem 5.10

Proof. When $c_1 = c_2 = 2$, the Lyapunov function is

$$\mathcal{L}^k = m(X^k) - m^* + \frac{2\gamma}{\theta}\tilde{G}^k + \frac{4\gamma\eta\beta}{\theta}\tilde{P}^k + \frac{2\gamma}{\eta}P^k. \tag{66}$$

Choosing step size γ as Lemma G.1 and setting $c_1 = c_2 = 2$ leads to

$$\frac{1}{\gamma} \le \left(\frac{1}{\gamma_s} + 6L_m + \frac{12\lambda^2(1+\epsilon)L_m}{\mu} + 12\tilde{L}\sqrt{\frac{\beta}{\theta}} + 12\tilde{L}\sqrt{\frac{2\lambda^2(1+\epsilon)\beta}{\mu\theta}}\right) + \left(12L + 12L\sqrt{\frac{2\lambda^2(1+\epsilon)}{\mu}}\right)\cdot\frac{1}{\eta}. \tag{67}$$

We denote $C_1^\gamma = \frac{1}{\gamma_s} + 6L_m + \frac{12\lambda^2(1+\epsilon)L_m}{\mu} + 12\tilde{L}\sqrt{\frac{\beta}{\theta}} + 12\tilde{L}\sqrt{\frac{2\lambda^2(1+\epsilon)\beta}{\mu\theta}}$ and $C_2^\gamma = 12L + 12L\sqrt{\frac{2\lambda^2(1+\epsilon)}{\mu}}$ for convenience. Using Theorem 5.7, we have

$$\frac{1}{4K}\sum_{k=1}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=1}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \le \frac{C_1^\gamma\mathcal{L}^0}{K} + \frac{C_2^\gamma\mathcal{L}^0}{\eta K} + \frac{2\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{4\eta^3\beta\sigma^2}{\theta} + \frac{2\eta\sigma^2}{N}. \tag{68}$$

We choose $\eta = \min\{(\frac{C_2^\gamma N \mathcal{L}^0}{2\sigma^2 K})^{1/2}, (\frac{\theta C_2^\gamma \mathcal{L}^0}{2(1-\alpha)\sigma^2 K})^{1/3}, (\frac{\theta C_2^\gamma \mathcal{L}^0}{4\beta\sigma^2 K})^{1/4}\}$, so that $\frac{C_2^\gamma \mathcal{L}^0}{\eta K} \geq \frac{2\eta\sigma^2}{N}$, $\frac{C_2^\gamma \mathcal{L}^0}{\eta K} \geq \frac{2\eta^2(1-\alpha)\sigma^2}{\theta}$ and $\frac{C_2^\gamma \mathcal{L}^0}{\eta K} \geq \frac{4\eta^3\beta\sigma^2}{\theta}$, and at least one inequality takes equal. Hence, (68) can be bounded as

$$
\begin{aligned}
&\frac{1}{4K}\sum_{k=1}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=1}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \leq \frac{C_1^\gamma \mathcal{L}^0}{K} + 4\frac{C_2^\gamma \mathcal{L}^0}{\eta K} \\
\leq & \frac{C_1^\gamma \mathcal{L}^0}{K} + 4\Big(\frac{2\eta\sigma^2}{N} + \frac{2\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{4\eta^3\beta\sigma^2}{\theta}\Big) \\
\leq & \frac{C_1^\gamma \mathcal{L}^0}{K} + 4\Big(\frac{2\sigma^2 C_2^\gamma \mathcal{L}^0}{NK}\Big)^{\frac{1}{2}} + 4\Big(\frac{2\sigma^2(1-\alpha)}{\theta}\Big)^{\frac{1}{3}}\Big(\frac{C_2^\gamma \mathcal{L}^0}{K}\Big)^{\frac{2}{3}} + 4\Big(\frac{4\beta\sigma^2}{\theta}\Big)^{\frac{1}{4}}\Big(\frac{C_2^\gamma \mathcal{L}^0}{K}\Big)^{\frac{3}{4}},
\end{aligned} \tag{69}
$$

which leads to the conclusion. □

I. Proof of Theorem 5.11

Proof. When $c_1 = 0, c_2 = 1$, the Lyapunov function can be simplified as

$$
\mathcal{L}^k = m(X^k) - m^* + \frac{\gamma}{\eta}P^k. \tag{70}
$$

And Theorem 5.7 leads to

$$
\frac{1}{4K}\sum_{k=1}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=1}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \leq \frac{\mathcal{L}^0}{\gamma K} + \frac{\eta\sigma^2}{N}. \tag{71}
$$

Choosing step size γ as Lemma G.1 and setting $c_1 = 0, c_2 = 1$ leads to

$$
\frac{1}{\gamma} \leq \Big(\frac{1}{\gamma_s} + 6L_m + \frac{12\lambda^2(1+\epsilon)L_m}{\mu}\Big) + \Big(6\sqrt{2}L + 12L\sqrt{\frac{\lambda^2(1+\epsilon)}{\mu}}\Big)\cdot\frac{1}{\eta}. \tag{72}
$$

We denote $C_1^\gamma = \frac{1}{\gamma_s} + 6L_m + \frac{12\lambda^2(1+\epsilon)L_m}{\mu}$ and $C_2^\gamma = 6\sqrt{2}L + 12L\sqrt{\frac{\lambda^2(1+\epsilon)}{\mu}}$ for convenience, and (71) leads to

$$
\frac{1}{4K}\sum_{k=1}^{K-1}\mathbb{E}[\|\mathrm{grad} f(X^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=1}^{K-1}\mathbb{E}[\mathcal{N}(X^k)] \leq \frac{C_1^\gamma \mathcal{L}^0}{K} + \frac{C_2^\gamma \mathcal{L}^0}{\eta K} + \frac{\eta\sigma^2}{N}. \tag{73}
$$

Choosing $\eta = \sqrt{\frac{C_2^\gamma N \mathcal{L}^0}{\sigma^2 K}}$ leads to the conclusion.

□

J. Algorithm for Block-wise Problems

Algorithm 2 Block-wise EF-Landing

```
Require: starting point $\mathfrak{X}^0 \in \mathfrak{R}$; gradient bound $L'$; step size $\gamma > 0$; compressor $\mathcal{C}$; momentum $\eta \in (0, 1]$;
Each node initializes $\mathfrak{v}_i^0 = \nabla F(\mathfrak{X}^0; \xi_i^0)$, $\mathfrak{g}_i^0 = \mathcal{C}(\mathfrak{v}_i^0)$ for $i = 1, \ldots, N$; Master initializes $\mathfrak{g}^0 = \frac{1}{N}\sum_{i=1}^N \mathfrak{g}_i^0$.
for $k = 0, 1, \ldots, K-1$ do
    Master clips the gradient $\tilde{\mathfrak{g}}^k = \min\{1, L'/\|\mathfrak{g}^k\|\}\mathfrak{g}^k$;
    Master computes $\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)$ using (14);
    Master computes $\mathfrak{X}^{k+1} = \mathfrak{X}^k - \gamma\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)$ and broadcasts $\mathfrak{X}^{k+1}$ to all nodes.
    for all nodes $i = 1, \ldots, N$ in parallel do
        Compute momentum $\mathfrak{v}_i^{k+1} = (1-\eta)\mathfrak{v}_k^k + \eta\nabla F(\mathfrak{X}^{k+1}; \xi_i^{k+1})$;
        Compress $\mathfrak{c}_i^k$ via $\mathfrak{c}_i^k = \mathcal{C}(\mathfrak{v}_i^{k+1} - \mathfrak{g}_i^k)$ and send it to the master;
        Update local state $\mathfrak{g}_i^{k+1}$ via $\mathfrak{g}_i^{k+1} = \mathfrak{g}_i^k + \mathfrak{c}_i^k$.
    end for
    Master updates $\mathfrak{g}^{k+1}$ via $\mathfrak{g}^{k+1} = \mathfrak{g}^k + \frac{1}{N}\sum_{i=1}^N \mathfrak{c}_i^k$.
end for
```

K. Convergence Results of EF-Landing in Stochastic Scenarios for Block-wise Constraints

Theorem K.1 (Block-wise EF-Landing Convergence in Stochastic Scenario)**.** *Consider the problem* (13) *with block-wise constraints* (13)*. Letting Assumptions 3.2, 5.1, 5.2, 5.3 and 5.4 hold (in the sense of struct in $\mathfrak{R}$), if we choose step size γ as in Lemma G.1, by running Algorithm 1 for K iterations, we have*

$$\frac{1}{K}\sum_{k=1}^{K-1}\sum_{j=1}^{J}\mathbb{E}[\|\mathrm{grad}_j f(X_j^k)\|_F^2] \leq 4\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big),$$

$$\frac{1}{K}\sum_{k=1}^{K-1}\sum_{j=1}^{J}\mathbb{E}[\mathcal{N}_j(X_j^k)] \leq \frac{2}{\lambda\mu}\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big),$$

$$\frac{1}{K}\sum_{k=1}^{K-1}\mathbb{E}\Big[\Big\|\frac{\partial f}{\partial x^k}\Big\|_2^2\Big] \leq 2\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big),$$

where $\mathcal{O}(\frac{1}{\sqrt{NK}})$ is specified as

$$\mathcal{O}\Big(\frac{1}{\sqrt{NK}}\Big) = \frac{C_1^\gamma\mathcal{L}^0}{K} + 4\Big(\frac{2\sigma^2 C_2^\gamma\mathcal{L}^0}{NK}\Big)^{\frac{1}{2}}$$
$$+4\Big(\frac{2\sigma^2(1-\alpha)}{\theta}\Big)^{\frac{1}{3}}\Big(\frac{C_2^\gamma\mathcal{L}^0}{K}\Big)^{\frac{2}{3}} + 4\Big(\frac{4\beta\sigma^2}{\theta}\Big)^{\frac{1}{4}}\Big(\frac{C_2^\gamma\mathcal{L}^0}{K}\Big)^{\frac{3}{4}},$$

when choosing $\eta = \min\{(\frac{C_2^\gamma N\mathcal{L}^0}{2\sigma^2 K})^{1/2}, (\frac{\theta C_2^\gamma\mathcal{L}^0}{2(1-\alpha)\sigma^2 K})^{1/3}, (\frac{\theta C_2^\gamma\mathcal{L}^0}{4\beta\sigma^2 K})^{1/4}\}$. $\mathcal{L}^0$ is defined in (12)*, θ and β are defined in Theorem 5.7, C_1^γ, C_2^γ are two constants defined as $C_1^\gamma = \frac{1}{\gamma_s} + 6L_m + \frac{12\lambda^2(1+\epsilon)L_m}{\mu} + 12\tilde{L}\sqrt{\frac{\beta}{\theta}} + 12\tilde{L}\sqrt{\frac{2\lambda^2(1+\epsilon)\beta}{\mu\theta}}$ and $C_2^\gamma = 12L + 12L\sqrt{\frac{2\lambda^2(1+\epsilon)}{\mu}}$.*

Proof. We choose the merit function as:

$$m(\mathfrak{X}) = f(\mathfrak{X}) - \sum_{j=1}^{J} h_j(X_j) + \sum_{j=1}^{J}\mu\mathcal{N}_j(X_j), \tag{74}$$

where $h_j(X_j) = \dfrac{1}{2}\Big\langle \mathrm{sym}\Big(X_j^\top\dfrac{\partial f}{\partial X_j}\Big), X_j^\top X_j - I_{p_j}\Big\rangle$ and μ is a hyper-parameter specified later.

Noticing that partial functions like $h_j(X_j)$ satisfies $\frac{\partial h_j}{\partial X_k} = 0,\ \forall j \neq k$, we have

$$\langle \tilde{\Lambda}(\mathfrak{X}; \mathfrak{g}), \nabla m(\mathfrak{X}) \rangle = \sum_{j=1}^{J} \Big\langle \mathrm{skew}(\boldsymbol{g}_j X_j^\top) X_j + \lambda X_j (X_j^\top X_j - I_p), \frac{\partial f}{\partial X_j} - \nabla h_j(X_j) + \nabla \mu \mathcal{N}_j(X_j) \Big\rangle + \Big\langle \boldsymbol{g}_0, \frac{\partial f}{\partial x} \Big\rangle.$$

Noticing that for all $j = 1, \dots, J$, $\frac{\partial f}{\partial X_j}$ is L-Lipschitz smooth as long as $\nabla f(\mathfrak{X})$ is L-Lipschitz smooth, and $\frac{\partial f}{\partial X_j}$ is L'-bounded as long as $\nabla f(\mathfrak{X})$ is L'-bounded, using Lemma 5.6 with the same choice of μ, we have

$$\begin{aligned} \langle \tilde{\Lambda}(\mathfrak{X}; \mathfrak{g}), \nabla m(\mathfrak{X}) \rangle \geq & \frac{1}{4} \sum_{j=1}^{J} \Big\| \mathrm{skew}\Big(\frac{\partial f}{\partial X_j} X_j^\top \Big) X_j \Big\|_F^2 + \frac{1}{4} \sum_{j=1}^{J} \Big\| \mathrm{skew}\Big(\boldsymbol{g}_j X_j^\top \Big) X_j^\top \Big\|_F^2 - \sum_{j=1}^{J} \Big\| \boldsymbol{g}_j - \frac{\partial f}{\partial X_j} \Big\|_F^2 \\ & + \sum_{j=1}^{J} \lambda \mu \mathcal{N}(X_j) + \frac{1}{2} \|\boldsymbol{g}_0\|_2^2 + \frac{1}{2} \Big\| \frac{\partial f}{\partial x} \Big\|_2^2 - \Big\| \boldsymbol{g}_0 - \frac{\partial f}{\partial x} \Big\|_2^2, \end{aligned} \tag{75}$$

where for $\langle \boldsymbol{g}_0, \frac{\partial f}{\partial x} \rangle$ we use $\langle a, b \rangle = \frac{1}{2}\|a\|_2^2 + \frac{1}{2}\|b\|_2^2 - \frac{1}{2}\|a - b\|_2^2 \geq \frac{1}{2}\|a\|_2^2 + \frac{1}{2}\|b\|_2^2 - \|a - b\|_2^2$. The inequality is for the sake of $\|\mathfrak{g} - \nabla f(\mathfrak{X})\|^2 = \sum_{j=1}^{J} \|\boldsymbol{g}_j - \frac{\partial f}{\partial X_j}\|_F^2 + \|\boldsymbol{g}_0 - \frac{\partial f}{\partial x}\|_2^2$. Hence, using smoothness of $m(\mathfrak{X}^k)$ and the merit function bound (75), we have

$$\begin{aligned} m(\mathfrak{X}^{k+1}) \leq & m(\mathfrak{X}^k) + \langle m(\mathfrak{X}^k), \mathfrak{X}^{k+1} - \mathfrak{X}^k \rangle + \frac{L_m}{2} \|\mathfrak{X}^{k+1} - \mathfrak{X}^k\|^2 \\ = & m(\mathfrak{X}^k) - \gamma \langle \nabla m(\mathfrak{X}^k), \tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k \rangle + \frac{L_m \gamma^2}{2} \|\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)\|^2 \\ \leq & m(\mathfrak{X}^k) - \frac{\gamma}{4} \sum_{j=1}^{J} \|\mathrm{grad}_j f(X_j^k)\|_F^2 - \frac{\gamma}{4} \sum_{j=1}^{J} \|\mathrm{skew}\big(\tilde{\boldsymbol{g}}_j^k (X_j^k)^\top \big) X_j^k\|_F^2 - \gamma \lambda \mu \sum_{j=1}^{J} \mathcal{N}_j(X_j^k) \\ & + \frac{L_m \gamma^2}{2} \|\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)\|^2 - \frac{\gamma}{2} \|\tilde{\boldsymbol{g}}_0^k\|_2^2 - \frac{\gamma}{2} \Big\| \frac{\partial f}{\partial x^k} \Big\|_2^2 + \gamma \|\tilde{\mathfrak{g}}^k - \nabla f(\mathfrak{X}^k)\|^2 \end{aligned} \tag{76}$$

Denote $S^k = \sum_{j=1}^{J} \|\mathrm{skew}(\tilde{\boldsymbol{g}}_j^k (X_j^k)^\top) X_j^k\|_F^2$, following the same process as the proof of Theorem 5.7, we have

$$\begin{aligned} & \mathbb{E}[m(\mathfrak{X}^{k+1})] - m^* + \frac{c_1 \gamma}{\theta} \mathbb{E}[\tilde{G}^{k+1}] + \frac{2 c_1 \gamma \eta \beta}{\theta} \mathbb{E}[\tilde{P}^{k+1}] + \frac{c_2 \gamma}{\eta} \mathbb{E}[P^{k+1}] \\ \leq & \mathbb{E}[m(\mathfrak{X}^k)] - m^* + \frac{c_1 \gamma}{\theta} \mathbb{E}[\tilde{G}^k] + \frac{2 c_1 \gamma \eta \beta}{\theta} \mathbb{E}[\tilde{P}^k] + \frac{c_2 \gamma}{\eta} \mathbb{E}[P^k] \\ & - \frac{\gamma}{4} \sum_{j=1}^{J} \mathbb{E}[\|\mathrm{grad}_j f(X_j^k)\|_F^2] - \frac{\gamma}{4} \mathbb{E}[S^k] - \gamma \lambda \mu \sum_{j=1}^{J} \mathbb{E}[\mathcal{N}_j(X_j^k)] - \frac{\gamma}{2} \mathbb{E}\Big[\Big\| \frac{\partial f}{\partial x^k} \Big\|_2^2 \Big] - \frac{\gamma}{2} \mathbb{E}[\|\tilde{\boldsymbol{g}}_0^k\|_2^2] \\ & + \Big(\frac{2 c_1 \beta \tilde{L}^2}{\theta} (1 - \eta + \eta^3) \gamma^3 + \frac{c_2 L^2}{\eta^2} (1 - \eta)^2 (1 + \eta) \gamma^3 + \frac{L_m \gamma^2}{2} \Big) \mathbb{E}[\|\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)\|^2] \\ & + \frac{c_1 \gamma \eta^2 (1 - \alpha) \sigma^2}{\theta} + \frac{2 c_1 \gamma \eta^3 \beta \sigma^2}{\theta} + \frac{c_2 \gamma \eta \sigma^2}{N}. \end{aligned} \tag{77}$$

Noticing that $\|\tilde{\Lambda}(\mathfrak{X}^k; \tilde{\mathfrak{g}}^k)\|^2 \leq \sum_{j=1}^{J} \|\mathrm{skew}(\tilde{\boldsymbol{g}}_j (X_j^k)^\top) X_j^k\|_F^2 + 4\lambda^2 (1 + \epsilon) \sum_{j=1}^{J} \mathcal{N}_j(X_j^k) + \|\tilde{\boldsymbol{g}}_0\|_2^2$, if we choose $\gamma \leq \gamma_1$ defined in Theorem 5.7, the following inequality also holds, so that the influence of $\|\tilde{\boldsymbol{g}}_0\|_2^2$ can be eliminated.

$$-\frac{\gamma}{2} + \frac{2 c_1 \beta \tilde{L}^2}{\theta} (1 - \eta + \eta^3) \gamma^3 + \frac{c_2 L^2}{\eta^2} (1 - \eta)^2 (1 + \eta) \gamma^3 + \frac{L_m \gamma^2}{2} \leq 0. \tag{78}$$

Hence, we don't need any additional restriction for γ apart from $\gamma_s, \gamma_1, \gamma_2$ in Theorem 5.7. With the step size above, we have

$$\begin{aligned} \mathbb{E}[\mathcal{L}^{k+1}] \leq & \mathbb{E}[\mathcal{L}^k] - \frac{\gamma}{4} \sum_{j=1}^{J} \mathbb{E}[\|\mathrm{grad}_j f(X_j^k)\|_F^2] - \frac{\gamma \lambda \mu}{2} \sum_{j=1}^{J} \mathbb{E}[\mathcal{N}_j(X_j^k)] + \frac{\gamma}{2} \mathbb{E}\Big[\Big\| \frac{\partial f}{\partial x^k} \Big\|_2^2 \Big] \\ & + \frac{c_1 \gamma \eta^2 (1 - \alpha) \sigma^2}{\theta} + \frac{2 c_1 \gamma \eta^3 \beta \sigma^2}{\theta} + \frac{c_2 \gamma \eta \sigma^2}{N}. \end{aligned} \tag{79}$$

Summing up all inequalities from $k = 0$ to $K - 1$ and dividing by K, we have

$$\begin{aligned}
&\frac{1}{4K}\sum_{k=0}^{K-1}\sum_{j=1}^{J}\mathbb{E}[\|\mathrm{grad} f(X_j^k)\|_F^2] + \frac{\lambda\mu}{2K}\sum_{k=0}^{K-1}\sum_{j=0}^{J}\mathbb{E}[\mathcal{N}_j(X_j^k)] + \frac{1}{2K}\sum_{k=0}^{K-1}\left\|\frac{\partial f}{\partial x^k}\right\|_2^2 \\
\leq &\frac{\mathcal{L}^0}{\gamma K} + \frac{c_1\eta^2(1-\alpha)\sigma^2}{\theta} + \frac{2c_1\eta^3\beta\sigma^2}{\theta} + \frac{c_2\eta\sigma^2}{N}.
\end{aligned} \tag{80}$$

Follow the same process of proof of Theorem 5.10, we achieve the final result.

□

L. Further Numerical Experiments

L.1. Online PCA

In the experiment, the matrix $A \in \mathbb{R}^{N\times n}$ is synthetically generated, where $N = 20,000$ represents the number of samples, each of dimension $n = 5000$. The columns of A are independently sampled from the normal distribution $\mathcal{N}(0, UU^T + \sigma I_n)$, with $\sigma = 0.1$, and $U \in \mathbb{R}^{n\times p}$ is sampled from the Stiefel manifold using the uniform Haar distribution.

We conducted the experiment in deterministic scenarios and distributed the data across 4 nodes. The EF-Landing algorithm was tested for different values of p (100, 200, 500, and 1000). The results were compared with the QR retraction method, Euclidean gradient descent with ℓ_2 regularization (referred to as the Penalty method), and the Landing algorithm. The compression methods used included Top-K and QSGD(Alistarh et al., 2017). For the Top-K compression, the compression rate was set to 0.1, and for QSGD, the quantization level s was set to 8 for $p = 100$ and $p = 200$, and 16 for $p = 500$ and $p = 1000$. The experiment involved 600 iterations for all algorithms with a fixed learning rate, except for the QSGD algorithm when $p = 1000$, where a learning rate decay was applied during the iterations.

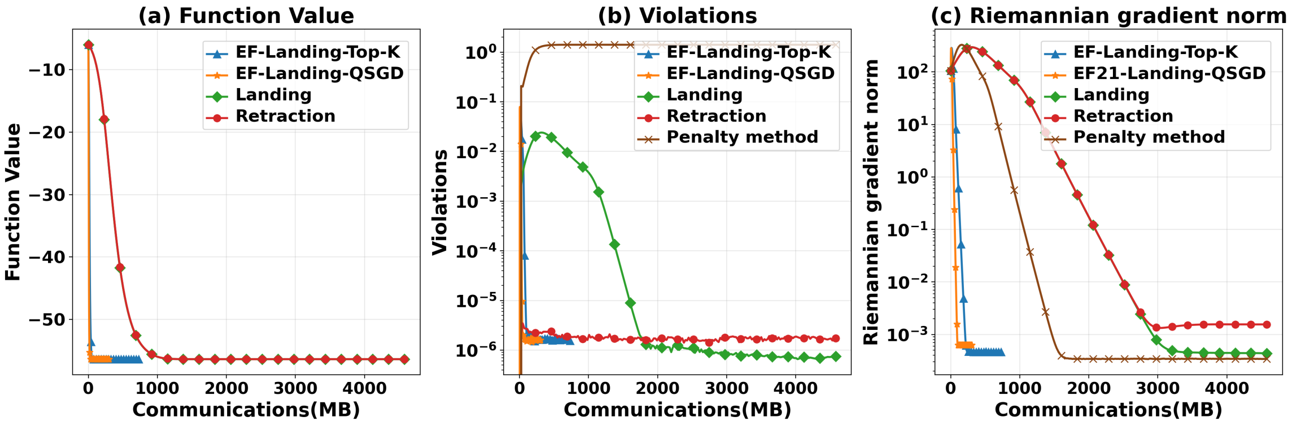

Figure 4: Performance of EF-Landing and other algorithms on online PCA, with $p = 100$

The corresponding experimental results for parameter p taking values of 100, 200, and 500 are shown in Figures 4, 5, and 6, respectively. We can observe that, regardless of the compression algorithm, EF-Landing consistently reduces the communication volume by at least half while still converging to the optimal value of the function and ensuring that the norm of Riemannian gradient approaches 0. Additionally, compared to the Landing algorithm, the EF-Landing algorithm requires less communication to satisfy the constraints on the variable X, specifically ensuring that $\mathcal{N}(X)$ is sufficiently small, as demonstrated in Figures 5 and 6.

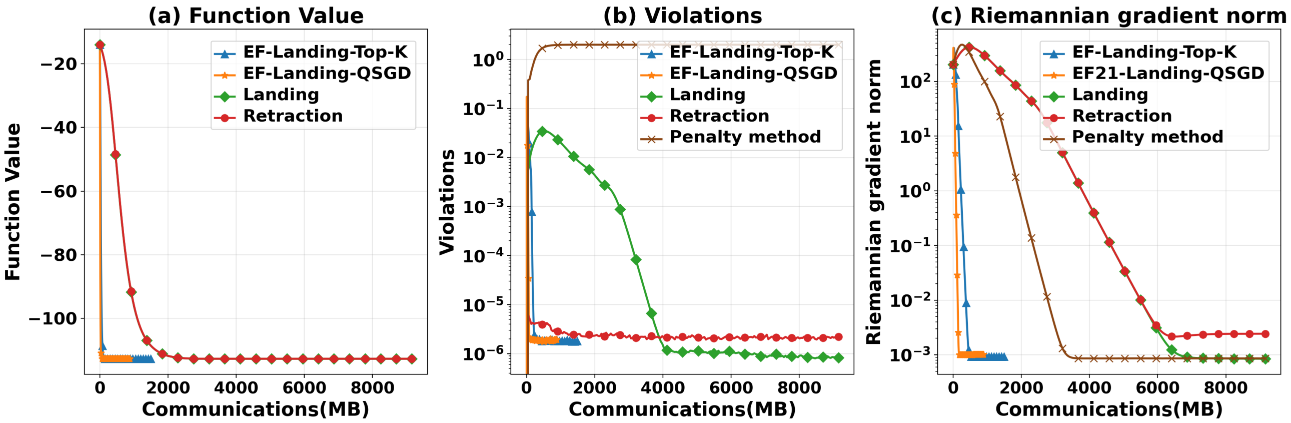

Figure 5: Performance of EF-Landing and other algorithms on online PCA, with $p = 200$

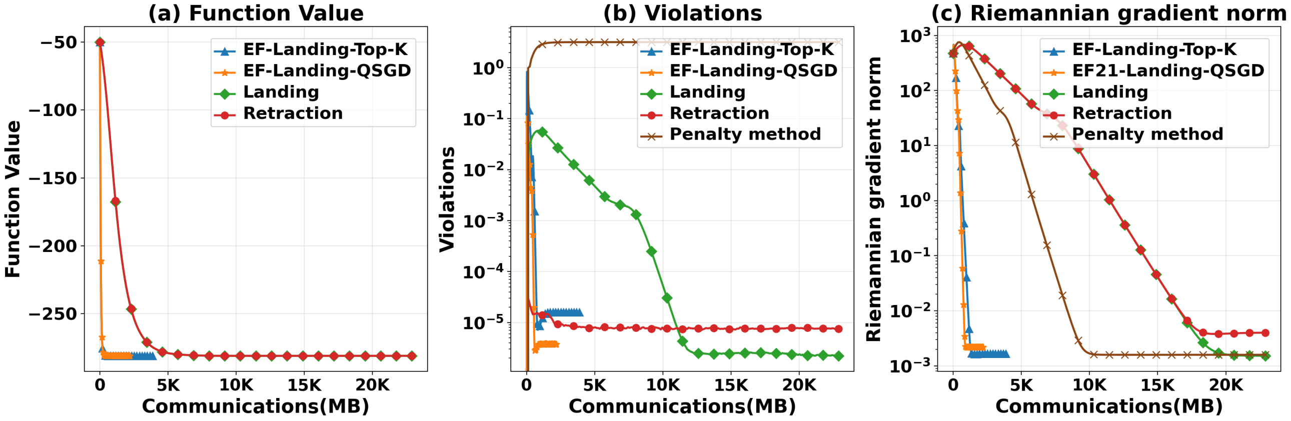

Figure 6: Performance of EF-Landing and other algorithms on online PCA, with $p = 500$

Compared to the QR retraction method, EF-Landing reduces a significant amount of computational effort by eliminating the need for QR decomposition, making the algorithm more efficient. On the other hand, the Penalty method is highly sensitive to the choice of λ. If λ is too small, the parameters may fail to satisfy the constraint, as shown in Figures 1 and 4 to 6. If λ is too large, the Riemannian gradient may fail to converge to zero and can lead to numerical instability. Although the Landing method can more accurately satisfy the orthogonality constraints, it requires several times more communication. By contrast, the EF-Landing algorithm remains highly efficient.

L.2. Neural Networks with Orthogonality Constraints

We also tested the performance of the EF-Landing algorithm on neural network models with orthogonal constraints applied to the convolutional layers. Orthogonal constraints are playing an increasingly important role in deep learning. For example, they can ensure the stability of gradient magnitudes during training (Arjovsky et al., 2016; Saxe et al., 2014), preventing issues like gradient explosion or vanishing. Furthermore, there has been growing attention on how to design orthogonal convolutions(Singla & Feizi, 2021; Boissin et al., 2025; Yu et al., 2022).

We conducted experiments on VGG16 and ResNet18, comparing the performance of the EF-Landing algorithm with the Landing and QR retraction algorithms. Specifically, we applied orthogonal constraints to the convolutional layers by reshaping the convolutional kernels to the size $n_{\text{out}} \times n_{\text{in}} n_x n_y$, where n_{in} and n_{out} represent the number of input and output channels, respectively, and n_x and n_y are the filter dimensions(Ablin et al., 2024). In cases where the reshaped matrix becomes wide instead of tall, we enforce orthogonality on its transpose.It is worth noting that the Block-wise Constraints formulation in (13) better suits this problem, as we impose constraints on each individual convolutional layer rather than on the entire neural network parameters X.

L.2.1. VGG16 ON MNIST

In this experiment, we use the built-in VGG16 model from PyTorch and initialize the convolutional layers using QR decomposition to enforce orthogonal constraints. To match the model architecture, we change the shape of the data to 224×224. The compression methods applied are Top-K and Rand-K, with a compression ratio of 0.2. The training data is uniformly distributed across 8 nodes. The initial learning rate γ is set to 0.01, which decays by a factor of 1/10 every 50 epochs. The batch size is set to 128, and the model is trained for a total of 150 epochs.

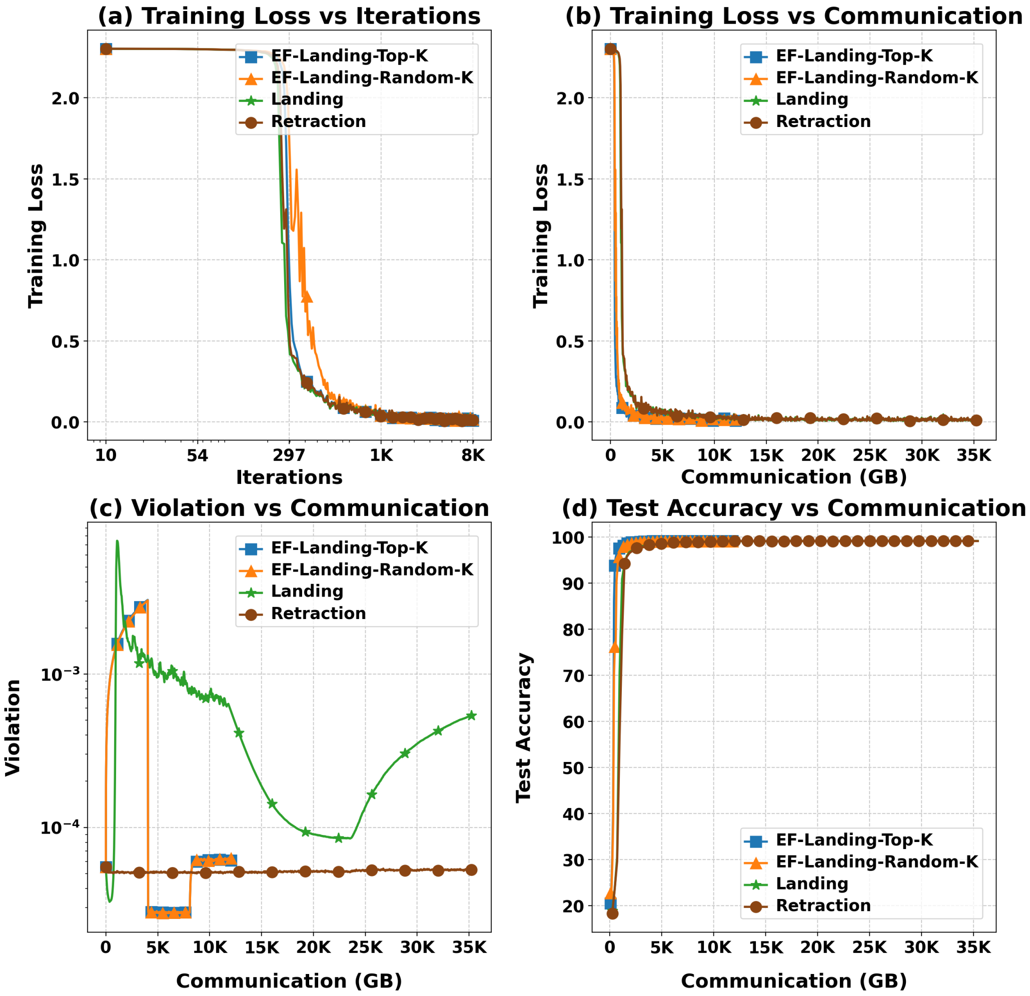

Figure 7: Performance of EF-Landing and other algorithms on deep learning with VGG16 on MNIST.

Figure 7 demonstrates that the EF-Landing algorithm achieves convergence of the training loss function and test accuracy to a steady state with less communication volume, and the accuracy on the test set stabilizes around 99%.In addition, the EF-Landing algorithm outperforms the Landing algorithm in satisfying the orthogonality constraint under this experiment, which also highlights the advantage of the EF-Landing algorithm to some extent.

L.2.2. VGG16 AND RESNET18 ON CIFAR-10

Additionally, we tested the performance of EF-Landing on the CIFAR-10 dataset in a single-node setting, using the Top-K and Rand-K compressors with a compression ratio of 0.2. We compared the results with the Landing and QR retraction methods. In the experiment, each algorithm was trained for 80 epochs, and after the 50th epoch, the learning rate was reduced to $\frac{1}{10}$ of its original value, with λ set to 1.0.

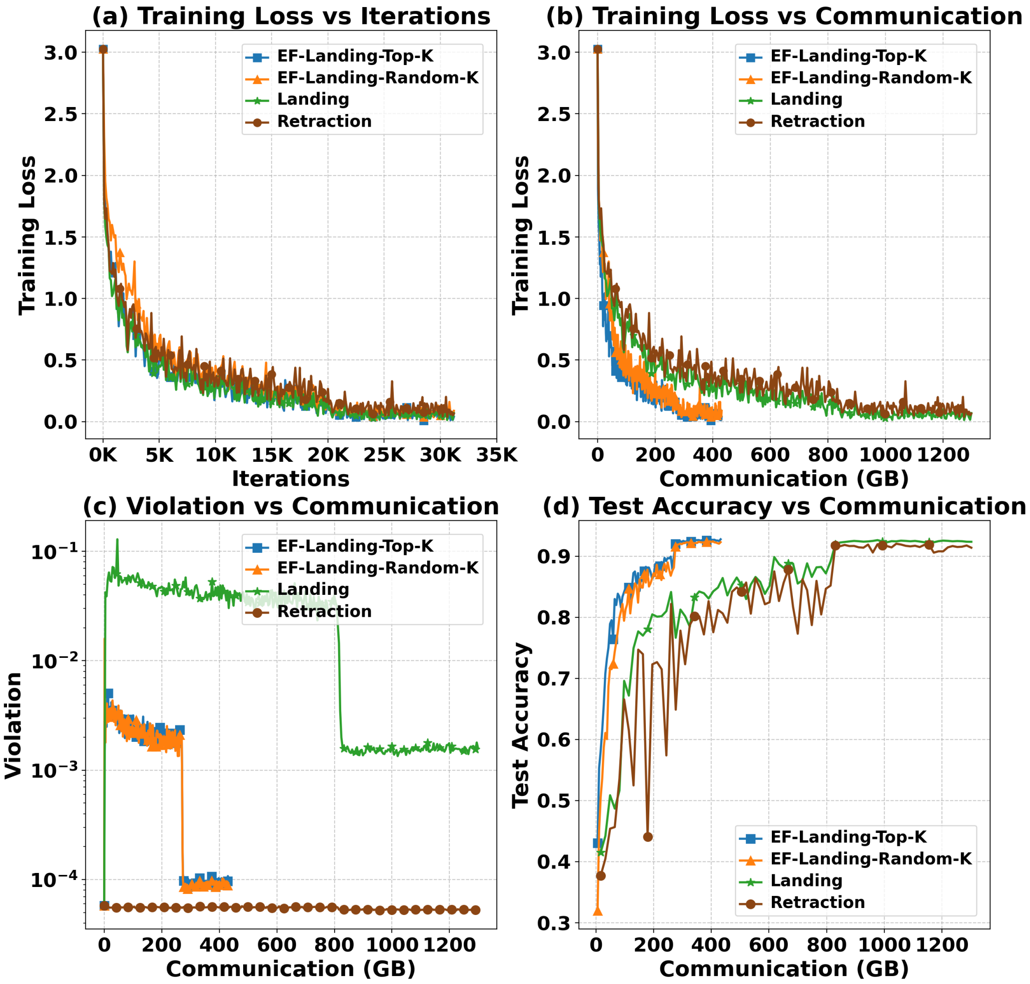

Figure 8: Performance of EF-Landing and other algorithms on deep learning with ResNet18 on CIFAR-10.

As shown in Figures 2 and 8, in terms of satisfying the constraint conditions, the EF-Landing algorithm not only reduces $\mathcal{N}(X)$ with less communication overhead compared to the Landing algorithm, but also satisfies the constraints more accurately . Furthermore, in terms of computational cost, the EF-Landing algorithm offers a clear advantage over the QR retraction method, being free of QR decomposition calculations.