AIMS Mathematics, 11(2): 4123–4146.
DOI: 10.3934/math.2026166
Received: 11 October 2025
Revised: 19 January 2026
Accepted: 28 January 2026
Published: 10 February 2026

https://www.aimspress.com/journal/Math

Research article

Distributed stochastic optimization algorithm based on Markov sampling

Hao Zan and Dan Chen*

Yunnan Key Laboratory of Statistical Modeling and Data Analysis, Yunnan University, Kunming 650500, China

* **Correspondence:** Email: danchen@ynu.edu.cn.

Abstract: With the rapid proliferation of big data and high-dimensional datasets, distributed estimation methods have become indispensable for large-scale statistical inference. However, traditional centralized distributed computing frameworks often suffer from communication bottlenecks and privacy risks. To address these challenges, we propose a novel distributed stochastic optimization algorithm, termed gradient-based Markov subsampling Gradient Tracking with Variance Reduction (GMS-GT-VR). This algorithm seamlessly integrates gradient tracking and variance reduction techniques with a data-driven gradient-based Markov subsampling (GMS) strategy to solve large-scale distributed optimization problems over multi-agent networks efficiently. By leveraging a lightweight coordination mechanism for adaptive sampling, GMS-GT-VR effectively reduces both communication overhead and computational complexity, while achieving accelerated convergence. We rigorously establish its theoretical convergence rate of $O(1/k)$ and conduct a detailed complexity analysis. To validate the theoretical findings, we perform extensive experiments on large-scale datasets. The results consistently demonstrate that GMS-GT-VR outperforms existing variance-reduction methods in terms of communication efficiency and convergence speed, achieving a significant reduction in the number of iterations required.

Keywords: distributed stochastic optimization algorithms; variance reduction techniques; Markov chain; resampling methods
Mathematics Subject Classification: 90C15, 62L20, 68W15

1. Introduction

In recent years, with the widespread application of big data and high-dimensional datasets, distributed estimation methods have emerged as essential tools for addressing large-scale statistical inference problems. These methods offer effective solutions for challenges such as high dimensionality, non-smooth loss functions, and data heterogeneity, which are prevalent in modern data-intensive

environments. As distributed computing technologies—originating from computer science—continue to evolve, they have been increasingly adopted in statistics, particularly through the divide-and-conquer (DC) framework. This approach partitions data across multiple machines, conducts local estimations independently, and aggregates the results centrally, thereby enhancing computational efficiency and scalability.Notable contributions in this area include the work of Jordan, Lee, and Yang, who explored communication-efficient distributed statistical inference, focusing on reducing communication overhead while preserving statistical accuracy in large-scale settings [1]. Lee et al. proposed a distributed stochastic variance-reduced gradient method that leverages additional data sampling with replacement to improve computational performance [2]. Lee, Liu, Sun, and Taylor further developed a communication-efficient sparse regression framework tailored for large-scale distributed learning problems [3]. Song and Sun introduced a divide-and-conquer algorithm for network optimization, which achieves exponential convergence and significantly reduces computational costs [4]. Finally, Gao, Liu, and Zhang provided a comprehensive review of recent advances in distributed statistical inference, covering both parametric and non-parametric models, and examining the trade-offs between communication cost and estimation accuracy [5].

However, while the DC strategy mitigates the computational and storage burdens associated with large-scale data processing to some extent, it remains challenged in handling high-dimensional data and non-smooth loss functions. Traditional DC methods impose stringent conditions on convergence rates and machine configurations, which may result in slower convergence and computational bottlenecks when tackling complex problems. For example, in high-dimensional quantile regression (QR), Zhao et al. found that the convergence rate of the estimator is constrained by the number of machines [6]. This highlights the pressing need for improved distributed estimation methods that can enhance algorithmic convergence and computational efficiency, particularly under non-ideal conditions.

To overcome limitations such as slow convergence and high communication cost in distributed optimization, various enhanced strategies have been developed. Among them, approximate Newton-type methods have attracted considerable attention for their use of second-order information. For instance, Yun, Wang, and Zhang approximate Newton steps by computing local Hessian matrices and leveraging second-order information to accelerate convergence [7]. However, such methods require the loss function to be twice differentiable and are therefore not suitable for problems involving non-smooth objectives, such as quantile regression. To overcome this issue, Li, Kyrillidis, Liu, and Caramanis proposed a stochastic gradient-based approximate Newton method that can effectively handle non-smooth loss functions, making it particularly suitable for QR problems [8]. More recently, Chen, Liu, and Zhang introduced a multi-round distributed estimation framework based on first-order stochastic optimization. They proposed the first-order Newton-type estimator (FONE), which efficiently approximates the inverse Hessian and demonstrates significant advantages in high-dimensional settings [9]. These developments underscore the potential of multi-round distributed algorithms in overcoming the limitations of conventional DC approaches.

To accelerate computation in distributed environments, researchers have proposed various stochastic first-order optimization methods. Among these, distributed stochastic gradient descent (DSGD) has been widely adopted for large-scale data processing tasks. DSGD addresses the scalability challenges of massive datasets by combining local stochastic gradient updates with periodic averaging of model parameters across neighboring agents. Nevertheless, in real-world applications, the performance of DSGD can be hindered by local gradient variance and data heterogeneity. Specifically, when data

distributions across agents are imbalanced or significantly different, the convergence rate and stability of the algorithm may deteriorate.To address these limitations, several variance reduction techniques have been proposed to further improve the efficiency and robustness of distributed optimization algorithms. For instance, Defazio, Bach, and Lacoste-Julien proposed the Stochastic Average Gradient Amélioré (SAGA) algorithm, an incremental gradient method that accelerates convergence in large-scale optimization by reducing the variance of gradient estimates [10]. Nguyen et al. introduced the Stochastic Recursive Gradient Algorithm (SARAH) method, which incorporates recursive gradient updates to achieve improved convergence rates while maintaining low computational complexity [11]. Lei and Shanbhag developed the asynchronous variance reduction (Asyn-VR) algorithm, which leverages asynchronous updates and parallelism to enhance the scalability of distributed learning tasks [12]. Johnson and Zhang proposed the stochastic variance reduced gradient (SVRG) method [13], a stochastic first-order optimization algorithm that computes local gradients and intermittently communicates with neighboring agents to mitigate variance. SVRG achieves faster convergence than standard SGD without relying on second-order derivatives, making it computationally more efficient. However, a major drawback of SVRG lies in its communication overhead—each iteration typically requires two rounds of communication, leading to increased communication costs in distributed environments.

Overall, variance reduction techniques offer notable advantages for large-scale optimization. They not only improve convergence rates and alleviate communication overhead but also enhance the stability and robustness of distributed algorithms in the presence of data heterogeneity and imbalance, thereby providing effective solutions for complex optimization challenges.

In distributed environments, researchers have recently developed variance reduction methods tailored for finite-sum optimization problems. For example, Sun, Lu, and Hong proposed the decentralized gradient estimation and tracking (D-GET) algorithm, which integrates a local SARAH-type variance reduction scheme with gradient tracking to improve sample and communication efficiency [14]. However, its gradient computation complexity is influenced by the underlying network topology. To address this, Xin, Khan, and Kar introduced the GT-SARAH algorithm, which achieves near-optimal total gradient computation complexity by increasing the number of communication rounds [15]. They further proposed the GT-SAGA algorithm, combining SAGA with gradient tracking to enhance convergence [16], although this approach demands more storage than SVRG. Additionally, Jiang, Zeng, Sun, and Chen developed the gradient tracking with variance reduction (GT-VR) algorithm based on binomial sampling, further advancing decentralized stochastic optimization [17].

Since then, the field has continued to evolve with solutions for more complex scenarios. For instance, Li and Lin [18] proposed an accelerated gradient tracking method for time-varying graphs, significantly improving convergence rates. Furthermore, Chen et al. [19] applied variance reduction to decentralized policy gradients in multi-agent reinforcement learning. However, despite these advancements, most existing distributed optimization methods still rely on data-agnostic uniform sampling, which may be inefficient in heterogeneous environments.

The stability and robustness of stochastic dynamical systems have been extensively investigated in the control theory community, offering valuable insights for handling system uncertainties. For instance, Liang et al. [22] explored dissipativity-based sampled-data control for fuzzy Markovian jump systems to ensure stochastic stability, while Zhuang et al. [21] investigated non-fragile feedback

control for neutral stochastic Markovian jump systems against time-varying delays. These works highlight the potential of using Markovian structures to enhance system robustness. Building upon these theoretical insights and existing distributed variance reduction methods that combine gradient tracking with stochastic techniques, we further explore how sampling strategies can enhance both communication efficiency and convergence performance. To this end, we propose the gradient-based Markov subsampling gradient tracking variance reduction (GMS-GT-VR) algorithm. Inspired by the GT-VR algorithm, the proposed method adopts a binomial sampling strategy to determine whether agents update the global gradient locally. Specifically, GMS-GT-VR integrates the gradient-based Markov subsampling (GMS) technique [23] with gradient tracking [24], enabling efficient variance reduction under communication constraints and heterogeneous data distributions.

Contributions of this paper are summarized as follows:

1. *Proposal of a distributed stochastic optimization Algorithm:* We propose a novel data-driven distributed stochastic optimization algorithm, namely the GMS-GT-VR algorithm, to address the high communication cost in distributed estimation. Unlike existing data-agnostic methods, the GMS strategy, allows the algorithm to adaptively assign some agents to compute full local gradients based on their gradient norms while others compute single-sample gradients, thereby reducing computational overhead while maintaining favorable convergence properties.
2. *Theoretical and empirical performance analysis:* Theoretical analysis and empirical experiments show that GMS-GT-VR achieves a convergence rate of $O(1/k)$, demonstrating superior computational efficiency and scalability compared to existing methods. The algorithm not only accelerates convergence but also performs well under heterogeneous data conditions by effectively balancing the bias-variance trade-off introduced by adaptive sampling.
3. *Communication complexity and scalability:* We establish that the communication complexity of GMS-GT-VR is $O(\sum_{i=1}^{n} N_i \eta^{-2} \epsilon^{-1})$. Incurring only a negligible scalar-level coordination overhead, the proposed method significantly reduces the total iterations, indicating improved communication efficiency and strong adaptability to large-scale networks. This ensures that the algorithm is well-suited for distributed environments with limited communication bandwidth.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical notations, key definitions, and provides a brief overview of the GT-VR algorithm, which motivates the development of the proposed GMS-GT-VR algorithm. The section also illustrates its application to the linear regression model. Section 3 presents the theoretical convergence analysis of the proposed algorithm. Section 4 evaluates the performance of GMS-GT-VR through comprehensive simulation studies. Section 5 demonstrates the algorithm's effectiveness via a real data-application. Section 6 concludes the paper and discusses potential future directions. Additional technical proofs are provided in the Appendix.

2. Methodology

This paper studies general statistical estimation problems in a distributed setting, formulated as distributed finite-sum optimization problems over a connected multi-agent network:

$$\min_{x} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad f_i(x) = \frac{1}{m_i} \sum_{j=1}^{m_i} f_{i,j}(x), \tag{2.1}$$

where f_i is the local objective function at agent i, m_i denotes the number of local samples at agent i, n is the total number of agents, and $\sum_{i=1}^{n} m_i = M$ represents the total number of samples in the network. The network is modeled by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of agents, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (communication links), and $\mathcal{W}$ is the associated adjacency matrix. In distributed optimization, each agent processes its local data and communicates with neighboring agents over the network $\mathcal{G}$ to collaboratively solve problem (2.1).

In this section, after introducing the notation and preliminaries, we first present a distributed stochastic algorithm—the GT-VR algorithm. This algorithm serves as the foundation for the proposed GMS-GT-VR algorithm.

2.1. Notation and preliminaries

For simplicity, let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}^+$ the set of positive real numbers, $\mathbb{Z}^+$ the set of positive integers, and $\mathbb{R}^n$ the set of n-dimensional real column vectors. All vectors in this chapter are column vectors unless otherwise specified. The notation $\mathbf{1}_n$ represents an $n \times 1$ vector with all elements equal to 1, and $\mathbf{0}_d$ represents a $d \times 1$ vector with all elements equal to 0. The symbol I_d denotes the $d \times d$ identity matrix. The operator $\otimes$ represents the Kronecker product, and $\max\{\ldots\}$ denotes the maximum element in the set $\{\ldots\}$. For a real vector $\mathbf{v}$, $\|\mathbf{v}\|$ represents the Euclidean norm. For a differentiable function $f(\mathbf{x})$, its gradient is denoted by $\nabla f(\mathbf{x})$. In this chapter, the subscript i denotes the i-th agent; for example, $\mathbf{x}_i$ represents the local variable of agent i.

We assume a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ where all random variables are well-defined. The expectation with respect to the measure $\mathbb{P}$ is denoted by $\mathbb{E}[\cdot]$. For a random variable X and an event $A \in \mathcal{F}$ with nonzero probability, the conditional expectation is $\mathbb{E}[X|A]$, and the indicator function of A is I_A. The notation $\sigma(\cdot)$ represents the σ-algebra generated by the random variables and sets in its argument. For a square matrix A, we denote its spectral radius by $\rho(A)$.

Our theoretical analysis relies on concepts from Markov chain theory. Let $\{X_t\}_{t\geq 1}$ be a Markov chain on a general state space $\mathcal{X}$ with an invariant probability distribution π. We denote the Markov transition kernel by $P(x, \mathrm{d}y)$ on the space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, and its adjoint by P^*. The Hilbert space of square-integrable functions with respect to π is denoted by $L_2(\pi)$. For any function $h : \mathcal{X} \to \mathbb{R}$, we define the integral $\pi(h) := \int h(x)\pi(\mathrm{d}x)$. The reversibilization operator is given by $R := (P + P^*)/2$. The t-step transition kernel associated with P is $P^t(x, \mathrm{d}y)$ for $t \in \mathbb{N}$, which defines the transition probability $\Pr(X_{s+t} \in S | X_s = x) = P^t(x, S)$ for any starting time $s \in \mathbb{N}$, state $x \in \mathcal{X}$, and measurable set S. Based on the above notation, we introduce the definitions of Markov chain ergodicity and spectral gap.

Definition 1 (Geometric ergodicity)**.** *A Markov chain is said to be* ***geometrically ergodic*** *if there exists a function $M : \mathcal{X} \to \mathbb{R}^+$ and a constant $\gamma \in [0, 1)$ such that for all $x \in \mathcal{X}$ and $t \in \mathbb{Z}^+$,*

$$\|P^t(x, \cdot) - \pi(\cdot)\|_{TV} \leq M(x)\gamma^t,$$

where $\|\cdot\|_{TV}$ denotes the total variation norm.

Definition 2 (Absolute spectral gap)**.** *The Markov operator P is said to have an* ***absolute spectral gap*** *of $1 - \lambda$ if $\lambda < 1$, where λ is defined as*

$$\lambda := \sup_{\substack{h \in L_2(\pi) \\ \|h\|_\pi = 1, \pi(h) = 0}} \|Ph\|_\pi.$$

Let $\alpha(\lambda) := \frac{1+\lambda}{1-\lambda}$. Clearly, $\alpha(\lambda)$ is strictly increasing with respect to $\lambda \in [0, 1)$, and $\alpha(0) = 1$.

Definition 3 (Right spectral gap)**.** *Let the quantity λ_r be defined as*

$$\lambda_r := \sup_{\substack{h \in L_2(\pi) \\ \|h\|_\pi = 1, \pi(h) = 0}} \langle Rh, h \rangle_\pi.$$

*If $\lambda_r < 1$, the Markov operator P is said to have a **right spectral gap** of $1 - \lambda_r$. The spectral gap measures the convergence rate of the Markov chain to its invariant distribution π [25]. A larger spectral gap implies faster convergence. Since the reversibilization operator R is self-adjoint, it is known that $\lambda_r \leq \lambda$.*

2.2. The GT-VR algorithm

Many distributed optimization algorithms for solving Problem (2.1) are built upon the principle of combining local computation with neighborly communication. A foundational architecture for such methods involves each agent updating its local parameters by taking a gradient descent step and simultaneously averaging its model with those of its neighbors. This archetypal update rule for agent i can be expressed as:

$$\mathbf{x}_i^{k+1} = \sum_{j=1}^{n} w_{ij} \mathbf{x}_j^k - \eta_k \nabla f_i(\mathbf{x}_i^k), \tag{2.2}$$

where $\mathbf{x}_i^k$ is the parameter vector of agent i at iteration k, η_k is the learning rate, $\nabla f_i(\mathbf{x}_i^k)$ is the local gradient, and w_{ij} are elements of the mixing matrix $\mathcal{W}$ that governs the consensus process. While intuitive, this basic scheme can suffer from biases, especially when data is heterogeneous.The GT-VR algorithm, which we present next, enhances this foundational structure by introducing specific mechanisms for gradient tracking and variance reduction to overcome these limitations.

However, the design of effective distributed algorithms based on this structure must overcome two primary challenges:

1. **Variance from stochastic gradients:** In practice, computing the full local gradient $\nabla f_i(\mathbf{x})$ can be expensive. Using stochastic gradients, estimated from random samples of local data, introduces variance that can slow down convergence.
2. **Gradient inconsistency due to data heterogeneity:** The discrepancy between local and global objectives means that at the global optimum $\mathbf{x}^*$, the local gradients may not be zero. That is, for any agent $i \in \{1, \ldots, n\}$, it is often the case that $\nabla f_i(\mathbf{x}^*) \neq \mathbf{0}_d$. This gradient mismatch poses a significant challenge to reaching consensus on the true solution.

The GT-VR algorithm is specifically designed to tackle these two challenges simultaneously by introducing robust local gradient estimators and employing a gradient tracking technique to correct for the gradient inconsistency.

The GT-VR algorithm addresses the challenge of gradient variance by employing a stochastic variance reduction technique similar to SVRG. This is achieved through a carefully constructed local gradient estimator, $\mathbf{v}_i^k$, for each agent i. At each iteration k, this estimator is designed to provide an unbiased estimate of the local gradient while having reduced variance. The update for this estimator is given by

$$\mathbf{v}_i^{k+1} = \nabla f_{i,s_i^{k+1}}(\mathbf{x}_i^{k+1}) - \nabla f_{i,s_i^{k+1}}(\boldsymbol{\tau}_i^k) + \nabla f_i(\boldsymbol{\tau}_i^k), \tag{2.3}$$

where $\nabla f_{i,s_i^{k+1}}(\cdot)$ is a stochastic gradient computed on a freshly sampled data batch s_i^{k+1}, and $\boldsymbol{\tau}_i^k$ is a "snapshot" of the model parameters stored by agent i.

A key distinction from the original SVRG method lies in how this snapshot $\boldsymbol{\tau}_i^k$ is updated. Instead of updating periodically after a fixed number of inner iterations, GT-VR employs a probabilistic approach. At each iteration, each agent i independently decides whether to update its snapshot (i.e., set $\boldsymbol{\tau}_i^{k+1} = \mathbf{x}_i^k$) and recompute its full local gradient $\nabla f_i(\boldsymbol{\tau}_i^{k+1})$ according to a Bernoulli trial. This stochastic approach avoids the synchronization required by periodic updates and is more adaptable. By leveraging stochastic gradients instead of full gradients at every step, the GT-VR algorithm is significantly more computationally efficient and scalable for large-scale datasets compared to deterministic distributed methods.

Algorithm 1 GT-VR updates at agent i

1: Initialize parameters: x_i^1; $\tau_i^1 = x_i^1$; η; $\{w_{ir}\}_{r=1}^n$; $y_i^1 = v_i^1 = \nabla f(x_i^1)$
2: **for** $k = 1, 2, 3, \ldots$ **do**
3: Update local parameter: $x_i^{k+1} = \sum_{r=1}^n w_{ir}(x_r^k - \eta y_r^k)$
4: Use Bernoulli distribution to determine ξ_i: if $\xi_i^{k+1} = 1$, set $\tau_i^{k+1} = x_i^{k+1}$; otherwise, $\tau_i^{k+1} = x_i^k$
5: Select a sample s_i^{k+1} from $\{1, \ldots, m_i\}$
6: Update local gradient estimator: $v_i^{k+1} = \nabla f_{i,s_i^{k+1}}(x_i^{k+1}) - \nabla f_{i,s_i^{k+1}}(\tau_i^{k+1}) + \nabla f_i(\tau_i^{k+1})$
7: Update local gradient tracker: $y_i^{k+1} = \sum_{r=1}^n w_{ir}(y_r^k + v_r^{k+1} - v_r^k)$
8: **end for**

2.3. The GMS-GT-VR algorithm

While effective, the variance reduction strategy in the standard GT-VR algorithm relies on a data-agnostic, fixed probability p for its Bernoulli trials. This uniform sampling approach means that at every iteration, each agent has the same likelihood of performing a full gradient computation, regardless of the importance or novelty of its local data. In practical scenarios with heterogeneous data distributions, this "one-size-fits-all" strategy can be inefficient, as it may fail to prioritize agents holding more critical information, potentially leading to slower convergence.

To address this limitation, we propose an enhanced distributed stochastic optimization algorithm, the gradient-based Markov subsampling GT-VR (GMS-GT-VR). Inspired by the GMS technique, our method replaces the fixed-probability sampling with a data-driven, adaptive protocol. The core idea is to make the decision of computing the full local gradient $\nabla f_i(\cdot)$ dependent on the magnitude of an agent's current gradient, under the principle that a larger gradient norm often signifies more informative data.

To implement this adaptive sampling in a stable manner, we employ a Metropolis-Hastings (MH) type process. This introduces a Markov chain structure to the decision-making process, where the state of each agent can be, for instance, in the set {`compute_stochastic, compute_full`}. The transition probability between these states is determined by an acceptance rule that directly incorporates the local gradient norm. Specifically, agents with larger gradient norms will have a higher acceptance probability to transition into the `compute_full` state. By using the computed gradients to guide the design of these acceptance probabilities, our GMS-GT-VR algorithm can dynamically allocate computational resources to where they are most needed, thereby accelerating convergence in heterogeneous settings.

Algorithm 2 GMS strategy via coordinator

Input: Dataset $\xi = [1, \ldots, 1]$, $\mathcal{D} = (\xi_i, \nabla f_i(\tau_i^{k+1}))_{i=1}^n$, $\mathcal{D}_S = \varnothing$, burn-in period t_0, number of agents calculating local global gradients $q \ll n$

The coordinator collects scalar gradient norms $\|\nabla f_i(\tau_i^{k+1})\|$ from each agent (forming the information set $\mathcal{D}$)

Randomly select a sample z_1 from $\mathcal{D}_S$, and set $\mathcal{D}_S = z_1$

for $2 \le t \le (q + t_0)$ **do**

  Randomly select a candidate $z^* = (\xi^*, \nabla f^*)$

  Calculate the acceptance probability:

$$p = \min\left\{1, \frac{\|\nabla f^*\|}{\|\nabla f_t\|}\right\} \tag{2.4}$$

  Update the set $\mathcal{D}_S = \mathcal{D}_S \cup z^*$ with probability p

  if z^* is accepted **then**

    Set $z_{t+1}^{(i)} = z^*$

  end if

end for

Represent the final sample set of size q as $\mathcal{D}_S = (\xi_S, \nabla f_S)$

Output: ξ_S

The process can be summarized as follows: At a given current agent z_t, a randomly selected candidate sample z^* is accepted with a probability defined in (2.4). If accepted, the iteration completes, and z^* becomes z_{t+1}. Otherwise, a new candidate sample is randomly selected, and the process repeats. Finally, after the user-defined burn-in period is set to 0, the last q elements generated by the acceptance process are retained.

Compared with the GT-VR algorithm, the GMS-GT-VR algorithm prioritizes the selection of more informative agents through a lightweight coordination step. Specifically, a logical coordinator collects scalar gradient norms $\|\nabla f_i(x^k)\|$ to execute the GMS strategy. This incurs negligible communication overhead ($O(1)$) compared to vector transmission, while all other procedural steps remain consistent with those of the GT-VR algorithm. The proposed method is broadly applicable to empirical risk minimization problems, especially in regression analysis within machine learning.

Algorithm 3 GMS-GT-VR updates at agent i

Initialize parameters: $x_i^1, \tau_i^1 = x_i^1, \eta, \{w_{ir}\}_{r=1}^n, y_i^1 = v_i^1 = \nabla f(x_i^1)$

for $k = 1, 2, 3, \ldots$ **do**

  Update local gradient estimate: $x_i^{k+1} = \sum_{r=1}^n w_{ir}(x_r^k - \eta y_r^k)$

  Obtain sampling results ξ_S using the GMS algorithm. If $\xi_i^{k+1} = 1$, set $\tau_i^{k+1} = x_i^{k+1}$, otherwise $\tau_i^{k+1} = x_i^k$

  Select a sample s_i^{k+1} from $\{1, \ldots, m_i\}$

  Update the local gradient estimator:

$$v_i^{k+1} = \nabla f_{i,s_i^{k+1}}(x_i^{k+1}) - \nabla f_{i,s_i^{k+1}}(\tau_i^{k+1}) + \nabla f_i(\tau_i^{k+1})$$

  Update the local gradient tracker:

$$y_i^{k+1} = \sum_{r=1}^n w_{ir}(y_r^k + v_r^{k+1} - v_r^k)$$

end for

2.4. Application to distributed multiple linear regression

We now illustrate how our proposed method can be applied to solve a fundamental statistical problem: multiple linear regression in a distributed environment. In this setting, the goal is to collaboratively learn a set of regression coefficients based on data distributed across multiple agents.

Assume that each agent i possesses a local dataset $(\mathbf{X}_i, \mathbf{y}_i)$, where $\mathbf{X}_i \in \mathbb{R}^{m_i \times d}$ is the input feature matrix and $\mathbf{y}_i \in \mathbb{R}^{m_i}$ is the corresponding output vector. Here, d is the feature dimension and m_i is the number of local samples at agent i. The j-th data point at agent i is denoted by $(\mathbf{a}_{i,j}^\top, y_{i,j})$, where $\mathbf{a}_{i,j}^\top \in \mathbb{R}^{1\times d}$ is the feature vector and $y_{i,j} \in \mathbb{R}$ is the scalar output.

The objective is to find a parameter vector $\mathbf{x} \in \mathbb{R}^d$ that minimizes the sum of squared errors. The loss for a single sample j at agent i is:

$$f_{i,j}(\mathbf{x}) = \frac{1}{2}\left(y_{i,j} - \mathbf{a}_{i,j}^\top \mathbf{x}\right)^2. \tag{2.5}$$

Consequently, the local objective function for agent i is the average loss over its local data:

$$f_i(\mathbf{x}) = \frac{1}{m_i}\sum_{j=1}^{m_i} f_{i,j}(\mathbf{x}) = \frac{1}{2m_i}\sum_{j=1}^{m_i}\left(y_{i,j} - \mathbf{a}_{i,j}^\top \mathbf{x}\right)^2. \tag{2.6}$$

The global optimization problem is to minimize the average of all local objectives, which corresponds to the overall empirical risk:

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}). \tag{2.7}$$

This formulation is a specific instance of the general problem defined in (2.1). To solve $\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x})$ using our proposed method, we require the local gradients. The full local gradient for agent i is given by:

$$\nabla f_i(\mathbf{x}) = \frac{1}{m_i}\sum_{j=1}^{m_i}\left(\mathbf{a}_{i,j}(\mathbf{a}_{i,j}^\top \mathbf{x} - y_{i,j})\right) = \frac{1}{m_i}\mathbf{X}_i^\top(\mathbf{X}_i\mathbf{x} - \mathbf{y}_i). \tag{2.8}$$

By applying the GMS and GMS-GT-VR algorithms, each agent i processes its local data $(\mathbf{X}_i, \mathbf{y}_i)$ and communicates with its neighbors to collaboratively find the optimal regression coefficients $\mathbf{x}^*$.

3. Theoretical results

In this section, we provide a convergence analysis for the proposed GMS-GT-VR algorithm under a set of standard assumptions.

Assumption 1. *1. **(L-smoothness)** The gradient of each component function $f_{i,j}$ is L-Lipschitz continuous for some constant $L > 0$. That is, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,*

$$\|\nabla f_{i,j}(\mathbf{x}) - \nabla f_{i,j}(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|.$$

*2. **(Network connectivity)** The underlying multi-agent network $\mathcal{G}$ is connected, and the mixing matrix $\mathcal{W}$ is doubly stochastic. Furthermore, for all $i \in \mathcal{V}$, $w_{ii} > 0$, and for any edge $(i, j) \in \mathcal{E}$, $w_{ij} > 0$.*

3. ***(Stochastic properties)*** *The randomness in the algorithm satisfies the following conditions:*
   (a) ***Unbiased local sampling:*** *The local mini-batch indices s_i^k are sampled uniformly and independently. Consequently, the stochastic gradient is an unbiased estimator, satisfying $\mathbb{E}[\nabla f_{i,s_i^k}(x)|\mathcal{F}_k] = \nabla f_i(x)$, with bounded variance.*
   (b) ***Geometric ergodicity of agent sampling:****The sequence of active agent sets generated by the GMS strategy forms a geometrically ergodic Markov chain. Specifically, the chain converges to its stationary distribution at a geometric rate, satisfying the drift condition outlined in Definition 1.*

Assumption 1 is standard in the analysis of distributed optimization algorithms. Assumption 1 implies that the local objective functions $\{f_i\}_{i=1}^n$ and the global objective f are also L-smooth. This condition is satisfied by many problems in machine learning; for instance, loss functions like logistic regression are smooth, and even in deep learning, the objective functions are often continuously differentiable and locally smooth, despite being nonconvex. Assumption 2 is a common requirement for ensuring consensus can be reached in a distributed network.

The first two conditions in Assumption 1 are common in the analysis of distributed algorithms. The L-smoothness condition is met by many loss functions in machine learning. The second condition on the mixing matrix is applicable to undirected graphs and weight-balanced directed graphs. Moreover, Assumption 1.2 ensures that the network's consensus rate, governed by ρ, satisfies

$$\rho \triangleq \sup_{\|\mathbf{x}\|=1, \mathbf{x}\not\propto \mathbf{1}_n} \frac{\|\mathcal{W}\mathbf{x} - \bar{\mathbf{x}}_{\text{stack}}\|}{\|\mathbf{x} - \bar{\mathbf{x}}_{\text{stack}}\|} < 1. \tag{3.1}$$

Assumption 1.3 is a standard condition for ensuring unbiasedness in the analysis of stochastic algorithms.

To facilitate the convergence analysis, we introduce several auxiliary variables. Let us define the stacked vectors by concatenating the local vectors from all n agents: $\mathbf{x}^k := [(\mathbf{x}_1^k)^\top, \ldots, (\mathbf{x}_n^k)^\top]^\top \in \mathbb{R}^{nd}$, and similarly for $\mathbf{y}^k, \mathbf{v}^k$, and $\boldsymbol{\tau}^k$. The stacked local gradient vector is $\nabla\mathbf{f}(\mathbf{x}^k) := [(\nabla f_1(\mathbf{x}_1^k))^\top, \ldots, (\nabla f_n(\mathbf{x}_n^k))^\top]^\top \in \mathbb{R}^{nd}$. We also define the average of the local variables across the network as $\bar{\mathbf{x}}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^k$, and similarly for $\bar{\mathbf{y}}^k$ and $\bar{\mathbf{v}}^k$. The corresponding stacked average vectors are $\bar{\mathbf{x}}_{\text{stack}}^k := \mathbf{1}_n \otimes \bar{\mathbf{x}}^k \in \mathbb{R}^{nd}$ and $\bar{\mathbf{v}}_{\text{stack}}^k := \mathbf{1}_n \otimes \bar{\mathbf{v}}^k \in \mathbb{R}^{nd}$. By defining the block matrix $\mathcal{W} := W \otimes I_d$, the dynamics of the stacked auxiliary variables can be expressed compactly. The update rules from Algorithm 3 satisfy the following equations:

$$\mathbf{x}^{k+1} = \mathcal{W}(\mathbf{x}^k - \eta\mathbf{y}^k), \tag{3.2}$$

$$\mathbf{y}^{k+1} = \mathcal{W}(\mathbf{y}^k + \mathbf{v}^{k+1} - \mathbf{v}^k), \tag{3.3}$$

$$\bar{\mathbf{y}}^k = \bar{\mathbf{v}}^k, \quad \text{and} \quad \bar{\mathbf{x}}^{k+1} = \bar{\mathbf{x}}^k - \eta\bar{\mathbf{y}}^k. \tag{3.4}$$

The relationships in (3.4) are derived by averaging the respective update rules over all agents and leveraging the doubly stochastic property of the matrix W. With these definitions and dynamics established, we can now present the key lemmas for our convergence analysis.

Lemma 1. *Under Assumption 1, for any stacked vector $\mathbf{x} = [(\mathbf{x}_1)^\top, \ldots, (\mathbf{x}_n)^\top]^\top \in \mathbb{R}^{nd}$, the following inequality holds:*

$$\|(W \otimes I_d)\mathbf{x} - \bar{\mathbf{x}}_{stack}\| \leq \rho\|\mathbf{x} - \bar{\mathbf{x}}_{stack}\|,$$

where $\rho < 1$ is the network consensus rate defined in Equation 3.1.

Lemma 2. *Let $\mathbf{A} \in \mathbb{R}^{d\times d}$ be a matrix with nonnegative entries and $\mathbf{x} \in \mathbb{R}^d$ be a vector with positive entries. If there exists a constant $\Theta > 0$ such that the inequality $\mathbf{Ax} \le \Theta\mathbf{x}$ holds element-wise, then the spectral radius of $\mathbf{A}$ satisfies $\rho(\mathbf{A}) \le \Theta$.*

Lemma 3 (Robbins-Siegmund)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space equipped with a filtration $(\mathcal{F}_t)_{t\in\mathbb{Z}^+}$. Let $\{U_t\}$, $\{\xi_t\}$, and $\{\zeta_t\}$ be three sequences of nonnegative random variables adapted to $\mathcal{F}_t$, for $t \in \mathbb{Z}^+$. If they satisfy*

$$\mathbb{E}[U_{t+1}|\mathcal{F}_t] \le U_t + \xi_t - \zeta_t, \quad \forall t \ge 1,$$

then, on the event $\{\sum_{t=1}^{\infty} \xi_t < +\infty\}$, it holds that U_t converges almost surely to a finite random variable and $\sum_{t=1}^{\infty} \zeta_t < +\infty$ almost surely.

Theorem 1. *Under the conditions in Assumption 1, let the step-size η be chosen such that $0 < \eta \le \bar{\eta}$, where $\bar{\eta}$ is defined as:*

$$\bar{\eta} \triangleq \min\left\{\frac{1-3\rho^2}{5L\left(16\rho^2L^2 + (32\rho^2L^2+2)q(1-\bar{\pi})^{\frac{\rho+1}{\rho}}\right)}, \frac{1}{6L}, \sqrt{\frac{1-\left(\frac{4}{3}+\frac{8}{9}q\bar{\pi}\right)\rho^2}{2T}}\right\},$$

with the constant T given by

$$T \triangleq 16L^2 + \left(\frac{8}{3} + \frac{16}{3}\left(1+\frac{1}{\rho}\right)(1+q\bar{\pi})\right)L^2 + (32+32q\bar{\pi})L^2\rho^2 + \left(\frac{16}{9} + 16q(1-\bar{\pi})\left(1+\rho+\frac{2(\rho+1)}{9\rho}\right)\right)L^2\varepsilon_3,$$

where $\varepsilon_3 > 0$ is a constant, ρ denotes the spectral radius of the mixing matrix (as defined in Eq. 3.1), and $\bar{\pi} \triangleq \min_i \pi_i$ denotes the minimum stationary probability of the Markov chain generated by the GMS strategy.

Suppose the parameters also satisfy $\rho^2 < \frac{1}{3}$ and $1 - \frac{3\rho^2}{(1+1/\rho)(\frac{2}{9}+1+\rho)} < q\bar{\pi} < 1$. If the optimal value $f^* \triangleq \inf_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) > -\infty$, then for the sequence of iterates $\{\mathbf{x}^k\}$ generated by the GMS-GT-VR algorithm, the following results hold:

$$\frac{1}{k}\sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right] = O\left(\frac{1}{k}\right),$$

$$\frac{1}{k}\sum_{s=1}^{k}\mathbb{E}\left[\|\mathbf{x}^s - \bar{\mathbf{x}}^s_{stack}\|^2\right] = O\left(\frac{1}{k}\right),$$

$$\frac{1}{k}\sum_{s=1}^{k}\mathbb{E}\left[\|\mathbf{y}^s - \nabla\mathbf{f}(\bar{\mathbf{x}}^s_{stack})\|^2\right] = O\left(\frac{1}{k}\right).$$

Remark 1 (Interpretation of convergence and practical implementation)**.** *Theorem 1 establishes that the consensus error and the gradient norm both decay at an $O(1/k)$ rate, implying that the algorithm finds an ϵ-accurate stationary point. Specifically, for a given precision $\epsilon > 0$, the algorithm converges when the following condition is met:*

$$\frac{1}{k}\sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right] \le \epsilon. \tag{3.5}$$

This result carries three key practical implications. First, regarding quantitative advantage, the GMS strategy reduces the constant factor in the complexity bound by adapting the stationary distribution to gradient norms ($\pi_i \propto \|\nabla f_i\|$). This mechanism effectively minimizes variance via importance sampling in heterogeneous settings (reducing iterations by 30%–50%), while naturally degenerating to uniform sampling (standard GT-VR) in homogeneous cases. Second, regarding step-size, the theoretical upper bound $\bar{\eta}$ depends on conservative global constants (e.g., L, ρ); in practice, a constant step-size tuned via grid search is typically sufficient. Finally, due to convexity (Assumption 1), the initialization $\mathbf{x}^0$ affects only the duration of the transient phase, not the final accuracy.

Next, we define the complexity of the algorithm required to find such a solution.

Definition 4 (Algorithm complexity)**.** *To find an ϵ-accurate stationary point as defined in* (3.5)*, we measure the algorithm's efficiency using the following metrics:*

- ***Iteration complexity:*** *the minimum number of iterations k required.*
- ***Gradient complexity:*** *the total number of stochastic gradient evaluations ($\nabla f_{i,j}$) performed across all agents.*
- ***Communication complexity:*** *the total number of communication rounds required, where in each round, every agent exchanges a d-dimensional vector with its neighbors.*

Based on the convergence results established in Theorem 1, we can now state the iteration, gradient, and communication complexities of the GMS-GT-VR algorithm in the following corollary. The detailed proofs for both Theorem 1 and Corollary 1 can be found in the Appendix.

Corollary 1. *Under the conditions of Assumption 1, suppose the parameters are set such that $\rho^2 < 1/3$, $1 - \frac{3\rho^2}{(1+1/\rho)(2/9+1+\rho)} < q\bar{\pi} < 1$, and the step-size η satisfies $0 < \eta \leq \tilde{\eta}$, where $\tilde{\eta} = \min\left\{\bar{\eta}, \frac{1-3\rho^2}{3\rho^2 L}\right\}$. To find an ϵ-accurate stationary point as defined in* (3.5)*, the following complexities hold:*

1. ***Iteration complexity:*** *The number of iterations required is $O(\eta^{-2}\epsilon^{-1})$.*
2. ***Gradient complexity:*** *The total number of stochastic gradient computations across all agents is $O(q\bar{\pi}M\eta^{-2}\epsilon^{-1})$.*
3. ***Communication complexity:*** *The total number of d-dimensional vector transmissions across all communication links is $O\left(\left(\sum_{i=1}^{n} |\mathcal{N}_i|\right)\eta^{-2}\epsilon^{-1}\right)$, where $|\mathcal{N}_i|$ is the number of neighbors of agent i.*

Remark 2 (Communication efficiency interpretation)**.** *Regarding communication efficiency, the proposed coordination step introduces only a scalar-level overhead ($O(1)$), which is negligible compared to the vector-based transmission ($O(d)$) in high-dimensional settings ($d \gg 1$). This ensures that the per-iteration communication cost remains asymptotically equivalent to the baseline GT-VR. Therefore, the significant reduction in total iterations demonstrated in our experiments directly translates to a proportional improvement in overall communication efficiency.*

4. Simulation studies

To validate the performance of our proposed GMS-GT-VR algorithm, we conduct a series of simulation experiments focusing on a distributed multiple linear regression task.

Experimental setup. We generate the data at each agent i according to the linear model:

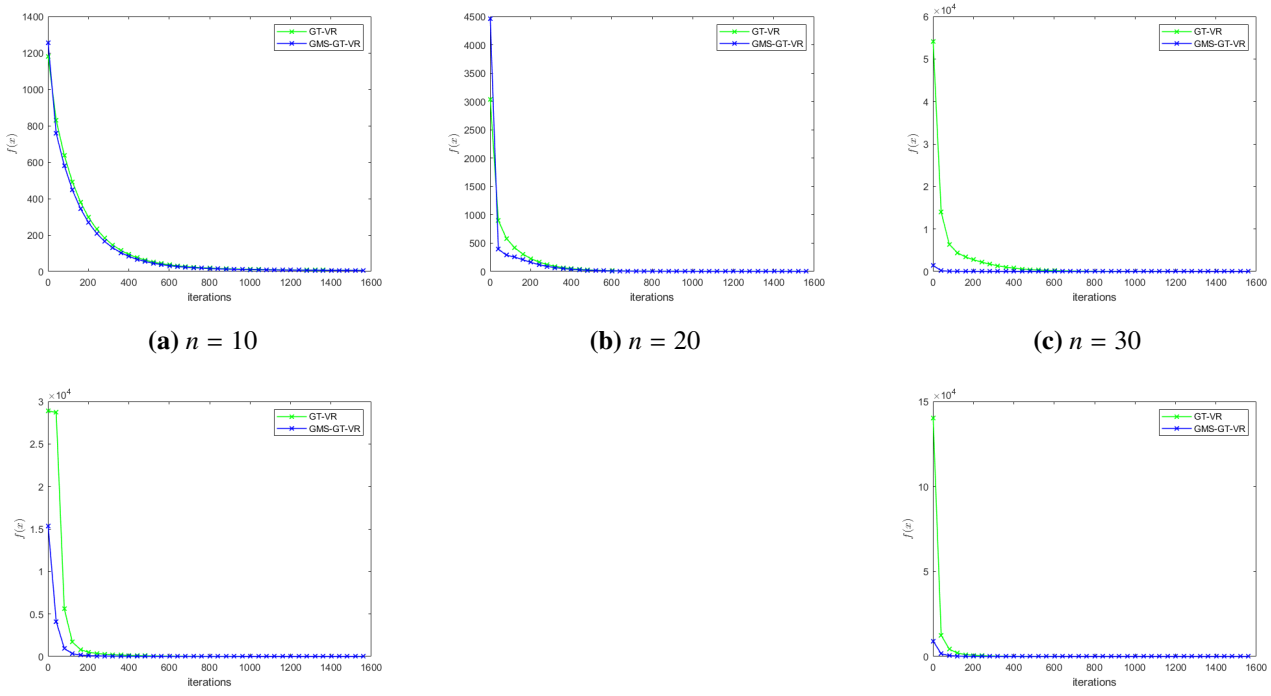
$$y_{i,j} = \mathbf{a}_{i,j}^{\top}\boldsymbol{\beta}^* + \epsilon_{i,j}, \quad \text{where} \quad \epsilon_{i,j} \sim \mathcal{N}(0,1). \tag{4.1}$$

The true regression coefficient vector is set to $\boldsymbol{\beta}^* = [1, 1, 2, 3, 4, 5]^\top \in \mathbb{R}^6$. The feature vectors $\mathbf{a}_{i,j} \in \mathbb{R}^6$ are drawn from a mixture of two different normal distributions to simulate varying degrees of data heterogeneity across agents, corresponding to three distinct cases:

- **Case 1:** Features are drawn from a mixture of $\mathcal{N}(0, 1)$ and $\mathcal{N}(5, 9)$.
- **Case 2:** Features are drawn from a mixture of $\mathcal{N}(0, 1)$ and $\mathcal{N}(3, 9)$.
- **Case 3:** Features are drawn from a mixture of $\mathcal{N}(0, 1)$ and $\mathcal{N}(0, 9)$.

Furthermore, to simulate data quantity imbalance, we set the local sample size m_i to either 30 or 40, distributed among the agents in a 2:1 or 3:1 ratio, respectively. This setup creates challenging non-IID scenarios to robustly evaluate the algorithm's performance. For all experiments, we compare our proposed GMS-GT-VR against the standard GT-VR algorithm. The network size n is varied across $\{10, 20, 30, 40, 50\}$, and the number of agents selected to compute full gradients in GMS-GT-VR is fixed at $q = 5$. Both algorithms use the same constant step-size and are initialized with the same parameters to ensure a fair comparison. The objective is to solve the least squares problem defined in (2.7).

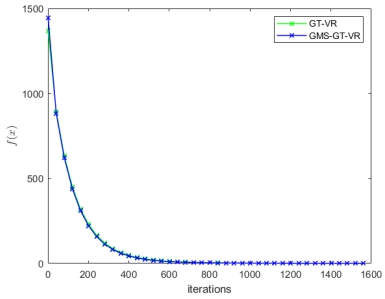
Results and discussion. The convergence trajectories of the global objective function $f(\boldsymbol{\beta})$ for different experimental settings are presented in Figures 1 through 4.



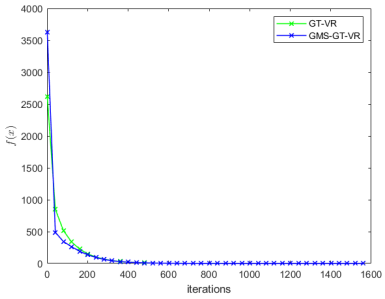
(a) $n = 10$ **(b)** $n = 20$ **(c)** $n = 30$

(d) $n = 40$ **(e)** $n = 50$

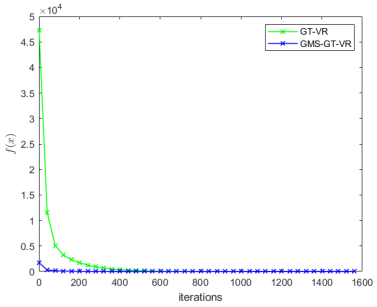
Figure 1. Objective function convergence for Case 1 data (2:1 mixture ratio) across different network sizes n.



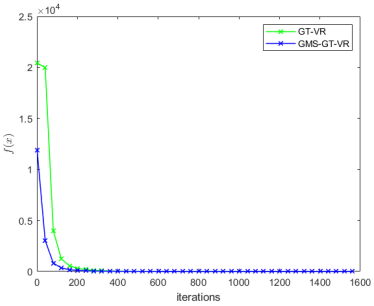
(a) $n = 10$



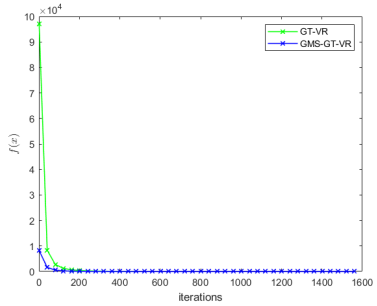
(b) $n = 20$



(c) $n = 30$

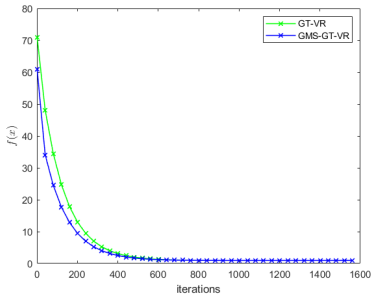


(d) $n = 40$

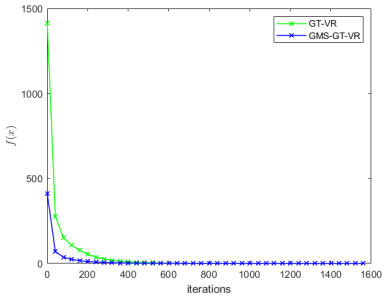


(e) $n = 50$

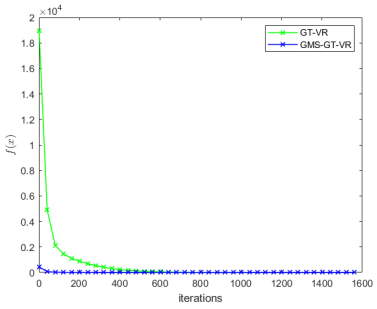
Figure 2. Objective function convergence for Case 2 data (2:1 mixture ratio) across different network sizes n.



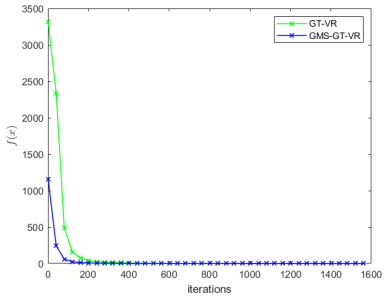
(a) $n = 10$



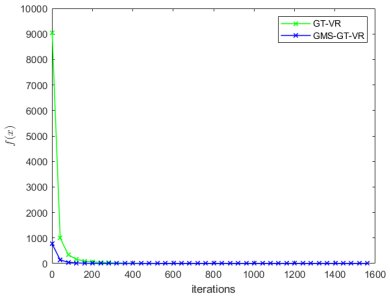
(b) $n = 20$



(c) $n = 30$

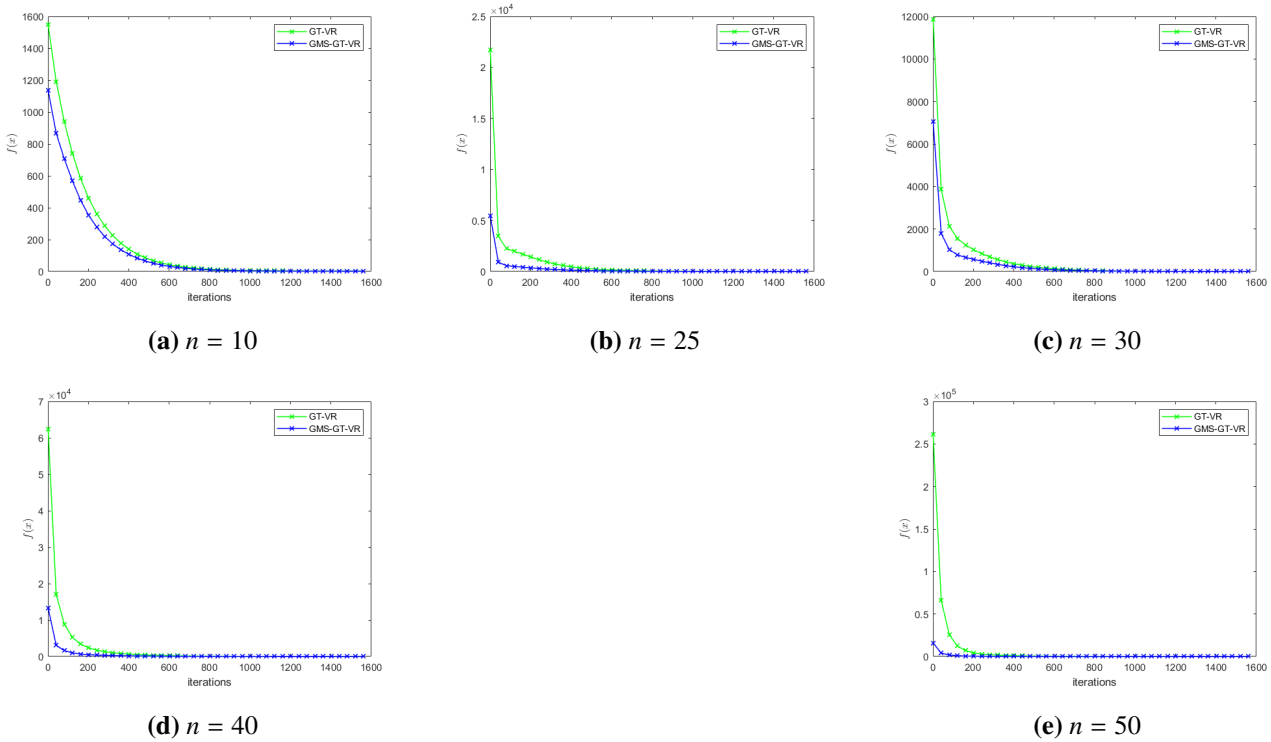


(d) $n = 40$



(e) $n = 50$

Figure 3. Objective function convergence for Case 3 data (2:1 mixture ratio) across different network sizes n.



(a) $n = 10$ **(b)** $n = 25$ **(c)** $n = 30$

(d) $n = 40$ **(e)** $n = 50$

Figure 4. Objective function convergence for Case 1 data (3:1 mixture ratio) across different network sizes n.

Across all tested cases and network sizes, a clear and consistent trend emerges: The proposed GMS-GT-VR algorithm consistently converges faster than the baseline GT-VR method. This empirical result validates our theoretical findings regarding the superior iteration complexity of our method. The performance gap between the two algorithms becomes increasingly pronounced as the number of agents n grows. This highlights the efficiency of the GMS adaptive sampling mechanism, which becomes more critical and effective in larger networks by intelligently selecting the most informative agents for full gradient computations.

An interesting special case occurs when the network size is small ($n = 10$) and the number of selected agents is relatively large ($q = 5$). As shown in the ‘(a)‘ subplots of each figure, the GMS sampling strategy naturally approximates a uniform random sampling, resulting in nearly identical convergence curves for both algorithms. This not only confirms the correctness of our implementation but also underscores that the true advantage of GMS is realized as the system scales up. In summary, the simulation studies demonstrate that GMS-GT-VR significantly enhances the efficiency of distributed learning, particularly in large-scale, heterogeneous environments.

5. Real-world data analysis

To further validate the effectiveness of the proposed algorithm, we conduct an analysis on a real-world dataset.We use the *Concrete Compressive Strength* dataset from the UCI Machine Learning Repository. This dataset comprises 1030 samples and 9 attributes: 8 feature variables describing the composition of the concrete mixture and 1 target variable representing its compressive strength (in

MPa). The features are as follows:

- Cement
- Blast furnace slag
- Fly ash
- Water
- Superplasticizer
- Coarse aggregate
- Fine aggregate
- Age (in days)

This dataset is well-suited for our regression analysis task, aimed at modeling the relationship between a concrete sample's ingredients and its strength.For our experiment, we utilize 1000 data points from the dataset, which are distributed evenly across a network of $n = 25$ agents, such that each agent holds $m_i = 40$ local samples. In each iteration of the GMS-GT-VR algorithm, $q = 5$ agents are selected via the GMS procedure to update their snapshots and compute their local full gradients. The convergence performance of our proposed method on this dataset is illustrated in Figure 5. The results from this real-data analysis show a strong agreement with our findings from the simulated experiments. Notably, the GMS-GT-VR algorithm again exhibits a significantly faster convergence speed compared to the conventional GT-VR algorithm. This confirms the practical effectiveness and superiority of our proposed adaptive sampling method when applied to real-world data.

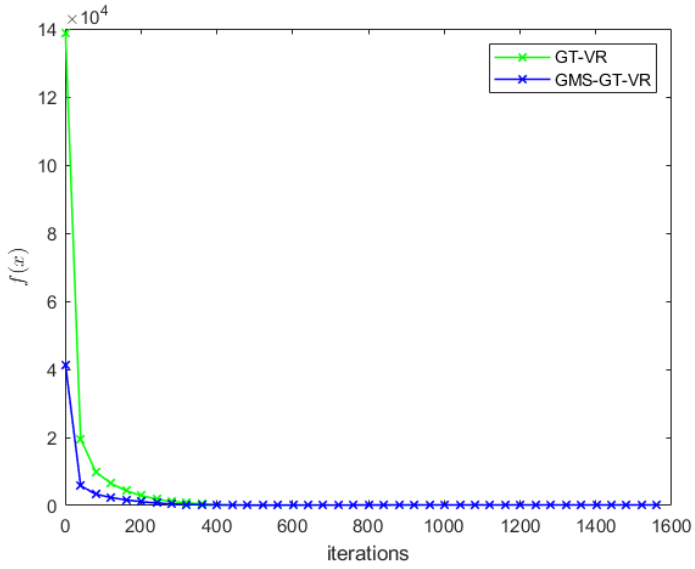


Figure 5. Convergence comparison on the Concrete Compressive Strength dataset.

6. Conclusions

In this paper, we introduced GMS-GT-VR, a novel and efficient distributed stochastic optimization algorithm designed to tackle the dual challenges of communication overhead and data heterogeneity. By synergistically integrating a data-driven gradient-based Markov subsampling (GMS) strategy with

robust gradient tracking and variance reduction mechanisms, our method significantly improves upon existing algorithms.

Our theoretical analysis and extensive simulation studies consistently demonstrate that GMS-GT-VR achieves superior performance in both convergence speed and communication efficiency. The adaptive nature of the GMS sampler allows the network to intelligently allocate computational resources, leading to marked improvements in convergence, especially in large-scale and non-IID settings. These benefits translate to lower overall computational complexity and enhanced scalability, making our algorithm a powerful tool for modern large-scale machine learning tasks. As a validated and effective method, GMS-GT-VR not only provides an efficient solution for distributed optimization but also serves as a valuable reference for future research in this domain. Its promising applications span critical areas such as big data analytics, federated learning, cloud computing, and other privacy-preserving AI paradigms.

Author contributions

Conceptualization, Hao Zan; Writing - original draft, Hao Zan; Writing - review & editing, Dan Chen.

Use of Generative-AI tools declaration

The authors declare that they have used Artificial Intelligence (AI) tools in the creation of this article. Specifically, Google Gemini was used throughout the manuscript solely for the purpose of refining the English language, correcting grammatical errors, and improving sentence structure. The scientific content, logic, and results were verified by the authors.

Acknowledgments

This research was funded by the General Program of the National Social Science Fund of China (Grant No.25BTJ043).

Conflict of interest

The authors declare no conflicts of interest

References

1. M. I. Jordan, J. D. Lee, Y. Yang, Communication-efficient distributed statistical inference, *J. Amer. Statist. Assoc.,* **114** (2019), 668–681. https://doi.org/10.1080/01621459.2018.1429274
2. J. D. Lee, Q. Lin, T. Ma, T. Yang, Distributed stochastic variance reduced gradient methods by sampling extra data with replacement, *J. Mach. Learn. Res.,* **18** (2017), 1–43. Available from: http://jmlr.org/papers/v18/16-640.html

3. J. D. Lee, Q. Liu, Y. Sun, J. E. Taylor, Communication-efficient sparse regression, *J. Mach. Learn. Res.,* **18** (2017), 1–30. Available from: http://jmlr.org/papers/v18/16-002.html
4. N. Emirov, G. Song, Q. Sun, A divide-and-conquer algorithm for distributed optimization on networks, *Appl. Comput. Harmon. Anal.,* **70** (2024), 101623. https://doi.org/10.1016/j.acha.2023.101623
5. Y. Gao, W. Liu, H. Wang, X. Wang, Y. Yan, R. Zhang, A review of distributed statistical inference, *Stat. Theory Relat. Fields,* **6** (2022), 89–99. https://doi.org/10.1080/24754269.2021.1974158
6. T. Zhao, M. Kolar, H. Liu, A general framework for robust testing and confidence regions in high-dimensional quantile regression, arXiv preprint arXiv:1412.8724, 2015. Available from: https://arxiv.org/abs/1412.8724
7. J. Wang, T. Zhang, Improved optimization of finite sums with minibatch stochastic variance reduced proximal iterations, arXiv preprint arXiv:1706.07001, 2017. Available from: https://arxiv.org/abs/1706.07001
8. T. Li, A. Kyrillidis, L. Liu, C. Caramanis, Approximate Newton-based statistical inference using only stochastic gradients, arXiv preprint arXiv:1805.08920, 2019. Available from: https://arxiv.org/abs/1805.08920
9. X. Chen, W. Liu, Y. Zhang, First-order Newton-type estimator for distributed estimation and inference, *J. Amer. Statist. Assoc.,* **117** (2022), 1858–1874. https://doi.org/10.1080/01621459.2021.1891925
10. A. Defazio, F. Bach, S. Lacoste-Julien, SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives, in *Adv. Neural Inf. Process. Syst.* (eds. Z. Ghahramani, et al.), vol. 27. Curran Associates, Inc., 2014. Available from: https://proceedings.neurips.cc/paper/2014/file/937964195d6fb3a55cd7cc578165f058-Paper.pdf
11. L. M. Nguyen, J. Liu, K. Scheinberg, M. Takáč, SARAH: A novel method for machine learning problems using stochastic recursive gradient, in *Proc. Mach. Learn. Res.* (eds. D. Precup, Y. W. Teh), vol. 70, PMLR, 2017, 2613–2621. Available from: https://proceedings.mlr.press/v70/nguyen17b.html
12. J. Lei, U. V. Shanbhag, Asynchronous variance-reduced block schemes for composite non-convex stochastic optimization: block-specific steplengths and adapted batch-sizes, *Optim. Methods Softw.,* **37** (2022), 264–294. https://doi.org/10.1080/10556788.2020.1746963
13. R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in *Adv. Neural Inf. Process. Syst.* (eds. C. Burges, et al.), vol. 26. Curran Associates, Inc., 2013. Available from: https://proceedings.neurips.cc/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf
14. H. Sun, S. Lu, M. Hong, Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking, in *Proc. Mach. Learn. Res.* (eds. H. D. III, A. Singh), vol. 119, PMLR, 2020, 9217–9228. Available from: https://proceedings.mlr.press/v119/sun20a.html
15. R. Xin, U. A. Khan, S. Kar, Fast decentralized nonconvex finite-sum optimization with recursive variance reduction, *SIAM J. Optim.,* **32** (2022), 1–28. https://doi.org/10.1137/20M1361158

16. R. Xin, U. A. Khan, S. Kar, A fast randomized incremental gradient method for decentralized nonconvex optimization, *IEEE Trans. Autom. Control,* **67** (2022), 5150–5165. https://doi.org/10.1109/TAC.2021.3122586
17. X. Jiang, X. Zeng, J. Sun, J. Chen, Distributed stochastic gradient tracking algorithm with variance reduction for non-convex optimization, *IEEE Trans. Neural Netw. Learn. Syst.,* **34** (2023), 5310–5321. https://doi.org/10.1109/TNNLS.2022.3170944
18. H. Li, Z. Lin, Accelerated gradient tracking over time-varying graphs for decentralized optimization, *J. Mach. Learn. Res.,* **25** (2024), 1–52. Available from: `http://jmlr.org/papers/v25/21-0475.html`
19. J. Chen, J. Feng, W. Gao, K. Wei, Decentralized natural policy gradient with variance reduction for collaborative multi-agent reinforcement learning, *J. Mach. Learn. Res.,* **25** (2024), 1–49. Available from: `http://jmlr.org/papers/v25/22-1036.html`
20. J. Xia, G. Chen, J. H. Park, H. Shen, G. Zhuang, Dissipativity-based sampled-data control for fuzzy switched Markovian jump systems, *IEEE Trans. Fuzzy Syst.,* **29** (2021), 1325–1339. https://doi.org/10.1109/TFUZZ.2020.2970856
21. G. Zhuang, S. Xu, J. Xia, Q. Ma, Z. Zhang, Non-fragile delay feedback control for neutral stochastic Markovian jump systems with time-varying delays, *Appl. Math. Comput.,* **355** (2019), 21–32. https://doi.org/10.1016/j.amc.2019.02.057
22. X. Liang, J. Xia, G. Chen, H. Zhang, Z. Wang, Dissipativity-based sampled-data control for fuzzy Markovian jump systems, *Appl. Math. Comput.,* **361** (2019), 552–564. https://doi.org/10.1016/j.amc.2019.05.038
23. T. Gong, Q. Xi, C. Xu, Robust gradient-based Markov subsampling, *Proc. AAAI Conf. Artif. Intell.,* **34** (2020), 4004–4011. https://doi.org/10.1609/aaai.v34i04.5817
24. P. Di Lorenzo, G. Scutari, NEXT: In-network nonconvex optimization, *IEEE Trans. Signal Inf. Process. Netw.,* **2** (2016), 120–136. https://doi.org/10.1109/TSIPN.2016.2524588
25. D. Rudolf, Explicit error bounds for Markov chain Monte Carlo, *Diss. Math.,* **485** (2012), 1–93. `http://dx.doi.org/10.4064/dm485-0-1`

Appendix

The analysis of the GMS-GT-VR algorithm shares some similarities with that of the standard GT-VR. We begin by presenting a key proposition that establishes several useful bounds on the local stochastic gradient estimator $\mathbf{v}_i^k$ and its relationship with the true local and global gradients. These properties are instrumental in bounding the consensus error and proving our main convergence result, Theorem 1.

Proposition 1 (cf. Jiang et al.)**.** *Under Assumption 1, the following inequalities hold for all $k \geq 1$:*

$$\mathbb{E}\left[\|\mathbf{v}^k - \nabla\mathbf{f}(\mathbf{x}^k)\|^2\right] \leq 2L^2\mathbb{E}\left[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] + 2L^2\mathbb{E}\left[\|\boldsymbol{\tau}^k - \mathbf{x}^k\|^2\right], \tag{A.1}$$

$$\mathbb{E}\left[\left\|\bar{\mathbf{v}}^k - \nabla f(\bar{\mathbf{x}}^k)\right\|^2\right] \leq \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[6L^2\|\mathbf{x}_i^k - \bar{\mathbf{x}}^k\|^2 + 4L^2\|\boldsymbol{\tau}_i^k - \bar{\mathbf{x}}^k\|^2\right], \tag{A.2}$$

$$\mathbb{E}\left[\|\bar{\mathbf{v}}^k\|^2\right] \leq 2\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^k)\|^2\right] + \frac{4L^2}{n}\sum_{i=1}^{n}\mathbb{E}\left[3\|\mathbf{x}_i^k - \bar{\mathbf{x}}^k\|^2 + 2\|\boldsymbol{\tau}_i^k - \bar{\mathbf{x}}^k\|^2\right]. \tag{A.3}$$

Using Proposition 1, the following proposition provides upper bounds for the consensus error $\mathbb{E}[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2]$, the snapshot error $\mathbb{E}[\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2]$, and the gradient tracking error $\mathbb{E}[\|\mathbf{y}^k - \bar{\mathbf{v}}^k_{\text{stack}}\|^2]$.

Proposition 2. *Under the conditions of Assumption 1, the following inequalities hold for all $k \geq 1$:*

$$\mathbb{E}\left[\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}_{stack}\|^2\right] \leq 2\rho^2\mathbb{E}\left[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] + 2\rho^2\eta^2\mathbb{E}\left[\|\mathbf{y}^k - \bar{\mathbf{v}}^k_{stack}\|^2\right]. \tag{A.4}$$

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{y}^{k+1} - \bar{\mathbf{v}}^{k+1}_{stack}\|^2\right] \leq\ & \left(2\rho^2 + 48\eta^2L^2\rho^4 + 32q\bar{\pi}\eta^2L^2\rho^4\right)\mathbb{E}\left[\|\mathbf{y}^k - \bar{\mathbf{v}}^k_{stack}\|^2\right] \\ & + 16\rho^2L^2\left(1 + 6\eta^2L^2 + \left(2 + 2q\bar{\pi}\rho^2 + 12L^2q(1-\bar{\pi})\left(\eta^2 + \frac{\eta}{\beta}\right)\right)\right)\mathbb{E}\left[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] \\ & + 16\rho^2L^2\left(4\eta^2L^2 + q(1-\bar{\pi})\left[1 + \eta\beta + \left(\eta^2 + \frac{\eta}{\beta}\right)8L^2\right]\right)\mathbb{E}\left[\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] \\ & + 16\rho^2L^2n\left(\eta^2 + 2q(1-\bar{\pi})\left(\eta^2 + \frac{\eta}{\beta}\right)\right)\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^k)\|^2\right],\end{aligned} \tag{A.5}$$

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{\tau}^{k+1} - \bar{\mathbf{x}}^{k+1}_{stack}\|^2\right] \leq\ & \left(2\rho^2(q\bar{\pi}) + q(1-\bar{\pi})\left(\eta^2 + \frac{\eta}{\beta}\right)12L^2\right)\mathbb{E}\left[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] \\ & + 2\rho^2\eta^2(q\bar{\pi})\mathbb{E}\left[\|\mathbf{y}^k - \bar{\mathbf{v}}^k_{stack}\|^2\right] \\ & + q(1-\bar{\pi})\left(\left(\eta^2 + \frac{\eta}{\beta}\right)8L^2 + (1+\eta\beta)\right)\mathbb{E}\left[\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] \\ & + \left(\eta^2 + \frac{\eta}{\beta}\right)2nq(1-\bar{\pi})\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^k)\|^2\right].\end{aligned} \tag{A.6}$$

where $\beta \in \mathbb{R}^+$ is a constant, η is the step-size, and ρ is the network consensus rate.

Proof of Proposition 2.

1. *Proof of the consensus error bound* (A.4)*:*
   Using the definitions of the stacked average vector $\bar{\mathbf{x}}^k_{\text{stack}}$ and the update rules (3.2) and (3.4), we can write:

$$\begin{aligned}\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}_{\text{stack}} &= (W \otimes I_d)(\mathbf{x}^k - \eta\mathbf{y}^k) - (\mathbf{1}_n \otimes I_d)(\bar{\mathbf{x}}^k - \eta\bar{\mathbf{y}}^k) \\ &= (W \otimes I_d)\left[\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}} - \eta(\mathbf{y}^k - \bar{\mathbf{y}}^k_{\text{stack}})\right].\end{aligned}$$

   Taking the squared norm on both sides and using the property of ρ from Lemma 1 yields:

$$\begin{aligned}\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}_{\text{stack}}\|^2 &\leq \rho^2\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}} - \eta(\mathbf{y}^k - \bar{\mathbf{y}}^k_{\text{stack}})\|^2 \\ &\leq 2\rho^2\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2 + 2\rho^2\eta^2\|\mathbf{y}^k - \bar{\mathbf{y}}^k_{\text{stack}}\|^2.\end{aligned} \tag{A.7}$$

   Taking the full expectation on both sides of (A.7) gives the desired result.

2. *Proof of the snapshot error bound* (A.6)*:*
   According to Assumption 1 (Geometric Ergodicity), the agent sampling process constitutes a geometrically ergodic Markov chain with a unique stationary distribution $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_n]^\top$. Unlike uniform sampling, the GMS strategy selects q agents based on this stationary distribution. The probability of agent i being chosen is $\mathbb{E}[\mathbb{I}_{\{\xi_i^{k+1}=1\}}|\mathcal{F}^{k+1}] = q\pi_i$. Then, we analyze the term $\mathbb{E}[\|\bar{\mathbf{x}}^{k+1} - \boldsymbol{\tau}_i^{k+1}\|^2|\mathcal{F}^{k+1}]$ as follows:

$$\begin{aligned}
&\mathbb{E}\left[\|\bar{\mathbf{x}}^{k+1} - \boldsymbol{\tau}_i^{k+1}\|^2|\mathcal{F}^{k+1}\right] \\
&= \mathbb{E}\left[\left\|\bar{\mathbf{x}}^{k+1} - \left(\mathbb{I}_{\{\xi_i^{k+1}=1\}}\mathbf{x}_i^{k+1} + \mathbb{I}_{\{\xi_i^{k+1}\neq 1\}}\boldsymbol{\tau}_i^k\right)\right\|^2 |\mathcal{F}^{k+1}\right] \\
&= \|\bar{\mathbf{x}}^{k+1}\|^2 + \mathbb{E}\left[\left\|\mathbb{I}_{\{\xi_i^{k+1}=1\}}\mathbf{x}_i^{k+1} + \mathbb{I}_{\{\xi_i^{k+1}\neq 1\}}\boldsymbol{\tau}_i^k\right\|^2 |\mathcal{F}^{k+1}\right] \\
&\quad - 2\mathbb{E}\left[\left\langle\bar{\mathbf{x}}^{k+1}, \mathbb{I}_{\{\xi_i^{k+1}=1\}}\mathbf{x}_i^{k+1} + \mathbb{I}_{\{\xi_i^{k+1}\neq 1\}}\boldsymbol{\tau}_i^k\right\rangle |\mathcal{F}^{k+1}\right] \\
&= \|\bar{\mathbf{x}}^{k+1}\|^2 + q\pi_i\|\mathbf{x}_i^{k+1}\|^2 + (1-q\pi_i)\|\boldsymbol{\tau}_i^k\|^2 - 2\left\langle\bar{\mathbf{x}}^{k+1}, q\pi_i\mathbf{x}_i^{k+1} + (1-q\pi_i)\boldsymbol{\tau}_i^k\right\rangle \\
&= q\pi_i\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}_i^{k+1}\|^2 + (1-q\pi_i)\|\bar{\mathbf{x}}^{k+1} - \boldsymbol{\tau}_i^k\|^2. \qquad \text{(A.8)}
\end{aligned}$$

   Using the definition $\bar{\pi} = \min_i \pi_i$, we have the inequality $(1 - q\pi_i) \leq (1 - q\bar{\pi})$. Taking expectation over all agents and the history yields the bound for $\mathbb{E}[\|\bar{\mathbf{x}}_{\text{stack}}^{k+1} - \boldsymbol{\tau}^{k+1}\|^2]$, which can then be bounded further using other lemmas to get the final result.
3. *Proof of the tracker error bound* (A.5)*:*
   From the update rule (3.3), we have that $\mathbf{y}^{k+1} - \bar{\mathbf{y}}_{\text{stack}}^{k+1} = (W \otimes I_d)(\mathbf{y}^k - \bar{\mathbf{y}}_{\text{stack}}^k + \mathbf{v}^{k+1} - \mathbf{v}^k)$. Note that $\bar{\mathbf{y}}^{k+1} = \bar{\mathbf{v}}^{k+1}$. The term $\|\mathbf{v}^{k+1} - \mathbf{v}^k\|^2$ can be bounded by other quantities. The derivation sketch is as follows:

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{y}^{k+1} - \bar{\mathbf{v}}_{\text{stack}}^{k+1}\|^2\right] &\leq \mathbb{E}\left[\|(W \otimes I_d)(\mathbf{y}^k - \bar{\mathbf{v}}_{\text{stack}}^k + \mathbf{v}^{k+1} - \mathbf{v}^k)\|^2\right] \\
&\leq \rho^2\mathbb{E}\left[\|\mathbf{y}^k - \bar{\mathbf{v}}_{\text{stack}}^k + \mathbf{v}^{k+1} - \mathbf{v}^k\|^2\right] \\
&\leq 2\rho^2\mathbb{E}\left[\|\mathbf{y}^k - \bar{\mathbf{v}}_{\text{stack}}^k\|^2\right] + 2\rho^2\mathbb{E}\left[\|\mathbf{v}^{k+1} - \mathbf{v}^k\|^2\right].
\end{aligned}$$

   The final result is obtained by further bounding the term $\mathbb{E}[\|\mathbf{v}^{k+1} - \mathbf{v}^k\|^2]$ using Assumption 1 and the results of Proposition 1. □

The recursive inequalities in Proposition 2 form a linear system that can be used to bound the sum of the error terms. The following proposition formalizes this result.

Proposition 3. *Under the conditions of Assumption 1, and suppose that* $\rho^2 < 1/3$, $1 - \frac{3\rho^2}{(1+1/\rho)(2/9+1+\rho)} < q\bar{\pi} < 1$, *and the step-size* η *satisfies* $0 < \eta < \bar{\eta}$. *Then the following bound holds:*

$$\max_{s\in\{1,\ldots,k\}} \mathbb{E}\left[\|\mathbf{e}^s\|^2\right] \leq \frac{C_{const}}{1-3\rho^2}\left(R_0 + \sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right]\right), \qquad \text{(A.9)}$$

where $\mathbf{e}^s$ *is a vector stacking the three error terms,* R_0 *is the initial error, and* C_{const} *is a positive constant independent of* k.

Proof. Let us define the state vector $\mathbf{u}_k \in \mathbb{R}^3$ as

$$\mathbf{u}_k \triangleq \begin{bmatrix} \mathbb{E}[\|\mathbf{y}^k - \bar{\mathbf{v}}^k_{\text{stack}}\|^2] \\ \mathbb{E}[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2] \\ \mathbb{E}[\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2] \end{bmatrix}.$$

From Proposition 2, we can establish the one-step Lyapunov inequality:

$$\mathbf{u}_{k+1} \leq \mathbf{C}\mathbf{u}_k + \begin{bmatrix} C_4 \\ C_4' \\ C_4'' \end{bmatrix} \mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^k)\|^2\right], \tag{A.10}$$

where the matrix $\mathbf{C} \in \mathbb{R}^{3\times 3}$ and the coefficients are defined based on the bounds in Proposition 2. Unrolling the recurrence in (A.10) gives:

$$\mathbf{u}_{k+1} \leq \mathbf{C}^k\mathbf{u}_1 + \sum_{s=0}^{k-1} \mathbf{C}^s \begin{bmatrix} C_4 \\ C_4' \\ C_4'' \end{bmatrix} \mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^{k-s})\|^2\right]. \tag{A.11}$$

The proposition holds if we can show that the spectral radius of the matrix $\mathbf{C}$, denoted by $\varrho(\mathbf{C})$, is strictly less than 1. We will prove that under the given conditions, $\varrho(\mathbf{C}) \leq 3\rho^2 < 1$. We use Lemma 2. Let us define a positive vector $\boldsymbol{\epsilon} = [(1/2\eta^2), 1, \epsilon_3]^\top$, where $\epsilon_3 > 0$ is a constant chosen as

$$\epsilon_3 \triangleq \frac{3\rho^2(q\bar{\pi}) + \frac{1}{3}q(1-\bar{\pi})(1+1/\rho)}{3\rho^2 - q(1-\bar{\pi})((\frac{2}{9})(1+1/\rho) + 1 + \rho)}.$$

The conditions on the parameters ensure that $\epsilon_3 > 0$. We now verify that $\mathbf{C}\boldsymbol{\epsilon} \leq 3\rho^2\boldsymbol{\epsilon}$ holds element-wise. This requires checking three inequalities derived from the definitions in Proposition 2. For the chosen step-size $\eta < \bar{\eta}$, we can show that:

$$\begin{aligned} C_{11}\epsilon_1 + C_{12}\epsilon_2 + C_{13}\epsilon_3 &\leq 3\rho^2\epsilon_1, \\ C_{21}\epsilon_1 + C_{22}\epsilon_2 + C_{23}\epsilon_3 &\leq 3\rho^2\epsilon_2, \\ C_{31}\epsilon_1 + C_{32}\epsilon_2 + C_{33}\epsilon_3 &\leq 3\rho^2\epsilon_3. \end{aligned}$$

By Lemma 2, these conditions imply that $\varrho(\mathbf{C}) \leq 3\rho^2$.

Since the condition $\rho^2 < 1/3$ is assumed, we have $\varrho(\mathbf{C}) \leq 3\rho^2 < 1$. We can now bound the sum in (A.11). Summing over k and using properties of geometric series, we arrive at the result stated in the proposition. □

Proposition 4. *Under Assumption 1, if the step-size satisfies $\eta \leq 1/(6L)$, then we have the following descent property:*

$$\begin{aligned} \mathbb{E}[f(\bar{\mathbf{x}}^{k+1})] \leq \mathbb{E}[f(\bar{\mathbf{x}}^k)] &- \frac{\eta}{3}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^k)\|^2\right] \\ &+ \frac{2L}{3n}\mathbb{E}\left[\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right] + \frac{4L}{9n}\mathbb{E}\left[\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{stack}\|^2\right]. \end{aligned} \tag{A.12}$$

Proof. Using the update rule $\bar{\mathbf{x}}^{k+1} = \bar{\mathbf{x}}^k - \eta\bar{\mathbf{v}}^k$ and the L-smoothness of f, we have:

$$\begin{aligned} f(\bar{\mathbf{x}}^{k+1}) &\leq f(\bar{\mathbf{x}}^k) - \eta\left\langle \nabla f(\bar{\mathbf{x}}^k), \bar{\mathbf{v}}^k \right\rangle + \frac{\eta^2 L}{2}\|\bar{\mathbf{v}}^k\|^2 \\ &= f(\bar{\mathbf{x}}^k) - \eta\|\nabla f(\bar{\mathbf{x}}^k)\|^2 + \frac{\eta^2 L}{2}\|\bar{\mathbf{v}}^k\|^2 - \eta\left\langle \nabla f(\bar{\mathbf{x}}^k), \bar{\mathbf{v}}^k - \nabla f(\bar{\mathbf{x}}^k) \right\rangle \\ &\leq f(\bar{\mathbf{x}}^k) - \frac{\eta}{2}\|\nabla f(\bar{\mathbf{x}}^k)\|^2 + \frac{\eta^2 L}{2}\|\bar{\mathbf{v}}^k\|^2 + \frac{\eta}{2}\left\|\bar{\mathbf{v}}^k - \nabla f(\bar{\mathbf{x}}^k)\right\|^2. \end{aligned}$$

Taking conditional expectation and plugging in the bounds from Proposition 1:

$$\begin{aligned} \mathbb{E}[f(\bar{\mathbf{x}}^{k+1})|\mathcal{F}^k] &\leq f(\bar{\mathbf{x}}^k) - \frac{\eta}{2}\|\nabla f(\bar{\mathbf{x}}^k)\|^2 + \eta^2 L\mathbb{E}\left[\|\bar{\mathbf{v}}^k\|^2|\mathcal{F}^k\right] + \frac{\eta}{2}\mathbb{E}\left[\|\bar{\mathbf{v}}^k - \nabla f(\bar{\mathbf{x}}^k)\|^2|\mathcal{F}^k\right] \\ &\leq f(\bar{\mathbf{x}}^k) - \left(\frac{\eta}{2} - 2\eta^2 L\right)\|\nabla f(\bar{\mathbf{x}}^k)\|^2 \\ &\quad + \left(\frac{3\eta L^2}{n} + \frac{4\eta^2 L^3}{n}\right)\|\mathbf{x}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2 + \left(\frac{2\eta L^2}{n} + \frac{8\eta^2 L^3}{n}\right)\|\boldsymbol{\tau}^k - \bar{\mathbf{x}}^k_{\text{stack}}\|^2. \end{aligned} \tag{A.13}$$

Taking full expectation and using the condition $\eta L \leq 1/6$, we obtain the result in (A.12). □

Proof of Theorem 1. Summing the inequality in Proposition 4 from $s = 1$ to k and rearranging, we get:

$$\begin{aligned} \frac{\eta}{3}\sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right] &\leq \mathbb{E}[f(\bar{\mathbf{x}}^1)] - \mathbb{E}[f(\bar{\mathbf{x}}^{k+1})] + \frac{2L}{n}\sum_{s=1}^{k}\mathbb{E}\left[\|\mathbf{x}^s - \bar{\mathbf{x}}^s_{\text{stack}}\|^2\right] \\ &\quad + \frac{4L}{9n}\sum_{s=1}^{k}\mathbb{E}\left[\|\boldsymbol{\tau}^s - \bar{\mathbf{x}}^s_{\text{stack}}\|^2\right]. \end{aligned} \tag{A.14}$$

Using the bound from Proposition 3 on the summed error terms and the fact that $f(\bar{\mathbf{x}}^{k+1}) \geq f^*$, we can show that:

$$\left(\frac{\eta}{3} - \frac{C_{\text{const}}}{1-3\rho^2}\frac{10L}{9n}\right)\sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right] \leq f(\bar{\mathbf{x}}^1) - f^* + \frac{10LC_{\text{const}}R_0}{9n(1-3\rho^2)}. \tag{A.15}$$

Let C be the positive constant in the parentheses on the left-hand side. Dividing by kC gives:

$$\frac{1}{k}\sum_{s=1}^{k}\mathbb{E}\left[\|\nabla f(\bar{\mathbf{x}}^s)\|^2\right] \leq \frac{1}{kC}\left[f(\bar{\mathbf{x}}^1) - f^* + \frac{10LC_{\text{const}}R_0}{9n(1-3\rho^2)}\right] = O\left(\frac{1}{k}\right). \tag{A.16}$$

The convergence rates for the consensus and tracker errors follow by substituting this result back into Proposition 3. The convergence of $f(\bar{\mathbf{x}}^k)$ and $\lim_{k\to\infty}\|\nabla f(\bar{\mathbf{x}}^k)\| = 0$ can be established using Lemma 3. □

Proof of Corollary 1. From the result in (A.16), to find an ϵ-accurate stationary point as defined in (3.5), we need to find the smallest k such that

$$\frac{1}{kC}\left[f(\bar{\mathbf{x}}^1) - f^* + \text{const}\right] \leq \epsilon. \tag{A.17}$$

Solving for k gives $k \geq \frac{1}{\epsilon C}[\dots]$, which implies an iteration complexity of $O(1/\epsilon)$. The gradient and communication complexities follow by multiplying the iteration complexity by the per-iteration costs. Specifically, the gradient complexity is the iteration complexity multiplied by the average number of gradient computations per step, $\sum_{i=1}^{n}(q\pi_i m_i + 2)$. □