Secure Distributed Optimization Under Gradient Attacks

Shuhua Yu, *Graduate Student Member, IEEE*, and Soummya Kar, *Fellow, IEEE*

***Abstract*—In this article, we study secure distributed optimization against arbitrary gradient attacks in multi-agent networks. In distributed optimization, there is no central server to coordinate local updates, and each agent can only communicate with its neighbors on a predefined network. We consider the scenario where out of n networked agents, a fixed but unknown fraction ρ of the agents are under arbitrary gradient attacks in that their stochastic gradient oracles return arbitrary information to derail the optimization process, and the goal is to minimize the sum of local objective functions on unattacked agents. We propose a distributed stochastic gradient method that combines local variance reduction and clipping (CLIP-VRG). We show that, in a connected network, when the unattacked local objective functions are convex and smooth, share a common minimizer, and their sum is strongly convex, CLIP-VRG leads to almost sure convergence of the iterates to the exact sum cost minimizer at all agents. We quantify a tight upper bound on the fraction ρ of attacked agents in terms of problem parameters such as the condition number of the associated sum cost that guarantee exact convergence of CLIP-VRG, and characterize its asymptotic convergence rate. Finally, we empirically demonstrate the effectiveness of the proposed method under gradient attacks on both synthetic and real-world image classification datasets.**

***Index Terms*—Distributed optimization, multi-agent networks, security, resilience, gradient descent, variance reduction.**

I. Introduction

In THIS article, we study the problem of secure distributed optimization in peer-to-peer multi-agent networks under arbitrary gradient attacks. In distributed optimization over n networked agents, each agent $i \in [n]$ holds a local objective function f_i, has access to stochastic gradients of its local f_i via a local and private stochastic gradient oracle, and may only communicate with its direct neighbors defined by an inter-agent communication graph to cooperatively minimize the aggregated objective function $\sum_{i\in[n]} f_i$. The sum-cost minimization and its stochastic variants as described above have emerged as natural abstractions of performing various distributed signal processing and machine learning tasks and seen extensive research over the past decade with the primary focus of building distributed stochastic gradient like procedures based on consensus [1], [2] or diffusion processes [3] with proven convergence or learning guarantees [4].

This article studies the adversarial setting where a fixed but unknown fraction ρ of agents are under gradient attacks in that their stochastic gradient oracles return arbitrary adversarial information when queried during algorithm execution. In distributed network based settings such a scenario arises in which the local cost functions (often reflecting the local data at the agents) are manipulated by an adversary, corresponding to the practical class of data injection attacks.[1] Denoting by $\mathcal{A}$ the set of agents, unknown apriori, whose gradients are potentially attacked and by $\mathcal{N}$ the non-attacked agents such that $|\mathcal{A}| + |\mathcal{N}| = n$, in the adversarial scenario, instead of minimizing the global aggregate $\sum_{i\in[n]} f_i$, we aim to solve

$$\text{minimize}_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} f_i(\mathbf{x}). \tag{1}$$

When $\mathcal{N} = [n]$, problem (1) reduces to the classical distributed optimization formulation (in non-adversarial environments) as discussed above.

Many machine learning and statistical inference tasks are currently being implemented via decentralized computation paradigms such as Federated Learning [5], [6] instead of the classical cloud-based paradigm, to deal with scalability, robustness and privacy issues. As server-worker paradigms, where there is a central server to coordinate local model updates, are prone to single point of failures and bottlenecks, *fully* distributed paradigms that distribute the computation task among multiple entities have gained increasing attention [4], [7], [8]. However, distributed data and communication pose additional challenges such as data integrity concerns. As many distributed optimization methods are based on stochastic gradient computation, gradient attacks induced by malicious data injection is a serious security concern that needs to be addressed. For example, the adversary can inject malicious data points to some participating agents in decentralized machine learning training [9], or corrupt sensor measurements in statistical inference over sensor networks [10], [11]. In these settings, undefended distributed algorithms can be arbitrarily misled by such gradient attacks, thus motivating the current work.

Manuscript received 27 October 2022; revised 1 March 2023; accepted 2 May 2023. Date of publication 19 May 2023; date of current version 26 May 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ketan Rajawat. This work was supported by the Office of Naval Research under Grant N00014-21-1-2547. *(Corresponding author: Shuhua Yu.)*

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: shuhuay@andrew.cmu.edu; soummyak@andrew.cmu.edu).

Digital Object Identifier 10.1109/TSP.2023.3277211

[1] In what follows, we will refer to such data injection attacks as gradient attacks as the constructed optimization procedures are based on first order gradient optimization. Although for abstraction purposes we model the attacks as affecting the gradients, it follows from a straightforward inspection of our algorithms that the proposed constructions carry over to other types of attacks that directly manipulate the data or cost functions at the agents.

Our work is closely related to Byzantine-robust distributed optimization. Byzantine attack [12] is the most difficult threat model considered in the distributed optimization literature in that Byzantine agents can deviate from the prescribed algorithm and send arbitrary adversarial messages to its neighbors. In contrast, the threat model considered in this article assumes that those attacked agents still preserve normal computation and communication capabilities, i.e. still follow the prescribed algorithmic procedures. Thus, our threat model is weaker and subsumed by the Byzantine attack. While data attacks are subsumed by the Byzantine model, it is important to note that certain types of Byzantine behavior may be addressed or detected by the use of appropriate cryptographic protocols (e.g. digital signatures) [13]), whereas, data (gradient) attacks, even via passive honest agents as considered in this work, may be hard to detect due to the heterogeneous nature of agent data and noise, thus making standard cryptographic solutions inadequate. This is why the data injection attack model, albeit weaker than the most general Byzantine model, is of interest in its own right. To cope with Byzantine agents, different approaches to aggregate information from neighbors [14] have been proposed. The works [15], [16], [17], [18] use trimmed average of model updates from neighbors, but this approach requires that the majority of non-Byzantine agents' neighbors are also non-Byzantine, while in our threat model we do not have such strict requirements on the distribution of attacked agents over networks. If data are independent and identically distributed (i.i.d.), [19] proposes to use local data points to evaluate the models communicated from neighbors and only trust those with good performance on local data. Algorithmically, the most related method to the approach proposed in this article, designated as CLIP-VRG, is SCCLIP proposed in [20] that combines local momentum and self-centered clipping on differences between local and incoming models. In contrast, the proposed CLIP-VRG approach uses decaying stepsizes to achieve local variance reduction, and the clipping operator in CLIP-VRG is applied on a suitably constructed gradient estimators instead of model differences. More practically, CLIP-VRG uses predefined clipping sequences, whereas, SCCLIP relies on information that may not be accessible in a distributed environment for obtaining its clipping thresholds. In addition, [21] proposes a TV-regularized approximation of the Byzantine-free optimization problem along with a subgradient method that converges to a neighborhood of the optimal solution. Exact minimum is also achievable if some redundancy condition and a complete communication graph are assumed [22]. Our "common minimizer" assumption is mildly stronger than the redundancy assumption in [22], but our algorithm applies to general connected network topology and is based on stochastic gradients while [22] requires exact (full) gradient. Note that there exists a trade-off between redundancy and the robustness to adversarial attacks. In the Byzantine threat model, the adversarial agents have unrestricted capabilities, so more stringent forms of network connectivity conditions (as well as a limit on the number of Byzantine agents) need to be enforced to ensure that the non-Byzantine agents retain access to useful information [14], [17], [22]. While in our setup, due to a more restricted attack model, the robustness requirement boils down to having simply a connected network and a limit on the fraction of agents attacked, both of which can be readily checked.

Our work is also related to the resilient distributed parameter estimation in terms of threat model considered. The papers [10], [23] consider sensor attacks in distributed parameter estimation that can arbitrarily manipulate sensor measurements but assume that the computation and communication capabilities of sensors remain intact, which is similar to the gradient attack scenario studied in this work. To counter sensor measurement attacks, a distributed estimator based on saturated local update is proposed in [10], [23] which is the first clipping based method to achieve resilience in distributed inference to the best of our knowledge. From an optimization perspective, this method is indeed a gossip-type distributed stochastic gradient method with local gradient clipping. In contrast, CLIP-VRG applies to a wider range of distributed optimization models compared to the de facto least-squares model considered in [11], [23]. From an algorithmic perspective, CLIP-VRG uses a constant consensus mixing matrix while [11], [23] employ a sequence of time-varying mixing matrices following the *consensus+innovations* framework [24]; additionally, CLIP-VRG further involves gradient estimators to achieve variance reduction locally, which is discussed in more detail in Remark 5. The article [25] studies robust distributed estimation over networks when measurements are corrupted by impulsive noise. In [25], the impulsive noise corrupt measurements of *all* networked agents, in contrast to *a bounded fraction* of agents in this article, but the considered impulsive noise is assumed to be zero-mean, symmetric and stationary with bounded variance, whereas, the gradient attack in our work is arbitrary. We refer the reader to [11] and references therein for a broader survey on more countermeasures to achieve resilient distributed inference in multi-agent networks.

Other Related Works: Distributed optimization has been extensively studied with various algorithmic frameworks and constructions [26], including distributed (sub)gradient descent methods [1], [27], gradient tracking based methods [7], [28], [29], acceleration schemes [30], variance reduction schemes [31], primal-dual methods [32], [33], [34], to name a few. Distributed optimization has also been studied taking into account different communication topologies [35], [36], compressed communication [37], data heterogeneity [38] and data privacy [39]. Although adversarial robustness of distributed optimization is relatively less studied, in server-worker type setups with a central trustworthy server, several approaches to achieve Byzantine robustness have been proposed. In these approaches the central server employs robust gradient aggregators such as the median [40], [41], [42], [43], geometric median [44], concentration filtering [45], [46], signSGD [47], [48], gradient clipping [49], and worker momentum [50]. When the central server also has access to the training data, the server can score the incoming gradients and abandon those abnormal ones [51], [52], and may also reach exact minimum by exploiting redundancy [53]. In the case that the probability of an agent being Byzantine or trustworthy follows a two-state Markov Chain, [54] proposes a method with temporal and spatial robust aggregation, and gradient normalization. In addition, in the decentralized optimization setting with a trustworthy server, Byzantine robustness combined with other challenging constraints have also been studied, including privacy [55], asynchronous decentralized computing [56], and in particular data heterogeneity. Given distributed heterogeneous data, and that the attackers may take advantage of the variance of good workers over time [57] making it hard to distinguish Byzantine workers, methods such as bucketing [58], RSA [48] and concentration filtering [46] have been developed to counter this issue.

Main Contributions: The main contributions of this article are as follows: (1) We consider an arbitrary gradient attack model that is relatively unexplored in the context of distributed optimization; (2) We develop a distributed stochastic gradient based method, i.e., CLIP-VRG, that combines local variance reduced gradient estimation and clipping, and analytically and empirically illustrate its robust performance against gradient attacks; (3) For convex and smooth unattacked local objective functions that share a common minimizer, if the sum of unattacked objective functions is strongly convex, we prove that CLIP-VRG asymptotically and almost surely converges to the exact minimizer as long as the proportion ρ of attacked agents satisfies $\rho < 1/(1+\kappa)$, where κ is the condition number of the aggregated unattacked objective function.

Notations: We use $[n] = \{1, 2, \ldots, n\}$ to denote the set of all network agents, and $|\cdot|$ to denote cardinality for an argument set such as $\mathcal{N}$. We use $\|\cdot\|$ to denote Euclidean norm for vectors and $\|\cdot\|_2$ for spectral norm of matrices, respectively. We use $\mathrm{diag}(\cdot)$ to denote the diagonal matrix whose diagonal entries are components of the argument vector. We use $\mathbf{1}_p$ to denote the column vector of ones of length p, and bold lower case and upper case letters to denote vectors and matrices, respectively. Equalities or inequalities that involve random variables hold true almost surely. Random variables are often indexed by a superscript or subscript ω to indicate their sample path behavior.

Organizations: In Section II, we formalize the problem assumptions and discuss their implications. In Section III, we develop CLIP-VRG and present our main theoretical results. Section IV details the implementations and empirical performance of CLIP-VRG for a regularized logistic regression model on both synthetic and image classification datasets. The proofs of the main results are provided in Section V, whereas, Section VI concludes the article.

II. Problem Setup

Referring to the scenario in (1), recall, by $\mathcal{A}$ we denote the set of agents whose stochastic gradient oracles are arbitrarily manipulated by some adversary. For simplicity, let $a = |\mathcal{N}|$, $b = |\mathcal{A}|$, $a + b = n$, and the fraction of attacked agents $\rho = b/n$. Agents aim to minimize f as defined in (1) by local computations and communications with neighbors, i.e., in a distributed manner, as is common in the distributed consensus or gossip based computing literature [1], [2], [3]. We consider iterative processes that at each agent $i \in [n]$ generate a sequence of state vectors $\{\mathbf{x}_i^t : t \geq 0\}$, where $\mathbf{x}_i^0$ is algorithm initialization treated as constant. At each iteration t, each agent i calls a local stochastic gradient oracle that returns $\mathbf{m}_i(\mathbf{x}_i^t)$ (shortened as $\mathbf{m}_i^t$) at query $\mathbf{x}_i^t$. For $i \in \mathcal{N}$, we can write

$$\mathbf{m}_i^t = \nabla f_i(\mathbf{x}_i^t) + \boldsymbol{\xi}_i^t,$$

where $\boldsymbol{\xi}_i^t$ denotes stochastic gradient noise. We work with a rich enough probability triple $(\Omega, \mathcal{F}, \mathbb{P})$, and define the natural filtration as the σ-algebra generated as

$$\mathcal{F}_t = \sigma\left(\left\{\boldsymbol{\xi}_i^k : 0 \leq k \leq t-1, i \in \mathcal{N}\right\}\right),$$

where $\mathcal{F}_t$ intuitively represents historical information associated with algorithm iterates up to iteration $t-1$. We make the following assumption on the stochastic gradient oracles that also formalizes the threat model considered in this work.

Assumption 1: At each iteration t: for each regular agent $i \in \mathcal{N}$, we have $\mathbf{m}_i^t = \nabla f_i(\mathbf{x}_i^t) + \boldsymbol{\xi}_i^t$ where $\mathbb{E}(\boldsymbol{\xi}_i^t \mid \mathcal{F}_t) = 0$ and $\mathbb{E}(\|\boldsymbol{\xi}_i^t\|^2 \mid \mathcal{F}_t) \leq \sigma^2$, and the set $\{\boldsymbol{\xi}_i^t : i \in \mathcal{N}\}$ is mutually independent; for each attacked agent $i \in \mathcal{A}$, $\mathbf{m}_i^t$ is arbitrary, but $\mathbf{m}_i^t$ need not be $\mathcal{F}_t$ measurable, i.e., the attacker may access arbitrary information (not available to the algorithm) to design the attack. The sets $\mathcal{N}$ and $\mathcal{A}$ are fixed but apriori unknown. We further assume that agents in $\mathcal{A}$, although suffering from potential gradient (data) attacks, are otherwise non-adversarial and follow recommended algorithmic protocols as specified.

Remark 1: Our gradient model works for general expected risk minimization. In machine learning or statistical inference tasks, f_i can be defined as

$$f_i(\mathbf{x}) := \mathbb{E}_{\vartheta \sim \mathcal{D}_i} F_i(\mathbf{x}, \vartheta),$$

where F_i is defined on local data samples ϑ_i that with distribution $\mathcal{D}_i$, and the unattacked stochastic gradient oracles return $\nabla F_i(\mathbf{x}, \vartheta)$. If f_i is defined as a finite sum over data points, then one can still sample stochastic gradients from mini-batches.

Note that this threat model only involves attack on gradient oracles but attacked agents will still follow the recommended algorithmic procedures, i.e., we assume data injection attacks on a subset of networked agents but otherwise the agents themselves are non-Byzantine. The noise model for regular agents is standard. The gradient attacks are allowed to be arbitrary, which subsumes all specific data injection attack designs.

Assumption 2: For each unattacked agent $i \in \mathcal{N}$, f_i is L-smooth, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|.$$

Assumption 3: For each unattacked agent $i \in \mathcal{N}$, f_i is convex and twice differentiable. The average objective on unattacked agents f is μ-strongly convex, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

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

Since f is also L-smooth by definition (1) and Assumption 2, we can define the condition number of f as $\kappa := L/\mu$.

Note that this strong convexity assumption is made on the average of local objective functions in $\mathcal{N}$ instead of for every single $i \in \mathcal{N}$.

Assumption 4: We assume that there exists a common global minimizer $\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^d} f_i(\mathbf{x})$ for every $i \in \mathcal{N}$.

Remark 2: Note that we do not assume that $\mathbf{x}^*$ in Assumption 4 is the unique minima of each f_i in $\mathcal{N}$. In particular, the individual f_i's may have multiple non-overlapping minima that are not minimizers of the global function f, and hence the agents need to collaborate to find the minimizer of f. Clearly, in the adversarial setting, this task is further complicated by the presence of an adversary that aims to hinder such coordination to converge to a minimizer of the global cost f.

This assumption clearly subsumes the case where each agent $i \in \mathcal{N}$ performs the same machine learning tasks on i.i.d. (independent and identically distributed) data distributions by optimizing the same strongly convex such as logistic regression. On the other hand, it also includes the case where data is not i.i.d across agents such as in distributed sensing. For example, in the problem of distributed linear parameter estimation where the parameter to be estimated is globally observable based on the collective data [23], there exists a unique global minimizer of the associated global risk function, but the data distribution on different agents can be heterogeneous when agents have different observation matrices, and hence each agent may have multiple minimizers associated with their local risk functions that are not globally optimal. We also elaborate on this case by using experiments in Section IV part A.

Distributed network-based algorithms typically involve one information mixing step to aggregate decision variables from neighboring agents, the *neighborhoods* being specified by an inter-agent communication graph $\mathcal{G}$.

Assumption 5: The inter-agent communication graph $\mathcal{G}$ is undirected and connected.

Suppose in each step of local computation each agent holds variable $\mathbf{x}_i^{t+1/2}$, we consider fixed nonnegative mixing parameters w_{ij} in update $\mathbf{x}_i^t = \sum_{j=1}^n w_{ij}\mathbf{x}_j^{t+1/2}$. We make the following standard assumption on the mixing matrix $\mathbf{W}$ composed of entries w_{ij}.

Assumption 6: The nonnegative weight matrix $\mathbf{W}$ satisfies that $w_{ij} \neq 0$ only if there is a communication link between agent i and j in graph $\mathcal{G}$, or $i = j$. Further, $\mathbf{W}$ is real symmetric, doubly stochastic, and has eigenvalues $1 = \lambda_1(\mathbf{W}) > |\lambda_2(\mathbf{W})| \geq \ldots \geq |\lambda_n(\mathbf{W})|$ with $\beta := |\lambda_2(\mathbf{W})| \in [0, 1)$.

Remark 3: Note that, under Assumption 5, there exists a $\mathbf{W}$ satisfying Assumption 6 (see [59]). In particular, in the special case when the graph $\mathcal{G}$ is complete, we may choose $\mathbf{W} = (1/n)\mathbf{1}\mathbf{1}^\top$ which recovers computing in centralized scenarios, then the problem setup reduces to the Byzantine distributed optimization where a trusted central server is present and ρ-fraction working clients may report adversarial gradients [40], while our convergence analyses remain applicable.

Assumption 7: The fraction ρ of attacked agents satisfies $\rho < 1/(1+\kappa)$.

Remark 4: We use an example to show that, for the family of all strongly convex and smooth global objective functions, $\rho < 1/(1+\kappa)$ is indeed tight for the considered threat model, i.e., recovering the exact minimum is impossible when $\rho \geq 1/(1+\kappa)$. Suppose a set of $2\,m$ agents hold the same object function x^2 (note $\kappa = 1$ in this case), and there exists an algorithm $\mathcal{M}$ with which each agent can resiliently find the optimal solution 0 when m agents, i.e., $\rho = 1/(1+\kappa) = 1/2$, are under arbitrary gradient attack. Let every attacked agent simulate objective function $(x-1)^2$, then

$$\mathcal{M}\left(\underbrace{x^2, \ldots, x^2}_{m \text{ regular agents}}, \underbrace{(x-1)^2, \ldots, (x-1)^2}_{m \text{ attacked agents}}\right) = 0. \qquad (2)$$

Now, if we set the local objectives on all $2\,m$ agents as $(x-1)^2$, and m agents are under arbitrary gradient attack and say they simulate local objective functions of x^2, then, by our hypothesis (2), and Assumption 1 that the set of attacked agents are unknown to $\mathcal{M}$, $\mathcal{M}$ will lead to solution 0 instead of the true optimizer 1. Hence, such $\mathcal{M}$ will not exist and the upper bound in Assumption 7 cannot be relaxed. On the other hand, for the subfamily of global objective functions that correspond to the same κ, for example $\kappa = 1/3$, $1/(1+\kappa)$ may not be tight when considering all algorithms in general.

TABLE I
PARAMETERS FOR ALGORITHM 1

Step size	$\alpha_t = c_\alpha(t+\varphi)^{-\tau_\alpha}$.
VR step size	$\eta_t = c_\eta(t+\varphi)^{-\tau_\eta}$.
Clipping threshold	$\gamma_t = c_\gamma(t+\varphi)^{-\tau_\gamma}$.
Constraints	$c_\alpha, c_\gamma > 0, , c_\eta \in (0,1)$, $0 < 2\tau_\gamma < \tau_\alpha < \min(1, 1-\tau_\gamma)$, $\tau_\eta = 2(\tau_\alpha + \tau_\gamma)/3$, $\varphi > 1/\big(1 - \beta^{1/(\tau_\alpha+\tau_\gamma)}\big) - 1$.
Example	$\beta = 0.5$, $\alpha_t = 0.5(t+1)^{-5/6}$, $\eta_t = 0.5(t+1)^{-23/36}$, $\gamma_t = 10(t+1)^{-1/8}$.

III. Algorithm Development and Main Results

We next develop our algorithm CLIP-VRG, see the tabular description Algorithm 1 and its parameter list Table I for details. In the distributed computation setup, each agent $i \in [n]$ holds a local iterative decision variable $\mathbf{x}_i^t \in \mathbb{R}^d$ at iteration t. Recall that each regular agent $i \in \mathcal{N}$ computes a stochastic gradient $\mathbf{m}_i^t$. We apply variance reduction and clipping operations on the stochastic (and possibly attacked) gradients at all agents with the following intended outcome: For regular agents $\mathcal{N}$, the variance reduction scheme is expected to yield a strongly consistent gradient estimator from $\mathbf{m}_i^t$; For attacked agents $\mathcal{A}$, the clipping operation can bound the influence of adversarial gradients.

By the smoothness property in Assumption 2, if the distance between consecutive iterates converges to 0, the difference between consecutive true gradients also converges to 0. Thus, we employ a local recursive averaging scheme to reduce the variance of local gradient estimates on regular agents. To this end, we develop a recursive gradient estimator $\mathbf{v}_i^t$ computed as,

$$\mathbf{v}_i^0 = \mathbf{m}_i^0,\ \mathbf{v}_i^{t+1} = (1-\eta_t)\mathbf{v}_i^t + \eta_t\mathbf{m}_i^{t+1}, \forall i \in [n], \qquad (3)$$

with the variance reduction (VR) step size

$$\eta_t = c_\eta(t+\varphi)^{-\tau_\eta} \in (0,1), \qquad (4)$$

for some positive constants c_η, τ_η, and some positive integer φ specified in Table I.

Algorithm 1: CLIP-VRG.

```
Input: $\alpha_t, \gamma_t, \eta_t$.
Initialization: $\mathbf{x}_i^0 = \mathbf{x}_j^0, \forall i, j \in [n]$.
for $t = 0, \ldots, T-1$ do
    for agent $i \in [n]$ in parallel do
        Query stochastic gradient oracle that returns $\mathbf{m}_i^t$;
        Update
        $\mathbf{v}_i^t = \begin{cases} \mathbf{m}_i^t, & t = 0, \\ (1-\eta_{t-1})\mathbf{v}_i^{t-1} + \eta_{t-1}\mathbf{m}_i^t, & t \geq 1, \end{cases}$;
        Compute $k_i^t = \begin{cases} 1, & \|\mathbf{v}_i^t\| \leq \gamma_t, \\ \gamma_t \|\mathbf{v}_i^t\|^{-1}, & \|\mathbf{v}_i^t\| > \gamma_t, \end{cases}$;
        Send $\mathbf{x}_i^t - \alpha_t k_i^t \mathbf{v}_i^t$ to all neighbors of agent $i$;
        Update $\mathbf{x}_i^{t+1} = \sum_{j=1}^n w_{ij}\left(\mathbf{x}_j^t - \alpha_t k_j^t \mathbf{v}_j^t\right)$;
    end
end
Output: $\{\mathbf{x}_i^T\}_{i \in [n]}$.
```

We use clipping to combat arbitrary adversarial gradient attacks on attacked agents $\mathcal{A}$. Specifically, we use the following distributed clipped gradient method with decaying clipping threshold, i.e., for each $i \in [n]$,

$$\mathbf{x}_i^{t+1} = \sum_{j=1}^n w_{ij}\left(\mathbf{x}_j^t - \alpha_t k_j^t \mathbf{v}_j^t\right), \tag{5}$$

where the w_{ij}'s are entries of the matrix $\mathbf{W}$ as in Assumption 6, the adaptive clipping coefficient k_i^t is defined as

$$k_i^t = \begin{cases} 1, & \|\mathbf{v}_i^t\| \leq \gamma_t, \\ \gamma_t \|\mathbf{v}_i^t\|^{-1}, & \|\mathbf{v}_i^t\| > \gamma_t, \end{cases}$$

and the clipping threshold γ_t and step size α_t are defined as

$$\gamma_t = c_\gamma (t+\varphi)^{-\tau_\gamma}, \tag{6}$$

$$\alpha_t = c_\alpha (t+\varphi)^{-\tau_\alpha} \in (0,1), \tag{7}$$

for some positive constants $c_\gamma, \tau_\gamma, c_\alpha, \tau_\alpha$ to be chosen.

We outline the procedures of CLIP-VRG in Algorithm 1, and list the parameters, i.e., three decaying sequences $\alpha_t, \beta_t, \gamma_t$, as well as their design requirements in Table I. Under this setup, we present the main results as follows.

Theorem 1: Under Assumptions 1–6, suppose that $\alpha_t, \gamma_t, \eta_t$ are taken as in (4), (6), and (7) for some tunable positive constants $c_\alpha, c_\gamma, c_\eta$, with $\tau_\eta = 2(\tau_\alpha + \tau_\gamma)/3, 2\tau_\gamma < \tau_\alpha < \min(1, 1-\tau_\gamma)$ and integer $\varphi > 1/(1-\beta^{1/(\tau_\alpha+\tau_\gamma)}) - 1$. Then, for all $i \in [n]$, for every $0 < \tau < \min(\tau_\gamma, (\tau_\alpha - 2\tau_\gamma)/3)$, we have

$$\mathbb{P}\left(\lim_{t\to\infty} (t+1)^\tau \|\mathbf{x}_i^t - \mathbf{x}^*\| = 0\right) = 1.$$

Corollary 1: Under Assumptions 1–6, we can choose $\tau_\alpha, \tau_\gamma, \tau_\eta$ in Theorem 1 to achieve that for any $i \in [n]$, any ϵ with $0 < \epsilon < 1/3$, almost surely,

$$\lim_{t\to\infty} (t+1)^{1/3-\epsilon}\left(f(\mathbf{x}_i^t) - f(\mathbf{x}^*)\right) = 0.$$

Remark 5: Theorem 1 states that asymptotically, the algorithm iterates $\mathbf{x}_i^t$ of any agent i almost surely converge to the exact minimum of f. The convergence rate is sublinear and depends on the choices of τ_α, τ_γ. To obtain Corollary 1 from Theorem 1 we solve a linear program involving τ_α, τ_γ to maximize $\min(\tau_\gamma, \tau_\alpha/3 - 2\tau_\gamma/3)$ under the constraints $2\tau_\gamma \leq \tau_\alpha \leq \min(1, 1-\tau_\gamma)$, which leads to the optimal assignment $\tau_\alpha = 5/6, \tau_\gamma = 1/6$. Thus, we can achieve a convergence rate that is arbitrarily close to $\mathcal{O}(t^{-1/6})$ for $\|\mathbf{x}_i^t - \mathbf{x}^*\|$. Since f is also L-smooth, using a standard descent lemma [60], we have $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

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

Taking $\mathbf{x} = \mathbf{x}^*$ and using the fact that $\nabla f(\mathbf{x}^*) = 0$ that follows from Assumption 3–4, we have $f(\mathbf{x}_i^t) - f(\mathbf{x}^*) \leq (L/2)\|\mathbf{x}_i^t - \mathbf{x}^*\|^2$, and thus the convergence rate for $f(\mathbf{x}_i^t) - f(\mathbf{x}^*)$ can be arbitrarily close to $\mathcal{O}(t^{-1/3})$. Note that our convergence rate is established in the almost sure sense, while most existing works in distributed stochastic optimization prove mean square convergence [3], [8], [20], [29] to the exact minimum by decaying step sizes or to some neighborhood around the optimum using constant step sizes, with the exception of [61] (see also references therein). The almost sure convergence is beneficial in that it provides convergence guarantee for nearly every algorithm sample path instance.

Note that CLIP-VRG does pay a price in convergence rate for the sake of security. In a one machine setup, the almost sure convergence rate of stochastic gradient descent for strongly convex and smooth objective function can be designed to be arbitrarily close to $\mathcal{O}(t^{-1})$ [62]. In the distributed setup with no gradient attack, [61] also proves almost sure convergence arbitrarily close to $\mathcal{O}(t^{-1})$ for distributed stochastic gradient with the aid of gradient tracking.

Finally, we point out some technical differences with respect to [23] that also uses decaying thresholds to achieve perfect recovery in distributed estimation. A key difference in our construction is the use of an additional gradient estimator (locally at each agent) based on instantaneous stochastic gradients. Intuitively, these gradient estimators lead to asymptotically decaying gradient variance which is required for convergence as the clipping operation induces additional nonlinearities. By leveraging the special form of the linear regression type cost models studied in [23], this step was essentially bypassed in lieu of a simple arithmetic mean of gradients which is not applicable in the current scenario. The addition of the non-trivial gradient estimators in turn necessitate different choices of the algorithm parameters (weight sequences and thresholds) and hence new proof techniques. However, although CLIP-VRG applies to more general convex models, we point out that the SAGE algorithm developed in [23] is optimized for linear regression models and for such setups yields a $\mathcal{O}(t^{-1/4})$ almost sure convergence rate of the agent iterates to the true minimizer, which is better than the $\mathcal{O}(t^{-1/6})$ rate we obtain for more general convex models in the adversarial setting.

Remark 6 (Proof Sketch): The proofs for Theorem 1 proceed in three steps. First, we show that in Lemma 1, thanks to the local clipping operation in Algorithm 1, in a almost sure sample path sense, the local iterate $\mathbf{x}_i^t$ converges to the network average iterate $\bar{\mathbf{x}}^t = (1/n)\sum_{i\in[n]}\mathbf{x}_i^t$; Further, its convergence rate is quantified, thus enabling us to conveniently focus on the behavior of $\bar{\mathbf{x}}^t$ in subsequent proofs. Second, in Lemma 2 and Lemma 3, we show that for regular agents $\mathcal{N}$, the developed recursive estimator $\mathbf{v}_i^t$ for the corresponding true gradient $\nabla f_i(\mathbf{x}_i^t)$ is strongly consistent and its convergence rate is also quantified. Third, we focus on the dynamics of the network average $\bar{\mathbf{x}}^t$ and show that the sequence $\{\bar{\mathbf{x}}^t\}_{t\geq 0}$ *effectively* evolves as a gradient descent type process with errors including consensus errors (from the first step), gradient estimation errors (second step), and biases introduced by local clippings, and adversarial gradient attacks. This third step is technical and, among other complexities, requires a careful analysis of the biases resulting from clipping regular (unattacked) stochastic gradients and those resulting from attacks. The derivation is achieved by considering two cases: Case 1, if $\bar{\mathbf{x}}^t$ enters some region as characterized in Lemma 5, we show that $\bar{\mathbf{x}}^t$ would stay in this region and converge to $\mathbf{x}^*$ at the same sublinear rate as clipping threshold γ_t; Case 2, if $\bar{\mathbf{x}}^t$, as described in Lemma 6, never falls into Case 1, then for each iteration t, we can lower bound the set of clipping coefficients $\{k_i^t : i \in \mathcal{N}\}$, that effectively leads the $\bar{\mathbf{x}}^t$ sequence to behave as a time-varying contractive process (due to convexity and smoothness of the objective functions) with a (controlled) clipping bias, enabling us to obtain the convergence to $\mathbf{x}^*$ at another sublinear rate. Combining these two cases concludes our analysis.

IV. Experiments

In this section, we compare the numerical performance of CLIP-VRG, the baseline distributed stochastic gradient descent (DSGD) which is the most common construction employed in non-adversarial settings, as well as Byzantine-resilient algorithms BRIDGE [18] and SCClip [20]. The DSGD algorithm we implement is adapted from the diffusion variant studied in [63], i.e., each agent in parallel updates its local estimate $\mathbf{x}_i^t$ as follows,

$$\mathbf{x}_i^{t+1} = \sum_{j=1}^{n} w_{ij}(\mathbf{x}_j^t - \alpha_t \mathbf{m}_j^t).$$

SCClip relies on some practically unknown network-wide parameters to compute clipping thresholds, which we empirically choose in our experiments as suggested by Remark 4 in [20]. BRIDGE is a DSGD-type of algorithmic framework, and we implement, by applicability, its three realizations grounded on different aggregation rules for regular agents to combine iterate information from their neighbors, i.e., BRIDGE-T based on coordinate-wise trimmed mean, BRIDGE-M based on coordinate-wise median, BRIDGE-K based on Krum, a type of secure aggregation studied in [41]. Note that SCClip enforces stringent constraints on the weight matrix for entries associated with Byzantine agents, and BRIDGE-T requires that regular agents have "enough" number of regular neighbors, and these conditions may be hard to verify and are not necessarily satisfied in our experiments. Neither BRIDGE-K or BRIDGE-M has convergence guarantee. In addition, all algorithms are initialized from the zero vector, use Metropolis weights [64] as mixing matrix $\mathbf{W}$, and are fine tuned using grid search in all experiments.

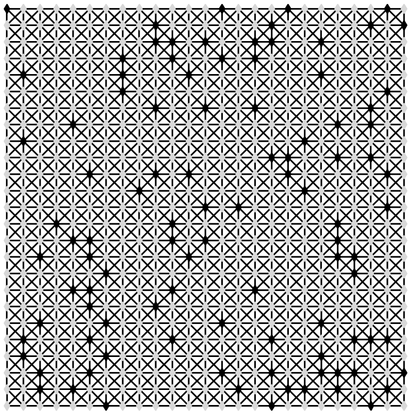


Fig. 1. 2D grid network of 625 agents.

A. Distributed Heterogeneous Measurements

In this synthetic example, we demonstrate the effectiveness of CLIP-VRG in a distributed heterogeneous measurement model. We consider an undirected 2D grid network that consists of 625 agents as shown in Fig. 1, and as indicated by inter-agent links, each agent can only communicate with its direct neighbors or agents at its diagonal position. In Fig. 1, green nodes represent regular agents in $\mathcal{N}$ while black nodes are agents in $\mathcal{A}$ that have adversarial stochastic gradient oracles (arbitrarily corrupted sensor measurements). The network of agents aims to estimate a long vector of 625 environment parameters $\boldsymbol{\theta}_*$ with each component of $\boldsymbol{\theta}_*$ corresponding to the true scalar environment parameter at the location of each agent. However, each agent only has noisy measurements on environment parameters at positions within distance of 5 units (the side length of each cell in the grid is 1 unit), so these agents need to collaborate to estimate $\boldsymbol{\theta}_*$ that has network-wide information. Specifically, for each agent $i \in \mathcal{N}$, we consider the following measurement model,

$$\mathbf{y}_i^t = \mathbf{H}_i\boldsymbol{\theta}_* + \mathbf{w}_i^t,$$

where each row of measurement matrix $\mathbf{H}_i$ is a canonical basis vector of length 625 that measures one component of $\boldsymbol{\theta}_*$ and $\{\mathbf{w}_i^t\}_{t\geq 0}$ are i.i.d. zero mean Gaussian noises. The sensing matrix $\mathbf{H}_i$ is defined to enable agent i to measure all components of $\boldsymbol{\theta}_*$ that are in positions within distance of 5 units from agent i. For example, for agent i at the center of the grid, $\mathbf{H}_i$ has 46 rows, but if agent i is at the corner of the grid then $\mathbf{H}_i$ has 26 rows. To recover the true parameter vector $\boldsymbol{\theta}_*$, we formulate an ℓ_2 loss minimization problem over regular agents $\mathcal{N}$,

$$\text{minimize}_{\mathbf{x}\in\mathbb{R}^{625}} \sum_{i\in\mathcal{N}} \mathbb{E}_{\mathbf{w}_i}\|\mathbf{H}_i\mathbf{x} - \mathbf{y}_i\|^2. \tag{8}$$

In this example, as shown in Fig. 1, we randomly sample 100 agents that are under gradient attacks. In our setting, each local objective function

$$f_i = \mathbb{E}_{\mathbf{w}_i}\|\mathbf{H}_i\mathbf{x}_i - \mathbf{y}_i\|^2$$

is convex and smooth, the true aggregated objective function $\sum_{i\in\mathcal{N}} f_i$ is strongly convex with condition number $\kappa \approx 4.35$,

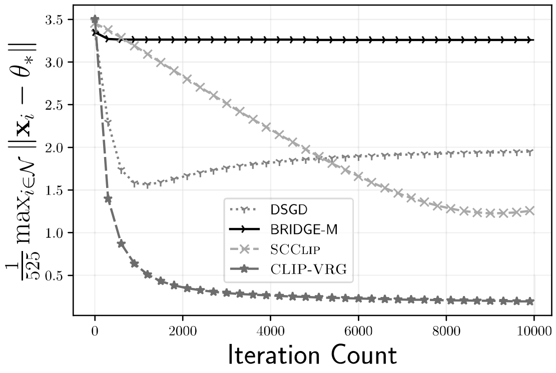


Fig. 2. Maximum ℓ_2 estimation errors comparison.

and $\boldsymbol{\theta}_*$ is an optimal solution for each f_i. Note that each f_i may have infinitely many minimizers, see also the general discussion in Remark 2. Per Assumption 7, the fraction of attacked agents in this experiment is less than $625/(1+\kappa) \approx 117$. We sample $\boldsymbol{\theta}_*$ from $[-40, 180]^{625}$ (range of temperature), and design the variance of Gaussian noise as 10. In our formulation, stochastic gradients only have nonzero entries corresponding to components being measured. For example, the stochastic gradients computed by an agent at the corner of the grid only have 26 nonzero components. To simulate gradient attack, for agents in $\mathcal{A}$, we set the nonzero components of their gradients as -200 persistently. Note that in this example the stochastic gradient computed on a regular agent $i \in \mathcal{N}$ is $2\mathbf{H}_i^\top(\mathbf{H}_i\mathbf{x}_i^t - \mathbf{H}_i\boldsymbol{\theta}_* + \mathbf{w}_i^t)$ with gradient noise being $2\mathbf{H}_i^\top\mathbf{w}_i^t$, and the gradient noise here satisfies the conditions in Assumption 1 due to our choice of $\mathbf{w}_i^t$'s as i.i.d. Gaussians.

We compare the maximum ℓ_2 estimation errors over all regular agents $\mathcal{N}$, i.e., $(1/525)\max_{i\in\mathcal{N}}\|\mathbf{x}_i^t - \boldsymbol{\theta}_*\|_2$ of for DSGD, CLIP-VRG, SCClip and BRIDGE-M (other variants are not applicable), in Fig. 2. After careful tuning: for DSGD, we pick $\alpha_t = 22/(t+1)$; for CLIP-VRG, we choose $\alpha_t = 220(t+1)^{-0.82}, \gamma_t = 600(t+1)^{-0.17}, \eta_t = 7(t+1)^{-0.66}$; for SCClip, we compute momentum step size, model step size, and clipping thresholds as suggested in the article [20]; for BRIDGE-M, we use the same step size as DSGD. Under the considered gradient attacks, DSGD fails to converge to the true parameter $\boldsymbol{\theta}_*$ and diverges after some iterations, BRIDGE-M does not make any progress, SCClip converges to some neighborhood of the optimum then starts to diverge, and only CLIP-VRG resiliently minimizes the ℓ_2 error towards 0.

B. Distributed Binary Classification

In this experiment we use real-world image classification datasets to test the efficacy of CLIP-VRG. We study a scenario where each networked agent solves the same empirical risk minimization formulation for a binary classification task to simulate distributed learning or inference on homogeneous data. Specifically, each agent has the same dataset $\{\boldsymbol{\theta}_i, \xi_i\}$ and tries to minimize a regularized logistic regression objective

$$\ell(\mathbf{x}, \{\boldsymbol{\theta}_i, \xi_i\}_{i=1,\ldots,n}) = \frac{1}{n}\sum_{i=1}^{n} \ln\left(1 + e^{-\mathbf{x}^\top\boldsymbol{\theta}_i\xi_i}\right) + \frac{\lambda}{2}\|\mathbf{x}\|_2^2,$$

where $\boldsymbol{\theta}_i$ denotes the feature vector of ith data point, ξ_i is its corresponding binary label in $\{-1, 1\}$, and λ is a regularization parameter to control overfitting. We perform this experiment on two graph topologies with different datasets.

In the first experiment setup, we consider an undirected geometric random graph of 100 agents, among which 25 agents are under arbitrary gradient attack (See Fig. 3, black nodes represent attacked agents). Each agent has access to the same Fashion-MNIST dataset [65], and use data points with labels "pullover" and "coat". Each label has 5000 training data points and 2000 test data points, and each agent solves the same regularized logistic regression formulation with $\lambda = 0.1$ on training data. For regular agents in $\mathcal{N}$, mini-batch stochastic gradients from 200 data points are sampled by shuffling at each iteration, but for attacked agents in $\mathcal{A}$, at each iteration their gradient oracles persistently return $c\mathbf{1}_{784}$ for constant $c \approx 0.714$. In this setup, we can estimate the upper bound on the fraction of attacked agents in Assumption 7 is $1/(1+\kappa) \geq 0.26$, which is larger than the actual fraction of the attacked agents 0.25. Note that Assumptions 2–7 are all satisfied in this experiment setup, except that Assumption 1 is not necessarily satisfied since regularized logistic regression may have unbounded gradient when $\mathbf{x}$ is unbounded. In Fig. 3, we compare the performance of DSGD, BRIDGE-M, SCClip and CLIP-VRG in terms of average test accuracy on test data points and the average optimality gap of training loss, i.e., $\ell(\mathbf{x}^t, \{\boldsymbol{\theta}_i, \xi_i\}_{i=1,\ldots,n}) - \ell(\mathbf{x}_*, \{\boldsymbol{\theta}_i, \xi_i\}_{i=1,\ldots,n})$, over regular agents. Under the aforementioned attacks, CLIP-VRG achieves comparable test accuracy compared with the baseline DSGD without attack, but CLIP-VRG converges slower in terms of the optimality gap. This validates our discussions in Remark 5 that the best achievable almost sure convergence rate of CLIP-VRG is inferior with respect to the theoretically achievable in non-adversarial environments. (In this experiment, we use the optimized τ_α, τ_γ as in Remark 5). From Fig. 3 we see that CLIP-VRG outperforms both BRIGE-M and SCClip while BRIDGE-M does obtain good test accuracy. Note that BRIDGE-M is an empirical method, and SCClip's underperformance may be due to that the mixing matrix in this setup is far away from what SCClip assumes.

In the second experiment setup, we use a simpler topology, a connected cycle of 15 agents where each agent has 8 neighbors, and 3 agents are under arbitrary gradient attacks (see Fig. 4, black agents represent attacked agents). Each agent has access to the same binary classification dataset a9a [65], which contains 32561 training data points and 16281 test data points. We compare the performance of DSGD, BRIDGE-K, BRIDGE-M, BRIDGE-T, SCClip, and CLIP-VRG in terms of the average test accuracy and average optimality gap of training loss, over regular agents. In this setup, we choose a regularization parameter $\lambda = 1/32561$, and each attacked agent receives persistent gradient $c\mathbf{1}_{123}$ for a constant $c \approx 1.8$. Fig. 4 shows that, under gradient attacks, CLIP-VRG performs at a comparable level with BRIDGE-K, BRIDGE-M, BRIDGE-T. Furthermore, CLIP-VRG is computationally cheaper than these counterparts, at least in terms of constant factor. BRIDGE-M and BRIDGE-T involves finding some coordinate-wise percentile of incoming vectors, BRIDGE-K even requires computing

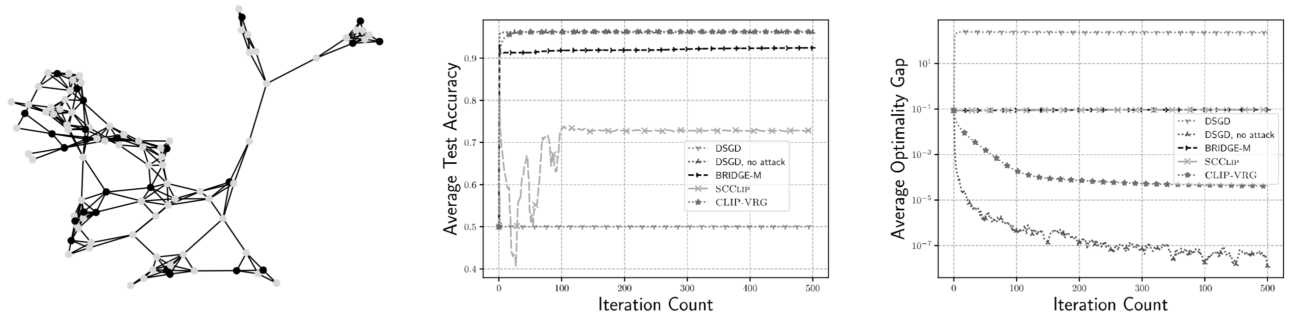


Fig. 3. An undirected random geometric graph of 100 agents with Fashion-MNIST dataset. Performance comparison of DSGD, BRIDGE-M, SCCLIP, and CLIP-VRG under persistent gradient attacks; and DSGD without attack as baseline.

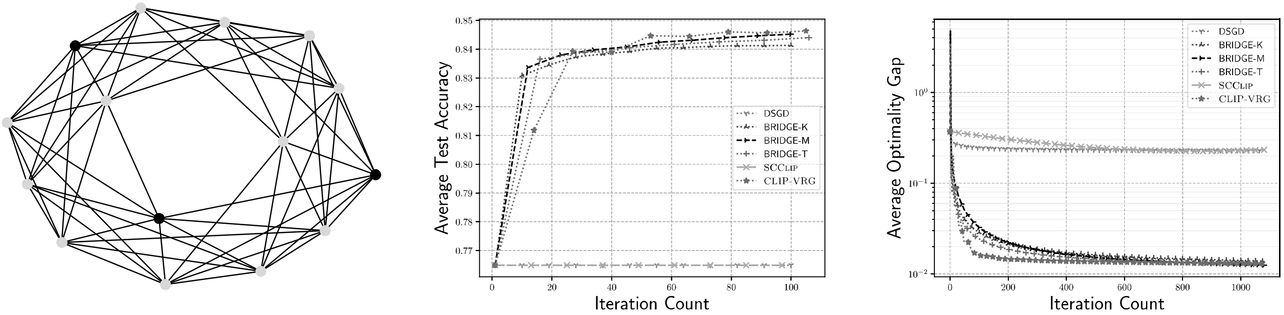


Fig. 4. An connected cycle of 15 agents with a9a dataset. Performance comparison of DSGD, BRIDGE-K, BRIDGE-M, BRIDGE-T, SCCLIP and CLIP-VRG under persistent gradient attacks.

pair-wise ℓ_2 distance among incoming vectors. Overall, the proposed CLIP-VRG demonstrates its effectiveness and some advantages.

V. Proof of Theorem 1

Define the network average of all local decision variables at iteration t as $\bar{\mathbf{x}}^t := (1/n)\sum_{i=1}^n \mathbf{x}_i^t$. Then, by the double stochasticity of $\mathbf{W}$, averaging (5) over all $i \in [n]$ gives that

$$\bar{\mathbf{x}}^{t+1} = \bar{\mathbf{x}}^t - \frac{\alpha_t}{n}\sum_{i=1}^n k_i^t \mathbf{v}_i^t. \tag{9}$$

Define long vectors and diagonal matrix

$$\mathbf{x}^t = \begin{bmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_n^t \end{bmatrix}, \mathbf{v}^t = \begin{bmatrix} \mathbf{v}_1^t \\ \vdots \\ \mathbf{v}_n^t \end{bmatrix},$$

$$\mathbf{K}^t = \operatorname{diag}([k_1^t, \dots, k_n^t]) \otimes \mathbf{I}_d.$$

Then, all local updates at iteration t can be summarized as

$$\mathbf{x}^{t+1} = (\mathbf{W} \otimes \mathbf{I}_d)(\mathbf{x}^t - \alpha_t \mathbf{K}^t \mathbf{v}^t). \tag{10}$$

We first develop the following lemma to estimate consensus error $\|\mathbf{x}^t - \mathbf{1}_n \otimes \bar{\mathbf{x}}^t\|$, i.e., the distance between local variables $\mathbf{x}_i^t$ and network average $\bar{\mathbf{x}}^t$.

Lemma 1: Take integer $\varphi > 1/(1-\beta^{1/(\tau_\alpha+\tau_\gamma)}) - 1$. Then, the iterate $\mathbf{x}^t$ generated by CLIP-VRG satisfies that for any constant $c \geq \max(\beta/[(\varphi/(1+\varphi))^{\tau_\alpha+\tau_\gamma} - \beta], \beta(1 + 1/\varphi)^{\tau_\alpha+\tau_\gamma})$, we have $\forall t \geq 1$,

$$\|\mathbf{x}^t - \mathbf{1}_n \otimes \bar{\mathbf{x}}^t\| \leq \sqrt{n}\sum_{s=0}^{t-1} \beta^{t-s}\alpha_s\gamma_s \leq c\sqrt{n}\alpha_t\gamma_t. \tag{11}$$

Proof: By the definition of k_i^t, we have

$$\forall i \in [n],\ \|k_i^t \mathbf{v}_i^t\| \leq \gamma_t. \tag{12}$$

From (10) we have

$$\mathbf{x}^t = (\mathbf{W} \otimes \mathbf{I}_d)^t \mathbf{x}^0 - \sum_{s=0}^{t-1} (\mathbf{W} \otimes \mathbf{I}_d)^{t-s} \alpha_s \mathbf{K}^s \mathbf{v}^s. \tag{13}$$

Then,

$$\begin{aligned}
&\|\mathbf{x}^t - \mathbf{1}_n \otimes \bar{\mathbf{x}}^t\| \\
&= \| \left(\mathbf{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \mathbf{I}_d\right)\mathbf{x}^t \| \\
&= \| \left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \mathbf{I}_d - \mathbf{I}_{nd}\right) \sum_{s=0}^{t-1} (\mathbf{W} \otimes \mathbf{I}_d)^{t-s}\alpha_s \mathbf{K}^s \mathbf{v}^s \| \\
&\leq \sum_{s=0}^{t-1} \| \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top - \mathbf{W}^{t-s}\|_2 \|\alpha_s \mathbf{K}^s \mathbf{v}^s\| \\
&\overset{(12)}{\leq} \sqrt{n}\sum_{s=0}^{t-1} \beta^{t-s}\alpha_s\gamma_s.
\end{aligned}$$

In the second equality above, we exploited the fact that

$$\left(\mathbf{I}_{nd} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top \otimes \mathbf{I}_d\right)(\mathbf{W}\otimes\mathbf{I}_d)^t\mathbf{x}^0 = 0, \tag{14}$$

owing to the initialization $\mathbf{x}_i^0 = \mathbf{x}_j^0$ for all $i, j \in [n]$ and the assumption that $\mathbf{W}$ is doubly stochastic. In the first inequality above, we used the Assumption 6 that $\mathbf{W}$ is real symmetric so $\mathbf{W}^{t-s}$ has eigenvalues $(\lambda_i(\mathbf{W}))^{t-s}$ for $i = 1, \ldots, n$, and $\mathbf{W}$ is stochastic thus the eigenvector of $\mathbf{W}$ associated with 1 is $(1/\sqrt{n})\mathbf{1}_n$. We next show by induction that for some positive $c := c(\beta, \tau_\alpha, \tau_\gamma, \varphi)$, we have

$$\sum_{s=0}^{t-1}\beta^{t-s}\alpha_s\gamma_s \le c\alpha_t\gamma_t. \tag{15}$$

First, for $t = 1$, if we take

$$c \ge \beta\left(1 + \frac{1}{\varphi}\right)^{\tau_\alpha+\tau_\gamma}, \tag{16}$$

then we have

$$\beta\alpha_0\gamma_0 \le c\alpha_1\gamma_1. \tag{17}$$

Next, suppose (15) holds true for some $t \ge 1$. Given that, we want to ensure that (15) also holds for $t + 1$, which is

$$\sum_{s=0}^{t}\beta^{t+1-s}\alpha_s\gamma_s = \beta\sum_{s=0}^{t-1}\beta^{t-s}\alpha_s\gamma_s + \beta\alpha_t\gamma_t \le c\alpha_{t+1}\gamma_{t+1}.$$

Since we have that (15) holds for t, to ensure the above relation, it suffices to have

$$c\beta\alpha_t\gamma_t + \beta\alpha_t\gamma_t \le c\alpha_{t+1}\gamma_{t+1}, \forall t \ge 0,$$

which can be rearranged as

$$c(\alpha_{t+1}\gamma_{t+1} - \beta\alpha_t\gamma_t) \ge \beta\alpha_t\gamma_t, \forall t \ge 0. \tag{18}$$

Then, we take φ such that $\alpha_{t+1}\gamma_{t+1} > \beta\alpha_t\gamma_t$ for all $t \ge 0$, which is equivalent to

$$\varphi > \frac{1}{1 - \beta^{1/(\tau_\alpha+\tau_\gamma)}} - 1. \tag{19}$$

Now that $\alpha_{t+1}\gamma_{t+1} - \beta\alpha_t\gamma_t > 0$, dividing it from both sides of (18) leads to

$$c \ge \frac{\beta\alpha_t\gamma_t}{\alpha_{t+1}\gamma_{t+1} - \beta\alpha_t\gamma_t}, \forall t \ge 0,$$

and taking the maximum of the right hand side gives that

$$c \ge \frac{\beta}{(\frac{\varphi}{1+\varphi})^{\tau_\alpha+\tau_\gamma} - \beta}. \tag{20}$$

Taking positive c, φ that satisfy (16), (19)–(20), and combing with the base case (17) completes the proof of this lemma. Note that we determine the choice of φ for the purpose of characterizing the constant c in (11), and all of $\varphi, c_\alpha, c_\gamma$ only affect the scaling constants of convergence rate. Instead, the choices of τ_α and τ_γ are crucial in the asymptotic rate. □

We next try to upper bound the gradient estimation errors on regular agents. As an intermediate result, we first show the following lemma.

Lemma 2: Let $\{z_t\}$ be an $\mathbb{R}_+$ stochastic sequence. Let $\mathcal{G}_{t+1}$ be the σ-algebra generated from $\{z_t\}_{t=1}^k$. Suppose that for some positive constants $c_1, c_2, 0 < a < 1$ and $a < b < a + 1$, $\{z_t\}$ satisfies that

$$\mathbb{E}\left(z_{t+1} \mid \mathcal{G}_{t+1}\right) \le (1 - c_1(t+1)^{-a})z_t + c_2(t+1)^{-b}. \tag{21}$$

Then, we have that for any $0 < \epsilon_0 < b - a$,

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^{b-a-\epsilon_0}z_t = 0\right) = 1.$$

Proof: The proof is adapted from the proof of Lemma 1 in [66]. Applying Lemma 7 in the Appendix leads to that $\forall 0 < \epsilon_0 < b - a$,

$$\lim_{t\to\infty}(t+1)^{b-a-\epsilon_0}\mathbb{E}\left(z_t\right) = 0. \tag{22}$$

Now, we fix ϵ_0. Since $0 < b - a - \epsilon_0 < 1$, $(t+1)^{b-a-\epsilon_0}$ is a concave function of t, and thus

$$(t+2)^{b-a-\epsilon_0} \le (t+1)^{b-a-\epsilon_0}[1 + (b - a - \epsilon_0)(t+1)^{-1}].$$

Multiplying the both sides of the above relation into the both sides of (21) we obtain that for sufficiently large t, there exist some constants c_3, c_4 such that

$$\begin{aligned}
&(t+2)^{b-a-\epsilon_0}\mathbb{E}\left(z_{t+1} \mid \mathcal{G}_{t+1}\right)\\
\le\ &\left[1 - \frac{c_1}{(t+1)^a} + \frac{b-a-\epsilon_0}{t+1} - \frac{c_1(b-a-\epsilon_0)}{(t+1)^{a+1}}\right]\\
&\cdot(t+1)^{b-a-\epsilon_0}z_t + \frac{c_2}{(t+1)^{a+\epsilon_0}}\left(1 + \frac{b-a-\epsilon_0}{t+1}\right)\\
\le\ &\left(1 - \frac{c_3}{(t+1)^a}\right)(t+1)^{b-a-\epsilon_0}z_t + \frac{c_4}{(t+1)^{a+\epsilon_0}}.
\end{aligned} \tag{23}$$

Define the process

$$\begin{aligned}
V(t) &= (t+1)^{b-a-\epsilon_0}z_t\\
&\quad - \sum_{i=0}^{t-1}\left[\left(\Pi_{j=i+1}^{t-1}(1 - \frac{c_3}{(j+1)^a}\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right].
\end{aligned}$$

Using Lemma 8 in the Appendix we obtain that

$$\lim_{t\to\infty}\sum_{i=0}^{t-1}\left[\left(\Pi_{j=i+1}^{t-1}(1 - \frac{c_3}{(j+1)^a}\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right] = 0, \tag{24}$$

where we used the convention that $\Pi_{j=i+1}^{t-1}(1 - c_3(j+1)^{-a}) = 1$ for $j = i - 1$. Also note that we can split

$$\begin{aligned}
&\sum_{i=0}^{t}\left[\left(\Pi_{j=i+1}^{t}\left(1 - \frac{c_3}{(j+1)^a}\right)\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right]\\
=\ &\left[1 - \frac{c_3}{(t+1)^a}\right]\sum_{i=0}^{t-1}\left[\left(\Pi_{j=i+1}^{t-1}\left(1 - \frac{c_3}{(j+1)^a}\right)\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right]\\
&+ \frac{c_4}{(t+1)^{a+\epsilon_0}}.
\end{aligned}$$

Denote $\mathcal{H}_{k+1}$ the natural filtration of the process $\{(t+1)^{b-a-\epsilon_0}\}$, and note that $V(t)$ is adapted to this filtration. Then, by the independence condition,

$$\begin{aligned} &\mathbb{E}(V(t+1)\mid \mathcal{H}_{t+1})\\ &= \mathbb{E}\left((t+2)^{b-a-\epsilon_0} z_{t+1}\mid \mathcal{H}_{t+1}\right)\\ &\quad -\sum_{i=0}^{t}\left[\left(\Pi_{j=i+1}^{t}(1-\frac{c_3}{(j+1)^a}\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right]\\ &\underset{(23)}{\le} \left[1-\frac{c_3}{(t+1)^a}\right](t+1)^{b-a-\epsilon_0} z_t + \frac{c_4}{(t+1)^{a+\epsilon_0}}\\ &\quad -\sum_{i=0}^{t}\left[\left(\Pi_{j=i+1}^{t}(1-\frac{c_3}{(j+1)^a}\right)\frac{c_4}{(i+1)^{a+\epsilon_0}}\right]\\ &= \left[1-\frac{c_3}{(t+1)^a}\right]V(t)\\ &\le V(t). \end{aligned}$$

Thus, $\{V(t)\}$ is a supermartingale. By (24), $V(t)$ is bounded from below. It follows that there exists a finite random variable V_* such that $\mathbb{P}(\lim_{t\to\infty} V(t) = V_*) = 1$. Thus, with (24) we have

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^{b-a-\epsilon_0} z_t = V_*\right) = 1.$$

Then, by Fatou's lemma and (22), we have

$$\begin{aligned} 0 &\le \mathbb{E}\left(\lim_{t\to\infty}(t+1)^{b-a-\epsilon_0} z_t\right)\\ &\le \liminf_{t\to\infty}(t+1)^{b-a-\epsilon_0}\mathbb{E}(z_t) = 0. \end{aligned}$$

Therefore, we have $\mathbb{P}(\lim_{t\to\infty}(t+1)^{b-a-\epsilon_0} z_t = 0) = 1$. □

Lemma 3: Take $\tau_\eta = 2(\tau_\alpha+\tau_\gamma)/3$ and $0 < 2\tau_\gamma < \tau_\alpha < 1$. For any $i \in \mathcal{N}$, for any $0 < \epsilon < \tau_\eta/2$, almost surely, there exists some constant c_p such that $\|\mathbf{v}_i^t - \nabla f_i(\mathbf{x}_i^t)\| \le c_p(t+1)^{-(0.5\tau_\eta-\epsilon)}$.

Proof: First, we bound the one step difference $\|\mathbf{x}_i^{t+1}-\mathbf{x}_i^t\|^2$. By Jensen's inequality,

$$\begin{aligned} &\|\mathbf{x}_i^{t+1}-\mathbf{x}_i^t\|^2\\ &= \left\|\sum_{j=1}^{n} w_{ij}(\mathbf{x}_j^t - \alpha_t k_j^t \mathbf{v}_j^t) - \mathbf{x}_i^t\right\|^2\\ &\le 2\sum_{j=1}^{n} w_{ij}\left(\|\mathbf{x}_j^t-\mathbf{x}_i^t\|^2 + \alpha_t^2\|k_j^t\mathbf{v}_j^t\|^2\right). \end{aligned}$$

By Lemma 1, for any $i,j\in[n]$ and $i\neq j$,

$$\|\mathbf{x}_j^t-\mathbf{x}_i^t\|^2 \le 2\|\mathbf{x}_j^t-\bar{\mathbf{x}}^t\|^2 + 2\|\mathbf{x}_i^t-\bar{\mathbf{x}}^t\|^2 \le 2nc^2\alpha_t^2\gamma_t^2.$$

Together with (12), we obtain

$$\|\mathbf{x}_i^{t+1}-\mathbf{x}_i^t\|^2 \le (4nc^2+2)\alpha_t^2\gamma_t^2.$$

Next, we establish the recursion for gradient estimation errors. From (3), we have

$$\begin{aligned} &\mathbf{v}_i^{t+1} - \nabla f_i(\mathbf{x}_i^{t+1})\\ &= (1-\eta_t)(\mathbf{v}_i^t - \nabla f_i(\mathbf{x}_i^t))\\ &\quad + (1-\eta_t)(\nabla f_i(\mathbf{x}_i^t) - \nabla f_i(\mathbf{x}_i^{t+1})) + \eta_t\boldsymbol{\xi}_i^{t+1}. \end{aligned} \tag{25}$$

Note that $\mathcal{F}_{t+1}$ is generated from $\{\{\boldsymbol{\xi}_i^s\}_{i\in\mathcal{N},0\le s\le t}\}$. Define $\mathbf{p}_i^t := \mathbf{v}_i^t - \nabla f_i(\mathbf{x}_i^t)$. By (25) and the assumptions on the gradient noises on good agents, we have, for $t\ge 1$,

$$\begin{aligned} &\mathbb{E}(\|\mathbf{p}_i^{t+1}\|^2\mid\mathcal{F}_{t+1})\\ &\le (1-\eta_t)^2\|\mathbf{p}_i^t\|^2 + \eta_t^2\mathbb{E}(\|\boldsymbol{\xi}_i^{t+1}\|^2\mid\mathcal{F}_{t+1})\\ &\quad + (1-\eta_t)^2\mathbb{E}(\|\nabla f_i(\mathbf{x}_i^t)-\nabla f_i(\mathbf{x}_i^{t+1})\|^2\mid\mathcal{F}_{t+1})\\ &\quad + 2(1-\eta_t)^2\langle\mathbf{p}_i^t, \nabla f_i(\mathbf{x}_i^t)-\nabla f_i(\mathbf{x}_i^{t+1})\rangle\\ &\le (1-\eta_t)^2\|\mathbf{p}_i^t\|^2 + \eta_t^2\sigma^2\\ &\quad + (1-\eta_t)^2 L^2(4nc^2+2)\alpha_t^2\gamma_t^2 + (1-\eta_t)^2\\ &\quad \cdot\left[\frac{\eta_t}{2}\|\mathbf{p}_i^t\|^2 + \frac{2}{\eta_t}\mathbb{E}\left(\|\nabla f_i(\mathbf{x}_i^t)-\nabla f_i(\mathbf{x}_i^{t+1})\|^2\mid\mathcal{F}_{t+1}\right)\right]\\ &\le (1-\eta_t)^2\left(1+\frac{\eta_t}{2}\right)\|\mathbf{p}_i^t\|^2 + \eta_t^2\sigma^2\\ &\quad + (1-\eta_t)^2 L^2\left(1+\frac{2}{\eta_t}\right)(4nc^2+2)\alpha_t^2\gamma_t^2\\ &\le (1-\eta_t)\|\mathbf{p}_i^t\|^2 + \eta_t^2\sigma^2 + \frac{3L^2}{\eta_t}(4nc^2+2)\alpha_t^2\gamma_t^2 \end{aligned}$$

where in the last inequality we used $0<\eta_t<1$ and thus

$$0 < (1-\eta_t)^2\left(1+\frac{\eta_t}{2}\right) = 1-\eta_t+\frac{1}{2}\eta_t^3 - \frac{1}{2}\eta_t \le 1-\eta_t.$$

We take $\tau_\eta = 2(\tau_\alpha+\tau_\gamma)/3$. Then, we obtain

$$\begin{aligned} &\mathbb{E}\left(\|\mathbf{p}_i^{t+1}\|^2\mid\mathcal{F}_{t+1}\right)\\ &\le (1-\eta_t)\|\mathbf{p}_i^t\|^2\\ &\quad + \left[c_\eta^2\sigma^2 + \frac{3L^2}{c_\eta}(4nc^2+2)c_\alpha^2c_\gamma^2\right](t+\varphi)^{-2\tau_\eta}. \end{aligned}$$

Then, since $0<2\tau_\gamma<\tau_\alpha<1$, we have $0<\tau_\eta<1$. Using Lemma 2, we obtain that for any $0<2\epsilon<\tau_\eta$,

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^{\tau_\eta-2\epsilon}\|\mathbf{p}_i^t\|^2 = 0\right) = 1,$$

and thus the lemma follows. Note that the choice of c_η does not change the asymptotic rate of $\|\mathbf{p}_i^t\|$, and τ_η is determined by τ_α, τ_γ. □

To show that the network average $\bar{\mathbf{x}}^t$ almost surely converges to the minimum $\mathbf{x}^*$, we discuss two exclusive cases. We first show the existence of a local convergence region. Before that, we present a standard result in convex optimization.

Lemma 4: Suppose function $h:\mathbb{R}^d\to\mathbb{R}$ is μ-strongly convex and L-smooth with minimizer $\mathbf{x}^*$. Then, the iterates generated by gradient descent $\mathbf{x}'=\mathbf{x}-\alpha\nabla h(\mathbf{x})$ with stepsize $0<\alpha\le 2/(L+\mu)$ satisfy that $\|\mathbf{x}'-\mathbf{x}^*\|\le(1-\alpha\mu)\|\mathbf{x}-\mathbf{x}^*\|$.

Proof: We prove this lemma for completeness and defer the proof to the Appendix. □

Lemma 5: Take $\tau_\eta=2(\tau_\alpha+\tau_\gamma)/3$ and $0<2\tau_\gamma<\tau_\alpha<1$. Define the auxiliary threshold

$$\overline{\gamma}_t=\frac{\gamma_t}{L}-\frac{p_t}{L}-c\sqrt{n}\alpha_t\gamma_t, \tag{26}$$

where $p_t=c_p(t+1)^{-(0.5\tau_\eta-\epsilon)}$ for some c_p and arbitrary small $0<\epsilon<0.5\tau_\eta$. Almost surely, there exist some constant c_p, finite T_0 such that if for some $t\ge T_0$ we have $\|\bar{\mathbf{x}}^t-\mathbf{x}^*\|\le\overline{\gamma}_t$, then $\|\bar{\mathbf{x}}^t-\mathbf{x}^*\|\le\overline{\gamma}_t$ for all $t\ge T_0$.

Proof: Consider the sample path $\omega\in\Omega$ such that for some $c_{p,\omega}$ we have for all $i\in\mathcal{N}$,

$$\begin{aligned}\|\mathbf{p}_i^{t,\omega}\|&=\|\mathbf{v}_i^{t,\omega}-\nabla f_i(\mathbf{x}_i^{t,\omega})\|\\&\le p_{t,\omega}=c_{p,\omega}(t+1)^{-(0.5\tau_\eta-\epsilon)}.\end{aligned}$$

By Lemma 3, such sample paths have probability measure 1. Since $\tau_\alpha>2\tau_\gamma$, we have

$$\tau_\gamma<(\tau_\alpha+\tau_\gamma)/3-\epsilon=0.5\tau_\eta-\epsilon,$$

for arbitrarily small ϵ. Then, in (26), γ_t decays slower than $p_{t,\omega}$ and $\alpha_t\gamma_t$, and thus there exists some finite t_1 such that $\forall t\ge t_{1,\omega}$, $\overline{\gamma}_{t,\omega}>0$. We decompose the update of network average $\bar{\mathbf{x}}^{t,\omega}$,

$$\begin{aligned}\bar{\mathbf{x}}^{t+1,\omega}&=\bar{\mathbf{x}}^{t,\omega}-\frac{\alpha_t}{n}\sum_{i=1}^{n}k_i^{t,\omega}\mathbf{v}_i^{t,\omega}\\&=\bar{\mathbf{x}}^{t,\omega}-\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}(\nabla f_i(\mathbf{x}_i^{t,\omega})+\mathbf{p}_i^{t,\omega})\\&\quad-\frac{\alpha_t}{n}\sum_{i\in\mathcal{A}}k_i^{t,\omega}\mathbf{v}_i^{t,\omega}.\end{aligned}\tag{27}$$

By Assumption 4, we have for any $i\in\mathcal{N},\nabla f_i(\mathbf{x}^*)=0$. Since f_i is L-smooth and by the lemma hypothesis,

$$\begin{aligned}\|\mathbf{v}_i^{t,\omega}\|&=\|\nabla f_i(\mathbf{x}_i^{t,\omega})+\mathbf{p}_i^{t,\omega}\|\\&\le\|\nabla f_i(\mathbf{x}_i^{t,\omega})-\nabla f_i(\mathbf{x}^*)\|+\|\mathbf{p}_i^{t,\omega}\|\\&\le L\|\mathbf{x}_i^{t,\omega}-\mathbf{x}^*\|+p_{t,\omega}\\&\le L\left(\|\mathbf{x}_i^{t,\omega}-\bar{\mathbf{x}}^{t,\omega}\|+\|\bar{\mathbf{x}}^{t,\omega}-\mathbf{x}^*\|\right)+p_{t,\omega}\\&\overset{(11)}{\le}\gamma_t.\end{aligned}\tag{28}$$

Thus, $k_i^{t,\omega}=1$ for all $i\in\mathcal{N}$. Then, we take $t_{2,\omega}$ as the least $t\ge t_{1,\omega}$ such that $\alpha_t\le 2/[(1-\rho)(\mu+L)]$. By Lemma 4, for $t\ge t_{2,\omega}$, from (27) we have

$$\begin{aligned}&\|\bar{\mathbf{x}}^{t+1,\omega}-\mathbf{x}^*\|\\&\le\left\|\bar{\mathbf{x}}^{t,\omega}-\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}\nabla f_i(\bar{\mathbf{x}}^{t,\omega})-\mathbf{x}^*\right\|\\&\quad+\left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}(\nabla f_i(\bar{\mathbf{x}}^{t,\omega})-\nabla f_i(\mathbf{x}_i^{t,\omega}))\right\|\\&\quad+\left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}\mathbf{p}_i^{t,\omega}\right\|+\left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{A}}k_i^{t,\omega}\mathbf{v}_i^{t,\omega}\right\|\\&\underset{(11)}{\le}[1-\alpha_t\mu(1-\rho)]\left\|\bar{\mathbf{x}}^{t,\omega}-\mathbf{x}^*\right\|\\&\quad+cL\sqrt{1-\rho}\alpha_t^2\gamma_t+(1-\rho)\alpha_t p_{t,\omega}+\rho\alpha_t\gamma_t\\&\underset{(26)}{\le}[1-\alpha_t\mu(1-\rho)]\overline{\gamma}_{t,\omega}\\&\quad+cL\sqrt{1-\rho}\alpha_t^2\gamma_t+(1-\rho)\alpha_t p_{t,\omega}\\&\quad+\rho\alpha_t(L\overline{\gamma}_{t,\omega}+p_{t,\omega}+cL\sqrt{n}\alpha_t\gamma_t)\\&=[1-\alpha_t(\mu(1-\rho)-\rho L)]\overline{\gamma}_{t,\omega}\\&\quad+cL(\sqrt{1-\rho}+\sqrt{n}\rho)\alpha_t^2\gamma_t+\alpha_t p_{t,\omega}.\end{aligned}\tag{29}$$

We next show that, for large enough t, (29) implies that $\|\bar{\mathbf{x}}^{t+1,\omega}-\bar{\mathbf{x}}^*\|\le\overline{\gamma}_{t+1,\omega}$. Define

$$\begin{aligned}\Delta_{t,\omega}&=(t+\varphi)^{\tau_\gamma}\overline{\gamma}_{t,\omega}\\&=\frac{c_\gamma}{L}-\frac{c_{p,\omega}}{L(t+\varphi)^{(\tau_\alpha-2\tau_\gamma)/3-\epsilon}}-\frac{c\sqrt{n}c_\alpha c_\gamma}{(t+\varphi)^{\tau_\alpha}}.\end{aligned}\tag{30}$$

Since $\tau_\alpha>2\tau_\gamma$ and ϵ is arbitrarily small, $\Delta_{t,\omega}$ is increasing in t, and

$$\begin{aligned}\overline{\gamma}_{t+1,\omega}&=\frac{\Delta_{t+1,\omega}}{(t+\varphi+1)^{\tau_\gamma}}\\&\ge\frac{\Delta_{t,\omega}}{(t+\varphi+1)^{\tau_\gamma}}=\left(\frac{t+\varphi}{t+\varphi+1}\right)^{\tau_\gamma}\overline{\gamma}_{t,\omega}.\end{aligned}$$

By (29), to show $\|\bar{\mathbf{x}}^{t+1}-\bar{\mathbf{x}}^*\|\le\overline{\gamma}_{t+1}$, it suffices to have

$$\begin{aligned}&[1-\alpha_t(\mu(1-\rho)-\rho L)]\overline{\gamma}_{t,\omega}\\&\quad+cL(\sqrt{1-\rho}+\sqrt{n}\rho)\alpha_t^2\gamma_t+\alpha_t p_{t,\omega}\\&\le\left(\frac{t+\varphi}{t+\varphi+1}\right)^{\tau_\gamma}\overline{\gamma}_{t,\omega}.\end{aligned}$$

By the choice of $t_{2,\omega}$, the left hand side of the above inequality is positive. Thus, we can divide $\overline{\gamma}_{t,\omega}=\Delta_{t,\omega}(t+\varphi)^{-\tau_\gamma}$ from both sides and it leads to that

$$\begin{aligned}1-\alpha_t&\left[\mu(1-\rho)-\rho L-\frac{cL(\sqrt{1-\rho}+\sqrt{n}\rho)c_\alpha c_\gamma}{\Delta_{t,\omega}(t+\varphi)^{\tau_\alpha}}\right.\\&\left.-\frac{c_{p,\omega}}{\Delta_{t,\omega}(t+\varphi)^{(\tau_\alpha-2\tau_\gamma)/3-\epsilon}}\right]\le\left(\frac{t+\varphi}{t+\varphi+1}\right)^{\tau_\gamma}.\end{aligned}\tag{31}$$

By Assumption 7 we have $\mu(1-\rho)-\rho L>0$, together with $(\tau_\alpha-2\tau_\gamma)/3-\epsilon>0$, there exists some finite $t_{3,\omega}\ge t_{2,\omega}$ such that for $t\ge t_{3,\omega}$ the left side of (31) is strictly less than 1. Using $1-x\le e^{-x}$ for $x\ge 0$, to show (31) it suffices to have

$$\frac{\alpha_t}{\tau_\gamma}\left[\mu(1-\rho)-\rho L-\frac{cL(\sqrt{1-\rho}+\sqrt{n}\rho)c_\alpha c_\gamma}{\Delta_{t,\omega}(t+\varphi)^{\tau_\alpha}}\right.$$

$$-\frac{c_{p,\omega}}{\Delta_{t,\omega}(t+\varphi)^{(\tau_\alpha-2\tau_\gamma)/3-\epsilon}}\Bigg] \geq \ln\frac{t+\varphi+1}{t+\varphi}. \tag{32}$$

Since $\Delta_{t,\omega}$ monotonically increases to c_γ/L, we can find a finite $t_{4,\omega}$ as the least $t \geq t_{3,\omega}$ such that $\Delta_{t,\omega} \geq c_\gamma/(2\,L)$. Then, for $t \geq t_{4,\omega}$, using $\ln(1+x) \leq x$ for $x \geq 0$, to show (32) it suffices to have

$$\frac{c_\alpha}{\tau_\gamma}\left[\mu(1-\rho) - \rho L - \frac{2cL^2(\sqrt{1-\rho}+\sqrt{n}\rho)c_\alpha}{(t+\varphi)^{\tau_\alpha}}\right.$$
$$\left. - \frac{2c_{p,\omega}L}{c_\gamma(t+\varphi)^{(\tau_\alpha-2\tau_\gamma)/3-\epsilon}}\right] \geq \frac{1}{(t+\varphi)^{1-\tau_\alpha}},$$

which holds true for some finite $t_{5,\omega} \geq t_{4,\omega}$ since $0 < \tau_\alpha < 1$, and thus for all $t \geq t_{5,\omega}$. Taking $T_{0,\omega} = t_{5,\omega}$ concludes the proof. □

Lemma 6: Choose $\tau_\alpha, \tau_\gamma, \overline{\gamma}_t$ as in Lemma 5, and in addition $\tau_\alpha + \tau_\gamma < 1$. Suppose that for all $t \geq 0$, we have $\|\bar{\mathbf{x}}^t - \mathbf{x}^*\| > \overline{\gamma}_t$. Then, we have for any $0 < \tau < (\tau_\alpha - 2\tau_\gamma)/3$,

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^\tau\|\bar{\mathbf{x}}^t - \mathbf{x}^*\| = 0\right) = 1.$$

Proof: Consider the sample path $\omega \in \Omega$ such that Lemma 5 holds, and since such set of ω has probability measure 1, results holds on such path ω will hold almost surely. Similar to (28), we have for any $i \in \mathcal{N}$,

$$\begin{aligned}\|\mathbf{v}_i^{t,\omega}\| &\leq \|\nabla f_i(\mathbf{x}_i^{t,\omega})\| + p_{t,\omega} \\ &\leq L\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| + cL\sqrt{n}\alpha_t\gamma_t + p_{t,\omega}.\end{aligned} \tag{33}$$

Step I. Setup recursion: Define

$$\hat{k}^{t,\omega} = \frac{\gamma_t}{L\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| + cL\sqrt{n}\alpha_t\gamma_t + p_{t,\omega}}. \tag{34}$$

Then, by (33) we have

$$\hat{k}^{t,\omega} \leq \gamma_t\|\mathbf{v}_i^{t,\omega}\|^{-1}. \tag{35}$$

Recall that by the definition of $\overline{\gamma}_{t,\omega}$ in (26),

$$\gamma_t = L\overline{\gamma}_{t,\omega} + cL\sqrt{n}\alpha_t\gamma_t + p_{t,\omega}.$$

By the lemma hypothesis that $\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \geq \overline{\gamma}_{t,\omega}$, we have $\hat{k}^{t,\omega} \leq 1$. Thus,

$$\hat{k}^{t,\omega} \leq k_i^{t,\omega} = \min\left(1, \gamma_t\|\mathbf{v}_i^{t,\omega}\|^{-1}\right). \tag{36}$$

By Assumption 3, for $i \in \mathcal{N}$, f_i is twice differentiable and convex, by the mean value theorem, there exists some matrix $\mathbf{M}_i^{t,\omega} \succeq 0$ such that

$$\nabla f_i(\bar{\mathbf{x}}^{t,\omega}) - \nabla f_i(\mathbf{x}^*) = \mathbf{M}_i^{t,\omega}(\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*).$$

Define

$$\mathbf{M}^{t,\omega} = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}}\mathbf{M}_i^{t,\omega}.$$

By Assumption 3 and 2, we have

$$0 \preceq \mathbf{M}_i^{t,\omega} \preceq L\mathbf{I},\ \mu\mathbf{I} \preceq \mathbf{M}^{t,\omega} \preceq L\mathbf{I}.$$

From (27), we have the relation

$$\begin{aligned}&\|\bar{\mathbf{x}}^{t+1,\omega} - \mathbf{x}^*\| \\ &\leq \left\|\bar{\mathbf{x}}^{t,\omega} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\nabla f_i(\bar{\mathbf{x}}^{t,\omega}) - \mathbf{x}^*\right\| \\ &\quad + \left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}(\nabla f_i(\bar{\mathbf{x}}^{t,\omega}) - \nabla f_i(\mathbf{x}_i^{t,\omega}))\right\| \\ &\quad + \left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{p}_i^{t,\omega}\right\| + \left\|\frac{\alpha_t}{n}\sum_{i\in\mathcal{A}}k_i^{t,\omega}\mathbf{v}_i^{t,\omega}\right\|.\end{aligned} \tag{37}$$

We have that

$$\begin{aligned}&\|\bar{\mathbf{x}}^{t,\omega} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\nabla f_i(\bar{\mathbf{x}}^{t,\omega}) - \mathbf{x}^*\| \\ &\leq \left\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^* - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}(\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*)\right\| \\ &\leq \left\|\mathbf{I} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}\right\|_2 \left\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\right\|.\end{aligned} \tag{38}$$

Let $\lambda_1(\cdot)$ denote the largest eigenvalue. Take T_1 as the least t such that $\alpha_t < 1/[L(1-\rho)]$, then for $t \geq T_1$, the symmetric matrix $\mathbf{I} - (\alpha_t/n)\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}$ is positive definite, and thus

$$\left\|\mathbf{I} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}\right\|_2 = \lambda_1\left(\mathbf{I} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}\right),$$

$$\left\|\mathbf{I} - \frac{\alpha_t\hat{k}^{t,\omega}}{n}\sum_{i\in\mathcal{N}}\mathbf{M}_i^{t,\omega}\right\|_2 = \lambda_1\left(\mathbf{I} - \frac{\alpha_t\hat{k}^{t,\omega}}{n}\sum_{i\in\mathcal{N}}\mathbf{M}_i^{t,\omega}\right).$$

Next, we use the fact that for any pair of symmetric matrices A, B,

$$\lambda_1(A+B) \leq \lambda_1(A) + \lambda_1(B).$$

It follows that

$$\begin{aligned}&\lambda_1\left(\mathbf{I} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}\right) \\ &= \lambda_1\left(\mathbf{I} - \frac{\alpha_t\hat{k}^{t,\omega}}{n}\sum_{i\in\mathcal{N}}\mathbf{M}_i^{t,\omega}\right) \\ &\quad + \lambda_1\left(\frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}(\hat{k}^{t,\omega} - k_i^{t,\omega})\mathbf{M}_i^{t,\omega}\right)\end{aligned}$$

Since for all $i \in \mathcal{N}$, $\hat{k}^{t,\omega} - k_i^{t,\omega} \leq 0$ and $\mathbf{M}_i^{t,\omega}$ is positive semidefinite, we have

$$\left\|\mathbf{I} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\mathbf{M}_i^{t,\omega}\right\|_2 \leq \left\|\mathbf{I} - \frac{\alpha_t\hat{k}^{t,\omega}}{n}\sum_{i\in\mathcal{N}}\mathbf{M}_i^{t,\omega}\right\|_2. \tag{39}$$

Combing relations (38) and (39) we obtain

$$\left\|\bar{\mathbf{x}}^{t,\omega} - \frac{\alpha_t}{n}\sum_{i\in\mathcal{N}}k_i^{t,\omega}\nabla f_i(\bar{\mathbf{x}}^{t,\omega}) - \mathbf{x}^*\right\|$$

$$\begin{aligned}
&\le \|\mathbf{I} - \alpha_t(1-\rho)\hat{k}^{t,\omega}\mathbf{M}^{t,\omega}\|_2\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \\
&\le [1 - \alpha_t\mu(1-\rho)\hat{k}^{t,\omega}]\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\|.
\end{aligned}$$

Combing the above with (36) (37) leads to the recursion

$$\begin{aligned}
&\|\bar{\mathbf{x}}^{t+1,\omega} - \mathbf{x}^*\| \\
&\le [1 - \alpha_t\mu(1-\rho)\hat{k}^{t,\omega}]\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \\
&\quad + cL\sqrt{1-\rho}\alpha_t^2\gamma_t + (1-\rho)\alpha_t p_{t,\omega} + \rho\alpha_t\gamma_t \\
&= \left[1 - \alpha_t[\mu(1-\rho) - \rho L]\hat{k}^{t,\omega}\right]\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \\
&\quad + cL(\sqrt{1-\rho} + \rho\sqrt{n})\alpha_t^2\gamma_t + \alpha_t p_{t,\omega}. \qquad (40)
\end{aligned}$$

Step II. Lower bound for $\hat{k}^{t,\omega}$: We next show that there exists some positive constant $c_{k,\omega}$ such that

$$\hat{k}^{t,\omega} \ge c_{k,\omega}(t+\varphi)^{-\tau_\gamma}. \qquad (41)$$

By the definition of $\hat{k}^{t,\omega}$ in (34), to show (41) it suffices to show that

$$\sup_{t\ge 0}\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| < \infty, \qquad (42)$$

which is equivalent to show that $\sup_{t\ge T_1}\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| < \infty$. Define the system

$$\begin{aligned}
\hat{m}_{t+1,\omega} &= \left[1 - \frac{(\mu(1-\rho) - \rho L)\,\alpha_t\gamma_t}{Lm_{t,\omega} + cL\sqrt{n}\alpha_t\gamma_t + p_{t,\omega}}\right]m_{t,\omega} \\
&\quad + cL(\sqrt{1-\rho} + \rho\sqrt{n})\alpha_t^2\gamma_t + \alpha_t p_{t,\omega}, \\
m_{t+1,\omega} &= \max\left(\hat{m}_{t+1,\omega}, m_{t,\omega}\right),
\end{aligned}$$

with initial condition $m_{T_1,\omega} = \|\bar{\mathbf{x}}^{T_1,\omega} - \mathbf{x}^*\|$. Note that since $2\tau_\alpha + \tau_\gamma > 4\tau_\alpha/3 + \tau_\gamma/3$, for some constant $c_{m,\omega}$ we have

$$\begin{aligned}
&cL(\sqrt{1-\rho} + \rho\sqrt{n})\alpha_t^2\gamma_t + \alpha_t p_{t,\omega} \\
&= c_{m,\omega}(t+\varphi)^{-(\frac{4}{3}\tau_\alpha + \frac{1}{3}\tau_\gamma - \epsilon)}.
\end{aligned}$$

Then, since $\tau_\alpha > 2\tau_\gamma$, the dynamics of $\{m_{t,\omega}\}_{t\ge T_1}$ falls into the pursuit of Lemma 9 in Appendix and leads to $\sup_{t\ge T_1} m_{t,\omega} < \infty$. By (40) and the definition of $m_{t,\omega}$, $\forall t \ge T_1$, $\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \le m_{t,\omega}$, and thus (42) follows.

Step III. Convergence rate: Combing (40) and (41), we have

$$\begin{aligned}
&\|\bar{\mathbf{x}}^{t+1,\omega} - \mathbf{x}^*\| \\
&\le \left[1 - c_{k,\omega}c_\alpha[\mu(1-\rho) - \rho L](t+\varphi)^{-(\tau_\alpha+\tau_\gamma)}\right]\|\bar{\mathbf{x}}^{t,\omega} - \mathbf{x}^*\| \\
&\quad + c_{m,\omega}(t+\varphi)^{-(\frac{4}{3}\tau_\alpha + \frac{1}{3}\tau_\gamma - \epsilon)}.
\end{aligned}$$

Then, with $\tau_\alpha + \tau_\gamma < 1$, we can apply Lemma 7 in Appendix to obtain the desired convergence rate. □

Proof of Theorem 1: From Lemma 5 and Lemma 6, we have for every $0 \le \tau < \min(\tau_\gamma, (\tau_\alpha - 2\tau_\gamma)/3)$,

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^\tau\|\bar{\mathbf{x}}^t - \mathbf{x}^*\| = 0\right) = 1.$$

By the triangle inequality and (11), for every $i \in [n]$

$$\begin{aligned}
\|\mathbf{x}_i^t - \mathbf{x}^*\| &\le \|\bar{\mathbf{x}}^t - \mathbf{x}^*\| + \|\mathbf{x}_i^t - \bar{\mathbf{x}}^t\| \\
&\le \|\bar{\mathbf{x}}^t - \mathbf{x}^*\| + c\sqrt{n}\alpha_t\gamma_t.
\end{aligned}$$

Since $\tau < \tau_\alpha + \tau_\gamma$, we have

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^\tau\|\mathbf{x}_i^t - \mathbf{x}^*\| = 0\right) = 1.$$

□

VI. Conclusion

In this article, we have studied a relatively unexplored threat model in distributed optimization, namely that of arbitrary gradient attacks, and we have proposed a distributed stochastic gradient method CLIP-VRG that combines local variance reduced gradient estimation and clipping to achieve resilience in the presence of such threats. We have identified a readily computable upper bound, determined by the condition number of the aggregate (strongly convex) objective function, on the fraction of attacked agents that may be tolerated by the proposed CLIP-VRG scheme. Under some similarity conditions among local objective functions and the above mentioned attack threshold condition, we have established the almost sure convergence of CLIP-VRG to the exact minimum, which is empirically supported by experiments on both synthetic and real-world image classification datasets. Future directions include extending CLIP-VRG to nonconvex or finite sum objective functions where different upper tolerance bounds on the fraction of attacked agents and new similarity conditions may be needed, involving techniques to further mitigate the impact of data heterogeneity such as gradient tracking, and understanding the convergence with respect to other notions.

Appendix

Proof of Lemma 4: Define $p(\mathbf{x}) = h(\mathbf{x}) - (\mu/2)\|\mathbf{x} - \mathbf{x}^*\|^2$, then p is convex and $(L-\mu)$-smooth. If $L > \mu$, we have (Theorem 2.1.5 in [67])

$$\langle\nabla p(\mathbf{x}), \mathbf{x} - \mathbf{x}^*\rangle \ge \frac{1}{L-\mu}\|\nabla p(\mathbf{x})\|^2. \qquad (43)$$

Since

$$\mathbf{x}' - \mathbf{x}^* = (1-\alpha\mu)(\mathbf{x} - \mathbf{x}^*) - \alpha\nabla p(\mathbf{x}), \qquad (44)$$

we obtain

$$\begin{aligned}
&\|\mathbf{x}' - \mathbf{x}^*\|^2 \\
&= (1-\alpha\mu)^2\|\mathbf{x} - \mathbf{x}^*\|^2 - 2\alpha(1-\alpha\mu)\langle\mathbf{x} - \mathbf{x}^*, \nabla p(\mathbf{x})\rangle \\
&\quad + \alpha^2\|\nabla p(\mathbf{x})\|^2 \\
&\overset{(43)}{\le} (1-\alpha\mu)^2\|\mathbf{x} - \mathbf{x}^*\|^2 - \frac{\alpha[2 - \alpha(\mu+L)]}{L-\mu}\|\nabla p(\mathbf{x})\|^2.
\end{aligned}$$

Since $\alpha \le 2/(\mu+L)$, the second term of the above display is nonnegative, so the desired relation is obtained. If $L = \mu$, then h is a quadratic function and it turns out

$$h(\mathbf{x}) = h(\mathbf{x}^*) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^*\|^2,$$

and $\nabla p(\mathbf{x}) = 0$, so (44) reduces to $\mathbf{x}' - \mathbf{x} = (1-\alpha\mu)(\mathbf{x} - \mathbf{x}^*)$, and taking Euclidean norms on both sides completes the proof. □

Lemma 7 (Lemma 5 in [68]): Consider the scalar positive sequences $0 < u_t \le 1$ that

$$u_t = u_0(t+1)^{-a}, \text{ and } w_t = w_0(t+1)^{-b},$$

with $0 \le a < 1$ and $a < b$. Then for

$$y_{t+1} = (1-u_t)y_t + w_t,$$

for every $0 < \epsilon < b-a$, $\lim_{t\to\infty}(t+1)^{b-a-\epsilon}y_t = 0$.

Lemma 8 (Lemma 25 in [2]): Let the sequences $\{u_t\}, \{w_t\}$ be given by

$$u_t = u_0(t+1)^{-a}, w_t = w_0(t+1)^{-b}$$

where $u_0, w_0, a \ge 0$, and $b > a$. The for arbitrary fixed j,

$$\lim_{t\to\infty}\sum_{k=j}^{t-1}\left[\left(\Pi_{l=k+1}^{t-1}(1-u(t))\right)w_t\right] = 0.$$

Lemma 9: Consider a scalar dynamical system $\{m_t\}$ of the form

$$\hat{m}_{t+1} = \left(1 - \frac{u_t}{m_t + v_t}\right)m_t + w_t,$$
$$m_{t+1} = \max\left(|\hat{m}_{t+1}|, |m_t|\right),$$

where $m_0 > 0$ and v_t is a positive decaying sequence, and

$$u_t = \frac{u_0}{(t+1)^a},\ w_t = \frac{w_0}{(t+1)^b},$$

for some positive constants $b > a, u_0, w_0$. Then, it satisfies that $\sup_{t>0} m_t < \infty$.

Proof: We prove the lemma by showing that there exists some finite T such that for all $t \ge T$, $m_{t+1} = m_t$. Notice that a sufficient condition for $m_{t+1} = m_t$ is $|\hat{m}_{t+1}| \le |m_t|$ and $m_t > 0$.

By definition, $\{m_t\}$ is a nondecreasing positive sequence, so $m_t + v_t > m_0$. By the definition of u_t, take $t_0 = \lceil (u_0/m_0)^{1/a} - 1 \rceil$, then all $t > t_0$ we have $u_t/(m_t + v_t) < 1$. Then, for $t \ge t_0$, we have $m_t > 0, \hat{m}_t > 0$ and $|\hat{m}_{t+1} \le |m_t|$ reduces to $\hat{m}_{t+1} \le m_t$.

$$m_t - \hat{m}_{t+1} = \frac{u_t m_t}{m_t + v_t} - w_t.$$

Since m_t is nondecreasing and v_t is decaying, we have

$$\frac{m_t}{m_t + v_t} = \frac{1}{1 + (v_t/m_t)} \ge \frac{m_{t_0}}{m_{t_0} + v_{t_0}} := c_0 > 0.$$

Then, by the definitions of u_t, w_t, for

$$t \ge t_0' := \max\left(t_0, \left\lceil \left(\frac{w_0}{c_0 u_0}\right)^{\frac{1}{b-a}} - 1 \right\rceil\right),$$

we have

$$m_t - \hat{m}_{t+1} \ge c_0 u_t - w_t > 0.$$

Therefore, taking $T = t_0'$ we have for all $t \ge T$, $m_{t+1} = m_t$, and thus $\sup_{t\ge 0} m_t = m_T < \infty$. □

REFERENCES

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3575–3605, Jun. 2012.

[3] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[4] A. Nedic, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 92–101, May 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artif. Intell. Statist.*, pp. 1273–1282, 2017.

[6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[7] R. Xin, S. Pu, A. Nedić, and U. A. Khan, "A general framework for decentralized optimization with first-order methods," *Proc. IEEE Proc. IRE*, vol. 108, no. 11, pp. 1869–1889, Nov. 2020.

[8] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5336–5346.

[9] S. Yu, Y. Chen, and S. Kar, "Resilient decentralized optimization in multi-agent networks with data injection attack," in *Proc. 55th Asilomar Conf. Signals, Syst., Comput.*, 2021, pp. 1032–1036.

[10] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient distributed estimation: Sensor attacks," *IEEE Trans. Autom. Control*, vol. 64, no. 9, pp. 3772–3779, Sep. 2019.

[11] Y. Chen, S. Kar, and J. M. F. Moura, "The Internet of Things: Secure distributed inference," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 64–75, Sep. 2018.

[12] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[13] E. Gorbunov, A. Borzunov, M. Diskin, and M. Ryabinin, "Secure distributed training at scale," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 7679–7739.

[14] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," 2022, *arXiv:2206.04568*.

[15] L. Su and N. H. Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2227–2233, May 2021.

[16] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[17] Z. Yang and W. U. Bajwa, "ByRDIE: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 4, pp. 611–627, Dec. 2019.

[18] C. Fang, Z. Yang, and W. U. Bajwa, "BRIDGE: Byzantine-resilient decentralized gradient descent," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 610–626, 2022.

[19] S. Guo et al., "Byzantine-resilient decentralized stochastic gradient descent," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 4096–4106, Jun. 2022.

[20] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust decentralized learning via self-centered clipping," 2022, *arXiv:2202.01545*.

[21] J. Peng, W. Li, and Q. Ling, "Byzantine-robust decentralized stochastic optimization over static and time-varying networks," *Signal Process.*, vol. 183, 2021, Art. no. 108020.

[22] N. Gupta and N. H. Vaidya, "Fault-tolerance in distributed optimization: The case of redundancy," in *Proc. 39th Symp. Princ. Distrib. Comput.*, 2020, pp. 365–374.

[23] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient distributed parameter estimation with heterogeneous data," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4918–4933, Oct. 2019.

[24] S. Kar and J. M. F. Moura, "Consensus + innovations distributed inference over networks: Cooperation and sensing in networked systems," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 99–109, May 2013.

[25] S. Al-Sayed, A. M. Zoubir, and A. H. Sayed, "Robust distributed estimation by networked agents," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 3909–3921, Aug. 2017.

[26] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[27] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[28] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[29] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Program.*, vol. 187, no. 1, pp. 409–457, 2021.

[30] D. Jakovetić, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[31] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[32] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[33] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, Jun. 2014.

[34] Z. Wang, J. Zhang, T.-H. Chang, J. Li, and Z.-Q. Luo, "Distributed stochastic consensus optimization with momentum for nonconvex nonsmooth problems," *IEEE Trans. Signal Process.*, vol. 69, pp. 4486–4501, 2021.

[35] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized SGD via matching decomposition sampling," in *Proc. 6th Indian Control Conf.*, 2019, pp. 299–300.

[36] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin, "Exponential graph is provably efficient for decentralized deep training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 13975–13987.

[37] A. Koloskova, S. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3478–3487.

[38] T. Vogels et al., "Relaysum for decentralized deep learning on heterogeneous data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 28004–28015.

[39] E. Cyffers, M. Even, A. Bellet, and L. Massoulié, "Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022.

[40] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.

[41] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 118–128.

[42] R. Guerraoui et al., "The hidden vulnerability of distributed learning in Byzantium," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.

[43] C. Xie, O. Koyejo, and I. Gupta, "Zeno: Byzantine-suspicious stochastic gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6893–6901.

[44] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.

[45] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 4618–4628.

[46] D. Data and S. Diggavi, "Byzantine-resilient SGD in high dimensions on heterogeneous data," in *Proc. IEEE Int. Symp. Inf. Theory*, 2021, pp. 2310–2315.

[47] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "SignSGD with majority vote is communication efficient and fault tolerant," in *Proc. 9th Int. Conf. Learn. Representations*, 2019.

[48] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 1544–1551.

[49] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for Byzantine robust optimization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5311–5319.

[50] E. M. E. Mhamdi, R. Guerraoui, and S. L. A. Rouault, "Distributed momentum for Byzantine-resilient stochastic gradient descent," in *Proc. 9th Int. Conf. Learn. Representations*, 2021.

[51] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Robust fully asynchronous SGD," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10495–10503.

[52] J. Regatti, H. Chen, and A. Gupta, "Bygars: Byzantine SGD with arbitrary number of attackers," 2020, *arXiv:2006.13421*.

[53] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 903–912.

[54] B. Turan, C. A. Uribe, H.-T. Wai, and M. Alizadeh, "Robust distributed optimization with randomly corrupted gradients," *IEEE Trans. Signal Process.*, vol. 70, pp. 3484–3498, 2022.

[55] B. Zhu et al., "Byzantine-robust federated learning with optimal statistical rates," *Artif. Intell. Statist.*, pp. 3151–3178, 2023.

[56] G. Damaskinos et al., "Asynchronous byzantine machine learning (the case of SGD)," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1145–1154.

[57] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 8635–8645.

[58] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via bucketing," in *Proc. 10th Int. Conf. Learn. Representations*, 2022.

[59] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[60] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.

[61] R. Xin, U. A. Khan, and S. Kar, "Variance-reduced decentralized stochastic optimization with accelerated convergence," *IEEE Trans. Signal Process.*, vol. 68, pp. 6255–6271, 2020.

[62] O. Sebbouh, R. M. Gower, and A. Defazio, "Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball," in *Proc. Conf. Learn. Theory*, 2021, pp. 3935–3971.

[63] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

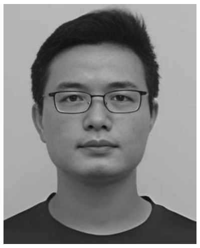
[64] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 63–70.

[65] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[66] S. Yu, Y. Chen, and S. Kar, "Dynamic median consensus over random networks," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 5695–5702.

[67] Y. Nesterov et al., *Lectures on Convex Optimization*, vol. 137. Berlin, Germany: Springer, 2018.

[68] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.



Shuhua Yu (Graduate Student Member, IEEE) received the B.Eng. degree in computer science and engineering from The Chinese University of Hong Kong, Shenzhen, China, in May 2019. He is currently working toward the Ph.D. degree with Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include optimization, machine learning, and multi-agent systems.

Soummya Kar (Fellow, IEEE) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, Kharagpur, India, in May 2005, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010. He is currently a Professor of electrical and computer engineering with Carnegie Mellon University. From June 2010 to May 2011, he was with the Electrical Engineering Department, Princeton University, Princeton, NJ, USA, as a Postdoctoral Research Associate. He has authored or coauthored more than 250 articles in journals and conference proceedings and holds multiple patents in his research field which include decision-making in large-scale networked systems, stochastic systems, and machine learning.