Tackling Data Heterogeneity: A New Unified Framework for Decentralized SGD with Sample-induced Topology

Yan Huang [1] **Ying Sun** [2] **Zehan Zhu** [1] **Changzhi Yan** [1] **Jinming Xu** [1]

Abstract

We develop a general framework unifying several gradient-based stochastic optimization methods for empirical risk minimization problems both in centralized and distributed scenarios. The framework hinges on the introduction of an augmented graph consisting of nodes modeling the samples and edges modeling both the inter-device communication and intra-device stochastic gradient computation. By designing properly the topology of the augmented graph, we are able to recover as special cases the renowned Local-SGD and DSGD algorithms, and provide a unified perspective for variance-reduction (VR) and gradient-tracking (GT) methods such as SAGA, Local-SVRG and GT-SAGA. We also provide a unified convergence analysis for smooth and (strongly) convex objectives relying on a proper structured Lyapunov function, and the obtained rate can recover the best known results for many existing algorithms. The rate results further reveal that VR and GT methods can effectively eliminate data heterogeneity within and across devices, respectively, enabling the exact convergence of the algorithm to the optimal solution. Numerical experiments confirm the findings in this paper.

1. Introduction

With the increasing popularity of large-scale machine learning, distributed stochastic optimization methods have sparked considerable interest to improve learning efficiency in both academia and industry (Lian et al., 2017; Boyd et al., 2011). In contrast to the typical centralized/parameter-server architecture (Dean et al., 2012; Lian et al., 2015; Stich, 2019) where a center node coordinates the entire optimization process, which usually becomes the bottleneck, distributed structure has its unique advantage in improving computation and communication efficiency (Nedić et al., 2018). Besides, since the data is locally maintained by each node, data privacy can be well preserved for data-sensitive application domains (Li et al., 2019).

We consider the prototypical empirical risk minimization problem collaboratively solved by a set of agents over a communication network. The overall objective of the agents is to seek an optimal solution $x^* \in \mathbb{R}^d$ that solves the following finite-sum problem:

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{n}\sum_{i=1}^{n}\left(f_i(x) := \frac{1}{m}\sum_{j=1}^{m}\underbrace{f(x,\xi_{i,j})}_{f_{ij}(x)}\right), \quad (1)$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ is the local private loss function accessible only by the associated node $i \in \mathcal{N} := \{1, 2, \cdots, n\}$, and $\{\xi_{i,j}\}_{j=1}^m$ denote the data samples locally stored at node $i \in \mathcal{N}$ and $\{f_{ij}\}$ denote the corresponding loss functions.

Problem (1) has been extensively studied over the last decade and enormous distributed algorithms, e.g., Nedic and Ozdaglar (2009); Nedic et al. (2010); Lobel and Ozdaglar (2011); Yuan et al. (2016); Pu et al. (2020), have been proposed to solve this problem. The readers are referred to the recent survey paper (Nedić et al., 2018) and the references therein. Among these algorithms, the distributed gradient decent (DGD) algorithm (Nedic and Ozdaglar, 2009) is a simple yet effective method. However, it suffers from steady-state error when employing constant stepsizes due to the fact that the fixed point of DGD is inherently not consensual (Yuan et al., 2016). To bridge the gap between centralized gradient decent and DGD, a gradient-tracking scheme is introduced to overcome the above issue (Xu et al., 2015; Di Lorenzo and Scutari, 2016; Nedic et al., 2017; Qu and Li, 2017), which introduces an auxiliary variable to estimate the full gradient ∇f (the sum of local gradients) leveraging dynamic average consensus (Zhu and Martínez, 2010) so as to make the fixed point consensual. The introduction of gradient tracking scheme allow us to effectively account for the heterogeneity among local data sets (also known as *external* variance among data distribution of nodes).

[1]College of Control Science and Engineering, Zhejiang University, Hangzhou, China [2]School of Electrical Engineering and Computer Science, The Pennsylvania State University, PA 16802, USA. Correspondence to: Jinming Xu <jimmyxu@zju.edu.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Variance-reduced (VR) and local methods. All of the above-mentioned algorithms are deterministic and require evaluating the local full gradient ∇f_i at each iteration, leading to high computational complexity (Hong et al., 2017). A natural way to reduce the computational complexity is to use stochastic gradients to approximate the local full gradient. To this end, a vast number of distributed stochastic optimization algorithms are proposed for the problem (1), such as DSGD (Ram et al., 2009), D-PSGD (Lian et al., 2017) and SGP (Assran et al., 2019), and just to name a few. These stochastic optimization algorithms work quite well in practice but usually require diminishing stepsizes to attenuate the variance of the stochastic gradient, a.k.a. *internal* (sample) variance. An effective solution to this is to employ the idea of variance-reduction and learn the local full gradient ∇f_i iteratively, as did in SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014), L-SVRG (Hofmann et al., 2015; Qian et al., 2021) and SARAH (Nguyen et al., 2017) to eliminate the internal variance, yielding faster convergence such as D-SAGA (Calauzenes and Roux, 2017). To avoid high communication burden among nodes, one may trade computation with communication by performing several local gradient steps between two consecutive communication steps, which leads to communication-efficient algorithms, such as Local-SGD (Khaled et al., 2020) and Local-SVRG (Gorbunov et al., 2021); or employing periodic global averaging (PGA) for speeding up consensus, such as Gossip-PGA (Chen et al., 2021).

Gradient-tracking-based (GT) stochastic methods. The gradient-tracking scheme have been also recently incorporated into various stochastic optimization algorithms as a key step to eliminate the *external* variance. For example, DSGT (Pu and Nedić, 2020) and DSA (Mokhtari and Ribeiro, 2016) can achieve higher accuracy than that of DSGD by removing the bias due to the heterogeneity among local data sets. Nevertheless, gradient-tracking, by its nature, is still unable to eliminate the data sample variance. This naturally leads to the integration of variance-reduction methods with the gradient-tracking scheme. For instance, Sun et al. (2020) employ a scheme with both gradient-tracking and variance-reduction to solve a smooth (probably non-convex) problem and show that it converges to a stationary point sublinearly. Li et al. (2021) proposed a similar algorithm with a nested loop structure for the sake of improving its overall complexity. Xin et al. (2020) and Jiang et al. (2022) consider a similar GT-VR framework and obtain a linear rate for strongly convex problems and $\mathcal{O}\left(1/k\right)$ rate for non-convex setting, respectively. Similar attempts have been recently made towards composite optimization problems (Ye et al., 2020). There are also many other efforts made to solve Problem (1), such as those based on approximate Newton-type methods (Li et al., 2020) and acceleration schemes (Scaman et al., 2017; Hendrikx et al., 2021).

Unified framework for first-order stochastic optimization algorithms. There have been some efforts made to unify the aforementioned algorithms. In particular, Hu et al. (2017) unify several variance-reduction methods by establishing the intrinsic connection between stochastic optimization methods and dynamic jump systems. Wang and Joshi (2021) consider to use a time-varying mixing matrix to model Cooperative SGD which can recover several existing non-variance-reduced methods. Building on this, Koloskova et al. (2020) propose a unified framework for Decentralized (Gossip) SGD by employing changing topology and multiple local updates. To incorporate variance-reduction methods, Gorbunov et al. (2020) study a general framework that can account for variance-reduction, importance sampling, mini-batch sampling, leading to a unified theory of variance reduced and non-variance-reduced SGD methods. The authors also extend this framework to cover Local-SGD and variance-reduced SGD methods, which recovers the rate in (Koloskova et al., 2020). However, to the best of our knowledge, there is no such a framework for distributed first-order stochastic optimization algorithms that can recover all the above-mentioned VR-, GT-, Local- and PGA-based methods both in centralized and decentralized settings.

1.1. Our contribution

In this work, we develop a new unified framework based on the introduction of an augmented graph whose nodes model the data samples. Leveraging a proper sampling strategy on the augmented graph, this framework allows us to recover many existing algorithms as well as their corresponding best known rates. In contrast to the existing frameworks as mentioned above, the proposed framework not only contain them as special cases but also enable us to easily design new efficient algorithms with guaranteed rates, especially those employing gradient-tracking scheme, yielding a broader range of methods to be incorporated. The main contributions of this paper are summarized as follows:

- **New unified framework for algorithm design and analysis.** The proposed framework unify various gradient-based stochastic optimization methods both in centralized and distributed scenarios. With proper sampling strategies on the augmented graph, we can easily recover these VR-, GT-, Local- and PGA-based methods, as well as their proper combinations (see Table 1). Besides, this framework also provides a new unifying perspective for GT- and VR-based schemes, which are otherwise two separate approaches before, and show an equivalence of Local-SGD, Gossip-PGA and DSDG in terms of iteration complexity once their expected topology connectivity is same.

- **Recovering various existing algorithms along with the best known rates.** A unified convergence analysis

is provided, which relies on a proper Lyapunov function for smooth and (strongly) convex objectives. The obtained rates either recover the existing best known rates or are new for certain algorithms under our settings, including SAGA, Local-SGD, DSGD, Local-SVRG and GT-SAGA (see Table 1). These rates also show the clear dependence of the convergence performance on the above-mentioned schemes, such as GT, VR, Local update, and PGA. The theoretical results further reveal that VR- and GT-based methods are usually needed to achieve exact convergence in scenarios where data heterogeneity is a key concern.

- **New efficient algorithms with provable rates.** Our framework allows us to easily come up with new efficient algorithms with proper design of the sampling strategy on the augmented graph and provides the corresponding rate guarantee as well, such as Local-SAGA, PGA-SAGA, PGA-GT-SAGA, which are not formally analyzed before (see Table 1). Moreover, the proposed framework provides more flexibility in design of network topology which can be of multi-layer structure in certain scenarios for communication efficiency.

2. Problem Formulation

We consider solving Problem (1) over a peer-to-peer network, modeled as an augmented directed graph (digraph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} := \{1, 2, ..., M\}$ with $M = nm$ denotes the set of nodes modeling the data samples and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges consisting of ordered pairs (i, j) modeling the virtual/actual communication link from j to i. We then make the following blanket assumptions on the cost functions of problem (1).

Assumption 1. *Each $f_i : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and L-smooth, i.e., for any $x, x' \in \mathbb{R}^d$*

$$\langle \nabla f_i(x) - \nabla f_i(x'), x - x' \rangle \geqslant \mu \|x - x'\|^2, \quad (2)$$

$$\|\nabla f_i(x) - \nabla f_i(x')\| \leqslant L \|x - x'\|. \quad (3)$$

Assumption 2. *(Bounded data heterogeneity at optimum) Let $x^* \in \arg\min_x f(x)$. For each $f_{ij} : \mathbb{R}^d \to \mathbb{R}$, $i \in [n], j \in [m]$, there exist positive constants σ^*, ζ^* such that*

$$\frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} \|\nabla f_{ij}(x^*) - \nabla f_i(x^*)\|^2 \leqslant \sigma^*, \quad (4)$$

$$\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(x^*)\|^2 \leqslant \zeta^*. \quad (5)$$

Remark 1. *The parameter σ^*, ζ^* are defined to measure the local gradient sampling variance and data heterogeneity across devices, respectively. Notice that we do not require the data heterogeneity be bounded uniformly for all x but only at the optimum x^*, which is weaker than the previous works such as (Lian et al., 2017; Wang and Joshi, 2021).*

Assumption 3. *(Averaged smoothness) For all $i \in [n]$ and $\forall x, x' \in \mathbb{R}^d$, we have*

$$\frac{1}{m} \sum_{j=1}^{m} \|\nabla f_{ij}(x) - \nabla f_{ij}(x')\|^2 \leqslant 2L \left(f_i(x) - f_i(x') - \langle \nabla f_i(x'), x - x' \rangle \right). \quad (6)$$

Assumption 3 is automatically satisfied if we assume that each f_{ij} is convex and L-smooth.

3. Sample-wise Push-Pull Framework

In this section, we introduce the sample-wise Push-Pull framework for the finite-sum problem (1). To this end, we use an augmented graph $\mathcal{G}$ with a two-level structure as depicted in Figure 1 to illustrate the key ideas.

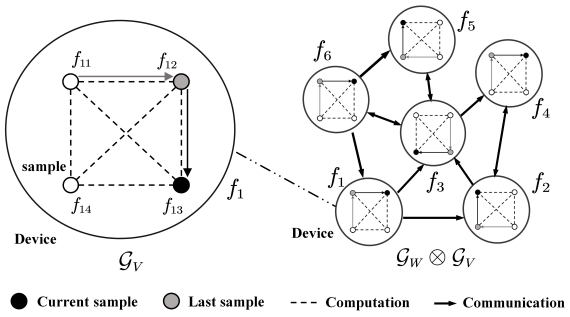

Figure 1. A two-level augmented graph for $n = 6$ and $m = 4$. The graph $\mathcal{G}_V$ on the left refers to a fully connected underlying graph within one device, and the solid arrows represent the *local* stochastic gradient computation for updating parameters. The graph $\mathcal{G} = \mathcal{G}_W \otimes \mathcal{G}_V$ on the right represents the augmented graph induced by the samples locally stored at all the devices.

We first model each device as a super-node connected by the actual communication network $\mathcal{G}_W$. Then, we replace each device i with a virtual fully connected graph $\mathcal{G}_V$ whose nodes model the samples $\xi_{i,j}$, yielding an augmented graph $\mathcal{G} = \mathcal{G}_W \otimes \mathcal{G}_V$. As we will show shortly, implementation of stochastic optimization algorithms can be deemed as recursively sampling the edges of $\mathcal{G}$, which consists in: **i)** locally updating the parameter of sample node j in device i using the gradient of f_{il} when the edge $(j, l) \in \mathcal{G}_V$ associated with device i; **ii)** performing an actual communication step between devices i and s when the edge $(i, s) \in \mathcal{G}_W$.

As a convention in decentralized optimization, we associate each node in $\mathcal{G}$ a local variable $x_i \in \mathbb{R}^d$ that serves as a copy of the global variable x. Besides, each node i also maintains an auxiliary variable $y_i \in \mathbb{R}^d$ that estimates the full gradient $\nabla f(x)$. For brevity, we use $X, Y \in \mathbb{R}^{M \times d}$ to denote matrices stacking the x_i's and y_i's, respectively:

$$X := [x_1, x_2, \cdots, x_M]^T, \; Y := [y_1, y_2, \cdots, y_M]^T.$$

Accordingly, the collective gradient vectors of all local objective functions at X is denoted as

$$\nabla F(X) := \left[\nabla f_1(x_1), \cdots, \nabla f_M(x_M)\right]^T.$$

Now, we are ready to introduce our algorithmic framework, termed sample-wise Push-Pull (SPP), for Problem (1):

$$X_{k+1} = R_k X_k - \alpha \Gamma_k Y_k, \tag{7a}$$
$$Y_{k+1} = C_k Y_k + \nabla F(X_{k+1}) - \nabla F(X_k), \tag{7b}$$

where α is a constant step-size, and $R_k, C_k \in \mathbb{R}^{M\times M}$ are time-varying weight matrices (to be properly designed) for reaching consensus (pull operation) among nodes and tracking the overall average gradient (push operation), respectively; $\Gamma_k \in \mathbb{R}^{M\times M}$ is a random sampling matrix for selecting edges of the augmented graph.

In what follows, we explain the role of the above three key parameters of Γ_k, R_k and C_k over an augment graph with the two-level structure as shown in Figure 1.

Sampling on augmented graph. In our proposed SPP framework, both the processes of actual communication among devices and local stochastic gradient computation for updating parameters can be regarded as selecting a subset of nodes from the augmented graph $\mathcal{G} = \mathcal{G}_W \otimes \mathcal{G}_V$. In particular, each device i elects its samples (nodes in $\mathcal{G}_V$) participating in the update at iteration k, indicated by a binary-valued vector $\boldsymbol{e}_{i,k} \in \{0,1\}^{m\times 1}$: the j-th element $e_{i,k}^j$ is set to be one if the ξ_{ij} is selected and zero otherwise. The concatenated vector $\boldsymbol{e}_k = [\boldsymbol{e}_{1,k}^T, \cdots, \boldsymbol{e}_{n,k}^T]^T \in \mathbb{R}^M$ then indicates the identities of all the samples in the network selected at iteration k. Denote by b_k the number of selected samples of each device at iteration k, i.e., $b_k = \mathbf{1}_m^T e_{i,k}, \forall i \in [n]$. Inherited from the weight matrix of $\mathcal{G}$ at iteration k, which is $W_k \otimes \mathbf{1}\mathbf{1}^T$, the sampling matrix becomes:

$$\Gamma_k := \Lambda_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}, \tag{8}$$

where $\Lambda_k = \mathbf{diag}(\boldsymbol{e}_k)$ denotes the elected samples at iteration k who will send messages to the nodes picked by Λ_{k+1}. Indeed, Γ_k models the virtual/actual message passing among sample node from iteration k to $k+1$.

Then, we consider the following sampling strategy, which is commonly used for many distributed learning problems.

Assumption 4. *For all $k \geqslant 0$, each device $i \in [n]$ independently and uniformly selects b_k data samples from its local datasets at random without replacement.*

Intra and inter consensus guarantee. The main purpose of the weight matrix R_k is to ensure consensus of estimates within and across devices, which is designed as follows:

$$R_k := \mathbf{I}_M - \Lambda_{k+1} + \Gamma_k, \tag{9}$$

where the term $\mathbf{I}_M - \Lambda_{k+1}$ represents that the parameters kept at the nodes that are not sampled remain unchanged at iteration $k+1$. The term Γ_k, as defined in (8), denotes the message passing from samples ξ_k to ξ_{k+1} over the augmented graph, and only samples in ξ_{k+1} perform update. Note that in such design, R_k is *row-stochastic*. There are indeed two consensus processes involved in the above process: 1) *consensus within each device*, which is guaranteed by the fully connectivity of $\mathcal{G}_V$, meaning that the latest parameters can be always sent to the current sample node; 2) *consensus across devices*, which is ensured by the proper design of weight matrix W_k that has been incorporated in Λ_k.

Accurate full-gradient estimation for tackling data heterogeneity. In order to obtain an accurate estimate on gradient descent direction, variance-reduction and gradient-tracking methods are widely used to eliminate the variance of the stochastic gradient within and across the devices, respectively. In the proposed framework, we properly design the doubly-stochastic weight matrix C_k corresponding to the augmented graph $\mathcal{G}_W \otimes \mathcal{G}_V$ as

$$C_k := G_k \otimes V_k. \tag{10}$$

Since C_k is doubly-stochastic, we have by induction,

$$\frac{\mathbf{1}_M^T}{M} Y_k = \frac{\mathbf{1}_M^T}{M} \nabla F(X_k), \forall k \geqslant 0, \tag{11}$$

which enables all the nodes to track the full gradient $\nabla f(x)$. In fact, it will become clear that $G_k \in \mathbb{R}^{n\times n}$ is meant for gradient-tracking across devices while $V_k \in \mathbb{R}^{m\times m}$ is dedicated to variance-reduction within device.

Thus, we can properly choose these above weight matrices Γ_k, R_k, C_k to recover existing algorithms with or without GT and VR operations. We use $\mathcal{A}(\Gamma_k, R_k, C_k)$ to denote algorithms generated from the proposed SPP framework. For ease of presentation, we will use the above same setting for Γ_k, R_k and C_k throughout the paper.

Remark 2. *In contrast to the existing frameworks, our proposed SPP framework provides a more general prospective for algorithm design based on sampling of an augmented graph, i.e., we allow for adopting various consensus schemes, gradient estimation methods and their combinations by different choices of the three key matrices R_k, C_k and Γ_k. Our framework covers several recently proposed frameworks, such as Local-SGD (Gorbunov et al., 2021), Cooperative SGD (Wang and Joshi, 2021), Decentralized (Gossip) SGD (Koloskova et al., 2020). Indeed, none of these works consider both gradient-tracking and variance-reduction with changing topology and local updates. Besides, the proposed framework has the potential to recover Multi-Level Local-SGD (Castiglia et al., 2020) by properly designing a topology of hierarchical structures.*

4. Existing Algorithms as Special Cases and Beyond

In this section, we show how the proposed SPP framework can, indeed, recover a large number of existing algorithms as special cases. To this end, we introduce a projection matrix $S_k \in \mathbb{R}^{n\times M}$, which is defined as

$$S_k := \left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right) \frac{\Lambda_k}{b_k}, \tag{12}$$

that can reduce the dimension of the general SPP algorithm from M (number of samples) to n (number of physical devices), yielding an algorithm that can be implemented efficiently on actual devices with new updating variables:

$$\hat{X}_k := S_k X_k, \quad \hat{Y}_k := S_k Y_k. \tag{13}$$

Now, we will use the cases of SAGA/L-SVRG ($n = 1$), and GT-SAGA ($n > 1$) to illustrate the equivalence between the recovered algorithm and the general SPP algorithm. More details on the recovery of various existing algorithms can be found in Appendix A.

Recovering SAGA/L-SVRG. Under the proposed SPP framework, we choose the parameters (Γ_k, R_k, C_k) as defined in (8), (9) and (10) for SAGA with:

$$W_k = G_k = 1,\ V_k = \mathbf{J}_m,\ \forall k \geqslant 1. \tag{14}$$

We denote by $\boldsymbol{s}_k$ the mini-batch of randomly selected $b \in [1, m]$ sample nodes at iteration k, then we can derive the recursion of decision variable:

$$\hat{x}_{k+1} = \hat{x}_k - \alpha \hat{y}_k, \tag{15}$$

where $(\hat{x}_k, \hat{y}_k)$ is an instance of $\left(\hat{X}_k, \hat{Y}_k\right)$ with $n = 1$. Then, by noticing that only the randomly selected sample nodes performs update, i.e, $x_{s,k+1} = \hat{x}_{k+1}$, $\forall s \in \boldsymbol{s}_{k+1}$, while the other sample nodes remain unchanged, i.e., $x_{j,k+1} = x_{j,k}, \forall j \notin \boldsymbol{s}_{k+1}$, we can derive the recursion of full-gradient estimation variable:

$$\begin{aligned}\hat{y}_{k+1} &= \frac{\mathbf{1}_m^T}{m}\nabla F\left(X_k\right) + S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\ &= \frac{1}{m}\sum_{j=1}^{m}\nabla f_j\left(x_{j,k}\right)\\ &\quad + \frac{1}{b}\sum_{s\in\boldsymbol{s}_{k+1}}\left(\nabla f_s\left(\hat{x}_{k+1}\right) - \nabla f_s\left(x_{s,k}\right)\right),\end{aligned} \tag{16}$$

The above procedures imply that $\nabla F\left(X_k\right)$ actually plays the role of gradient table (Defazio et al., 2014). Thus, the original SAGA is recovered.

Different from SAGA, L-SVRG, (Qian et al., 2021) reduces the stochastic gradient variance by performing full gradient update with a certain probability, it can be also recovered by SPP with the following stochastic matrix V_k varies as:

$$\begin{cases} V_k = \mathbf{I}_m, & b_k = b, \quad w.p. \quad 1-p,\\ V_k = \mathbf{J}_m, & b_k = m, \quad w.p. \quad p. \end{cases} \tag{17}$$

Then, we obtain the recovered L-SVRG algorithm by S_k,

$$\begin{aligned}\hat{x}_{k+1} &= \hat{x}_k - \alpha\hat{y}_k,\\ \hat{y}_{k+1} &= \frac{1}{m}\sum_{j=1}^{m}\nabla f_j\left(x_{j,t_{k+1}}\right)\\ &\quad + \frac{1}{b_{k+1}}\sum_{s\in\boldsymbol{s}_{k+1}}\left(\nabla f_s\left(\hat{x}_{k+1}\right) - \nabla f_s\left(x_{s,t_{k+1}}\right)\right),\end{aligned} \tag{18}$$

where $t_{k+1} < k+1$ denotes the latest iteration before $k+1$ performing full gradient update, i.e., $b_{t_{k+1}} = m$.

Recovering GT-SAGA. In contrast to the centralized setting, there exists extra data heterogeneity among devices ($n > 1$). Gradient tracking methods are proposed to further eliminating the global data heterogeneity. We recover GT-SAGA (Xin et al., 2020) by choosing (Γ_k, R_k, C_k) as defined in (8), (9) and (10) with:

$$W_k = G_k = W,\ V_k = \mathbf{J}_m,\ \forall k \geqslant 1. \tag{19}$$

Then, multiplying both sides of (7) with S_{k+1} and using the fact that $S_{k+1}R_k = S_{k+1}\Gamma_k = W_k S_k$ (c.f., Lemma (1)), we can obtain a reduced algorithm as follows:

$$\begin{aligned}\hat{X}_{k+1} &= W\left(\hat{X}_k - \alpha\hat{Y}_k\right),\\ \hat{Y}_{k+1} &= WS_kY_k + S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\ &= W\hat{Y}_k - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right).\end{aligned} \tag{20}$$

Further, noticing that $\nabla F\left(X_k\right)$ resembles the gradient table of all samples at iteration k, GT-SAGA is thus recovered under the proposed SPP framework. See Appendix A for more detailed recovery of various other algorithms.

New Efficient Algorithms. It should be noted that our proposed SPP framework can further recover a large family of algorithms by proper combination of existing algorithms, such as Local-SVRG, Gossip-PGA. In so doing, we can easily design some new efficient algorithms that have not been formally proposed before with provable rates, such as Local-SAGA, PGA-SAGA, PGA-GT-SAGA (see Table 1).

5. Convergence Analysis

In this section, we provide a unified convergence analysis of the proposed SPP framework for both strongly convex and general convex cases. To this end, we first define

Table 1. Topology design and complexity comparison for the algorithms that can be recovered and analysed by our SPP framework under **strongly convex** setting. The choices of W_k (consensus), V_k (variance-reduction) and G_k (gradient-tracking) are presented with corresponding values of (ρ_W, r, p, q). Notice that "$W_k = 1$" implies centralized scenario with only one device. The $\log \frac{1}{\varepsilon}$ factor is omitted when the sub-linear terms are dominant. The mark "*" implies that the best-known rate is recovered by our convergence analysis under the same settings; "†" denotes that the obtained rate is new under our settings.

Algorithm	W_k	V_k	G_k	(ρ_W, r, p, q)	Obtained Complexity ($\tilde{\mathcal{O}}(\cdot)$)	Results
SAGA (Defazio et al., 2014) *	1	$\mathbf{J}_m$	1	$\left\{0, 1, 1, \frac{b}{m}\right\}$	$\left(\frac{L}{\mu} + \frac{m}{b}\right) \log \frac{1}{\varepsilon}$	Cor. 2
L-SVRG (Qian et al., 2021) *	1	$\{\mathbf{I}_m, \mathbf{J}_m\}$	1	$\{0, 1, p, 1\}$	$\left(\frac{L}{\mu} + \frac{1}{p}\right) \log \frac{1}{\varepsilon}$	Cor. 2
Local-SGD (Khaled et al., 2020) *	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{1, r, 0, 0\}$	$\frac{L}{\mu r} + \frac{\sigma^*}{nb\mu^2\varepsilon} + \sqrt{\frac{(1-r)L}{\mu^3 r\varepsilon}\left(\frac{\zeta^*}{r} + \frac{\sigma^*}{b}\right)}$	Cor. 1
DSGD (Lian et al., 2017) *	W	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{\rho_W, 0, 0, 0\}$	$\frac{L}{\mu(1-\rho_W)} + \frac{\sigma^*}{nb\mu^2\varepsilon} + \sqrt{\frac{\rho_W L}{\mu^3(1-\rho_W)\varepsilon}\left(\frac{\zeta^*}{1-\rho_W} + \frac{\sigma^*}{b}\right)}$	Cor. 1
Gossip-PGA (Chen et al., 2021)	$\{W, \mathbf{J}_n\}$	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{\rho_W, r, 0, 0\}$	$\frac{L}{\mu\left(1-\rho_{r,W}\right)} + \frac{\sigma^*}{nb\mu^2\varepsilon} + \sqrt{\frac{\rho_{r,W} L}{\mu^3\left(1-\rho_{r,W}\right)\varepsilon}\left(\frac{\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right)}$	Cor. 1
Local-SAGA (New)	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\mathbf{J}_m$	$\mathbf{I}_n$	$\left\{1, r, 1, \frac{b}{m}\right\}$	$\frac{L}{\mu r} + \frac{m}{b} + \sqrt{\frac{(1-r)L\zeta^*}{\mu^3 r^2\varepsilon}}$	Cor. 2
Local-SVRG (Gorbunov et al., 2021) †	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\{\mathbf{I}_m, \mathbf{J}_m\}$	$\mathbf{I}_n$	$\{1, r, p, 1\}$	$\frac{L}{\mu r} + \frac{1}{p} + \sqrt{\frac{(1-r)L\zeta^*}{\mu^3 r^2\varepsilon}}$	Cor. 2
D-SAGA (Calauzenes and Roux, 2017)†	W	$\mathbf{J}_m$	$\mathbf{I}_n$	$\left\{\rho_W, 0, 1, \frac{b}{m}\right\}$	$\frac{L}{\mu(1-\rho_W)} + \frac{m}{b} + \sqrt{\frac{\rho_W L\zeta^*}{\mu^3(1-\rho_W)^2\varepsilon}}$	Cor. 2
PGA-SAGA (New)	$\{W, \mathbf{J}_n\}$	$\mathbf{J}_m$	$\mathbf{I}_n$	$\left\{\rho_W, r, 1, \frac{b}{m}\right\}$	$\frac{L}{\mu\left(1-\rho_{r,W}\right)} + \frac{m}{b} + \sqrt{\frac{\rho_{r,W} L\zeta^*}{\mu^3\left(1-\rho_{r,W}\right)^2\varepsilon}}$	Cor. 2
GT-SAGA (Xin et al., 2020) †	W	$\mathbf{J}_m$	W	$\left\{\rho_W, 0, 1, \frac{b}{m}\right\}$	$\left(\frac{L}{\mu(1-\rho_W)^2} + \frac{m}{b}\right) \log \frac{1}{\varepsilon}$	Cor. 3
PGA-GT-SAGA (New)	$\{W, \mathbf{J}_n\}$	$\mathbf{J}_m$	W_k	$\left\{\rho_W, r, p, \frac{b}{m}\right\}$	$\left(\frac{L}{\mu\left(1-\rho_{r,W}\right)^2} + \frac{m}{b}\right) \log \frac{1}{\varepsilon}$	Cor. 3

$p := P(V_k = \mathbf{J}_m)$ as the probability of performing local variance-reduction; $q := \mathbb{E}[b_k/m | V_k = \mathbf{J}_m]$ the expected ratio of batch-size while performing variance-reduction; $r := P(W_k = \mathbf{J}_n)$ the probability of adopting global averaging. In order to characterize the algorithm to be incorporated into the proposed framework, we also need to specify the non-negative matrices W_k, G_k, V_k which correspond to mixing, gradient-tracking and variance-reduction, respectively, as given by the following assumption.

Assumption 5. *The non-negative matrices W_k, G_k, V_k are independently and randomly chosen as*

$$W_k \in \{W, \mathbf{J}_n\}, \ G_k \in \{\mathbf{I}_n, W_k\}, \ V_k \in \{\mathbf{I}_m, \mathbf{J}_m\},$$

for all $k \geqslant 0$, and W_k is doubly stochastic, i.e., $\mathbf{1}_n^T W_k = \mathbf{1}_n^T, W_k \mathbf{1}_n = \mathbf{1}_n$, and satisfies

$$\rho_{r,W} := \mathbb{E}\left[\|W_k - \mathbf{J}_n\|_2^2\right] = (1-r)\rho_W < 1, \qquad (21)$$

where $\rho_W := \|W - \mathbf{J}_n\|_2^2$.

Remark 3. *Assumption 5 does not require a contraction property of W_k at each iteration but in the sense of expectation. For instance, W_k in Local-SGD can be randomly or periodically chosen from $\{\mathbf{I}_n, \mathbf{J}_n\}$, or simply set as $W_k = W$ with $\rho_W < 1$ for $k \geqslant 0$ in DSGD (c.f., Appendix A).*

Main results. For convergence analysis, we define the average variables of X_k and Y_k as follows:

$$\bar{x}_k := \frac{\mathbf{1}_n^T}{n} S_k X_k, \quad \bar{y}_k := \frac{\mathbf{1}_n^T}{n} S_k Y_k. \qquad (22)$$

Then, we are ready to establish the convergence rates of SPP for smooth and strongly convex objective functions under both centralized and distributed settings. To this end, we first construct a general Lyapunov function consisting of several error terms as follows:

$$\begin{aligned} T_{k+1} := c_0 \underbrace{\|\bar{x}_{k+1} - x^*\|^2}_{\text{optimal gap}} + c_1 \underbrace{\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2}_{\text{consensus error}} \\ + c_2 \underbrace{\|\nabla F(X_{t_k}) - \nabla F(\mathbf{1}_M x^*)\|^2}_{\text{delayed VR error}} \\ + c_3 \underbrace{\|\nabla F(X_k) - \nabla F(\mathbf{1}_M x^*)\|^2}_{\text{VR error}} \\ + c_4 \underbrace{\left\|\hat{Y}_{k+1} - \mathbf{1}_n \bar{y}_{k+1}\right\|^2}_{\text{GT error}}, \end{aligned} \qquad (23)$$

where $c_0, c_1, c_2, c_3, c_4 \geqslant 0$ are constants to be properly determined (see Appendix B for more details). Note that the vanishing of the Lyapunov function implies the attainment of the optimum for strongly convex objective functions.

Now, we proceed to present our main results[1] for algorithms $\mathcal{A}(\Gamma_k, R_k, C_k)$ under different parameter settings and we summarize their complexity in Table 1 for comparison.

Theorem 1. *Consider algorithms $\mathcal{A}(\cdot,\cdot,C_k \equiv \mathbf{I}_M)$ generated from the SPP framework with a constant batch-size of b. Suppose Assumption 1-5 hold and $\mu > 0$. Let*

$$c_0 = 1,\ c_1 = (1-r)\frac{8\alpha L(4\alpha L+1)}{n(1-\rho_{r,W})}, c_2 = c_3 = c_4 = 0$$

and the step-size satisfy

$$\alpha = \min\left\{\mathcal{O}\left(\frac{1}{L}\right), \mathcal{O}\left(\frac{1-\rho_{r,W}}{L\sqrt{\rho_{r,W}}}\right)\right\}. \tag{24}$$

Then, we have for all $k \geqslant 0$

$$\mathbb{E}[T_{k+1}] \leqslant \left(1 - \min\left\{\alpha\mu, \frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}[T_k] + \frac{2\alpha^2\sigma^*}{nb} + (1-r)\frac{16\alpha^3 L\rho_{r,W}}{1-\rho_{r,W}}\left(\frac{4\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right). \tag{25}$$

Remark 4. *Theorem 1 yields a known tightest convergence rate as Koloskova et al. (2020) for a class of decentralized optimization algorithms without adopting VR or GT schemes, i.e., Local-SGD, DSGD (see Table. 1).*

Corollary 1. *Under the conditions in Theorem 1, there exists a suitable upper-bound of step-size α (refer to supplementary) such that $\mathbb{E}[T_K] \leqslant \varepsilon$ after at most the following number of iterations K:*

$$K \geqslant \tilde{\mathcal{O}}\left(\frac{L}{\mu(1-\rho_{r,W})}\log\frac{\mathbb{E}[T_0]}{\varepsilon}\right) + \tilde{\mathcal{O}}\left(\frac{\sigma^*}{n\mu^2\varepsilon} + \sqrt{\frac{\rho_{r,W}L}{\mu^3(1-\rho_{r,W})\varepsilon}\left(\frac{\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right)}\right), \tag{26}$$

where $\tilde{\mathcal{O}}$ hides logarithmic factors.

Then, we provide the convergence analysis for algorithms that adopt VR schemes, such as Local-SVRG, Local-SAGA (new) and D-SAGA, in order to eliminate the internal variance σ^* due to gradient sampling.

Theorem 2. *Consider algorithms $\mathcal{A}(\cdot,\cdot,C_k \equiv \mathbf{I}_n \otimes V_k)$ generated from the SPP framework with $p > 0$. Suppose Assumption 1-5 hold and $\mu > 0$. Let*

$$c_0 = 1, c_1 = \frac{20L\alpha}{n(1-\rho_{r,W})}, c_2 = \frac{5\alpha^2}{Mp}, c_3 = \frac{16\alpha^2}{Mq}, c_4 = 0$$

and the step-size satisfy

$$\alpha = \mathcal{O}\left(\frac{1-\rho_{r,W}}{L}\right). \tag{27}$$

[1] All proofs can be found in Appendix 5.

Then, we have for all $k \geqslant 0$

$$\mathbb{E}[T_{k+1}] \leqslant \left(1 - \min\left\{\alpha\mu, \frac{pq}{2}, \frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}[T_k] + \frac{80\alpha^3 L\rho_{r,W}}{(1-\rho_{r,W})^2}\zeta^*. \tag{28}$$

Corollary 2. *Under the conditions in Theorem 2, there exists a suitable upper-bound of step-size α (refer to supplementary) such that we have $\mathbb{E}[T_K] \leqslant \varepsilon$ after at most the number of iterations K:*

$$K \geqslant \mathcal{O}\left(\frac{1}{pq} + \frac{L}{\mu(1-\rho_{r,W})}\right)\log\left(\frac{\mathbb{E}[T_0]}{\varepsilon}\right) + \mathcal{O}\left(\sqrt{\frac{\rho_{r,W}L\zeta^*}{(1-\rho_{r,W})^2\mu^3\varepsilon}}\right). \tag{29}$$

Finally, we provide the convergence analysis for algorithms adopting both VR and GT schemes, such as GT-SAGA and PGA-GT-SAGA (new) which are capable of removing both the internal (σ^*) and external variance (ζ^*) induced by the data heterogeneity within and across devices.

Theorem 3. *Consider algorithms $\mathcal{A}(\cdot,\cdot,C_k = W_k \otimes \mathbf{J}_m)$ generated from the SPP framework. Suppose Assumption 1-5 hold and $\mu > 0$. Let*

$$c_0 = 1,\ c_1 = \frac{1-\rho_{r,W}}{n\rho_{r,W}(1+\rho_{r,W})},\ c_2 = 0,$$
$$c_3 = \frac{20\alpha^2}{Mq(1-\rho_{r,W})^2},\ c_4 = \frac{8\alpha^2}{n(1-\rho_{r,W})}$$

and the step-size satisfy

$$\alpha = \mathcal{O}\left(\frac{(1-\rho_{r,W})^2}{L}\right). \tag{30}$$

Then, we have for all $k \geqslant 0$

$$\mathbb{E}[T_{k+1}] \leqslant \left(1 - \min\left\{\alpha\mu,\ \frac{q}{2},\ \frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}[T_k]. \tag{31}$$

Corollary 3. *Under the same conditions in Theorem 3, we have $\mathbb{E}[T_K] \leqslant \varepsilon$ after at most the number of iterations K:*

$$K \geqslant \tilde{\mathcal{O}}\left(\left(\frac{L}{\mu(1-\rho_{r,W})^2} + \frac{1}{q}\right)\log\frac{\mathbb{E}[T_0]}{\varepsilon}\right). \tag{32}$$

Remark 5. *It follows from the above theorems that one can always effectively remove the data heterogeneity within and across devices by properly choosing VR and GT schemes, respectively. Besides, the above results also establish a clear dependency of the complexity on the parameters related to network, cost functions and algorithm design.*

Remark 6. *In Table 1, we provide the complexity results for several well known existing and some new algorithms for smooth and strongly convex objectives under our proposed SPP framework. With proper choices of W_k, V_k and G_k, the complexity induced by Theorem 1 yields a best-known convergence rate for Local-SGD and DSGD , i.e., matching the results in (Koloskova et al., 2020). We also recover the complexity results of SAGA and L-SVRG in the centralized scenario from Theorem 2; Besides, we enhance the result of GT-SAGA (Xin et al., 2020) in that the dependency on the condition number of objectives is improved from $(L/\mu)^2$ to L/μ (c.f., Th. 3) in our slightly stronger settings. We also provide several new algorithms with provable rates, such as Local-SAGA, PGA-SAGA and PGA-GT-SAGA, which have not been formally proposed and analyzed yet.*

For general convex cases ($\mu = 0$), we provide the obtained complexity of the algorithms that can be recovered and analysed by the SPP framework in Table 2. In particular, we can also recover the best-known sub-linear rates of SAGA, L-SVRG, DSGD and Local-SGD. Moreover, we establish the sublinear rate of GT-SAGA in general convex settings, which is not yet considered in Xin et al. (2020) (c.f., Theorem 4-6). All proofs can be found in Appendix D.

6. Experimental Results

In this section, we report some experiments to verify the theoretical findings for the proposed framework under different settings of data heterogeneity and topology[2].

Experiment Settings. We train a regularized logistic regression classifier on both CIFAR-10 and Fashion-MNIST (F-MNIST) datasets over a network of n nodes each of which locally stores m data samples, which reads:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\Bigg(\underbrace{\ell(x,\xi_{ij}) + \frac{\lambda}{2}\|x\|^2}_{f_{ij}}\Bigg) \quad (33)$$

with the cross-entropy loss ℓ defined as:

$$\ell(x,\xi_{ij}) := -\sum_{c=1}^{10}\phi_{ij}^c \log\left(1+\exp\left(-x^T\theta_{i,j}\right)\right)^{-1}, \quad (34)$$

where $\lambda = 0.001$ is the regularization parameter, $\theta_{i,j} \in \mathbb{R}^d$ and $\phi_{ij}^c \in \{-1, 1\}$ is the feature vector and the label of class $c \in [10]$ associated with sample ξ_{ij}, respectively. The datasets and parameters we used are summarized in Table 3 with $r = 0.05$ for certain algorithms with global averaging.

Data heterogeneity. We consider datasets with unbalanced label distribution, which is an important type of data heterogeneity for classification problems in distributed settings (Hsieh et al., 2020), and control the level of data heterogeneity through an arithmetic sequence with a difference of h to allocate training samples of each class (c.f., Eq. (106, 107) in Appendix E). Fig. 2 plots the testing accuracy of the algorithms to be compared under different settings of heterogeneous label distributions: i) independent and identically (label) distributed (IID) datasets on the top row, i.e., $h = 0$; ii) non-IID datasets with $h = 124$ on the middle row; iii) A highly unbalanced label distribution denoted by $h = h_{\max}$ on the bottom row (c.f., Table 6). It follows from Fig. 2 that, when the labels are evenly distributed (top row), the VR-based algorithms (such as SAGA, D-SAGA, Local-SAGA, denoted in solid lines) outperform the others without VR schemes. Furthermore, when the level of data heterogeneity increases (middle and bottom rows), GT-SAGA and PGA-GT-SAGA that adopt GT schemes can maintain relatively high testing accuracy while those without GT will degrade dramatically, especially for those that also do not employ VR schemes, which implies that GT-based methods are more robust against the data heterogeneity. The above experimental results verify the effectiveness of adopting VR and GT schemes in scenarios where datasets are heterogeneous, which corroborates the Theorem 1-3.

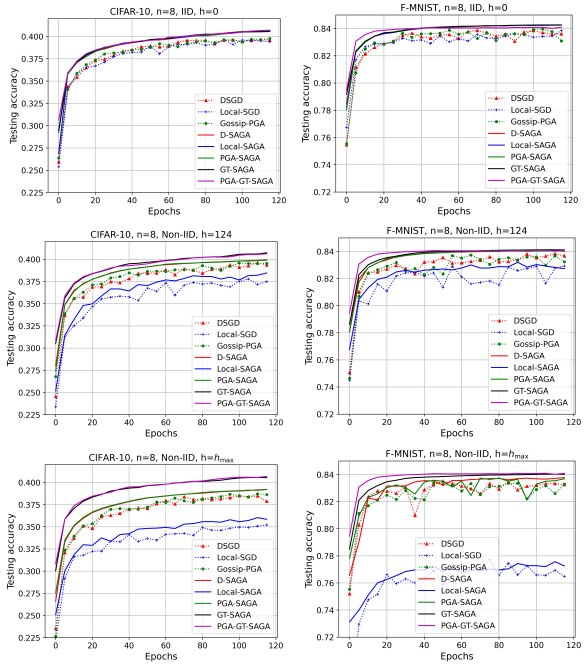

Figure 2. Performance comparison of DSGD, Local-SGD, Gossip-PGA, D-SAGA, Local-SAGA, PGA-SAGA, GT-SAGA and PGA-GT-SAGA on directed ring graph with $n = 8$ under different settings of data heterogeneity.

Topology Dependence. To verify the dependency of the impact of data heterogeneity on the performance against the connectivity of the topology, we conduct several experiments (with a fixed value of $h = 20$) on different graphs: directed ring with $n = 8$, $\rho_W \approx 0.92$; exponential graph

[2]More experimental results can be found in Appendix E.

Table 2. Topology design and complexity comparison for the algorithms that can be recovered and analysed by our SPP framework under **convex** settings ($\mu = 0$). The choices of W_k (consensus), V_k (variance-reduction) and G_k (gradient-tracking) are presented with corresponding values of (ρ_W, r, p, q). Notice that "$W_k = 1$" implies centralized scenario with only one device. The mark "*" implies that the best-known rate is recovered by our analysis under the same settings; "†" denotes that the obtained rate is new under our settings; $\mathcal{O}(\cdot)$ hides the initial error $\|x_0 - x^*\|^2$.

Algorithm	W_k	V_k	G_k	(ρ_W, r, p, q)	**Obtained Complexity** ($\mathcal{O}(\cdot)$)	**Results**
SAGA (Defazio et al., 2014) *	1	$\mathbf{J}_m$	1	$\{0, 1, 1, \frac{b}{m}\}$	$\frac{mL}{b\varepsilon}$	Cor. 5
L-SVRG (Qian et al., 2021) *	1	$\{\mathbf{I}_m, \mathbf{J}_m\}$	1	$\{0, 1, p, 1\}$	$\frac{L}{p\varepsilon}$	Cor. 5
Local-SGD (Khaled et al., 2020) *	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{1, r, 0, 0\}$	$\frac{L}{r\varepsilon} + \frac{\sigma^*}{nb\varepsilon^2} + \frac{1}{\varepsilon^{3/2}}\sqrt{\frac{(1-r)L}{r}\left(\frac{\zeta^*}{r} + \frac{\sigma^*}{b}\right)}$	Cor. 4
DSGD (Lian et al., 2017) *	W	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{\rho_W, 0, 0, 0\}$	$\frac{L}{(1-\rho_W)\varepsilon} + \frac{\sigma^*}{nb\varepsilon^2} + \frac{1}{\varepsilon^{3/2}}\sqrt{\frac{\rho_W L}{1-\rho_W}\left(\frac{\zeta^*}{1-\rho_W} + \frac{\sigma^*}{b}\right)}$	Cor. 4
Gossip-PGA (Chen et al., 2021)	$\{W, \mathbf{J}_n\}$	$\mathbf{I}_m$	$\mathbf{I}_n$	$\{\rho_W, r, 0, 0\}$	$\frac{L}{(1-\rho_{r,W})\varepsilon} + \frac{\sigma^*}{nb\varepsilon^2} + \frac{1}{\varepsilon^{3/2}}\sqrt{\frac{\rho_{r,W} L}{1-\rho_{r,W}}\left(\frac{\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right)}$	Cor. 4
Local-SAGA (New)	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\mathbf{J}_m$	$\mathbf{I}_n$	$\{1, r, 1, \frac{b}{m}\}$	$\frac{mL}{br\varepsilon} + \frac{m\sqrt{(1-r)L\zeta^*}}{br\varepsilon^{3/2}}$	Cor. 5
Local-SVRG (Gorbunov et al., 2021) †	$\{\mathbf{I}_n, \mathbf{J}_n\}$	$\{\mathbf{I}_m, \mathbf{J}_m\}$	$\mathbf{I}_n$	$\{1, r, p, 1\}$	$\frac{L}{rp\varepsilon} + \frac{\sqrt{(1-r)L\zeta^*}}{rp\varepsilon^{3/2}}$	Cor. 5
D-SAGA (Calauzenes and Roux, 2017)†	W	$\mathbf{J}_m$	$\mathbf{I}_n$	$\{\rho_W, r, 1, \frac{b}{m}\}$	$\frac{mL}{b(1-\rho_W)\varepsilon} + \frac{m\sqrt{\rho_W L\zeta^*}}{b(1-\rho_W)\varepsilon^{3/2}}$	Cor. 5
PGA-SAGA (New)	$\{W, \mathbf{J}_n\}$	$\mathbf{J}_m$	$\mathbf{I}_n$	$\{\rho_W, r, 1, \frac{b}{m}\}$	$\frac{mL}{b\left(1-\rho_{r,W}\right)\varepsilon} + \frac{m\sqrt{\rho_{r,W} L\zeta^*}}{b\left(1-\rho_{r,W}\right)\varepsilon^{3/2}}$	Cor. 5
GT-SAGA (Xin et al., 2020) †	W	$\mathbf{J}_m$	W	$\{\rho_W, 0, 1, \frac{b}{m}\}$	$\frac{mL}{b(1-\rho_W)^2\varepsilon}$	Cor. 6
PGA-GT-SAGA (New)	$\{W, \mathbf{J}_n\}$	$\mathbf{J}_m$	W_k	$\{\rho_W, r, p, \frac{b}{m}\}$	$\frac{mL}{b\left(1-\rho_{r,W}\right)^2\varepsilon}$	Cor. 6

Table 3. Summary of the experimental setup.

Dataset	**Node** (n)	**#Train**	**#Test**	**BS** ($n \times b$)	**SS** (α)
F-MNIST	{8, 50}	60000	10000	200	0.05
CIFAR-10	{8, 50}	50000	10000	400	0.008

BS: Batch-size; SS: Step-size.

with $n = 50$, $\rho_W \approx 0.99$ for DSGD, D-SAGA and GT-SAGA, respectively, whose testing accuracy results are plotted in Fig. 3. It follows from Fig. 3 that the performance of both DSGD and D-SAGA will be degraded when the connectivity of the graph becomes worse while GT-SAGA maintains relatively high testing accuracy since it removes data heterogeneity ζ^* by employing GT-schemes.

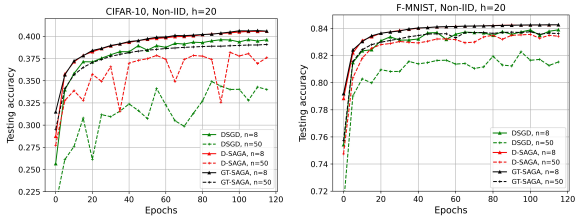

Figure 3. Performance comparison of DSGD, D-SAGA, and GT-SAGA on graphs with $n = 8, 50$.

7. Conclusions

This paper develops a new unified framework for first-order stochastic gradient methods both in centralized and distributed scenarios. The proposed framework is able to recover many existing stochastic algorithms along with their corresponding rates. It also enable us to easily design new efficient algorithms by proper design of sampling strategies on the augmented graph. This framework is especially suitable for scenarios where data heterogeneity is a key concern. Since the framework heavily depends on the underlying augmented graph, it is of great interest and importance to design a proper augmented graph and sampling strategy to account for different important scenarios.

Acknowledgements

The work of Huang, Zhu, Yan, and Xu has been supported in parts by National Natural Science Foundation of China under Grants 62003302, 62088101, 61922058 and 62173225; and in parts by the Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province. The work of Sun has been supported by the Office of Naval Research, under the grant N00014-21-1-2673.

References

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. (2019). Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR.

Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Calauzenes, C. and Roux, N. L. (2017). Distributed SAGA: Maintaining linear convergence rate with limited communication. *arXiv preprint arXiv:1705.10405*.

Castiglia, T., Das, A., and Patterson, S. (2020). Multi-level local SGD: Distributed SGD for heterogeneous hierarchical networks. In *International Conference on Learning Representations*.

Chen, Y., Yuan, K., Zhang, Y., Pan, P., Xu, Y., and Yin, W. (2021). Accelerating gossip SGD with periodic global averaging. In *International Conference on Machine Learning*, pages 1791–1802. PMLR.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., et al. (2012). Large scale distributed deep networks. *Advances in neural information processing systems*, 25.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.

Di Lorenzo, P. and Scutari, G. (2016). Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136.

Gorbunov, E., Hanzely, F., and Richtárik, P. (2020). A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR.

Gorbunov, E., Hanzely, F., and Richtárik, P. (2021). Local SGD: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3556–3564. PMLR.

Gut, A. (2005). *Probability: A Graduate Course.* New York, NY : Springer Science+Business Media, Inc.

Hendrikx, H., Bach, F., and Massoulie, L. (2021). An optimal algorithm for decentralized finite-sum optimization. *SIAM Journal on Optimization*, 31(4):2753–2783.

Hofmann, T., Lucchi, A., Lacoste-Julien, S., and McWilliams, B. (2015). Variance reduced stochastic gradient descent with neighbors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2305–2313.

Hong, M., Hajinezhad, D., and Zhao, M.-M. (2017). Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks. In *International Conference on Machine Learning*, pages 1529–1538. PMLR.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. (2020). The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR.

Hu, B., Seiler, P., and Rantzer, A. (2017). A unified analysis of stochastic optimization methods using jump system theory and quadratic constraints. In *Conference on Learning Theory*, pages 1157–1189. PMLR.

Jiang, X., Zeng, X., Sun, J., and Chen, J. (2022). Distributed stochastic gradient tracking algorithm with variance reduction for non-convex optimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323.

Khaled, A., Mishchenko, K., and Richtárik, P. (2020). Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR.

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. (2020). A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR.

Li, B., Cen, S., Chen, Y., and Chi, Y. (2020). Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. In *International Conference on Artificial Intelligence and Statistics*, pages 1662–1672. PMLR.

Li, B., Li, Z., and Chi, Y. (2021). DESTRESS: Computation-optimal and communication-efficient decentralized nonconvex finite-sum optimization. *arXiv preprint arXiv:2110.01165*.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2019). On the convergence of FedAvg on non-iid data. In *International Conference on Learning Representations*.

Lian, X., Huang, Y., Li, Y., and Liu, J. (2015). Asynchronous parallel stochastic gradient for nonconvex optimization. *Advances in Neural Information Processing Systems*, 28.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. (2017). Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30.

Lobel, I. and Ozdaglar, A. (2011). Distributed subgradient methods for convex optimization over random networks. *IEEE Trans. Autom. Control*, 56(6):1291–1306.

Mokhtari, A. and Ribeiro, A. (2016). DSA: Decentralized double stochastic averaging gradient algorithm. *The Journal of Machine Learning Research*, 17(1):2165–2199.

Nedić, A., Olshevsky, A., and Rabbat, M. G. (2018). Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976.

Nedic, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61.

Nedic, A., Ozdaglar, A., and Parrilo, P. A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938.

Nesterov, Y. (2003). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. (2017). SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2613–2621.

Pu, S. and Nedić, A. (2020). Distributed stochastic gradient tracking methods. *Mathematical Programming*, pages 1–49.

Pu, S., Shi, W., Xu, J., and Nedic, A. (2020). Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*.

Qian, X., Qu, Z., and Richtárik, P. (2021). L-SVRG and L-Katyusha with arbitrary sampling. *Journal of Machine Learning Research*, 22(112):1–47.

Qu, G. and Li, N. (2017). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260.

Ram, S. S., Nedić, A., and Veeravalli, V. V. (2009). Asynchronous gossip algorithms for stochastic optimization. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3581–3586. IEEE.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pages 3027–3036. PMLR.

Stich, S. U. (2019). Local SGD converges fast and communicates little. In *ICLR 2019-International Conference on Learning Representations*, number CONF.

Sun, H., Lu, S., and Hong, M. (2020). Improving the sample and communication complexity for decentralized nonconvex optimization: Joint gradient estimation and tracking. In *International Conference on Machine Learning*, pages 9217–9228. PMLR.

Wang, J. and Joshi, G. (2021). Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. *Journal of Machine Learning Research*, 22(213):1–50.

Xin, R., Khan, U. A., and Kar, S. (2020). Variance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68:6255–6271.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2015). Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 2055–2060.

Ye, H., Xiong, W., and Zhang, T. (2020). PMGT-VR: A decentralized proximal-gradient algorithmic framework with variance reduction. *arXiv preprint arXiv:2012.15010*.

Yuan, K., Ling, Q., and Yin, W. (2016). On the convergence of decentralized gradient descent. *SIAM J. on Optim.*, 26(3):1835–1854.

Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329.

Appendix

Contents

A. Recovering Existing Algorithms

In this section, we show that most of existing algorithms can be recovered by the proposed SPP framework. To this end, we first recall our proposed SPP framework (7) as follows:

$$\begin{aligned} X_{k+1} &= R_k X_k - \alpha \Gamma_k Y_k, \\ Y_{k+1} &= C_k Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right), \end{aligned}$$

where

$$\Gamma_k = \Lambda_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}, \quad R_k = \mathbf{I}_M - \Lambda_{k+1} + \Gamma_k, \quad C_k = G_k \otimes V_k.$$

Note that the matrices R_k is row-stochastic and C_k is doubly-stochastic such that we have by induction

$$\frac{\mathbf{1}_M^T}{M} Y_k = \frac{\mathbf{1}_M^T}{M} \nabla F\left(X_k\right), \forall k \geq 0.$$

Also, let us recall the notions with dimension mentioned in the main text:

$$\begin{aligned} X_k &:= \left[x_{1,k}, x_{2,k}, \cdots, x_{M,k}\right]^T \in \mathbb{R}^{M\times d}, \quad x_{i,k} \in \mathbb{R}^{1\times d} \quad \forall i, \\ Y_k &:= \left[y_{1,k}, y_{2,k}, \cdots, y_{M,k}\right]^T \in \mathbb{R}^{M\times d}, \quad y_{i,k} \in \mathbb{R}^{1\times d} \quad \forall i, \\ \nabla F\left(X_k\right) &:= \left[\nabla f_1\left(x_{1,k}\right), \cdots, \nabla f_M\left(x_{M,k}\right)\right]^T \in \mathbb{R}^{M\times d}, \\ \hat{X}_k &:= S_k X_k = \left[\hat{x}_{1,k}^T, \hat{x}_{2,k}^T, \cdots, \hat{x}_{n,k}^T\right]^T \in \mathbb{R}^{n\times d}, \\ \hat{Y}_k &:= S_k Y_k = \left[\hat{y}_{1,k}^T, \hat{y}_{2,k}^T, \cdots, \hat{y}_{n,k}^T\right]^T \in \mathbb{R}^{n\times d}, \\ \bar{x}_k &:= \frac{\mathbf{1}_n^T}{n}\hat{X}_k = \frac{\mathbf{1}_n^T}{n} S_k X_k \in \mathbb{R}^{1\times d}, \\ \bar{y}_k &:= \frac{\mathbf{1}_n^T}{n}\hat{Y}_k = \frac{\mathbf{1}_n^T}{n} S_k Y_k \in \mathbb{R}^{1\times d}, \\ x^* &:= \operatorname*{argmin}_x f\left(x\right) \in \mathbb{R}^{1\times d}. \end{aligned} \tag{35}$$

Now, we show that how each algorithm mentioned in this paper can be recovered with particular choices of $(\Gamma_k,\ R_k, C_k)$ as well as the projection matrix $S_k := \left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_k}{b_k}$ (c.f., Eq. (12)).

The following lemma related to Kronecker product is crucial in recovering these algorithms.

Lemma 1. *Suppose Assumption 4 holds. Then we have for all $k \geqslant 0$,*

$$S_{k+1} R_k = S_{k+1}\Gamma_k = W_k S_k. \tag{36}$$

Proof. Since $\Lambda_k\left(\mathbf{I}_M - \Lambda_k\right) = 0$ according to the definition of Λ_k, we have $S_{k+1}R_k = S_{k+1}\Gamma_k$. Further, we obtain that

$$\begin{aligned} S_{k+1}R_k = S_{k+1}\Gamma_k &= \left(\left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_{k+1}}{b_{k+1}}\right)\left(\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}\right) \\ &= \frac{1}{b_k b_{k+1}}\begin{bmatrix} \ddots & & \\ & \mathbf{1}_m^T\Lambda_{k+1}^i & \\ & & \ddots \end{bmatrix}\begin{bmatrix} \cdots & w_{1j,k}\mathbf{1}\mathbf{1}^T\Lambda_k^j & \cdots \\ & \vdots & \\ \cdots & w_{nj,k}\mathbf{1}\mathbf{1}^T\Lambda_k^j & \cdots \end{bmatrix} \\ &= \frac{1}{b_k}\begin{bmatrix} \cdots & w_{1j,k}\mathbf{1}_m^T\Lambda_k^j & \cdots \\ & \vdots & \\ \cdots & w_{nj,k}\mathbf{1}_m^T\Lambda_k^j & \cdots \end{bmatrix} = W_k\left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_k}{b_k}, \end{aligned} \tag{37}$$

which completes the proof. □

A.1. Recovering (centralized) SAGA, L-SVRG and SARAH

To recover the algorithms with VR schemes, we set the matrices $(\mathit{\Gamma}_k,\ R_k,\ C_k)$ as follows:

$$\mathit{\Gamma}_k = \mathit{\Lambda}_{k+1}\mathbf{1}\mathbf{1}^T\frac{\mathit{\Lambda}_k}{b_k},\quad R_k = \mathbf{I}_m - \mathit{\Lambda}_{k+1} + \mathit{\Lambda}_{k+1}\mathbf{1}\mathbf{1}^T\frac{\mathit{\Lambda}_k}{b_k},\quad C_k = V_k.$$

For SAGA. The matrices W_k, G_k and V_k are set as follows (c.f., Section 4 for more details for these matrices):

$$W_k = G_k = 1,\ V_k = \mathbf{J}_m,\ k \geqslant 1. \tag{38}$$

We denote by $\boldsymbol{s}_k$ the set of randomly selected sample nodes at iteration k with constant mini-batch size $b \in [1, m]$. Then, multiplying S_{k+1} for both sides of (7) and invoking Lemma 1, we can derive the following recursions:

$$\begin{aligned}
\hat{x}_{k+1} &= S_{k+1}\left(R_k X_k - \alpha \mathit{\Gamma}_k Y_k\right) = \hat{x}_k - \alpha\hat{y}_k,\\
\hat{y}_{k+1} &= S_{k+1}\left(\mathbf{J}_m Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_{t_k}\right)\right)\\
&= \frac{\mathbf{1}_m^T}{m}\nabla F\left(X_k\right) + S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\
&= \frac{1}{m}\sum_{j=1}^{m}\nabla f_j\left(x_{j,k}\right) + \frac{1}{b}\sum_{s\in\boldsymbol{s}_{k+1}}\left(\nabla f_s\left(\hat{x}_{k+1}\right) - \nabla f_s\left(x_{s,k}\right)\right),
\end{aligned} \tag{39}$$

where $(\hat{x}_k, \hat{y}_k)$ is the specific instance of $\left(\hat{X}_k, \hat{Y}_k\right)$ with $n = 1$, and $s \in \boldsymbol{s}_{k+1}$ represents the index of sample randomly picked at iteration $k+1$. It is worth noting that only the picked sample nodes in $s \in \boldsymbol{s}_{k+1}$ performs updates at each iteration. As a result, the collective gradient vector $\nabla F\left(X_k\right)$ keeps the historical gradients of all samples, resembling the role of the table storing the gradients in the original SAGA algorithm (Defazio et al., 2014).

For L-SVRG. The matrix V_k corresponding to variance reduction and batch-size b_k vary as:

$$\begin{cases} V_k = \mathbf{I}_m, & b_k = b, \quad w.p. \quad 1-p\\ V_k = \mathbf{J}_m, & b_k = m, \quad w.p. \quad p \end{cases},$$

which indicates that the algorithm performs full gradient update with probability p. Then, by Lemma 1 and Lemma 5, we derive the recursions of the recovered L-SVRG algorithm with the projection matrix S_{k+1}:

$$\begin{aligned}
\hat{x}_{k+1} &= S_{k+1}\left(R_k X_k - \alpha \mathit{\Gamma}_k Y_k\right) = \hat{x}_k - \alpha\hat{y}_k,\\
\hat{y}_{k+1} &= S_{k+1}\left(V_k Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\
&= \frac{1}{m}\sum_{j=1}^{m}\nabla f_j\left(x_{j,t_{k+1}}\right) + \frac{1}{b_{k+1}}\sum_{s\in\boldsymbol{s}_{k+1}}\left(\nabla f_s\left(\hat{x}_{k+1}\right) - \nabla f_s\left(x_{s,t_{k+1}}\right)\right),
\end{aligned} \tag{40}$$

where $t_{k+1} < k+1$ denotes the latest iteration before $k+1$ performing the full gradient update, i.e., $b_{t_{k+1}} = m$, and $x_{j,t_{k+1}} = \hat{x}_{t_{k+1}}, \forall j \in [m]$. The original L-SVRG algorithm (Qian et al., 2021) is thus recovered.

For SARAH. As with SAGA and L-SVRG, we can verify that SARAH can be also recovered from the SPP framework by setting:

$$V_k = \mathbf{J}_m,\quad b_k = \begin{cases} b, & w.p. \quad 1-p\\ m, & w.p. \quad p \end{cases}, \tag{41}$$

which indicates that SARAH always performs variance reduction (since $V_k = \mathbf{J}_m$) while using dynamic sampling strategy. Therefore, we can obtain the original SARAH algorithm (Nguyen et al., 2017) by the projection matrix S_k, which shares the same recursions of SAGA in (39) while intermittently performing full gradient update. It can be further revealed from Table 4 that SARAH is, indeed, a mixing of SAGA and L-SVRG.

Table 4. A unified perspective for SAGA, L-SVRG and SARAH under SPP framework

Algorithms	SAGA	L-SVRG	**SARAH**
b_k	b	$\{b,\ m\}$	$\{b,\ m\}$
V_k	$\mathbf{J}_m$	$\{\mathbf{I}_m, \mathbf{J}_m\}$	$\mathbf{J}_m$

A.2. Recovering Local-SAGA, D-SAGA and PGA-SAGA

We choose the parameters $(\mathit{\Gamma}_k,\ R_k,\ C_k)$ as follow:

$$\mathit{\Gamma}_k = \mathit{\Lambda}_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\mathit{\Lambda}_k}{b_k}, \quad R_k = \mathbf{I}_M - \mathit{\Lambda}_{k+1} + \mathit{\Lambda}_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\mathit{\Lambda}_k}{b_k}, \quad C_k = \left(\mathbf{I}_n \otimes V_k\right).$$

For Local-SAGA. We set $G_k = \mathbf{I}_n, V_n = \mathbf{J}_m$, and the mixing matrix W_k corresponding to the actual communication topology varies as:

$$W_k = \begin{cases} \mathbf{I}_n, & w.p. \quad 1-r \\ \mathbf{J}_n, & w.p. \quad r \end{cases}.$$

From Lemma 1, we derive the following recursion of the recovered Local-SAGA algorithm:

$$\begin{aligned} \hat{X}_{k+1} &= S_{k+1}\left(R_k X_k - \alpha \mathit{\Gamma}_k Y_k\right) = W_k\left(\hat{X}_k - \alpha \hat{Y}_k\right), \\ \hat{Y}_{k+1} &= S_{k+1}\left(\left(\left(\mathbf{I}_n \otimes \mathbf{J}_m\right) Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\right) \\ &= \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_k\right) + S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right). \end{aligned} \tag{42}$$

Then, by noticing that the decision variables X_{k+1} and X_k are only different at the rows corresponding to samples at iteration $k+1$, we have

$$S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right) = \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right), \tag{43}$$

which implies

$$\hat{Y}_{k+1} = \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_k\right). \tag{44}$$

By doing so, we get the recovered Local-SAGA algorithm:

$$\hat{X}_{k+1} = W_k\left(\hat{X}_k - \alpha\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_k\right)\right), \tag{45}$$

which indicates that each device performs SAGA over their local datasets, and carry out global communication at a probability of r to reach consensus .

For D-SAGA and PGA-SAGA. Similar to Local-SAGA, the only difference between them are the mixing matrix W_k, in specific, we can recover D-SAGA by choosing $W_k = W$ for $k \geqslant 0$, and PGA-SAGA by choosing W_k varying as:

$$W_k = \begin{cases} W, & w.p. \quad 1-r \\ \mathbf{J}_n, & w.p. \quad r \end{cases}. \tag{46}$$

A.3. Recovering Local-SVRG and D-SVRG

For Local-SVRG. The matrix W_k corresponding to the actual communication topology, and V_k corresponding to variance reduction vary as:

$$W_k = \begin{cases} \mathbf{I}_n, & w.p. \quad 1-r \\ \mathbf{J}_n, & w.p. \quad r \end{cases}, \quad \begin{cases} V_k = \mathbf{I}_m, & b_k = b, \quad w.p. \quad 1-p \\ V_k = \mathbf{J}_m, & b_k = m, \quad w.p. \quad p \end{cases},$$

which indicates that each device performs full gradient update with probability p (L-SVRG) over their local datasets, and global communication to reach consensus with probability r.

Then, similar to Local-SAGA, we can drive the recursions of the recovered Local-SVRG algorithm with S_{k+1}:

$$\begin{aligned}\hat{X}_{k+1} &= S_{k+1}\left(R_k X_k - \alpha \Gamma_k Y_k\right) = W_k\left(\hat{X}_k - \alpha \hat{Y}_k\right),\\ \hat{Y}_{k+1} &= \left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_{k+1}}{b_{k+1}}\left(\left(\mathbf{I}_n \otimes V_k\right)Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\ &= \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_{t_{k+1}}\right) + \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_{t_{k+1}}\right)\right).\end{aligned} \tag{47}$$

Noticing that $t_{k+1} < k+1$ denotes the iteration before $k+1$ performing full gradient update, which thus recovers the original Local-SVRG algorithm (Gorbunov et al., 2021).

For D-SVRG. Similar to Local-SVRG, it is straightforward to recover D-SVRG by choosing the mixing matrix $W_k = W$ for $k \geqslant 0$.

A.4. Recovering GT-SAGA

We choose the parameters $(\Gamma_k,\, R_k,\, C_k)$ as follow:

$$\Gamma_k = \Lambda_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}, \quad R_k = \mathbf{I}_M - \Lambda_{k+1} + \Lambda_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}, \quad C_k = \left(W_k \otimes V_k\right),$$

which indicates performing both gradient tracking and variance reduction. Then, using Lemma 1, we can derive the recursions of the recovered GT-SAGA algorithm:

$$\begin{aligned}\hat{X}_{k+1} &= S_{k+1}\left(R_k X_k - \alpha \Gamma_k Y_k\right) = W_k\left(\hat{X}_k - \alpha \hat{Y}_k\right),\\ \hat{Y}_{k+1} &= \left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_{k+1}}{b_{k+1}}\left(\left(W_k \otimes \mathbf{J}_m\right)Y_k + \nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right)\\ &= W_k\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)Y_k + S_{k+1}\nabla F\left(X_{k+1}\right) - S_{k+1}\nabla F\left(X_k\right)\\ &= W_k\hat{Y}_k - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_{k+1}\right) - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_k\right),\end{aligned} \tag{48}$$

where in the last equality we have used the following facts:

$$\begin{aligned}&\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)Y_k = \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\Lambda_k Y_k = \hat{Y}_k,\\ &S_{k+1}\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right) = \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\left(\nabla F\left(X_{k+1}\right) - \nabla F\left(X_k\right)\right).\end{aligned}$$

Noticing that $\nabla F\left(X_k\right)$, indeed, plays the role of the table of SAGA storing the historical gradients, we thus recover the original GT-SAGA algorithm (Xin et al., 2020).

New efficient algorithms. It should be noted that we can further design various new algorithms by properly choosing the matrices W_k, G_k and V_k to obtain suitable performance for different scenarios.

B. Technical Preliminaries

The key idea of the proofs for the main results in Section 5 is based on the proper design of the following Lyapunov function as defined in (23), which we recall here:

$$\begin{aligned}T_{k+1} :=\ & c_0\left\|\bar{x}_{k+1} - x^*\right\|^2 + c_1\left\|\hat{X}_{k+1} - \mathbf{1}_n\bar{x}_{k+1}\right\|^2\\ &+ c_2\left\|\nabla F\left(X_{t_k}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 + c_3\left\|\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 + c_4\left\|\hat{Y}_k - \mathbf{1}_n\bar{y}_k\right\|^2,\end{aligned}$$

where $c_0, c_1, c_2, c_3, c_4 \geqslant 0$ are parameters to be properly determined. In particular, the above Lyapunov function consists of the following five error terms in the sense of expectation:

- **Optimality gap**: $\mathbb{E}\left[\left\|\bar{x}_k - x^*\right\|^2\right]$;
- **Consensus error across proxy nodes**: $\mathbb{E}\left[\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2\right]$;
- **Delayed variance-reduction error**: $\mathbb{E}\left[\left\|\nabla F\left(X_{t_k}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2\right]$;
- **Variance-reduction error** : $\mathbb{E}\left[\left\|\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2\right]$;
- **Gradient tracking error**: $\mathbb{E}\left[\left\|\hat{Y}_k - \mathbf{1}_n \bar{y}_k\right\|^2\right]$.

Proof Sketch: To obtain the main results, we first need to establish the evolution of each of the above error terms denoted as $\boldsymbol{e}_k$ (c.f., Lemma 9-14) to come up with a recursive dynamics: $\boldsymbol{e}_{k+1} \leqslant A\boldsymbol{e}_k + \boldsymbol{b}$. Then, by properly choosing a non-negative non-zero coefficient vector $\boldsymbol{c} := [c_0, \cdots, c_4]^T$ such that we have $\boldsymbol{c}^T A \leqslant \varrho \boldsymbol{c}^T$ with $\varrho < 1$, we are able us to obtain the convergence results as stated in Theorem (Corollary) 1-3 for smooth and strongly convex objectives ($\mu > 0$). Similar procedures can be applied to obtain the sub-linear rates for general convex cases ($\mu = 0$).

Before proceeding to the main proofs, we first introduce some lemmas that will be crucial in the subsequent analysis. Besides, we denote by $\mathcal{F}_k$ the σ-algebra generated by $\{\Lambda_0, R_0, C_0, \cdots, \Lambda_{k-1}, R_{k-1}, C_{k-1}\}$, and define $\mathbb{E}\left[\cdot|\mathcal{F}_k\right]$ as the conditional expectation given $\mathcal{F}_k$. We also recall the following definitions:

$$\rho_W := \|W - \mathbf{J}_n\|_2^2,\ r := P\left(W_k = \mathbf{J}_n\right),\ p := P\left(V_k = \mathbf{J}_m\right),\ q := \mathbb{E}\left[b_k/m | V_k = \mathbf{J}_m\right],$$

where ρ_W denotes the spectral gap of the fixed mixing matrix W; r represents the probability of adopting global averaging; p represents the probability of performing local variance reduction (i.e., updating the gradient table kept by each device); q represents the expected batch-size of samples while performing variance reduction.

B.1. Supporting Lemmas

In this section, we first provide some lemmas that will be used in the subsequent analysis.

Lemma 2. *(Nesterov, 2003) Suppose Assumption 1 hold. Then, for any $x, x' \in \mathbb{R}^d$, we have*

$$\left\|\nabla f_i\left(x\right) - \nabla f_i\left(x'\right)\right\|^2 \leqslant 2L\left(f_i\left(x\right) - f_i\left(x'\right) - \left\langle \nabla f_i\left(x'\right), x - x'\right\rangle\right). \tag{49}$$

Lemma 3. *Let $\boldsymbol{x} \in \mathbb{R}^n$ be a random vector and $A \in \mathbb{R}^{n\times n}$ be a random matrix. Then, if A is independent on $\boldsymbol{x}$, we have*

$$\mathbb{E}\left[\left\|A\boldsymbol{x}\right\|^2\right] = \mathbb{E}\left[\boldsymbol{x}^T A^T A \boldsymbol{x}\right] = \mathbb{E}\left[\boldsymbol{x}^T \mathbb{E}\left[A^T A | \boldsymbol{x}\right] \boldsymbol{x}\right] \leqslant \rho\left(\mathbb{E}\left[A^T A\right]\right) \mathbb{E}\left[\left\|\boldsymbol{x}\right\|^2\right]. \tag{50}$$

The above lemma follows directly from the smoothing lemma (Gut, 2005) and thus its proof is omitted.

Lemma 4. *Suppose Assumption 4 hold. Then, we have for all $k \geqslant 0$,*

$$\rho(\mathbb{E}\left[\left(S_k\right)^T S_k\right]) \leqslant \frac{1}{m}, \tag{51}$$

and

$$\rho\left(\mathbb{E}\left[\left(\frac{\mathbf{1}_n^T}{n} S_k\right)^T \left(\frac{\mathbf{1}_n^T}{n} S_k\right)\right]\right) \leqslant \frac{1}{M}, \tag{52}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

Proof. Under Assumption 4, we know that each node i performs independent identically distributed mini-batch sampling without replacement from finite m data samples locally. Therefore, by Lemma 3, we have

$$\begin{aligned}
&\mathbb{E}\left[\left(S_k\right)^T S_k\right] = \mathbb{E}\left[\frac{\Lambda_k}{b_k}\left(\mathbf{I}_n \otimes \mathbf{1}_m \mathbf{1}_m^T\right)\frac{\Lambda_k}{b_k}\right] \\
&= \mathbb{E}\left[\frac{1}{\left(b_k\right)^2}\left(\frac{C_{m-1}^{b_k-1}}{C_m^{b_k}}\mathbf{I}_M + \frac{C_{m-2}^{b_k-2}}{C_m^{b_k}}\left(\mathbf{I}_n \otimes \left(\mathbf{1}_m \mathbf{1}_m^T - \mathbf{I}_m\right)\right)\right)\right] \\
&= \mathbb{E}\left[\frac{1}{\left(b_k\right)^2}\left(\frac{b_k}{m}\left(1 - \frac{b_k - 1}{m-1}\right)\mathbf{I}_M + \frac{b_k\left(b_k-1\right)}{m\left(m-1\right)}\left(\mathbf{I}_n \otimes \mathbf{1}_m \mathbf{1}_m^T\right)\right)\right],
\end{aligned}$$

where $C_m^{b_k}$ denote the number of b_k combinations from the set $[m]$. Then, using Gershgorin circle theorem, we have

$$\rho\left(\mathbb{E}\left[\left(S_k\right)^T S_k\right]\right) \leqslant \frac{1}{m},$$

and

$$\rho\left(\mathbb{E}\left[\left(\frac{\mathbf{1}_n^T}{n}S_k\right)^T\left(\frac{\mathbf{1}_n^T}{n}S_k\right)\right]\right) \leqslant \rho\left(\frac{\mathbf{1}_n\mathbf{1}_n^T}{n^2}\right)\rho\left(\mathbb{E}\left[\left(S_k\right)^T S_k\right]\right) = \frac{1}{M},$$

which completes the proof. □

Lemma 5. *Suppose $C_k = \mathbf{I}_M$ hold for $k > t_k$. Then, we have by induction that*

$$Y_k = C_{t_k} Y_{t_k} + \nabla F\left(X_k\right) - \nabla F\left(X_{t_k}\right). \tag{53}$$

The above lemma shows that there will be a delay in the recursion of Y_k when the estimation on the full gradient is not adopted at each iteration (i.e., $C_k = \mathbf{I}_M$), such as SVRG and L-SVRG algorithms.

Lemma 6. *Under the proposed SPP framework, we have for $k > 0$*

$$X_{k+1} = \left(\mathbf{I}_M - \Lambda_{k+1}\right)X_k + \Lambda_{k+1}\left(\hat{X}_{k+1} \otimes \mathbf{1}_m\right), \tag{54}$$

and

$$\nabla F\left(X_k\right) = \left(\mathbf{I}_M - \Lambda_k\right)\nabla F\left(X_{k-1}\right) + \Lambda_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right). \tag{55}$$

Proof. By (8), (9), we have

$$\begin{aligned}
X_{k+1} &= \left(\mathbf{I}_M - \Lambda_{k+1}\right)X_k + \Lambda_{k+1}\left(W_k \otimes \mathbf{1}\mathbf{1}^T\right)\frac{\Lambda_k}{b_k}\left(X_k - \alpha Y_k\right) \\
&= \left(\mathbf{I}_M - \Lambda_{k+1}\right)X_k + \Lambda_{k+1}\left(W_k \otimes \mathbf{1}_m\right)\left(\mathbf{I}_n \otimes \mathbf{1}_m^T\right)\frac{\Lambda_k}{b_k}\left(X_k - \alpha Y_k\right) \\
&= \left(\mathbf{I}_M - \Lambda_{k+1}\right)X_k + \Lambda_{k+1}\left(\hat{X}_{k+1} \otimes \mathbf{1}_m\right),
\end{aligned}$$

where in the second equality we have used the property of Kronecker product that $\left(A \otimes B\right)\left(C \otimes D\right) = AC \otimes BD$. In what follows, by noticing that $\Lambda_k \nabla F\left(X_k\right)$ represents selecting the rows of $\nabla F\left(X_k\right)$ corresponding to the randomly selected samples at iteration k by Λ_k, we have

$$\begin{aligned}
\nabla F\left(X_k\right) &= \nabla F\left(\left(\mathbf{I}_M - \Lambda_k\right)X_{k-1} + \Lambda_k\left(\hat{X}_k \otimes \mathbf{1}_m\right)\right) \\
&= \left(\mathbf{I}_M - \Lambda_k + \Lambda_k\right)\nabla F\left(\left(\mathbf{I}_M - \Lambda_k\right)X_{k-1} + \Lambda_k\left(\hat{X}_k \otimes \mathbf{1}_m\right)\right) \\
&= \Lambda_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right) + \left(\mathbf{I}_M - \Lambda_k\right)\nabla F\left(X_{k-1}\right),
\end{aligned}$$

which completes the proof. □

Lemma 6 illustrates the update rule of decision variables over the augmented graph, that is, only the sampled nodes (samples) perform update, while other nodes that are not sampled remain unchanged.

Lemma 7. *Suppose Assumption 4 and 5 hold. Then we have for all $k > 0$*

$$\mathbb{E}\left[\bar{y}_k|\mathcal{F}_k\right] = r\nabla f\left(\bar{x}_k\right) + \left(1-r\right) g_k, \tag{56}$$

where

$$g_k = \frac{\mathbf{1}_M^T}{M}\nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right), \tag{57}$$

and $r = P\left(W_k = \mathbf{J}_m\right)$ denotes the probability of performing global averaging at each iteration.

Proof. By the definition of $\bar{y}_k$ in (22), we have $\bar{y}_k = \frac{\mathbf{1}_n^T}{n} S_k Y_k$, and

$$\begin{aligned}
\mathbb{E}\left[\bar{y}_k|\mathcal{F}_k\right] \overset{(7b)}{=}& \mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k\left(C_{k-1}Y_{k-1} + \nabla F\left(X_k\right) - \nabla F\left(X_{k-1}\right)\right)|\mathcal{F}_k\right] \\
=& \mathbb{E}\left[\left(\frac{\mathbf{1}_M^T}{M} Y_{k-1} + \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_k\right) - \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_{k-1}\right)\right)|\mathcal{F}_k\right] \\
\overset{(11)}{=}& \mathbb{E}\left[\left(\frac{\mathbf{1}_M^T}{M} \nabla F\left(X_{k-1}\right) + \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_k\right) - \frac{\mathbf{1}_M^T}{M} \nabla F\left(X_{k-1}\right)\right)|\mathcal{F}_k\right] \\
=& \mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_k\right)|\mathcal{F}_k\right],
\end{aligned} \tag{58}$$

where in the second equality we have used the fact that $\mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k\right] = \frac{\mathbf{1}_n^T}{n}\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right) = \frac{\mathbf{1}_M^T}{M}$ due to Assumption 4 and C_{k-1} is doubly-stochastic. Then, by the Eq. (55) in Lemma 6, we obtain

$$\begin{aligned}
\mathbb{E}\left[\bar{y}_k|\mathcal{F}_k\right] &= \mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k\left[\nabla F\left(\Lambda_k\left(\hat{X}_k \otimes \mathbf{1}_m\right) + \left(\mathbf{I}_M - \Lambda_k\right) X_{k-1}\right)\right]|\mathcal{F}_k\right] \\
&= \mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k\left[\Lambda_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right) + \left(\mathbf{I}_M - \Lambda_k\right)\nabla F\left(X_{k-1}\right)\right]|\mathcal{F}_k\right] \\
&= \mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right)|\mathcal{F}_k\right],
\end{aligned} \tag{59}$$

where in the last equality we have used the fact that $\Lambda_k\left(\mathbf{I}_M - \Lambda_k\right) = 0$. Therefore, by the law of total probability, we obtain

$$\begin{aligned}
\mathbb{E}\left[\bar{y}_k|\mathcal{F}_k\right] &= r\mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right)|\mathcal{F}_k, W_k = \mathbf{J}_n\right] + \left(1-r\right)\mathbb{E}\left[\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right)|\mathcal{F}_k, W_k = W\right] \\
&= r\nabla f\left(\bar{x}_k\right) + \left(1-r\right)\frac{\mathbf{1}_M^T}{M}\nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right),
\end{aligned}$$

which completes the proof. □

Lemma 7 shows that the expectation of $\bar{y}_k$ is a linear combination of the full gradient evaluated at the averaged decision variable and that evaluated at the decision variable of proxy nodes.

In the next lemma, we define an error term σ_k (resembling internal variance) which can be bounded by the variance reduction errors, and establish the upper bounds under three particular settings discussed in the main text.

Lemma 8. *Suppose Assumption 1-5 hold. Denote*

$$\sigma_k := n\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(\mathbf{1}_M x^*\right) + \frac{\mathbf{1}_n^T}{n} S_k C_{t_k} Y_{t_k} - \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_{t_k}\right) - \nabla f\left(x^*\right)\right\|^2 |\mathcal{F}_k\right], \tag{60}$$

where $t_k < k$ represents the latest iteration before k that $C_{t_k} \neq \mathbf{I}_M$. Then we have, for all $k > 0$, if we choose $C_k \equiv \mathbf{I}_M$ and $b_k = b$,

$$\sigma_k \leqslant \frac{\sigma^*}{b}, \tag{61}$$

else if $C_k = \mathbf{I}_n \otimes V_k$ *with* $p > 0$ *holds,*

$$\begin{aligned}\sigma_k \leqslant & \frac{1-p}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{t_{k-1}}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{I}_m\right] \\ & + \frac{p}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right],\end{aligned} \tag{62}$$

else if $C_k = W_k \otimes \mathbf{J}_m$ *holds,*

$$\sigma_k \leqslant \frac{1}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right]. \tag{63}$$

Proof. By the definition of σ_k and t_k, for $C_k \equiv \mathbf{I}_M$ in the first case, which implies $t_k = 0$, we obtain by noticing $Y_0 = \nabla F\left(X_0\right)$ and $b_k = b$ that

$$\begin{aligned}\sigma_k &= \frac{1}{n}\mathbb{E}\left[\left\|\mathbf{1}_n^T S_k \nabla F\left(\mathbf{1}_M x^*\right) - \mathbf{1}_n^T\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right] \\ &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left(\left\|\frac{1}{b_k}\sum_{j\in\xi_{i,k}}\left(\nabla f_{ij}\left(x^*\right) - \nabla f_i\left(x^*\right)\right)\right\|^2 |\mathcal{F}_k\right) \\ &\leqslant \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left(\frac{1}{b_k^2}\sum_{j\in\xi_{i,k}}\left\|\nabla f_{ij}\left(x^*\right) - \nabla f_i\left(x^*\right)\right\|^2 |\mathcal{F}_k\right) \leqslant \frac{\sigma^*}{b},\end{aligned} \tag{64}$$

where we used the fact that $\mathbb{E}\left[\nabla f_{ij}\left(x^*\right) - \nabla f_i\left(x^*\right)\right] = 0$, $j \in \xi_i$.

For the case that $C_k = \mathbf{I}_n \otimes V_k$ with $p > 0$, recalling that C_k is column-stochastic, then we have

$$\begin{aligned}\sigma_k &= n\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n}\left(\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_{t_k}\right) - S_k\nabla F\left(X_{t_k}\right) + S_k\nabla F\left(\mathbf{1}_M x^*\right) - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2 |\mathcal{F}_k\right] \\ &= \frac{1}{n}\mathbb{E}\left[\left\|\left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(X_{t_k}\right) - S_k\nabla F\left(X_{t_k}\right) + S_k\nabla F\left(\mathbf{1}_M x^*\right) - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right],\end{aligned} \tag{65}$$

where in the second equality we used $\mathbb{E}\left[S_k\right] = \mathbf{I}_n \otimes \frac{\mathbf{1}_m^T}{m}$ induced by Assumption 4. Furthermore, noticing the fact that $\mathbb{E}\left[\left\|X - \mathbb{E}\left[X\right]\right\|^2\right] \leqslant \mathbb{E}\left[\left\|X\right\|^2\right]$, we get

$$\begin{aligned}\sigma_k &\leqslant \frac{1}{n}\mathbb{E}\left[\left\|S_k\nabla F\left(X_{t_k}\right) - S_k\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right] \\ &\leqslant \frac{1-p}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{t_{k-1}}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{I}_m\right] \\ &\quad + \frac{p}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right],\end{aligned} \tag{66}$$

where in the last inequality we have used the law of total probability.

For $C_k = W_k \otimes \mathbf{J}_m$ in the third case, we have $p = 1$ and

$$\sigma_k \leqslant \frac{1}{M}\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right].$$

which completes the proof. □

Lemma 8 shows that the error term σ_k is related to the optimization error when the variance-reduction methods are adopted (see Lemma 12 and 13), otherwise, there will be a fixed upper bound on σ_k as defined in Assumption 2.

Now we are going to bound each error term in the Lyapunov function (23) and then combine them together to obtain a recursion of the constructed Lyapunov function with proper choice of $c_0 - c_4$.

B.2. The Recursion of Optimality Gap

Lemma 9. *Suppose Assumption 1-5 hold. Then, we have for all $k > 0$,*

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{x}_{k+1}-x^*\right\|^2|\mathcal{F}_k\right] \leqslant (1-\alpha\mu)\left\|\bar{x}_k-x^*\right\|^2 - \left(2\alpha-8\alpha^2L\right)\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) \\ + (1-r)\frac{\left(4\alpha^2L^2+\alpha L\right)}{n}\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2 + \frac{2\alpha^2\sigma_k}{n}.\end{aligned} \tag{67}$$

Proof. Using (22), we have

$$\bar{x}_{k+1} = \bar{x}_k - \alpha\bar{y}_k,$$

which implies that

$$\left\|\bar{x}_{k+1}-x^*\right\|^2 = \left\|\bar{x}_k-x^*\right\|^2 - 2\alpha\left\langle\bar{y}_k,\bar{x}_k-x^*\right\rangle + \left\|\bar{y}_k\right\|^2.$$

Then by lemma 6 and 7, we get

$$\begin{aligned}&\mathbb{E}\left[\left\|\bar{x}_{k+1}-x^*\right\|^2|\mathcal{F}_k\right] \\ &\leqslant \left\|\bar{x}_k-x^*\right\|^2 - 2\alpha\mathbb{E}\left[\left\langle\bar{y}_k,\bar{x}_k-x^*\right\rangle|\mathcal{F}_k\right] + \alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right] \\ &= \left\|\bar{x}_k-x^*\right\|^2 - 2\alpha\left\langle r\nabla f\left(\bar{x}_k\right)+(1-r)\,g_k,\bar{x}_k-x^*\right\rangle + \alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right] \\ &\overset{(a)}{\leqslant} (1-\alpha\mu r)\left\|\bar{x}_k-x^*\right\|^2 - 2\alpha r\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) - 2\alpha(1-r)\left\langle g_k,\bar{x}_k-x^*\right\rangle + \alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right] \\ &\overset{(b)}{\leqslant} (1-\alpha\mu)\left\|\bar{x}_k-x^*\right\|^2 - 2\alpha\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) + \frac{\alpha L(1-r)}{n}\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2 + \alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right],\end{aligned}$$

where in the inequality (a) we have used the fact that f is μ-strongly convex:

$$-2\alpha\left\langle r\nabla f\left(\bar{x}_k\right),\bar{x}_k-x^*\right\rangle \leqslant -2r\alpha\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) - r\alpha\mu\left\|\bar{x}_k-x^*\right\|^2,$$

and for the inequality (b), noticing that $\hat{X}_k := S_kX_k = \left[\hat{x}_{1,k}^T,\hat{x}_{2,k}^T,\cdots,\hat{x}_{n,k}^T\right]^T \in \mathbb{R}^{n\times d}$, and the fact that each f_i is μ-strongly convex and L-smooth, we have:

$$\begin{aligned}&-2\alpha(1-r)\left\langle g_k,\bar{x}_k-x^*\right\rangle \\ &= -2\alpha(1-r)\left\langle\frac{\mathbf{1}_n^T}{n}\left(\mathbf{I}_n\otimes\frac{\mathbf{1}_m^T}{m}\right)\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right),\bar{x}_k-x^*\right\rangle \\ &= -2\alpha(1-r)\frac{1}{n}\sum_{i=1}^n\left\langle\nabla f_i\left(\hat{x}_{i,k}\right),\bar{x}_k-x^*\right\rangle \\ &= -2\alpha(1-r)\frac{1}{n}\sum_{i=1}^n\left(\left\langle\nabla f_i\left(\hat{x}_{i,k}\right),\hat{x}_{i,k}-x^*\right\rangle + \left\langle\nabla f_i\left(\hat{x}_{i,k}\right),\bar{x}_k-\hat{x}_{i,k}\right\rangle\right) \\ &\leqslant 2\alpha(1-r)\frac{1}{n}\sum_{i=1}^n\left(f_i\left(x^*\right)-f_i\left(\hat{x}_{i,k}\right)-\frac{\mu}{2}\left\|\hat{x}_{i,k}-x^*\right\|^2\right) \\ &+ 2\alpha(1-r)\frac{1}{n}\sum_{i=1}^n\left(f_i\left(\hat{x}_{i,k}\right)-f_i\left(\bar{x}_k\right)+\frac{L}{2}\left\|\hat{x}_{i,k}-\bar{x}_k\right\|^2\right) \\ &= -2\alpha(1-r)\left[f\left(\bar{x}_k\right)-f\left(x^*\right)\right] - \alpha\mu(1-r)\left\|\bar{x}_k-x^*\right\| + \frac{\alpha L(1-r)}{n}\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2.\end{aligned}$$

Further, we can bound the last term in the inequality (b) (note that $\nabla f(x^*) = 0$)

$$\begin{aligned}
&\mathbb{E}\left\|\bar{y}_k - \nabla f\left(x^*\right)|\mathcal{F}_k\right\|^2 \\
&\overset{(22)}{=} \mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_k\right) + \frac{\mathbf{1}_n^T}{n} S_k C_{k-1} Y_{k-1} - \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_{k-1}\right) - \nabla f\left(x^*\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant 2\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left[\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M \bar{x}_k\right) + \nabla F\left(\mathbf{1}_M \bar{x}_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|^2 |\mathcal{F}_k\right] \\
&+ \underbrace{2\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(\mathbf{1}_M x^*\right) + \frac{\mathbf{1}_n^T}{n} S_k C_{t_k} Y_{t_k} - \frac{\mathbf{1}_n^T}{n} S_k \nabla F\left(X_{t_k}\right) - \nabla f\left(x^*\right)\right\|^2 |\mathcal{F}_k\right]}_{:=\sigma_k/n} \\
&\leqslant \underbrace{4\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left(\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M \bar{x}_k\right)\right)\right\|^2 |\mathcal{F}_k\right]}_{S_1} \\
&+ \underbrace{4\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left(\nabla F\left(\mathbf{1}_M \bar{x}_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2 |\mathcal{F}_k\right]}_{S_2} + \frac{2\sigma_k}{n}.
\end{aligned} \tag{68}$$

In what follows, under Assumption 3, we bound the terms S_1 and S_2 respectively. For S_1, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left(\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M \bar{x}_k\right)\right)\right\|^2 |\mathcal{F}_k\right] \\
&\overset{(54)}{=} \mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left(\nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right) - \nabla F\left(\mathbf{1}_M \bar{x}_k\right)\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant \frac{1}{n}\sum_{i=1}^n \frac{1}{m}\sum_{j=1}^m \left\|\nabla f_{ij}\left(\hat{x}_{i,k}\right) - \nabla f_{ij}\left(\bar{x}_k\right)\right\|^2 \overset{(6)}{\leqslant} \frac{L^2}{n}\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2,
\end{aligned} \tag{69}$$

and for S_2, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{\mathbf{1}_n^T}{n} S_k \left(\nabla F\left(\mathbf{1}_M \bar{x}_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant \frac{1}{n}\sum_{i=1}^n \frac{1}{m}\sum_{j=1}^m \left\|\nabla f_{ij}\left(\bar{x}_k\right) - \nabla f_{ij}\left(x^*\right)\right\|^2 \overset{(6)}{\leqslant} 2L\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right).
\end{aligned} \tag{70}$$

Then, combining (68), (69) and (70), we get

$$\mathbb{E}\left\|\bar{y}_k - \nabla f\left(x^*\right)|\mathcal{F}_k\right\|^2 \leqslant 4\left(1-r\right)\frac{L^2}{n}\rho_W\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 + 8L\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right) + \frac{2\sigma_k}{n}. \tag{71}$$

which completes the proof. □

B.3. The Recursion of Consensus Error

For simplicity, we first recall the following notations:

$$r = P\left(W_k = \mathbf{J}_m\right), \quad \rho_{r,W} = \left(1-r\right)\rho_W, \quad \rho_W = \left\|W - \mathbf{J}_n\right\|^2.$$

Then, we bound the consensus error for two types of algorithms respectively. In particular, the results for the algorithms adopting only variance-reduction schemes are provided in Lemma 10 while the results for those adopting both variance-reduction and gradient-tracking schemes are given in Lemma 11.

Lemma 10. *Suppose Assumption 1-5 hold. Then, if we set $C_k = \mathbf{I}_n \otimes V_k$, we have for all $k > 0$*

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant \left(\frac{1+\rho_{r,W}}{2} + \frac{4\alpha^2 L^2 \rho_{r,W}\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}}\right)\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 \\
&+ \frac{8n\alpha^2 L \rho_{r,W}\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}}\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right) + \alpha^2 \rho_{r,W}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + n\sigma_k\right).
\end{aligned} \tag{72}$$

Proof. First of all, recalling the definition of $\hat{X}_k$ in (13), $\bar{x}_k$ in (22) and $\rho_{r,W} := (1-r)\,\rho_W$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2 |\mathcal{F}_k\right] \\
&= (1-r)\,\mathbb{E}\left[\left\|W_k\left[\hat{X}_k - \alpha\hat{Y}_k\right] - \mathbf{1}_n\left(\bar{x}_k - \alpha\bar{y}_k\right)\right\|^2 |\mathcal{F}_k, W_k = W\right] \\
&+ r\mathbb{E}\left[\left\|W\left[\hat{X}_k - \alpha\hat{Y}_k\right] - \mathbf{1}_n\left(\bar{x}_k - \alpha\bar{y}_k\right)\right\|^2 |\mathcal{F}_k, W_k = \mathbf{J}_n\right] \\
&= (1-r)\,\mathbb{E}\left[\left\|W\left[\hat{X}_k - \alpha\hat{Y}_k\right] - \mathbf{1}_n\left(\bar{x}_k - \alpha\bar{y}_k\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant (1-r)\left\|W - \mathbf{J}_n\right\|_2^2 \mathbb{E}\left[\left\|\left(\hat{X}_k - \mathbf{1}_n\bar{x}_k\right) - \alpha S_k Y_k\right\|^2 |\mathcal{F}_k\right]. \\
&= \rho_{r,W}\mathbb{E}\left[\left\|\left(\hat{X}_k - \mathbf{1}_n\bar{x}_k\right) - \alpha S_k Y_k\right\|^2 |\mathcal{F}_k\right].
\end{aligned} \tag{73}$$

Then, by Lemma 5, we get

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant \rho_{r,W}\mathbb{E}\left[\left\|\left(\hat{X}_k - \mathbf{1}_n\bar{x}_k\right) - \alpha S_k\left(C_{t_k}Y_{t_k} + \nabla F\left(X_k\right) - \nabla F\left(X_{t_k}\right)\right)\right\|^2 |\mathcal{F}_k\right] \\
&\overset{(c)}{\leqslant} \rho_{r,W}\mathbb{E}\left[\left\|\left(\hat{X}_k - \mathbf{1}_n\bar{x}_k\right) - \alpha\left(S_k\nabla F\left(X_k\right) - S_k\nabla F\left(\mathbf{1}_M x^*\right) + \left(\mathbf{I}_n \otimes \frac{\mathbf{1}^T}{m}\right)\nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2 |\mathcal{F}_k\right] \\
&+ \alpha^2\rho_{r,W}\underbrace{\mathbb{E}\left[\left\|S_k\left(C_{t_k}Y_{t_k} - \nabla F\left(X_{t_k}\right)\right) + S_k\nabla F\left(\mathbf{1}_M x^*\right) - \left(\mathbf{I}_n \otimes \frac{\mathbf{1}^T}{m}\right)\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right]}_{:=n\sigma_k} \\
&\overset{(d)}{\leqslant} \rho_{r,W}\left(1+\beta\right)\left\|\hat{X}_k - \mathbf{1}_n\bar{x}_k\right\|^2 + 2\alpha^2\rho_{r,W}\left(1+\frac{1}{\beta}\right)\left(\mathbb{E}\left[\left\|S_k\nabla F\left(X_k\right) - S_k\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right]\right) \\
&+ 2\alpha^2\rho_{r,W}\left(1+\frac{1}{\beta}\right)n\zeta^* + \alpha^2\rho_{r,W}n\sigma_k,
\end{aligned} \tag{74}$$

where $t_k < k$ denotes the latest iteration before k that $V_{t_k} = \mathbf{J}_m$, and thus $C_{t_k} = \mathbf{I}_n \otimes \mathbf{J}_m$. Besides, in the inequality (c) we have used the fact that $\mathbb{E}\left[\left\|X - \mathbb{E}\left[X\right]\right\|^2\right] \leqslant \mathbb{E}\left[\left\|X\right\|^2\right]$, and in the inequality (d) we have used Young inequality with parameter $\beta = \frac{1-\rho_{r,W}}{2\rho_{r,W}}$. Furthermore, noticing that

$$\begin{aligned}
&\mathbb{E}\left[\left\|S_k\nabla F\left(X_k\right) - S_k\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right] \\
&= \mathbb{E}\left[\left\|S_k\nabla F\left(X_k\right) - S_k\nabla F\left(\mathbf{1}_M\bar{x}_k\right) + S_k\nabla F\left(\mathbf{1}_M\bar{x}_k\right) - S_k\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant 2L^2\left\|\hat{X}_k - \mathbf{1}_n\bar{x}_k\right\|^2 + 4nL\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right),
\end{aligned} \tag{75}$$

we thus complete the proof of the lemma. □

Lemma 11. *Suppose Assumption 1-5 hold. Then, if we set $C_k = W_k \otimes \mathbf{J}_m$, we have for all $k > 0$*

$$\mathbb{E}\left[\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2 |\mathcal{F}_k\right] \leqslant \rho_{r,W}\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 + \frac{\alpha^2 \rho_{r,W}\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}} \mathbb{E}\left[\left\|\hat{Y}_k - \mathbf{1}_n \bar{y}_k\right\|^2 |\mathcal{F}_k\right]. \tag{76}$$

Proof. Recalling the definition of $\hat{X}_k$ in (13) and $\bar{x}_k$ in (22), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{X}_{k+1} - \mathbf{1}_n \bar{x}_{k+1}\right\|^2 |\mathcal{F}_k\right] \\
&= (1-r)\,\mathbb{E}\left[\left\|\left(W_k - \mathbf{J}_n\right)\left(\hat{X}_k - \mathbf{1}_n \bar{x}_k\right) - \alpha\left(W - \mathbf{J}_n\right)\left(\hat{Y}_k - \mathbf{1}_n \bar{y}_k\right)\right\|^2 |\mathcal{F}_k\right] \\
&\leqslant \rho_W\,(1-r)\,(1+\beta)\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 + \alpha^2 \rho_W\,(1-r)\left(1+\frac{1}{\beta}\right)\mathbb{E}\left[\left\|\hat{Y}_k - \mathbf{1}_n \bar{y}_k\right\|^2 |\mathcal{F}_k\right],
\end{aligned} \tag{77}$$

where in the last inequality we have used Young inequality. Setting $\beta = \frac{1-\rho_{r,W}}{2\rho_{r,W}}$ completes the proof. □

B.4. The Recursion of Variance Reduction Error

In the following lemma, we show that the VR error will be decaying when the variance-reduction methods such as SAGA, L-SVRG, are adopted. For ease of presentation, we recall the notions as follows:

$$q := \rho\left(\mathbb{E}\left[\Lambda_{k+1} | V_k = \mathbf{J}_m\right]\right), \quad r := P\left(W_k = \mathbf{J}_n\right).$$

Lemma 12. *Suppose Assumption 1-5 hold. Then, we have for all $k > 0$*

$$\begin{aligned}
&\mathbb{E}\left[\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M x^*\right) |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&\leqslant (1-q)\,\mathbb{E}\left[\left\|\left[\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&+ 2\,(1-r)\,mqL^2\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 + 4MqL\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right).
\end{aligned} \tag{78}$$

Proof. Using Lemma 5, we have

$$\nabla F\left(X_k\right) = \left(\mathbf{I}_M - \Lambda_k\right)\nabla F\left(X_{k-1}\right) + \Lambda_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right).$$

Then, we further obtain

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F\left(X_k\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&= \mathbb{E}\left[\left\|\left(\mathbf{I}_M - \Lambda_k\right)\nabla F\left(X_{k-1}\right) + \Lambda_k \nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&= \mathbb{E}\left[\left\|\ \left(\mathbf{I}_M - \Lambda_k\right)\left[\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right]\ \right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&+ \mathbb{E}\left[\left\|\Lambda_k\left[\nabla F\left(\hat{X}_k \otimes \mathbf{1}_m\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&\overset{(69),(70)}{\leqslant} (1-q)\,\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right) - \nabla F\left(\mathbf{1}_M x^*\right)\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] \\
&+ 2\,(1-r)\,mqL^2 \mathbb{E}\left[\left\|\hat{X}_k - \mathbf{1}_n \bar{x}_k\right\|^2 |\mathcal{F}_k, V_{k-1} = \mathbf{J}_m\right] + 4MqL\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right).
\end{aligned}$$

□

B.5. The Recursion of Delay Error of Variance Reduction

When the variance-reduction methods with random local loops, such as L-SVRG, are adopted, an additional delay error term will appear and should be dealt with properly for convergence analysis.

Lemma 13. *Suppose Assumption 1-5 hold. Then, we have for all $k > 0$,*

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F\left(X_{t_k}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant (1-p)\,\mathbb{E}\left[\left\|\nabla F\left(X_{t_{k-1}}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k, V_{k-1}=\mathbf{I}_m\right] \\
&+ p\mathbb{E}\left[\left\|\left[\nabla F\left(X_{k-1}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|^2|\mathcal{F}_k, V_{k-1}=\mathbf{J}_m\right].
\end{aligned} \tag{79}$$

The proof for the above lemma is straightforward leveraging the definition of t_k as well as the conditional expectation on the choices of $V_k - 1$, which is thus omitted.

B.6. The Recursion of Gradient Tracking Error

The following lemma shows that the error caused by the data heterogeneity across devices is decaying with iteration when gradient-tracking schemes are adopted.

Lemma 14. *Suppose Assumption 1-5 hold, if we set $C_k = W_k \otimes \mathbf{J}_m$, then for all $k > 0$*

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{Y}_{k+1}-\mathbf{1}_n\bar{y}_{k+1}\right\|^2|\mathcal{F}_k\right] \\
&\leqslant \left(\frac{1+\rho_{r,W}}{2}+\frac{8\alpha^2L^2\rho_{r,W}\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}}\right)\mathbb{E}\left[\left\|Y_k-\mathbf{1}_M\bar{y}_k\right\|^2|\mathcal{F}_k\right] \\
&+\frac{4\left(1+\rho_{r,W}\right)L^2}{1-\rho_{r,W}}\left(4+(1-r)\,q+4\left(1-r\right)\alpha^2L^2\right)\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2 \\
&+\frac{2\left(1+\rho_{r,W}\right)}{m\left(1-\rho_{r,W}\right)}\left(1-q+\frac{4\alpha^2L^2}{n}\right)\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|_2^2|\mathcal{F}_k\right] \\
&+\frac{4n\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}}\left(8\alpha^2L^3+4L+2qL\right)\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right).
\end{aligned} \tag{80}$$

Proof. Recalling the definition of $\hat{Y}_k$ in (13) and $\bar{y}_k$ in (22), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\hat{Y}_{k+1}-\mathbf{1}_n\bar{y}_{k+1}\right\|^2|\mathcal{F}_k\right] \\
&=\mathbb{E}\left[\left\|\left(\mathbf{I}_n-\mathbf{J}_n\right)\left(W_k\hat{Y}_k+S_{k+1}\left(\nabla F\left(X_{k+1}\right)-\nabla F\left(X_k\right)\right)\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant \frac{1+\rho_{r,W}}{2}\mathbb{E}\left[\left\|\hat{Y}_k-\mathbf{1}_n\bar{y}_k\right\|^2|\mathcal{F}_k\right]+\frac{1+\rho_{r,W}}{1-\rho_{r,W}}\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(X_{k+1}\right)-\nabla F\left(X_k\right)\right)\right\|^2|\mathcal{F}_k\right],
\end{aligned}$$

where we have used the Young inequality. Then, we first bound the last term:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(X_{k+1}\right)-\nabla F\left(X_k\right)\right)\right\|^2|\mathcal{F}_k\right] \\
&=\mathbb{E}\left[\left\|S_{k+1}\nabla F\left(X_{k+1}\right)-S_{k+1}\nabla F\left(\mathbf{1}_M x^*\right)+S_{k+1}\nabla F\left(\mathbf{1}_M x^*\right)-S_{k+1}\nabla F\left(X_k\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant 2\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(\hat{X}_{k+1}\otimes\mathbf{1}_m\right)-\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)+\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2|\mathcal{F}_k\right] \\
&+\frac{2}{m}\mathbb{E}\left[\left\|\nabla F\left(X_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant 4L^2\underbrace{\mathbb{E}\left[\left\|\hat{X}_{k+1}-\hat{X}_k\right\|^2|\mathcal{F}_k\right]}_{S_3}+\underbrace{4\mathbb{E}\left[\left\|S_{k+1}\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)-S_{k+1}\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k\right]}_{S_4} \\
&+\frac{2}{m}\mathbb{E}\left[\left\|\nabla F\left(X_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k\right],
\end{aligned}
$$

In what follows, we further bound the terms of S_3 and S_4, respectively. For S_3, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\hat{X}_{k+1}-\hat{X}_k\right\|^2|\mathcal{F}_k\right] \\
&=\mathbb{E}\left[\left\|W_k\hat{X}_k-\hat{X}_k-\alpha W_k\hat{Y}_k\right\|^2|\mathcal{F}_k\right] \\
&\leqslant\mathbb{E}\left[\left\|W_k\hat{X}_k-\hat{X}_k\right\|^2|\mathcal{F}_k\right]+\alpha^2\mathbb{E}\left[\left\|W_k\hat{Y}_k-\mathbf{1}_n\bar{y}_k+\mathbf{1}_n\bar{y}_k\right\|^2|\mathcal{F}_k\right]-2\alpha\left\langle W_k\hat{X}_k-\hat{X}_k,W_k\hat{Y}_k\right\rangle \\
&=\mathbb{E}\left[\left\|W_k\hat{X}_k-\hat{X}_k\right\|^2|\mathcal{F}_k\right]+\alpha^2\mathbb{E}\left[\left\|W_k\hat{Y}_k-\mathbf{1}_n\bar{y}_k\right\|^2|\mathcal{F}_k\right] \\
&+n\alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right]-2\alpha\left\langle W_k\hat{X}_k-\hat{X}_k,W_k\hat{Y}_k-\mathbf{1}_n\bar{y}_k\right\rangle \\
&\leqslant 2\mathbb{E}\left[\|W_k-\mathbf{I}_n\|\right]_2^2\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2+2\alpha^2\rho_{r,W}\mathbb{E}\left[\left\|\hat{Y}_k-\mathbf{1}_n\bar{y}_k\right\|^2|\mathcal{F}_k\right]+n\alpha^2\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right].
\end{aligned}
$$

Then, for S_4, recalling the relations in (69) and (70), we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|S_{k+1}\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)-S_{k+1}\nabla F\left(\mathbf{1}_M x^*\right)\right\|^2|\mathcal{F}_k\right] \\
&=\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)-\nabla F\left(\mathbf{1}_M\bar{x}_k\right)+\nabla F\left(\mathbf{1}_M\bar{x}_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant 2\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(\hat{X}_k\otimes\mathbf{1}_m\right)-\nabla F\left(\mathbf{1}_M\bar{x}_k\right)\right)\right\|^2|\mathcal{F}_k\right]+2\mathbb{E}\left[\left\|S_{k+1}\left(\nabla F\left(\mathbf{1}_M\bar{x}_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right)\right\|^2|\mathcal{F}_k\right] \\
&\leqslant 2L^2\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2+4nL\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right).
\end{aligned}
$$

Combining all these inequalities above, together with the inequality of $\mathbb{E}\left[\left\|\bar{y}_k\right\|^2|\mathcal{F}_k\right]$ in (68), we complete the proof. □

Remark 7. *It follows from 14, together with Lemma 10 with Lemma 11, that the error caused by the data heterogeneity across devices is decaying when gradient-tracking methods are adopted, otherwise the heterogeneity constant ζ^* as defined in Assumption 2 will not be eliminated.*

C. Proofs in Section 5

In this section, we provide the proofs of Theorem 1-3, followed by the corresponding Corollary 1-3 for the strongly convex and smooth objectives.

C.1. Proof of Theorem 1

Proof. For algorithms $\mathcal{A}(\cdot,\cdot,C_k\equiv\mathbf{I}_M)$, recalling the following parameter settings of Lyapunov function (23)

$$c_0=1,\ c_1=\frac{8(1-r)\alpha L(4\alpha L+1)}{n(1-\rho_{r,W})},\ c_2=c_3=c_4=0,$$

and using Lemma 8, 9, 10, we can derive by properly rearranging terms that

$$\begin{aligned}
\mathbb{E}[T_{k+1}] &= c_0\mathbb{E}\left[\|\bar{x}_{k+1}-x^*\|^2\right]+c_1\mathbb{E}\left[\left\|\hat{X}_{k+1}-\mathbf{1}_n\bar{x}_{k+1}\right\|^2\right]\\
&\leqslant (1-\alpha\mu)c_0\mathbb{E}\left[\|\bar{x}_k-x^*\|_2^2\right]+\left(1-\frac{1-\rho_{r,W}}{8}\right)\mathbb{E}\left[\left\|\hat{X}_k-\mathbf{1}_n\hat{x}_k\right\|^2\right]\\
&+\underbrace{\left((1-r)\frac{64\alpha^3L^2(4\alpha L+1)\rho_{r,W}(1+\rho_{r,W})}{(1-\rho_{r,W})^2}-\left(2\alpha-8\alpha^2L\right)\right)}_{e_1}\mathbb{E}\left[f(\bar{x}_k)-f(x^*)\right]\\
&+\underbrace{\left((1-r)\frac{\left(4\alpha^2L^2+\alpha L\right)}{n}-(1-r)\frac{8\alpha L(4\alpha L+1)}{n(1-\rho_{r,W})}\frac{1-\rho_{r,W}}{8}\right)}_{e_2}\mathbb{E}\left[\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2\right]\\
&+\frac{2\alpha^2\sigma^*}{nb}+(1-r)\frac{8\alpha^3L(4\alpha L+1)\rho_{r,W}}{(1-\rho_{r,W})}\left(\frac{4\zeta^*}{1-\rho_{r,W}}+\frac{\sigma^*}{b}\right).
\end{aligned}\tag{81}$$

It can be thus verified that if the step-size satisfies

$$\alpha\leqslant\min\left\{\frac{1}{5L},\frac{1-\rho_{r,W}}{4L\sqrt{\rho_{r,W}(1+\rho_{r,W})}},\frac{1-\rho_{r,W}}{12L\sqrt{2\rho_{r,W}(1+\rho_{r,W})}}\right\}=\min\left\{\mathcal{O}\left(\frac{1}{L}\right),\mathcal{O}\left(\frac{1-\rho_{r,W}}{L\sqrt{\rho_{r,W}}}\right)\right\},\tag{82}$$

then the coefficient $e_0,e_1\leqslant 0$, and we thus have

$$\mathbb{E}[T_{k+1}]\leqslant\left(1-\min\left\{\alpha\mu,\frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}[T_k]+\frac{2\alpha^2\sigma^*}{nb}+(1-r)\frac{16\alpha^3L\rho_{r,W}}{1-\rho_{r,W}}\left(\frac{4\zeta^*}{1-\rho_{r,W}}+\frac{\sigma^*}{b}\right),$$

which completes the proof. □

C.2. Proof of Corollary 1

Proof. According to Theorem 1, we have

$$\mathbb{E}[T_k]\leqslant\left(1-\min\left\{\alpha\mu,1-\frac{1-\rho_{r,W}}{8}\right\}\right)^k\mathbb{E}[T_0]+\frac{1}{\min\left\{\alpha\mu,\frac{1-\rho_{r,W}}{8}\right\}}\left(\alpha^2D_1+\alpha^3D_2\right),\tag{83}$$

where

$$D_1=\frac{2\sigma^*}{nb},\ D_2=(1-r)\frac{8L\rho_{r,W}(4\alpha L+1)}{1-\rho_{r,W}}\left(\frac{4\zeta^*}{1-\rho_{r,W}}+\frac{\sigma^*}{b}\right).$$

Let γ denote the resulting upper bound on step size α in (82), and give another extra upper bound as follow:

$$\alpha\leqslant\min\left\{\gamma,\frac{\ln\left(\max\left\{2,\frac{\mu^2K^2}{D_1},\frac{\mu^3K^3}{D_2}\right\}\right)}{\mu K}\right\},\tag{84}$$

Then, we consider two possible situations. First, if $\gamma \leqslant \frac{\ln\left(\max\left\{2, \min\left\{\frac{\mu^2K^2}{D_1}, \frac{\mu^3K^3}{D_2}\right\}\right\}\right)}{\mu K}$ holds, then we let $\alpha = \gamma$, and obtain

$$\begin{aligned}
\mathbb{E}\left[T_k\right] &\leqslant \left(1 - \min\left(\gamma\mu, \frac{1-\rho_{r,W}}{8}\right)\right)^K \mathbb{E}\left[T_0\right] + \frac{1}{\min\left(\gamma\mu, \frac{1-\rho_{r,W}}{8}\right)}\left(\gamma^2 D_1 + \gamma^3 D_2\right) \\
&\leqslant \exp\left(-\min\left(\gamma\mu, \frac{1-\rho_{r,W}}{8}\right)K\right)\mathbb{E}\left[T_0\right] + \frac{1}{\min\left(\gamma\mu, \frac{1-\rho_{r,W}}{8}\right)}\left(\gamma^2 D_1 + \gamma^3 D_2\right) \\
&\leqslant \exp\left(-\min\left(\gamma\mu, \frac{1-\rho_{r,W}}{8}\right)K\right) + \frac{D_1}{\mu^2 K} + \frac{D_2}{\mu^3 K^2} + \frac{D_1}{\left(1-\rho_{r,W}\right)\mu^2 K^2} + \frac{D_2}{\left(1-\rho_{r,W}\right)\mu^3 K^3}.
\end{aligned} \tag{85}$$

To get $\mathbb{E}\left[T_K\right] \leqslant \varepsilon$, we have

$$K \geqslant \tilde{\mathcal{O}}\left(\left(\frac{1}{\gamma\mu} + \frac{1}{1-\rho_{r,W}}\right)\log\frac{\mathbb{E}\left[V_0\right]}{\varepsilon} + \frac{\sigma^*}{nb\mu^2\varepsilon} + \sqrt{\frac{\rho_{r,W}L}{\mu^3\left(1-\rho_{r,W}\right)\varepsilon}\left(\frac{4\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right)}\right) \geqslant \frac{1}{1-\rho_{r,W}}. \tag{86}$$

Second, if $\left\{\gamma, \frac{1-\rho_{r,W}}{8\mu}\right\} \geqslant \frac{\ln\left(\max\left\{2, \min\left\{\frac{\mu^2K^2}{D_1}, \frac{\mu^3K^3}{D_2}\right\}\right\}\right)}{\mu K}$ holds, we set

$$\alpha = \frac{\ln\left(\max\left\{2, \min\left\{\frac{\mu^2K^2}{D_1}, \frac{\mu^3K^3}{D_2}\right\}\right\}\right)}{\mu K},$$

and then obtain

$$\begin{aligned}
\mathbb{E}\left[T_K\right] &\leqslant \left(1-\alpha\mu\right)^K \mathbb{E}\left[T_0\right] + \frac{1}{\mu}\left(\gamma D_1 + \gamma^2 D_2\right) \\
&= \exp\left(-\frac{\ln\left(\max\left\{2, \min\left\{\frac{\mu^2K^2}{D_1}, \frac{\mu^3K^3}{D_2}\right\}\right\}\right)}{\mu K}\mu K\right) + \frac{1}{\mu}\left(\gamma D_1 + \gamma^2 D_2\right) \\
&= \tilde{\mathcal{O}}\left(\frac{D_1}{\mu^2 K} + \frac{D_2}{\mu^3 K^2}\right).
\end{aligned}$$

In all, substituting the value of γ and hiding the logarithmic factors and constants, we have $\mathbb{E}\left[T_K\right] \leqslant \varepsilon$ after at most the following number of iterations:

$$K \geqslant \tilde{\mathcal{O}}\left(\left(\frac{L}{\mu\left(1-\rho_{r,W}\right)}\right)\log\frac{\mathbb{E}\left[T_0\right]}{\varepsilon} + \frac{\sigma^*}{nb\mu^2\varepsilon} + \sqrt{\frac{\rho_{r,W}L}{\mu^3\left(1-\rho_{r,W}\right)\varepsilon}\left(\frac{4\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right)}\right).$$

which completes the proof. □

C.3. Proof of Theorem 2

Proof. In order to obtain the convergence rate, we choose the following parameters for the Lyapunov function:

$$c_0 = 1, c_1 = \frac{20L\alpha}{n\left(1-\rho_{r,W}\right)}, c_2 = \frac{5\alpha^2}{Mp}, c_3 = \frac{16\alpha^2}{Mq}, c_4 = 0. \tag{87}$$

Then, using Lemma 8, 9, 10, 12 and 13, if

$$\frac{1+\rho_{r,W}}{2} + \frac{4\alpha^2L^2\rho_{r,W}\left(1+\rho_{r,W}\right)}{1-\rho_{r,W}} \leqslant \frac{3+\rho_{r,W}}{4},$$

which implies

$$\alpha \leqslant \frac{1-\rho_{r,W}}{4L\sqrt{\rho_{r,W}\left(1+\rho_{r,W}\right)}},$$

we can derive the recursion of Lyapunov function as follows:

$$
\begin{aligned}
\mathbb{E}\left[T_{k+1}\right] &= \mathbb{E}\left[\left\|\bar{x}_{k+1}-x^*\right\|_2^2\right] + c_1\mathbb{E}\left[\left\|\hat{X}_{k+1}-\mathbf{1}_n\bar{x}_{k+1}\right\|^2\right] \\
&\quad + c_2\mathbb{E}\left[\left\|\nabla F\left(X_{t_k}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|_2^2\right] + c_3\mathbb{E}\left[\nabla F\left(X_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right] \\
&\leqslant \underbrace{\left(\frac{160\alpha^3L^2\rho_{r,W}\left(1+\rho_{r,W}\right)}{\left(1-\rho_{r,W}\right)^2}-\left(2\alpha-72\alpha^2L\right)\right)}_{e_1}\mathbb{E}\left[f\left(\bar{x}_k\right)-f\left(x^*\right)\right] \\
&\quad + \underbrace{\left((1-r)\frac{\left(4\alpha^2L^2+\alpha L\right)}{n}+(1-r)\frac{32\alpha^2L^2}{n}-\frac{5L\alpha}{2n}\right)}_{e_2}\mathbb{E}\left[\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2\right] \\
&\quad + \underbrace{\left(\frac{2\alpha^2}{n}\frac{1-p}{M}+\frac{20\alpha^3L\rho_{r,W}}{\left(1-\rho_{r,W}\right)}\frac{1-p}{M}-\frac{5\alpha^2}{2M}\right)}_{e_3}\mathbb{E}\left[\left\|\nabla F\left(X_{t_{k-1}}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|_2^2\right] \\
&\quad + \underbrace{\left(\frac{2\alpha^2}{n}\frac{p}{M}+\frac{20\alpha^3L\rho_{r,W}}{\left(1-\rho_{r,W}\right)}\frac{p}{M}+\frac{5\alpha^2}{M}-\frac{8\alpha^2}{M}\right)}_{e_4}\mathbb{E}\left[\left\|\left[\nabla F\left(X_{k-1}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|_2^2\right] \\
&\quad + (1-\alpha\mu)\,\mathbb{E}\left[\left\|\bar{x}_k-x^*\right\|_2^2\right] + \left(1-\frac{1-\rho_{r,W}}{8}\right)c_1\mathbb{E}\left[\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2\right] \\
&\quad + \left(1-\frac{p}{2}\right)c_2\mathbb{E}\left[\left\|\nabla F\left(X_{t_{k-1}}\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right\|_2^2\right] \\
&\quad + \left(1-\frac{q}{2}\right)c_3\mathbb{E}\left[\left\|\left[\nabla F\left(X_k\right)-\nabla F\left(\mathbf{1}_M x^*\right)\right]\right\|_2^2\right] + \frac{80\alpha^3L\rho_{r,W}}{\left(1-\rho_{r,W}\right)^2}\zeta^*.
\end{aligned}
$$

It can be verified that if the step-size satisfies:

$$
\alpha \leqslant \min\left\{\frac{1}{64L}, \frac{1-\rho_{r,W}}{40L}, \frac{1-\rho_{r,W}}{16L\sqrt{\rho_{r,W}\left(1+\rho_{r,W}\right)}}\right\} = \mathcal{O}\left(\frac{1-\rho_{r,W}}{L}\right), \tag{88}
$$

then we have $e_1, e_2, e_3, e_4 \leqslant 0$, and

$$
\mathbb{E}\left[T_{k+1}\right] \leqslant \left(1-\min\left\{\alpha\mu, \frac{pq}{2}, \frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}\left[T_k\right] + \frac{80\alpha^3L\rho_{r,W}}{\left(1-\rho_{r,W}\right)^2}\zeta^*,
$$

which completes the proof. □

C.4. Proof of Corollary 2

Proof. Similar to the proof of Corollary 1, let γ denote the resulting upper bound on step size α in (88), and give another extra upper bound as follow:

$$
\alpha \leqslant \min\left\{\gamma, \frac{\ln\left(\max\left\{2, \frac{\mu^3K^2}{D_1}\right\}\right)}{\mu K}\right\}, \tag{89}
$$

where $D_1 = \frac{80L\rho_{r,W}}{(1-\rho_{r,W})^2}\zeta^*$. Then, for a constant iteration K such that either $\gamma \leqslant \frac{\ln\left(\max\left\{2, \frac{\mu^3K^2}{D_1}\right\}\right)}{\mu K}$ or

$$
\frac{\ln\left(\max\left\{2, \frac{\mu^3K^2}{D_1}\right\}\right)}{\mu K} \leqslant \min\left\{\frac{1-\rho_{r,W}}{8\mu}, \frac{pq}{2\mu}\right\},
$$

Then, substituting the value of γ and hiding the logarithmic factors and constants, we have $\mathbb{E}\left[T_K\right] \leqslant \varepsilon$ after at most the following number of iterations

$$K \geqslant \mathcal{O}\left(\frac{1}{pq}+\frac{L}{\mu\left(1-\rho_{r,W}\right)}\right)\log\left(\frac{\mathbb{E}\left[T_0\right]}{\varepsilon}\right)+\mathcal{O}\left(\sqrt{\frac{\rho_{r,W}L\zeta^*}{\left(1-\rho_{r,W}\right)^2\mu^3\varepsilon}}\right),$$

which completes the proof. □

C.5. Proof of Theorem 3

Proof. For algorithms $\mathcal{A}\left(\cdot,\cdot,C_k=W_k\otimes\mathbf{J}_m\right)$, recalling the following parameter settings of Lyapunov function (23)

$$c_0=1,\ c_1=\frac{1-\rho_{r,W}}{n\rho_{r,W}\left(1+\rho_{r,W}\right)},\ c_2=0,\ c_3=\frac{20\alpha^2}{Mq\left(1-\rho_{r,W}\right)^2},\ c_4=\frac{8\alpha^2}{n\left(1-\rho_{r,W}\right)}$$

and using Lemma 8, 9, 11, 12, 14, we have

$$\begin{aligned}\mathbb{E}\left[T_{k+1}\right] &\leqslant \left(1-\min\left\{\alpha\mu,\frac{q}{2},\frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}\left[T_k\right]\\ &+e_1\mathbb{E}\left[f\left(\bar{x}_k\right)-f\left(x^*\right)\right]+e_2\mathbb{E}\left[\left\|\hat{X}_k-\mathbf{1}_n\bar{x}_k\right\|^2\right]\\ &+e_3\mathbb{E}\left[\left\|\nabla F\left(X_{k-1}\right)-\nabla F\left(\mathbf{1}_Mx^*\right)\right\|^2\right]+e_4\mathbb{E}\left[\left\|\hat{Y}_k-\mathbf{1}_n\bar{y}_k\right\|^2\right],\end{aligned} \tag{90}$$

where

$$\begin{aligned}e_1&=\frac{32\alpha^2L\left(1+\rho_{r,W}\right)}{\left(1-\rho_{r,W}\right)^2}\left(8\alpha^2L^2+4+2q\right)+\frac{80\alpha^2L}{\left(1-\rho_{r,W}\right)^2}-\left(2\alpha-8\alpha^2L\right),\\ e_2&=\frac{32\alpha^2L^2\left(1+\rho_{r,W}\right)}{n\left(1-\rho_{r,W}\right)^2}\left(4+\left(1-r\right)q+4\left(1-r\right)\alpha^2L^2\right)+\left(1-r\right)\frac{80\alpha^2L^2}{n\left(1-\rho_{r,W}\right)^2}\\ &\quad+\left(1-r\right)\frac{\left(4\alpha^2L^2+\alpha L\right)}{n}-\frac{\left(1-\rho_{r,W}\right)^2}{4n\rho_{r,W}\left(1+\rho_{r,W}\right)},\\ e_3&=\frac{16\alpha^2\left(1+\rho_{r,W}\right)}{M\left(1-\rho_{r,W}\right)^2}\left(1-q+\frac{4\alpha^2L^2}{n}\right)+\frac{2\alpha^2}{M}-\frac{20\alpha^2}{2M\left(1-\rho_{r,W}\right)^2},\\ e_4&=\frac{\alpha^2}{n}-\frac{8\alpha^2}{8n}=0.\end{aligned}$$

It can be verified that if the step-size satisfies:

$$\alpha\leqslant\min\left\{\frac{1}{8L},\frac{1-\rho_{r,W}}{4L\sqrt{2\rho_{r,W}\left(1+\rho_{r,W}\right)}},\frac{\left(1-\rho_{r,W}\right)^2}{528L}\right\}=\mathcal{O}\left(\frac{\left(1-\rho_{r,W}\right)^2}{L}\right), \tag{91}$$

then all the coefficients satisfy $e_1,e_2,e_3,e_4\leqslant 0$, which further leads to

$$\mathbb{E}\left[T_{k+1}\right]\leqslant\left(1-\min\left\{\alpha\mu,\ \frac{q}{2},\ \frac{1-\rho_{r,W}}{8}\right\}\right)\mathbb{E}\left[T_k\right]. \tag{92}$$

We thus complete the proof. □

C.6. Proof of Corollary 3

Proof. By Theorem 3, we have

$$\mathbb{E}\left[T_k\right]\leqslant\max\left\{1-\alpha\mu,\frac{q}{2},\frac{1-\rho_{r,W}}{8}\right\}^k\mathbb{E}\left[T_0\right],$$

where

$$\alpha = \mathcal{O}\left(\frac{(1-\rho_{r,W})^2}{L}\right),$$

then similar to the proof of Corollary 1, we have $\mathbb{E}[T_K] \leqslant \varepsilon$ after at most the following number of iterations:

$$K \geqslant \tilde{\mathcal{O}}\left(\left(\frac{L}{\mu(1-\rho_{r,W})^2} + \frac{1}{q}\right)\log\frac{\mathbb{E}[T_0]}{\varepsilon}\right).$$

□

D. Sub-linear Convergence Analysis for Convex Problems

In this section, we give the corresponding sub-linear convergence rate in section 5 for smooth and convex ($\mu = 0$) objectives, the proofs are also based on the Lyapunov function (23).

Theorem 4. *Consider algorithms belonging to $\mathcal{A}(\cdot,\cdot,C_k \equiv \mathbf{I}_M)$. Suppose Assumption 1-5 hold and $\mu = 0$. Let*

$$c_0 = 1,\ c_1 = (1-r)\frac{8L(4\alpha L+1)}{n(1-\rho_{r,W})},\ c_2 = c_3 = c_4 = 0$$

and the step-size satisfy

$$\alpha \leqslant \min\left\{\frac{1}{5L}, \frac{(1-\rho_{r,W})}{24\mathrm{L}\sqrt{\rho_{r,W}(1+\rho_{r,W})}}\right\} = \min\left\{\mathcal{O}\left(\frac{1}{L}\right), \mathcal{O}\left(\frac{1-\rho_{r,W}}{\mathrm{L}\sqrt{\rho_{r,W}}}\right)\right\}, \tag{93}$$

then we have for all $k > 0$

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(\bar{x}_k) - f(x^*)] \leqslant \frac{5\|\bar{x}_0 - x^*\|^2}{\alpha K} + \frac{10\alpha\sigma^*}{nb} + (1-r)\frac{72\alpha^2\mathrm{L}\rho_{r,W}}{n(1-\rho_{r,W})}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + \frac{n\sigma^*}{b}\right). \tag{94}$$

Proof. According to the proof of Theorem 1 and noticing $\mu = 0$, we further let the step size satisfies

$$\alpha \leqslant \frac{(1-\rho_{r,W})}{24\mathrm{L}\sqrt{\rho_{r,W}(1+\rho_{r,W})}},$$

which implies that

$$\left(2\alpha - 8\alpha^2 L\right) - (1-r)\frac{64\alpha^3\mathrm{L}^2(4\alpha L+1)\rho_{r,W}(1+\rho_{r,W})}{(1-\rho_{r,W})^2} \geqslant \left(\alpha - 4\alpha^2 L\right).$$

Combining the upper bound of step-size in (82), then we have

$$\begin{aligned}&\left(\alpha - 4\alpha^2 L\right)\left(f(\hat{x}_k) - f(x^*)\right)\\ &\leqslant \mathbb{E}[T_k] - \mathbb{E}[T_{k+1}] + \frac{2\alpha^2\sigma^*}{nb} + (1-r)\frac{8\alpha\mathrm{L}(4\alpha L+1)}{n(1-\rho_{r,W})}\alpha^2\rho_{r,W}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + \frac{n\sigma^*}{b}\right).\end{aligned} \tag{95}$$

Then, summing over k from 0 to $K-1$, we further obtain

$$\begin{aligned}&\frac{1}{K}\sum_{k=1}^{K-1}\left(f(\bar{x}_k) - f(x^*)\right)\\ &\leqslant \frac{\mathbb{E}[T_0]}{\alpha(1-4\alpha L)K} + \frac{2\alpha\sigma^*}{nb(1-4\alpha L)} + (1-r)\frac{8\alpha^2\mathrm{L}(4\alpha L+1)\rho_{r,W}}{n(1-\rho_{r,W})(1-4\alpha L)}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + \frac{n\sigma^*}{b}\right)\\ &\leqslant \frac{5\|\bar{x}_0 - x^*\|^2}{\alpha K} + \frac{10\alpha\sigma^*}{nb} + (1-r)\frac{72\alpha^2\mathrm{L}\rho_{r,W}}{n(1-\rho_{r,W})}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + \frac{n\sigma^*}{b}\right),\end{aligned}$$

which completes the proof. □

Corollary 4. *Under the same conditions in Theorem 4, suppose the step-size satisfy*

$$\alpha \leqslant \left\{\gamma, \frac{1}{\sqrt{H_1 K}}, \frac{1}{\sqrt[3]{H_2 K}}\right\},$$

where γ denotes the resulting upper bound on α in (93) and

$$H_1 = \frac{\sigma^*}{nb\left\|x_0 - x^*\right\|^2},\ H_2 = \frac{72\mathrm{L}\rho_{r,W}\left(1-r\right)}{\left(1-\rho_{r,W}\right)\left\|x_0 - x^*\right\|^2}\left(\frac{4\zeta^*}{1-\rho_{r,W}} + \frac{\sigma^*}{b}\right).$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f\left(\bar{x}_k\right) - f\left(x^*\right)\right] \leqslant \varepsilon$$

after at most the following iterations:

$$K \geqslant \left(\frac{1}{\gamma\varepsilon} + \frac{\sigma^*}{nb\varepsilon^2} + \frac{1}{\varepsilon^{3/2}}\sqrt{\frac{\rho_{r,W}L\left(b\zeta^* + \left(1-\rho_{r,W}\right)\sigma^*\right)}{b\left(1-\rho_{r,W}\right)^2}}\right)\left\|\bar{x}_0 - x^*\right\|^2. \tag{96}$$

Proof. According to the obtained sub-linear rate in Theorem 4 and the upper bound on α, we have three cases:

$$\gamma \leqslant \left\{\frac{1}{\sqrt{H_1 K}}, \frac{1}{\sqrt[3]{H_2 K}}\right\}, \quad \frac{1}{\sqrt{H_1 K}} \leqslant \left\{\gamma, \frac{1}{\sqrt[3]{H_2 K}}\right\}, \quad \frac{1}{\sqrt[3]{H_2 K}} \leqslant \left\{\gamma, \frac{1}{\sqrt{H_1 K}}\right\}.$$

Then, we can obtain

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f\left(\bar{x}_k\right) - f\left(x^*\right)\right] \\
&\leqslant \frac{5\left\|\bar{x}_0 - x^*\right\|^2}{\min\left\{\gamma, \frac{1}{\sqrt{H_1 K}}, \frac{1}{\sqrt[3]{H_2 K}}\right\}K} + \frac{10\sigma^*}{nb}\frac{1}{\sqrt{H_1 K}} \\
&+ \left(1-r\right)\frac{72L\rho_{r,W}}{n\left(1-\rho_{r,W}\right)}\left(\frac{4n\zeta^*}{1-\rho_{r,W}} + \frac{n\sigma^*}{b}\right)\frac{1}{\sqrt[3]{H_2 K}} \leqslant \varepsilon.
\end{aligned}$$

Plugging H_1 and H_2 into the above inequality and merging the similar terms, we obtain the complexity in (96). □

Theorem 5. *Consider algorithms $\mathcal{A}\left(\cdot,\cdot,C_k = \mathbf{I}_n \otimes V_k\right)$ with $p > 0$. Suppose Assumption 1-5 hold and $\mu = 0$. Let*

$$c_0 = 1,\ c_1 = \frac{20L\alpha}{n\left(1-\rho_{r,W}\right)},\ c_2 = \frac{5\alpha^2}{Mp},\ c_3 = \frac{16\alpha^2}{Mq},\ c_4 = 0$$

and the step-size satisfy

$$\alpha \leqslant \min\left\{\frac{1}{64L}, \frac{1-\rho_{r,W}}{40L}, \frac{1-\rho_{r,W}}{32L\sqrt{\rho_{r,W}\left(1+\rho_{r,W}\right)}}\right\} = \mathcal{O}\left(\frac{1-\rho_{r,W}}{L}\right). \tag{97}$$

Then, we have

$$\frac{1}{K}\sum_{k=1}^{K-1}\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right) \leqslant \frac{\mathbb{E}\left[T_0\right]}{\alpha K} + \frac{80\alpha^2 L\rho_{r,W}}{\left(1-\rho_{r,W}\right)^2}\zeta^*. \tag{98}$$

Proof. According to the proof of Theorem 2 and noticing $\mu = 0$, let the step size satisfy

$$\alpha \leqslant \frac{1-\rho_{r,W}}{32L\sqrt{\rho_{r,W}\left(1+\rho_{r,W}\right)}},$$

which implies that

$$\frac{160\alpha^3 L^2\rho_{r,W}\left(1+\rho_{r,W}\right)}{n\left(1-\rho_{r,W}\right)^2} \geqslant \left(\alpha - 36\alpha^2 L\right).$$

Combining the upper bound of step-size in (88), then we get

$$\left(\alpha - 36\alpha^2 L\right)\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right) \leqslant \mathbb{E}\left[T_k\right] - \mathbb{E}\left[T_{k+1}\right] + \frac{80\alpha^3 L\rho_{r,W}}{\left(1-\rho_{r,W}\right)^2}\zeta^*. \tag{99}$$

Summing over k from 0 to $K-1$, we have

$$\frac{1}{K}\sum_{k=1}^{K-1}\left(f\left(\bar{x}_k\right) - f\left(x^*\right)\right) \leqslant \frac{\mathbb{E}\left[T_0\right]}{\alpha\left(1-36\alpha L\right)K} + \frac{80\alpha^2 L\rho_{r,W}}{\left(1-36\alpha L\right)\left(1-\rho_{r,W}\right)^2}\zeta^* \leqslant \frac{\mathbb{E}\left[T_0\right]}{\alpha K} + \frac{80\alpha^2 L\rho_{r,W}}{\left(1-\rho_{r,W}\right)^2}\zeta^* \tag{100}$$

where

$$\begin{aligned} T_0 &= \left\|x_0 - x^*\right\|^2 + \mathcal{O}\left(\frac{\alpha^2}{pq}\right)\left\|\nabla f\left(x_0\right) - \nabla f\left(x^*\right)\right\|^2 \\ &\leqslant \mathbb{E}\left[\left\|x_0 - x^*\right\|^2\right] + \mathcal{O}\left(\frac{\alpha^2}{pq}\right)L^2\left\|x_0 - x^*\right\|^2 = \mathcal{O}\left(\frac{1}{pq}\right)\left\|x_0 - x^*\right\|^2. \end{aligned}$$

□

Corollary 5. *Under the same conditions in Theorem 5, suppose that the step-size*

$$\alpha \leqslant \min\left\{\gamma, \frac{1}{\sqrt{H_1 K}}\right\},$$

where γ is the resulting upper bound of α in (97) and

$$H_1 = \frac{80L\rho_{r,W}pq}{\left(1-\rho_{r,W}\right)^2\left\|x_0 - x^*\right\|^2}\zeta^*.$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f\left(\bar{x}_k\right) - f\left(x^*\right)\right] \leqslant \varepsilon$$

after at most the following iterations:

$$K \geqslant \left(\frac{1}{\gamma pq\varepsilon} + \frac{1}{pq\varepsilon^{3/2}}\sqrt{\frac{\rho_{r,W}L\zeta^*}{\left(1-\rho_{r,W}\right)^2}}\right)\left\|x_0 - x^*\right\|^2. \tag{101}$$

Proof of Corollary 5 is similar to the proof of Corollary 4 and thus omitted.

Theorem 6. *Consider the algorithms $\mathcal{A}\left(\cdot,\cdot,C_k = W_k \otimes \mathbf{J}_m\right)$. Suppose Assumption 1-5 hold and $\mu = 0$. Let*

$$c_0 = 1,\ c_1 = \frac{1-\rho_{r,W}}{n\rho_{r,W}\left(1+\rho_{r,W}\right)},\ c_2 = 0,\ c_3 = \frac{20\alpha^2}{Mq\left(1-\rho_{r,W}\right)^2},\ c_4 = \frac{8\alpha^2}{n\left(1-\rho_{r,W}\right)}$$

and the step-size satisfy

$$\alpha \leqslant \min\left\{\frac{1}{8L}, \frac{1-\rho_{r,W}}{4L\sqrt{2\rho_{r,W}\left(1+\rho_{r,W}\right)}}, \frac{\left(1-\rho_{r,W}\right)^2}{1056L}\right\} = \mathcal{O}\left(\frac{\left(1-\rho_{r,W}\right)^2}{L}\right). \tag{102}$$

Then, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f\left(\bar{x}_k\right) - f\left(x^*\right)\right] \leqslant \frac{\left(1-\rho_W\right)^2}{2\alpha\left(\left(1-\rho_W\right)^2 - \left(640 + 10\left(1-\rho_W\right)^2\right)\alpha L\right)K}\mathbb{E}\left[T_0\right]. \tag{103}$$

Proof. According to the proof of Theorem 2 and noticing $\mu = 0$, let the step size satisfy

$$\alpha \leqslant \frac{(1-4\alpha L)\left(1-\rho_{r,W}\right)^2}{1056L},$$

which implies that

$$\frac{32\alpha^2 L\left(1+\rho_{r,W}\right)}{\left(1-\rho_{r,W}\right)^2}\left(8\alpha^2L^2+4+2q\right)+\frac{80\alpha^2 L}{\left(1-\rho_{r,W}\right)^2} \leqslant \left(\alpha-4\alpha^2L\right).$$

Combining the upper bound of step-size in (91), then we get

$$\left(\alpha-4\alpha^2L\right)\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) \leqslant \mathbb{E}\left[T_k\right]-\mathbb{E}\left[T_{k+1}\right].$$

Summing over k from 0 to $K-1$, we have

$$\frac{1}{K}\sum_{k=0}^{K-1}\left(f\left(\bar{x}_k\right)-f\left(x^*\right)\right) \leqslant \frac{\mathbb{E}\left[T_0\right]}{\left(1-\alpha L\right)\alpha K},$$

where

$$\begin{aligned}\mathbb{E}\left[T_0\right] &= \left\|x_0-x^*\right\|^2+\mathcal{O}\left(\frac{\alpha^2}{Mq\left(1-\rho_{r,W}\right)^2}\right)\left\|\nabla F\left(\mathbf{1}_M x_0\right)\right\|^2 \\ &= \left\|x_0-x^*\right\|^2+\mathcal{O}\left(\frac{\alpha^2L^2}{q\left(1-\rho_{r,W}\right)^2}\right)\left\|x_0-x^*\right\|^2=\mathcal{O}\left(\frac{1}{q}\left\|x_0-x^*\right\|^2\right).\end{aligned} \tag{104}$$

□

Corollary 6. *Under the conditions same in Theorem 6, we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[f\left(\bar{x}_k\right)-f\left(x^*\right)\right] \leqslant \varepsilon$$

after at most the following iterations:

$$K \geqslant \mathcal{O}\left(\frac{1}{\gamma q\varepsilon}\right)\left\|x_0-x^*\right\|^2. \tag{105}$$

Proof of Corollary 6 is similar to the proof of Corollary 4 and thus omitted.

E. Additional Experiments

Table 5. Summary of the experimental setup.

Dataset	**Node** (n)	**# Train**	**# Test**	**Dimension**	**BS** $(n \times b)$	**SS** (α)	λ
F-MNIST	{8, 20, 50}	60000	10000	784	200	0.05	0.001
CIFAR-10	{8, 20, 50}	50000	10000	3072	400	0.008	0.001

In this section, we further verify our theoretical findings by several extra experiments. The experiment setting is recalled here. We train a regularized logistic regression classifier on both CIFAR-10 and Fashion-MNIST (F-MNIST) datasets over a network of n nodes each of which locally stores m data samples, as defined in (33) and (34):

$$\min_{x\in\mathbb{R}^d} f\left(x\right) := \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\Bigg(\underbrace{\ell\left(x,\xi_{ij}\right)+\frac{\lambda}{2}\left\|x\right\|^2}_{f_{ij}}\Bigg),$$

with the cross-entropy loss ℓ:

$$\ell\left(x,\xi_{ij}\right) := -\sum_{c=1}^{10}\phi_{ij}^c\log\left(1+\exp\left(-x^T\theta_{i,j}\right)\right)^{-1},$$

where λ is a regularization parameter, ξ_{ij} represents the j-th sample of node i with feature vector $\theta_{i,j}\in\mathbb{R}^d$ and label $\phi_{ij}^c\in\{-1,1\}$ of class c. The datasets and parameters we used are summarized in Table 5 with $r=0.05$ for corresponding algorithms. All the algorithms are executed on a server with 8 GPUs (NVIDIA RTX 2080Ti).

Table 6. Unbalanced label distribution of training samples with $h = 20$ for CIFAR-10.

Node \ Label	1	2	3	4	5	6	7	8	9	10
1	555	575	595	615	635	655	675	695	625	625
2	575	595	615	635	655	675	695	555	625	625
3	595	615	635	655	675	695	555	575	625	625
4	615	635	655	675	695	555	575	595	625	625
5	635	655	675	695	555	575	595	615	625	625
6	655	675	695	555	575	595	615	635	625	625
7	675	695	555	575	595	615	635	655	625	625
8	695	555	575	595	615	635	655	675	625	625

Table 7. A highly unbalanced label distribution of training samples denoted by $h = h_{\max}$ for CIFAR-10.

Node \ Label	1	2	3	4	5	6	7	8	9	10
1	1000	0	0	0	1000	1000	1000	1000	625	625
2	1000	1000	0	0	0	1000	1000	1000	625	625
3	1000	1000	1000	0	0	0	1000	1000	625	625
4	1000	1000	1000	1000	0	0	0	1000	625	625
5	1000	1000	1000	1000	1000	0	0	0	625	625
6	0	1000	1000	1000	1000	1000	0	0	625	625
7	0	0	1000	1000	1000	1000	1000	0	625	625
8	0	0	0	1000	1000	1000	1000	1000	625	625

Heterogeneously Split Dataset. In order to generate different levels of data heterogeneity among devices, we split the samples of 10 classes into $n = \{8, 20, 50\}$ nodes with different *label distributions* for both Fashion-MNIST and CIFAR-10 datasets. In particular, we denote by $m_{i,c}$ the number of samples of class $c \in [10]$ allocated to node $i \in [n]$, then we generate the unbalanced label distribution through cyclic arithmetic sequences with a difference of h, i.e, for the case $n = 8$, we have

$$m_{i,c} = \begin{cases} m_0 + h(i + c - 2), \ \forall i, c \in [n] \\ \frac{M}{10n}, \ \forall i \in [n], c \notin [n] \end{cases}, \quad \text{s.t.,} \ \sum_{i=1}^{n} m_{i,c} = \frac{M}{10}, \ \sum_{c=1}^{10} m_{i,c} = \frac{M}{n}, \tag{106}$$

where M denotes the total number of samples to be allocated, m_0 denotes the initial term of the arithmetic sequence, and h is the difference between the consecutive terms which we used to represent the level of data heterogeneity. We also consider the cases $n = 20, 50$ which are larger than the number of classes, then the label distribution $\{m_{i,c}\}$ is generated as follow:

$$m_{i,c} = m_0 + h((i + c - 2)\,\%10), \forall i \in [n], c \in [10], \quad \text{s.t.,} \ \sum_{i=1}^{n} m_{i,c} = \frac{M}{10}, \ \sum_{c=1}^{10} m_{i,c} = \frac{M}{n}, \tag{107}$$

where $\%$ denotes the remainder operator. Intuitively, the larger the value of h, the more heterogeneous the local datasets will be. For example, let $n = 8$, $h = 20$ and $m_0 = 555$, we get the allocation strategy of training samples on CIFAR-10 dataset by (106) in Table 6. Furthermore, we also design a highly unbalanced label distribution of CIFAR-10 denoted by $h_{\max}$ in Table 7 (can be adapted to F-MNIST), which indicates that each node only has samples of 7 classes while other classes of samples are inaccessible. As a result, it is very difficult to learn a global model for 10-class image classification task in a distributed manner (c.f. Fig. 2).

Topology Dependence. We provide more experiments to verify the topology dependence of different algorithms, as we reported in the theoretical results in the main text. We compare various algorithms that can be recovered in the proposed SPP framework on heterogeneously split datasets ($h = 20$) over graphs: i) directed ring with $n = 8$; ii) directed ring with $n = 20$; iii) geometric graph with $n = 50$. The results are summarized in Fig. 4. It follows from the figure that the algorithms adopting VR and/or GT schemes (solid lines) outperform the others both in terms of testing accuracy and training loss, especially on the graph with worse connectivity (i.e., $n = 50, \rho_W \approx 0.99$), which verifies the dependency of the performance of the algorithms on the sampling variance, data heterogeneity, and the connectivity of the graph.

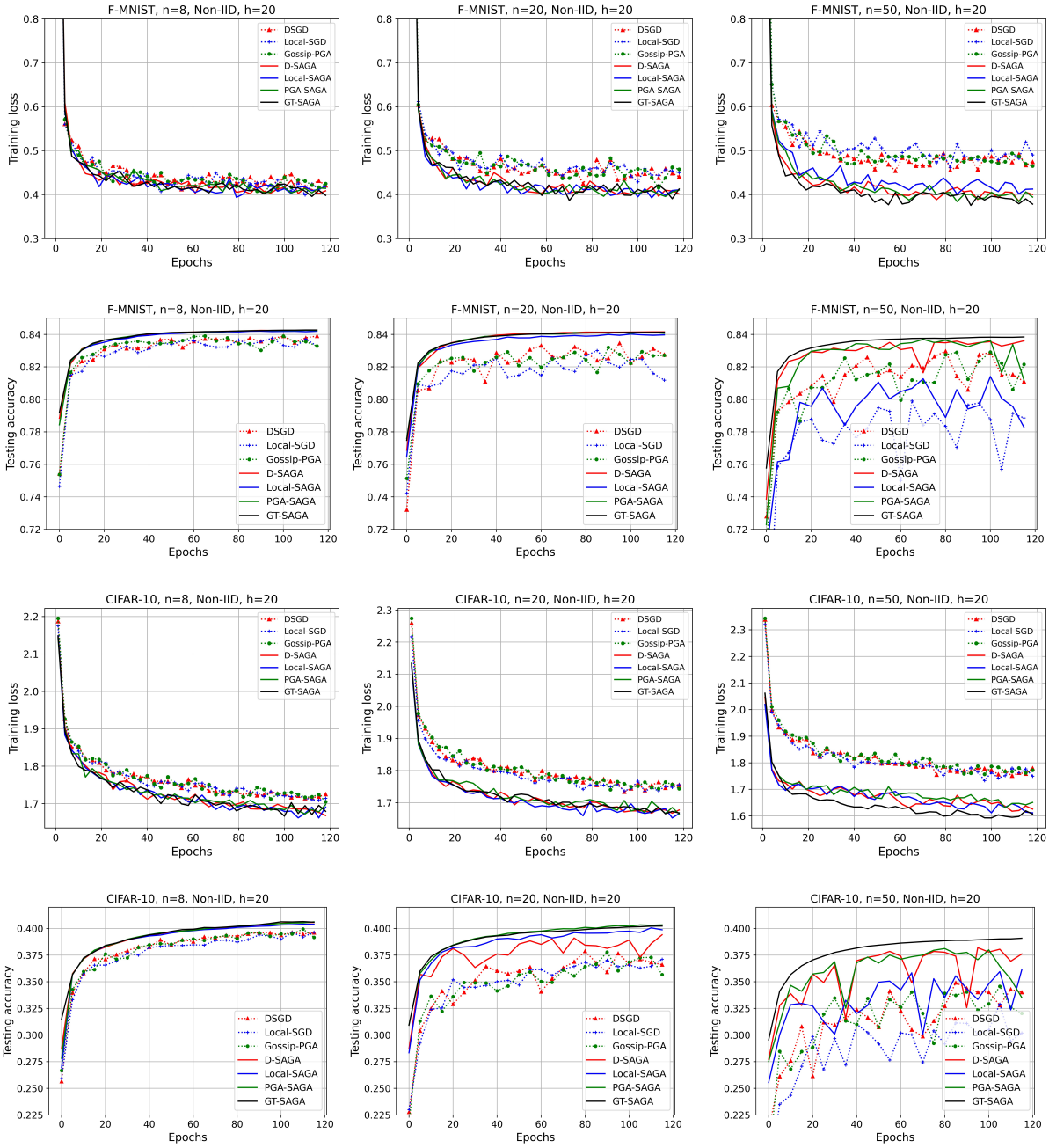

Figure 4. Performance comparison of DSGD, Local-SGD, Gossip-PGA, D-SAGA, Local-SAGA, PGA-SAGA and GT-SAGA over three graphs: i) directed ring with $n = 8$ (first column); ii) directed ring with $n = 20$ (second column); iii) geometric graph with $n = 50$ (third column). The sub-figures on the first two rows plot the training loss and testing accuracy of the algorithms on Fashion-MNIST dataset, respectively, and the sub-figures on the last two rows plot the training loss and testing accuracy on CIFAR-10 dataset.