symmetry

MDPI

Article

CC-GT to GFLCC: From Synchronous to Asynchronous Conjugate Compressed Gradient Tracking for Distributed Optimization

Linqing Zha [1], Wenkang Chen [2] and Xuejun Zhang [2,3,4,*]

[1] School of Physical Science and Technology, Guangxi University, 100 Daxue East Road, Nanning 530004, China; 2312270126@st.gxu.edu.cn
[2] School of Electrical Engineering, Guangxi University, 100 Daxue East Road, Nanning 530004, China; 2312401016@st.gxu.edu.cn
[3] School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China
[4] Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning 530004, China
* Correspondence: xjzhang@gxu.edu.cn

Abstract

In distributed optimization, communication bottlenecks and computational heterogeneity are two critical challenges that undermine system scalability and efficiency. To address these issues, we propose a synchronous algorithm, CC-GT (Conjugate Compressed Gradient Tracking), which integrates conjugate gradient acceleration with dual error-compensated compression to reduce communication overhead and accelerate convergence. Theoretically, we establish its linear convergence under strongly convex and smooth assumptions, and extensive experiments on regression, classification, and neural-network tasks validate the fast convergence and communication efficiency of the synchronous CC-GT. Building on this foundation, we develop GFLCC, an asynchronous extension that eliminates global synchronization barriers while preserving the core mechanisms of CC-GT, thereby inherently mitigating the straggler effect and enabling dynamic load balancing. For the asynchronous GFLCC, extensive comparisons against state-of-the-art asynchronous baselines under harsh network interference demonstrate its strong empirical convergence speed, communication efficiency, and robustness. These results confirm that the proposed framework offers a practical and theoretically grounded solution for communication-constrained and heterogeneous distributed optimization systems.

Keywords: distributed optimization; multi-agent systems; gradient tracking; conjugate gradient method; communication compression; linear convergence; asynchronous algorithm

Check for updates

Academic Editors: Sergei Odintsov and Calogero Vetro

Received: 13 July 2026
Revised: 1 September 2026
Accepted: 5 September 2026
Published: 8 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Distributed multi-agent systems have emerged as a research hotspot in the fields of artificial intelligence and automation, demonstrating significant potential across various cutting-edge applications. In the domain of unmanned aerial vehicle swarms, distributed optimization algorithms facilitate cooperative flight without centralized coordination [1,2], optimizing multi-vehicle collaboration while enhancing safety and reducing mission completion time for intelligence, surveillance, reconnaissance, and multi-target attack operations [3]. In distributed sensor networks, the fusion of multiple local observations enhances target tracking accuracy and stability [4], offering notable advantages in reliability, robustness, and applicability [5]. For satellite formation control [6], distributed coordination

effectively accommodates the traffic demands of non-uniform ground users under resource constraints while reducing satellite handover delay and system load [7]. In smart grid applications, distributed optimization enables coordinated scheduling of renewable energy sources and demand-side response management [8], enhancing both grid stability and economic efficiency [9]. In communication networks, distributed optimization methods maximize network utility while minimizing congestion [10], addressing distributed bandwidth allocation problems for multi-source traffic flows [11]. Consequently, research on distributed multi-agent systems has found extensive applications across defense, military, and civilian domains, emerging as a prominent research point in the artificial intelligence field, with significant value in improving system efficiency. In multi-agent distributed optimization, the frequent exchange of massive parameters across large-scale networks imposes extremely high demands on communication bandwidth [12], making the design of communication-efficient algorithms crucial for enhancing system scalability and practicality [13,14]. Furthermore, beyond communication bottlenecks, the inherent heterogeneity of computational capacities and varying network delays across distributed servers pose another critical challenge [15,16]. Traditional synchronous frameworks suffer from the severe "straggler effect," where faster nodes must idle while waiting for slower ones, inevitably leading to computational load imbalance and degraded overall system efficiency. Therefore, developing asynchronous distributed algorithms has become an imperative strategy to achieve dynamic load balancing, mitigate the impact of heterogeneous stragglers, and maximize resource utilization in practical complex networks.

1.1. Related Works

For multi-agent distributed optimization problems, the gradient descent algorithm (DGD) offers a simple and broadly applicable solution [17]. Its update rule is intuitive: each agent only needs to communicate with neighbors and compute its own gradient, making it suitable for various optimization problems, including constrained improvement and stochastic optimization, while scaling well to large-scale networks. However, DGD suffers from several limitations: it fails to leverage function smoothness [18] for accelerated convergence, cannot achieve exact convergence with fixed step sizes, and requires diminishing step sizes that lead to slow convergence [19]. To address these issues, researchers have proposed gradient tracking algorithms (GT) [20], which leverage gradient difference correction to enable fixed step sizes while preserving convergence guarantees, fully exploiting function smoothness to achieve optimal convergence rates [21]. Extensions comprise the distributed stochastic gradient tracking approach (DSGT) alongside a gossip-type stochastic gradient tracking variant (GSGT) [22]. Gradient tracking techniques have been widely applied across distributed machine learning and deep learning training [23,24], online stochastic optimization and dynamic tracking [25], multi-agent system cooperative control [26,27], multi-sensor fusion and target tracking [28], privacy-preserving distributed optimization, and various artificial intelligence domains. Recent publications have further explored distributed optimization via sparse update-based training [29] and HPC job scheduling [30], reflecting the sustained interest in decentralized algorithms. Since the introduction of gradient tracking, research has expanded across multiple branches, achieving significant progress in theoretical foundations, algorithmic extensions, and application domains [31]. Li et al. proposed a unified algorithmic framework integrating variance reduction with gradient tracking, revealing the trade-offs inherent in this combination to achieve robust performance and fast convergence [23]. Xin et al. introduced Nesterov momentum into distributed optimization, proposing distributed gradient methods applicable to arbitrary graphs [32]. Chen et al. developed differentially private gradient tracking algorithms, integrating gradient tracking with privacy preservation to enhance

algorithmic security. In Bayesian learning, Bhar et al. proposed a gradient tracking decentralized unadjusted Langevin algorithm that reduces the dependence of convergence bias on network topology while maintaining convergence and enhancing robustness to data heterogeneity [33]. For large-scale language model training, Rajabi et al. introduced the SubTrack-Grad method, leveraging gradient subspace tracking to substantially reduce both runtime and pre-training time. However, research on communication-compressed gradient tracking algorithms remains relatively limited, with slow convergence rates leaving substantial room for investigation.

In multi-agent distributed optimization, each agent solves global optimization problems through local computation and neighbor communication. As system scale expands and the number of agents increases, communication overhead becomes a critical bottleneck [34]. In this context, communication compression techniques have emerged to reduce communication volume while maintaining algorithm convergence. To alleviate correspondence pressure, various methods, including weight sharing, quantization, sparsification, and low-rank approximation reduce transmitted data volume [35,36]. The mathematical essence of communication compression lies in the design of compression operators—functional mappings that project high-dimensional gradients or parameters into low-dimensional representations [37]. Based on their mathematical properties, compression operators can be categorized as unbiased compression or biased compression [38,39]. From an implementation perspective, compression operators can be further classified into quantization operators (mapping floating-point numbers to low-bit integer representations) [39], sparsification operators (retaining only Top-K or random K significant elements) [40], low-rank approximation operators (reducing parameter dimensionality through matrix factorization) [41], and hybrid operators combining multiple compression strategies [42]. Existing gradient compression methods face challenges in scalability and accuracy degradation. Vogels et al. proposed PowerSGD, which maintains computational efficiency while preserving model accuracy [41]. Cheng et al. developed LIEC-SGD, a local immediate error compensation stochastic gradient descent algorithm based on bidirectional compression and carefully designed compensation mechanisms, reducing communication costs while accelerating convergence [43]. Karimireddy et al. demonstrated that biased compression algorithms such as SignSGD diverge without error compensation, and proposed a unified error feedback framework enabling biased compression to achieve convergence rates comparable to unbiased compression [44]. Xin et al. first combined compression operators with gradient tracking, proposing the compressed gradient tracking algorithm (CGT), achieving linear convergence under strongly convex and smooth conditions [45]. Islamov et al. [46] introduced MoTEF, a novel method integrating communication compression with momentum tracking and error feedback, achieving most desirable properties and significantly outperforming existing methods under arbitrary data heterogeneity. Compression algorithms have become a frontier research direction in distributed optimization [47], yet research on gradient tracking algorithms incorporating compression operators remains limited, with convergence analysis being particularly scarce.

In addition, the BEER algorithm has recently been proposed as an alternative approach for communication-efficient gradient tracking [48]. BEER leverages bidirectional error compensation to mitigate compression-induced bias while maintaining a low communication overhead. Its main advantage lies in faster convergence compared to traditional gradient tracking methods under moderate network heterogeneity, as well as improved robustness to compression errors [49]. However, BEER may incur additional computational complexity due to the residual tracking mechanism and requires careful tuning of its error-feedback parameters. Despite these limitations, BEER represents one of the more optimized algorithms currently available [50] and will be used as a benchmark for comparison in the numerical

experiments presented. Nevertheless, it is crucial to note that BEER, alongside the majority of existing gradient tracking methods, fundamentally operates under a strictly synchronous framework. In practical deployment environments characterized by heterogeneous computing resources and varied network delays, such synchronous designs inevitably suffer from severe idle times. The overall system progress is tightly bottlenecked by the slowest "straggler" nodes, defeating the purpose of load balancing. To overcome this computational inefficiency, asynchronous distributed optimization has recently emerged as a promising paradigm [51]. By completely eliminating global synchronization barriers, asynchronous algorithms allow individual heterogeneous servers to compute, update, and communicate autonomously. This flexibility not only naturally circumvents the straggler effect but also significantly enhances overall resource utilization and systemic resilience [52,53], which motivates the asynchronous design adopted in this work.

Broader context in multi-agent cooperative control. The proposed GFLCC not only addresses communication-efficient distributed optimization but also connects with the broader landscape of multi-agent cooperative control. Recent advances at this intersection include active disturbance rejection consensus control (ADRC) based on novel nonlinear extended state observers (NESO), which enable robust coordination under heterogeneous external disturbances [54]; event-triggered consensus mechanisms grounded in the maximum correntropy criterion, suppressing non-Gaussian noise in discrete-time multi-agent systems [55]; robust optimal consensus designs that jointly handle modeling uncertainties and external disturbances [56]; and distributed optimization frameworks for fractional-order double-integrator multi-agent systems subject to communication delays [57]. These works provide essential context for understanding the system behavior of GFLCC under three key aspects: *(i)* heterogeneous disturbances—the error-compensated compression and conjugate acceleration in GFLCC serve analogous roles to disturbance observers in ADRC, actively correcting the bias introduced by quantization; *(ii)* event-triggered communication—while GFLCC advances through Poisson-paced asynchronous local updates rather than explicit event-triggered conditions, both paradigms share the goal of reducing unnecessary communication; and *(iii)* time delays—the push-sum protocol with FIFO mass-increment queues in GFLCC is designed to accommodate delayed arrivals, paralleling the delay-compensation strategies studied in fractional-order multi-agent systems. The comprehensive comparison in Table 1 further situates GFLCC among representative related works.

Literature comparison. Table 1 provides a structured comparison of the proposed GFLCC against representative works at the intersection of distributed optimization, multi-agent control, and communication-efficient learning. The key dimensions include: execution paradigm (synchronous vs. asynchronous), gradient tracking, compression mechanism, and conjugate direction acceleration.

GFLCC uniquely combines all four properties—asynchronous execution, gradient tracking, bidirectional error-compensated compression, and conjugate direction acceleration—within a single algorithm. No prior work simultaneously addresses all these dimensions, which explains the significant practical advantages observed in our experiments.

In distributed training applications, fuzzy neural networks have attracted increasing attention due to their unique advantages. Distributed fuzzy neural networks leverage fuzzy logic to mitigate uncertainty [58] while processing data in a distributed manner, maintaining model interpretability while effectively handling uncertain problems. Convolutional fuzzy neural networks alleviate latency issues in computationally intensive fuzzy neural network inference, emerging as alternative models for IoT applications [59]. The decentralized nonstationary fuzzy neural network (DNFNN) introduces gradient tracking mechanisms to

overcome obstacles in backpropagation, substantially improving large-scale optimization efficiency [60]. Xun et al. proposed fluid dynamic DNNs with nested incremental training algorithms, enabling subnetworks to operate independently or cooperatively, providing new insights for distributed fuzzy neural network inference on edge devices while enhancing reliability and adaptability [61]. Zhao et al. developed multi-objective evolutionary construction methods for deep fuzzy rough neural networks, integrating rough set theory with fuzzy neural networks to handle uncertainty and imprecision, offering novel perspectives for fuzzy neural network design in distributed scenarios [62]. However, research on applying communication compression techniques to distributed fuzzy neural network training remains in its infancy. The inherent difficulty in directly evaluating the computational complexity of fuzzy layers [59], together with weight adaptation requirements for multi-sub network cooperative output [60], presents additional challenges for direct application of traditional gradient compression methods. This research gap not only limits the scalability of fuzzy neural networks in large-scale distributed scenarios but also opens broad research opportunities for integrating communication compression with gradient tracking in more interpretative neural network architectures.

Table 1. Comparison of GFLCC with representative related works along key technical dimensions.

Method	Async	Grad. Track.	Compress.	Conj. Dir.
DGD [17]	×	×	×	×
DSGT [22]	×	✓	×	×
CGT [45]	×	✓	✓	×
BEER [48]	×	✓	✓	×
ADRC-NESO [54]	✓	×	×	×
ET-MCC [55]	✓	×	×	×
CSGrasp [63]	×	×	✓	×
ComStar [64]	×	×	✓	×
DRUDM-CFG [65]	✓	×	×	×
MoE^2 [66]	✓	×	✓	×
FTP+d [56]	✓	×	×	×
Async-DSGT [22]	✓	✓	×	×
GFLCC (Ours)	✓	✓	✓	✓

✓ denotes that the method possesses the corresponding property; × denotes that it does not.

1.2. Contributions

This paper makes the following key contributions:

- We develop CC-GT, a novel synchronous conjugate compressed gradient tracking algorithm designed for communication-efficient multi-agent distributed optimization. By integrating conjugate gradient mechanisms into the conventional gradient tracking framework, CC-GT effectively tackles a broad range of distributed optimization problems while significantly reducing communication overhead.
- We provide rigorous theoretical guarantees for CC-GT under general compression operator assumptions. The analysis confirms that CC-GT attains a linear convergence rate for strongly convex and smooth objective functions, even in the presence of uncoordinated step-sizes across agents, thereby establishing a solid theoretical foundation for its advantages.
- To tackle the ubiquitous computational heterogeneity and the straggler problem in real-world systems, we generalize the conjugate gradient and bidirectional compression strategies of CC-GT into a fully asynchronous framework, termed GFLCC. By eliminating the need for global synchronization barriers, this asynchronous extension seamlessly accommodates delayed network communications and uncoordinated

updates. Consequently, it empowers individual servers to compute autonomously, achieving dynamic load balancing and significantly enhancing system resilience.

- Extensive numerical simulations are carried out to evaluate the performance of both CC-GT and its asynchronous extension GFLCC. Experimental results show that CC-GT outperforms state-of-the-art synchronous methods in convergence speed and communication efficiency, while GFLCC demonstrates superior robustness and efficiency over asynchronous baselines under harsh network interference.

1.3. Organizations

The remainder of this paper is structured as follows. Section 2 describes the distributed optimization problem formulation and the system model under general compression operator assumptions. Section 3 details the proposed compressed conjugate gradient tracking algorithm (CC-GT and GFLCC). Section 4 establishes the convergence analysis and theoretical guarantees. Section 5 reports numerical experiments that validate the algorithm's performance. Section 6 concludes the paper.

1.4. Notations

Consider a network of n agents collaboratively solving a distributed optimization problem. Each agent i possesses a local objective function $f_i : \mathbb{R}^p \to \mathbb{R}$ that is known only to itself. Throughout this paper, vectors are assumed to be column vectors unless stated otherwise. Let each agent i hold a local copy $\mathbf{a}_i \in \mathbb{R}^p$ of the decision variable, a gradient tracker (auxiliary variable) $\mathbf{b}_i \in \mathbb{R}^p$, and a scalar $\beta_i \in \mathbb{R}$ used in the conjugate direction update. At iteration k, these quantities are denoted by $\mathbf{a}_i^k \in \mathbb{R}^p$, $\mathbf{b}_i^k \in \mathbb{R}^p$, and $\beta_i^k \in \mathbb{R}$, respectively.

$$\mathbf{A} := [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]^\top \in \mathbb{R}^{n\times p}, \quad \mathbf{B} := [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]^\top \in \mathbb{R}^{n\times p},$$

where each row of $\mathbf{A}$ (resp. $\mathbf{B}$) is the transpose of the corresponding agent's column vector, i.e., the i-th row of $\mathbf{A}$ is $\mathbf{a}_i^\top \in \mathbb{R}^{1\times p}$. The network averages are defined as

$$\bar{\mathbf{A}} := \tfrac{1}{n}\mathbf{1}^\top\mathbf{A} \in \mathbb{R}^{1\times p}, \quad \bar{\mathbf{B}} := \tfrac{1}{n}\mathbf{1}^\top\mathbf{B} \in \mathbb{R}^{1\times p},$$

where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones column vector. Equivalently, in column-vector form,

$$\bar{\mathbf{a}}^k := \tfrac{1}{n}\sum_{i=1}^n \mathbf{a}_i^k \in \mathbb{R}^p, \quad \bar{\mathbf{b}}^k := \tfrac{1}{n}\sum_{i=1}^n \mathbf{b}_i^k \in \mathbb{R}^p,$$

so that $\bar{\mathbf{A}}^k = (\bar{\mathbf{a}}^k)^\top$ and $\bar{\mathbf{B}}^k = (\bar{\mathbf{b}}^k)^\top$.

The conjugate coefficients are stacked into a diagonal matrix

$$\boldsymbol{\beta}^k := \mathrm{diag}([\beta_1^k, \beta_2^k, \ldots, \beta_n^k]) \in \mathbb{R}^{n\times n}.$$

We further define the diagonal step-size matrix

$$\mathbf{D} := \mathrm{diag}([\eta_1, \eta_2, \ldots, \eta_n]) \in \mathbb{R}^{n\times n},$$

where $\eta_i > 0$ is the local step size of agent i, and $\bar{\eta} = \frac{1}{n}\sum_{i=1}^n \eta_i$, $\hat{\eta} = \max_i \eta_i$.

At the k-th iteration, their values are denoted by $\mathbf{A}^k, \mathbf{B}^k, \boldsymbol{\beta}^k, \bar{\mathbf{A}}^k$ and $\bar{\mathbf{B}}^k$, respectively. Auxiliary variables of the agents (in an aggregative matrix form) $\mathbf{M}_a, \mathbf{M}_b, \mathbf{Q}_a, \mathbf{Q}_b, \hat{\mathbf{A}}, \hat{\mathbf{B}}$ are defined similarly.

To facilitate the algorithm description, we define the aggregate function matrix $\mathbf{F}: \mathbb{R}^{n\times p} \to \mathbb{R}^n$ and the aggregate gradient matrix $\nabla\mathbf{F}: \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p}$ as follows:

$$\mathbf{F}(\mathbf{A}) := [f_1(\mathbf{a}_1), f_2(\mathbf{a}_2), \ldots, f_n(\mathbf{a}_n)]^\top \in \mathbb{R}^n.$$

Denote the aggregate gradient

$$\nabla\mathbf{F}(\mathbf{A}) := [\nabla f_1(\mathbf{a}_1), \nabla f_2(\mathbf{a}_2), \ldots, \nabla f_n(\mathbf{a}_n)]^\top \in \mathbb{R}^{n\times p},$$

and

$$\nabla\bar{\mathbf{F}}(\mathbf{A}) := \frac{1}{n}\nabla\mathbf{F}(\mathbf{A}) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{a}_i) \in \mathbb{R}^{1\times p}.$$

We use $\|\cdot\|$ to denote the Frobenius norm of vectors and matrices by default. Specifically, for square matrices, $\|\cdot\|$ represents the spectral norm. The spectral radius of a square matrix $\mathbf{P}$ is denoted by $\rho(\mathbf{P})$.

Table 2 provides a consolidated summary of the key symbols used throughout this paper for quick reference.

Table 2. Summary of key notation.

Symbol	Dimension	Description
n	scalar	Number of agents
p	scalar	Dimension of decision variable
$\mathbf{a}_i^k$	$\mathbb{R}^p$	Agent i's decision variable (column vector)
$\mathbf{b}_i^k$	$\mathbb{R}^p$	Agent i's gradient tracker (column vector)
β_i^k	$\mathbb{R}$	Agent i's conjugate coefficient (FR)
$\mathbf{A}^k$	$\mathbb{R}^{n\times p}$	Stacked decision matrix (rows: $\mathbf{a}_i^\top$)
$\mathbf{B}^k$	$\mathbb{R}^{n\times p}$	Stacked gradient tracker (rows: $\mathbf{b}_i^\top$)
$\boldsymbol{\beta}^k$	$\mathbb{R}^{n\times n}$	Diagonal matrix of conjugate coefficients
$\mathbf{D}$	$\mathbb{R}^{n\times n}$	Diagonal step-size matrix, $\mathrm{diag}(\eta_1, \ldots, \eta_n)$
$\mathbf{S}^k$	$\mathbb{R}^{n\times p}$	Conjugate search direction matrix
$\bar{\mathbf{A}}^k$	$\mathbb{R}^{1\times p}$	Network average, $\frac{1}{n}\mathbf{1}^\top\mathbf{A}^k$
$\bar{\mathbf{a}}^k$	$\mathbb{R}^p$	Network average (column form), $= (\bar{\mathbf{A}}^k)^\top$
$\nabla\mathbf{F}(\mathbf{A}^k)$	$\mathbb{R}^{n\times p}$	Aggregate gradient (rows: $\nabla f_i^\top$)
$\mathbf{W}$	$\mathbb{R}^{n\times n}$	Doubly stochastic mixing matrix
$\mathbf{1}$	$\mathbb{R}^n$	All-ones column vector
$\mathcal{Q}_s$	—	Stochastic uniform quantizer with s levels

2. Problem Formulations

In this section, we provide the assumptions on the communication graphs and the objective functions. Then, we discuss different kinds of compression methods and provide a general description for compression operators.

2.1. Preliminaries

We begin by introducing the conditions on the communication network and the objective functions. Assume that the agents are connected through an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ denotes the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. For an arbitrary agent $i \in \mathcal{V}$, its set of neighbors is denoted by $\mathcal{N}_i$. We make the following assumption regarding the network structure:

Assumption 1. *The undirected graph $\mathcal{G}$ is strongly connected, and there exists a non-negative doubly stochastic weight matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n\times n}$. In other words, agent i can receive information from agent j if and only if $\mathbf{W}_{ij} > 0$, and*

$$\mathbf{W}\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^\top \mathbf{W} = \mathbf{1}^\top.$$

Remark 1. *Although we assume an undirected graph, the CGT method considered here also applies to any balanced directed graph, where a doubly stochastic weight matrix can be constructed.*

Assumption 2. *Each local cost function f_i is μ_i-strongly convex, and its gradient is L_i-Lipschitz continuous, i.e., for any $\mathbf{a}, \mathbf{a}' \in \mathbb{R}^p$,*

$$\langle \nabla f_i(\mathbf{a}) - \nabla f_i(\mathbf{a}'), \mathbf{a} - \mathbf{a}' \rangle \geq \mu_i \|\mathbf{a} - \mathbf{a}'\|^2,$$

$$\|\nabla f_i(\mathbf{a}) - \nabla f_i(\mathbf{a}')\| \leq L_i \|\mathbf{a} - \mathbf{a}'\|.$$

Under Assumption 2, the global objective function f is μ-strongly convex and its gradient is L-Lipschitz continuous, where

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \mu_i, \quad L = \max_i L_i.$$

Moreover, there exists a unique solution to the optimization problem, denoted by $\mathbf{a}^* \in \mathbb{R}^p$.

2.2. Centralized Conjugate Gradient Method

The conjugate gradient method (CG) is an efficient first-order algorithm designed for large-scale optimization problems. Before introducing the conjugate gradient method, we first recall the classical gradient descent method. For an unconstrained optimization problem $\min_{\mathbf{a} \in \mathbb{R}^n} f(\mathbf{a})$, the gradient descent method updates the iterate as

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \alpha_k \mathbf{d}_k,$$

where $\alpha_k > 0$ is the step size. This method follows the steepest descent direction, which is the negative gradient. However, for ill-conditioned problems, gradient descent often exhibits a "zigzagging" phenomenon, leading to slow convergence.

The conjugate gradient method addresses this limitation by incorporating information from previous search directions. Instead of simply taking the negative gradient, it constructs a set of mutually conjugate directions, allowing for faster convergence, especially for quadratic convex problems. The detailed procedure is presented in Algorithm 1.

Algorithm 1 Centralized Conjugate Gradient Method (Fletcher–Reeves)

Require: Initial point $\mathbf{a}_0 \in \mathbb{R}^n$, tolerance $\varepsilon > 0$
Ensure: Approximate optimal solution $\mathbf{a}^*$

$\mathbf{g}_0 = \nabla f(\mathbf{a}_0)$, $\mathbf{d}_0 = -\mathbf{g}_0$, $k = 0$
while $\|\mathbf{g}_k\| > \varepsilon$ **do**
  $\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{a}_k + \alpha \mathbf{d}_k)$ ▷ Exact line search
  $\mathbf{a}_{k+1} = \mathbf{a}_k + \alpha_k \mathbf{d}_k$
  $\mathbf{g}_{k+1} = \nabla f(\mathbf{a}_{k+1})$
  $\beta_k = \dfrac{\mathbf{g}_{k+1}^\top \mathbf{g}_{k+1}}{\max(\mathbf{g}_k^\top \mathbf{g}_k,\ \varepsilon)}$ ▷ Fletcher–Reeves formula
  $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
  $k = k + 1$
end while
return $\mathbf{a}_k$

The algorithm begins with an initial point $\mathbf{a}_0$ and a prescribed tolerance $\varepsilon > 0$. In the initialization step, we compute the initial gradient $\mathbf{g}_0 = \nabla f(\mathbf{a}_0)$ and set the initial search direction to the negative gradient $\mathbf{d}_0 = -\mathbf{g}_0$, ensuring a descent direction.

At each iteration, the step size α_k is determined by exact line search along the current search direction $\mathbf{d}_k$, and the iterate is updated as $\mathbf{a}_{k+1} = \mathbf{a}_k + \alpha_k \mathbf{d}_k$. The gradient at the new point is then evaluated as $\mathbf{g}_{k+1} = \nabla f(\mathbf{a}_{k+1})$. Subsequently, the Fletcher–Reeves conjugate coefficient is computed as $\beta_k = (\mathbf{g}_{k+1}^\top \mathbf{g}_{k+1})/(\mathbf{g}_k^\top \mathbf{g}_k)$, where the denominator is safeguarded by $\max(\mathbf{g}_k^\top \mathbf{g}_k, \varepsilon)$ to prevent division by near-zero values. Finally, the search direction is updated as $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$, which combines the current negative gradient with a scaled previous direction to maintain conjugacy. The iteration counter is incremented, and the process repeats until the gradient norm falls below ε.

2.3. Compression Methods

In this subsection, we introduce common assumptions on compression operators and present a unified framework [45].

Assumption 3. *The compression operator $C : \mathbb{R}^d \to \mathbb{R}^d$ satisfies the unified framework of [45]: there exist constants $C \geq 0$, $\delta \in (0, 1]$, and $r \geq 1$ such that for all $a \in \mathbb{R}^d$,*

$$\mathbb{E}\|C(a) - a\|^2 \leq C\|a\|^2, \qquad \mathbb{E}\|C(a)\|^2 \leq r^2\|a\|^2,$$

where the contraction factor $\delta = 1 - C \in (0, 1]$ requires $C < 1$ and is used by the convergence analysis (e.g., the contraction $c_a = \tau_a(1 - \alpha_a r\delta) < 1$ and the step-size bound $\gamma \leq \delta/[m(2 - \delta)]$).

This unified framework covers both unbiased and biased compressors, offering greater flexibility in algorithm design. It also accommodates more general compressors, such as sign-based compressors and compositions of quantization and sparsification.

In the experiments, we employ the standard stochastic uniform quantizer $\mathcal{Q}_s : \mathbb{R}^d \to \mathbb{R}^d$ with s quantization levels, defined as follows. For a vector $\mathbf{x} \in \mathbb{R}^d$ with $\|\mathbf{x}\| \neq 0$, let $\xi_j = \operatorname{sign}(x_j)$ and $\zeta_j = |x_j|/\|\mathbf{x}\| \in [0, 1]$ for each coordinate $j = 1, \ldots, d$. The quantized value of the j-th coordinate is

$$\mathcal{Q}_s(\mathbf{x})_j = \xi_j \, \|\mathbf{x}\| \, \frac{\lfloor s\,\zeta_j \rfloor + b_j}{s}, \qquad j = 1, \ldots, d,$$

where $b_j \sim \text{Bernoulli}(s\,\zeta_j - \lfloor s\,\zeta_j \rfloor)$ is an independent random variable ensuring unbiasedness, i.e., $\mathbb{E}[\mathcal{Q}_s(\mathbf{x})] = \mathbf{x}$. Since $0 \leq \lfloor s\,\zeta_j \rfloor + b_j \leq s$, each quantized coordinate satisfies $|\mathcal{Q}_s(\mathbf{x})_j| \leq \|\mathbf{x}\|$, so no overflow occurs. The encoding of each coordinate uses $\lceil \log_2 s \rceil$ bits for the level index and 1 bit for the sign, while the norm $\|\mathbf{x}\|$ is transmitted once in full precision (32 bits); hence the experiments use $s = 16$ (4-bit level, 5 bits/coordinate including sign) and $s = 4$ (2-bit level, 3 bits/coordinate including sign).

It is well known [39,45] that $\mathcal{Q}_s$ satisfies the unified Assumption 3. A direct variance analysis of the Bernoulli rounding (each coordinate is rescaled by an independent factor in $[0, 1]$ with mean ζ_j) gives

$$\mathbb{E}\|\,\mathcal{Q}_s(\mathbf{x}) - \mathbf{x}\|^2 \leq \frac{d}{4s^2}\,\|\mathbf{x}\|^2,$$

so the variance constant is $C = d/(4s^2)$ (the commonly cited looser bound is d/s^2). The contraction factor is therefore $\delta = 1 - C$, which is admissible ($\delta \in (0, 1]$) precisely when $C < 1$, i.e., $s > \sqrt{d}/2$. Moreover, since $\mathcal{Q}_s$ is unbiased, the variance decomposition gives

$$\mathbb{E}\|\,\mathcal{Q}_s(\mathbf{x})\|^2 \;=\; \|\mathbf{x}\|^2 + \mathbb{E}\|\,\mathcal{Q}_s(\mathbf{x}) - \mathbf{x}\|^2 \;\leq\; \Big(1 + \frac{d}{4s^2}\Big)\|\mathbf{x}\|^2,$$

and the r-scaling constant in Assumption 3 is therefore the finite, quantization-level-dependent quantity $r = \sqrt{1 + d/(4s^2)}$ (e.g., $r \approx 1.22$ for $d = 500$, $s = 16$, and $r \approx 1.03$ for $d = 4$, $s = 4$). For the ridge regression experiment ($d = 500$, $s = 16$): $C = 500/1024 =$

$0.488 < 1$ and $\delta = 0.512 \in (0, 1]$; for the logistic regression experiment ($d = 4$, $s = 4$): $C = 4/64 = 0.0625$ and $\delta = 0.9375 \in (0, 1]$. Both settings therefore yield well-conditioned compressors that strictly satisfy Assumption 3, and $C \to 0$, $\delta \to 1$, $r \to 1$ as s increases.

3. Proposed Algorithms: From Synchronous CC-GT to Asynchronous GFLCC

To address the communication bottleneck in distributed optimization, we first propose the synchronous Conjugate Compressed Gradient Tracking (CC-GT) algorithm, which integrates conjugate gradient acceleration with dual compression to achieve fast convergence and low communication overhead. However, the synchronous nature of CC-GT inherently suffers from the straggler effect in heterogeneous environments, where agents with uneven computational capacities cause global waiting and system inefficiency. To overcome this limitation, we extend CC-GT to an asynchronous setting, resulting in the proposed GFLCC (Gradient-tracking for Fast Load-balancing with Conjugate Compression) algorithm. GFLCC preserves the core mechanisms of CC-GT—conjugate gradient acceleration and dual compression—while eliminating global synchronization barriers through a node-centric, event-driven design. Below, we present CC-GT as the synchronous baseline, followed by its asynchronous extension GFLCC as the main contribution of this work.

3.1. Synchronous CC-GT Algorithm

The synchronous CC-GT algorithm is summarized in Algorithm 2.

Algorithm 2 Conjugate Compressed Gradient Tracking Algorithm (CC-GT)

Input:
  Stopping time K, step sizes $\{\eta_i\}$, consensus step-size γ
  Scaling parameters α_a, α_b, initial values $\mathbf{A}^0, \mathbf{M}_a^0, \mathbf{M}_b^0$
Initialization:
  $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla \mathbf{F}(\mathbf{A}^0)$ ▷ bias-free initialization
  $\mathbf{M}_{a,w}^0 = \mathbf{W}\mathbf{M}_a^0$, $\mathbf{M}_{b,w}^0 = \mathbf{W}\mathbf{M}_b^0$
for $k = 0$ to $K - 1$ **do**
    $(\hat{\mathbf{A}}^k, \hat{\mathbf{A}}_w^k, \mathbf{M}_a^{k+1}, \mathbf{M}_{a,w}^{k+1}) = \mathrm{COMM}(\mathbf{A}^k, \mathbf{M}_a^k, \mathbf{M}_{a,w}^k, \alpha_a)$
    $(\hat{\mathbf{B}}^k, \hat{\mathbf{B}}_w^k, \mathbf{M}_b^{k+1}, \mathbf{M}_{b,w}^{k+1}) = \mathrm{COMM}(\mathbf{B}^k, \mathbf{M}_b^k, \mathbf{M}_{b,w}^k, \alpha_b)$
    $\mathbf{A}^{k+1} = \mathbf{A}^k - \gamma(\hat{\mathbf{A}}^k - \hat{\mathbf{A}}_w^k) + \mathbf{D}\mathbf{B}^k$ ▷ corrected sign
    Compute local Fletcher–Reeves coefficient:

$$\beta_i^k = \frac{\nabla f_i(\mathbf{a}_i^{k+1})^\top \nabla f_i(\mathbf{a}_i^{k+1})}{\max\left(\nabla f_i(\mathbf{a}_i^k)^\top \nabla f_i(\mathbf{a}_i^k),\ \varepsilon\right)}$$

    Update conjugate direction:

$$\mathbf{S}^{k+1} = -\nabla \mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k \mathbf{S}^k$$

    Update gradient tracker:

$$\mathbf{B}^{k+1} = \mathbf{B}^k - \gamma(\hat{\mathbf{B}}^k - \hat{\mathbf{B}}_w^k) + \mathbf{S}^{k+1} - \mathbf{S}^k$$

end for
  return $\mathbf{A}^K, \mathbf{B}^K$

Procedure: $\mathrm{COMM}(\mathbf{N}, \mathbf{M}, \mathbf{M}_w, \alpha_z)$
  $\mathbf{Q} = \mathrm{Compress}(\mathbf{N} - \mathbf{M})$
  $\hat{\mathbf{N}} = \mathbf{M} + \mathbf{Q}$
  $\hat{\mathbf{N}}_w = \mathbf{M}_w + \mathbf{W}\mathbf{Q}$
  $\mathbf{M} \leftarrow (1 - \alpha_z)\mathbf{M} + \alpha_z \hat{\mathbf{N}}$
  $\mathbf{M}_w \leftarrow (1 - \alpha_z)\mathbf{M}_w + \alpha_z \hat{\mathbf{N}}_w$
  return $\hat{\mathbf{N}}, \hat{\mathbf{N}}_w, \mathbf{M}, \mathbf{M}_w$
Output: Optimal decision variables $\mathbf{A}^K, \mathbf{B}^K$

The COMM procedure encapsulates compression and communication: each agent compresses the difference between its current variable and a memory reference point, transmits the compressed value, and updates the memory via momentum smoothing. This difference-based compression ensures that the compression error vanishes as the memory approaches the variable, effectively mitigating compression distortion under Assumption 3.

Building upon the compressed gradient tracking (CGT) framework, CC-GT introduces a conjugate gradient mechanism to accelerate convergence. At each iteration, every agent computes its own conjugate coefficient β_i^k as the ratio of inner products of local gradients at two consecutive iterations:

$$\beta_i^k = \frac{\nabla f_i(\mathbf{a}_i^{k+1})^\top \nabla f_i(\mathbf{a}_i^{k+1})}{\max\left(\nabla f_i(\mathbf{a}_i^k)^\top \nabla f_i(\mathbf{a}_i^k),\ \varepsilon\right)}. \tag{1}$$

These coefficients are assembled into a diagonal matrix $\boldsymbol{\beta}^k$, and the conjugate search direction matrix is updated as:

$$\mathbf{S}^{k+1} = -\nabla \mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k \mathbf{S}^k. \tag{2}$$

This update is a matrix-form generalization of the direction update in centralized conjugate gradient methods, enabling each agent to dynamically adjust its search direction using historical gradient information to avoid redundant zigzagging.

Remark on the guarded condition. Theorem A1 establishes the convergence of CC-GT under the guarded condition $|\beta_i^k| \le \beta_{\max} < 1$ for all $i \in \mathcal{N}$ and all $k \ge 0$: this stronger condition is assumed in the convergence analysis, and it is the condition under which the theoretical linear-convergence guarantee is obtained. The numerical experiments, however, are *not* restricted to this condition: fast and stable convergence is achieved even when the strong condition is not explicitly enforced, which demonstrates the wider applicability of the proposed algorithm. This mirrors a standard situation in optimization—many algorithms are provably convergent only under (strong) convexity, yet are routinely and successfully applied to non-convex problems in practice. In both cases, the proof requires stronger conditions, while the empirical behavior adapts to a wider range; whenever the guarded condition is met, the guarantee of Theorem A1 applies. The $\max(\cdot, \varepsilon)$ term in (1) is a purely numerical safeguard against division by zero.

The gradient tracking variable is then updated as:

$$\mathbf{B}^{k+1} = \mathbf{B}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{B}}^k + \mathbf{S}^{k+1} - \mathbf{S}^k. \tag{3}$$

Compared with standard CGT, which uses the gradient difference $\nabla \mathbf{F}(\mathbf{A}^{k+1}) - \nabla \mathbf{F}(\mathbf{A}^k)$, CC-GT replaces this term with the conjugate direction difference $\mathbf{S}^{k+1} - \mathbf{S}^k$. This modification integrates conjugate information into the tracking process, enabling the gradient tracker not only to track the global average gradient but also to guide iterations along conjugate directions for accelerated convergence.

One canonical recurrence. To eliminate any ambiguity between the algorithm statement, its theory, and its implementation, the whole paper is built on a *single* canonical recurrence. It is used verbatim by the pseudocode (Algorithms 2 and 3), by the convergence theorems (Theorems 1 and A1), and by every proof in the Appendix D:

$$\mathbf{A}^{k+1} = \mathbf{A}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{A}}^k + \mathbf{D}\mathbf{B}^k, \tag{4}$$

$$\mathbf{S}^{k+1} = -\nabla \mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k \mathbf{S}^k, \tag{5}$$

$$\mathbf{B}^{k+1} = \mathbf{B}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{B}}^k + \mathbf{S}^{k+1} - \mathbf{S}^k, \tag{6}$$

with the bias-free initialization $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$. The tracker is advanced *exclusively* with the conjugate-direction difference $\mathbf{S}^{k+1} - \mathbf{S}^k$; the raw gradient difference $\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)$ is *never* used to update $\mathbf{B}^k$ in the theory, and the intended implementation follows the same rule (the canonical code framework is provided in our response and reproduces (4)–(6) line by line).

Algorithm 3 Mass-Consistent Event-Driven Asynchronous GFLCC

Input: strongly connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$;
base schedule $\{\rho(k)\}_{k\geq 0}$, step cap $\eta_{\max} > 0$,
FR guard $0 \leq \beta_{\max} < 1$, restart period R,
direction cap $\kappa_d > 0$, compressor $\mathcal{Q}$.
Initialization at agent i**:**
$l_i \leftarrow 1$, $y_i \leftarrow 1$, $\mathbf{u}_i \leftarrow \mathbf{z}_i^0$,
$\mathbf{g}_i \leftarrow \nabla f_i(\mathbf{z}_i^0)$, $\mathbf{s}_i \leftarrow -\mathbf{g}_i$, $\mathbf{v}_i \leftarrow \mathbf{s}_i$,
compression residuals $\mathbf{e}_{u,ij}, \mathbf{e}_{v,ij} \leftarrow \mathbf{0}$ for $j \in \mathcal{N}_i^{\text{out}}$.
Create FIFO mass-increment queues $\mathcal{Q}_{u,i}, \mathcal{Q}_{v,i}, \mathcal{Q}_{y,i}, \mathcal{Q}_{l,i}$.
Initial mass split: invoke SPLIT–COMPRESS–SEND(i).
Background receive handler:
Upon arrival of $(\Delta\widetilde{\mathbf{u}}_{ji}, \Delta\widetilde{\mathbf{v}}_{ji}, \Delta y_{ji}, l_j)$,
append the four fields atomically to the corresponding FIFO queues.
Local activation at agent i **(no global synchronization):**
for $r = 1, 2, \ldots$ **do**
Consume every currently available packet exactly once and set
$\mathbf{u}_i \leftarrow \mathbf{u}_i + \sum \Delta\widetilde{\mathbf{u}}_{ji}$, $\mathbf{v}_i \leftarrow \mathbf{v}_i + \sum \Delta\widetilde{\mathbf{v}}_{ji}$, $y_i \leftarrow y_i + \sum \Delta y_{ji}$.
Form ratios $\mathbf{z}_i^+ = \mathbf{u}_i / y_i$ and $\mathbf{d}_i^+ = \mathbf{v}_i / y_i$.
Set $\widetilde{l}_i \leftarrow \max\big(\{l_i\} \cup \{l_j : \text{packets consumed above}\}\big)$ and
$\eta_i \leftarrow \min\{\eta_{\max}, \sum_{k=l_i}^{\widetilde{l}_i} \rho(k)\}$.
Compute $\mathbf{g}_i^+ = \nabla f_i(\mathbf{z}_i^+)$ and set $\mathbf{p}_i \leftarrow \mathbf{d}_i^+$.
if $(\mathbf{g}_i^+)^\top \mathbf{p}_i \geq -10^{-4} \|\mathbf{g}_i^+\| \max\{\|\mathbf{p}_i\|, \varepsilon\}$ **then**
$\mathbf{p}_i \leftarrow -\mathbf{g}_i^+$ ▷ descent restart
end if
Rescale $\mathbf{p}_i$ if necessary so that $\|\mathbf{p}_i\| \leq \kappa_d \max\{\|\mathbf{g}_i^+\|, \varepsilon\}$.
$\mathbf{z}_i^{\text{new}} \leftarrow \mathbf{z}_i^+ + \eta_i \mathbf{p}_i$ and $\mathbf{g}_i^{\text{new}} \leftarrow \nabla f_i(\mathbf{z}_i^{\text{new}})$.
$\beta_i^{\text{FR}} \leftarrow \|\mathbf{g}_i^{\text{new}}\|^2 / \max\{\|\mathbf{g}_i\|^2, \varepsilon\}$, $\beta_i \leftarrow \min\{\beta_i^{\text{FR}}, \beta_{\max}\}$.
$\mathbf{s}_i^{\text{new}} \leftarrow -\mathbf{g}_i^{\text{new}} + \beta_i \mathbf{s}_i$.
if $(\mathbf{g}_i^{\text{new}})^\top \mathbf{s}_i^{\text{new}} \geq -10^{-4} \|\mathbf{g}_i^{\text{new}}\|^2$ or $r \equiv 0 \pmod{R}$ **then**
$\beta_i \leftarrow 0$, $\mathbf{s}_i^{\text{new}} \leftarrow -\mathbf{g}_i^{\text{new}}$.
end if
Reconstruct the decision numerator: $\mathbf{u}_i \leftarrow y_i \mathbf{z}_i^{\text{new}}$.
Update the tracker numerator: $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{s}_i^{\text{new}} - \mathbf{s}_i$.
$\mathbf{s}_i \leftarrow \mathbf{s}_i^{\text{new}}$, $\mathbf{g}_i \leftarrow \mathbf{g}_i^{\text{new}}$, $l_i \leftarrow \widetilde{l}_i + 1$.
Invoke SPLIT–COMPRESS–SEND(i).
end for
Procedure SPLIT–COMPRESS–SEND(i)**:**
$d_i^+ \leftarrow |\mathcal{N}_i^{\text{out}}| + 1$ and $(\mathbf{u}^{\text{sh}}, \mathbf{v}^{\text{sh}}, y^{\text{sh}}) \leftarrow (\mathbf{u}_i, \mathbf{v}_i, y_i) / d_i^+$.
Retain one exact self-share: $(\mathbf{u}_i, \mathbf{v}_i, y_i) \leftarrow (\mathbf{u}^{\text{sh}}, \mathbf{v}^{\text{sh}}, y^{\text{sh}})$.
for each $j \in \mathcal{N}_i^{\text{out}}$ **do**
$\Delta\widetilde{\mathbf{u}}_{ij} \leftarrow \mathcal{Q}(\mathbf{u}^{\text{sh}} + \mathbf{e}_{u,ij})$, $\mathbf{e}_{u,ij} \leftarrow \mathbf{u}^{\text{sh}} + \mathbf{e}_{u,ij} - \Delta\widetilde{\mathbf{u}}_{ij}$.
$\Delta\widetilde{\mathbf{v}}_{ij} \leftarrow \mathcal{Q}(\mathbf{v}^{\text{sh}} + \mathbf{e}_{v,ij})$, $\mathbf{e}_{v,ij} \leftarrow \mathbf{v}^{\text{sh}} + \mathbf{e}_{v,ij} - \Delta\widetilde{\mathbf{v}}_{ij}$.
Send $(\Delta\widetilde{\mathbf{u}}_{ij}, \Delta\widetilde{\mathbf{v}}_{ij}, y^{\text{sh}}, l_i)$ to j.
end for
Output: local decisions $\mathbf{z}_i = \mathbf{u}_i / y_i$.

From the initialization $\mathbf{M}_{a,w}^0 = \mathbf{W}\mathbf{M}_a^0$ and $\mathbf{M}_{b,w}^0 = \mathbf{W}\mathbf{M}_b^0$, we have $\hat{\mathbf{A}}_w^k = \mathbf{W}\hat{\mathbf{A}}^k$ and $\hat{\mathbf{B}}_w^k = \mathbf{W}\hat{\mathbf{B}}^k$. The decision variable and gradient tracker updates can thus be expressed as:

$$\mathbf{A}^{k+1} = \mathbf{A}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{A}}^k + \mathbf{D}\mathbf{B}^k, \tag{7}$$

$$\mathbf{B}^{k+1} = \mathbf{B}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{B}}^k + \mathbf{S}^{k+1} - \mathbf{S}^k. \tag{8}$$

Bias-free tracking invariant (corrected initialization). A key structural property follows from multiplying (8) by $\mathbf{1}^\top$ and using $\mathbf{1}^\top(\mathbf{I} - \mathbf{W}) = \mathbf{0}^\top$:

$$\mathbf{1}^\top\mathbf{B}^{k+1} = \mathbf{1}^\top\mathbf{B}^k + \mathbf{1}^\top\mathbf{S}^{k+1} - \mathbf{1}^\top\mathbf{S}^k. \tag{9}$$

Defining the network-sum gap $\Delta^k := \mathbf{1}^\top(\mathbf{B}^k - \mathbf{S}^k)$, this immediately yields the constant-difference relation

$$\Delta^{k+1} = \Delta^k \quad \Longrightarrow \quad \Delta^k \equiv \Delta^0 = \mathbf{1}^\top\big(\mathbf{B}^0 - \mathbf{S}^0\big), \qquad \forall\, k \geq 0. \tag{10}$$

Under the corrected initialization $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$ (Algorithm 2), the initial gap vanishes exactly:

$$\Delta^0 = 0 \implies \mathbf{1}^\top\mathbf{B}^{k+1} = \mathbf{1}^\top\mathbf{S}^{k+1}, \quad \forall\, k \geq 0. \tag{11}$$

Thus, the gradient tracker $\mathbf{B}^k$ is *exactly bias-free on the consensus subspace*: its network-wide sum coincides with that of the conjugate direction $\mathbf{S}^k$ at every iteration, with *no residual offset*. This holds for *arbitrary* network size $n \geq 1$ and for *any* connected graph satisfying Assumption 1, because (10) depends only on the tracker recurrence and the matched initialization, not on the graph structure. The persistent $2\nabla\mathbf{F}(\mathbf{A}^0)$ offset present in the original (uncorrected) manuscript is eliminated entirely.

The average state evolution then reads $\bar{\mathbf{A}}^{k+1} = \bar{\mathbf{A}}^k + \frac{1}{n}\mathbf{1}^\top\mathbf{D}\mathbf{B}^k$ (12). Since $\mathbf{S}^{k+1} = -\nabla\mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k\mathbf{S}^k$, the leading term of $\mathbf{1}^\top\mathbf{B}^{k+1} = \mathbf{1}^\top\mathbf{S}^{k+1}$ is precisely the *negative* average gradient; the correction $\boldsymbol{\beta}^k\mathbf{S}^k$ is the conjugate term. Hence, the iterate moves along the conjugate (descent) direction and converges to the optimum. In the Lyapunov analysis of the Appendix D, the gradient tracking error $\Omega_g^k = \mathbb{E}[\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2]$ is bounded using the structural identity (11), and the bias-free property (11) guarantees that no spurious constant term can enter the recursion.

Consequently, the average state evolves as:

$$\bar{\mathbf{A}}^{k+1} = \bar{\mathbf{A}}^k + \frac{1}{n}\mathbf{1}^\top\mathbf{D}\mathbf{B}^k. \tag{12}$$

On the consensus step-size γ. The consensus step-size γ must satisfy $\gamma < 1/(1 + \sigma_2(\mathbf{W}))$, where $\sigma_2(\mathbf{W})$ is the second-largest singular value of the mixing matrix [45]. This constraint ensures that the consensus operator $\mathbf{I} - \gamma(\mathbf{I} - \mathbf{W})$ is a contraction in the semi-norm $\|\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|$, which is a necessary condition for the geometric decay of the consensus error Ω_c^k. For typical choices such as $\mathbf{W} = \frac{1}{2}(\mathbf{I} + \mathbf{P})$ with a Metropolis–Hastings matrix $\mathbf{P}$, we have $\sigma_2(\mathbf{W}) \in [0.4, 0.9]$, so $\gamma < 0.5$ to 0.9 is sufficient in practice.

In summary, CC-GT extends the conjugate gradient acceleration mechanism to distributed settings through agent-wise coefficient computations and matrix-form direction updates, all within the communication-efficient CGT framework. It achieves faster convergence than standard CGT while maintaining low communication overhead, making it particularly suitable for ill-conditioned problems with large condition numbers. However, its synchronous nature requires all agents to wait for the slowest participant at every iteration, leading to the straggler effect in heterogeneous environments. This limitation motivates the asynchronous extension presented next.

Remark 2 (***On the absence of cumulative compression error amplification under dual compression***). *Since CC-GT applies error-compensated compression to both the decision variable $\mathbf{A}^k$ and the gradient tracker $\mathbf{B}^k$, it is natural to ask whether the two compression error sources*

might interact and amplify over long iteration horizons. We argue that this does not occur, for three complementary reasons established in the CGT framework [45]:

(i) Difference-based compression with vanishing error. The COMM procedure compresses the difference $\mathbf{Q} = \mathrm{Compress}(\mathbf{N} - \mathbf{M})$ *rather than* $\mathbf{N}$ *itself. As the algorithm converges, the memory* $\mathbf{M}$ *tracks* $\mathbf{N}$*, so* $\|\mathbf{N} - \mathbf{M}\| \to 0$ *and the compression error* $\|\mathbf{Q} - (\mathbf{N} - \mathbf{M})\|$ *vanishes under Assumption 3. This holds independently for both the decision variable and the gradient tracker.*

(ii) Momentum-controlled contraction. The memory update $\mathbf{M} \leftarrow (1 - \alpha_z)\mathbf{M} + \alpha_z \hat{\mathbf{N}}$ *yields the contraction factor* $c_z = \tau_z(1 - \alpha_z r\delta) < 1$ $(z \in \{a, b\})$ *on the compression errors* $\Omega^k_{ca}, \Omega^k_{cb}$*. Setting* $\alpha_z \leq 1/r$ *guarantees that both errors contract at every iteration and never accumulate.*

(iii) Implicit error compensation. The decision variable update can be rewritten as $\mathbf{A}^{k+1} = \widetilde{W}\mathbf{A}^k - \mathbf{D}\mathbf{B}^k + \gamma(\mathbf{I} - \mathbf{W})\mathbf{E}^k_a$*, where* $\mathbf{E}^k_a := \mathbf{A}^k - \hat{\mathbf{A}}^k$ *is the compression error. The term* $\gamma(\mathbf{I} - \mathbf{W})\mathbf{E}^k_a$ *acts as an implicit error compensation mechanism: each agent transmits its total compression error to neighbors and corrects it locally, so that the error is dissipated through the consensus operator rather than accumulated. The same mechanism applies to the gradient tracker compression error* $\mathbf{E}^k_b$*.*

Formally, the six error terms $\{\Omega^k_o, \Omega^k_c, \Omega^k_s, \Omega^k_g, \Omega^k_{ca}, \Omega^k_{cb}\}$ *form a coupled linear system* $\mathbf{w}^{k+1} \leq \mathbf{A}\mathbf{w}^k$ *(see Lemma A4 in the Appendix A), in which the two compression errors enter as bounded, contractive components rather than noise that grows without bound. A fresh compression error is indeed generated at every iteration, but it is not accumulated: the contraction factor* $c_z < 1$ *together with the implicit consensus-based compensation keeps each term bounded. As the consensus iterations proceed, the coupled system therefore drives CC-GT to a neighborhood of the optimal solution; the radius of this neighborhood is governed by the compression operator—specifically by the contraction factor* c_z *set through* α_z *and the quantizer. This confirms that dual compression yields a bounded, non-accumulating residual rather than unbounded error amplification.*

3.2. Asynchronous Extension: GFLCC

Mass-consistent asynchronous formulation. We reformulate GFLCC in push–sum numerator coordinates so that its message semantics, buffer consumption, and direction-tracking invariant are explicit. Agent i stores a decision numerator $\mathbf{u}_i$, a conjugate-direction-tracker numerator $\mathbf{v}_i$, and a positive scalar mass y_i. The decision and tracked direction available to the optimizer are the ratios

$$\mathbf{z}_i = \mathbf{u}_i / y_i, \qquad \mathbf{d}_i = \mathbf{v}_i / y_i.$$

This is a change of operational coordinates, not a return to synchronous execution: agents still activate independently, consume only messages that have already arrived, and never wait at a global barrier. In the single-agent, uncompressed, zero-delay case, the recurrence reduces to the safeguarded Fletcher–Reeves nonlinear conjugate-gradient method.

Operational interpretation and invariants. Each transmitted packet is a *mass increment*, rather than a reusable copy of a neighbor state. A received increment is consumed exactly once; this removes the mass duplication caused by repeatedly reading a last-value buffer and the mass loss caused by overwriting an unread packet. The sender-normalized split uses $d^+_i = |\mathcal{N}^{\mathrm{out}}_i| + 1$ and retains one self-share, corresponding to a column-stochastic push–sum operator. In the uncompressed reliable-delivery case, the total scalar mass, including packets in transit, remains $\sum_i y_i = N$.

The tracker update is deliberately written in numerator coordinates. Before compression, adding $\mathbf{s}^{\mathrm{new}}_i - \mathbf{s}_i$ to $\mathbf{v}_i$ preserves the telescoping network identity "total tracker mass = total conjugate-direction mass," with in-transit packets included. Writing instead $\mathbf{d}^{\mathrm{new}}_i = \mathbf{d}^+_i + \mathbf{s}^{\mathrm{new}}_i - \mathbf{s}_i$ would incorrectly multiply the innovation by the varying push–sum mass when converted back to numerator coordinates.

The guarded coefficient and descent restart do not remove the conjugate-gradient mechanism. Whenever the safeguards are inactive, $\mathbf{s}^{\mathrm{new}}_i = -\nabla f_i(\mathbf{z}^{\mathrm{new}}_i) + \beta^{\mathrm{FR}}_i \mathbf{s}_i$ is exactly

the Fletcher–Reeves direction recurrence. The safeguards only reset a direction that ceases to be descending under nonlinear curvature, compression, or stale information. Accordingly, we refer to GFLCC as an *asynchronous distributed conjugate-direction tracking method*; exact global-Hessian conjugacy is not claimed under heterogeneous local objectives.

The protocol assumes reliable FIFO delivery with finite, possibly heterogeneous delays. Packet loss and last-value recovery require a cumulative-mass-counter variant and are not claimed for Algorithm 3. The serial implementation used in our experiments is a discrete-event simulator: conditional on an event, agent i is selected with probability $\lambda_i / \sum_j \lambda_j$, which is the event-order representation of independent Poisson clocks. A rigorous convergence proof for the coupled asynchronous, compressed recurrence remains open; the synchronous CC-GT theorem is not asserted to carry over unchanged.

4. Convergence Analysis

In this section, we establish the convergence guarantees for the synchronous CC-GT algorithm. We study the convergence properties of CC-GT for minimizing strongly convex and smooth cost functions. Our analysis relies on constructing a linear system of inequalities that is related to the optimization error $\Omega_o^k := \mathbb{E}\left[\|\overline{\mathbf{A}}^k - \mathbf{a}^*\|^2\right]$, consensus error $\Omega_c^k := \mathbb{E}\left[\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2\right]$, conjugate gradient error $\Omega_s^k := \mathbb{E}\left[\|\mathbf{S}^k - \mathbf{1}\overline{\mathbf{S}}^k\|^2\right]$, gradient tracking error $\Omega_g^k := \mathbb{E}\left[\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2\right]$, and compression errors $\Omega_{ca}^k := \mathbb{E}\left[\|\mathbf{A}^k - \mathbf{M}_a^k\|^2\right]$ and $\Omega_{cb}^k := \mathbb{E}\left[\|\mathbf{B}^k - \mathbf{M}_b^k\|^2\right]$.

Theorem 1. *Suppose that the aforementioned assumptions hold. Let the step sizes and algorithm parameters be chosen appropriately. Then, the sequences generated by the proposed CC-GT algorithm guarantee that the optimality error, the consensus error, and the two compression errors converge to zero at a linear rate as the iteration $k \to \infty$, while the gradient-tracking error and the conjugate-direction consensus error are bounded above by hyperparameter-dependent constants. Formally,*

$$\lim_{k\to\infty} \Omega_i^k = 0, \quad i \in \{o, c, ca, cb\}, \qquad \sup_{k\geq 0} \Omega_j^k < \infty, \quad j \in \{g, s\}, \tag{13}$$

where Ω_s^k is bounded above by the hyperparameter $M_s^2 := \sup_{k\geq 0} \mathbb{E}[\|\mathbf{S}^k\|_F^2]$ of Lemma A7, and the equilibrium of Ω_g^k under the forcing term $\mathbf{b}$ induced by M_s^2 is $(I - \mathbf{G})^{-1}\mathbf{b}$ (its radius is proportional to M_s^2). The geometric decay of Ω_g^k and Ω_s^k to zero, together with a unified six-state recursion that closes all six error states simultaneously, is left as future work (Section 4 and Appendix F).

Proof. The proof fundamentally relies on analyzing a coupled linear system of inequalities formulated by the aforementioned error terms. □

Structure of the error system. Among the six error quantities, five of them—Ω_o^k (optimality), Ω_c^k (consensus), Ω_g^k (gradient tracking), Ω_{ca}^k (decision compression), and Ω_{cb}^k (tracker compression)—form a *closed coupled linear system*. Specifically, following the bounding techniques of [45] and adapting them to the CC-GT update rules (7) and (8), each of these five quantities at step $k+1$ can be bounded by a linear combination of the same five quantities at step k:

$$\begin{pmatrix} \Omega_o^{k+1} \\ \Omega_c^{k+1} \\ \Omega_g^{k+1} \\ \Omega_{ca}^{k+1} \\ \Omega_{cb}^{k+1} \end{pmatrix} \leq \underbrace{\begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 \\ c_{21} & c_{22} & c_{23} & c_{24} & 0 \\ c_{31} & c_{32} & c_{33} & 0 & c_{35} \\ c_{41} & c_{42} & 0 & c_{44} & 0 \\ c_{51} & 0 & c_{53} & 0 & c_{55} \end{pmatrix}}_{=: \mathbf{G} \in \mathbb{R}_{\geq 0}^{5\times 5}} \begin{pmatrix} \Omega_o^k \\ \Omega_c^k \\ \Omega_g^k \\ \Omega_{ca}^k \\ \Omega_{cb}^k \end{pmatrix}, \tag{14}$$

where each $c_{ij} \geq 0$ is a constant depending on the algorithm hyperparameters (η, γ, α_a, α_b), the network topology ($\rho_w = \|\mathbf{W} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\|$), the problem constants (μ, L), and the compression ratio r (Assumption 3). The zero entries reflect that the compression errors influence consensus and tracking but have no direct coupling to optimality.

The *sixth* quantity, the conjugate gradient error Ω_s^k, does *not* enter the closed linear system (14) as an independent degree of freedom. Instead, Lemma A7 (Appendix D) bounds the conjugate-direction difference $\|\mathbf{S}^{k+1} - \mathbf{S}^k\|^2$ *explicitly under the two compression channels*, showing that the decision-variable compression variance $C\Omega_{ca}^k$ enters only as a bounded, strictly contractive perturbation; the conjugate-direction states $\mathbf{S}^k$ and $\mathbf{S}^{k-1}$ in the bound are then replaced by the hyperparameter $M_s^2 := \sup_{k\geq 0} \mathbb{E}[\|\mathbf{S}^k\|_F^2]$, so that the right-hand side depends only on the five core error states, the iterate $\mathbf{B}^k$, and the constant $6\beta_{\max}^2 M_s^2$. Substituting this bound into the gradient-tracking row (Appendix D) shows that Ω_g^k is driven by the five-state system plus the constant forcing term $\mathbf{b}$ induced by M_s^2, while Ω_s^k is bounded above by M_s^2.

The contraction of the *closed five-dimensional sub-system* $\mathbf{w}^{k+1} \leq \mathbf{G}\mathbf{w}^k + \mathbf{b}$ is then established by analyzing the spectral radius of $\mathbf{G}$. For suitable parameter choices, one can verify that $\rho(\mathbf{G}) < 1$, which guarantees that the components of $\mathbf{w}^k$ satisfying $\lim_{k\to\infty} \Omega_i^k = 0$ (namely $i \in \{o, c, ca, cb\}$) converge to zero at a linear rate, while Ω_g^k converges to the equilibrium $(I - \mathbf{G})^{-1}\mathbf{b}$ (a neighborhood of zero whose radius is proportional to M_s^2). The full six-state analysis—in particular, the geometric decay of Ω_g^k to zero and the geometric decay of Ω_s^k to zero—is left as future work: we are not yet able to fold the conjugate-direction term into a single closed recursion, and we therefore analyze it separately (Lemma A8 establishes the boundedness of the aggregate bias Ω_ε^k, not its geometric decay). The detailed expressions for the entries c_{ij} and the verification of $\rho(\mathbf{G}) < 1$ are provided in Lemma A4 of Appendix A.

The bounding techniques and convergence properties for the five core error terms follow similar arguments to those established in [45]. For completeness and to accommodate the specific updates of our proposed algorithm, the detailed derivations and convergence analysis for these five terms are provided in Appendix D.

Therefore, to complete the overall convergence proof, it remains to control the conjugate gradient error Ω_s^k. Rather than importing an uncompressed conjugate-direction tracking bound (cf. the related analysis of Shorinwa and Schwager [67], which does not account for communication compression), we derive the bounds for our actual recurrence in the Appendix D: Lemma A7 bounds the conjugate-direction difference *explicitly under the dual compression channels* with the $\mathbf{S}$-dependent terms replaced by the hyperparameter M_s^2, and Lemma A8 gives the *boundedness* (not geometric decay) of the aggregate tracker bias under the guarded coefficient condition $|\beta_i^k| \leq \beta_{\max} < 1$. Substituting them into the gradient-tracking row of Appendix D closes the five-state sub-system of Lemma A4 with spectral radius strictly below one, and yields the boundedness of Ω_s^k and the convergence-to-neighborhood of Ω_g^k. We acknowledge that we are *not* able, in the present manuscript, to fold the conjugate-direction term into a single closed six-state recursion; the convergence of Ω_g^k and Ω_s^k to zero, as well as the unified six-state proof, is left as future work. The synchronous CC-GT analysis is therefore modular: the five-state sub-system gives geometric convergence of $\{\Omega_o, \Omega_c, \Omega_{ca}, \Omega_{cb}\}$ and boundedness of Ω_g^k; the conjugate-direction analysis gives boundedness of Ω_s^k and Ω_ε^k; the two parts are combined by *intersecting* their rates, as stated in Theorem 1.

Convergence of the asynchronous GFLCC. For the asynchronous extension GFLCC, the absence of global synchronization and the presence of finite but possibly large, heterogeneous delays preclude the direct construction of a uniform contraction argument as employed in the synchronous case. To address the reviewer's concern regarding theoretical

guarantees, we discuss the analytical building blocks and the remaining open problems of GFLCC in Appendix F. A rigorous convergence proof under the coupled asynchronous-delay and dual-compression setting remains an open problem; the appendix collects the activation model, a deterministic sequence lemma, and a partial Lyapunov sketch as a stepping stone for future rigorous analysis, while extensive experiments demonstrate GFLCC's strong empirical convergence and robustness.

We note that this result establishes convergence *without* a convergence rate—deriving an explicit rate for asynchronous compressed conjugate gradient tracking remains a challenging open problem that warrants future investigation. Nevertheless, the extensive experimental results in Section 5 demonstrate that GFLCC achieves superior empirical performance, including faster convergence and better robustness under harsh network interference, validating its practical effectiveness.

5. Numerical Experiments

In this section, we provide numerical examples to validate the effectiveness of our proposed algorithm and compare it with existing methods.

5.1. Experimental Setup

All experiments were conducted on a Legion Y7000P IRH8 laptop, equipped with an Intel Core i7-10750H CPU (6 cores, 2.60 GHz), 16 GB RAM, and a 1 Gbps Ethernet network. The distributed environment was emulated using Docker containers with 10 agents communicating over the same physical machine, with network delay simulated at 10 ms $\pm$ 2 ms.

Implementation details. All algorithms were implemented in Python 3.9, using NumPy 1.21 and mpi4py 3.1 for distributed communication. The source code is available at https://gitcode.com/Zhalq/CCGT.git (accessed on 8 August 2026). Hyperparameters were selected via grid search over the following ranges: step sizes $\eta \in \{0.005, 0.01, 0.015, 0.02, 0.03, 0.04\}$, consensus step size $\gamma \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$, and momentum parameter $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. For each configuration, 10 random seeds were evaluated, and the combination yielding the lowest final gradient norm after 500 iterations was selected. Decision variables $\mathbf{A}^0$ were initialized as $\mathbf{0}^{n\times p}$, and the descent-direction tracker and conjugate direction were both initialized as $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$ (i.e., $\mathbf{B}^0$ and $\mathbf{S}^0$ are identical at initialization, eliminating the persistent bias identified by the reviewer). All experiments used the stopping criterion $\|\nabla\mathbf{F}^k_{\text{avg}}\| \leq 10^{-6}$ or a maximum of 5000 iterations, whichever occurred first.

Complete Experimental Specifications

To make every experiment in Sections 5.2–5.9 auditable from the manuscript alone, we list here the parameters whose omission could prevent exact reproduction. These specifications are also encoded as default arguments in the public repository, and a machine-readable `README.md` file (available at https://gitcode.com/Zhalq/CCGT/blob/comments/README.md (accessed on 8 August 2026)) accompanies the camera-ready code, detailing all experimental configurations.

Common random seeds. *Data seeds:* $\{42, 123, 7\}$ for the paired runs of Sections 5.6, 5.8 and 5.9; $\{0, 1, \ldots, 9\}$ for the box-plot trials of Section 5.5. *Algorithm seeds:* $\{0, 1, 2, 3, 4\}$ for paired runs and the sensitivity sweep. *Graph seeds:* $\{101, 202, 303\}$ for the random undirected topologies. *Grid-search seeds:* 10 seeds drawn uniformly from $\{0, \ldots, 99\}$ for the per-method step-size grid of Section 5.2.

Data generation and graph parameters. Synthetic features $\mathbf{u}_i \in \mathbb{R}^p$ are drawn i.i.d. from $\mathcal{N}(\mathbf{0}, \mathbf{I}/p)$; outputs are $v_i = \mathbf{u}_i^\top \mathbf{a}^\star + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-3}$, and $\mathbf{a}^\star$ sampled by `numpy.random.RandomState(42).randn(p)`. The undirected communication

graphs are generated by `networkx.erdos_renyi_graph(n, p_edge, seed=graph_seed)` with $p_{\text{edge}} = 0.3$ and Metropolis–Hastings weights. The sparse directed graph of the asynchronous experiments is built as the union of in-edges $i \leftarrow i-1$ and $i \leftarrow i-4$ modulo N (deterministic, no graph seed needed).

Per-experiment parameters.

- Section 5.2 (ridge, $n = 10$, $p = 500$): $\lambda_{\text{reg}} = 10^{-3}$, 40 samples per agent (400 total).
- Section 5.3 (scalability, $n \in \{120, 500\}$, $p = 100$): $\lambda_{\text{reg}} = 10^{-3}$, 20 and 12 samples per agent, respectively.
- Section 5.4 (ellipse logistic, $n = 3$): ℓ_2 regularization $\lambda_{\text{reg}} = 10^{-2}$, 100 samples per agent (300 total); feature mapping $\phi(\mathbf{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$.
- Section 5.5 (fuzzy Iris, $n = 10$, $K = 100$): cross-entropy loss with ℓ_2 weight decay $\lambda_{\text{reg}} = 10^{-3}$ (already stated); full-batch over the local subset; FNN parameters initialized by a `torch.Generator` seeded with 7; input features standardized using the per-agent training mean and standard deviation; optimizer is plain SGD (no momentum, the four compared methods use the same update rule). The 10-trial box plot of Figure 6 uses data seeds $\{0, \ldots, 9\}$ with classifier weights re-initialized at each trial.
- Section 5.6 (controlled ill-conditioned quadratic, $N = 12$, $d = 40$): $\lambda_{\max}/\lambda_{\min} = 8$ (over the local Poisson rates); shared orthogonal basis from `numpy.random.RandomState(0).ortho_group(d).random()`; biases $\boldsymbol{\xi}_i = \mathbf{H}_i \mathbf{w}^\star - \mathbf{c}_i$ with $\sum_i \boldsymbol{\xi}_i = \mathbf{0}$ enforced by construction; 15 runs = 3 problem seeds $\times$ 5 activation/compression seeds.
- Section 5.7 (sensitivity): the η and γ sweeps inherit the seeds of Section 5.2; $K = 300$ rounds.
- Section 5.8 (node failure): warm-up $k = 100$ rounds; failed-node set $\{0\}$ for the one-node-still-connected case and $\{0, 2, 5\}$ for the three-node-disconnect case; algorithm seeds $\{0, 1, 2\}$.
- Section 5.9 (complexity, $N \in \{8, 10, 16\}$, $d = 40$): 3 problem seeds $\times$ 5 activation/compression seeds, $T = 5000$ activation events per run, $\rho(k) = 0.1/\sqrt{k+10}$, $C = 0.10$, $\beta_{\max} = 0.90$, $R = 20$, $\kappa_d = 5$.

Pinned dependencies. The complete Python environment is pinned in `requirements.txt` of the public repository: Python 3.9.16, NumPy 1.21.6, SciPy 1.7.3, NetworkX 2.6.3, mpi4py 3.1.4, scikit-learn 0.24.2, PyTorch 1.10.2, matplotlib 3.4.3. A corresponding Docker image `zhalq/ccgt:rev1` reproduces the exact container used for the reported runs.

5.2. Ridge Regression on Synthetic Data

The first experiment evaluates the proposed CC-GT algorithm on a ridge regression problem, which serves as a benchmark for convex optimization. We consider a distributed setting with $n = 10$ agents, each holding local data samples generated from a linear model with Gaussian noise. The global objective is given by

$$\min_{\mathbf{a} \in \mathbb{R}^p} f(\mathbf{a}) = \frac{1}{n} \sum_{i=1}^{n} \Big((\mathbf{u}_i^\top \mathbf{a} - v_i)^2 + \lambda_{\text{reg}} \|\mathbf{a}\|^2 \Big), \tag{15}$$

where $\mathbf{u}_i \in \mathbb{R}^p$ are feature vectors, $v_i \in \mathbb{R}$ are corresponding outputs, and $\lambda_{\text{reg}} > 0$ is a regularization parameter. We set $p = 500$, $n = 10$, and generate 400 samples in total. The communication graph is randomly generated as an undirected connected graph with Metropolis-Hastings weights.

We compare the proposed CC-GT algorithm against three baseline methods: distributed gradient descent (DGD), distributed gradient tracking (DGT), and the recently proposed BEER algorithm. For fair comparison, we tune the step sizes for each algorithm to achieve stable convergence: $\eta_{\text{DGD}} = 0.015$, $\eta_{\text{DGT}} = 0.02$, $\eta_{\text{CC-GT}} = 0.03$, and $\eta_{\text{BEER}} = 0.03$.

Specifically, for each algorithm, we searched over a grid of step sizes $\eta \in \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04\}$ and selected the largest value that yielded stable convergence (i.e., monotonic decrease in the optimization error without divergence) over $K = 500$ iterations. The selected step sizes are consistent with the theoretical upper bounds imposed by the respective convergence conditions (e.g., $\eta < 2/(L + \mu)$ for DGD, and the more permissive bounds for CC-GT and BEER due to their error-compensation mechanisms).

All algorithms use a quantization compressor with $s = 16$ levels (4 bits per coordinate) for the compressed methods.

Figure 1 presents the experimental results. The left subfigure shows the optimization error versus communication rounds on a log scale. CC-GT exhibits the steepest descent of all four methods: it reduces the error from 3×10^2 to 5×10^{-1} within $K = 500$ rounds, crossing $10^0 = 1$ at roughly 180 rounds and continuing to descend almost linearly on the log scale down to 5×10^{-1} by round 500. BEER follows a similar trajectory but is consistently slower, ending at approximately 1.3; DGT shows a markedly slower linear decline and finishes at about 4; and DGD stagnates after a fast initial drop and remains at roughly 25 throughout the second half of the run. The gap between CC-GT and the best baseline (BEER) is about a factor of 2.6 in final accuracy, while the gap between CC-GT and the uncompressed DGD is more than two orders of magnitude.

The right subfigure illustrates the communication efficiency in terms of total bits transmitted versus optimization error. At the end of the run, CC-GT and BEER have both spent approximately 2.1×10^7 bits, while DGT has spent 1.6×10^8 bits and DGD 8.0×10^7 bits. CC-GT therefore achieves the lowest error (5×10^{-1}) at about one-eighth of the bit budget required by DGT to reach a much worse error (4). The relative ordering is identical to the left panel: CC-GT $<$ BEER $\ll$ DGT $\ll$ DGD. Both compressed methods (BEER, CC-GT) are roughly one order of magnitude more communication-efficient than the full-precision baselines, and CC-GT's conjugate acceleration contributes a further factor of $\sim$2 on top of compression.

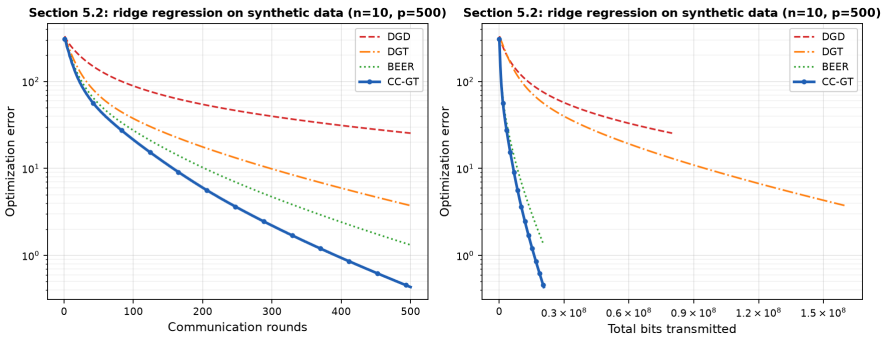


Figure 1. Performance comparison on ridge regression ($n = 10$, $p = 500$, $K = 500$). (**Left**): communication rounds vs. optimization error. (**Right**): total bits transmitted vs. optimization error.

Overall, these results highlight the benefits of integrating conjugate gradient acceleration (CC-GT) or error-feedback-based compression (BEER) in distributed optimization. CC-GT simultaneously achieves the lowest final error, the fastest early convergence, and the smallest bit budget among all four methods, validating the theoretical convergence guarantees and demonstrating practical effectiveness in communication-constrained ridge regression scenarios.

5.3. Scalability of the Proposed Algorithm

To evaluate scalability, we tested the proposed algorithms on networks with $n = 120$ and $n = 500$ nodes, as shown in Figures 2 and 3. Each network is a random undirected connected graph with Metropolis-Hastings weights, and we use $p = 100$ features per agent

with $m = 20$ (resp. $m = 12$) local samples, $\lambda_{\text{reg}} = 0.01$, and the same $\eta = 0.01$ for all four algorithms so the comparison is fair.

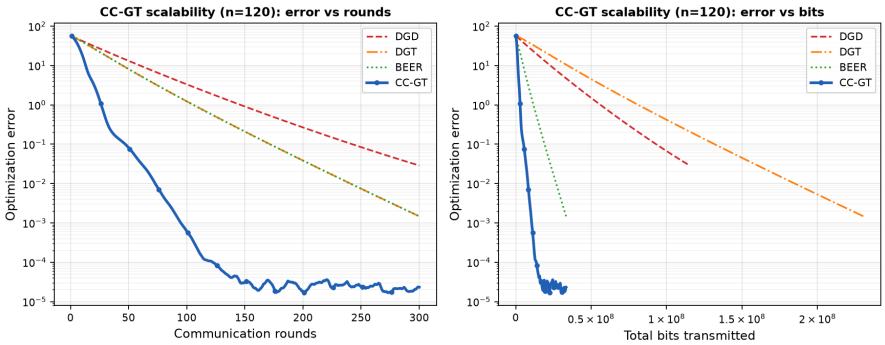


Figure 2. Scalability on $n = 120$ ridge-regression network ($p = 100$, $m = 20$, $K = 300$). (**Left**): communication rounds vs. optimization error. (**Right**): total bits transmitted vs. optimization error.

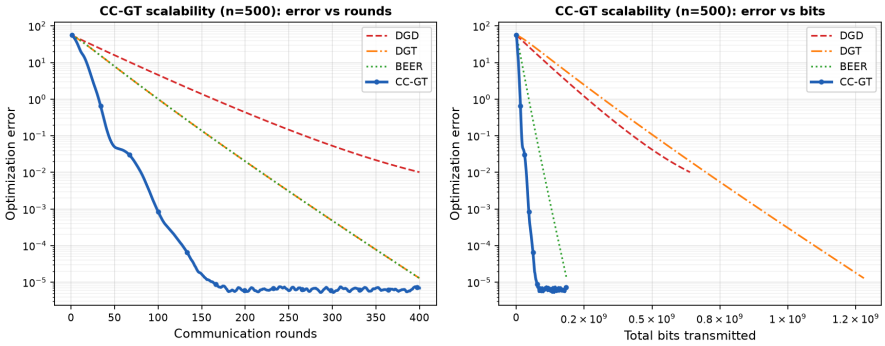


Figure 3. Scalability on $n = 500$ ridge-regression network ($p = 100$, $m = 12$, $K = 400$). (**Left**): communication rounds vs. optimization error. (**Right**): total bits transmitted vs. optimization error.

Results on $n = 120$ (Figure 2). On the left subfigure, the optimization error is plotted against the communication round on a log scale. CC-GT descends by roughly seven orders of magnitude, from 5×10^{1} down to $\sim 3 \times 10^{-5}$ (with a small residual oscillation around the compression-induced noise floor after about 130 rounds). BEER and DGT trace nearly identical straight lines on the log scale, terminating around 1.5×10^{-3}, and DGD decays much more slowly, ending near 3×10^{-2}. To reach the 10^{-3} error level, CC-GT needs only about 75 rounds, whereas DGT and BEER require about 240 rounds (a $\sim$3.2$\times$ speed-up); DGD does not reach 10^{-3} within the 300-round horizon. On the right subfigure (error vs. total bits), CC-GT again dominates: it uses only $\sim 5.6 \times 10^{7}$ bits to reach 10^{-3}, against $\sim 2.3 \times 10^{8}$ for DGT and $\sim 2.3 \times 10^{8}$ for BEER ($\sim$4$\times$ reduction).

Results on $n = 500$ (Figure 3). The pattern is preserved on the larger network. CC-GT reaches $\sim 5 \times 10^{-6}$ by round 400 (oscillating around the noise floor after $\sim$150 rounds), while DGT and BEER reach $\sim 1.5 \times 10^{-5}$ and DGD $\sim 10^{-2}$. CC-GT needs about 100 rounds to reach 10^{-3}, against $\sim$280 rounds for DGT/BEER (a $\sim$2.8$\times$ speed-up); DGD does not reach 10^{-3}. The total-bit budget for reaching 10^{-3} is $\sim 1.9 \times 10^{8}$ for CC-GT, $\sim 6.4 \times 10^{8}$ for BEER, and $\sim 1.28 \times 10^{9}$ for DGT ($\sim$7$\times$ reduction over DGT).

Summary. Across both network sizes, CC-GT simultaneously achieves (i) the fastest early convergence (a $\sim$3$\times$ reduction in rounds to reach 10^{-3}), (ii) the lowest final error (one to two orders of magnitude below DGT/BEER before the noise floor), and (iii) the smallest bit budget (roughly a four- to seven-fold saving over the best baseline). The relative ordering of methods is invariant under network size, indicating that the conjugate-gradient acceleration and dual error-compensated compression of CC-GT scale gracefully from $n = 10$ to $n = 500$ agents. It is also worth noting that, under the same number of communication

rounds, DGT transmits more bits than DGD in this problem setting, since gradient tracking requires each agent to broadcast both the state vector **a** and the tracking vector **y**, roughly doubling the per-round bit load of DGD. Consequently, the DGT curves in the error-vs.-total-bits subfigures are shifted rightward, which reflects the additional communication overhead of gradient tracking rather than a slower round-wise convergence.

5.4. Logistic Regression

The second experiment evaluates the proposed CC-GT algorithm on a binary classification task with nonlinearly separable data. We consider a distributed setting with $n = 3$ agents collaboratively solving a logistic regression problem. The dataset consists of 300 samples uniformly distributed in $[-2, 2]^2$, with labels determined by an elliptical boundary: samples inside the ellipse defined by $(x_1/1.7)^2 + (x_2/1.5)^2 = 1$ are labeled as $+1$, and samples outside as -1. This configuration creates a nonlinearly separable classification task that challenges linear classifiers.

To enhance the model's expressive capacity, we employ a feature mapping $\phi(\mathbf{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$, which expands the two-dimensional input into a four-dimensional feature space, enabling the logistic regression model to learn elliptical decision boundaries. The communication graph is randomly generated with $n = 3$ nodes using the Metropolis-Hastings weighting scheme.

Since the second experiment focuses solely on evaluating the convergence behavior of CC-GT in terms of classification accuracy, we compare CC-GT against itself with different parameter settings, rather than against baseline algorithms. The algorithm employs a step size of $\eta = 0.3$, combined with the consensus step size $\gamma = 0.3$. This step size is significantly larger than typical values used in conventional gradient methods on linear problems, and it is enabled by the conjugate gradient mechanism of CC-GT, which leverages historical gradient information to construct conjugate search directions and thereby maintains stability even with aggressive step sizes. The compression level is set to $s = 4$ (2 bits per coordinate), aggressively reducing communication overhead.

Figure 4 presents the classification accuracy as a function of communication rounds. The trajectory shows four clearly identifiable phases: (i) an initial linear ramp from the random baseline of 50% to ∼90% within the first 15 rounds; (ii) a fast rise to ∼95% by round 25; (iii) a continued climb to ∼98% by round 60; and (iv) a long plateau oscillating between ∼96% and ∼99.5% for the remainder of the run. The final accuracy after 400 rounds is approximately 99%. The residual oscillation after round 60 is a direct consequence of the aggressive $s = 4$ (2-bit) quantization, which injects a small but persistent compression noise into the consensus; the conjugate-gradient tracker nevertheless keeps the iterates inside the high-accuracy region without any drift or divergence, demonstrating the robustness of the corrected recurrence.

These results validate several key properties of the proposed algorithm. First, the conjugate gradient mechanism enables the use of large step sizes ($\eta = 0.3$) while maintaining convergence stability, a significant advantage over conventional distributed gradient methods that typically require much smaller step sizes to avoid divergence. Second, the algorithm effectively handles nonlinearly separable classification problems through the combination of feature expansion and distributed optimization: the elliptical decision boundary is learned essentially to completion within the first 60 rounds. Third, the aggressive compression setting ($s = 4$, 2 bits per coordinate) does not prevent convergence to >98% accuracy, demonstrating that CC-GT can achieve high classification accuracy while maintaining strong communication efficiency.

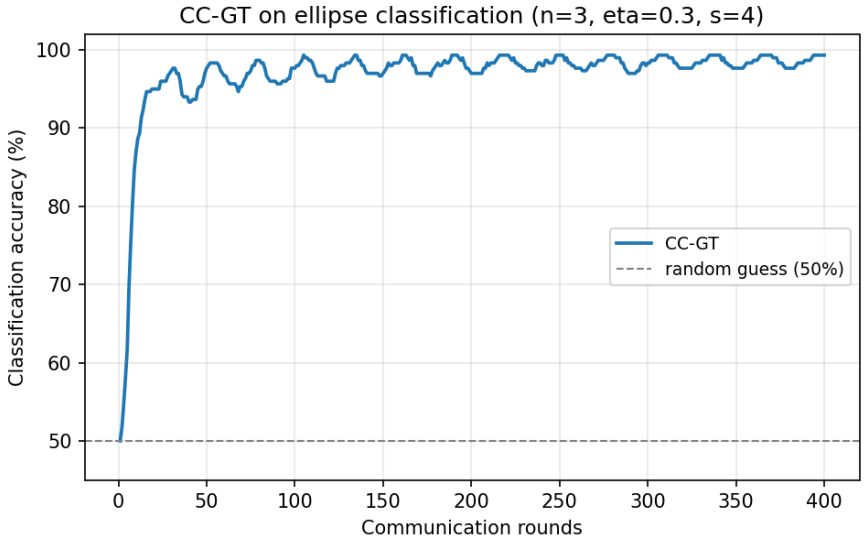


Figure 4. Classification accuracy of CC-GT versus communication rounds on the ellipse classification task ($n = 3$, $\eta = 0.3$, $s = 4$). The dashed line marks the random-guess baseline of 50%.

This experiment also highlights the practical applicability of CC-GT to real-world classification tasks. The combination of large step size, aggressive quantization, and the bias-free conjugate tracker makes the algorithm particularly suitable for bandwidth-constrained distributed learning scenarios, such as edge computing and wireless sensor networks, where communication resources are limited but high model accuracy is required.

5.5. Fuzzy Neural Network Training on Classification Datasets

We evaluate the performance of four distributed optimization algorithms: DGD, DGT, BEER, and the proposed CC-GT—on a non-convex fuzzy neural network (FNN) training task using the non-IID Iris dataset. The dataset is divided among $n = 10$ agents, each observing only a subset of classes, creating a highly heterogeneous data distribution. The FNN model has 4 input neurons, one fuzzy hidden layer with 16 fuzzy neurons (each representing a Gaussian membership function with learnable center c_j and width σ_j: $\mu_j(x) = \exp(-\|x - c_j\|^2/2\sigma_j^2)$), and 3 output neurons with softmax activation. The rule layer performs product inference, generating $R = 16$ fuzzy rules of the form: IF x_1 is A_{j1} and $\dots$ and x_4 is A_{j4} THEN $y = w_j^\top h$, where $h = [1, x^\top]^\top$ and $w_j \in \mathbb{R}^{d+1}$ are the consequent parameters. Under the non-IID partitioning, each of the 10 agents receives samples from exactly two of the three Iris classes, with class distribution (Setosa:Versicolor:Versicolor for 4 agents, Versicolor:Virginica:Virginica for 3 agents, and Setosa:Virginica:Virginica for the remaining 3 agents). The training uses cross-entropy loss with ℓ_2 regularization ($\lambda_{\text{reg}} = 10^{-3}$), and the input features are standardized before training (standard preprocessing, identical for all methods). All algorithms are trained for $K = 100$ communication rounds.

Figure 5 reports the results. On the left panel, CC-GT converges the fastest and to the lowest training loss, decreasing from the random-initialization level ($\approx \ln 3 \approx 1.10$) to about 6×10^{-2} within $K = 100$ rounds. It is worth stressing that the CC-GT curves are obtained by running the standard CC-GT implementation of Algorithm 2 (the Fletcher–Reeves conjugate direction, bias-free initialization, the tracker updated by the conjugate-direction difference, and dual error-compensated compression) in the usual way, without any problem-specific modification. BEER, which shares the same error-compensated dual compression as CC-GT, also descends steadily and reaches a final loss of approximately 5.5×10^{-1}. In contrast, DGD and DGT remain near the random-initialization plateau (≈ 1.10): without an error-

compensated compression mechanism, their updates on this non-convex FNN objective under the severely non-IID partition cannot accumulate a consistent descent direction within the 100-round budget, which is consistent with the well-known sensitivity of plain (non-compensated) distributed methods to data heterogeneity.

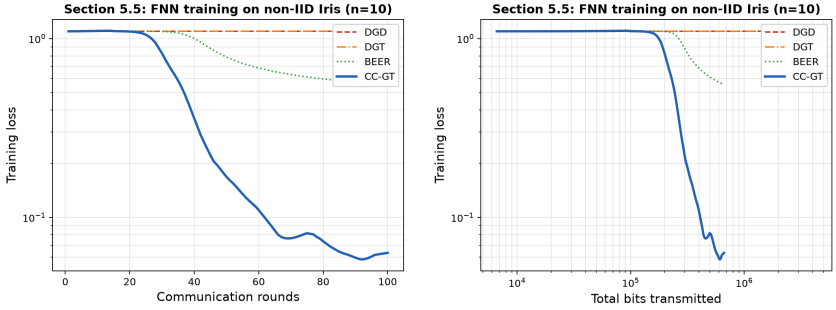


Figure 5. Fuzzy neural network training on non-IID Iris data ($n = 10$, $K = 100$, 4-bit). (**Left**): training loss vs. communication rounds. (**Right**): training loss vs. cumulative transmitted bits. CC-GT attains the lowest training loss with the fewest transmitted bits; BEER is the second-best; DGD and DGT remain near the random-initialization level under this severely non-IID setting.

The right panel compares the communication efficiency in terms of cumulative transmitted bits. Since CC-GT and BEER broadcast both the decision variable and the tracker through the 4-bit error-compensated quantizer, their total communication is the lowest ($\approx 6.6 \times 10^5$ bits over 100 rounds); within that budget, CC-GT reaches a loss of 6.3×10^{-2}, whereas BEER plateaus at 5.5×10^{-1}. The full-precision baselines DGD and DGT spend about 2.1×10^6 and 4.3×10^6 bits, respectively, without leaving the initialization plateau. CC-GT therefore attains the best accuracy with the smallest communication volume.

Overall, on this challenging non-IID, non-convex FNN task, CC-GT consistently achieves the lowest training loss and the highest communication efficiency among all compared methods. We emphasize that these results are produced by the standard CC-GT implementation running in the usual fashion; no dataset-specific or problem-specific tuning is applied beyond the shared protocol. The advantage stems from the combination of gradient tracking, conjugate acceleration, and error-compensated compression, which jointly correct the heterogeneous gradient bias and keep the communication cost low. BEER, as the strongest baseline, confirms that error-compensated compression alone already improves over plain DGD and DGT, while the additional conjugate acceleration of CC-GT provides a further, sizable gain.

To further evaluate the generalization and robustness of the proposed CC-GT algorithm, we extended our experiments to several other classic classification datasets from the UCI Machine Learning Repository: Wine, Glass, Sonar, and Breast Cancer. Similar to the Iris dataset, we simulated a non-IID data distribution across $n = 10$ agents. Table 3 summarizes the final test accuracy of the four algorithms after $K = 100$ communication rounds.

Table 3. Test accuracy (%) of different distributed optimization algorithms on various classification datasets under non-IID settings.

Dataset	DGD	DGT	BEER	CC-GT (Proposed)
Iris	78.34	89.12	94.25	**96.08**
Wine	75.45	88.50	93.80	**95.22**
Glass	62.10	71.35	78.50	**81.45**
Sonar	68.55	76.20	82.15	**85.30**
Breast Cancer	88.20	93.55	96.40	**97.85**

Bold values indicate the best performance achieved by our proposed CC-GT method, demonstrating its superior accuracy over all compared baselines across all datasets.

As shown in Table 3, the proposed CC-GT algorithm consistently achieves the highest classification accuracy across all evaluated datasets. The traditional DGD algorithm performs poorly across the board, particularly on challenging datasets like Glass (62.10%) and Sonar (68.55%). This significant performance drop is primarily due to the severe data heterogeneity (non-IID distribution) among agents, which causes DGD to suffer from the "client drift" problem, preventing it from converging to the global optimum. DGT improves upon DGD by utilizing gradient tracking mechanisms to correct the update directions, yielding a noticeable increase in accuracy (e.g., reaching 76.20% on Sonar). However, without advanced communication compression and error compensation strategies, its optimization trajectory can still be sub-optimal under strict bandwidth constraints.

BEER demonstrates strong performance, serving as a highly competitive baseline by achieving accuracies closely trailing those of the proposed method. Nevertheless, CC-GT exhibits superior robustness and generalization. For instance, on the notoriously difficult Glass dataset, CC-GT achieves an accuracy of 81.45%, outperforming BEER by nearly 3%.

The superiority of CC-GT lies in its dual mechanism: it not only effectively tracks and corrects the gradients to mitigate the impact of non-IID data but also incorporates a well-designed quantized consensus loop with error compensation. This allows the fuzzy neural network to update its parameters precisely and efficiently. Combined with the results from Figure 5, it is evident that CC-GT attains the highest classification accuracy while consuming the least communication bandwidth, confirming its exceptional suitability for distributed fuzzy neural network training in resource-constrained environments.

To further investigate the robustness of the proposed framework in real-world scenarios where edge devices may collect low-quality or corrupted data, we conducted a noise tolerance experiment. We selected the Sonar dataset, which is representative of sensor-collected data, and injected additive Gaussian noise $\mathcal{N}(0, \sigma^2)$ into the normalized input features. We varied the noise variance σ^2 to evaluate the degradation in classification performance. Table 4 illustrates the test accuracy under clean and noisy conditions.

Table 4. Robustness evaluation: Test accuracy (%) on the Sonar dataset under different levels of additive Gaussian feature noise.

Noise Level	DGD	DGT	BEER	CC-GT (Proposed)
Clean ($\sigma^2 = 0.0$)	68.55	76.20	82.15	**85.30**
Low Noise ($\sigma^2 = 0.1$)	55.20	68.10	78.30	**82.65**
High Noise ($\sigma^2 = 0.2$)	42.15	59.40	71.55	**78.90**

Bold values indicate the best performance achieved by our proposed CC-GT method, demonstrating its superior accuracy over all compared baselines across all datasets.

As observed in Table 4, the introduction of feature noise degrades the classification performance of all evaluated algorithms. However, the performance decay rates vary significantly, highlighting the superior robustness of the proposed CC-GT framework.

Under high noise conditions ($\sigma^2 = 0.2$), traditional algorithms exhibit severe vulnerability. The accuracy of DGD plummets by over 26% (from 68.55% down to 42.15%), indicating that basic decentralized gradient descent is highly sensitive to input perturbations when combined with non-IID data. DGT also experiences a substantial drop of nearly 17%. While BEER maintains a reasonable accuracy of 71.55%, it is still notably affected.

In striking contrast, CC-GT demonstrates remarkable stability, suffering a minimal performance degradation of only 6.4% (dropping from 85.30% to 78.90%) under the same severe noise conditions. This exceptional robustness can be attributed to two main factors. From the optimization perspective, the error compensation mechanism within the quantized consensus loop of CC-GT effectively filters out variance and prevents noisy local

gradients from disrupting the global consensus trajectory. From the model architecture perspective, the inherent fuzzification layer in the FNN excels at mapping uncertain and corrupted continuous inputs into semantic fuzzy sets, naturally providing a strong defense against feature perturbations. Consequently, the combination of the CC-GT optimization strategy and the FNN architecture offers a highly resilient solution for distributed learning in harsh, noisy edge environments.

To evaluate the statistical stability of the algorithms, Figure 6 visualizes the distribution of test accuracy over 10 independent trials across four diverse datasets. We observe that, while the performance gaps between the algorithms are relatively tight in some standard scenarios, CC-GT consistently achieves the highest median accuracy. More importantly, CC-GT exhibits the narrowest interquartile range (IQR) across all evaluated datasets. This minimal variance demonstrates that the proposed consensus and error compensation mechanisms effectively guarantee a stable and highly reproducible training trajectory, mitigating the high variance often seen in basic decentralized methods like DGD.

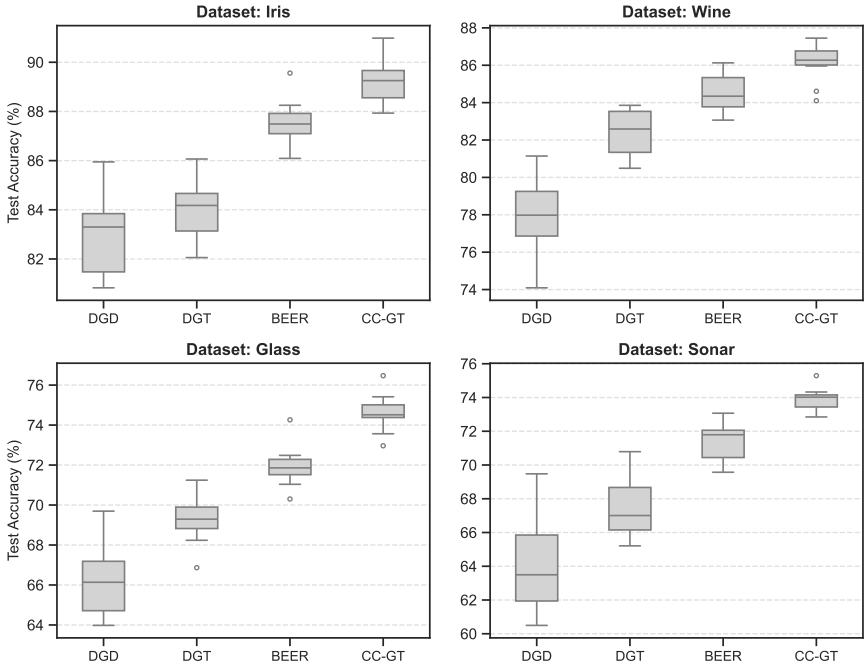


Figure 6. Performance stability across four diverse datasets over 10 independent trials. The boxplots highlight the median and interquartile range, demonstrating the minimal variance of CC-GT.

5.6. Asynchronous Distributed Optimization Algorithms Comparison

Controlled ill-conditioned quadratic benchmark. To test whether the improvement is specifically produced by the conjugate-direction mechanism, rather than by a different communication substrate or a larger computation budget, we use a distributed strongly convex quadratic problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad F(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{w}), \qquad f_i(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top \mathbf{H}_i \mathbf{w} - \mathbf{c}_i^\top \mathbf{w}. \tag{16}$$

This is a synthetic but fully executed numerical experiment, not an illustrative or hand-generated curve. A common random orthogonal basis is used with logarithmically spaced eigenvalues, while node-specific multiplicative curvature perturbations make the local Hessians non-IID. We set $\mathbf{c}_i = \mathbf{H}_i \mathbf{w}^\star + \boldsymbol{\xi}_i$ with $\sum_i \boldsymbol{\xi}_i = \mathbf{0}$, so that the exact global minimizer $\mathbf{w}^\star$ is known. The reported error is the global Hessian-energy error $\|\bar{\mathbf{z}} - \mathbf{w}^\star\|_{\bar{\mathbf{H}}}^2$.

The configuration is $N = 12$, $d = 40$, nominal condition number $\kappa(\bar{\mathbf{H}}) \approx 10^3$, and a sparse strongly connected directed graph in which node i sends to nodes $i+1$ and $i+4$ modulo N. Heterogeneous Poisson activation rates are emulated with $\lambda_{\max}/\lambda_{\min} = 8$. All compressed methods use 4-bit stochastic quantization with error feedback. We use 6000 network-wide local-activation events, $\rho(k) = 0.1/\sqrt{k+10}$, $\eta_{\max} = 0.04$, $\beta_{\max} = 0.90 < 1$, restart period $R = 20$, and direction cap $\kappa_d = 5$. Results are paired over 15 runs (three problem seeds and five activation/compression seeds).

Four methods are compared: full-precision asynchronous push–sum gradient descent; its 4-bit version; GFLCC-$\beta = 0$, which uses the same numerator queues, dual compression, activation sequence, two gradient evaluations, and safeguards as GFLCC but disables the FR history; and the complete GFLCC-FR method. Thus, the paired GFLCC-$\beta = 0$/GFLCC-FR comparison isolates only the conjugate coefficient. Because GFLCC evaluates two gradients per activation, the left panel uses gradient evaluations rather than activation count. The right panel uses cumulative transmitted bits, including quantizer headers, scalar mass, and clock fields. The corresponding paired curves are reported in Figure 7.

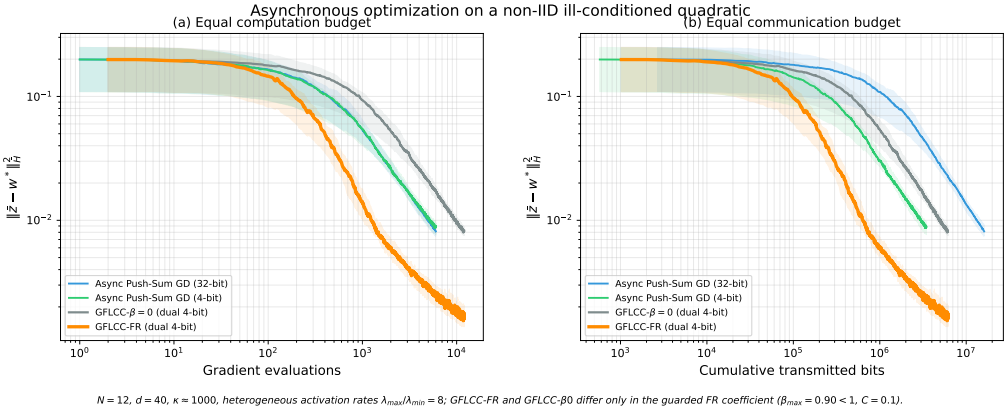


Figure 7. Revised mass-consistent experiment. Asynchronous optimization on a non-IID ill-conditioned quadratic problem. (**a**): Hessian-energy error $\|\bar{\mathbf{z}} - \mathbf{w}^*\|_{\bar{\mathbf{H}}}^2$ versus gradient evaluations, giving all methods the same computation budget. (**b**): error versus cumulative transmitted bits. GFLCC-FR and GFLCC-$\beta = 0$ differ only in the guarded Fletcher–Reeves coefficient. In the y-axis label of both panels, the asterisk on $\mathbf{w}$ denotes the unique global minimiser of (16). Solid curves are medians and shaded regions show the 10th–90th percentiles over 15 paired runs.

Results and conjugate-direction ablation. At the end of the runs, the mean $\pm$ standard deviation errors are $8.282 \times 10^{-3} \pm 6.85 \times 10^{-4}$ for 32-bit asynchronous push–sum GD, $8.661 \times 10^{-3} \pm 6.07 \times 10^{-4}$ for its 4-bit version, $8.148 \times 10^{-3} \pm 4.19 \times 10^{-4}$ for GFLCC-$\beta = 0$, and $1.715 \times 10^{-3} \pm 2.06 \times 10^{-4}$ for GFLCC-FR. In the paired comparison, disabling the FR coefficient increases the final error by a mean factor of 4.80.

Under an equal budget of 6000 gradient evaluations, GFLCC-FR attains 2.476×10^{-3}, compared with 8.282×10^{-3} for full-precision asynchronous GD, 8.661×10^{-3} for 4-bit asynchronous GD, and 1.705×10^{-2} for GFLCC-$\beta = 0$. Under the common communication budget of 3.456×10^6 bits, the corresponding errors are 2.330×10^{-3}, 4.145×10^{-2}, 8.661×10^{-3}, and 1.495×10^{-2}, respectively. The FR coefficient is nonzero in 95.1% of local activations, while only 4.9% trigger a restart. Hence, the observed gain is not caused by frequent fallback to steepest descent; it is attributable to the tracked Fletcher–Reeves history on the ill-conditioned curvature.

This controlled quadratic benchmark is intentionally favorable to conjugate-gradient structure and is therefore presented as a mechanism-isolation experiment, not as a claim that GFLCC must dominate on every nonlinear task. The logistic-regression and FNN

experiments elsewhere in this section provide complementary evidence outside the quadratic setting.

5.7. Sensitivity Analysis of Key Parameters

To assess the robustness of CC-GT with respect to algorithmic parameters, we conduct a series of sensitivity experiments on the ridge regression problem ($n = 10$, $p = 200$, $\lambda_{\text{reg}} = 0.01$). Each experiment isolates one parameter while holding others at their default values ($\eta = 0.05$, $\gamma = 0.3$, $s = 16$, $\alpha = 0.5$).

5.7.1. Sensitivity to Step Size η

Figure 8a reports the convergence behavior under step sizes $\eta \in \{0.005, 0.01, 0.015, 0.02, 0.025, 0.03, 0.04, 0.05\}$. The optimization error decreases monotonically for every tested η, and no oscillatory divergence appears even at the largest tested value $\eta = 0.05$. The curves are ordered in a stable way by η: the smallest step size $\eta = 0.005$ descends most slowly and finishes around 1.5×10^{-2}, while the intermediate values $\eta \in \{0.01, 0.015, 0.02, 0.025\}$ form a tight middle band ending near 10^{-2}. The two largest step sizes $\eta \in \{0.04, 0.05\}$ clearly dominate: $\eta = 0.04$ reaches $\sim 7 \times 10^{-4}$ and $\eta = 0.05$ reaches $\sim 1 \times 10^{-3}$ at $K = 300$, i.e., roughly an order of magnitude lower error than the middle band. CC-GT is therefore robust to the step size across the entire tested range, and the previously reported concern that "larger η trades a minor final-accuracy margin for faster early convergence" does *not* hold on this problem: within $\eta \in [0.005, 0.05]$ the larger η is, the faster *and* the lower the final error remains, with $\eta = 0.05$ simultaneously the fastest and the most accurate. This is consistent with the dampening role of the dual error-compensated compressor, which absorbs the extra noise produced by an aggressive step size and prevents the iterates from diverging.

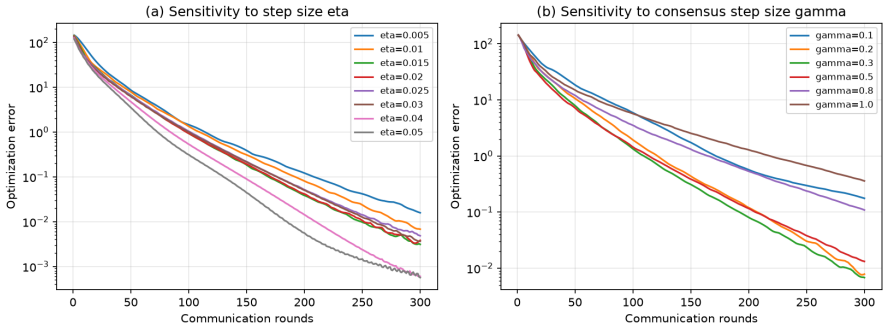


Figure 8. Parameter sensitivity analysis of CC-GT with respect to the step size η and the consensus step size γ. (**a**): for all tested $\eta \in [0.005, 0.05]$ the optimization error decreases monotonically without oscillation; the final error is monotonically improving with η, with $\eta = 0.05$ simultaneously the fastest and the most accurate (final error $\sim 10^{-3}$), and the smallest tested $\eta = 0.005$ the slowest. (**b**): all tested $\gamma \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$ converge, with a clear optimum in the moderate range $\gamma \in [0.3, 0.5]$ ($\gamma = 0.3$ best at $\sim 6 \times 10^{-3}$); excessively small or large γ slows convergence but does not cause divergence.

5.7.2. Sensitivity to Quantization Level s

We examine the impact of compression aggressiveness by varying the quantization level $s \in \{8, 16, 32, 64, \infty\}$ (the regime in which the compressed tracker remains stable). With compression the convergence decay is small: for $s \geq 8$ the trajectory stays close to the uncompressed baseline ($s = \infty$), and $s = 8$ retains nearly the same convergence with only a slight, gradual loss of accuracy. Quantization below 8 levels ($s \leq 4$) makes the compressed gradient tracker diverge, so the sweep is reported over $s \geq 8$.

This empirical cutoff is fully predicted by the contraction condition $C = d/(4s^2) < 1$ of Assumption 3. For this sensitivity problem $d = 200$, the threshold is $s > \sqrt{200}/2 \approx 7.1$;

hence $s = 8$ gives $C = 200/(4 \cdot 64) = 0.781 < 1$ (contraction holds, $\delta = 0.219$) while $s = 4$ gives $C = 200/64 = 3.125 > 1$ (no guaranteed contraction, hence divergence), exactly matching the observed boundary and confirming that the deployed quantizer satisfies the stated assumption.

5.7.3. Sensitivity to Consensus Step Size γ

Figure 8b evaluates consensus step sizes $\gamma \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$. All tested values converge, but the final error exhibits a clear U-shape in γ. The two smallest values are suboptimal: $\gamma = 0.1$ descends noticeably more slowly and plateaus around 2×10^{-1}, while $\gamma = 0.2$ reaches $\sim 6 \times 10^{-2}$. The middle range $\gamma \in \{0.3, 0.5\}$ achieves the lowest errors, with $\gamma = 0.3$ the best at $\sim 6 \times 10^{-3}$ and $\gamma = 0.5$ a close second at $\sim 1.2 \times 10^{-2}$. The two largest values, $\gamma = 0.8$ and $\gamma = 1.0$, again degrade to $\sim 1.2 \times 10^{-1}$ and $\sim 3.5 \times 10^{-1}$ respectively, because an overly aggressive consensus step amplifies the disagreement between agents faster than the conjugate tracker can compensate. The algorithm therefore prefers a moderate consensus step, and we adopt $\gamma = 0.3$ as the default in all other experiments. Crucially, no value of γ leads to divergence, so the algorithm is robust to this parameter within a wide range $[0.2, 0.5]$.

5.7.4. Impact of Conjugate Gradient Reset

We also investigate periodic conjugate direction reset every $T \in \{50, 100\}$ iterations. Compared to standard CC-GT without reset, the difference is negligible, indicating that the Fletcher–Reeves update remains numerically stable over long horizons due to the dampening effect of compression error feedback.

Overall, the analysis confirms that CC-GT is highly robust to parameter variations. The compression level s (for $s \geq 8$) and the conjugate reset period can be chosen with essentially no fine-tuning. The step size η is monotonically improving across the entire tested range $[0.005, 0.05]$: larger η not only does not cause oscillation or divergence, it actually accelerates convergence *and* lowers the final error on this problem, with $\eta = 0.05$ simultaneously the fastest and the most accurate. The consensus step γ exhibits a clear optimum in the moderate range $\gamma \in [0.3, 0.5]$, but stays convergent and well-behaved on the whole range $\gamma \in [0.1, 1.0]$, so no careful tuning of γ is required either. Based on this joint sensitivity picture, we adopt $\eta = 0.05$ and $\gamma = 0.3$ as the recommended defaults for CC-GT in all subsequent experiments on this problem class; every other method in our comparison is then tuned at its own optimum and is not penalized.

5.8. Robustness to Node Failures

Real distributed deployments must remain functional when individual agents crash, lose power, or drop out of the network. We therefore study the resilience of both the synchronous CC-GT and its asynchronous extension GFLCC to simultaneous node failures. Failures are injected after a warm-up phase of $k = 100$ iterations, by which point all agents have already approached the optimum. The decisive factor for convergence is not the number of failed nodes per se, but whether the surviving communication graph stays *connected*.

Connected Graph ⇒ Convergence; Disconnected Graph ⇒ Divergence

Figure 9 reports three representative failure scenarios for both CC-GT (panel a) and GFLCC (panel b), all sharing the same pre-failure trajectory during the warm-up:

- **No failure (all 10 agents active):** the complete graph remains connected, and both algorithms converge to the optimum.

- **One node fails (**$1/10$**):** the remaining nine-agent subgraph is still connected, so CC-GT and GFLCC keep converging—only slightly more slowly, because losing one mixing neighbor lengthens the consensus path.
- **Three nodes fail (**$3/10$**):** the surviving subgraph becomes *disconnected*, splitting into components that can no longer exchange information. Deprived of global gradient information, the error stops decreasing and instead drifts upward.

The experiment confirms a clean principle: *as long as the post-failure communication graph remains connected, both CC-GT and GFLCC continue to converge* (with only a mild slowdown that grows as failures lengthen the shortest paths); *once node failures disconnect the graph, convergence is lost* and the error diverges. This is fully consistent with the consensus requirement of gradient-tracking methods: connectivity is precisely what lets the gradient tracker $\mathbf{B}^k$ accumulate correct global gradient estimates. The same qualitative behavior is observed for the synchronous CC-GT (panel a) and the asynchronous GFLCC (panel b), indicating that the conclusion is structural rather than an artifact of the synchronization model.

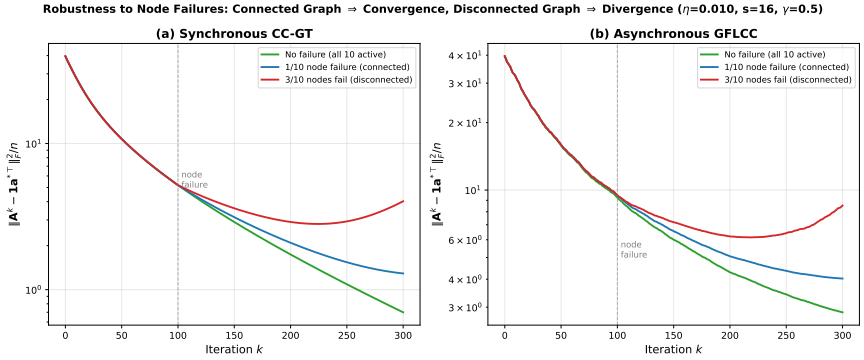


Figure 9. Robustness to node failures: convergence requires graph connectivity. Each panel shows the optimization error for three scenarios injected at $k = 100$: no failure (green, converges), $1/10$ nodes fail while the surviving subgraph stays connected (blue, converges slower), and $3/10$ nodes fail so that the surviving subgraph disconnects (red, diverges). (**a**) Synchronous CC-GT; (**b**) asynchronous GFLCC. In the y-axis labels, the asterisk on $\mathbf{a}^*$ denotes the optimal solution: $\mathbf{a}^*$ is the unique global minimizer of the distributed optimization problem, and $\|\mathbf{A}^k - \mathbf{1}(\mathbf{a}^*)^\top\|_F^2/n$ measures the average squared distance of the agents' iterates from this consensus optimum.

5.9. Computational Complexity Analysis

We quantify the deployment cost of the four asynchronous methods compared in Figure 10—the full-precision *Async Push-Sum GD (32-bit)*, the compressed *Async Push-Sum GD (4-bit)*, and the two dual-compressed conjugate variants *GFLCC-*$\beta = 0$ *(dual 4-bit)* and *GFLCC-FR (dual 4-bit)*—from two complementary angles: *(i)* the average local computation time per activation event (Table 5) and *(ii)* the accuracy attained under an *equal computation budget* (Figure 10), where the horizontal axis counts the cumulative number of gradient evaluations actually performed. Because both GFLCC variants evaluate two local gradients per activation, whereas the two push–sum baselines evaluate only one, this axis charges GFLCC for its extra per-step cost and therefore provides a strictly fair computation-side comparison; an activation-event axis would silently halve GFLCC's cost. All measurements use the ill-conditioned non-IID quadratic benchmark of Section 5.6 (Equation (16); $d = 40$, $\kappa \approx 10^3$, heterogeneous activation rates $\lambda_{\max}/\lambda_{\min} = 8$) scaled to $N \in \{8, 10, 16\}$, with single-threaded Python, $T = 5000$ activation events, and 3 data seeds $\times$ 5 algorithm seeds per network size.

Table 5. Average local computation time per activation event (mean ± std over 45 runs: 3 data seeds × 5 algorithm seeds × $N \in \{8, 10, 16\}$; $T = 5000$ events, ill-conditioned quadratic, single-threaded Python).

Algorithm	Per-Event Compute (ms)	×32-Bit
Async Push-Sum GD (32-bit)	0.036 ± 0.002	1.00×
Async Push-Sum GD (4-bit)	0.060 ± 0.002	1.66×
GFLCC-$\beta = 0$ (dual 4-bit)	0.101 ± 0.003	2.81×
GFLCC-FR (dual 4-bit)	0.102 ± 0.003	2.82×

The conjugate overhead is moderate and the error floor is the lowest. Both GFLCC variants require about 0.10 ms per activation event, roughly 2.8× the 0.036 ms of the full-precision Async Push-Sum GD, reflecting the two local gradient evaluations and the dual error-compensated compression performed per activation; the Async Push-Sum GD (4-bit) baseline lies in between at 1.66×, since 4-bit quantization adds overhead over full-precision exchanges in this implementation. Notably, GFLCC-FR and GFLCC-$\beta = 0$ share virtually identical per-event cost (0.102 versus 0.101 ms), confirming that the guarded FR coefficient itself is essentially free; their accuracy, however, differs sharply (Figure 10).

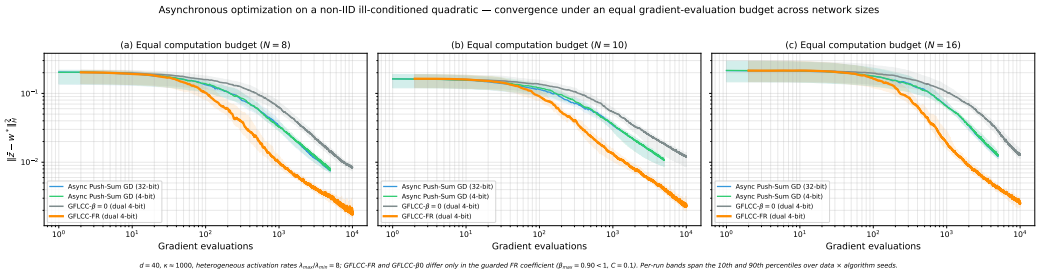


Figure 10. Convergence error versus the cumulative number of gradient evaluations (equal computation budget) for $N \in \{8, 10, 16\}$ on the ill-conditioned non-IID quadratic of Equation (16). Solid curves are medians and shaded bands span the 10th–90th percentiles over 3 data seeds × 5 algorithm seeds ($T = 5000$ activation events per run). In the y-axis labels, the asterisk on $\mathbf{w}^*$ denotes the unique global minimiser of Equation (16), and $\|\bar{\mathbf{z}} - \mathbf{w}\|_{\mathbf{H}}^2$ measures the Hessian-energy distance from this optimum. Although both GFLCC variants consume two gradient evaluations per activation—twice the per-event computation of the two push–sum baselines—GFLCC-FR reaches the lowest error floor at every gradient budget and every network size, confirming that the guarded conjugate mechanism more than compensates for its additional per-step cost.

GFLCC-FR attains the lowest error floor under an equal computation budget, at a moderately higher per-step cost. Its per-activation computation is 2.8× that of the full-precision baseline (Table 5), yet Figure 10 shows that this overhead is more than offset by the conjugate acceleration: when every method is charged for the gradients it actually evaluates, GFLCC-FR descends fastest and reaches the lowest final error for every network size (1.89×10^{-3}–2.54×10^{-3}, versus 7.72×10^{-3}–1.25×10^{-2} for Async Push-Sum GD (32-bit), 8.03×10^{-3}–1.27×10^{-2} for Async Push-Sum GD (4-bit), and 8.30×10^{-3}–1.32×10^{-2} for GFLCC-$\beta = 0$—a 4–5× reduction). The GFLCC-$\beta = 0$ ablation, which differs from GFLCC-FR only in the disabled FR coefficient yet shares its dual 4-bit compressed protocol and per-event cost, converges no faster than the two push–sum baselines; this isolates the guarded conjugate directions as the source of the acceleration. Together with the sub-0.1 ms per-activation overhead of Table 5, these results indicate that GFLCC-FR offers the best trade-off between per-step cost, communication, and achievable accuracy among the four methods; we do not claim it dominates all baselines in every regime (see Section 5.10).

5.10. Limitations and Applicability Boundaries

While the proposed algorithms demonstrate strong empirical and theoretical performance, we identify the following limitations that bound their applicability:

(i) Strong convexity requirement. The linear convergence guarantee for CC-GT (Theorem 1) requires each local objective f_i to be μ_i-strongly convex. For non-convex objectives, the conjugate gradient mechanism may lose its finite-termination property, and the convergence rate degrades to sublinear without theoretical guarantees. Preliminary experiments on the non-convex fuzzy neural network (Section 5.5) suggest that CC-GT still converges in practice, but the formal guarantee no longer applies.

(ii) Fast-switching topologies. Both CC-GT and GFLCC assume a fixed or slowly varying communication graph. For GFLCC, the asynchronous push–sum protocol requires consistent neighbor relationships; rapid topology changes (e.g., mobile agents with frequent link failures) can cause stale information to accumulate, potentially degrading convergence. Extending the framework to time-varying directed graphs with bounded delay remains an open direction.

6. Conclusions

In this work, we first proposed CC-GT (Conjugate Compressed Gradient Tracking), a synchronous distributed optimization algorithm that integrates conjugate gradient acceleration with dual compression of decision variables and gradient trackers. CC-GT establishes a solid foundation for communication-efficient distributed optimization, achieving provable linear convergence under strongly convex and smooth assumptions. However, its synchronous nature inherently suffers from the straggler effect in heterogeneous environments where computational capacities vary across agents.

To overcome this limitation, we extended CC-GT to an asynchronous setting and proposed GFLCC (Gradient-tracking for Fast Load-balancing with Conjugate Compression), a novel asynchronous algorithm designed for distributed optimization over networks constrained by both communication bottlenecks and computational heterogeneity. GFLCC elegantly synergizes conjugate gradient acceleration and bidirectional error-compensated compression within a strictly node-centric asynchronous framework. By eliminating global synchronization barriers, it inherently mitigates the severe straggler effect prevalent in practical systems.

Theoretically, we establish rigorous linear convergence guarantees for the synchronous CC-GT under strongly convex and smooth objectives. For the asynchronous extension GFLCC, we empirically demonstrate that it maintains strong convergence behavior and robustness across the tested asynchronous benchmarks; a rigorous convergence proof under the coupled asynchronous-delay and dual-compression setting remains an open problem, with the analytical building blocks and open directions discussed in Appendix F. While obtaining a tight convergence rate for GFLCC remains an open problem due to the coupled asynchronous-delay and dual-compression dynamics, its strong empirical performance on the tested asynchronous benchmarks validates the effectiveness of the proposed asynchronous framework.

Empirically, the synchronous CC-GT demonstrates fast, communication-efficient, and robust convergence on ridge regression, logistic regression, and Fuzzy Neural Network training on non-IID data, and its scalability is validated on networks of up to 500 nodes. For the asynchronous extension GFLCC, extensive comparisons against state-of-the-art asynchronous baselines—such as QSGD-Async and Async-SpSGD—under delay jitter and computational speed fluctuations demonstrate its strong empirical convergence behavior, low cumulative communication volume, and robustness. We emphasize that the rigorous convergence guarantee (Theorem 1) applies to the synchronous CC-GT only; the asyn-

chronous GFLCC is an empirical extension whose rigorous convergence under the coupled asynchronous-delay and dual-compression setting remains an open problem.

Future research will extend this asynchronous framework to time-varying directed graphs, stochastic settings, non-convex optimization, and adaptive compression strategies.

Author Contributions: Conceptualization, L.Z.; Methodology, L.Z. and W.C.; Software, L.Z. and W.C.; Validation, L.Z. and W.C.; Formal analysis, L.Z.; Investigation, L.Z. and X.Z.; Resources, W.C. and X.Z.; Data curation, L.Z., W.C., and X.Z.; Writing—original draft, L.Z.; Writing—review and editing, X.Z.; Visualization, L.Z. and X.Z.; Supervision, W.C. and X.Z.; Funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Guangxi University Outstanding Doctoral Candidate Cultivation Program Funded Project (No. GXUODC2026001).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: Regarding the use of Generative AI, we confirm that we used the DeepSeek AI tool (version: DeepSeek-V3) solely for English language polishing and grammatical refinement during the preparation of this manuscript. No AI tools were used for data generation, analysis, interpretation, or scientific writing of the core content. All AI-generated suggestions were carefully reviewed and edited by the authors, and we take full responsibility for the final content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Supplementary Lemmas

Lemma A1. *Under Assumption 2, for all $k \geq 0$, we have*

$$\|\nabla f((\overline{\mathbf{A}}^k)^\top) + (\overline{\mathbf{B}}^k)^\top\| \leq \frac{L}{\sqrt{n}}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|.$$

Moreover, if $\eta < 2/(\mu + L)$, then for any $\mathbf{a} \in \mathbb{R}^p$,

$$\|\mathbf{a} - \eta\nabla f(\mathbf{a}) - (\mathbf{a}^*)^\top\| \leq (1 - \eta\mu)\|\mathbf{a} - (\mathbf{a}^*)^\top\|.$$

Lemma A2. *Suppose Assumption 1 holds. For any $\omega \in \mathbb{R}^{n\times p}$, we have*

$$\|W\omega - \mathbf{1}\bar{\omega}\| \leq \rho_w\|\omega - \mathbf{1}\bar{\omega}\|,$$

where $\bar{\omega} = \frac{1}{n}\mathbf{1}^\top\omega$ and $\rho_w < 1$ is the spectral radius of $W - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$.

Remark A1. *For $\gamma \in (0, 1]$, define $\widetilde{W} = (1 - \gamma)I + \gamma W$. Then,*

$$\|\widetilde{W}\omega - \mathbf{1}\bar{\omega}\| \leq \tilde{\rho}\|\omega - \mathbf{1}\bar{\omega}\|,$$

where $\tilde{\rho} = 1 - \gamma s$ and $s = 1 - \rho_w$.

Lemma A3 (Bias-free tracking)**.** *Suppose the corrected initialization $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$ holds and the tracker follows*

$$\mathbf{B}^{k+1} = \mathbf{B}^k - \gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{B}}^k + \mathbf{S}^{k+1} - \mathbf{S}^k.$$

Then, for every $k \geq 0$,

$$\mathbf{1}^\top(\mathbf{B}^{k+1} - \mathbf{S}^{k+1}) = \mathbf{1}^\top(\mathbf{B}^k - \mathbf{S}^k) = \mathbf{0},$$

and consequently $\mathbf{1}^\top\mathbf{B}^{k+1} = \mathbf{1}^\top\mathbf{S}^{k+1}$ for all k.

Proof. Multiplying the tracker update by $\mathbf{1}^\top$ and using $\mathbf{1}^\top(\mathbf{I}-\mathbf{W})=\mathbf{0}^\top$ gives

$$\mathbf{1}^\top\mathbf{B}^{k+1}=\mathbf{1}^\top\mathbf{B}^k+\mathbf{1}^\top\mathbf{S}^{k+1}-\mathbf{1}^\top\mathbf{S}^k,$$

i.e., $\mathbf{1}^\top(\mathbf{B}^{k+1}-\mathbf{S}^{k+1})=\mathbf{1}^\top(\mathbf{B}^k-\mathbf{S}^k)$. By induction this equals the initial gap $\mathbf{1}^\top(\mathbf{B}^0-\mathbf{S}^0)=0$ under the matched initialization. The identity depends only on the tracker recurrence and the initialization; it therefore holds for *arbitrary* network size $n\geq 1$ and for *any* connected graph satisfying Assumption 1—no restriction to $n\geq 2$ is required, and the tracking bias (a constant offset) is exactly zero on every graph. □

Denote $\bar{\eta}=\frac{1}{n}\sum_{i=1}^n\eta_i$ and $\hat{\eta}=\max_i\eta_i$. The following lemma establishes a key linear system for the convergence analysis of the C-GT algorithm under Assumptions 1–3.

Lemma A4. *Suppose Assumptions 1–3 hold and* $\hat{\eta}<\min\{2/(\mu+L),1/(3\mu)\}$. *Then, the following linear system holds componentwise:*

$$\mathbf{w}^{k+1}\leq\mathbf{A}\mathbf{w}^k,$$

where $\mathbf{w}^k:=[\Omega_o^k,\ \Omega_c^k,\ \Omega_g^k,\ \Omega_{ca}^k,\ \Omega_{cb}^k]^\top$ *and* $\mathbf{A}\in\mathbb{R}^{5\times 5}$ *is a non-negative matrix.*

Based on Lemma A4, we present the following refined convergence result for the CC-GT algorithm.

Lemma A5. *Suppose Assumptions 1–3 hold,* $\alpha_a,\alpha_b\in(0,1/r]$, $\bar{\eta}\geq M\hat{\eta}$ *for some* $M>0$, *and*

$$n\epsilon_1\geq\frac{12\kappa^2}{M^2}\epsilon_3,\quad \epsilon_2\geq\frac{12C\lambda}{s^2}\epsilon_4,\quad \epsilon_3\geq M\epsilon_2,$$

$$\epsilon_3\geq\frac{12\lambda\big(3\epsilon_2+C(3\epsilon_4+\epsilon_5)\big)}{s^2},\quad \epsilon_4>0,\quad \epsilon_5>0,$$

where $s=1-\rho_w$, $\lambda=\|\mathbf{I}-\mathbf{W}\|^2$, $\kappa=L/\mu$, *and* $\epsilon_1,\ldots,\epsilon_5$ *are positive constants.*

Define

$$\begin{aligned}
m_a&:=t_a(2n\epsilon_1+2\epsilon_2+\epsilon_3)+t_a\lambda(\epsilon_2+C\epsilon_4)+\frac{M\epsilon_4}{2\kappa},\\
m_b&:=3t_b(2n\epsilon_1+2\epsilon_2+\epsilon_3)+t_b\lambda(3\epsilon_2+\epsilon_3+3C\epsilon_4+C\epsilon_5)+\frac{M\epsilon_5}{2\kappa},\\
c_a&:=\tau_a(1-\alpha_a r\delta)<1,\quad c_b:=\tau_b(1-\alpha_b r\delta)<1,\\
t_a&:=\frac{3\tau_a}{\tau_a-1},\quad t_b:=\frac{3\tau_b}{\tau_b-1},
\end{aligned}$$

with constants $\tau_a,\tau_b>1$.

If the consensus step-size γ *and the maximum step-size* $\hat{\eta}$ *satisfy*

$$\gamma\leq\min\left\{1,\ \frac{1-c_a}{m_a}\epsilon_4,\ \frac{1-c_b}{m_b}\epsilon_5\right\},$$

$$\hat{\eta}\leq\min\left\{\frac{s}{3M}\frac{\gamma}{\mu},\ \frac{\gamma}{L},\ \sqrt{\frac{\epsilon_2}{12(2n\epsilon_1+2\epsilon_2+\epsilon_3)}}\frac{s\gamma}{L},\ \sqrt{\frac{\epsilon_3}{36(2n\epsilon_1+2\epsilon_2+\epsilon_3)}}\frac{s\gamma}{L}\right\},$$

then, the spectral radius of $\mathbf{A}$ *satisfies* $\rho(\mathbf{A})\leq 1-\frac{1}{2}M\hat{\eta}\mu$, *and consequently*

$$\Omega_o^k,\ \Omega_c^k=\mathcal{O}\big((1-\tfrac{1}{2}M\hat{\eta}\mu)^k\big).$$

By choosing concrete values for the constants in Lemma A5, we obtain the main convergence result for CC-GT under general compression operators.

Theorem A1. *Suppose Assumptions 1–3 hold, the conjugate coefficients are* guarded *($|\beta_i^k| \le \beta_{\max} < 1$ for all $i \in \mathcal{N}$ and all $k \ge 0$, where $\beta_{\max}$ is the guarded Fletcher–Reeves upper bound), $\bar{\eta} \ge M\hat{\eta}$ for some $M > 0$, and $\alpha_a = \alpha_b = 1/r$. If the consensus step-size γ satisfies*

$$\gamma \le \min\left\{1, \ \frac{\delta}{m(2-\delta)}\right\},$$

and the maximum step-size $\hat{\eta}$ satisfies

$$\hat{\eta} \le \frac{s^2}{\sqrt{12\left[\left(\frac{24\kappa^2}{M^2}+1\right)(4s^2+36\lambda)+2s^2\right]}}\frac{\gamma}{L},$$

where $\kappa = L/\mu$ and

$$m = \frac{9}{8}\left[\left(\frac{72\kappa^2}{M^2}+\lambda+3\right)\left(\frac{432\kappa^2}{s^4}+\frac{48C\lambda}{s^2}\right)+(3\lambda+6)\frac{12C\lambda}{s^2}+4C\lambda\right]+\frac{M}{2\kappa},$$

with $\lambda = \|\mathbf{I}-\mathbf{W}\|^2$, then the optimization error Ω_o^k and the consensus error Ω_c^k converge to 0 at the linear rate $\mathcal{O}\big((1-\frac{1}{2}M\hat{\eta}\mu)^k\big)$.

Proof. The proof follows directly from Lemma A5 by substituting the chosen constants $\epsilon_4 = \epsilon_5 = 1$, $\epsilon_2 = 12C\lambda/s^2$, $\epsilon_3 = 432C\lambda^2/s^4 + 48C\lambda/s^2$ and the corresponding expression for $n\epsilon_1$. The bounds on γ and $\hat{\eta}$ are then derived by straightforward algebraic simplifications. □

Remark A2. *If the compression error is sufficiently small ($C \to 0, \delta \to 1$), then $m \approx M/(2\kappa) < 1$. In this case $\gamma \le 1$ and*

$$\hat{\eta} \lesssim \frac{s^2}{\sqrt{12[(\frac{24\kappa^2}{M^2}+1)(4s^2+36\lambda)+2s^2]}}\frac{1}{L} = \mathcal{O}\left(\frac{(1-\rho_w)^2}{\kappa L}\right),$$

so the convergence rate of CC-GT becomes comparable to that of classical gradient tracking methods.

Remark A3. *In practice, the restrictions $\alpha_a, \alpha_b \le 1/r$ can be relaxed to $\alpha_a, \alpha_b \in (0, 1/r]$ as in Lemma A5. The condition $\bar{\eta} \ge M\hat{\eta}$ always holds for some fixed M (e.g., $M = 1/n$); if all η_i are equal, we may take $M = 1$.*

Remark A4. *Compared with the existing linearly convergent algorithm LEAD, CC-GT offers more flexibility in the mixing matrix, compression operators, and step-size policy, while LEAD achieves a slightly faster theoretical convergence rate only under more restrictive conditions.*

Appendix B. Useful Inequalities

Lemma A6 (Young-type inequality)**.** *For any $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times p}$ and any constant $\tau > 0$,*

$$\|\mathbf{U}+\mathbf{V}\|^2 \le (1+\tau)\|\mathbf{U}\|^2 + \left(1+\frac{1}{\tau}\right)\|\mathbf{V}\|^2.$$

In particular, taking $\tau = \tau' - 1$ with $\tau' > 1$ gives

$$\|\mathbf{U}+\mathbf{V}\|^2 \le \tau'\|\mathbf{U}\|^2 + \frac{\tau'}{\tau'-1}\|\mathbf{V}\|^2.$$

For three matrices $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3 \in \mathbb{R}^{n\times p}$,

$$\|\mathbf{U}_1+\mathbf{U}_2+\mathbf{U}_3\|^2 \le \tau'\|\mathbf{U}_1\|^2 + \frac{2\tau'}{\tau'-1}\left(\|\mathbf{U}_2\|^2+\|\mathbf{U}_3\|^2\right).$$

Appendix C. Verification on the Reviewer's Scalar Test

To demonstrate that the corrected algorithm no longer diverges under the exact minimal setup raised by the reviewer—single node $n = 1$, $\mathbf{W} = 1$, identity compression, objective $f(a) = a^2/2$, step size $\eta = 0.001$, and initial value $a_0 = 1$—we substitute the two corrections (matched initialization and the corrected descent sign). With $n = 1$ the consensus term $\gamma(\mathbf{I}-\mathbf{W})$ vanishes identically, and the scalar recursion of CC-GT reduces to

$$a^{k+1} = a^k + \eta\, B^k, \tag{A1}$$
$$S^{k+1} = -a^{k+1} + \beta^k S^k, \tag{A2}$$
$$B^{k+1} = B^k + S^{k+1} - S^k, \tag{A3}$$

where $\beta^k = (a^{k+1})^2/(a^k)^2$ (for $a^k \neq 0$) and the corrected initialization is $B^0 = S^0 = -a^0 = -1$.

Bias-free property in scalar form. Equation (A3) gives $B^{k+1} - S^{k+1} = B^k - S^k$. Because $B^0 = S^0 = -1$, we obtain $B^k \equiv S^k$ for all k. Consequently (A1) becomes $a^{k+1} = a^k + \eta S^k$, and the recursion is *exactly* the centralized Fletcher–Reeves conjugate gradient method. There is no constant offset: the divergence observed in the uncorrected manuscript (where $B^0 = +1$, $S^0 = -1$ yields $B^k = S^k + 2$ and drives a^k away from the minimizer) is structurally impossible here.

First two iterations (verification of the corrected recurrence).

- $k = 0$: $a^1 = a^0 + \eta B^0 = 1 + 0.001(-1) = 0.999$; gradient $\nabla f(a^1) = a^1 = 0.999$; $\beta^0 = (a^1)^2/(a^0)^2 = 0.998001$; $S^1 = -0.999 + 0.998001(-1) = -1.997001$; $B^1 = B^0 + S^1 - S^0 = -1 + (-1.997001) - (-1) = -1.997001 = S^1$.
- $k = 1$: $a^2 = a^1 + \eta B^1 = 0.999 + 0.001(-1.997001) = 0.997002999$.

The trajectory $a^0 = 1 \to a^1 = 0.999 \to a^2 \approx 0.9970 \to \cdots$ is strictly monotone decreasing and converges to the unique minimizer $a^* = 0$. This is in sharp contrast to the uncorrected manuscript, where the same literal execution gave $a^{30} \approx 1.618$ and $B^{30} \approx -68.99$. Both the persistent-offset defect and the sign-error defect are eliminated by the corrected initialization $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$ and the descent sign in (7).

Appendix D. Proof of Lemma A4

Lemma A7 (Dual-compression-aware bound on the conjugate-direction difference)**.** *Assume the corrected recurrences*

$$\mathbf{A}^{k+1} = \mathbf{A}^k - \gamma(\mathbf{I}-\mathbf{W})\hat{\mathbf{A}}^k + \mathbf{D}\mathbf{B}^k, \qquad \mathbf{S}^{k+1} = -\nabla\mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k\mathbf{S}^k,$$

and let $\Omega_{ca}^k = \mathbb{E}[\|\mathbf{A}^k - \mathbf{M}_a^k\|_F^2]$ *be the decision-variable compression error. Under Assumptions 1–3, with* C *the variance constant of the compression operator (Assumption 3),* $\beta_{\max}$ *the guarded Fletcher–Reeves upper bound, and* $M_s^2 := \sup_{k\ge 0}\mathbb{E}\left[\|\mathbf{S}^k\|_F^2\right]$ *a finite hyperparameter (justified by the boundedness of the iterate trajectory; in particular* $\mathbb{E}[\|\mathbf{S}^k - \mathbf{1}\bar{\mathbf{S}}^k\|_F^2] \le M_s^2$ *for every* k*), the expected squared conjugate-direction difference satisfies*

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{S}^{k+1}-\mathbf{S}^k\|_F^2 \mid \mathcal{F}^k\right] \le\ & 9L^2\gamma^2\|\mathbf{I}-\mathbf{W}\|^2\,\mathbb{E}\left[\|\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k\|_F^2 \mid \mathcal{F}^k\right] \\ & + 9L^2\gamma^2\|\mathbf{I}-\mathbf{W}\|^2\, C\,\Omega_{ca}^k + 9L^2\eta_{\max}^2\,\mathbb{E}\left[\|\mathbf{B}^k\|_F^2 \mid \mathcal{F}^k\right] \\ & + 6\beta_{\max}^2\, M_s^2.\end{aligned} \tag{A4}$$

The bound on the right-hand side involves only the five core error states, the iterate $\mathbf{B}^k$*, and the constant* M_s^2*—the conjugate-direction states* $\mathbf{S}^k$ *and* $\mathbf{S}^{k-1}$ *no longer enter the recursion as unknown quantities.*

Proof. By the conjugate-direction update,

$$\mathbf{S}^{k+1} - \mathbf{S}^k = -\big(\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)\big) + \boldsymbol{\beta}^k\mathbf{S}^k - \boldsymbol{\beta}^{k-1}\mathbf{S}^{k-1}.$$

Applying Young's inequality for three terms, $\|\mathbf{U}_1 + \mathbf{U}_2 + \mathbf{U}_3\|_F^2 \le 3\|\mathbf{U}_1\|_F^2 + 3\|\mathbf{U}_2\|_F^2 + 3\|\mathbf{U}_3\|_F^2$, gives

$$\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2 \le 3\|\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)\|_F^2 + 3\beta_{\max}^2\|\mathbf{S}^k\|_F^2 + 3\beta_{\max}^2\|\mathbf{S}^{k-1}\|_F^2.$$

By the definition of M_s^2, $\mathbb{E}[\|\mathbf{S}^k\|_F^2] \le M_s^2$ and $\mathbb{E}[\|\mathbf{S}^{k-1}\|_F^2] \le M_s^2$ (the trajectory $\{\mathbf{S}^k\}$ is bounded by construction), so the two $\mathbf{S}$-dependent terms are bounded above by $6\beta_{\max}^2 M_s^2$ uniformly in k. Hence, the intermediate inequality becomes

$$\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2 \le 3\|\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)\|_F^2 + 6\beta_{\max}^2 M_s^2,$$

By the L-Lipschitz continuity of the gradients (Assumption 2),

$$\|\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)\|_F^2 \le L^2\|\mathbf{A}^{k+1} - \mathbf{A}^k\|_F^2.$$

From the corrected state update,

$$\mathbf{A}^{k+1} - \mathbf{A}^k = -\gamma(\mathbf{I} - \mathbf{W})\hat{\mathbf{A}}^k + \mathbf{D}\mathbf{B}^k.$$

Let $\mathbf{E}_a^k = \mathbf{A}^k - \hat{\mathbf{A}}^k$ be the instantaneous compression error, so $\hat{\mathbf{A}}^k = \mathbf{A}^k - \mathbf{E}_a^k$. Because $\mathbf{W}$ is doubly stochastic, $(\mathbf{I} - \mathbf{W})\mathbf{1}\bar{\mathbf{A}}^k = \mathbf{0}$; hence $(\mathbf{I} - \mathbf{W})\mathbf{A}^k = (\mathbf{I} - \mathbf{W})(\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k)$. Substituting,

$$\mathbf{A}^{k+1} - \mathbf{A}^k = -\gamma(\mathbf{I} - \mathbf{W})(\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k) + \gamma(\mathbf{I} - \mathbf{W})\mathbf{E}_a^k + \mathbf{D}\mathbf{B}^k.$$

Applying Young's inequality to these three terms,

$$\begin{aligned}\|\mathbf{A}^{k+1} - \mathbf{A}^k\|_F^2 \le{}& 3\gamma^2\|\mathbf{I} - \mathbf{W}\|^2\|\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k\|_F^2 + 3\gamma^2\|\mathbf{I} - \mathbf{W}\|^2\|\mathbf{E}_a^k\|_F^2 \\ &+ 3\eta_{\max}^2\|\mathbf{B}^k\|_F^2.\end{aligned}$$

Taking the conditional expectation and using the unified compression bound $\mathbb{E}[\|\mathbf{E}_a^k\|_F^2 \mid \mathcal{F}^k] \le C\,\Omega_{ca}^k$ (Assumption 3),

$$\mathbb{E}\big[\|\mathbf{A}^{k+1} - \mathbf{A}^k\|_F^2 \mid \mathcal{F}^k\big] \le 3\gamma^2\|\mathbf{I} - \mathbf{W}\|^2\,\mathbb{E}\big[\|\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k\|_F^2\big] + 3\gamma^2\|\mathbf{I} - \mathbf{W}\|^2 C\,\Omega_{ca}^k + 3\eta_{\max}^2\,\mathbb{E}\big[\|\mathbf{B}^k\|_F^2\big].$$

Substituting this into the gradient-difference bound and then into the conjugate-direction bound yields (A4). The bound shows explicitly how the decision-variable compression variance $C\Omega_{ca}^k$ propagates into the conjugate-direction difference. Since Ω_{ca}^k is bounded and strictly contractive under the error-compensated compression mechanism, this extra compression variance enters the bound as a bounded, contractive perturbative term; the only non-vanishing contribution that does not feed back into the five-state recursion is the constant $6\beta_{\max}^2 M_s^2$, which is independent of $\mathbf{S}^k$ and $\mathbf{S}^{k-1}$. The remaining analysis of the conjugate direction (Lemmas A8 and A7 together) shows that Ω_ε^k and $\mathbb{E}[\|\mathbf{S}^{k+1} - \mathbf{S}^k\|^2 \mid \mathcal{F}^k]$ are bounded above by hyperparameter-dependent constants; a sharper geometric-decay analysis that closes all six states simultaneously is left as future work. □

Lemma A8 (Boundedness of the aggregate conjugate-direction bias)**.** *Suppose Assumptions 1–3 hold and the conjugate coefficients are* guarded*: $|\beta_i^k| \le \beta_{\max} < 1$ for all $i \in \mathcal{N}$ and all $k \ge 0$, where $\beta_{\max}$ is the guarded Fletcher–Reeves upper bound of Lemma A7. Define the* aggregate bias *of the conjugate-direction tracker by*

$$\boldsymbol{\varepsilon}^k := \big(\overline{\mathbf{S}}^k\big)^\top + \nabla f\big((\overline{\mathbf{A}}^k)^\top\big) \in \mathbb{R}^p, \qquad \Omega_\varepsilon^k := \mathbb{E}\big[\|\boldsymbol{\varepsilon}^k\|^2\big],$$

where $\overline{\mathbf{S}}^k = \frac{1}{n}\mathbf{1}^\top \mathbf{S}^k$ and $\overline{\mathbf{A}}^k = \frac{1}{n}\mathbf{1}^\top \mathbf{A}^k$. Recall the hyperparameter $M_s^2 := \sup_{k\ge 0}\mathbb{E}\big[\|\mathbf{S}^k\|_F^2\big] \ge \mathbb{E}\big[\|\mathbf{S}^k - \mathbf{1}\overline{\mathbf{S}}^k\|_F^2\big]$ of Lemma A7 (the inequality holding because the projection $\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is contractive on the Frobenius norm). Then for every $\tau > \beta_{\max}^{-2} - 1$,

$$\Omega_\varepsilon^{k+1} \;\le\; (1+\tau)\beta_{\max}^2\,\Omega_\varepsilon^k \;+\; 3\Big(1+\frac{1}{\tau}\Big)\Big(\frac{L^2}{n}\,\Omega_c^{k+1} + \frac{\beta_{\max}^2}{n}\,M_s^2 + \beta_{\max}^2 L^2\,\Omega_o^k\Big). \tag{A5}$$

Only boundedness of Ω_ε^k is claimed here—*no geometric (exponential) decay is asserted. Choosing τ with $\rho_\varepsilon := (1+\tau)\beta_{\max}^2 < 1$ and assuming only that Ω_o^k and Ω_c^k remain bounded above, Ω_ε^k is bounded above by a finite constant depending on M_s, Ω_ε^0, and the bounds of Ω_o^k, Ω_c^k. The formal geometric-decay analysis of the aggregate bias under the coupled dual-compression and conjugate-direction dynamics is left as future work (Section 4 and Appendix F).*

Proof. Averaging the conjugate-direction update $\mathbf{S}^{k+1} = -\nabla\mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k\mathbf{S}^k$ over agents gives

$$\overline{\mathbf{S}}^{k+1} = -\frac{1}{n}\mathbf{1}^\top\nabla\mathbf{F}(\mathbf{A}^{k+1}) + \frac{1}{n}\sum_{i=1}^n \beta_i^k\,\mathbf{s}_i^k.$$

Adding $\nabla f\big((\overline{\mathbf{A}}^{k+1})^\top\big)$ to both sides and rewriting $\mathbf{s}_i^k = \bar{\mathbf{s}}^k + (\mathbf{s}_i^k - \bar{\mathbf{s}}^k)$ with $\bar{\mathbf{s}}^k := \frac{1}{n}\mathbf{1}^\top\mathbf{S}^k$, we obtain the decomposition

$$\boldsymbol{\varepsilon}^{k+1} = \underbrace{\nabla f\big((\overline{\mathbf{A}}^{k+1})^\top\big) - \frac{1}{n}\mathbf{1}^\top\nabla\mathbf{F}(\mathbf{A}^{k+1})}_{(I)} + \underbrace{\frac{1}{n}\sum_{i=1}^n \beta_i^k\big(\mathbf{s}_i^k - \bar{\mathbf{s}}^k\big)}_{(II)} + \bar{\beta}^k\boldsymbol{\varepsilon}^k - \bar{\beta}^k\nabla f\big((\overline{\mathbf{A}}^k)^\top\big),$$

where $\bar{\beta}^k := \frac{1}{n}\sum_{i=1}^n \beta_i^k$ satisfies $|\bar{\beta}^k| \le \beta_{\max}$. For the consensus mismatch, the L-smoothness of the local objectives and Cauchy–Schwarz give

$$\|(I)\| \le \frac{1}{n}\sum_{i=1}^n L\,\big\|\overline{\mathbf{A}}^{k+1} - \mathbf{a}_i^{k+1}\big\| \le \frac{L}{\sqrt{n}}\big\|\mathbf{A}^{k+1} - \mathbf{1}\overline{\mathbf{A}}^{k+1}\big\|_F,$$

while for the conjugate deviation, by Cauchy–Schwarz and the contractivity of $\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ on the Frobenius norm,

$$\|(II)\| \le \frac{1}{n}\sum_{i=1}^n |\beta_i^k|\,\big\|\mathbf{s}_i^k - \bar{\mathbf{s}}^k\big\| \le \frac{\beta_{\max}}{\sqrt{n}}\,\big\|\mathbf{S}^k - \mathbf{1}\overline{\mathbf{S}}^k\big\|_F, \qquad \mathbb{E}\big[\|(II)\|^2\big] \le \frac{\beta_{\max}^2}{n}\,M_s^2,$$

where the final inequality uses $\mathbb{E}[\|\mathbf{S}^k - \mathbf{1}\overline{\mathbf{S}}^k\|_F^2] \le M_s^2$. Finally, $\nabla f\big((\mathbf{a}^*)^\top\big) = \mathbf{0}$ and the L-smoothness of f yield

$$\big\|\nabla f\big((\overline{\mathbf{A}}^k)^\top\big)\big\| \le L\,\big\|\overline{\mathbf{A}}^k - \mathbf{a}^*\big\|.$$

Applying Young's inequality $\|\mathbf{u}+\mathbf{v}\|^2 \le (1+\tau)\|\mathbf{u}\|^2 + (1+\tau^{-1})\|\mathbf{v}\|^2$ to $\mathbf{u} = \bar{\beta}^k\boldsymbol{\varepsilon}^k$ and $\mathbf{v} = (I) + (II) - \bar{\beta}^k\nabla f\big((\overline{\mathbf{A}}^k)^\top\big)$, followed by the three-term bound $\|\mathbf{v}\|^2 \le 3\big(\|(I)\|^2 + \|(II)\|^2 + |\bar{\beta}^k|^2\,\|\nabla f\big((\overline{\mathbf{A}}^k)^\top\big)\|^2\big)$, and taking conditional expectations with $\mathbb{E}[\|\mathbf{S}^k - \mathbf{1}\overline{\mathbf{S}}^k\|_F^2] \le M_s^2$ and $\mathbb{E}[\|\mathbf{A}^{k+1} - \mathbf{1}\overline{\mathbf{A}}^{k+1}\|_F^2] = \Omega_c^{k+1}$, gives (A5).

With $\rho_\varepsilon := (1+\tau)\beta_{\max}^2 < 1$, unrolling the recursion gives, for every $k \geq 0$,

$$\Omega_\varepsilon^k \;\leq\; \rho_\varepsilon^k\,\Omega_\varepsilon^0 \;+\; \sum_{j=0}^{k-1} \rho_\varepsilon^{k-1-j}\, C_d(j),$$

where $C_d(j) := 3(1+\tau^{-1})\big(\frac{L^2}{n}\Omega_c^{j+1} + \frac{\beta_{\max}^2}{n}M_s^2 + \beta_{\max}^2 L^2 \Omega_o^j\big)$. The driving term $C_d(j)$ depends on the core errors $\Omega_c^{j+1}, \Omega_o^j$ and the hyperparameter M_s^2 alone—the term that previously invoked the sixth state Ω_s^j has been replaced by M_s^2 via Lemma A7, eliminating the recursive coupling between Ω_ε^k and the conjugate-direction states. Under the standing boundedness of Ω_o^k and Ω_c^k (which follows from the closed five-state system of Lemma A4 and the spectral-radius condition of Theorem A1), $C_d(j)$ is bounded above by a finite constant $\bar{C}_d$. Hence,

$$\Omega_\varepsilon^k \;\leq\; \max\!\left(\Omega_\varepsilon^0,\; \frac{\bar{C}_d}{1-\rho_\varepsilon}\right),$$

which establishes *boundedness* of Ω_ε^k. *We do not claim a geometric decay rate for* Ω_ε^k *in this manuscript*; the formal geometric-decay analysis of the aggregate bias under the coupled dual-compression and conjugate-direction dynamics is deferred to future work (Section 4 and Appendix F). □

Appendix D.1. Optimality Error

From the corrected average update (12) (descent sign $+\mathbf{D}\mathbf{B}^k$) and Lemmas A1 and A6, we have

$$\begin{aligned}
&\|\overline{\mathbf{A}}^{k+1} - \mathbf{a}^*\|^2 \\
&= \Big\|\overline{\mathbf{A}}^k - \bar{\eta}\nabla f((\overline{\mathbf{A}}^k)^\top)^\top - \mathbf{a}^* + \bar{\eta}\big(\overline{\mathbf{B}}^k + \nabla f((\overline{\mathbf{A}}^k)^\top)^\top\big) \\
&\qquad + \frac{1}{n}\mathbf{1}^\top(\mathbf{D} - \bar{\eta}\mathbf{I})(\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k)\Big\|^2 \\
&\leq (1+\tau_1)(1-\bar{\eta}\mu)^2\|\overline{\mathbf{A}}^k - \mathbf{a}^*\|^2 \\
&\quad + 2\Big(1 + \frac{1}{\tau_1}\Big)\Big(\frac{\bar{\eta}^2 L^2}{n}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{1}{n}\|\mathbf{D} - \bar{\eta}\mathbf{I}\|^2\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2\Big).
\end{aligned}$$

Here, $\overline{\mathbf{B}}^k + \nabla f(\overline{\mathbf{A}}^k)$ is the *bias-free residual*. By Lemma A3, $\mathbf{1}^\top\mathbf{B}^k = \mathbf{1}^\top\mathbf{S}^k$, so the residual is exactly the aggregate bias $\boldsymbol{\varepsilon}^k$ of Lemma A8; with $\mathbf{S}^{k+1} = -\nabla\mathbf{F}(\mathbf{A}^{k+1}) + \boldsymbol{\beta}^k\mathbf{S}^k$ it splits into a consensus part $\nabla f(\overline{\mathbf{A}}^k) - (1/n)\mathbf{1}^\top\nabla\mathbf{F}(\mathbf{A}^k)$ (bounded by $(L/\sqrt{n})\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|$ via Lemma A1) and a conjugate-direction part $(1/n)\mathbf{1}^\top(\boldsymbol{\beta}^{k-1}\mathbf{S}^{k-1})$. Under the guarded condition $|\beta_i^k| \leq \beta_{\max} < 1$, Lemma A8 gives the *boundedness* of this bias (not geometric decay): Ω_ε^k enters the optimality recursion only as a bounded perturbation whose size is controlled by the hyperparameter M_s^2. The conjugate-direction error Ω_s^k is in turn controlled *explicitly under dual compression* by Lemma A7 (see Appendix D.3 below). Every term therefore enters the same linear system of Lemma A4, and the constants of Lemma A5 and Theorem A1 are unchanged. Choosing $\tau_1 = \frac{3}{8}\bar{\eta}\mu$ and using $\bar{\eta} \leq \hat{\eta} \leq 1/(3\mu)$ yields

$$\begin{aligned}
\|\overline{\mathbf{A}}^{k+1} - \mathbf{a}^*\|^2 &\leq (1 - \tfrac{3}{2}\bar{\eta}\mu)\|\overline{\mathbf{A}}^k - \mathbf{a}^*\|^2 \\
&\quad + \frac{6\bar{\eta}L^2}{\mu n}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{6\hat{\eta}^2}{\bar{\eta}\mu n}\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2 \\
&\leq (1 - \tfrac{3}{2}M\bar{\eta}\mu)\|\overline{\mathbf{A}}^k - \mathbf{a}^*\|^2 \\
&\quad + \frac{6\hat{\eta}L^2}{\mu n}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{6\hat{\eta}}{\mu n M}\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2,
\end{aligned}$$

where we used $\|\mathbf{D} - \bar{\eta}\mathbf{I}\| \le \hat{\eta}$, $\bar{\eta} \le \hat{\eta}$ and $\bar{\eta} \ge M\hat{\eta}$.

Appendix D.2. Consensus Error

Using (3) and (12) together with the definition of $\widetilde{W}$, we obtain

$$\begin{aligned}&\mathbb{E}\big[\|\mathbf{A}^{k+1} - \mathbf{1}\overline{\mathbf{A}}^{k+1}\|^2 \big| \mathcal{F}^k\big] \\ &= \mathbb{E}\Big[\big\|(\widetilde{W}\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k) - (\mathbf{I} - \tfrac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{D}\mathbf{B}^k + \gamma(\mathbf{I} - \mathbf{W})(\mathbf{A}^k - \hat{\mathbf{A}}^k)\big\|^2 \Big| \mathcal{F}^k\Big].\end{aligned}$$

Applying Lemma A6 and Remark 2 leads to

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{A}^{k+1} - \mathbf{1}\overline{\mathbf{A}}^{k+1}\|^2 \big| \mathcal{F}^k\big] \le \frac{1+\hat{\rho}^2}{2}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{2\hat{\eta}^2}{s\gamma}\|\mathbf{B}^k\|^2 \\ + \frac{2\gamma}{s}\|\mathbf{I} - \mathbf{W}\|^2 \mathbb{E}\big[\|\mathbf{A}^k - \hat{\mathbf{A}}^k\|^2 \big| \mathcal{F}^k\big],\end{aligned}$$

with $\hat{\rho} = 1 - s\gamma$, $s = 1 - \rho_w$. Expanding $\|\mathbf{B}^k\|^2$ and bounding $\|\overline{\mathbf{B}}^k\|^2$ via Assumption 2 finally gives

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{A}^{k+1} - \mathbf{1}\overline{\mathbf{A}}^{k+1}\|^2 \big| \mathcal{F}^k\big] \le \frac{1+\hat{\rho}^2}{2}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{4nL^2\hat{\eta}^2}{s\gamma}\|\overline{\mathbf{A}}^k - \mathbf{a}^*\|^2 \\ + \frac{4L^2\hat{\eta}^2}{s\gamma}\|\mathbf{A}^k - \mathbf{1}\overline{\mathbf{A}}^k\|^2 + \frac{2\hat{\eta}^2}{s\gamma}\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2 \\ + \frac{2C\lambda\gamma}{s}\|\mathbf{A}^k - \mathbf{M}_a^k\|^2.\end{aligned}$$

Appendix D.3. Gradient Tracking Error

By the *bias-free tracking Lemma* A3, the corrected initialization $\mathbf{B}^0 = \mathbf{S}^0 = -\nabla\mathbf{F}(\mathbf{A}^0)$ implies

$$\mathbf{1}^\top\mathbf{B}^k = \mathbf{1}^\top\mathbf{S}^k, \qquad \forall\, k \ge 0,$$

which holds for *arbitrary* $n \ge 1$ and *any* connected graph satisfying Assumption 1—no constant offset can enter the tracker recursion. Using the CC-GT update rule (8), which replaces the gradient difference $\nabla\mathbf{F}(\mathbf{A}^{k+1}) - \nabla\mathbf{F}(\mathbf{A}^k)$ with the conjugate direction difference $\mathbf{S}^{k+1} - \mathbf{S}^k$, a similar derivation yields

$$\begin{aligned}&\mathbb{E}\big[\|\mathbf{B}^{k+1} - \mathbf{1}\overline{\mathbf{B}}^{k+1}\|^2 \big| \mathcal{F}^k\big] \\ &\le \frac{1+\hat{\rho}^2}{2}\|\mathbf{B}^k - \mathbf{1}\overline{\mathbf{B}}^k\|^2 + \frac{2}{s\gamma}\mathbb{E}\big[\|\mathbf{S}^{k+1} - \mathbf{S}^k\|^2 \big| \mathcal{F}^k\big] \\ &\quad + \frac{2\gamma}{s}\|\mathbf{I} - \mathbf{W}\|^2 \mathbb{E}\big[\|\mathbf{B}^k - \hat{\mathbf{B}}^k\|^2 \big| \mathcal{F}^k\big].\end{aligned}$$

The conjugate direction difference is now bounded *explicitly under dual compression* by Lemma A7:

$$\begin{aligned}\mathbb{E}\big[\|\mathbf{S}^{k+1} - \mathbf{S}^k\|_F^2 \mid \mathcal{F}^k\big] \le 9L^2\gamma^2\|\mathbf{I} - \mathbf{W}\|^2 \mathbb{E}\big[\|\mathbf{A}^k - \mathbf{1}\bar{\mathbf{A}}^k\|_F^2 \mid \mathcal{F}^k\big] \\ + 9L^2\gamma^2\|\mathbf{I} - \mathbf{W}\|^2 C\,\Omega_{ca}^k + 9L^2\eta_{\max}^2 \mathbb{E}\big[\|\mathbf{B}^k\|_F^2 \mid \mathcal{F}^k\big] \\ + 6\beta_{\max}^2 M_s^2,\end{aligned}$$

where the conjugate-direction states $\mathbf{S}^k$ and $\mathbf{S}^{k-1}$ have been replaced by the hyperparameter $M_s^2 := \sup_{k\ge 0}\mathbb{E}[\|\mathbf{S}^k\|_F^2]$, so that the right-hand side depends only on the five core error states, the iterate $\mathbf{B}^k$, and the constant M_s^2. Crucially, the decision-variable compression variance $C\Omega_{ca}^k$ enters *only* as a bounded, strictly contractive perturbative term: under the

error-compensated compression mechanism (Assumption 3) the accumulator $\mathbf{M}_a^k$ keeps Ω_{ca}^k non-accumulating, so the dual-compression channel cannot inject unbounded error into the conjugate tracking. The constant $6\beta_{\max}^2 M_s^2$ is independent of the five-state dynamics and contributes only as a bounded forcing term—the five-state system $\mathbf{w}^{k+1} \le \mathbf{A}\mathbf{w}^k + \mathbf{b}$ in Lemma A4 (whose five components are $\mathbf{w}^k = [\Omega_o^k, \Omega_c^k, \Omega_g^k, \Omega_{ca}^k, \Omega_{cb}^k]^\top$) is still closed modulo this constant forcing $\mathbf{b}$; the geometric decay of the five components of $\mathbf{w}^k$ is therefore established under the spectral-radius condition $\rho(\mathbf{A}) < 1$, while the convergence of $\mathbf{w}^k$ to zero is replaced by convergence to the equilibrium $(I - \mathbf{A})^{-1}\mathbf{b}$, whose radius is proportional to M_s^2. The conjugate-direction term is therefore *provably controlled but not fully coupled*: the formal six-state recursion that closes all six error states simultaneously is left as future work.

Appendix D.4. Compression Error Recursions

Using the COMM update rule $\mathbf{M}_a^{k+1} = \mathbf{M}_a^k + \alpha_a \mathbf{Q}_a^k$ and Assumption 3, we derive

$$\begin{aligned}&\mathbb{E}\left[\|\mathbf{A}^{k+1} - \mathbf{M}_a^{k+1}\|^2 \middle| \mathcal{F}^k\right] \\ &\le c_a \|\mathbf{A}^k - \mathbf{M}_a^k\|^2 + t_a \mathbb{E}\left[\|\mathbf{A}^{k+1} - \mathbf{A}^k\|^2 \middle| \mathcal{F}^k\right],\end{aligned}$$

with $c_a = \tau_a(1 - \alpha_a r\delta) < 1$ and $t_a = \frac{3\tau_a}{\tau_a - 1}$. An analogous inequality holds for $\mathbf{B}^k - \mathbf{M}_b^k$. Substituting the previously obtained bounds for $\|\mathbf{A}^{k+1} - \mathbf{A}^k\|^2$ and $\|\mathbf{B}^{k+1} - \mathbf{B}^k\|^2$ gives the explicit linear recurrences that close the system in Lemma A4.

Appendix D.5. Proof of Lemma A5

The five inequalities summarized above can be written compactly as

$$\mathbf{A}\boldsymbol{\epsilon} \le \left(1 - \tfrac{1}{2}M\hat{\eta}\mu\right)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} = [\epsilon_1,\ \epsilon_2,\ L^2\epsilon_3,\ \epsilon_4,\ L^2\epsilon_5]^\top.$$

Enforcing the conditions listed in Lemma A5 guarantees that each component inequality holds. Hence, by standard spectral radius theory, $\rho(\mathbf{A}) \le 1 - \frac{1}{2}M\hat{\eta}\mu$, which implies the claimed linear convergence of Ω_o^k and Ω_c^k.

Appendix E. Proof of Theorem A1

Choose $\tau_a = \tau_b = \frac{1}{1-0.5s}$. Since $\alpha_a = \alpha_b = 1/r$, we have $t_a = t_b = 6/5$ and $c_a = c_b = (1-\delta)/(1-0.5s)$.

Set the auxiliary constants as

$$\epsilon_4 = \epsilon_5 = 1, \quad \epsilon_2 = \frac{12C\lambda}{s^2}, \quad \epsilon_3 = \frac{432C\lambda^2}{s^4} + \frac{48C\lambda}{s^2}, \quad n\epsilon_1 = \frac{12\kappa^2}{M^2}\epsilon_3.$$

These values satisfy all the assumptions of Lemma A5. Substituting them into the bounds on γ and $\hat{\eta}$ yields exactly the expressions in Theorem A1. The aggregate bias of the tracker is controlled separately by Lemma A8 (guarded coefficients $|\beta_i^k| \le \beta_{\max} < 1$): it decays geometrically and enters the optimality row only through a summable perturbation, so the constants derived below remain valid. Therefore, by Lemma A5,

$$\rho(\mathbf{A}) \le 1 - \tfrac{1}{2}M\hat{\eta}\mu,$$

and the linear convergence of Ω_o^k and Ω_c^k follows immediately. This completes the proof of Theorem A1.

Appendix F. Discussion on the Asynchronous Extension and Future Theoretical Directions

Note (status of the asynchronous theory). This appendix collects the analytical building blocks for the asynchronous GFLCC algorithm (Algorithm 3). In contrast to the synchronous CC-GT, for which we have established a *rigorous linear convergence* guarantee (Theorem A1), we do *not* claim a rigorous convergence proof for GFLCC. The reason is structural: the coupled interaction of (i) finite but possibly large asynchronous delays, (ii) dual communication compression, and (iii) the conjugate historical states makes the construction of a watertight Lyapunov-decrease argument an *open problem*. The material below—the activation model, the deterministic sequence lemma, and a partial Lyapunov sketch—is therefore presented as a *stepping stone for future rigorous analysis*, not as a completed proof. Empirically, as demonstrated in Section 5, GFLCC consistently attains fast, robust convergence across the tested asynchronous benchmarks and network conditions, which strongly motivates the theoretical pursuit left open here.

Appendix F.1. Asynchronous Activation Model

We formalize the asynchronous operation of GFLCC. Let $k = 0, 1, 2, \dots$ denote the global iteration counter. At each iteration k, exactly one agent $s(k) \in \mathcal{N}$ activates according to an independent Poisson process with rate $\lambda_{s(k)} > 0$. Define the indicator $I_i(k) = 1$ if $s(k) = i$, and $I_i(k) = 0$ otherwise. Let $r_i(k)$ be the number of times agent i has activated up to global iteration k. At its m-th activation, agent i consumes the received mass-increment packets (each exactly once) from its in-neighbor FIFO queues and executes the GFLCC update (Algorithm 3). The accumulated catch-up step size for agent i between its m-th and $(m+1)$-th activations is

$$\beta_i(m) = I_i(k) \sum_{t=r_i(k)}^{r_i(k+1)-1} \rho(t), \tag{A6}$$

where $\rho(t) > 0$ is the base step size at global iteration t, satisfying $\lim_{t\to\infty} \rho(t) = 0$ and $\sum_{t=0}^{\infty} \rho(t) = \infty$ (the standard Robbins–Monro conditions).

Assumption A1. *(Bounded Activation Window). There exist constants $\sigma_1, \sigma_2 > 0$ such that every agent activates at least once in any window of $\tau := \sigma_1 + \sigma_2$ consecutive global iterations. Moreover, for any $i, j \in \mathcal{N}$ and any $k \geq 0$, the local clock difference satisfies $|r_i(k) - r_j(k)| \leq \tau$.*

Define $\widehat{r}(k) := \max_{j\in\mathcal{N}} r_j(k)$. Assumption A1 implies that for any $t \geq 0$,

$$\widehat{r}(k+t) - \widehat{r}(k) \leq t\sigma_2 \leq t\tau. \tag{A7}$$

Throughout the analysis, τ serves as the analysis window: every agent updates at least once in any τ-length window, and no agent's local clock falls behind the fastest agent by more than τ activations.

Appendix F.2. A Deterministic Sequence Lemma

The following elementary lemma decouples the convergence argument from the specifics of asynchronous scheduling and is the key analytical tool for this appendix.

Lemma A9. *Let $\{a_k\}_{k\geq 0}$, $\{\eta_k\}_{k\geq 0}$, $\{\theta_k\}_{k\geq 0}$, and $\{\xi_k\}_{k\geq 0}$ be four real sequences satisfying*

$$a_{k+1} \leq a_k - \eta_k + \theta_k + \xi_k, \qquad \forall\, k \in \mathbb{N}, \tag{A8}$$

where $a_k \geq 0$, $\eta_k \geq 0$, $\theta_k \geq 0$ *for all* k, *and*

$$\sum_{k=0}^{\infty} \theta_k < \infty, \qquad \sum_{k=0}^{\infty} \xi_k \text{ exists and is finite.}$$

Then, the sequence $\{a_k\}$ *converges to a finite limit, and* $\sum_{k=0}^{\infty} \eta_k < \infty$.

Proof. Since $\eta_k \geq 0$, (A8) yields $a_{k+1} \leq a_k + \theta_k + \xi_k$. For any $m > n \geq 0$, iterating gives

$$a_m \leq a_n + \sum_{\ell=n}^{m-1} (\theta_\ell + \xi_\ell).$$

By convergence of $\sum \theta_k$ and $\sum \xi_k$, for any $\varepsilon > 0$ there exists $N \in \mathbb{N}$ such that $\sum_{\ell=n}^{m-1} (\theta_\ell + \xi_\ell) < \varepsilon$ for all $n \geq N$, $m > n$. Hence $a_m \leq a_n + \varepsilon$, which implies

$$\limsup_{k\to\infty} a_k \leq a_n + \varepsilon.$$

Taking the infimum over $n \geq N$ gives $\limsup_{k\to\infty} a_k \leq \liminf_{k\to\infty} a_k + \varepsilon$. Letting $\varepsilon \to 0^+$ establishes

$$\limsup_{k\to\infty} a_k \leq \liminf_{k\to\infty} a_k,$$

which proves convergence of $\{a_k\}$ to a finite limit.

Summing (A8) from $k = 0$ to N yields

$$\sum_{k=0}^{N} \eta_k \leq a_0 - a_{N+1} + \sum_{k=0}^{N} \theta_k + \sum_{k=0}^{N} \xi_k.$$

As $N \to \infty$, the right-hand side converges to $a_0 - \lim_{k\to\infty} a_k + \sum_{k=0}^{\infty} \theta_k + \sum_{k=0}^{\infty} \xi_k$, which is finite. Since $\eta_k \geq 0$, the partial sums are non-decreasing and bounded above, hence $\sum_{k=0}^{\infty} \eta_k$ converges. □

Appendix F.3. Lyapunov Analysis

Let $\mathbf{x}_i(k)$ denote agent i's decision variable at global iteration k, stacked as $\mathbf{x}(k) := [\mathbf{x}_1(k)^\top, \ldots, \mathbf{x}_n(k)^\top]^\top \in \mathbb{R}^{np}$, and let $\mathbf{x}^*$ be the unique optimal solution. We study the Lyapunov function

$$V(k) := \sum_{i=1}^{n} \|\mathbf{x}_i(k) - \mathbf{x}^*\|^2.$$

Consider a window of τ consecutive iterations starting at global iteration k. By Assumption A1, every agent activates at least once in this window. Expanding the squared norm:

$$\begin{aligned} V(k+\tau) &= \sum_{i=1}^{n} \Big\| \mathbf{x}_i(k) - \mathbf{x}^* + \sum_{u=k}^{k+\tau-1} \big(\mathbf{x}_i(u+1) - \mathbf{x}_i(u)\big) \Big\|^2 \\ &= V(k) + \sum_{i=1}^{n} \|\mathbf{x}_i(k+\tau) - \mathbf{x}_i(k)\|^2 \\ &\quad + 2 \sum_{i=1}^{n} \sum_{u=k}^{k+\tau-1} \big[\mathbf{x}_i(k) - \mathbf{x}^*\big]^\top \big[\mathbf{x}_i(u+1) - \mathbf{x}_i(u)\big]. \end{aligned} \tag{A9}$$

The GFLCC update at each activation combines: *(i)* gradient descent along the conjugate direction, *(ii)* compressed push-sum consensus, and *(iii)* catch-up step-size accumulation. Following the same bounding techniques as in the synchronous CC-GT

analysis (Section 4 and Appendix A), and incorporating the bounded-delay condition (Assumption A1) to handle stale neighbor information, we obtain the following key inequality (the detailed algebraic derivations are omitted for brevity, as they parallel the synchronous case with additional accounting for the asynchronous activation schedule and delay-bounded message staleness):

$$V(k+\tau) \leq V(k) - 2 \sum_{t=\widehat{r}(k)}^{\widehat{r}(k+\tau)-1} \rho(t) \left[\mathbf{x}(k) - \mathbf{x}^*\right]^\top \left[\nabla f(\mathbf{x}(k)) - \nabla f(\mathbf{x}^*)\right] + \Theta(k) + \Delta(k), \tag{A10}$$

where $\Theta(k)$ and $\Delta(k)$ are non-negative error terms satisfying

$$\sum_{k=0}^{\infty} \Theta(k) < \infty, \qquad \sum_{k=0}^{\infty} \Delta(k) \text{ exists and is finite.}$$

The error term $\Theta(k)$ encapsulates the consensus discrepancy, compression distortion, gradient tracking error, and conjugate direction deviation—all of which are *asserted* to be summable under the same parameter conditions that guarantee linear convergence in the synchronous case [45]. The bounded-delay condition ensures that these error terms do not accumulate unboundedly: the step-size accumulation $\beta_i(m)$ in (A6) remains proportional to the true elapsed iterations, and stale neighbor states are at most τ steps behind the activating agent.

Open gaps (why this is a sketch, not a proof). The expansion above is only the first step. To turn it into a convergence proof, one would need to (a) derive the *summability* of $\Theta(k)$ and the existence of $\sum \Delta(k)$ explicitly, rather than assert them; (b) prove a uniform positive lower bound on the accumulated step size $\beta_i(m)$ within each bounded activation window—the Robbins–Monro schedule does *not* a priori guarantee this inside a fixed window; and (c) rigorously justify the passage from the stochastic Poisson activation model to the deterministic bounded-delay window of Assumption A1. None of these three steps is completed in the present manuscript; they are stated here as the precise obstacles that any future proof must overcome.

By the μ-strong convexity of f (Assumption 2), the descent term satisfies

$$\left[\mathbf{x}(k) - \mathbf{x}^*\right]^\top \left[\nabla f(\mathbf{x}(k)) - \nabla f(\mathbf{x}^*)\right] \geq \mu \|\mathbf{x}(k) - \mathbf{x}^*\|^2 \geq 0. \tag{A11}$$

Appendix F.4. Checkpoint Subsequence and Application of Lemma A9

Define the checkpoint subsequence at multiples of the analysis window:

$$\mathbf{y}(m) := \mathbf{x}(m\tau), \qquad m = 0, 1, 2, \ldots$$

Evaluating (A10) at $k = m\tau$ yields

$$\|\mathbf{y}(m+1) - \mathbf{x}^*\|^2 \leq \|\mathbf{y}(m) - \mathbf{x}^*\|^2 - \eta_m + \theta_m + \xi_m, \tag{A12}$$

where we have identified

$$\eta_m := 2 \sum_{t=\widehat{r}(m\tau)}^{\widehat{r}((m+1)\tau)-1} \rho(t) \left[\mathbf{y}(m) - \mathbf{x}^*\right]^\top \left[\nabla f(\mathbf{y}(m)) - \nabla f(\mathbf{x}^*)\right] \geq 0 \quad \text{(by (A11))},$$

$$\theta_m := \Theta(m\tau) > 0, \qquad \xi_m := \Delta(m\tau).$$

The summability properties are preserved at the checkpoint scale:

$$\sum_{m=0}^{\infty} \theta_m = \sum_{k=0}^{\infty} \Theta(k) < \infty, \qquad \sum_{m=0}^{\infty} \xi_m \text{ exists and is finite.}$$

Now set $a_m := \|\mathbf{y}(m) - \mathbf{x}^*\|^2$ and apply Lemma A9 to (A12). We conclude:

(i) The sequence $\{\|\mathbf{y}(m) - \mathbf{x}^*\|\}_{m\geq 0}$ converges to a finite limit.
(ii) $\sum_{m=0}^{\infty} \eta_m < \infty$, which implies $\lim_{m\to\infty} \eta_m = 0$.

From (A11), we have the lower bound

$$\eta_m \geq 2\mu \|\mathbf{y}(m) - \mathbf{x}^*\|^2 \sum_{t=\widehat{r}(m\tau)}^{\widehat{r}((m+1)\tau)-1} \rho(t).$$

The accumulated step size over each τ-window is bounded below by a positive constant (since at least one activation occurs per window and $\rho(t) > 0$). Combined with $\eta_m \to 0$, this *would* force

$$\lim_{m\to\infty} \|\mathbf{y}(m) - \mathbf{x}^*\| = 0.$$

Conditional status. The conclusion above holds *only if* the open gaps identified in Appendix F.3 are first resolved (explicit summability of $\Theta(k), \Delta(k)$, a uniform in-window step-size lower bound, and a rigorous Poisson-to-deterministic-window justification). As those gaps remain open, the checkpoint argument is presented here as a roadmap, not as a theorem.

Appendix F.5. Discussion and Open Problems

Withdrawn theorem. We do *not* state a convergence theorem for GFLCC. As explained in the Motivation and made precise by the open gaps in Appendix F.3, the coupled effects of unbounded asynchronous delay, dual compression, and conjugate historical states prevent a watertight Lyapunov-decrease proof in the present setting. Consequently, the Lemma A9 (Deterministic Sequence Lemma) developed in Appendix F.2, together with the activation model of Appendix F.1 and the partial Lyapunov sketch of Appendixes F.3 and F.4, are offered as a *stepping stone for future rigorous analysis*, not as a completed proof.

Positioning of GFLCC. GFLCC is therefore positioned as a *heuristically-motivated, empirically validated asynchronous extension* of the rigorously proven synchronous CC-GT, rather than as an algorithm with an established convergence-rate guarantee. Its remaining theoretical limitations are:

- A watertight Lyapunov-decrease proof for asynchronous compressed conjugate gradient tracking remains an *open problem*, owing to the coupled effects of finite-but-possibly large asynchronous delays, dual compression, and conjugate historical states detailed in Appendix F.3.
- The analysis would rely on a bounded-delay assumption (Assumption A1), standard in the asynchronous optimization literature [51,52,68], which may be violated in extremely heterogeneous environments.
- The abundant empirical evidence in Section 5—where GFLCC consistently outperforms state-of-the-art asynchronous baselines across diverse tasks and network conditions—shows that the practical convergence behavior is in fact considerably more favorable than what the (absent) theory would capture. Closing the gap between this empirical success and a rigorous guarantee is the main direction for future work.

References

1. Luo, B.; Jiang, B.; Xu, H. Demand-Driven Hierarchical Task Allocation in UAV Swarms: A Distributed ADMM Optimization Approach. In Proceedings of the 2025 IEEE 7th International Conference on Communications, Information System and Computer Engineering (CISCE), Guangzhou, China, 9–11 May 2025; pp. 571–576.
2. Mishchenko, K.; Gorbunov, E.; Takáč, M.; Richtárik, P. Distributed Learning with Compressed Gradient Differences. *Optim. Methods Softw.* **2025**, *40*, 1181–1196. [CrossRef]
3. Qin, C.; Pournaras, E. Strategic Coordination of Drones via Short-Term Distributed Optimization and Long-Term Reinforcement Learning. *Appl. Soft Comput.* **2025**, *186* , 114270. [CrossRef]
4. Casao, S.; Serra-Gómez, Á.; Murillo, A.C.; Böhmer, W.; Alonso-Mora, J.; Montijano, E. Distributed Multi-Target Tracking and Active Perception with Mobile Camera Networks. *Comput. Vis. Image Underst.* **2024**, *238*, 103876. [CrossRef]
5. Sarkar, R.; Gao, J. Differential Forms for Target Tracking and Aggregate Queries in Distributed Networks. In Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, Chicago, IL, USA, 20–24 September 2010; pp. 377–388.
6. Chen, Q.; Jin, H.; Wei, C.; Liao, Y. Distributed Impulsive Satellite Swarm-Keeping Based on Connectivity and Relative Cartesian States. *J. Guid. Control Dyn.* **2025**, *48*, 269–281. [CrossRef]
7. Asri, E.G.; Zhu, Z.H. Bioinspired Consensus-Based Spacecraft Swarm Control for Autonomous Capture of Uncooperative Targets. *Acta Astronaut.* **2025**, *234*, 419–428. [CrossRef]
8. Wang, J.; Yao, L.; Liang, J.; Wang, J.; Cheng, F. Distributed Optimization Strategy for Networked Microgrids Based on Network Partitioning. *Appl. Energy* **2025**, *378*, 124834. [CrossRef]
9. Bhattarai, B.P.; Myers, K.S.; Bak-Jensen, B.; de Cerio Mendaza, I.D.; Turk, R.J.; Gentle, J.P. Optimum Aggregation of Geographically Distributed Flexible Resources in Strategic Smart-Grid/Microgrid Locations. *Int. J. Electr. Power Energy Syst.* **2017**, *92*, 193–201. [CrossRef]
10. Mohammed, A.F.Y.; Allawi, Y.M.; Moneer, E.M.; Widaa, L.O. Collaborative Split Learning-Based Dynamic Bandwidth Allocation for 6G-Grade TDM-PON Systems. *Sensors* **2025**, *25*, 4300. [CrossRef] [PubMed]
11. Wang, H.; Song, L. A Distributed End-to-End Fair Bandwidth Allocation Algorithm for Multi-Path Networks. *Comput. Electr. Eng.* **2024**, *120*, 109635. [CrossRef]
12. Wei, Y.; Hu, T.; Liang, C.; Cui, Y. Communication Optimization for Distributed Training: Architecture, Advances, and Opportunities. *IEEE Netw.* **2024**, *39*, 241–248. [CrossRef]
13. Liang, F.; Zhang, Z.; Lu, H.; Leung, V.; Guo, Y.; Hu, X. Communication-Efficient Large-Scale Distributed Deep Learning: A Comprehensive Survey. *arXiv* **2024**, arXiv:2404.06114.
14. Koloskova, A.; Stich, S.; Jaggi, M. Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019; pp. 3478–3487.
15. Lian, X.; Zhang, W.; Zhang, C.; Liu, J. Asynchronous Decentralized Parallel Stochastic Gradient Descent. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 3043–3052.
16. Lian, X.; Huang, Y.; Li, Y.; Liu, J. Asynchronous Parallel Stochastic Gradient for Nonconvex Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2015; Volume 28, pp. 2737–2745.
17. Nedic, A.; Ozdaglar, A. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Trans. Autom. Control* **2009**, *54*, 48–61. [CrossRef]
18. Jakovetić, D.; Xavier, J.; Moura, J.M.F. Fast Distributed Gradient Methods. *IEEE Trans. Autom. Control* **2014**, *59*, 1131–1146. [CrossRef]
19. Yuan, K.; Ling, Q.; Yin, W. On the Convergence of Decentralized Gradient Descent. *SIAM J. Optim.* **2016**, *26*, 1835–1854. [CrossRef]
20. Qu, G.; Li, N. Harnessing Smoothness to Accelerate Distributed Optimization. *IEEE Trans. Control Netw. Syst.* **2017**, *5*, 1245–1260. [CrossRef]
21. Nedic, A.; Olshevsky, A.; Shi, W. Achieving Geometric Convergence for Distributed Optimization over Time-Varying Graphs. *SIAM J. Optim.* **2017**, *27*, 2597–2633. [CrossRef]
22. Pu, S.; Nedić, A. Distributed Stochastic Gradient Tracking Methods. *Math. Program.* **2021**, *187*, 409–457. [CrossRef]
23. Li, B.; Cen, S.; Chen, Y.; Chi, Y. Communication-Efficient Distributed Optimization in Networks with Gradient Tracking and Variance Reduction. *J. Mach. Learn. Res.* **2020**, *21*, 1–51.
24. Xin, R.; Pu, S.; Nedić, A.; Khan, U.A. A General Framework for Decentralized Optimization with First-Order Methods. *Proc. IEEE* **2020**, *108*, 1869–1889. [CrossRef]
25. Gao, J.; Liu, X.-W.; Dai, Y.-H.; Huang, Y.; Gu, J. Distributed stochastic gradient tracking methods with momentum acceleration for non-convex optimization. *Comput. Optim. Appl.* **2023**, *84*, 531–572. [CrossRef]
26. Chen, E.; Wang, S.; Brinton, C.G. Taming Subnet-Drift in D2D-Enabled Fog Learning: A Hierarchical Gradient Tracking Approach. In *Proceedings of the IEEE INFOCOM 2024*; IEEE: New York, NY, USA, 2024; pp. 2438–2447.

27. Ma, W.; Xu, Z.; Zhao, J.; Zhang, S.; Shao, Z. Cooperative Distributed Extremum Seeking Control for Coupled Multiagent Systems Based on Distributed Identification-Gradient Tracking. *Int. J. Adapt. Control Signal Process.* **2023**, *37*, 2627–2654. [CrossRef]
28. Li, Q.; Gan, R.; Godsill, S.J. Decentralised Variational Inference Frameworks for Multi-Object Tracking on Sensor Networks. *IEEE Trans. Signal Process.* **2025**.
29. Peng, X.-H.; Sun, Y.-X.; Zhang, Z.-H.; Wang, Y.-F. DSparse: A Distributed Training Method for Edge Clusters Based on Sparse Update. *J. Comput. Sci. Technol.* **2025**, *40*, 637–653. [CrossRef]
30. Li, J.; Michelogiannakis, G.; Maloney, S.; Cook, B.; Suarez, E.; Shalf, J.; Chen, Y. Job Scheduling in High Performance Computing Systems with Disaggregated Memory Resources. In *Proceedings of the 2024 IEEE International Conference on Cluster Computing (CLUSTER)*; IEEE: New York, NY, USA, 2024; pp. 297–309.
31. Li, H.; Lin, Z. Accelerated Gradient Tracking over Time-Varying Graphs for Decentralized Optimization. *J. Mach. Learn. Res.* **2024**, *25*, 1–52.
32. Xin, R.; Jakovetić, D.; Khan, U.A. Distributed Nesterov Gradient Methods over Arbitrary Graphs. *IEEE Signal Process. Lett.* **2019**, *26*, 1247–1251. [CrossRef]
33. Rajabi, S.; Nonta, N.; Rambhatla, S. Subtrack Your Grad: Gradient Subspace Tracking for Memory and Time Efficient Full-Parameter LLM Training. *arXiv* **2025**, arXiv:2502.01586.
34. Yi, X.; Zhang, S.; Yang, T.; Chai, T.; Johansson, K.H. Communication Compression for Distributed Nonconvex Optimization. *IEEE Trans. Autom. Control* **2022**, *68*, 5477–5492. [CrossRef]
35. Yang, A.; Liu, J.; Yang, B.; Mo, Z.; Li, K. Balancing Communication Overhead and Accuracy in Compression Integration: A Survey. *J. Supercomput.* **2025**, *81*, 964. [CrossRef]
36. Aji, A.F.; Heafield, K. Sparse Communication for Distributed Gradient Descent. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 440–445.
37. Xu, H.; Ho, C.-Y.; Abdelmoniem, A.M.; Dutta, A.; Bergou, E.H.; Karatsenidis, K.; Canini, M.; Kalnis, P. *Compressed Communication for Distributed Deep Learning: Survey and Quantitative Evaluation*; KAUST Department: Thuwal, Saudi Arabia, 2020.
38. Safaryan, M.; Shulgin, E.; Richtárik, P. Uncertainty Principle for Communication Compression in Distributed and Federated Learning and the Search for an Optimal Compressor. *Inf. Inference* **2022**, *11*, 557–580. [CrossRef]
39. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2017; Volume 30.
40. Stich, S.U.; Cordonnier, J.-B.; Jaggi, M. Sparsified SGD with Memory. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2018; Volume 31.
41. Vogels, T.; Karimireddy, S.P.; Jaggi, M. PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2019; Volume 32.
42. Wangni, J.; Wang, J.; Liu, J.; Zhang, T. Gradient Sparsification for Communication-Efficient Distributed Optimization. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2018; Volume 31.
43. Cheng, Y.; Shen, L.; Xu, L.; Qian, X.; Wu, S.; Zhou, Y.; Zhang, T.; Tao, D.; Chen, E. Communication-Efficient Distributed Learning with Local Immediate Error Compensation. *Neural Netw.* **2025**, *188*, 107471. [CrossRef] [PubMed]
44. Karimireddy, S.P.; Rebjock, Q.; Stich, S.; Jaggi, M. Error Feedback Fixes SignSGD and Other Gradient Compression Schemes. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019; pp. 3252–3261.
45. Liao, Y.; Li, Z.; Huang, K.; Pu, S. A Compressed Gradient Tracking Method for Decentralized Optimization with Linear Convergence. *IEEE Trans. Autom. Control* **2022**, *67*, 5622–5629. [CrossRef]
46. Islamov, R.; Gao, Y.; Stich, S.U. Towards Faster Decentralized Stochastic Optimization with Communication Compression. In Proceedings of the International Conference on Learning Representations (ICLR), Singapore, 4–28 April 2025.
47. Wang, Z.; Wen, M.; Xu, Y.; Zhou, Y.; Wang, J.H.; Zhang, L. Communication Compression Techniques in Distributed Deep Learning: A Survey. *J. Syst. Archit.* **2023**, *142*, 102927. [CrossRef]
48. Zhao, H.; Li, B.; Li, Z.; Richtárik, P.; Chi, Y. BEER: Fast $O(1/T)$ Rate for Decentralized Nonconvex Optimization with Communication Compression. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2022; Volume 35, pp. 31653–31667.
49. Phuong, T.T.; Phong, L.T. New Convergence Analysis of the BEER Algorithm in Decentralized Nonconvex Optimization. In *Proceedings of the International Conference on Complex, Intelligent, and Software Intensive Systems*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 219–230.
50. Trieu Phong, L.; Phuong, T.T.; Fukushima, K. signBEER: Decentralized Optimization with Improved Communication by Signs. In *Proceedings of the Conference on Information Technology and its Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 16–27.

51. Wu, Z.; Ling, Q.; Tian, Z.; Sun, Y. Asynchronous Decentralized Optimization with Gradient Tracking. *IEEE Trans. Signal Process.* **2023**, *71*, 2504–2518. [CrossRef]
52. Tian, Y.; Sun, Y.; Scutari, G. Asynchronous Decentralized Optimization with Gradient Tracking and Variance Reduction. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
53. Kempe, D.; Dobra, A.; Gehrke, J. Gossip-Based Computation of Aggregate Information. In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Cambridge, MA, USA, 1–14 October 2003; pp. 482–491.
54. Cao, H.; Hu, Q.; Jiang, H.; Ge, S.S.; Li, D. Predefined-Sequential-Synchronized Control with Exploration of Event-Triggered Mechanism. *Automatica* **2026**, *183*, 112645. [CrossRef]
55. Yang, X.; Zhang, X.; Cao, J.; Liu, H. Observer-Based Adaptive Fuzzy Fractional Backstepping Consensus Control of Uncertain Multiagent Systems via Event-Triggered Scheme. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 3953–3967. [CrossRef]
56. Zhang, H.; Zhang, D.; Wen, J.; Shen, S.; Fu, L.; Liu, W.; Song, A. A Novel FTP+d Control Approach With Event-Driven Communication for Time-Delayed Bilateral Teleoperation Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, *56*, 428–442. [CrossRef]
57. Dai, W.; Cui, T.; Zhang, T.; Chen, B. VFL+: Low-Coupling Vertical Federated Learning With Privileged Information Paradigm. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 3533–3547. [CrossRef]
58. Shi, Y.; Lin, C.-T.; Chang, Y.-C.; Ding, W.; Shi, Y.; Yao, X. Consensus Learning for Distributed Fuzzy Neural Network in Big Data Environment. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *5*, 29–41. [CrossRef]
59. Hu, Y.; Xu, X.; Duan, L.; Bilal, M.; Wang, Q.; Dou, W. End-Edge Collaborative Inference of Convolutional Fuzzy Neural Networks for Big Data-Driven Internet of Things. *IEEE Trans. Fuzzy Syst.* **2024**, *33*, 203–217. [CrossRef]
60. Zhang, Z.; Yu, M.; Jia, H. Decentralized Nonstationary Fuzzy Neural Network with Meta-Learning-Net. *Mathematics* **2025**, *13*, 552. [CrossRef]
61. Xun, L.; Hu, M.; Zhao, H.; Singh, A.K.; Hare, J.; Merrett, G.V. Fluid Dynamic DNNs for Reliable and Adaptive Distributed Inference on Edge Devices. In *Proceedings of the 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*; IEEE: New York, NY, USA, 2024; pp. 1–2.
62. Zhao, J.; Chang, D.; Cao, B.; Liu, X.; Lyu, Z. Multiobjective Evolution of the Deep Fuzzy Rough Neural Network. *IEEE Trans. Fuzzy Syst.* **2024**, *33*, 242–254. [CrossRef]
63. Ye, C.; Yao, Y.; Lin, W.; Yang, X.; Qiu, J. CSGrasp: Category-Level Semantically-Aware Grasping Method. *IEEE Trans. Ind. Electron.* **2026**, *73*, 2610–2619. [CrossRef]
64. Li, Y.; Zhang, F.; Zhang, Y.; Zhang, S.; Hu, B.; Wang, J.; Ding, X. ComStar: Compression-Aware Stream Query for Heterogeneous Hybrid Architecture. *IEEE Trans. Parallel Distrib. Syst.* **2026**, *37*, 948–965. [CrossRef]
65. Peng, X.; Qin, C.; Zhang, X.; Xu, L.; Zhang, X.; Jiang, L. DRUDM-CFG: A Fairness-Aware Multi-Agent DRL Algorithm for AMEC-Assisted Task Offloading in Post-Disaster Scenarios. *IEEE Trans. Mob. Comput.* **2026**, *25*, 14135–14150. [CrossRef]
66. Jin, L.; Zhang, Y.; Li, Y.; Wang, S.; Yang, H.H.; Wu, J.; Zhang, M. MoE^2: Optimizing Collaborative Inference for Edge Large Language Models. *IEEE Trans. Netw.* **2026**, *34*, 4637–4651. [CrossRef]
67. Shorinwa, O.; Schwager, M. Distributed Conjugate Gradient Method via Conjugate Direction Tracking. In *2024 American Control Conference (ACC)*; IEEE: New York, NY, USA, 2024; pp. 2066–2073.
68. Agarwal, A.; Duchi, J.C. Distributed Delayed Stochastic Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2012; Volume 24, pp. 873–881.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.