RW-LoRA: Communication-Efficient Decentralized LoRA Fine-Tuning via Random Walks

Xingran Chen, Rohit Bhagat, Ghadir Ayache, Rawad Bitar, Yanmin Gong, and Salim El Rouayheb

***Abstract*—Parameter-efficient fine-tuning methods such as LoRA have become a standard approach for adapting large foundation models. Adopting fine-tuning to distributed settings faces several challenges. Most existing distributed LoRA methods rely on centralized aggregation, and gossip-based LoRA requires repeated synchronization among multiple model copies. Both methods incur significant communication overhead and give rise to a bilinear mismatch due to multiple model aggregations. In this paper, we take a different perspective and propose a random-walk-based LoRA fine-tuning scheme. Instead of maintaining multiple model replicas, a single model token traverses the network and is updated sequentially using local fine-tuning objectives. This design eliminates the need for global synchronization, substantially reduces communication and computation costs, and avoids the bilinear mismatch. We provide rigorous convergence guarantees for non-convex objectives under standard assumptions. Through empirical results on multiple NLP tasks and graph topologies, we show that the proposed method achieves competitive task performance while requiring substantially less communication and computation than gossip-based LoRA.**

I. Introduction

Foundation models such as GPT-4 [1], LLaMA [2], and BERT [3] have transformed artificial intelligence, achieving strong performance on tasks ranging from translation to summarization [4], [5]. To achieve strong performance on domain-specific tasks, these models often need to be adapted using local or task-specific data. However, their scale, often ranging from 10^8 to 10^{11} parameters, makes full fine-tuning computationally expensive, communication-heavy, and prone to overfitting. Parameter-efficient fine-tuning (PEFT) methods, especially low-rank adaptation (LoRA), address this challenge by updating only a small number of additional parameters while keeping the backbone model frozen [6]. As a result, LoRA substantially reduces the computational and communication costs of fine-tuning.

The benefits of LoRA are especially important in distributed and resource-constrained applications, such as healthcare, edge personalization, and enterprise AI, where privacy, bandwidth, and data-sovereignty constraints often prevent raw data from being centralized. In such settings, fine-tuning must be performed through distributed collaboration across multiple data owners or edge nodes. Among them, federated LoRA methods [7]–[13] have become a predominant approach. However, these methods rely on centralized aggregation through a parameter server, which introduces communication and memory overhead and creates a single point of failure. To remove the need for central coordination, [14] recently proposed a gossip-based decentralized LoRA method. Nevertheless, gossip-based methods require nodes to exchange model updates with their neighbors repeatedly, which can still incur substantial communication overhead, especially on dense or bandwidth-limited networks.

More importantly, LoRA updates are of the form $W = BA$ where the model $W \in \mathbb{R}^{d_1 \times d_2}$ is factored into two low-rank matrices $A \in \mathbb{R}^{r \times d_2}$ and $B \in \mathbb{R}^{d_1 \times r}$ with $r \ll \min\{d_1, d_2\}$. In federated and gossip-based learning, the local models W_i of the nodes need to be averaged at every round. To avoid communicating W_i's, the nodes send their factor matrices A_i and B_i. Such methods face two challenges: (i) averaging the factors (as typically done in FL [7], [14]) introduces a mismatch, since $\sum_i B_i A_i \neq (\sum_i B_i)(\sum_i A_i)$; and (ii) computing $W_i = B_i A_i$ at the central server and then averaging the $W_i's$ result in a matrix that is not necessarily low-rank.

To address these limitations, we advocate token-based random-walk learning as a scalable alternative. A token representing the current model estimate moves randomly across the graph and is updated locally at each visited node [15]–[17]. Random-walk learning differs from consensus- and gossip-based methods in a fundamental way: it propagates a single mobile model rather than synchronizing many model copies. Each iteration requires only one node-to-node transfer, with communication on the order of the model size and independent of the network size. Propagating a single model avoids the mismatch of factor-wise aggregation and preserves the low-rank structure throughout training.

Our contributions are summarized as follows:

(i) We propose, to the best of our knowledge, the first random-walk-based LoRA fine-tuning algorithm. This design enables decentralized fine-tuning without a parameter server or synchronous neighbor-wise aggregation. We also provide rigorous convergence guarantees for non-convex objectives (Theorem 1).

(ii) We conduct experiments on several GLUE benchmark tasks using complete and ring communication graphs. Compared with the gossip-based decentralized LoRA

Xingran Chen is with the Engineering Systems and Design Pillar, Singapore University of Technology and Design, Singapore 487372 (E-mail: xingranc@ieee.org).

Ghadir Ayache is with the LinkedIn, New York, NY 10118, USA (E-mail: ayache.ghad@gmail.com).

Rawad Bitar is with the Chair of Communications Engineering, Technical University of Munich, 80333 Munich, Germany (E-mail: rawad.bitar@tum.de).

Yanmin Gong is with School of Engineering Medicine and the Department of Computer Science, Texas A&M University, College Station, TX 77843, USA (E-mail: yanmin.gong@tamu.edu).

Rohit Bhagat and Salim El Rouayheb are with Department of Electrical and Computer Engineering, Rutgers University, Piscataway Township, NJ 08854, USA (E-mail: {rb1395, sye8}@scarletmail.rutgers.edu).

baseline [4], RW-LoRA maintains strong task performance while significantly reducing both communication and computational overhead (Fig. 1). Moreover, by avoiding factor-wise aggregation of the LoRA matrices, RW-LoRA circumvents the bilinear mismatch issue present in gossip-based decentralized LoRA (Tables I and II).

II. System Model

We consider a set of nodes collaborating to fine-tune a large model. Each node holds a local dataset that can be used for fine-tuning, and nodes are able to communicate only with their neighbors. We represent the communication network as a graph, where each vertex corresponds to a node and each edge indicates that the two corresponding nodes can communicate directly. A token carrying the model parameters is passed from node to node and moves according to a random walk on this graph.

Definition 1 (Communication topologies and random walks)**.** A communication topology is defined as a finite directed graph $\mathcal{G} \triangleq (\mathcal{V}, E)$ with the set of nodes $\mathcal{V} = [N]$ and the set of edges $E \subseteq \binom{\mathcal{V}}{2}$. A random walk $\{v_t\}_{t\geqslant 0} : \Omega \to \mathcal{V}^{\mathbb{Z}_+}$ on this graph can be defined by the common transition probability matrix $P : \mathcal{V} \to \mathcal{M}(\mathcal{V})$, where the probability of transition from node u to node v in one time step at time $t \in \mathbb{Z}_+$, is $P_{uv} \triangleq \Pr(v_{t+1} = v \,|\, v_t = u)$.

Remark 1. Without loss of generality, we assume that $P_{uv} > 0$ for all nodes v connected to node u in graph $\mathcal{G}$. Therefore, transition matrix P is irreducible iff graph $\mathcal{G}$ is connected.

The stationary distribution describes the long-run fraction of time that the random walk spends at each node. Let P be the transition matrix defined in Definition 1. A probability π is called a stationary distribution if $\pi = \pi P$.

Let π_0 denote the initial distribution, and let $\pi_t = P^t\pi_0$ denote the distribution at time step t. For $\epsilon > 0$, the mixing time $\tau_{\text{mix}}(\epsilon)$ of $\{v_t\}_{t\geqslant 0}$ is defined as [15, Definition 2]:

$$\tau_{\text{mix}}(\epsilon) = \inf\left\{t \geqslant 1 | \forall \pi_0, d_{\text{TV}}(P^t\pi_0, \pi) \leqslant \epsilon\right\},$$

where $d_{\text{TV}}(\cdot,\cdot)$ is the total-variance distance and π_0 is the initial distribution.

We consider the decentralized stochastic optimization

$$\min_{\mathbf{w}} f(\mathbf{w}) = \min_{\mathbf{w}} \mathbb{E}_{v\sim\pi}\left[f_v(\mathbf{w})\right], \tag{1}$$

where $\mathbf{w}$ denotes the set of parameters to be fine-tuned (details are provided in Section III-A and Section IV), π is a probability distribution over the node set $\mathcal{V}$. In our setting, f is the global fine-tuning objective, while f_v is the local fine-tuning objective associated with the data stored at node v. Let $\mathcal{D}_v$ denote the local data distribution specific to user v, then the local fine-tuning objective f_v is given in stochastic form:

$$f_v(\mathbf{w}) = \mathbb{E}_{\xi_v\sim\mathcal{D}_v}\left[F_v(\mathbf{w};\xi_v)\right]. \tag{2}$$

Let $\{v_t\}_{t\geqslant 0}$ be a random walk on $\mathcal{V}$ with stationary distribution π. At time t, the model token is located at node v_t and is updated using only the local objective at that node [15]–[17]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla f_{v_t}(\mathbf{w}_t), \tag{3}$$

where $\eta > 0$ is the stepsize. (3) provides only a general random-walk update rule for random-walk-based learning. Its adaptation to the LoRA setting is presented in Section IV.

III. Preliminaries on LoRA

A. Low-Rank Adaptation

Consider a pre-trained model Φ_0[1] and let $W_0 \in \mathbb{R}^{d_1\times d_2}$ denote the weight matrix of one layer. In full fine-tuning, one would learn an update matrix $\Delta W \in \mathbb{R}^{d_1\times d_2}$ and use $W = W_0 + \Delta W$. To increase the efficiency of fine-tuning, LoRA keeps W_0 fixed and restricts the update to a low-rank form [6] $\Delta W = BA$, where $A \in \mathbb{R}^{r\times d_2}$, $B \in \mathbb{R}^{d_1\times r}$, and $r \ll \min\{d_1, d_2\}$. Thus, the adapted weight matrix becomes

$$W = W_0 + BA. \tag{4}$$

During the fine-tuning phase, LoRA optimizes the factors (also called adapters) A and B instead of directly updating ΔW. This reduces the number of trainable parameters from d_1d_2 to $r(d_1+d_2)$ and hence reduces the required computation and memory resources [7]. Throughout the paper, we refer to r as the LoRA rank, which is typically chosen from $\{2, 4, 8, 16\}$.

B. Federated LoRA-based Fine-Tuning

LoRA has become a popular approach for adapting foundation models [6]. To enable efficient fine-tuning across distributed data sources, LoRA has recently been incorporated into federated learning. Representative methods include FLoRA [7], FedALT [8], FedLoRA [9], FRLoRA [10], FedSA-LoRA [11], FedEx-LoRA [12], and FedMomentum [13]. These methods mainly differ in how they aggregate or personalize LoRA adapters across clients. For example, FLoRA, FedALT, and FedLoRA address heterogeneous or personalized LoRA adapters through stacking, personalized components, or global-local knowledge exchange. Other methods, including FRLoRA, FedSA-LoRA, FedEx-LoRA, and FedMomentum, improve aggregation efficiency or stability through residual updates, selective factor sharing, exactness-preserving corrections, or momentum-based aggregation. Most existing federated LoRA methods still rely on centralized aggregation through a server. As a result, they inherit the communication and coordination bottlenecks of classical federated learning, which become increasingly limiting for large foundation models.

C. Token-based Random-Walk Learning

Decentralized learning bypasses the need of a central node coordinating the process. The closest to our work is the recent work of [14]. The authors studied fully decentralized, gossip-based LoRA fine-tuning algorithm. They established convergence guarantees for non-convex objectives under standard

[1] Φ_0 denotes the initial neural network architecture, which consists of multiple layers parameterized by a set of weight matrices.

assumptions. In contrast, our work studies a random-walk-based LoRA method that avoids repeated server aggregation and neighbor-wise synchronization. This design reduces both communication and computational overhead.

Different from consensus- and gossip-based methods, token-based random-walk learning offers a different approach: a variable or model token moves across the graph according to a random walk and is updated using the local objective at each visited node. In this sense, each communication step directly moves the model to the next update location, rather than synchronizing several local models [15].

The literature on random-walk-based learning is broad, and we provide only a brief overview here. Seminal work in this area includes [16], [18], [19], which laid the foundation for optimization methods driven by Markov chain sampling. There are two directions to design and analyze token algorithms. First, [18] studied incremental optimization and subdifferentials methods under Markov chain sampling. Second, [16], [19] developed improved convergence guarantees for stochastic gradient and mirror-descent methods under Markovian sampling. A common feature of these results is that the convergence rate depends on the mixing time τ_{mix} of the underlying Markov chain, often through terms such as $\mathcal{O}\left(\frac{\tau_{\mathrm{mix}}}{T}\right) + \mathcal{O}\left(\sqrt{\frac{\tau_{\mathrm{mix}}}{T}}\right)$. For a more complete account of random-walk-based learning and token algorithms, we refer the reader to [15]. Recently, due to the attraction that random-walk based learning received, additional properties are being investigated. This includes privacy of the nodes' data [20], [21] and robustness of the learning algorithm to probabilistic and malicious node failure [22]–[24].

In this work, we shall restrict our attention to the convergence properties of random-walk learning for LoRA-based PEFT.

IV. The RW-LoRA Algorithm

In this section, we present the RW-LoRA algorithm, summarized in Algorithm 1.

As discussed in Section III-A, we fine-tune the model by optimizing the low-rank factors A and B. Specifically, the optimization problem in (1) can be written as:

$$\min_{A,B} f(W) = \min_{A,B} \mathbb{E}_{v\sim\pi}\left[f_v(W)\right], \tag{5}$$

where $A \in \mathbb{R}^{r\times d_2}$, $B \in \mathbb{R}^{d_1\times r}$, $f_v : \mathbb{R}^{d_1\times d_2} \to \mathbb{R}$ is the local fine-tuning objective of node v, and π is a pre-determined target distribution.

For simplicity, we use ∇ to denote the gradient with respect to W, i.e., ∇_W, and use ∇_A and ∇_B to denote the gradients with respect to A and B, respectively. We define

$$\nabla \tilde{f}_v(W) \triangleq \nabla F_v(W;\xi_v). \tag{6}$$

Given a target sampling distribution π $\big($assuming $\pi_v > 0$ for all $v \in \mathcal{V}\big)$, the Metropolis-Hastings algorithm [16] provides a principled way to construct a RW on $\mathcal{G}$ with a transition matrix P such that π is the stationary distribution of the Markov chain defined by P.[2] Let $\{v_t\}_{t\geqslant 0}$ be a random walk on $\mathcal{V}$ that evolves according to the transition matrix P. As mentioned in Section II, (3) provides a general random-walk update rule for vector-valued parameters. Since LoRA fine-tuning optimizes matrix-valued parameters, we extend this update rule to the two LoRA matrices A and B: At time t, the model token is located at node v_t, and the local fine-tuning objective at that node is used to update the LoRA matrices [4], [6],

$$A^{(t+1)} = A^{(t)} - \eta\nabla_A \tilde{f}_{v_t}(W^{(t)}), \tag{7}$$

$$B^{(t+1)} = B^{(t)} - \eta\nabla_B \tilde{f}_{v_t}(W^{(t)}), \tag{8}$$

where the initial weights $A^{(0)} \in \mathbb{R}^{r\times d_2}$, $[A^{(0)}]_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0,\sigma^2)$, $B^{(0)} = \mathbf{0} \in \mathbb{R}^{d_1\times r}$, and $W^{(0)} = W_0 + B^{(0)}A^{(0)}$, as adopted in [4].

Algorithm 1 RW-LoRA Algorithm

Require: Graph $\mathcal{G} = (\mathcal{V}, E)$, transition matrix P, stepsize η, LoRA rank r, number of iterations T, pre-trained weight matrix $W_0 \in \mathbb{R}^{d_1\times d_2}$

Initialize $A^{(0)} \sim \mathcal{N}(0,\sigma^2)$ and $B^{(0)} = 0$
Initialize the random walk at node $v_0 \in \mathcal{V}$
for $t = 0, 1, \ldots, T-1$, at node v_t **do**
  Form the LoRA-adapted weight $W^{(t)} = W_0 + B^{(t)}A^{(t)}$.
  Update the LoRA matrices via (7) and (8).
  Sample the next node according to $v_{t+1} \sim P_{v_t,\cdot}$.
  Transfer the token $(A^{(t+1)}, B^{(t+1)})$ to node v_{t+1}.
end for
return $A^{(T)}, B^{(T)}$ and $W^{(T)} = W_0 + B^{(T)}A^{(T)}$

V. Theoretical Results

This section establishes the convergence of RW-LoRA in the non-convex setting under Assumptions 1–5. These assumptions are stated and discussed immediately after the theorem.

Theorem 1. *Let Assumptions 1 ∼ 5 hold. Let* $\epsilon < \frac{1}{2}$, $\pi_{\min} = \min_{v\in\mathcal{V}}\{\pi_v\}$, $\tau \geqslant \tau_{\mathrm{mix}}(\pi_{\min}\epsilon)$, $T \geqslant \tau^2$, $\eta \leqslant \frac{1}{4L\sqrt{T}}$, *we obtain*[3]*:*

$$\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\left[\|\nabla_A f(W^{(t)})\|^2\right] + \mathbb{E}\left[\|\nabla_B f(W^{(t)})\|^2\right]\right) \leqslant \widetilde{\mathcal{O}}\left(\epsilon^2 + \frac{L\tau}{\sqrt{T}} + \frac{L\tau}{T}\right). \tag{9}$$

First, Theorem 1 shows that the proposed RW-LoRA algorithm converges to a stationary point in terms of the following first-order stationarity measure [4]:

$$\mathbb{E}\left[\|\nabla_A f(W^{(t)})\|\right] + \mathbb{E}\left[\|\nabla_B f(W^{(t)})\|\right].$$

This criterion is natural for LoRA fine-tuning because the trainable variables are the low-rank factors A and B, rather than the full update matrix ΔW. Hence, stationarity should be

[2]The transition matrix P depends on both the target sampling distribution π and the graph $\mathcal{G}$. We re-parameterize it as $P_{\pi,\mathcal{G}} \in \mathbb{R}_+^{N\times N}$, and simply write P when there is no risk of confusion.

[3]Here, $\widetilde{\mathcal{O}}(\cdot)$ hides logarithmic factors in the relevant problem parameters. For example, $\widetilde{\mathcal{O}}(1/\sqrt{T})$ may represent terms such as $\mathcal{O}(\log T/\sqrt{T})$ or $\mathcal{O}(\log(cT)/\sqrt{T})$, depending on the context.

evaluated with respect to these actual optimization variables. The complete proofs are provided in [25, Appendix A].

Second, Theorem 1 indicates that the convergence rate depends on the mixing time τ of the underlying Markov chain. In particular, our bound has the form $\widetilde{\mathcal{O}}\left(\epsilon^2 + \frac{L\tau}{\sqrt{T}} + \frac{L\tau}{T}\right)$. This is broadly consistent with the rates obtained in standard random-walk learning for vector-valued objectives, where the dependence on the number of iterations typically appears as $\mathcal{O}\left(\frac{\tau_{\text{mix}}}{T}\right) + \mathcal{O}\left(\sqrt{\frac{\tau_{\text{mix}}}{T}}\right)$ [16], [18], [19]. There are, however, two differences. First, our result is stated in $\widetilde{\mathcal{O}}$ notation, whereas the standard vector-valued setting is often expressed in $\mathcal{O}$ notation. Second, the coefficient of the $1/\sqrt{T}$ term scales linearly with the mixing time τ in our LoRA setting, while it scales as $\sqrt{\tau_{\text{mix}}}$ in the standard vector-valued case. This difference arises from the low-rank matrix factorization in LoRA: the trainable variables are the matrices A and B, and the update $\Delta W = BA$ introduces additional coupling between the two factors.

Assumption 1 (Submultiplicative property)**.** *The matrix norm $\|\cdot\|$ used in this paper is submultiplicative, i.e., $\|BA\| \leqslant \|B\|\|A\|$ for any compatible matrices A and B.*

This assumption is mild and is satisfied by most commonly used matrix norms, such as the spectral norm, Frobenius norm, 1-norm, and ∞-norm.

Assumption 2 (L-smooth, Assumption 4.1 in [4])**.** *We assume that each local fine-tuning objective f_v is L-smooth, i.e., for all $W, W' \in \mathbb{R}^{d_1 \times d_2}$, we have*

$$\|\nabla f_v(W) - \nabla f_v(W')\| \leqslant L\|W - W'\|.$$

Assumption 3 (Bounded gradient, Assumption 4.2 in [4])**.** *We assume that the stochastic gradients are unbiased and that their expected squared norm remains uniformly bounded:*

$$\mathbb{E}_{\xi_v \sim \mathcal{D}_v}\left[\nabla \tilde{f}_v(W)\right] = \nabla f_v(W),$$
$$\mathbb{E}_{\xi_v \sim \mathcal{D}_v}\left[\|\nabla \tilde{f}_v(W)\|^2\right] \leqslant c^2,\ v \in \mathcal{V},$$

where ξ_v represents a randomly sampled subset of training data from v-th node.

Assumption 3 is standard in decentralized optimization [4], [7], [9], [11]. It helps bound the magnitude of the update steps, as in the analysis in Lemmas 7 and 11 of [4]. For notational simplicity, we omit the subscript $\xi_v \sim \mathcal{D}_v$ when it is clear from the context.

Assumption 4 (Bounded variance, Assumption 5.1 in [26])**.** *There exists σ^2 such that for all $v \in \mathcal{V}$ and all $W \in \mathbb{R}^{d_1 \times d_2}$, we have:*

$$\mathbb{E}\left[\|\nabla \tilde{f}_v(W) - \nabla f(W)\|^2\right] \leqslant \sigma^2.$$

In Assumption 4, the expectation $\mathbb{E}$ is taken over two sources of randomness: (i) the randomness in data sampling at node v, i.e., $\xi_v \sim \mathcal{D}_v$, and (ii) the randomness in the global objective, as defined in (5). It is used to control the stochastic noise arising from the random-walk sampling process [26, Lemmas C.1 & C.2].

Assumptions 3 and 4 are both necessary and play distinct roles in our analysis. For vector-valued optimization variables, one of these assumptions is often sufficient, since the other is not needed for the convergence proof. In our setting, however, the optimization variables are the matrix-valued LoRA factors A and B, with $\Delta W = BA$. Because A and B are updated separately, dropping either assumption would prevent us from completing the convergence proof.

Assumption 5 (Assumption 4.3 in [4])**.** *There exists constants $a > 0$ and $b > 0$ such that: $\|A^{(t)}\| \leqslant a$, $\|B^{(t)}\| \leqslant b$ for all $0 \leqslant t \leqslant T$.*

In Assumption 5, the constants a and b are chosen uniformly over t. These constants may depend on the matrix dimensions, namely r, d_1, and d_2.

VI. Experiments

In this section, we compare the proposed RW-LoRA algorithm with a gossip-based decentralized LoRA baseline [4]. The results show that RW-LoRA significantly reduces both communication and computational overhead while avoiding the bilinear mismatch issue.

A. Experiment Settings

1) Setup: We use RoBERTa-base with 125M parameters [27] as the backbone model for all tasks. Following [14], we set the LoRA rank to $r = 16$. We use the default AdamW optimizer with learning rate 10^{-3}. The random-walk fine-tuning process is run over 10 nodes. At each communication round, the active node performs 1 local training steps with batch size $b = 32$ before passing the model token to the next node. All results are averaged over 10 runs with fixed random seed 42.

2) Datasets: We evaluate the proposed RW-LoRA algorithm on five GLUE datasets, covering two types of tasks[4]:

(i) Sentence-pair classification: We fine-tune the base model on MRPC [28], QQP [29], QNLI [29], [30], and MNLI [29], [31]. These tasks require the model to determine semantic equivalence, textual entailment, or whether a sentence contains the answer to a given question.

(ii) Sentiment classification: We fine-tune the base model on SST-2 [32], where the task is to classify each sentence as positive or negative.

3) Baseline: Decentralized learning for LoRA remains largely unexplored. To the best of our knowledge, [14] is the closest existing work to our setting. We therefore use its gossip-based federated LoRA method as our baseline.

4) Topologies: In our decentralized framework, nodes communicate exclusively along the edges of a fixed communication graph that connects 10 nodes. We focus on two topologies: ring graphs and complete graphs. These two topologies represent two extremes in network connectivity [23], [33]. The target distribution π is chosen as the uniform distribution, and the

[4]For each GLUE dataset, the training data are partitioned *i.i.d.* across 10 nodes.

corresponding transition probability matrix is constructed using the Metropolis–Hastings rule [16].

B. Experimental Results

a) Task Performance: The task performance of random-walk-based LoRA is reported in Tables I & II. Table I reports results on the complete graph, while Table II reports results on the ring topology[5]. In both tables, *Final Score* denotes the model performance at the last training round, whereas *Best Score* denotes the highest performance achieved over all training rounds.

	MRPC	SST2	QNLI	MNLI	QQP
RW (Final Scores)	**89.98**	**94.15**	**91.09**	**84.66**	**88.84**
Gossip (Final Scores)	83.57	93.23	90.17	83.66	86.71
RW (Best Scores)	**91.49**	**94.50**	**92.02**	**86.20**	**88.92**
Gossip (Best Scores)	83.57	93.23	90.17	84.06	86.71

TABLE I: Accuracy under Complete Graph

	MRPC	SST2	QNLI	MNLI	QQP
RW (Final Scores)	**90.75**	92.43	90.96	85.18	**88.54**
Gossip (Final Scores)	88.19	**92.89**	**91.23**	**85.76**	88.48
RW (Best Scores)	**93.02**	**94.50**	**92.00**	**86.07**	**88.93**
Gossip (Best Scores)	89.30	93.58	91.76	85.76	88.58

TABLE II: Accuracy under Ring Graph

On the complete graph, the proposed RW-LoRA algorithm outperforms gossip-based LoRA algorithm in both final and best accuracy. On the ring topology, the RW-LoRA algorithm achieves comparable final accuracy and higher best accuracy than gossip-based LoRA algorithm. Overall, these results show that the RW-LoRA algorithm improves upon the task performance of gossip-based LoRA algorithm.

b) Communication and Computation Efficiency: Under gossip-based LoRA, each node exchanges updates with its neighbors in every communication round. Thus, the per-node communication cost is $O(\text{degree} \times \text{model\_size})$, here $0.59M$ for the ring graph and $5.9M$ for the complete graph, and the total network-wide cost per round is $O(\text{network\_size} \times \text{degree} \times \text{model\_size})$, here $5.9M$ and $59M$ for the ring and complete graph, respectively. In contrast, random-walk-based LoRA involves only one active transmission per round, from the current node to the next node on the walk. Its total communication cost per round is therefore $O(\text{model\_size})$, here $0.59M$ for both graphs. Since each active transmission is accompanied by a local update, the computation cost scales with the communication cost.

Fig. 1 show the results on the QNLI dataset. We evaluate all methods on five datasets, and the accuracy curves exhibit similar trends across them. Due to space constraints, we report the accuracy curves for QNLI only. In Figs. 1a and 1c, the x-axis denotes the communication overhead (measured by the total number of bits transmitted), while in Figs. 1b and 1d, the x-axis denotes the computation overhead (measured by the total number of local SGD updates). The results show that the RW-LoRA algorithm reaches comparable or better accuracy with communication and computation cost that are order of magnitudes lower than the gossip-based LoRA algorithm. We note that here we observe one order of magnitude difference since we restrict the experiments to graphs with 10 nodes. For larger graphs, the difference in communication overhead will scale with the number of nodes and their degrees. This shows that the RW-LoRA algorithm is more resource-efficient, especially when communication and local computation are the main bottlenecks.

[5]In both tables, the performance metric for MRPC is F1, whereas the metric for all other datasets is accuracy. For simplicity of presentation, we use the term "Accuracy" in the tables to refer to the corresponding task performance metric.

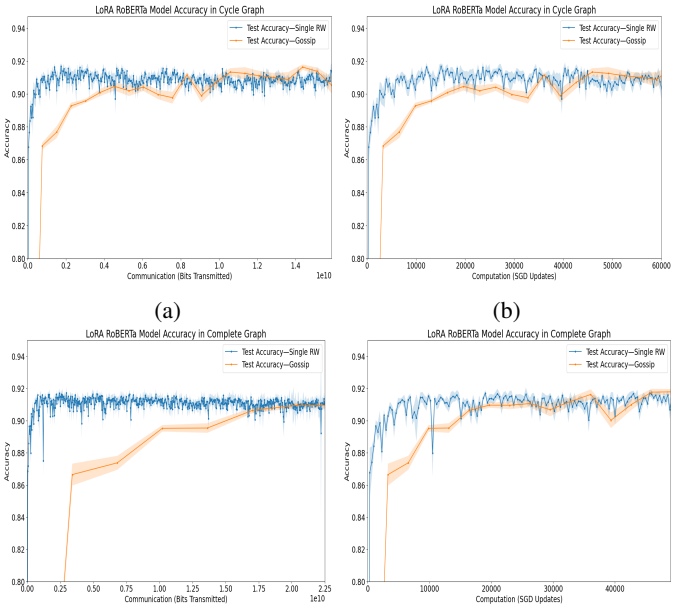

Fig. 1: (a) accuracy vs. communication overhead in a ring graph, (b) accuracy vs. computational overhead in a ring graph, (c) accuracy vs. communication overhead in a complete graph, and (d) accuracy vs. computational overhead in a complete graph.

VII. Conclusion

In this paper, we studied decentralized a random-walk-based LoRA framework. Unlike existing federated and gossip-based approaches that rely on multiple synchronized model copies, our method maintains a single mobile model that is updated sequentially across the network. We established convergence guarantees for non-convex objectives. Experimental results demonstrate that the proposed method achieves competitive performance while being substantially more resource-efficient than the gossip-based LoRA algorithm.

Future work will explore additional information-theoretic aspects of random-walk LoRA algorithms. In particular, it would be interesting to study how quantization and compression techniques can be combined with RW-LoRA to further reduce communication costs.

REFERENCES

[1] OpenAI, J. Achiam, S. Adler, and et.al, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[2] H. Touvron, T. Lavril, G. Izacard, and et.al, "Llama: Open and efficient foundation language models." *CoRR*, vol. abs/2302.13971, 2023. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr2302.html#abs-2302-13971

[3] J. Devlin, M. Chang, K. Lee, and et.al, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399

[4] S. Ghiasvand, M. Alizadeh, and R. Pedarsani, "Decentralized low-rank fine-tuning of large language models," arXiv:2501.15361, 2025.

[5] R. Bommasani, D. A. Hudson, E. Adeli, and et.al, "On the opportunities and risks of foundation models," *ArXiv*, vol. abs/2108.07258, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237091588

[6] E. J. Hu, Y. Shen, P. Wallis, and et. al, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[7] Z. Wang, Z. Shen, Y. He, and et.al, "FLoRA: federated fine-tuning large language models with heterogeneous low-rank adaptations," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 2024.

[8] J. Bian, L. Wang, L. Zhang, and et.al, "FedALT: Federated fine-tuning through adaptive local training with rest-of-world lora," *arXiv preprint arXiv:2503.11880*, 2025.

[9] L. Yi, H. Yu, G. Wang, and et.al, "Fedlora: Model-heterogeneous personalized federated learning with lora tuning," *ArXiv*, vol. abs/2310.13283, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264405713

[10] Y. Yan, C. Feng, W. Zuo, and et. al, "Federated residual low-rank adaption of large language models," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=e0rQRMUhs7

[11] P. Guo, S. Zeng, Y. Wang, and et. al, "Selective aggregation for low-rank adaptation in federated learning," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=iX3uESGdsO

[12] R. Singhal, K. Ponkshe, and P. Vepakomma, "FedEx-LoRA: Exact aggregation for federated and efficient fine-tuning of foundation models," in *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.

[13] P. Yan, Y. Hua, H. Wang, and et.al, "Fedmomentum: Preserving lora training momentum in federated fine-tuning," *ArXiv*, vol. abs/2603.08014, 2026.

[14] S. Ghiasvand, M. Alizadeh, and R. Pedarsani, "Decentralized low-rank fine-tuning of large language models," in *Proceedings of the 1st Workshop for Research on Agent Language Models (REALM 2025)*, Jul. 2025, pp. 334–345.

[15] M. Even, "Stochastic gradient descent under markovian sampling schemes," in *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[16] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2010.

[17] T. Sun, Y. Sun, and W. Yin, "On markov chain gradient descent," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 9918–9927.

[18] B. Johansson, M. Rabi, and M. Johansson, "A simple peer-to-peer algorithm for distributed optimization in sensor networks," *2007 46th IEEE Conference on Decision and Control*, pp. 4705–4710, 2007.

[19] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4064–4077, 2007.

[20] G. Ayache and S. E. Rouayheb, "Private weighted random walk stochastic gradient descent," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 452–463, 2021.

[21] M. Egger, S. Lage, R. Bitar, and et. al, "Source anonymity for private random walk decentralized learning," in *IEEE Information Theory Workshop (ITW)*. IEEE, 2025, pp. 1–6.

[22] M. Egger, G. Ayache, R. Bitar, and et. al, "Self-duplicating random walks for resilient decentralized learning on graphs," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2024, pp. 2960–2965.

[23] X. Chen, P. Parag, R. Bhagat, and et.al, "Self-creating random walks for decentralized learning under pac-man attacks," *arXiv preprint arXiv:2601.07674*, 2026.

[24] A. Khalesi and R. Bitar, "Fundamental limits of decentralized self-regulating random walks," *to be presented at the IEEE International Symposium on Information Theory*, 2026.

[25] X. Chen, R. Bhagat, G. Ayache, and et.al, "RW-LoRA: Communication-efficient decentralized lora fine-tuning via random walks," 2026. [Online]. Available: http://xingranchen.com/publications/LoRA.pdf

[26] M. Even, "Stochastic gradient descent under markovian sampling schemes," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23, 2023.

[27] Y. Liu, M. Ott, N. Goyal, and et. al, "RoBERTa: A robustly optimized bert pretraining approach," 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[28] B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[29] A. Wang, A. Singh, J. Michael, and et. al, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *BlackboxNLP@EMNLP*, 2018.

[30] P. Rajpurkar, J. J. Zhang, K. Lopyrev, and et.al, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[31] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *North American Chapter of the Association for Computational Linguistics*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3432876

[32] R. Socher, A. Perelygin, J. Wu, and et.al, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[33] X. Chen, P. Parag, R. Bhagat, and et. al, "Random walk learning and the pac-man attack," in *IEEE International Symposium on Information Theory (ISIT)*, 2026.

APPENDIX A
PROOF OF THEOREM 1

The proof consists of three parts. In the first part, we establish several useful lemmas. In the second part, we decompose the critical term $f(W^{(t+1)}) = f(B^{(t+1)}A^{(t+1)})$. In the third part, we apply a telescoping argument to obtain the desired result.

A. Useful lemmas

First of all, we provide the following useful lemmas.

Lemma 1. *The expected squared norms of $\nabla_A \tilde{f}_v(W)$ and $\nabla_B \tilde{f}_v(W)$ are bounded:*

$$\mathbb{E}\left[\|\nabla_A \tilde{f}_v(W)\|^2\right] \leqslant b^2c^2, \tag{10}$$

$$\mathbb{E}\left[\|\nabla_B \tilde{f}_v(W)\|^2\right] \leqslant a^2c^2. \tag{11}$$

Proof. Note that $W = BA$, according to (6), we have:

$$\begin{aligned}\nabla_A \tilde{f}_v(W) &= B^T \nabla \tilde{f}_v(W),\\ \nabla_B \tilde{f}_v(W) &= \nabla \tilde{f}_v(W) A^T.\end{aligned} \tag{12}$$

Taking norms on both sides of (12), and using the submultiplicative property in Assumption 1 together with Assumption 5, we obtain

$$\|\nabla_A \tilde{f}_v(W)\| \leqslant \|B\| \|\nabla \tilde{f}_v(W)\| \leqslant b\|\nabla \tilde{f}_v(W)\|, \tag{13}$$

$$\|\nabla_B \tilde{f}_v(W)\| \leqslant \|A\| \|\nabla \tilde{f}_v(W)\| \leqslant a\|\nabla \tilde{f}_v(W)\|. \tag{14}$$

According to Assumption 3, taking square and expectations on both sides of (13) and (14), we obtain:

$$\begin{aligned}\mathbb{E}\left[\|\nabla_A \tilde{f}_v(W)\|^2\right] &\leqslant b^2c^2,\\ \mathbb{E}\left[\|\nabla_B \tilde{f}_v(W)\|^2\right] &\leqslant a^2c^2.\end{aligned}$$

□

Lemma 2. *The expected squared differences $\mathbb{E}[\|A^{(t+1)} - A^{(t)}\|^2]$, $\mathbb{E}[\|B^{(t+1)} - B^{(t)}\|^2]$, and $\mathbb{E}[\|W^{(t+1)} - W^{(t)}\|^2]$ are upper bounded as follows:*

$$\mathbb{E}\left[\|A^{(t+1)} - A^{(t)}\|^2\right] \leqslant \eta^2 b^2 c^2, \tag{15}$$

$$\mathbb{E}\left[\|B^{(t+1)} - B^{(t)}\|^2\right] \leqslant \eta^2 a^2 c^2, \tag{16}$$

$$\mathbb{E}\left[\|W^{(t+1)} - W^{(t)}\|^2\right] \leqslant 3(\eta^2 b^4 c^2 + \eta^2 a^4 c^2 + \eta^4 a^2 b^2 c^4). \tag{17}$$

Proof. According to (7) and (8), we have:

$$\begin{aligned}\|A^{(t+1)} - A^{(t)}\|^2 &= \eta^2 \|\nabla_A \tilde{f}_{v_t}(W^{(t)})\|^2,\\ \|B^{(t+1)} - B^{(t)}\|^2 &= \eta^2 \|\nabla_B \tilde{f}_{v_t}(W^{(t)})\|^2.\end{aligned}$$

Taking expectations on both sides of above equalities, and based on (10) and (11), we have (15) and (16).

Next, we calculate $W^{(t+1)} - W^{(t)}$:

$$\begin{aligned}&W^{(t+1)} - W^{(t)} = B^{(t+1)}A^{(t+1)} - B^{(t)}A^{(t)}\\ &= \left(A^{(t)} - \eta \nabla_A \tilde{f}_{v_t}(W^{(t)})\right)\left(B^{(t)} - \eta \nabla_B \tilde{f}_{v_t}(W^{(t)})\right) - B^{(t)}A^{(t)}\\ &= -\eta \nabla_A \tilde{f}_{v_t}(W^{(t)})B^{(t)} - \eta A^{(t)} \nabla_B \tilde{f}_{v_t}(W^{(t)}) + \eta^2 \nabla_A \tilde{f}_{v_t}(W^{(t)}) \nabla_B \tilde{f}_{v_t}(W^{(t)}).\end{aligned}$$

Taking norms on both sides, by the triangle inequality, we obtain:

$$\begin{aligned}\|W^{(t+1)} - W^{(t)}\| \leqslant &\eta \|\nabla_A \tilde{f}_{v_t}(W^{(t)})B^{(t)}\| + \eta \|A^{(t)} \nabla_B \tilde{f}_{v_t}(W^{(t)})\|\\ &+\eta^2 \|\nabla_A \tilde{f}_{v_t}(W^{(t)}) \nabla_B \tilde{f}_{v_t}(W^{(t)})\|.\end{aligned}$$

Applying the submultiplicative property, we obtain

$$\begin{aligned}\|W^{(t+1)} - W^{(t)}\| \leqslant & \eta b \|\nabla_A \tilde{f}_{v_t}(W^{(t)})\| + \eta a \|\nabla_B \tilde{f}_{v_t}(W^{(t)})\| \\ & + \eta^2 \|\nabla_A \tilde{f}_{v_t}(W^{(t)})\| \|\nabla_B \tilde{f}_{v_t}(W^{(t)})\|.\end{aligned}$$

By Cauchy–Schwarz inequality, it follows that

$$\begin{aligned}\|W^{(t+1)} - W^{(t)}\|^2 \leqslant & 3\eta^2 b^2 \|\nabla_A \tilde{f}_{v_t}(W^{(t)})\|^2 + 3\eta^2 a^2 \|\nabla_B \tilde{f}_{v_t}(W^{(t)})\|^2 \\ & + 3\eta^4 \|\nabla_A \tilde{f}_{v_t}(W^{(t)})\|^2 \|\nabla_B \tilde{f}_{v_t}(W^{(t)})\|^2.\end{aligned}$$

Taking expectations on both sides of above equalities, based on (15) and (16), we obtain:

$$\mathbb{E}\left[\|W^{(t+1)} - W^{(t)}\|^2\right] \leqslant 3(\eta^2 b^4 c^2 + \eta^2 a^4 c^2 + \eta^4 a^2 b^2 c^4).$$

□

Lemma 3 (Lemma 4.10 in [4])**.** *Under Assumption 2, each local fine-tuning objective f_v is Lb^2-smooth with respect to A when B is fixed, and La^2-smooth with respect to B when A is fixed.*

Lemma 4. *Let $f(\cdot)$ be defined in* (5)*. Then, f is L-smooth.*

Proof. According to (5),

$$\nabla f(W) - \nabla f(W') = \mathbb{E}_{v\sim\pi}\left[\nabla f_v(W) - \nabla f_v(W')\right].$$

By the triangle inequality, we have:

$$\|\nabla f(W) - \nabla f(W')\| = \|\mathbb{E}_{v\sim\pi}\left[\nabla f_u(W) - \nabla f_v(W')\right]\| \leqslant \mathbb{E}_{v\sim\pi}\left\|\nabla f_v(W) - \nabla f_v(W')\right\|.$$

Based on Assumption 2, it follows that

$$\|\nabla f(W) - \nabla f(W')\|_F \leqslant \mathbb{E}_{v\sim\pi}\left[L\|W - W'\|\right] = L\|W - W'\|.$$

□

Lemma 5. *Let $f(\cdot)$ be defined in* (5)*. Then, f is Lb^2-smooth with respect to A when B is fixed, and La^2-smooth with respect to B when A is fixed.*

Proof. The proof directly follows from Lemma 3 and Lemma 4. □

Let π_0 denote the initial distribution, and let $\pi_t = P^t\pi_0$ denote the distribution at time step t. Let $\pi_{\min} = \min_{u\in\mathcal{V}} \pi_u$.

Lemma 6. *For $t \geqslant 0$ and if $v_t \sim \pi_t$ for $d_{TV}(\pi_t, \pi) \leqslant \frac{\pi_{\min}}{2}$, we have*

$$\mathbb{E}\left[\|\nabla \tilde{f}_{v_t}(W)\|^2\right] \leqslant 3\sigma^2 + 2\mathbb{E}\left[\|\nabla f(W)\|^2\right]$$

Proof. Since $d_{\mathrm{TV}}(\pi_t, \pi) \leqslant \frac{\pi_{\min}}{2}$, then for any $v \in \mathcal{V}$, we have

$$\Pr(v_t = v) \leqslant \pi_v + \pi_v/2 = \frac{3\pi_v}{2}.$$

By the triangle inequality, Cauchy–Schwarz inequality, and Assumption 4, it follows that

$$\begin{aligned}\mathbb{E}\left[\|\nabla \tilde{f}_{v_t}(W)\|^2\right] \leqslant & 2\mathbb{E}\left[\|\nabla \tilde{f}_{v_t}(W) - \nabla f(W)\|^2\right] + 2\mathbb{E}\left[\|\nabla f(W)\|^2\right] \\ \leqslant & 2\sum_{v\in\mathcal{V}} \Pr(v_t = v)\sigma^2 + 2\mathbb{E}\left[\|\nabla f(W)\|^2\right] \\ = & 3\sigma^2 + 2\mathbb{E}\left[\|\nabla f(W)\|^2\right].\end{aligned}$$

□

Lemma 7. *The expected squared norms of $\nabla f(W)$, $\nabla_A f(W)$ and $\nabla_B f(W)$ are bounded:*

$$\mathbb{E}\left[\|\nabla f(W)\|^2\right] \leqslant 2\sigma^2 + 2c^2, \tag{18}$$

$$\mathbb{E}\left[\|\nabla_A f(W)\|^2\right] \leqslant 2b^2(\sigma^2 + c^2), \tag{19}$$

$$\mathbb{E}\left[\|\nabla_B f(W)\|^2\right] \leqslant 2a^2(\sigma^2 + c^2). \tag{20}$$

Proof. By the triangle inequality and Cauchy–Schwarz inequality, we have

$$\mathbb{E}\left[\|\nabla f(W)\|^2\right] \leqslant 2\mathbb{E}\left[\|\nabla \tilde{f}_v(W) - \nabla f(W)\|^2\right] + 2\mathbb{E}\left[\|\nabla \tilde{f}_v(W)\|^2\right]$$

According to Assumption 3 and Assumption 4, we have

$$\mathbb{E}\left[\|\nabla f(W)\|^2\right] \leqslant 2\sigma^2 + 2c^2.$$

Note that $W = BA$, we have:

$$\nabla_A f(W) = B^T \nabla f(W),\ \nabla_B f(W) = \nabla f(W) A^T. \tag{21}$$

Applying the submultiplicative property and taking expectations on (21), we obtain:

$$\mathbb{E}\left[\|\nabla_A f(W)\|^2\right] \leqslant 2b^2(\sigma^2 + c^2)$$
$$\mathbb{E}\left[\|\nabla_B f(W)\|^2\right] \leqslant 2a^2(\sigma^2 + c^2).$$

B. Decomposition of $\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right]$

Since f is smooth, by the Descent Lemma[6], we have:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right] \leqslant& \mathbb{E}\left[f(B^{(t+1)}A^{(t)})\right] + \mathbb{E}\left\langle \nabla_A f(B^{(t+1)}A^{(t)}), A^{(t+1)} - A^{(t)} \right\rangle \\ &+ \frac{Lb^2}{2}\mathbb{E}\left[\|A^{(t+1)} - A^{(t)}\|^2\right].\end{aligned} \tag{22}$$

In (22), the expectation $\mathbb{E}$ is taken over the target sampling π and the randomness from the position of the RW at time t, i.e., v_t. Again, by the Descent Lemma, we further have:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t)})\right] \leqslant& \mathbb{E}\left[f(B^{(t)}A^{(t)})\right] + \mathbb{E}\left\langle \nabla_B f(B^{(t)}A^{(t)}), B^{(t+1)} - B^{(t)} \right\rangle \\ &+ \frac{La^2}{2}\mathbb{E}\left[\|B^{(t+1)} - B^{(t)}\|^2\right].\end{aligned} \tag{23}$$

Combine (22) and (23), we obtain:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right] \leqslant& \mathbb{E}\left[f(B^{(t)}A^{(t)})\right] + \mathbb{E}\left\langle \nabla_A f(B^{(t+1)}A^{(t)}), A^{(t+1)} - A^{(t)} \right\rangle \\ &+ \mathbb{E}\left\langle \nabla_B f(B^{(t)}A^{(t)}), B^{(t+1)} - B^{(t)} \right\rangle + \frac{Lb^2}{2}\mathbb{E}\left[\|A^{(t+1)} - A^{(t)}\|^2\right] \\ &+ \frac{La^2}{2}\mathbb{E}\left[\|B^{(t+1)} - B^{(t)}\|^2\right].\end{aligned} \tag{24}$$

By Cauchy–Schwarz inequality, we have

$$\begin{aligned}\mathbb{E}\left\langle \nabla_A f(B^{(t+1)}A^{(t)}), A^{(t+1)} - A^{(t)} \right\rangle \leqslant& \frac{1}{2}\mathbb{E}\left[\|\nabla_A f(B^{(t+1)}A^{(t)})\|^2\right] \\ &+ \frac{1}{2}\mathbb{E}\left[\|A^{(t+1)} - A^{(t)}\|^2\right].\end{aligned}$$

From (15) in Lemma 2 and (19) in Lemma 7,

$$\mathbb{E}\left\langle \nabla_A f(B^{(t+1)}A^{(t)}), A^{(t+1)} - A^{(t)} \right\rangle \leqslant b^2(\sigma^2 + c^2) + \eta^2 \frac{b^2 c^2}{2}. \tag{25}$$

According to the update rule (8), we have

$$\begin{aligned}\mathbb{E}\left\langle \nabla_B f(B^{(t)}A^{(t)}), B^{(t+1)} - B^{(t)} \right\rangle &= \mathbb{E}\left\langle \nabla_B f(B^{(t)}A^{(t)}), -\eta \nabla_B \tilde{f}_{v_t}(W^{(t)}) \right\rangle \\ &= -\eta \mathbb{E}\left\langle \nabla_B f(B^{(t)}A^{(t)}), \nabla_B \tilde{f}_{v_t}(W^{(t)}) \right\rangle \\ &= \mathbb{E}\left[-\eta \left\langle \nabla_B f(W^{(t)}), \nabla_B \tilde{f}_{v_t}(W^{(t)}) \right\rangle\right].\end{aligned} \tag{26}$$

[6]The descent lemma is a standard result for L-smooth differentiable functions. If f is L-smooth, then for any x, y,

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2.$$

.

Let $\epsilon < \frac{1}{2}$, $\tau \geqslant \tau_{\text{mix}}(\pi_{\min}\epsilon)$, and $t \geqslant \tau$, based on [26, Appendix C.1], we have:

$$\begin{aligned}\mathbb{E}\left[-\eta\left\langle\nabla_B\tilde{f}_{v_t}(W^{(t)}),\nabla_B f(W^{(t)})\right\rangle\right] &= \mathbb{E}\left[-\eta\left\langle\nabla_B\tilde{f}_{v_t}(W^{(t-\tau)}),\nabla_B f(W^{(t-\tau)})\right\rangle\right]\\ &+\mathbb{E}\left[-\eta\left\langle\nabla_B\tilde{f}_{v_t}(W^{(t)}),\nabla_B f(W^{(t)})-\nabla_B f(W^{(t-\tau)})\right\rangle\right]\\ &+\mathbb{E}\left[-\eta\left\langle\nabla_B\tilde{f}(W^{(t)})-\nabla_B\tilde{f}(W^{(t-\tau)}),\nabla_B f(W^{(t-\tau)})\right\rangle\right]\\ &\triangleq E_1+E_2+E_3.\end{aligned}$$

By the same analysis as in [26, Appendix C.1], we obtain the following three inequalities: (27), (28), and (29). First,

$$E_1 \leqslant -\frac{\eta}{4}\mathbb{E}\left[\|\nabla_B f(W^{(t-\tau)})\|^2\right]+\eta\epsilon^2\sigma^2. \tag{27}$$

By utilizing Assumption 2, we have:

$$E_2 \leqslant \frac{\eta^2 L}{2}\tau\mathbb{E}\left[\|\nabla_B\tilde{f}_{v_t}(W^{(t)})\|^2\right]+\frac{\eta^2 L}{2}\sum_{s=t-\tau}^{t-1}\mathbb{E}\left[\|\nabla_B\tilde{f}_{v_s}(W^{(s)})\|^2\right]. \tag{28}$$

Similarly, we have:

$$E_3 \leqslant \frac{\eta^2 L}{2}\tau\mathbb{E}\left[\|\nabla_B f(W^{(t-\tau)})\|^2\right]+\frac{\eta^2 L}{2}\sum_{s=t-\tau}^{t-1}\mathbb{E}\left[\|\nabla_B\tilde{f}_{v_s}(W^{(s)})\|^2\right] \tag{29}$$

Based on Lemma 2, Lemma 7, and (25) $\sim$ (29), we have:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right] \leqslant &\mathbb{E}\left[f(B^{(t)}A^{(t)})\right]+\eta\epsilon^2\sigma^2+b^2(\sigma^2+c^2)\\ &+\eta^2\left(\frac{3L}{2}\tau a^2c^2+\frac{c^2L}{2}(a^4+b^4)+\frac{b^2c^2}{2}\right)\\ &-\left(\frac{\eta}{4}-\frac{\eta^2L}{2}\tau\right)\mathbb{E}\left[\|\nabla_B f(W^{(t-\tau)})\|^2\right].\end{aligned} \tag{30}$$

In (22) and (23), we decompose $f(B^{(t+1)}A^{(t+1)})$ using the intermediate terms $f(B^{(t)}A^{(t+1)})$ and $f(B^{(t)}A^{(t)})$. By a similar analysis, we obtain:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right] \leqslant &\mathbb{E}\left[f(B^{(t)}A^{(t)})\right]+\eta\epsilon^2\sigma^2+a^2(\sigma^2+c^2)\\ &+\eta^2\left(\frac{3L}{2}\tau b^2c^2+\frac{c^2L}{2}(a^4+b^4)+\frac{a^2c^2}{2}\right)\\ &-\left(\frac{\eta}{4}-\frac{\eta^2L}{2}\tau\right)\mathbb{E}\left[\|\nabla_A f(W^{(t-\tau)})\|^2\right].\end{aligned} \tag{31}$$

Summing over (30) and (31), we derive:

$$\begin{aligned}\mathbb{E}\left[f(B^{(t+1)}A^{(t+1)})\right] \leqslant &\mathbb{E}\left[f(B^{(t)}A^{(t)})\right]+\eta\epsilon^2\sigma^2+\frac{a^2+b^2}{2}(\sigma^2+c^2)\\ &+\eta^2\left(\frac{3(a^2+b^2)L}{4}\tau c^2+\frac{c^2L}{2}(a^4+b^4)+\frac{c^2(a^2+b^2)}{4}\right)\\ &-\left(\frac{\eta}{4}-\frac{\eta^2L}{2}\tau\right)\left(\mathbb{E}\left[\|\nabla_A f(W^{(t-\tau)})\|^2\right]+\mathbb{E}\left[\|\nabla_B f(W^{(t-\tau)})\|^2\right]\right).\end{aligned} \tag{32}$$

C. Telescoping

Let $T \geqslant \tau^2$, $\eta \leqslant \frac{1}{4L\sqrt{T}}$, we have

$$\frac{\eta}{8} \leqslant \frac{\eta}{4}-\frac{\eta^2L}{2}\tau.$$

Then, summing for $\tau \leqslant t < T+\tau$, by telescoping (32), we obtain:

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbb{E}\left[\|\nabla_A f(W^{(t)})\|^2\right]+\mathbb{E}\left[\|\nabla_B f(W^{(t)})\|^2\right]\right)\\ &\leqslant \epsilon^2\sigma^2+\eta\Gamma+\frac{8\mathbb{E}\left[f(W^{(\tau)})\right]+4(a^2+b^2)(\sigma^2+c^2)}{\eta T},\end{aligned} \tag{33}$$

where

$$\Gamma = 6(a^2+b^2)L\tau c^2 + 4c^2L(a^4+b^4) + 2c^2(a^2+b^2).$$

Now, we upper bound $\mathbb{E}\left[f(W^{(\tau)})\right]$. According to Lemma 4, f is L-smooth. By the Descent Lemma, we have:

$$f(W^{(\tau)}) - f(W^{(\tau-1)}) \leqslant \langle \nabla f(W^{(\tau-1)}), W^{(\tau)} - W^{(\tau-1)} \rangle + \frac{L}{2}\|W^{(\tau)} - W^{(\tau-1)}\|^2.$$

By Cauchy–Schwarz inequality,

$$f(W^{(\tau)}) - f(W^{(\tau-1)}) \leqslant \left(\|\nabla f(W^{(\tau-1)})\| + \frac{L}{2}\right)\|W^{(\tau)} - W^{(\tau-1)}\|^2,$$

which implies

$$\mathbb{E}\left[f(W^{(\tau)})\right] - \mathbb{E}\left[f(W^{(\tau-1)})\right] \leqslant \left(\sqrt{\mathbb{E}\left[\|\nabla f(W^{(\tau-1)})\|^2\right]} + \frac{L}{2}\right)\|W^{(\tau)} - W^{(\tau-1)}\|^2.$$

Based on (18) in Lemma 7 and (17) in Lemma 2, we have:

$$\mathbb{E}\left[f(W^{(\tau)})\right] - \mathbb{E}\left[f(W^{(\tau-1)})\right] \leqslant 3\left(\sqrt{2\sigma^2 + 2c^2} + \frac{L}{2}\right)(\eta^2 b^4 c^2 + \eta^2 a^4 c^2 + \eta^4 a^2 b^2 c^4).$$

By applying the inequality above repeatedly, we obtain:

$$\mathbb{E}\left[f(W^{(\tau)})\right] \leqslant \mathbb{E}\left[f(W^{(0)})\right] + 3\tau\left(\sqrt{2\sigma^2 + 2c^2} + \frac{L}{2}\right)(\eta^2 b^4 c^2 + \eta^2 a^4 c^2 + \eta^4 a^2 b^2 c^4),$$

which implies $\mathbb{E}\left[f(W^{(\tau)})\right]$ is bounded.

Finally, let $\epsilon < \frac{1}{2}$, $\tau \geqslant \tau_{\text{mix}}(\pi_{\min}\epsilon)$, $T \geqslant \tau^2$, $\eta \leqslant \frac{1}{4L\sqrt{T}}$, we obtain:

$$\frac{1}{T}\sum_{t=1}^{T}\left(\mathbb{E}\left[\|\nabla_A f(W^{(t)})\|^2\right] + \mathbb{E}\left[\|\nabla_B f(W^{(t)})\|^2\right]\right) \leqslant \tilde{\mathcal{O}}\left(\epsilon^2 + \frac{L\tau}{\sqrt{T}} + \frac{L\tau}{T}\right).$$

□