CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

A Quantised Push-Sum Distributed Adaptive Momentum Algorithm for Optimisation Over Directed Networks

Qingguo Lü[1,2] | Mingyu Guan[1] | Chenglong He[1] | Chen Chen[2] | Xiangguang Dai[1] | Man-Fai Leung[3] | Wei Zhang[4]

[1]School of Computer Science and Engineering, Chongqing Three Gorges University, Chongqing, China | [2]College of Computer Science, Chongqing University, Chongqing, China | [3]School of Computing and Information Science, Faculty of Science and Engineering, Anglia Ruskin University, Chelmsford, UK | [4]College of Computer Science, Chongqing Polytechnic University of Electronic Technology, Chongqing, China

Correspondence: Xiangguang Dai (daixiangguang@163.com)

Received: 3 September 2025 | **Revised:** 20 November 2025 | **Accepted:** 10 December 2025

Keywords: adaptive momentum method | directed networks | distributed constrained optimisation | quantised communication | sublinear convergence

ABSTRACT

In this paper, we investigate a distributed constrained optimisation problem over directed networks. The agents in the networks conduct local computations and communications, endeavouring to collaboratively minimise the aggregation of all locally known convex cost functions subject to a global constraint set. However, since the agents are constantly transmitting information, most existing algorithms for this problem are prone to communication burdens, especially in large-scale networks under a limited communication bandwidth. Problems of this nature emerge in a number of applications, mostly evident in distributed classification tasks, distributed image restoration, distributed compressive sensing etc. To solve these kinds of problems, we propose an effective quantised push-sum distributed adaptive momentum (QPS-DADAM) algorithm. On the one hand, the QPS-DADAM algorithm employs the random quantiser to reduce the communication overhead and avoid the channel blockage. On the other hand, the QPS-DADAM algorithm incorporates the adaptive momentum method into the push-sum protocol to further accelerate the convergence over directed networks. Rigorous theoretical analyses are provided to illustrate that the QPS-DADAM algorithm converges sublinearly to the optimal solution. In addition, numerical simulations further demonstrate the efficacy of the QPS-DADAM algorithm and the correctness of the theoretical discoveries.

1 | Introduction

In the rapid advancement of modern technological landscapes, distributed optimisation algorithms have emerged as a linchpin in diverse applications, spanning economic dispatch [1, 2], signal processing [3, 4] and smart grids [5, 6]. However, the massive data transmission pressure, uneven agent computing capacity and real-time response requirements of real-world scenarios impose escalating demands on algorithmic robustness. Traditional centralised frameworks, heavily dependent on central entities for information aggregation, inherently suffer from critical limitations, including single-point failures, exorbitant synchronisation overheads, prohibitive long-distance communication costs and constrained scalability. In contrast, distributed optimisation paradigms address these challenges through collaborative multi-agent communication and parallel processing of large-scale computational tasks.

Literature review: Distributed optimisation, as a key method for solving collaborative decision-making problems in large-scale complex systems, has attracted extensive attention. The distributed subgradient descent method (DGD) proposed by Nedić and Ozdaglar [7] achieves the minimisation of the problem function through local information interaction between neighbouring agents, establishing a foundational framework for the design and analysis of distributed algorithms. However, the physical limitation of communication bandwidth in practical scenarios has become a prominent bottleneck. Most existing

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

https://doi.org/10.1049/cit2.70122

studies assume ideal bandwidth, yet data transmission in large-scale distributed systems often results in severe congestion. To address this, quantised communication technology has emerged as an important solution: By discretising continuous parameters into a finite set of discrete values, it reduces data volume while preserving key information, effectively lowering communication loads and driving research on distributed optimisation under quantised communication [8–12]. Xiong et al. [8] proposed a distributed quantised mirror descent algorithm for strongly convex optimisation over time-varying directed graphs, ensuring sublinear convergence in non-Euclidean spaces. Building on this foundation, Yuan et al. [9] investigated distributed online bandit optimisation under random quantization, developing an effective algorithm that achieves stable performance guarantees. Complementing these advances, Cao et al. [10] presented a comprehensive overview of communication-efficient distributed learning, systematically summarising the research progress and application prospects of quantised communication and other key technologies. Shifting focus to precision control, Pu et al. [11] proposed an iterative refinement quantisation mechanism that controls communication errors by dynamically adjusting precision, enabling linear convergence for unconstrained problems. Similarly, in a related vein, Riko et al. [12] alternated between gradient updates and quantised averaging to approximate centralised performance. Zhang et al. [13] proposed the Q-DOPFO algorithm, which balances quantisation precision and dynamic regret in time-varying networks while verifying the effectiveness of quantised communication in complex scenarios. However, existing quantised optimisation algorithms for directed networks often face trade-offs: For example, Nedić et al. [14] showed that some achieve convergence within a bounded error but with suboptimal rates, whereas Taheri et al. [15] noted that others rely on strict symmetry in quantisation precision, which is an impractical requirement in large-scale asymmetric scenarios.

Quantised communication eases bandwidth limits, but its performance depends on network topology, especially the asymmetric information flow in directed networks, which are challenges to convergence. Nedić et al. [14] first proved that quantised systems converge to the global mean within the bounded error. Alistarh et al. [16] and Reisizadeh et al. [17, 18] built convergence frameworks for master-slave and undirected networks, showing that constant-variance unbiased quantisers achieve optimal rates in convex and nonconvex settings. Xiong et al. [8] ensured sublinear convergence in non-Euclidean spaces by combining mirror descent with quantisation. For directed networks, Taheri et al. [15] proposed the QPS-DO algorithm, merging push-sum with quantised communication to reduce overhead and unify diverse topologies.

Beyond quantisation and topological adaptations, the learning rate design is crucial for mitigating quantisation-induced convergence fluctuations, as it directly influences the stability and speed of distributed optimisation. Adaptive momentum methods dynamically adjust learning rates using historical gradient information to reduce fluctuations and avoid local optima, and they have evolved from centralised to distributed optimisation contexts. In centralised settings, Duchi et al. [19] introduced an adaptive subgradient method for sparse data. Additionally, Kingma, Ba et al. [20] proposed the ADAM algorithm, which combines momentum and adaptivity while delivering strong performance for nonconvex problems. Meanwhile, Reddi et al. [21] corrected the convergence issues of the ADAM algorithm to ensure optimisation stability. In distributed systems, Okamoto et al. [22] introduced event-triggered adaptive gradients to reduce communication overhead. Shen et al. [23] developed adaptive online gradients with weight decay to enhance optimisation robustness. Nazari et al. [24] proposed the DADAM algorithm, a distributed online optimisation algorithm with adaptive momentum estimation that incorporates the advantages of the ADAM algorithm. This algorithm achieves smoother convergence, wider applicability and better performance than distributed algorithms without adaptive momentum by leveraging adaptive first-order and second-order momentum estimation. However, it cannot address bandwidth-limited scenarios or alleviate tight communication resource constraints, and it relies on doubly stochastic weight matrices. Zhang et al. [25] also designed adaptive ADMM-based methods. These works collectively confirm the value of adaptive momentum in distributed optimisation. Yet, the integration of adaptive momentum and quantised communication remains limited. Existing methods either ignore quantisation errors [22] or fail to balance momentum with quantised propagation [24], which reduces efficiency in bandwidth-limited directed networks. This paper advances the integration of quantisation, push-sum and adaptive momentum: It employs random quantisers to save bandwidth, adopts the push-sum protocol to maintain consistency in directed networks and leverages adaptive momentum to balance error and speed. This approach ultimately improves the efficiency of distributed optimisation.

Notably, recent breakthroughs in neural network modelling and meta-heuristic algorithms have spawned a series of innovative and high-impact studies that deliver remarkable practical value across diverse domains. Muhammad et al. [26] proposed a novel integration of attention-based LSTM with dilated CNN features for human action recognition. This design achieves highly robust performance by effectively capturing discriminative spatial-temporal features and outperforms traditional architectures. Travin et al. [27] developed an efficient and reliable statistical computational model. This model accurately predicts the composition of fission products, thereby providing strong technical support for the nuclear safety monitoring of nuclear power plants. Kiran et al. [28] presented an innovative tree-seed algorithm equipped with intelligent search mechanisms. This algorithm excels in continuous optimisation by better balancing exploration and exploitation, avoiding local optima more effectively than conventional methods. Zavadskas et al. [29] designed a practical intelligent expert system by integrating molecular fuzzy genetic algorithms with multi-problem particle swarm optimisation. This system offers well-structured decision support for circular-oriented project investments and effectively aligns economic returns, environmental benefits and resource efficiency. These outstanding works not only advance the theoretical frontier of their respective fields but also lay a solid foundation for practical applications in key areas such as action recognition, nuclear safety and circular economy development.

Drawing from the aforementioned discussions, the primary emphasis of this work is to devise an effective distributed algorithm capable of tackling general constrained optimisation problems over directed networks with limited communication

bandwidth. In detail, the core contributions are highlighted as follows:

1. An efficient quantised push-sum distributed adaptive momentum algorithm, named the QPS-DADAM algorithm, is proposed for the general constrained optimisation problems under the limited communication bandwidth of directed networks. Rigorous theoretical analysis demonstrates that the QPS-DADAM algorithm converges to the global optimal solution and achieves a sublinear convergence rate. Extensive simulations on real-world applications illustrate the efficacy of the QPS-DADAM algorithm.
2. Different from the algorithms in ref. [24], the QPS-DADAM algorithm considers quantised communication using a random quantiser, which reduces the communication overhead and avoids channel blocking when confronting a limited communication bandwidth in realistic communication environments. In addition, the QPS-DADAM algorithm is based on the push-sum strategy [30], which can effectively overcome the unbalanceness induced by the directed networks, that is, reaching a consensus at the optimal solution.
3. Motivated by the algorithms in ref. [15], but unlike them, the QPS-DADAM algorithm considers employing the adaptive momentum method [21, 24] to accelerate the convergence and reduce sensitivity to step sizes, thus achieving substantial performance improvement. In addition, the QPS-DADAM algorithm is implemented in a decentralised manner since no additional centres are required, alleviating much computational overhead.

Contributions compared with our past related works: Our prior work includes three works closely relevant to this study, and the QPS-DADAM algorithm achieves targeted enhancements for each: Compared with the Q-DADAM algorithm in ref. [31], the QPS-DADAM algorithm incorporates the push-sum protocol for directed networks, projected gradient descent for constrained optimisation and a dynamic quantisation strategy to balance bandwidth and precision. Different from the distributed stochastic subgradient-push algorithm in ref. [32], the QPS-DADAM algorithm integrates an ADAM-based adaptive momentum mechanism to accelerate convergence and reduce step-size sensitivity along with a dynamic quantised communication strategy and a projected gradient descent for constrained scenarios. Unlike the distributed dual averaging push algorithm in ref. [33], the QPS-DADAM algorithm incorporates an adaptive momentum mechanism, adopts a distinct projection approach and introduces a dynamic quantised communication strategy to save bandwidth.

Organisation: Section 2 introduces the related foundational theories. Section 3 designs the QPS-DADAM algorithm. Section 4 presents the convergence analysis of this algorithm. Section 5 evaluates the performance of the QPS-DADAM algorithm through various simulation experiments. Section 6 concludes this work.

Notation: For $\mathbf{x} = [x_1, ..., x_p]^{\mathrm{T}} \in \mathbb{R}^p$, $\mathbf{x} \in [a, b]^p$ means each $x_j \in [a, b]$, with $\mathbf{0}_p$ and $\mathbf{1}_p$ as all-zeros and all-ones vectors. $[N] = \{1, ..., N\}$ denotes integers from 1 to N. For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^p$, $\langle \mathbf{a}, \mathbf{b} \rangle$ is the Euclidean inner product, $\mathbf{a} \odot \mathbf{b}$ the element-wise product. For $\mathbf{A} \in \mathbb{R}^{p \times p}$, norms are $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_2$ and $\|\mathbf{A}\|_\infty$. The closed ball of radius R is $\mathbb{B}_R^d \triangleq \{\boldsymbol{x} \in \mathbb{R}^d | \, \|\boldsymbol{x}\| \le R\}$, $[\mathbf{A}_{ij}]$ is the (i, j) entry and $\sigma_i(\mathbf{A})$ the i-th eigenvalue. The projection operator onto a set $\mathcal{X}$ with respect to a positive semi-definite matrix $\mathbf{A}$ is defined as $\Pi_{\mathcal{X},\mathbf{A}}[\mathbf{x}] = \operatorname{argmin}_{y \in \mathcal{X}} \left\| \mathbf{A}^{\frac{1}{2}}(\mathbf{x} - \mathbf{y}) \right\|$. Agent i's vector at step t is $\mathbf{x}_{i,t}$, component d is $x_{i,t,d}$. $\operatorname{diag}(\mathbf{a})$ has diagonal $a_1, ..., a_p$, and $\mathbb{E}[\mathbf{x}]$ is the expectation of $\mathbf{x}$. Additionally, we use the symbol $\tilde{O}(\cdot)$ to denote an expression in the $O(\cdot)$ notation with logarithmic factors omitted.

2 | Problem Statement

In this section, we elaborate on the general constrained optimisation problem and present the practical application scenarios to form a comprehensive problem statement.

2.1 | Problem Formulation

In the realm of networked control systems, distributed optimisation has emerged as a pivotal approach to enhance optimisation efficiency, where the core lies in its inter-agent collaborative computation and information interaction. In particular, this work concerns the constrained optimisation of the form below:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}), \ \text{s.t.} \ \mathbf{x} \in \mathcal{X}, \tag{1}$$

where $\mathbf{x}$ denotes the global decision variable, f_i represents the local convex loss function at agent i and $\mathcal{X}$ is a convex set satisfying $\mathcal{X} \subseteq \mathbb{B}_R^d$.

Assumption 1. Defining $\mathbf{g}_i(\cdot, \xi_i) = \nabla f_i(\cdot, \xi_i)$ as the stochastic gradient of i with ξ_i denotes a random sample, there exists a positive constant G_∞ such that $\left\| \mathbf{g}_i(\mathbf{x}; \xi_i) \right\|_\infty \le G_\infty$ for all i.

Assumption 2. For any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, there exists a positive constant γ_∞ such that $\|\mathbf{x} - \mathbf{y}\|_\infty \le \gamma_\infty$.

Remark 2.1. In practical applications, Assumption 1 bounds the magnitude of the stochastic gradient ($\mathbf{g}_i$) to ensure optimisation stability and prevent extreme updates. This assumption is widely adopted in the analysis of gradient-based algorithms [34–36] for stochastic optimisation and is satisfied by a broad spectrum of functions, such as the Huber and Geman-McClure loss functions employed in robust optimisation [37] and the negative log-likelihood function in logistic regression [38]. Assumption 2 further bounds the constraint set $\mathcal{X}$ to enhance stability and support convergence analysis.

2.2 | Network Structure

The inter-agent communication within the system can be characterised as a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n intelligent agents, where the vertice set is $\mathcal{V} = \{1, 2, ..., n\}$ and

the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes directed connections in the network, that is, $(j, i) \in \mathcal{E}$ indicates that agent j can send information to agent i. The weight adjacency matrix of the network is denoted by $\mathbf{A} \in \mathbb{R}^{n \times n}$ with entries $[\mathbf{A}_{ij}] \in [0, 1]$, where $[\mathbf{A}_{ij}] > 0$ if there is a directed edge from agent j to agent i, and $[\mathbf{A}_{ij}] = 0$ otherwise. The weight $[\mathbf{A}_{ij}]$ quantifies the influence of agent j on agent i, and the in-neighbour of agent i is defined as $\mathcal{N}_i^{\text{in}} = \{j \in \mathcal{V} | [\mathbf{A}_{ij}] > 0\}$. The weight matrix $\mathbf{A}$ is column-stochastic if the normalisation condition $\sum_{i=1}^{n} [\mathbf{A}_{ij}] = 1$ holds for all $j \in \mathcal{V}$. Additionally, the network $\mathcal{G}$ is assumed to have a self-loop. Furthermore, the following assumptions are made regarding the network.

Assumption 3. The network $\mathcal{G}$ is directed and strongly connected. Furthermore, the matrix $\mathbf{A} \in [0, 1]^{n \times n}$ is column-stochastic and $[\mathbf{A}_{ii}] > 0$ for all $i \in \mathcal{V}$.

Remark 2.2. The self-loop ($[\mathbf{A}_{ii}] > 0$) in Assumption 3 is critical for distributed optimisation over directed networks. It maintains the column-stochastic property of matrix $\mathbf{A}$, preserves the local state of each agent to avoid over-reliance on neighbours and dilutes quantised communication noise to stabilise convergence.

2.3 | Application Examples

This subsection demonstrates the practical relevance of our research through two typical distributed constrained optimisation tasks, which can be rigorously formulated as distributed constrained optimisation problems over directed networks.

(a) Distributed image restoration for cross-regional visual transmission: A typical case of problem (1) in this paper is distributed constrained image restoration, which fits the theoretical framework of distributed constrained optimisation over time-invariant directed networks proposed herein. In practical scenarios, visual data are transmitted unidirectionally from edge cameras to regional agents and often degrade due to motion blur, Gaussian noise or compression artefacts. Raw high-resolution images generate heavy communication traffic, whereas bandwidth limits force low-bitrate transmission, and this leads to severe estimation deviations in unconstrained optimisation schemes. To address these challenges, the paper introduces a closed convex feasible set $\mathcal{X}$ to constrain pixel values within the physical range of images. Meanwhile, the QPS-DADAM algorithm is designed to balance communication efficiency and restoration accuracy, ensuring adaptability to bandwidth-constrained environments. From a mathematical modelling perspective, each agent i receives a degraded image $\mathbf{y}_i$ from its in-neighbour cameras. The image degradation process is modelled as $\mathbf{y}_i = \mathbf{H}_i\mathbf{x} + \mathbf{n}_i$, where $\mathbf{x}$ denotes the original clear image, $\mathbf{H}_i$ represents the degradation operator and $\mathbf{n}_i$ stands for zero-mean additive white Gaussian noise (AWGN). To collaboratively recover the original image $\mathbf{x}$, the local loss function for agent i is constructed as follows:

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{n} \left(\|\mathbf{H}_i\mathbf{x} - \mathbf{y}_i\|_2^2 + \frac{\lambda}{2n}\|\mathbf{x}\|_2^2 \right),$$

where $\mathcal{X}$ is defined as a convex set matching the physical value range of 8-bit image pixels, which restricts pixel values to a practically relevant interval. The regularisation parameter λ is introduced to suppress overfitting caused by noise. Each agent i holds multiple degraded versions of the same original image, with these degraded samples collected from different in-neighbour cameras. As designed for this collaborative restoration task, the local problem function corresponding to agent i is

$$f_i(\mathbf{x}) = \|\mathbf{H}_i\mathbf{x} - \mathbf{y}_i\|_2^2 + \frac{\lambda}{2n}\|\mathbf{x}\|_2^2.$$

The core goal is to collaboratively learn the optimal restored image $\mathbf{x}^*$ using distributed degraded data. The QPS-DADAM algorithm in the paper reduces communication overhead through $Q_t(\cdot)$-based quantisation, uses push-sum to handle unidirectional data flow in directed networks and leverages adaptive momentum to accelerate convergence.

(b) Distributed sparse signal compressive sensing for wireless sensor networks: The distributed collaborative sparse signal recovery task further illustrates problem (1) herein, serving as a specific application of distributed constrained optimisation over fixed directed networks. In industrial wireless sensor networks, a certain number of sensor agents are deployed, with a source emitting a sparse signal $\mathbf{x}$. Each sensor collects multiple groups of independent measurements, but limited battery capacity and communication bandwidth lead to rapid energy depletion when transmitting raw measurement vectors. Meanwhile, sensors only transmit data unidirectionally to the central agent, which necessitates the algorithm to adapt to directed network topologies. To address these challenges, the problem is modelled as an L_1-norm constrained sparse optimisation task. Here, the feasible set $\mathcal{X}$ is introduced to ensure the consistency of signal amplitude, whereas quantised communication technology is adopted to reduce the volume of data transmission, thereby alleviating bandwidth and energy constraints. From the perspective of measurement modelling, the measurement process of agent i follows the model $\mathbf{y}_{i,j} = \mathbf{O}_{i,j}\mathbf{x} + \mathbf{e}_{i,j}$. In this model, $\mathbf{y}_{i,j}$ denotes the j-th measurement obtained by agent i, $\mathbf{O}_{i,j}$ represents the i.i.d. Gaussian measurement matrix, and $\mathbf{e}_{i,j}$ stands for zero-mean Gaussian noise. The local loss function for this sparse signal recovery task is constructed as follows:

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{n} \left(\frac{1}{m} \sum_{j=1}^{m} \frac{1}{2}\|\mathbf{O}_{i,j}\mathbf{x} - \mathbf{y}_{i,j}\|_2^2 + \frac{\rho}{n}\|\mathbf{x}\|_1 \right),$$

where $\mathcal{X}$ is defined as a convex set that constrains signal amplitude within the physical range of the industrial equipment involved in the task. The regularisation parameter ρ is used to enhance the sparsity of the recovered signal and suppress the generation of nonzero elements caused by noise, which is designed based on the test requirements of the task. To align with problem (1), the local problem function for agent i is

$$f_i(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} \frac{1}{2}\|\mathbf{O}_{i,j}\mathbf{x} - \mathbf{y}_{i,j}\|_2^2 + \frac{\rho}{n}\|\mathbf{x}\|_1.$$

The core goal is to collaboratively recover the optimal sparse signal $\mathbf{x}^*$. The QPS-DADAM algorithm in the paper compresses

measurements via random quantisers, uses push-sum for unidirectional data flow and employs adaptive momentum to suppress noise-induced gradient fluctuations. Test results show a reconstructed signal error range of $[-0.02, 0.02]$, meeting industrial vibration monitoring requirements. In practice, the network satisfies Assumption 3 in the paper to ensure stable sensor consensus.

3 | Algorithm Design

This section elaborates on the design concept of the QPS-DADAM algorithm and compares it with the QPS-DO algorithm.

3.1 | The DADAM Algorithm

The DADAM algorithm utilises an adaptive momentum framework to guarantee convergence, initialised with $\mathbf{m}_{i,0} = \mathbf{v}_{i,0} = \mathbf{x}_{i,1} = \mathbf{0}_p$ and $\widehat{\mathbf{v}}_{i,0} = G_0^2\mathbf{1}_p$, where $G_0 > 0$, for all agents $i \in \mathcal{V}$. Its core update procedures are structured as follows:

$$\begin{cases} \mathbf{m}_{i,t} = \beta_1\mathbf{m}_{i,t-1} + (1-\beta_1)\mathbf{g}_{i,t}, \\ \mathbf{v}_{i,t} = \beta_2\mathbf{v}_{i,t-1} + (1-\beta_2)\mathbf{g}_{i,t} \odot \mathbf{g}_{i,t}, \\ \widehat{\mathbf{v}}_{i,t} = \beta_3\widehat{\mathbf{v}}_{i,t-1} + (1-\beta_3)\max(\widehat{\mathbf{v}}_{i,t-1}, \mathbf{v}_{i,t}), \\ \mathbf{x}_{i,t+1} = \Pi_{\mathcal{X}, \widehat{\mathbf{V}}_{i,t}^{1/2}}(\mathbf{z}_{i,t} - \alpha_t\widehat{\mathbf{V}}_{i,t}^{-1/2}\mathbf{m}_{i,t}), \end{cases}$$

where $\xi_{i,t}$ denotes stochastic noise, with the gradient defined as $\mathbf{g}_{i,t} = \nabla f_i(\mathbf{x}_{i,t}, \xi_{i,t})$, $\beta_1, \beta_2, \beta_3 \in [0, 1)$ are momentum coefficients, $\odot$ denotes element-wise multiplication, $\mathbf{W}$ is the network weight matrix and the aggregated state is given by $\mathbf{z}_{i,t} = \sum_{j\in\mathcal{N}_i}\mathbf{W}_{ij}\mathbf{x}_{j,t}$, $\alpha_t\widehat{\mathbf{V}}_{i,t}^{-1/2}$ is used to dynamically adapt the step size, where $\widehat{\mathbf{V}}_{i,t} = \mathrm{diag}(\widehat{\mathbf{v}}_{i,t})$. Specifically, $\widehat{\mathbf{v}}_{i,t}$ is a d-dimensional vector sequence derived from the raw second-order momentum $\mathbf{v}_{i,t}$ via max comparison with its historical value $\widehat{\mathbf{v}}_{i,t-1}$, and $\widehat{\mathbf{V}}_{i,t}$ is constructed by diagonalising $\widehat{\mathbf{v}}_{i,t}$, which embodies the adaptive momentum feature of the algorithm. The modified second moment $\widehat{\mathbf{v}}_{i,t}$ addresses convergence issues in the vanilla ADAM algorithm [21] by ensuring positive adaptive step components. However, the DADAM algorithm uses a doubly stochastic network weight matrix for inter-agent communication of local decision variables, which confines it to undirected networks. Meanwhile, the DADAM algorithm does not adopt quantisation techniques to cut communication overhead, preventing its use in bandwidth-constrained scenarios.

3.2 | The PS-DADAM Algorithm

The push-sum distributed adaptive momentum (PS-DADAM) algorithm is proposed to address the limitation of the DADAM algorithm being confined to undirected networks, integrating the push-sum consensus framework with the adaptive momentum mechanism to enable efficient constrained optimisation over directed networks. It is initialised with $\mathbf{m}_{i,0} = \mathbf{v}_{i,0} = \mathbf{x}_{i,1} = \mathbf{0}_p$, $\widehat{\mathbf{v}}_{i,0} = G_0^2\mathbf{1}_p$ and $\mathbf{y}_{i,0} = \mathbf{1}_p$ for all agents $i \in \mathcal{V}$, with $\mathbf{y}_{i,t}$ denoting the auxiliary vector for push-sum normalisation. Its core update procedures are structured as follows:

$$\begin{cases} \mathbf{m}_{i,t} = \beta_1\mathbf{m}_{i,t-1} + (1-\beta_1)\mathbf{g}_{i,t}, \\ \mathbf{v}_{i,t} = \beta_2\mathbf{v}_{i,t-1} + (1-\beta_2)\mathbf{g}_{i,t} \odot \mathbf{g}_{i,t}, \\ \widehat{\mathbf{v}}_{i,t} = \beta_3\widehat{\mathbf{v}}_{i,t-1} + (1-\beta_3)\max(\widehat{\mathbf{v}}_{i,t-1}, \mathbf{v}_{i,t}), \\ \mathbf{x}_{i,t+1} = \Pi_{\mathcal{X}, \widehat{\mathbf{V}}_{i,t}^{1/2}}(\mathbf{z}_{i,t} - \alpha_t\widehat{\mathbf{V}}_{i,t}^{-1/2}\mathbf{m}_{i,t}), \end{cases}$$

where $\xi_{i,t}$ denotes stochastic noise, the gradient is defined as $\mathbf{g}_{i,t} = \nabla f_i(\mathbf{z}_{i,t}, \xi_{i,t})$ and $\beta_1, \beta_2, \beta_3 \in [0, 1)$ are momentum coefficients. Here, $\odot$ represents the element-wise multiplication operation, $\mathbf{A}$ is a nondoubly stochastic matrix tailored for directed networks, and $\mathcal{N}_i^{\text{in}}$ denotes the set of in-neighbours of agent i. The term $\alpha_t\widehat{\mathbf{V}}_{i,t}^{-1/2}$ enables dynamic adaptation of the update step size, where $\widehat{\mathbf{V}}_{i,t}$ corresponds to the diagonal matrix constructed from modified second-order moments. Its specific mathematical form is given by $\widehat{\mathbf{V}}_{i,t} = \mathrm{diag}(\widehat{\mathbf{v}}_{i,t})$. Specifically, $\widehat{\mathbf{v}}_{i,t}$ is a d-dimensional vector sequence derived from the raw second-order momentum $\mathbf{v}_{i,t}$. This derivation process involves a max-value comparison between $\mathbf{v}_{i,t}$ and its historical counterpart $\widehat{\mathbf{v}}_{i,t-1}$. The design of $\widehat{\mathbf{v}}_{i,t}$ and the subsequent construction of $\widehat{\mathbf{V}}_{i,t}$ serve as a core manifestation of the adaptive momentum characteristic of the algorithm. $\mathbf{w}_{i,t}$ denotes the aggregated state of the local states of in-neighbours, $\mathbf{y}_{i,t}$ is the auxiliary vector for push-sum normalisation to eliminate nondoubly stochastic weight bias and $\mathbf{z}_{i,t}$ is the consensus state obtained by normalising the aggregated state. The modified second moment $\widehat{\mathbf{v}}_{i,t}$ inherits the advantage of the DADAM algorithm in addressing convergence issues in the vanilla ADAM algorithm [21] by ensuring positive adaptive step components. Compared with the DADAM algorithm, the PS-DADAM algorithm introduces the push-sum consensus mechanism via $\mathbf{w}_{i,t}, \mathbf{y}_{i,t}, \mathbf{z}_{i,t}$, breaking the confinement to undirected networks and enabling adaptation to asymmetric directed topologies. Meanwhile, it retains direct transmission of original states to ensure information integrity, suitable for scenarios with sufficient bandwidth and high precision requirements.

3.3 | Random Quantiser

The random quantiser plays a crucial role in the limited communication bandwidth scenarios.

Definition 1. [9] For a vector $\mathbf{x} \in \mathbb{R}^d$, we define the random quantiser $\mathbf{Q}_t(\mathbf{x}) \in \mathbb{R}^d$ as the time-varying quantiser of $\mathbf{x}$. At time $t \in [T]$, it satisfies the following conditions:

$$\mathbb{E}[\mathbf{Q}_t(\mathbf{x})] = \mathbf{x}, \quad \mathbb{E}\left[\|\mathbf{Q}_t(\mathbf{x}) - \mathbf{x}\|^2\right] \le \epsilon_{d,k_t}\|\mathbf{x}\|^2, \tag{2}$$

where ϵ_{d,k_t} is the quantisation resolution, which depends on both the quantisation level k_t and the dimension d.

Various random quantisers exist, such as the stochastic k-level quantiser [9] and the probabilistic time-varying quantiser [8]. We select the latter here due to its reduced computational burden and straightforward implementation. At time t, given an

n-dimensional vector $\mathbf{e} = [e_1, e_2, ..., e_n]^{\mathrm{T}}$, the quantisation result of the probabilistic time-varying quantiser is denoted as $\mathbf{Q}_t(\mathbf{e}) = [\mathbf{Q}_t(e_1), \mathbf{Q}_t(e_2), ..., \mathbf{Q}_t(e_n)]^{\mathrm{T}}$ with each $\mathbf{Q}_t(e_i)$ for $i \in [1, n]$ defined as follows:

$$\mathbf{Q}_t(e_i) = \begin{cases} \lfloor e_i \rfloor_t, & \text{with probability } \left(\lceil e_i \rceil_t - e_i\right)k_t, \\ \lceil e_i \rceil_t, & \text{with probability } \left(e_i - \lfloor e_i \rfloor_t\right)k_t, \end{cases} \tag{3}$$

where $\lfloor e_i \rfloor_t = \lfloor e_i \rfloor / k_t$ and $\lceil e_i \rceil_t = \lceil e_i \rceil / k_t$.

3.4 | The QPS-DADAM Algorithm

In scenarios of distributed optimisation, communication bandwidth limitations will impede the efficient transmission of information. To overcome this challenge, we propose the QPS-DADAM algorithm, a novel quantised push-sum distributed adaptive momentum algorithm, as detailed in Algorithm 1. The QPS-DADAM algorithm integrates the random quantiser [13, 39] and the adaptive momentum [24] method within the push-sum framework [15, 30] to enable efficient distributed constrained optimisation over directed networks, where the detailed solution flow of the algorithm is presented in Figure 1.

In particular, the QPS-DADAM algorithm employs the push-sum protocol for the collaborative aggregation of local loss functions $f_i(x)$ over directed networks, ensuring effective consensus among agents to support the distributed solution logic of the problem. To tackle insufficient bandwidth in practical communication environments, the algorithm adopts a random quantiser: This not only alleviates bandwidth constraints in the data transmission process of the problem but also enables effects similar to the exact convergence while significantly reducing communication costs, which is particularly valuable for scenarios where the network bandwidth is constrained or communication resources are expensive. For the global constraint $x \in \mathcal{X}$ specified in problem (1), the algorithm integrates popular projection operators to enforce the feasibility of the optimisation variable, ensuring compliance with the constraint set of the problem. Finally, to achieve better convergence speed towards the optimal solution of the problem, the algorithm incorporates an adaptive momentum mechanism, further enhancing its optimisation efficiency for the distributed scenarios targeted by problem (1). The QPS-DADAM algorithm includes the following steps.

Step 1 (Quantised push-sum consensus): At iteration $t \geq 0$, each agent i aggregates the quantised states $\mathbf{Q}_{t-1}(\mathbf{x}_{j,t-1})$ from its in-neighbours $j \in \mathcal{N}_i^{\mathrm{in}}$, normalises the aggregated information using an auxiliary variable and forms a consensus state for subsequent gradient computation as follows:

$$\begin{aligned} \mathbf{w}_{i,t} &= \sum_{j \in \mathcal{N}_i^{\mathrm{in}}} \mathbf{A}_{ij} \mathbf{Q}_{t-1}(\mathbf{x}_{j,t-1}), \quad \mathbf{z}_{i,t} = \mathbf{w}_{i,t} / \mathbf{y}_{i,t}, \\ \mathbf{y}_{i,t} &= \sum_{j \in \mathcal{N}_i^{\mathrm{in}}} \mathbf{A}_{ij} \mathbf{y}_{j,t-1}, \end{aligned} \tag{4}$$

where $\mathbf{w}_{i,t}$ is the weighted aggregation of quantised information received from incoming neighbours after quantisation by $\mathbf{Q}_{t-1}(\cdot)$. The quantisation operation $\mathbf{Q}_{t-1}(\cdot)$ compresses the state $\mathbf{x}_{j,t-1}$ to reduce inter-agent communication costs, with the quantisation error bounded by ϵ_{d,k_t}. The variable $\mathbf{y}_{i,t}$ serves as an auxiliary variable to normalise $\mathbf{w}_{i,t}$, eliminating the bias introduced by the nondoubly stochastic weights $\mathbf{A}_{ij}$ and ensuring the convergence of the push-sum mechanism in directed networks. The variable $\mathbf{z}_{i,t}$ is the consensus state obtained by balancing the information from neighbours and acts as a local proxy for gradient evaluation.

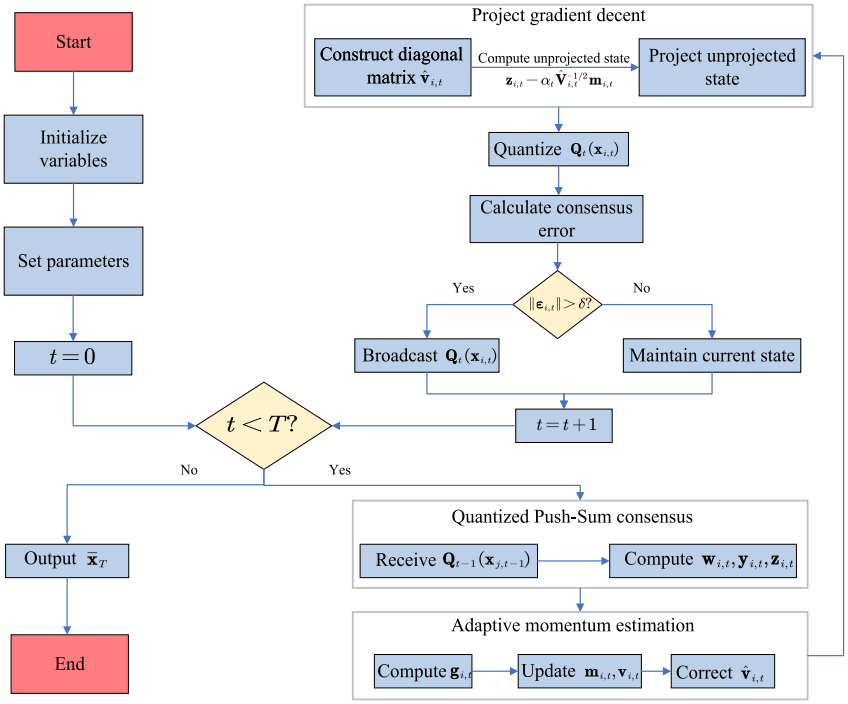


FIGURE 1 | Flow diagram of the QPS-DADAM algorithm.

Step 2 (Adaptive momentum estimation): To mitigate gradient noise and accelerate convergence, agent i updates the first- and second-order momentum by combining historical and current gradient information as follows:

$$\mathbf{m}_{i,t} = \beta_{1,t}\mathbf{m}_{i,t-1} + (1-\beta_{1,t})\mathbf{g}_{i,t},$$
$$\mathbf{v}_{i,t} = \beta_2\mathbf{v}_{i,t-1} + (1-\beta_2)\mathbf{g}_{i,t} \odot \mathbf{g}_{i,t},$$

where $\mathbf{g}_{i,t}$ denotes the stochastic gradient of agent i. Its calculation process is based on the consensus state $\mathbf{z}_{i,t}$ to align gradient benchmarks across agents in directed networks while leveraging random local samples $\xi_{i,t}$ to reflect the characteristics of the agent's local dataset, $\beta_{1,t}$ is a time-varying coefficient that balances the smoothing effect of historical gradients in early iterations and the responsiveness to current gradients in late iterations, β_2 controls the accumulation rate of historical squared gradient information, $\odot$ represents element-wise multiplication, a design that enables $\mathbf{v}_{i,t}$ to capture dimension-wise gradient fluctuations, and the modified second-order momentum $\widehat{\mathbf{v}}_{i,t}$ ensures the stability of adaptive step sizes.

To clarify the specific computational form of $\mathbf{g}_{i,t}$ and explicitly link it to the local loss function, consensus state and local samples, thereby guaranteeing the rationality of the gradient input for momentum updates, its definition is given as follows:

$$\mathbf{g}_{i,t} = \nabla f_{i,t}(\mathbf{z}_{i,t}, \xi_{i,t}).$$

Step 3 (Projection gradient descent): Then, the projection gradient descent step is further executed by projecting the gradient descent part onto the feasible set $\mathcal{X}$ (e.g., $\mathcal{X} = \mathbb{B}_R^d$ for norm-bounded states). Specifically,

$$\mathbf{x}_{i,t} = \Pi_{\mathcal{X},\widehat{\mathbf{V}}_{i,t}^{\frac{1}{2}}}\left[\mathbf{z}_{i,t} - \alpha_t\widehat{\mathbf{V}}_{i,t}^{-\frac{1}{2}}\mathbf{m}_{i,t}\right],$$

where $\widehat{\mathbf{V}}_{i,t}^{-\frac{1}{2}}$ is a diagonal matrix constructed from $\widehat{\mathbf{V}}_{i,t} = \mathrm{diag}(\widehat{\mathbf{v}}_{i,t})$ which adaptively adjusts dimension-wise step sizes, $\Pi_{\mathcal{X},\cdot}$ is used to find the point closest to the unprojected update result within $\mathcal{X}$ and $\mathcal{X}$ is task-specific; this projection operation guarantees $\mathbf{x}_{i,t} \in \mathcal{X}$, which not only maintains the feasibility of constrained optimisation but also avoids quantisation distortion in subsequent communication steps. The constraint handling mechanism of this algorithm includes the projection gradient descent step and adaptation to the feasible set $\mathcal{X}$. It draws on the classical framework from refs. [7, 40], with its rationality supported by theoretical analyses and verification results from both studies.

ALGORITHM 1 | The QPS-DADAM Algorithm.

Input: Momentum parameters $\{\beta_{1,t}\}_{t=1}^T, \beta_2, \beta_3 \in [0,1)$, learning rates $\{\alpha_t\}_{t=1}^T \subset \mathbb{R}_+$, $G_0 > 0$.
Output: $\overline{\mathbf{x}}_T = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_{i,T}$.
Initialise: For all $i \in \mathcal{V}$, set $\mathbf{m}_{i,0} = \mathbf{v}_{i,0} = \mathbf{x}_{i,0} = \mathbf{0}_p$, $\widehat{\mathbf{v}}_{i,0} = G_0^2\mathbf{1}_p$, $\mathbf{y}_{i,0} = \mathbf{1}_p$.
for $t = 1, 2, \ldots, T$ **do**
for $i = 1, 2, \ldots, n$ **do**
$\mathbf{w}_{i,t} = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}}\mathbf{A}_{ij}\mathbf{Q}_{t-1}(\mathbf{x}_{j,t-1})$,
$\mathbf{y}_{i,t} = \sum_{j\in\mathcal{N}_i^{\mathrm{in}}}\mathbf{A}_{ij}\mathbf{y}_{j,t-1}$,
$\mathbf{z}_{i,t} = \mathbf{w}_{i,t}/\mathbf{y}_{i,t}$,
$\mathbf{g}_{i,t} = \nabla f_{i,t}(\mathbf{z}_{i,t}, \xi_{i,t})$,
$\mathbf{m}_{i,t} = \beta_{1,t}\mathbf{m}_{i,t-1} + (1-\beta_{1,t})\mathbf{g}_{i,t}$,
$\mathbf{v}_{i,t} = \beta_2\mathbf{v}_{i,t-1} + (1-\beta_2)\mathbf{g}_{i,t} \odot \mathbf{g}_{i,t}$,
$\widehat{\mathbf{v}}_{i,t} = \beta_3\widehat{\mathbf{v}}_{i,t-1} + (1-\beta_3)\max(\widehat{\mathbf{v}}_{i,t-1}, \mathbf{v}_{i,t})$,
$\widehat{\mathbf{V}}_{i,t} = \mathrm{diag}(\widehat{\mathbf{v}}_{i,t})$,
$\mathbf{x}_{i,t} = \Pi_{\mathcal{X},\widehat{\mathbf{V}}_{i,t}^{1/2}}\left[\mathbf{w}_{i,t} - \alpha_t\widehat{\mathbf{V}}_{i,t}^{-1/2}\mathbf{m}_{i,t}\right]$.
Quantise $\mathbf{x}_{i,t}$ to get $\mathbf{Q}_t(\mathbf{x}_{i,t})$.
end for
end for

Remark 3.1. The core advantage of the QPS-DADAM algorithm lies in the integrates quantised communication, the push-sum protocol and adaptive momentum method, achieving a synergistic enhancement of communication efficiency and optimisation performance in directed network environments. The random quantisation technique effectively reduces inter-agent communication overhead, making it suitable for bandwidth-constrained distributed scenarios. The push-sum protocol addresses information asymmetry in directed networks, ensuring the effectiveness of consensus computation. The adaptive momentum mechanism accelerates the convergence process and reduces sensitivity to learning rates. Theoretical analysis demonstrates that the algorithm can converge to the global optimal solution under convexity conditions, and its engineering practicability has been validated in tasks such as distributed classification, image restoration and sparse signal processing.

3.5 | Comparison With QPS-DO Algorithm

The QPS-DO algorithm, proposed in ref. [15], is a foundational distributed optimisation method designed for directed networks, integrating push-sum consensus and random quantisation to alleviate bandwidth constraints while preserving convergence consistency with exact-communication counterparts. For all agents $i \in \mathcal{V}$, the QPS-DO algorithm initialises with an arbitrary initial state $\mathbf{x}_{i,0} \in \mathbb{R}^p$, an auxiliary normalisation variable $\mathbf{y}_{i,0} = \mathbf{1}_p$, a neighbour state estimation variable $\widehat{\mathbf{x}}_{i,0} = \mathbf{0}_p$ and an aggregated state variable $\mathbf{w}_{i,0} = \mathbf{0}_p$. Its core update framework is defined as follows:

$$\begin{cases} \widehat{\mathbf{x}}_{i,t} = \widehat{\mathbf{x}}_{i,t-1} + \mathbf{Q}_t(\mathbf{x}_{i,t-1} - \widehat{\mathbf{x}}_{i,t-1}), \\ \mathbf{z}_{i,t} = \dfrac{\sum_{j\in\mathcal{N}_i^{\mathrm{in}}}\mathbf{A}_{ij}\widehat{\mathbf{x}}_{j,t}}{\sum_{j\in\mathcal{N}_i^{\mathrm{in}}}\mathbf{A}_{ij}\mathbf{y}_{j,t-1}}, \\ \mathbf{x}_{i,t} = \mathbf{w}_{i,t} - \alpha\nabla f_i(\mathbf{z}_{i,t}, \xi_{i,t}), \end{cases}$$

where $\mathbf{A}_{ij}$ denotes a column-stochastic weight matrix with nonnegative entries and positive diagonal elements to enable consensus over directed networks, $\mathbf{Q}_t(\cdot)$ is an unbiased random quantiser that satisfies $\mathbb{E}[\mathbf{Q}_t(\mathbf{x})] = \mathbf{x}$ and bounded quantisation error $\mathbb{E}\left[\|\mathbf{Q}_t(\mathbf{x}) - \mathbf{x}\|^2\right] \le \omega^2\|\mathbf{x}\|^2$ $(\omega < 1)$, $\mathbf{z}_{i,t}$ is the normalised consensus state, with $\mathbf{y}_{i,t}$ eliminating bias from nondoubly stochastic weights and α is a fixed step size for gradient updates. Although the QPS-DO algorithm establishes a critical framework for quantised distributed optimisation over directed

networks, its limitations motivate the development of the QPS-DADAM algorithm. First, the QPS-DO algorithm lacks adaptive momentum mechanisms: its fixed step size α leads to slow convergence in non-convex scenarios and high sensitivity to step-size selection. This stands in contrast to the QPS-DADAM algorithm, which adopts a time-varying first-order momentum coefficient $\beta_{1,t}$ and a modified second-order momentum $\widehat{\mathbf{v}}_{i,t}$ to dynamically smooth gradient noise and adjust step sizes. Second, the QPS-DO algorithm uses a static quantisation level for $\mathbf{Q}_t(\cdot)$, creating a rigid trade-off between communication efficiency and optimisation accuracy: Higher quantisation levels reduce bandwidth but slow convergence, whereas lower quantisation introduces accumulated errors. The QPS-DADAM algorithm addresses this via a dynamic quantisation level k_t that balances early stage bandwidth savings and late-stage convergence precision. Third, the QPS-DO algorithm does not include a projection operation, restricting it to unconstrained optimisation problems as noted in ref. [15] in its problem formulation. In contrast, the QPS-DADAM algorithm incorporates the projection $\Pi_{\mathcal{X},\widehat{\mathbf{V}}_{i,t}^{-1/2}}[\cdot]$ to ensure feasibility in constrained scenarios, aligning with practical application needs.

Remark 3.2. As demonstrated in ref. [15], the QPS-DO algorithm matches the convergence rates of exact-communication methods for convex problems ($O(1/\sqrt{nT})$) over directed networks and achieves stable convergence with a rate of $O(1/\sqrt{T})$ for non-convex problems. The QPS-DADAM algorithm retains the core logic of the QPS-DO algorithm, which includes push-sum and quantisation, to leverage directed network compatibility and low communication overhead, while addressing the gaps of the QPS-DO algorithm in momentum adaptation, dynamic quantisation and constraint handling to enhance performance in complex real-world scenarios.

4 | Convergence Proof

In this section, we analyse the convergence of the QPS-DADAM algorithm and derive the upper bound of regret in convex scenarios.

4.1 | Auxiliary Lemmas

The subsequent theoretical analysis is based on the following auxiliary lemmas. Using these lemmas, we can establish supporting lemmas and derive the main results.

Lemma 1. *[41] If network $\mathcal{G}$ is time-invariant and directed strongly connected, then there exists a stochastic vector $\boldsymbol{\phi}$ (with components $\phi_i > 0$ and $\sum_{i=1}^{n}\phi_i = 1$), a constant $\mu > 0$, and a parameter $\nu \in (0,1)$ such that*

$$\left\|\mathbf{A}^t - \boldsymbol{\phi}\mathbf{1}^{\mathrm{T}}\right\| \le \mu\nu^t, \quad \sum_{j=1}^{n}\left|\left[\mathbf{A}^t\right]_{ij} - \frac{1}{n}\right| \le \mu\nu^t, \tag{5}$$

where $\mathbf{A}$ is the column-stochastic weight matrix corresponding to network $\mathcal{G}$.

Lemma 2. *[42] Under Assumption 1, it holds for all $i \in \mathcal{V}$ and $t \in [T]$, and the following inequalities hold:*

$$\left\|\widehat{\mathbf{v}}_{i,t}\right\|_\infty \le G_\infty^2, \quad \left\|\mathbf{m}_{i,t}\right\|_\infty \le G_\infty. \tag{6}$$

Lemma 3. *[24] Under the conditions that $\alpha_t = \alpha/\sqrt{t}$, $\{\beta_i\}_{i=1}^3 \in [0,1)$, $0 \le \beta_{1,t} \le \beta_1$ and $\eta = \beta_1/\beta_2 < 1$, then for each $i \in \mathcal{V}$ and $t \in [T]$, the following inequality holds:*

$$\sum_{t=1}^{T}\alpha_t\left\|\widehat{\mathbf{V}}_{i,t}^{-\frac{1}{2}}\mathbf{m}_{i,t}\right\| \le \frac{\alpha\sqrt{1+\log T}\sum_{d=1}^{p}\left\|\left|\mathbf{g}_{i,1:T,d}\right|^{\frac{1}{2}}\right\|}{\sqrt{G_0}\theta}, \tag{7}$$

where $\theta = (1-\eta)(1-\beta_1)\sqrt{(1-\beta_2)(1-\beta_3)}$ and $\mathbf{g}_{i,1:T,d} = \left[\mathbf{g}_{i,1,d}, \ldots, \mathbf{g}_{i,T,d}\right]^{\mathrm{T}}$.

4.2 | Supporting Lemmas

The following supporting lemma lays the foundation for the proof of the main results.

Lemma 4. *For all $i \in \mathcal{V}$ and $t \in [T]$, we have*

$$\begin{aligned}\left\langle\widehat{\mathbf{V}}_{i,t}^{-\frac{1}{2}}\mathbf{m}_{i,t}, \mathbf{Q}_{t-1}(\mathbf{x}_{i,t-1}) - \mathbf{x}\right\rangle \le \frac{1}{2\alpha_t}\left(\|\mathbf{x}-\mathbf{w}_{i,t}\|^2\right.\\ \left. - \|\mathbf{Q}_{t-1}(\mathbf{x}_{i,t-1}) - \mathbf{x}\|^2 - \|\mathbf{Q}_{t-1}(\mathbf{x}_{i,t-1}) - \mathbf{w}_{i,t}\|^2\right),\end{aligned} \tag{8}$$

where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{w}_{i,t} = \sum_{j=1}^{n}[\mathbf{W}]_{ij}\mathbf{Q}_{t-1}(\mathbf{x}_{j,t-1})$.

Proof. See Supporting Information S1. □

Lemma 5. *Under Assumptions 1–3, for each $i \in \mathcal{V}$ and $t \in [T]$, we have*

$$\left\|\mathbf{Q}_t(\mathbf{x}_{i,t}) - \overline{\mathbf{Q}}_t(\mathbf{x}_t)\right\| \le \mu\sum_{k=1}^{t}\nu^{t-k}\alpha_k M_{j,k}, \tag{9}$$

where $M_{j,k} = \max_{j\in[n]}\left\|\widehat{\mathbf{V}}_{j,k}^{-1/2}\mathbf{m}_{j,k}\right\|$.

Proof. See Supporting Information S1. □

Lemma 6. *Under Assumptions 1–3, for each $i \in \mathcal{V}$ and $t \in [T]$, we have*

$$\begin{aligned}\left\|\frac{n\overline{\mathbf{x}}_{t-1}}{P_y} - \mathbf{z}_{i,t}\right\|_2 \le \frac{3\mu R\sqrt{\varepsilon_{d,k_t}}}{(1-\nu)\delta} + \frac{4\mu R}{(1-\nu)\delta}\\ + \frac{2\mu^2(1+\mu\nu)}{(1-\nu)\delta}\sum_{k=1}^{t}\nu^{t-k}\alpha_k M_{j,k},\end{aligned} \tag{10}$$

where $P_y = \mathbf{1}^{\mathrm{T}}\mathbf{y}_0$.

Proof. See Supporting Information S1. □

Lemma 7. *[43, 44] Under Assumptions 1–3, for each $i \in \mathcal{V}$ and $t \in [T]$, we have*

$$\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{n}\left\|\mathbf{z}_{i,t}-\mathbf{z}^{*}\right\| \leq \frac{n^2\overline{h}_i}{2}+\frac{nR\mu\nu}{1-\nu}+\frac{4\sqrt{T}n\mu L\nu}{1-\nu}-\frac{2n\mu L\nu^2}{(1-\nu)^2}+2\sqrt{T}Ln + \frac{1+R\sqrt{\epsilon_{d,k_t}}}{\overline{h}_i}\left(\frac{3nT\mu R\sqrt{\epsilon_{d,k_t}}}{(1-\nu)\delta}+\frac{4nT\mu R}{(1-\nu)\delta}+\frac{2\mu^2(1+\mu\nu)\max_{j\in[n]}\overline{h}_j}{(1-\nu)^2\delta}\right), \tag{11}$$

where $\overline{h}_i = \frac{\alpha\sqrt{1+\log T}\sum_{d=1}^{p}\left\|\left|\mathbf{g}_{i,1:T,d}\right|^{\frac{1}{2}}\right\|}{\sqrt{G_0\theta}}$.

Proof. See Supporting Information S1. □

4.3 | Main Results

Theorem 1. *Under Assumptions 1–3, let* $\beta_{1,t}=\beta_1\lambda^{t-1}$ *with* $\lambda\in(0,1)$, $\{\beta_i\}_{i=1}^{3}\in[0,1)$, $\eta=\beta_1/\beta_2<1$, $y_0=\mathbf{1}$ *and* $\alpha_t=1/\sqrt{t}$. *Then, the following inequality holds:*

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{z}_{i,t}\right)-f(\mathbf{z}^{*})\right]\leqslant B\left(\frac{C+D}{T}+\frac{E}{\sqrt{T}}+F\right), \tag{12}$$

where B, C, D, E and F are defined as follows:

$$B=\sqrt{p}G_{\infty},$$

$$C=\frac{n^2\max_{j\in[n]}\overline{h}_j}{2}+\frac{nR\mu\nu}{1-\nu}-\frac{2n\mu L\nu^2}{(1-\nu)^2},$$

$$D=\frac{2\mu^2\left(1+R\sqrt{\epsilon_{d,k_t}}\right)(1+\mu\nu)}{(1-\nu)^2\delta}\cdot\frac{\max_{j\in[n]}\overline{h}_j}{\min_{j\in[n]}\overline{h}_j},$$

$$E=\frac{4n\mu L\nu}{1-\nu}+2nL,$$

$$F=\frac{\left(1+R\sqrt{\epsilon_{d,k_t}}\right)\left(3n\mu R\sqrt{\epsilon_{d,k_t}}+4n\mu R\right)}{(1-\nu)\delta\min_{j\in[n]}\overline{h}_j}.$$

Proof. Proof: See Supporting Information S1. □

Remark 4.1. The theorem establishes a sublinear regret bound $\tilde{O}(1/\sqrt{T})$, confirming the near-optimal convergence of the QPS-DADAM algorithm in convex scenarios. The bound quantifies the trade-off between communication efficiency and optimisation accuracy, with stronger network connectivity accelerating convergence exponentially. The adaptive momentum method mitigates learning rate sensitivity, outperforming fixed-step algorithms in nonstationary environments.

Remark 4.2. The regret bound in Theorem 1 integrates the quantisation efficiency term $R\sqrt{\epsilon_{d,k_t}}$, which balances communication bandwidth and optimisation accuracy by controlling the trade-off between quantisation precision and data transmission volume; the topological influences parameter $\mu/(1-\nu)$, reflecting the exponential mixing property of strongly connected networks as established in Lemma 1 and the momentum normalisation mechanism via $\widehat{\mathbf{V}}_{i,t}^{-1/2}\mathbf{m}_{i,t}$, which stabilises gradient updates under quantised communication by adapting to historical gradient statistics. This theoretical decomposition provides a rigorous basis for optimising quantisation precision and network topology in bandwidth-constrained distributed systems, ensuring the algorithm maintains convergence stability while reducing communication overhead.

5 | Simulation Experiments Results

To evaluate the performance of the QPS-DADAM algorithm, we conduct extensive experiments on various benchmark problem settings using Python libraries such as NumPy and PyTorch. The experiments are executed on a computer with an Intel i5-10400 processor, 32 GB of RAM and an NVIDIA RTX 2060 GPU. The code for all numerical experiments is available at https://github.com/hello-ikun-zqk/qps_agl.git.

For the distributed classification task, we compare the proposed QPS-DADAM algorithm with existing counterparts using representative datasets. Parameter configurations are strictly aligned with the convergence analysis in Section 4. Experimental results show that the QPS-DADAM algorithm achieves convergent performance comparable to that of existing algorithms under theoretical parameter constraints, which verifies the validity of our theoretical analysis. This analysis covers the algorithm's adaptability to directed networks and its ability to handle classification task constraints. Concurrently, we also evaluate the QPS-DADAM algorithm under diverse quantisation strategies to validate the quantiser, and results show that the quantiser effectively reduces communication costs while maintaining stable classification performance. In directed network environments, the distributed image restoration task enables agents to collaboratively recover clear images from degraded observations affected by blur and noise. The distributed sparse signal compressive sensing task allows agents to achieve joint sparse signal reconstruction from measurements, with reconstruction errors confined to a small range that meets practical precision requirements. Both tasks successfully meet practical application requirements and verify the practicality of the QPS-DADAM algorithm in real-world distributed scenarios.

5.1 | Distributed Classification Task

To assess the convergence and communication efficiency of the QPS-DADAM algorithm, we use four datasets: Fashion-MNIST, MNIST, Covertype and Breast. The Breast dataset is used for binary classification (classifying tumours as malignant or benign), with each sample containing 30 features. We randomly select 445 instances for training and 114 for testing. The Covertype dataset is a multi-class dataset with 54 features per sample and is used to classify instances into seven vegetation classes. We randomly select 464,809 training samples and 116,203 test samples. The MNIST dataset comprises 28 × 28-pixel greyscale handwritten digit images, with 60,000 training and 10,000 test samples. Fashion-MNIST is a 10-class dataset with 60,000 training and 10,000 test samples and consists of 28 × 28-pixel greyscale fashion images. Compared with the MNIST dataset, Fashion-MNIST exhibits higher diversity and complexity, making its classification task more challenging.

In this test, all agents collaboratively train a classifier by jointly optimising the aggregated classification loss. This loss integrates error measurement and overfitting mitigation, which helps balance classification accuracy and model generalisation. For binary classification tasks, we formulate the distributed optimisation problem with two core components. One is the binary cross-entropy loss, which measures classification errors errors, and the other is a L_2 regularisation term, which controls the complexity of model parameters. The problem is expressed as follows:

$$\begin{aligned}\min_{\mathbf{w}_i\in\mathbb{R}^d} f(\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_n) = \frac{1}{n}\sum_{i=1}^{n}\Bigg[\frac{1}{m_i}\sum_{j=1}^{m_i}\Big(y_{i,j}\log\widehat{y}_{i,j} \\ +\Big(1-y_{i,j}\Big)\log\Big(1-\widehat{y}_{i,j}\Big)\Big)+\frac{\rho}{2}\|\mathbf{w}_i\|^2\Bigg], \\ \text{s.t.}\mathbf{w}_i\in\mathcal{X},\mathbf{w}_i=\mathbf{w}_j,\end{aligned} \tag{13}$$

where $\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_n$ denote the local model parameters of all agents. Each $\mathbf{w}_i\in\mathbb{R}^d$, with its dimension matching that of the feature space in the optimisation problem. The constraint $\mathbf{w}_i\in\mathcal{X}$ $(i=1,2,...,n)$ restricts each agent's local parameters to a feasible range with physical or mathematical significance. This restriction prevents extreme parameter values that may lead to update oscillations or divergence, thereby safeguarding the stability of parameter updates throughout the distributed optimisation process. In the network, n represents the total number of agents. For each agent i, m_i denotes the number of local training samples it holds. $y_{i,j}\in\{0,1\}$ is the true binary label of the j-th sample belonging to agent i. $\widehat{y}_{i,j}\in(0,1)$ denotes the predicted probability where the j-th sample of agent i belongs to the positive class, and it is calculated using the sigmoid function:

$$\widehat{y}_{i,j}=\sigma\big(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{i,j}\big)=\frac{1}{1+e^{-\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{i,j}}}. \tag{14}$$

For tasks with $C>2$ classes, we employ the categorical cross-entropy loss, defined as

$$f(\mathbf{w})=-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\sum_{c=1}^{C}y_{i,j}^{c}\log\widehat{y}_{i,j}^{c}. \tag{15}$$

We encode labels using one-hot vectors $y_{i,j}^c,\widehat{y}_{i,j}^c\in\{0,1,...,C-1\}$. The softmax function is used to normalise the predicted labels, defined as

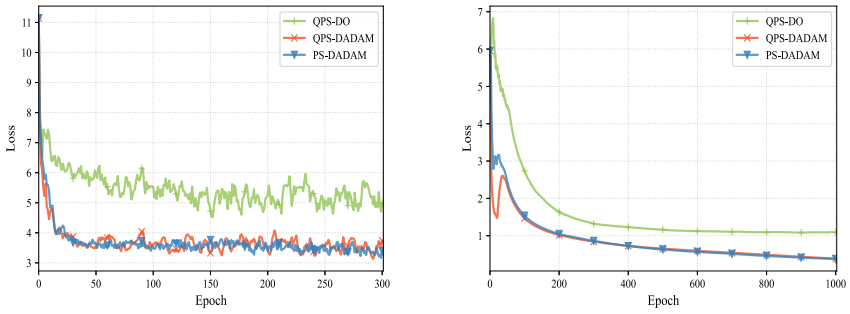
$$\widehat{y}_{i,j}^{c}=\frac{e^{\mathbf{w}_i^{c\mathrm{T}}\mathbf{x}_{i,j}}}{\sum_{k=1}^{C}e^{\mathbf{w}_i^{k\mathrm{T}}\mathbf{x}_{i,j}}}. \tag{16}$$

Consider a directed and strongly connected network with $n=10$ agents, and the training data are evenly distributed among the agents. Three algorithms are compared in this setup, namely the QPS-DADAM algorithm, the PS-DADAM algorithm and the QPS-DO algorithm. For the QPS-DADAM algorithm, the parameters are set as follows. The learning rate $\alpha_t=\alpha/\sqrt{t}$ with $\alpha=1$, exponential decay rates $\beta_1=0.9^t$, $\beta_2=0.999$, $\beta_3=0.9$, initial gradient scale $G_0=0.001$ and quantisation level $k_t=\lceil t^{1.5}\rceil$. PS-DADAM and QPS-DO adopt the same parameter settings as QPS-DADAM.

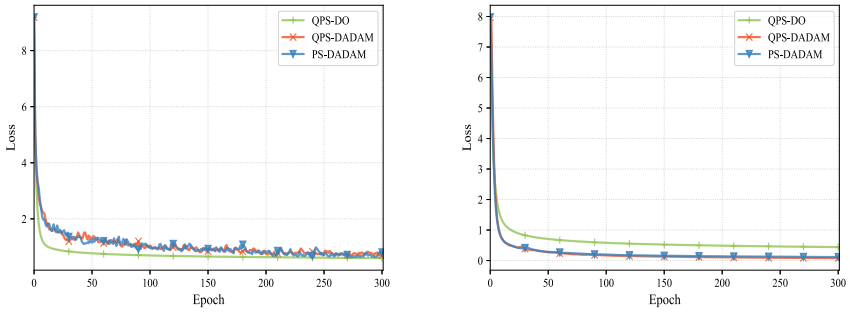
To clarify the attributes of the parameters used in the QPS-DADAM algorithm, we categorise them into three types based on their origin and adjustability: The first type is algorithm-innate parameters, which are fixed by the theoretical framework of the QPS-DADAM algorithm or the requirements of classification tasks, including the learning rate decay form $\alpha_t=\alpha/\sqrt{t}$, the time-varying form of $\beta_{1,t}$, the quantisation level form $k_t=\lceil t^{\gamma}\rceil$, cross-entropy loss functions and sigmoid/softmax activation functions; the second type is user-selectable parameters, which need to be set or adjusted by users to optimise the performance of the algorithm, covering the number of agents $n=10$, base learning rate $\alpha=1$, momentum coefficients $\beta_1=0.9$, $\beta_2=0.999$, $\beta_3=0.9$, initial gradient scale $G_0=0.001$, quantisation exponent $\gamma=1.5$ and regularisation coefficient ρ; the third type is random parameters, which are generated through random processes, specifically including the splitting of training/test sets, the distribution of data among agents and the selection of $\lfloor e_i\rfloor_t/\lceil e_i\rceil_t$ in the random quantiser.

Figure 2 demonstrates that the QPS-DADAM algorithm converges faster than the QPS-DO algorithm on the Covertype, Breast and Fashion-MNIST datasets. As shown in Figure 3, under identical test conditions, the QPS-DADAM algorithm achieves higher classification accuracy than the QPS-DO algorithm across all four datasets. Additionally, the confusion matrices in Figure 4 reveal that class predictions for each dataset are predominantly concentrated along the diagonal, indicating high overall classification accuracy. Notably, despite leveraging random quantisation to reduce communication overhead, the performance of the QPS-DADAM algorithm matches that of the PS-DADAM algorithm. This suggests that quantisation errors have a negligible impact on the convergence behaviour of the algorithm.

To evaluate the impact of different quantisation strategies on the communication bandwidth, we compare four configurations in this test: no quantisation, light quantisation, moderate quantisation and heavy quantisation. The comparison takes place in a setup with $n=10$ agents. We use logistic regression tasks on the Fashion-MNIST dataset to evaluate each configuration, with the QPS-DADAM algorithm employing parameter settings consistent with previous tests. In the no-quantisation scenario,

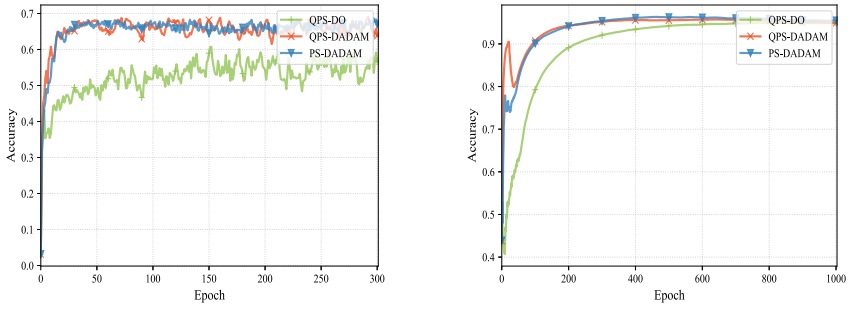


(a) Covertype dataset **(b)** Breast dataset

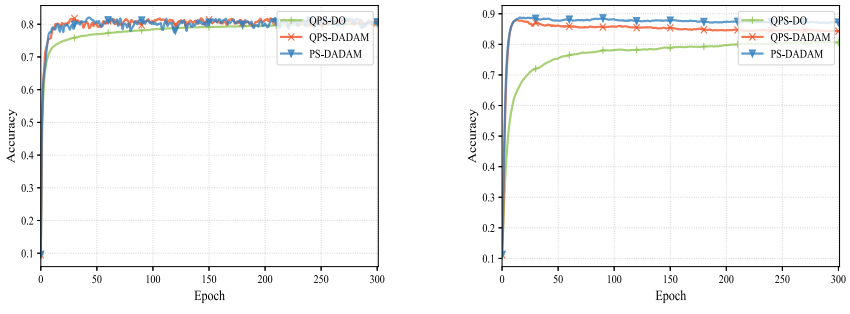


(c) Fashion-MNIST dataset **(d)** MNIST dataset

FIGURE 2 | Loss comparison of algorithms on multiple classification datasets.



(a) Covertype dataset **(b)** Breast dataset



(c) Fashion-MNIST dataset **(d)** MNIST dataset

FIGURE 3 | Accuracy comparison of algorithms on multiple classification datasets.

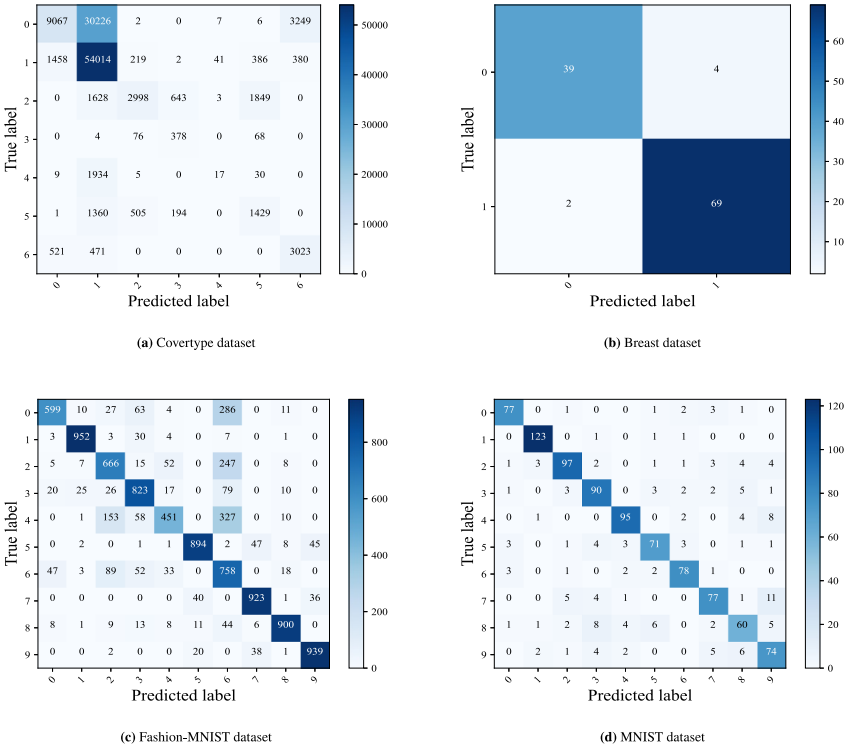
data are represented using 32-bit floating-point numbers, whereas to estimate communication volume under quantisation, we simulate data transmission between all agent pairs (i, j), where $i, j \in \{1, ..., 10\}$. Agent i first converts floating-point data $\mathbf{x} \in \mathbb{R}^d$ into a quantised integer array $\mathbf{q}^{\text{int}} = \lfloor \mathbf{x} \cdot k_t + 0.5 \rfloor$, using a random quantiser. The resulting integer array is then compressed into a byte stream via a dynamic bitwidth packaging mechanism: A 1-byte header parameter b, determined by $\max(\mathbf{q}^{\text{int}})$ recording the storage bitwidth and each element q_k^{int} (for $k \in [d]$), is encoded using exactly b bits to ensure lossless reconstruction of data dimension the and quantisation scale. Agent j uses header parameter b to reconstruct the integer array

and recovers the original values via inverse scaling $\mathbf{x}^{\text{rec}} = \mathbf{q}^{\text{int}}/k_t$. During each iteration, the algorithm measures the communication cost by accumulating the total number of bits in transmitted byte streams.

Figure 5a shows that the communication volume grows nearly linearly with iterations, with random quantisation steepening the slope as k_t increases, reflecting higher precision. Figure 5b illustrates the trade-off between quantisation and bandwidth: after 500 iterations, the total volume is 42,900.48 Mbits without quantisation, reduced to 32.3% (13,838.36 Mbits) under heavy quantisation ($k_t = \lceil t^{0.6} \rceil$), 60.5% under moderate ($k_t = \lceil t^{1.5} \rceil$) and 74.6% under light ($k_t = \lceil t^{3.0} \rceil$), confirming quantisation's bandwidth-saving effect.

To assess the QPS-DADAM algorithm in distributed deep learning, we test it on CIFAR-10 with a ResNet-18 variant. CIFAR-10 has 60,000 colour images ($32 \times 32 \times 3$) across 10 classes, split into 48,000 training and 12,000 test samples. Images are normalised, batch size is 128. For CIFAR-10, the input 7×7 conv (stride 2) is replaced by a 3×3 conv (stride 1), the following max-pooling is removed and the final fully connected layer outputs 10 classes. The model is initialised with ImageNet-pretrained weights and trained with multi-class cross-entropy loss (15). In the QPS-DADAM algorithm, the initial learning rate is $\alpha = 0.01$, with other hyperparameters in logistic regression.

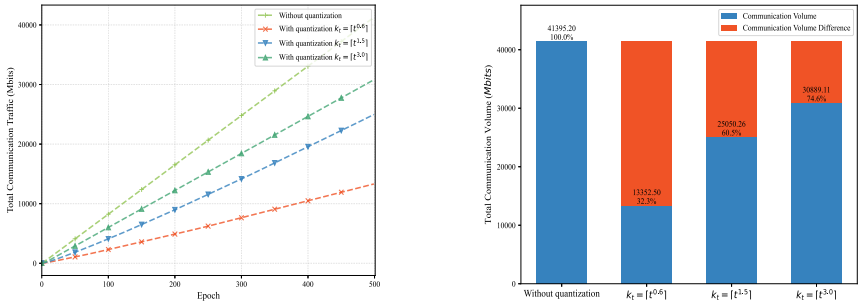
Figure 6 illustrates the optimisation trajectory of the ResNet-18 model using the proposed algorithm. Figure 6a depicts the classification accuracy and loss curves over the training epochs. The final test accuracy on CIFAR-10 stabilises at approximately 90.5%. Figure 6b shows consistent prediction accuracy across all classes, indicating no significant bias towards any class. These results collectively demonstrate that the QPS-DADAM algorithm effectively handles large-scale data and complex training tasks in strongly directed connected networks under limited-bandwidth constraints.



(a) Covertype dataset **(b)** Breast dataset

(c) Fashion-MNIST dataset **(d)** MNIST dataset

FIGURE 4 | Confusion matrices comparison of the QPS-DADAM algorithm on multiple classification datasets.



(a) The impact of various quantitative strategies on communication volume **(b)** Bar chart of total communication volume for each quantitative strategy

FIGURE 5 | The impact of different quantisation strategies on the communication bandwidth in the QPS-DADAM algorithm.

5.2 | Distributed Image Restoration

In this test, we present the test results of an image restoration task to demonstrate the efficacy of the QPS-DADAM algorithm. Specifically, each agent $i \in \mathcal{V}$ obtains a distinctly degraded version of a common image, with varying amounts of blur and noise. The agents aim to recover the original clear image from these degraded observations through collaborative optimisation. For each agent $i \in \mathcal{V}$, the linear degradation model is given as follows [45]:

$$\mathbf{y}_i = \mathbf{H}_i\mathbf{x} + \mathbf{n}_i, \tag{17}$$

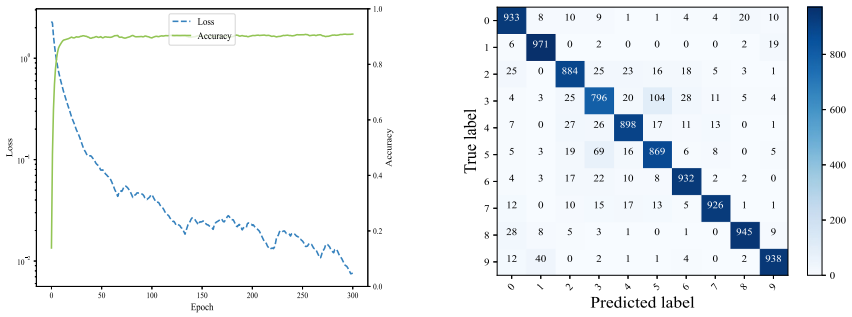
where $\mathbf{y}_i$ denotes the degraded image, $\mathbf{H}_i$ is the degradation operator and $\mathbf{n}_i$ represents the additive noise. When $\mathbf{H}_i = \mathbf{I}$, the degradation consists solely of additive noise $\mathbf{n}_i$ during acquisition and transmission. Another common degradation is motion blur, which is caused by relative motion between the object and the sensor during exposure. In this case, $\mathbf{H}_i$ represents convolution with a blur kernel and $\mathbf{n}_i$ represents additive white noise. This subsection focuses on motion blur. Figure 7 illustrates examples of partially degraded images.

To tackle the image restoration problem, we first convert each image into a vector, which converts traditional filtering operations into multiplication with toeplitz matrices for mathematical modelling. For the distributed collaborative image restoration task, we directly formulate the global constrained optimisation problem as follows:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\left[\|\mathbf{H}_i\mathbf{x} - \mathbf{y}_i\|^2 + \frac{\lambda}{2}\|\mathbf{x}\|^2\right], \mathbf{x} \in \mathcal{X}, \tag{18}$$

where $\mathbf{x}$ denotes the vectorised form of the original clear image, a single global optimisation variable shared by all agents and distinct from the local parameters $\{\mathbf{w}_i\}$ in classification tasks. $\mathcal{X} \subseteq \mathbb{B}_R^m$ is the feasible set that restricts $\mathbf{x}$ to a physically meaningful range. For example, $\mathcal{X} = [0, 255]^{784}$ for 28×28 greyscale Fashion-MNIST images to match 8-bit pixel values. For each agent $i \in \mathcal{V}$, $\mathbf{H}_i$ is its corresponding blur matrix, that is, the degradation operator, and $\mathbf{y}_i$ is the vectorised degraded image it holds. The term $\frac{\lambda}{2n}\|\mathbf{x}\|^2$ uses λ as the regularisation coefficient to enforce the smoothness of the restored image and prevent overfitting. The sum $\sum_{i=1}^{n}(\cdot)$ represents the aggregation of all agents' local losses to enable collaborative restoration.

To evaluate the restoration performance, we employ the peak signal-to-noise ratio (PSNR), defined as



(a) Loss value and accuracy **(b)** Confusion matrix

FIGURE 6 | Performance evaluation of the QPS-DADAM algorithm using ResNet18 on CIFAR-10.

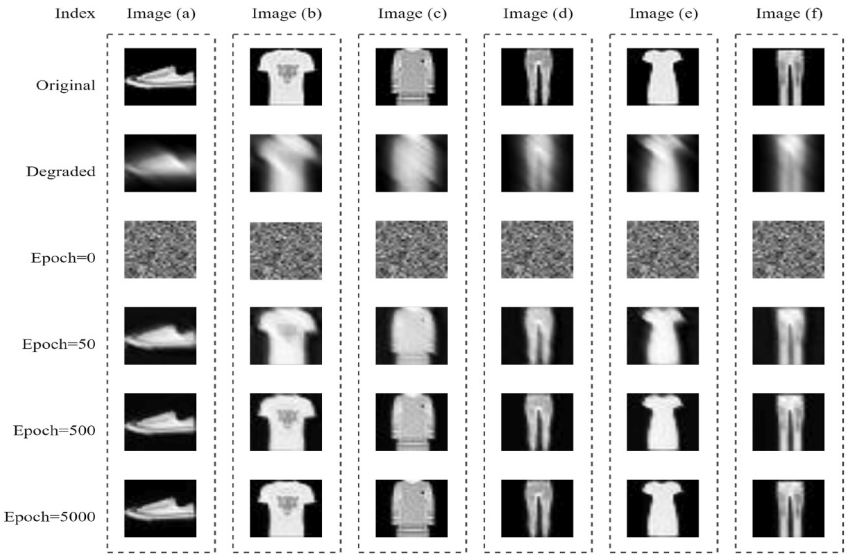


FIGURE 7 | Image restoration performance of QPS-DADAM on Fashion-MNIST.

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right),$$

where MAX denotes the maximum pixel value (e.g., 255 for 8-bit images). The mean squared error (MSE) is given by

$$\mathrm{MSE} = \frac{1}{u}\sum_{j=1}^{u}(\mathbf{x}_j - \hat{\mathbf{x}}_j)^2,$$

where $\mathbf{x}_j$ and $\hat{\mathbf{x}}_j$ denote the j-th pixel values of the original image and the restored image, respectively. We also employ the structural similarity index (SSIM) as a restoration performance metric. For images $\mathbf{x}$ and $\mathbf{y}$, the SSIM metric is defined as follows:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{\left(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1\right)\left(2\sigma_{\mathbf{xy}} + C_2\right)}{\left(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1\right)\left(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{y}}^2 + C_2\right)},$$

where $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{y}}$ denote the mean values of images $\mathbf{x}$ and $\mathbf{y}$, respectively. $\sigma_{\mathbf{x}}^2$ and $\sigma_{\mathbf{y}}^2$ represent the variances of $\mathbf{x}$ and $\mathbf{y}$, and $\sigma_{\mathbf{xy}}$ is the covariance. The constants $C_1 = (0.01L)^2$ and $C_2 = (0.03L)^2$ stabilise the denominator, where L is the dynamic range (e.g., $L = 255$ for 8-bit images). We set up a strongly connected directed network of $n = 20$ agents to perform collaborative image restoration, and the parameter settings are the same as in previous tests.

To clarify the attributes of the parameters used in the QPS-DADAM algorithm for image restoration tasks, we categorise them into three types based on their origin and adjustability. The first type is algorithm-innate parameters. These parameters are fixed by the theoretical framework of the QPS-DADAM algorithm or the requirements of image restoration tasks, including the learning rate decay form $\alpha_t = \alpha/\sqrt{t}$, the time-varying form of $\beta_{1,t}$, the quantisation level form $k_t = \lceil t^{\gamma} \rceil$, the image degradation model $\mathbf{y}_i = \mathbf{H}_i\mathbf{x} + \mathbf{n}_i$ and the calculation formulas for PSNR, MSE and SSIM. The second type is user-selectable parameters. These parameters are set or adjusted by users to optimise performance, covering the number of agents $n = 20$, base learning rate $\alpha = 1$, momentum coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\beta_3 = 0.9$, initial gradient scale $G_0 = 0.001$, quantisation exponent $\gamma = 1.5$, regularisation coefficient λ and feasible set $\mathcal{X}$. The third type is random parameters. These parameters are generated via random processes, including six test images randomly selected from Fashion-MNIST, the random generation of additive degradation noise $\mathbf{n}_i$ and the random distribution of degraded image data among agents. Given the $28\times$ 28 greyscale nature of Fashion-MNIST images, the restored vector $\mathbf{x}$ is constrained to the domain $\mathcal{X} = [0, 255]^{784}$, which serves as the feasible set for the optimisation task.

Figure 8 reports PSNR and SSIM over iterations, with all images reaching PSNR > 25 dB and SSIM > 0.9. Figure 7 illustrates the reconstruction process, verifying the effectiveness of the QPS-DADAM algorithm in distributed image restoration. To test scalability, we apply the QPS-DADAM algorithm to a 256 × 256 greyscale image from the Photographer dataset on $n = 32$ agents with parameters unchanged, restricting the solution to $\mathcal{X} = [0, 255]^{65536}$. Figure 9 shows clarity improving from random noise at epoch 0, with the final result achieving PSNR = 34.55 dB and SSIM = 0.9842. These results confirm the robustness of the QPS-DADAM algorithm across resolutions.

5.3 | Distributed Sparse Signal Compression Sensing

We consider a strongly connected directed network of n agents, where a source emits a sparse vector $x \in \mathbb{R}^q$. Each agent $i \in \mathcal{V}$ collects m independent measurements affected by sensor positions and biases. The j-th measurement uses matrix $O_{i,j} \in \mathbb{R}^{p\times q}$ with $p < q$, corrupted by zero-mean Gaussian noise $e_{i,j}$, yielding vectors $y_{i,j} \in \mathbb{R}^p$. We model each measurement as

$$\mathbf{y}_{i,j} = O_{i,j}x + \mathbf{e}_{i,j}, \tag{19}$$

where each $O_{i,j}$ has independent and identically distributed Gaussian entries with columns normalised to the unit norm. The noise $e_{i,j}$ is i.i.d., zero-mean and Gaussian. To collaboratively recover the sparse signal $\mathbf{w}$, we formulate the L_1-norm optimisation problem as

$$\min_{\mathbf{w}\in\mathbb{R}^q} f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}) + \frac{\rho}{n}\|\mathbf{w}\|_1, \quad \text{s.t. } \mathbf{w} \in \mathcal{X}, \tag{20}$$

where ρ is the regularisation parameter, $\mathbf{w} \in \mathcal{X}$ ($\mathcal{X} \subseteq \mathbb{B}_R^q$) and $f_i(\mathbf{w})$ is defined as

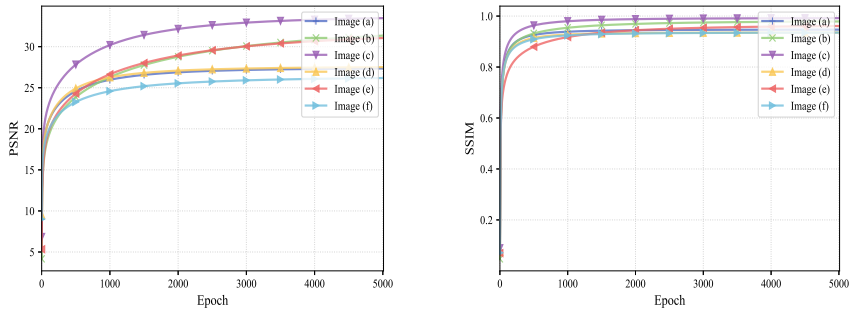
$$f_i(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{2}\left\|O_{i,j}\mathbf{w} - \mathbf{y}_{i,j}\right\|_2^2. \tag{21}$$

We set up a strongly connected directed network with $n = 50$ agents, with each agent $i \in \mathcal{V}$ collecting $m = 10$ observations. The sparse signal dimension is $q = 328$, and each observation is in $\mathbb{R}^{168}$ (i.e., $p = 168$). We enforce a sparsity level of $s = 88$ by randomly choosing 88 nonzero entries of $\mathbf{x}$. Hence, we constrain $\mathbf{x}$ to lie in $[-1, 1]^{328}$. We run the algorithm for 1000 iterations with a regularisation parameter of $\rho = 0.011$. For $k \in [q]$, we generate $\mathbf{x}$ according to a sinusoidal pattern as follows:

$$\mathbf{x}(k) = \sin(2\pi f \cdot k),$$

where the frequency $f = 5$ Hz. We then randomly set 240 entries of $\mathbf{x}$ to zero to enforce sparsity. The parameter settings of the QPS-DADAM algorithm are the same as in previous tests.

To clarify the attributes of the parameters used in the QPS-DADAM algorithm for sparse signal sensing tasks, we categorise them into three types based on their origin and adjustability. The first type is algorithm-innate parameters, which have fixed calculation rules or forms derived from the QPS-DADAM algorithm theory or sparse signal sensing task requirements, including the learning rate decay form $\alpha_t = \alpha/\sqrt{t}$, the time-varying form of $\beta_{1,t}$, the quantisation level form $k_t = \lceil t^{\gamma} \rceil$, the measurement model $\mathbf{y}_{i,j} = \mathbf{O}_{i,j}\mathbf{x} + \mathbf{e}_{i,j}$ and the L_1-norm optimisation problem form. The second type is user-selectable parameters, whose values are set or adjusted by users to optimise



(a) Peak signal-to-noise ratio

(b) Structural similarity index

FIGURE 8 | Performance comparison of the QPS-DADAM algorithm on Fashion-MNIST for image restoration.

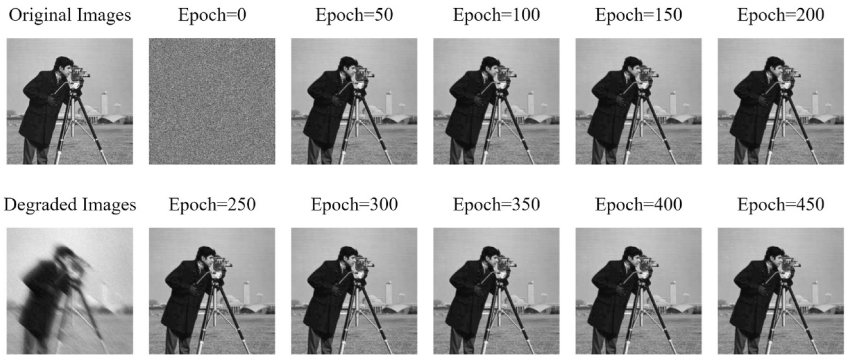


FIGURE 9 | Performance of the QPS-DADAM algorithm on Photographer for image restoration.

performance, covering the number of agents $n = 50$, number of observations per agent $m = 10$, signal dimension $q = 328$, observation dimension $p = 168$, signal sparsity $s = 88$, total iterations, regularisation parameter $\rho = 0.011$, signal frequency $f = 5$ Hz, feasible set $\mathcal{X} = [-1, 1]^{328}$, base learning rate $\alpha = 1$, momentum coefficients $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\beta_3 = 0.9$ and initial gradient scale $G_0 = 0.001$. The third type is random parameters, which are generated via random processes, including the random selection of 88 nonzero entries in $\mathbf{x}$, random zeroing of 240 entries in $\mathbf{x}$, random selection of three agents for signal visualisation, random Gaussian noise $\mathbf{e}_{i,j}$ and random distribution of measurement data among agents.

To visualise the input measurements, we randomly select signals from three agents as examples. Figure 10 shows two measurements from each of the agents 15, 37 and 59. Figure 11a shows the original sparse sinusoidal signal. Figure 11b shows the QPS-DADAM algorithm reconstruction, which closely matches the original in its overall shape. Figure 11c demonstrates the reconstruction error, which remains within $[-0.02, 0.02]$. Error values greater than 0 indicate overestimation, and error values less than 0 indicate underestimation. Despite the reduced precision from random quantisation, the QPS-DADAM algorithm effectively recovers the original signal. This demonstrates the algorithm's suitability for distributed sparse signal recovery.

Remark 5.1. It is worth noting that algorithm performance in distributed optimisation is inherently dependent on parameter configurations, and different hyperparameter choices may lead to varying test results. To ensure fair and credible comparisons between the QPS-DADAM algorithm, the PS-DADAM algorithm and the QPS-DO algorithm, all algorithms in our tests adopt unified or analogous key parameters, and these parameters include identical strongly connected directed network topologies, consistent time-varying learning rate rules ($\alpha_t = \alpha/\sqrt{t}$) and base learning rate ($\alpha = 1$), uniform momentum coefficients ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\beta_3 = 0.9$) and the same initial gradient scale. Detailed parameter settings for each task are explicitly presented in the corresponding subsections of Section 5, which minimises parameter-induced biases and ensures result reproducibility, such that the observed performance differences reflect the intrinsic advantages of each algorithm rather than inconsistent parameter effects.

6 | Conclusion

This paper has proposed an efficient quantised push-sum distributed adaptive momentum algorithm for general constrained optimisation problems over directed networks. Theoretical analysis has verified that the QPS-DADAM algorithm converges to the optimal solution of the problem. Numerical simulations of realistic distributed optimisation scenarios have been conducted to illustrate the performance of the QPS-DADAM algorithm. However, the computational efficiency of the algorithm is limited, mainly due to the element-wise operations in the adaptive momentum estimation module and the real-time bit allocation calculation in the dynamic quantiser module, and its ability to overcome privacy attacks remains uncertain. Future research will focus on developing methods to ensure the computational efficiency of the QPS-DADAM algorithm over dynamic directed networks. Additionally, the development of effective privacy-preserving the

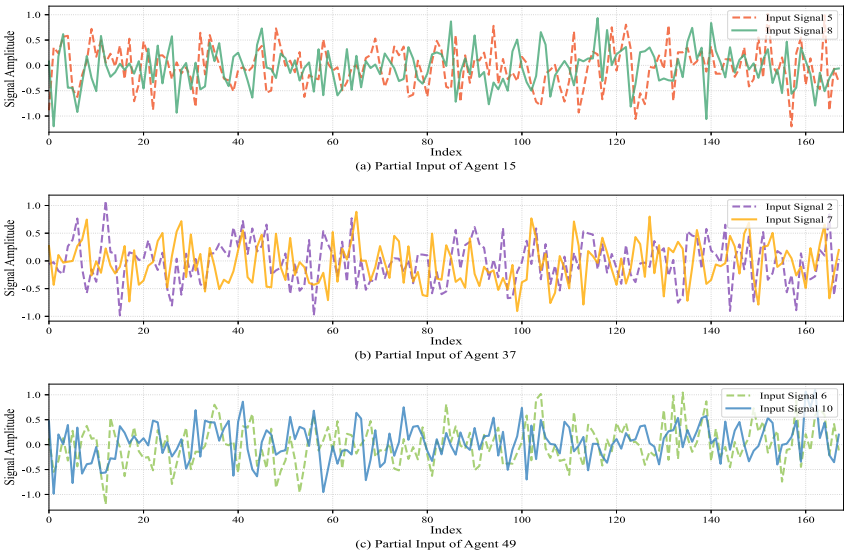


FIGURE 10 | Examples of input signals from three randomly selected agents.

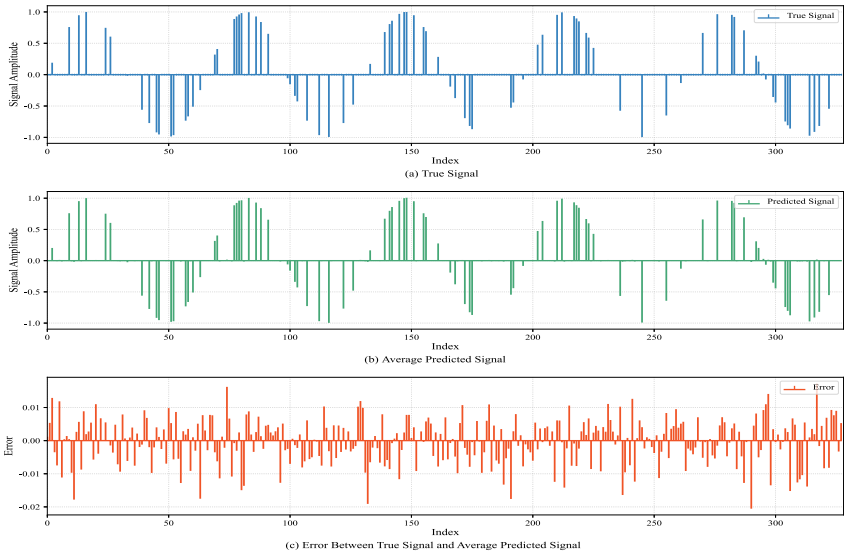


FIGURE 11 | Performance of the QPS-DADAM algorithm for sparse signal compressed sensing.

QPS-DADAM algorithm to achieve privacy security is also important.

Acknowledgements

This work was supported in part by the Natural Science Foundation of Chongqing (Innovation and Development Joint Fund) under Grant CSTB2025NSCQ-LZX0085 and CSTB2023NSCQ-LZX0135; the National Natural Science Foundation of China under Grants 62576053 and 62302068; the Science and Technology Research Programme of Chongqing Municipal Education Commission under Grants KJZD-M202201204, KJZD-K202201205 and KJQN202401232; the National Foreign Expert Programme under Grant S20250332; and the Chongqing Three Gorges University Graduate Research Innovation Project under Grants YJSKY25011.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Research data are not shared.

References

1. R. Yan, Y. Lin, N. Yu, and Y. Wu, "A Low-Carbon Economic Dispatch Model for Electricity Market With Wind Power Based on Improved Ant-Lion Optimisation Algorithm," *CAAI Transactions on Intelligent Technology* 8, no. 1 (2023): 29–39, https://doi.org/10.1049/cit2.12138.

2. Z. Chen, M. Yu, Z. Zhang, and H. Zeng, "A Power Allocation Algorithm Based on Q-Learning for Two-Way Relaying System With Dual-Media Parallel Communication," *IET Signal Processing* 17, no. 2 (2023): e12174, https://doi.org/10.1049/sil2.12174.

3. M. Hong, "A Distributed, Asynchronous, and Incremental Algorithm for Nonconvex Optimization: An Admm Approach," *IEEE Transactions on Control of Network Systems* 5, no. 3 (2018): 935–945, https://doi.org/10.1109/tcns.2017.2657460.

4. T. Yang, X. Yi, J. Wu, et al., "A Survey of Distributed Optimization," *Annual Reviews in Control* 47 (2019): 278–305, https://doi.org/10.1016/j.arcontrol.2019.05.006.

5. A. Gasparin, S. Lukovic, and C. Alippi, "Deep Learning for Time Series Forecasting: The Electric Load Case," *CAAI Transactions on Intelligence Technology* 7, no. 1 (2022): 1–25, https://doi.org/10.1049/cit2.12060.

6. N. Patari, V. Venkataramanan, A. Srivastava, D. K. Molzahn, N. Li, and A. Annaswamy, "Distributed Optimization in Distribution Systems: Use Cases, Limitations, and Research Needs," *IEEE Transactions on Power Systems* 37, no. 5 (2022): 3469–3481, https://doi.org/10.1109/TPWRS.2021.3132348.

7. A. Nedić and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control* 54, no. 1 (2009): 48–61, https://doi.org/10.1109/TAC.2008.2009515.

8. M. Xiong, B. Zhang, D. Yuan, and S. Xu, "Distributed Quantized Mirror Descent for Strongly Convex Optimization Over Time-Varying Directed Graph," *Science China Information Sciences* 65, no. 10 (2022): 202202, https://doi.org/10.1007/s11432-020-3275-3.

9. D. Yuan, B. Zhang, D. W. Ho, W. X. Zheng, and S. Xu, "Distributed Online Bandit Optimization Under Random Quantization," *Automatica* 146, no. 1 (2022): 110590, https://doi.org/10.1016/j.automatica.2022.110590.

10. X. Cao, T. Başar, S. Diggavi, et al., "Communication-Efficient Distributed Learning: An Overview," *IEEE Journal on Selected Areas in Communications* 41, no. 4 (2023): 851–873, https://doi.org/10.1109/jsac.2023.3242710.

11. Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization Design for Unconstrained Distributed Optimization," in *Proceedings of the American Control Conference* (2015), 1229–1234, https://doi.org/10.1109/ACC.2015.7170901.

12. A. I. Rikos, W. Jiang, T. Charalambous, et al., "Distributed Optimization With Gradient Descent and Quantized Communication," in *Proceedings of the 22nd World Congress of the International Federation of Automatic Control (IFAC)* (2023), 6433–6439, https://doi.org/10.1016/j.ifacol.2023.10.100.

13. W. Zhang, Y. Shi, B. Zhang, K. Lu, and D. Yuan, "Quantized Distributed Online Projection-Free Convex Optimization," *IEEE Control Systems Letters* 7, no. 1 (2023): 1837–1842, https://doi.org/10.1109/LCSYS.2023.3282021.

14. A. Nedić, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, "On Distributed Averaging Algorithms and Quantization Effects," *IEEE Transactions on Automatic Control* 54, no. 11 (2009): 2506–2517, https://doi.org/10.1109/TAC.2009.2031203.

15. H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Quantized Push-Sum for Gossip and Decentralized Optimization Over Directed Graphs," *arXiv Preprint arXiv:2002.09964* (2020), https://doi.org/10.48550/arXiv.2002.09964.

16. D. Alistarh, D. Grubic, J. Li, et al., "Qsgd: Communication-Efficient Sgd Via Gradient Quantization and Encoding," *Advances in Neural Information Processing Systems* 30 (2017): 1–12, https://doi.org/10.48550/arXiv.1610.02132.

17. A. Reisizadeh, H. Taheri, A. Mokhtari, et al., "Robust and Communication-Efficient Collaborative Learning," in *Proc. of Neural Information Processing Systems* (2019), 8386–8397, https://doi.org/10.48550/arXiv.2510.08311.

18. A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An Exact Quantized Decentralized Gradient Descent Algorithm," *IEEE Transactions on Signal Processing* 67, no. 19 (2019): 4934–4947, https://doi.org/10.1109/TSP.2019.2932876.

19. J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research* 12, no. 7 (2011): 1–39, https://dl.acm.org/doi/pdf/10.5555/1953048.2021068.

20. D. Kingma, J. B. Adam, "A Method for Stochastic Optimization," in *Proceedings of the Conference on Learning Representations* (2015), 6, https://doi.org/10.48550/arXiv.1412.6980.

21. S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," Computer Science, Accessed: Apr. 19, 2019, https://arxiv.org/abs/1904.09237.

22. K. Okamoto, N. Hayashi, and S. Takai, "Distributed Online Adaptive Gradient Descent With Event Triggered Communication," *IEEE Transactions on Control of Network Systems* 11, no. 2 (2023): 1–12, https://doi.org/10.1109/TCNS.2023.3294432.

23. X. Shen, D. Li, R. Fang, Y. Zhou, and X. Wu, "Distributed Adaptive Online Learning for Convex Optimization With Weight Decay," *Asian Journal of Control* 24, no. 2 (2022): 562–575, https://doi.org/10.1002/asjc.2489.

24. P. Nazari, D. A. Tarzanagh, and G. Michailidis, "Dadam: A Consensus-Based Distributed Adaptive Gradient Method for Online Optimization," *IEEE Transactions on Signal Processing* 70, no. 1 (2022): 6065–6079, https://doi.org/10.1109/TSP.2022.3223214.

25. Y. Zhang, Q. Ai, and Z. Li, "Admm-Based Distributed Response Quantity Estimation: A Probabilistic Perspective," *IET Generation, Transmission & Distribution* 14, no. 26 (2020): 6594–6602, https://doi.org/10.1049/iet-gtd.2020.1380.

26. K. Muhammad, Mustaqeem, A. Ullah, et al., "Human Action Recognition Using Attention Based Lstm Network With Dilated Cnn Features," *Future Generation Computer Systems* 125 (2021): 820–830, https://doi.org/10.1016/j.future.2021.06.045.

27. S. Travin, O. Gromov, and G. Duca, "Statistical Computational Model of Fission Products Composition of Irradiated Nuclear Fuel and Their Contribution to Gas-Aerosol Emissions of Nuclear Power Plants," *Romanian Journal of Information Science and Technology* 27, no. 3–4 (2024): 310–322, https://doi.org/10.59277/romjist.2024.3-4.05.

28. M. S. Kiran and H. Hakli, "A Tree–Seed Algorithm Based on Intelligent Search Mechanisms for Continuous Optimization," *Applied Soft Computing* 98 (2021): 106938, https://doi.org/10.1016/j.asoc.2020.106938.

29. E. K. Zavadskas, H. Dnçer, S. Yüksel, and S. Et, "Intelligent Expert Systems Using Molecular Fuzzy Genetic Algorithms and Multi-Objective Particle Swarm Optimization for Circular-Oriented Project Investments," *Romanian Journal of Information Science and Technology* 28, no. 3 (2025): 260–273, https://doi.org/10.59277/romjist.2025.3.02.

30. K. I. Tsianos, S. Lawlor, and M. G. Rabbat, "Push-Sum Distributed Dual Averaging for Convex Optimization," in *Proceedings of the IEEE Conference on Decision and Control*, (2012), 5453–5458, https://doi.org/10.1109/TSMC.2022.3200680.

31. Y. Li, C. Chen, K. Zhang, et al., "Q-dadam: A Quantized Distributed Online Optimization Algorithm With Adaptive Momentum," *IEEE Transactions on Control of Network Systems* 12, no. 2 (2025): 1540–1551, https://doi.org/10.1109/TCNS.2025.3526555.

32. Q. Lü, X. Liao, T. Xiang, H. Li, and T. Huang, "Privacy Masking Stochastic Subgradient-Push Algorithm for Distributed Online Optimization," *IEEE Transactions on Cybernetics* 51, no. 6 (2021): 3224–3237, https://doi.org/10.1109/TCYB.2020.2973221.

33. K. Zhang, X. Liao, and Q. Lü, "Privacy-Protected Decentralized Dual Averaging Push With Edge-Based Correlated Perturbations Over Time-Varying Directed Networks," *IEEE Transactions on Network Science and Engineering* 9, no. 6 (2022): 4145–4158, https://doi.org/10.1109/TNSE.2022.3195953.

34. L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning," *SIAM Review* 60, no. 2 (2018): 223–311, https://doi.org/10.1137/16m1080173.

35. H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications* (Springer Science & Business Media, 2003), https://doi.org/10.1007/b97441.

36. S. Bubeck, "Convex Optimization: Algorithms and Complexity," *Foundations and Trends in Machine Learning* 8, no. 3–4 (2015): 231–357, https://doi.org/10.1561/2200000050.

37. P. J. Huber, "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics* 35, no. 1 (1964): 73–101, https://doi.org/10.1007/978-1-4612-4380-9_35.

38. J. A. Nelder and R. W. M. Wedderburn, "Generalized Linear Models," *Journal of the Royal Statistical Society: Series A* 135, no. 3 (1972): 370–384, https://doi.org/10.2307/2344614.

39. G. Lan, S. Lee, and Y. Zhou, "Communication-Efficient Algorithms for Decentralized and Stochastic Optimization," *Mathematical Programming* 180, no. 1–2 (2020): 237–284, https://doi.org/10.1007/s10107-018-1355-4.

40. Q. Lü, X. Liao, H. Li, and T. Huang, "A Computation-Efficient Decentralized Algorithm for Composite Constrained Optimization," *IEEE Transactions on Signal and Information Processing over Networks* 6, no. 6 (2020): 774–789, https://doi.org/10.1109/TSIPN.2020.3037837.

41. A. Nedić and A. Olshevsky, Distributed Optimization Over Time-Varying Directed Graphs, (2013), https://doi.org/10.48550/arXiv.1303.2289.

42. Y. Wang, L. Lin, and J. Chen, "Communication-Compressed Adaptive Gradient Method for Distributed Nonconvex Optimization," *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)* 28–30 (2022): 6292–6320, https://doi.org/10.48550/arXiv.2111.00705.

43. Y. Niu, H. Li, Z. Wang, Q. Lu, D. Xia, and L. Ji, "A Distributed Stochastic Proximal-Gradient Algorithm for Composite Optimization," *IEEE Transactions on Control of Network Systems* 8, no. 3 (2021): 1383–1393, https://doi.org/10.1109/TCNS.2021.3065653.

44. Z. Yang, W. He, and Y. Yuan, "Differentially Private Distributed Optimization Over Diagraphs With Application to Image Deblurring," *IEEE Transactions on Control of Network Systems* 12, no. 1 (2024): 634–647, https://doi.org/10.1109/TCNS.2024.3395851.

45. D. Ren, W. Zuo, D. Zhang, L. Zhang, and M. H. Yang, "Simultaneous Fidelity and Regularization Learning for Image Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, no. 1 (2019): 284–299, https://doi.org/10.1109/TPAMI.2019.2926357.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.
Supporting Information S1: cit270122-sup-0001-suppl-data.pdf.