DISTRIBUTED OPTIMIZATION WITH DIRECTION-CONSTRAINED BIASED COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Error feedback (EF) is widely used to ensure convergence in distributed optimization with biased gradient compression. However, EF requires each node to maintain at least one additional model-sized numerical vector, introducing extra memory overhead. To reduce this memory overhead, we investigate an alternative approach that controls compression errors through directional constraints rather than numerical error accumulation. Our descent analysis identifies a sufficient condition for convergence based on the alignment between the global gradient and the aggregated compression error. Specifically, we introduce ACTION (Acute-Angle CondiTION), which requires their inner product to be non-negative. Since individual nodes cannot access the current global gradient before communication, we further propose L-ACTION (Lagged ACTION), which uses the previous aggregated compressed gradient as a reference. By enforcing the directional constraints coordinatewise, L-ACTION requires retaining only the signs of this reference as auxiliary state for compression error control, substantially reducing the memory footprint compared with the numerical error states maintained by EF. We establish convergence guarantees for ACTION under generalized biased compression without requiring compressor contractivity and show that L-ACTION achieves the same convergence rate. Mainstream compression methods can be modified to satisfy the directional constraints with minor adjustments. Experiments demonstrate lower training loss, higher test accuracy, and improved communication efficiency in several settings.

1 INTRODUCTION

The rapid growth of modern machine learning models has introduced substantial computational and memory challenges, making centralized training increasingly impractical. Distributed optimization (Yang et al., 2019; Liu et al., 2026) provides a natural solution by enabling multiple clients to collaboratively train models while distributing the computational burden across devices. In this work, we consider the nonconvex distributed optimization problem (Yuan et al., 2016; Ren et al., 2026):

$$\min_{x\in\mathbb{R}^d}\left\{f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(x)\right\}, \tag{1}$$

where $x\in\mathbb{R}^d$ denotes the model parameters, n is the number of clients, and $f_i(x)$ denotes the loss function of client i.

Distributed Gradient Descent (DGD, Alg. 1) (Yuan et al., 2016; Chen et al., 2018) achieves convergence by communicating full-precision gradients at each iteration. The communication overhead becomes a major bottleneck for large-scale distributed learning. To reduce communication costs, Distributed Compressed Gradient Descent (DCGD, Alg. 2) (Alistarh et al., 2017; Beznosikov et al., 2023; Chen et al., 2026a) applies gradient compression before communication. Existing convergence analyses for DCGD mainly rely on two classes of compressors (Huang et al., 2022; Condat et al., 2022), namely *unbiased compressors* (Alistarh et al., 2017; Mishchenko et al., 2025; Wangni et al., 2018) and *biased compressors* (Alistarh et al., 2018).

Biased compressors have attracted interest for their superior empirical performance in distributed training (Seide et al., 2014). However, directly applying biased compression to local gradients may

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

lead to divergence in distributed optimization (Beznosikov et al., 2023). Error feedback (EF) (Thomsen et al., 2025; Islamov et al., 2025; Gao et al., 2026) is a widely used approach to addressing this issue by retaining information lost during compression and incorporating it into subsequent updates. Representative methods include top-k+EF (Alistarh et al., 2018), EF-SignSGD (Karimireddy et al., 2019), and EF21 (Richtárik et al., 2021; 2024; Fatkhullin et al., 2025). Despite its effectiveness, EF introduces additional memory overhead by maintaining at least one model-sized numerical vector at each node. More elaborate EF-based methods maintain multiple such vectors: EControl (Gao et al., 2024; Thomsen et al., 2026) and EF21-SGDM (Fatkhullin et al., 2023) maintain two, while EF21-SGD2M (Fatkhullin et al., 2023) maintains three. As model sizes grow, these auxiliary states require increasing amounts of memory, motivating alternative approaches that control compression errors with lower auxiliary storage requirements. Moreover, existing convergence guarantees for biased compression typically require compressor contractivity, meaning that the squared compression error is bounded by a fixed fraction strictly less than one of the input's squared norm. This requirement limits their applicability to non-contractive compressors such as sign compression.

To address the above issues, we investigate an alternative approach that controls compression errors through directional constraints rather than numerical error accumulation. Under non-contractive biased compression (which we refer to as generalized biased compression; Definition 1), our descent analysis reveals how the alignment between the global gradient and the aggregated compression error affects objective descent. In particular, a negative inner product between them can hinder the descent induced by the update. This analysis motivates a sufficient condition for convergence, which we term ACTION (Acute-Angle CondiTION): $\langle \nabla f(x^t),\, g^t - \nabla f(x^t) \rangle \geq 0$, where g^t denotes the aggregated compressed gradient. Under smoothness, lower boundedness, and bounded gradient dissimilarity, we establish convergence guarantees for DCGD satisfying ACTION. These guarantees permit non-contractive compressors satisfying the proposed directional conditions.

To enforce ACTION, we design direction-constrained compressors that control the compression error relative to a reference direction. Using the current global gradient as the reference ensures that the aggregated compression error satisfies ACTION. However, this gradient is unavailable to individual nodes before communication, making the construction difficult to implement directly. To address this limitation, we propose L-ACTION (Lagged ACTION), which uses the previous aggregated compressed gradient as the reference. By enforcing the directional constraints coordinatewise, each node only needs to retain the signs of this reference rather than its full numerical values. This reduces the auxiliary memory required for error control compared with EF. We further show that L-ACTION achieves the same convergence rate as ACTION. Finally, mainstream compression methods can be modified to satisfy the proposed directional constraints with minor adjustments. Experiments on image and text classification demonstrate faster convergence and fewer communication rounds to target accuracy. The key contributions of this paper are:

- **Convergence with direction-constrained biased compression.** We identify the alignment between the global gradient and aggregated compression error as a key factor affecting objective descent. Motivated by this finding, we propose a direction-constrained framework that enforces ACTION. Under the stated assumptions, it guarantees convergence without EF even for non-contractive compressors.
- **Compression error control with sign-only auxiliary state.** We propose L-ACTION, which uses the previous aggregated compressed gradient as the reference for directional constraints, retaining only its signs instead of the model-sized numerical state required by EF. Furthermore, we show that L-ACTION achieves the same convergence rate as ACTION.
- **Direction-constrained compressors and empirical evaluation.** Mainstream compression methods, including quantization and sparsification, can be modified to satisfy the proposed directional constraints with minor adjustments. Experiments on image and text classification show that L-ACTION achieves lower training loss and higher test accuracy in several settings while reducing communication rounds to target accuracy.

2 Theoretical Results of Generalized Biased Compression

We consider the following assumptions commonly adopted in distributed optimization. First, we assume smoothness and lower boundedness (Li et al., 2021; Richtárik et al., 2021; Chen et al., 2026b).

Assumption 1 (Smoothness and Lower Boundedness)**.** *For each node i, the local objective function f_i has an L_i-Lipschitz continuous gradient, i.e., $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|, \quad \forall x, y \in \mathbb{R}^d$. Then f is L-smooth with $L \leq \frac{1}{n}\sum_{i=1}^n L_i$ and is lower bounded, i.e., $f^{\inf} := \inf_{x\in\mathbb{R}^d} f(x) > -\infty$.*

Second, we characterize the difference between local and global gradients using the following bounded gradient dissimilarity assumption (Lian et al., 2017; Li et al., 2021).

Assumption 2 (Bounded Gradient Dissimilarity)**.** *There exists a constant $\zeta \geq 0$ such that, for any $x \in \mathbb{R}^d$ and any node i, $\|\nabla f_i(x) - \nabla f(x)\|^2 \leq \zeta^2$.*

Finally, we define the class of compressors considered in this work. EF analyses commonly rely on compressor contractivity (Richtárik et al., 2021; Fatkhullin et al., 2025). Related approaches accommodate non-contractive compressors by requiring their scaled outputs to satisfy a contractivity condition (Yi et al., 2022; Xu et al., 2025; Liao et al., 2022). In contrast, our definition bounds the relative compression error of the original compressor, allowing any $\omega \geq 0$ without requiring contractivity of either the original or a scaled operator.

Definition 1 (Generalized Biased Compression)**.** *We say that a map $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is a generalized biased compressor if there exists a constant $\omega \geq 0$ such that $\|\mathcal{C}(x) - x\|^2 \leq \omega \|x\|^2$.*

Remark 1 (Comparison with EF)**.** *Our convergence analysis does not require contractivity of either the original compressor or a scaled operator. Representative EF methods rely on contractive compressors ($\omega < 1$) (Karimireddy et al., 2019; Beznosikov et al., 2023; Richtárik et al., 2021), while EF-BV accommodates non-contractive compressors through scaling that restores contractivity (Condat et al., 2022). These guarantees therefore do not directly extend to the full compressor class in Definition 1. In contrast, our analysis establishes convergence under the stated assumptions and directional conditions without imposing contractivity.*

2.1 Descent Analysis under Generalized Biased Compression

Building on the descent lemma in Li et al. (2021, Lemma 2), we analyze how compression errors affect objective descent, motivating our direction-constrained approach to convergence without compressor contractivity.

Lemma 1 (Descent Lemma with Inexact Gradient (Li et al., 2021))**.** *Suppose that f is L-smooth and $x^{t+1} = x^t - \eta g^t$, where $g^t \in \mathbb{R}^d$ is an **arbitrary** vector and $\eta > 0$, then:*

$$f(x^{t+1}) \leq f(x^t) - \frac{\eta}{2}\left\|\nabla f(x^t)\right\|^2 - \left(\frac{1}{2\eta} - \frac{L}{2}\right)\left\|x^{t+1} - x^t\right\|^2 + \frac{\eta}{2}\left\|g^t - \nabla f(x^t)\right\|^2. \tag{2}$$

Remark 2 (Challenge of Generalized Biased Compressors)**.** *The descent lemma consists of three key terms. The first term, $\|\nabla f(x^t)\|^2$, represents the ideal gradient term. Since its coefficient is negative, it directly promotes objective descent. The second term, $\left\|x^{t+1} - x^t\right\|^2$, represents the magnitude of the parameter update. When the learning rate satisfies $\eta < 1/L$, this term also contributes to descent. The last term, $\|g^t - \nabla f(x^t)\|^2$, represents the gradient error. Since its coefficient is positive, it hinders objective descent. Therefore, the gradient error term is the key obstacle to convergence. When the compression error is non-contractive, this term may offset the descent induced by the ideal gradient term. This explains why existing convergence analyses typically require contractive compression errors and highlights the key challenge of generalized biased compressors.*

We further analyze the descent relation in Lemma 1 by substituting the update rule $x^{t+1} - x^t = -\eta g^t$ and expanding $\|g^t\|^2 = \|\nabla f(x^t)\|^2 + \|g^t - \nabla f(x^t)\|^2 + 2\langle \nabla f(x^t), g^t - \nabla f(x^t)\rangle$. This yields:

$$\begin{aligned} f(x^{t+1}) \leq f(x^t) &- \left(\eta - \frac{L\eta^2}{2}\right)\left\|\nabla f(x^t)\right\|^2 \\ &+ \frac{L\eta^2}{2}\left\|g^t - \nabla f(x^t)\right\|^2 - (\eta - L\eta^2)\left\langle \nabla f(x^t), g^t - \nabla f(x^t)\right\rangle. \end{aligned} \tag{3}$$

This transformation reduces the first-order effect of the gradient error with respect to the learning rate to a second-order term. Consequently, when the learning rate satisfies $\eta < 1/L$, the error term becomes dominated by the descent term for sufficiently small η. However, this transformation introduces an additional first-order inner-product term $\langle \nabla f(x^t), g^t - \nabla f(x^t)\rangle$, whose sign is not determined. If this inner product is non-negative, it further promotes objective descent.

2.2 ACTION: CONVERGENCE GUARANTEES FOR GENERALIZED BIASED COMPRESSION

Motivated by the aforementioned theoretical analysis, we propose **ACTION** (Acute-Angle Condition), as defined below.

Definition 2 (ACTION). *For any iteration t, we say that the compressed gradient g^t satisfies ACTION if $\langle \nabla f(x^t), g^t - \nabla f(x^t) \rangle \geq 0$.*

ACTION indicates that the compressed gradient does not deviate from the descent direction of the original gradient. In other words, the compressed gradient still preserves a valid descent component along the original gradient direction. ACTION ensures that the compressed global gradient forms an acute angle with the original gradient. According to Definition 2, ACTION implies that $\langle \nabla f(x^t), g^t \rangle \geq \|\nabla f(x^t)\|^2 \geq 0$.

Fig. 1 illustrates two optimization trajectories starting from the same point. With ACTION, the compressed gradients satisfy the directional constraints illustrated by the shaded regions, and the blue trajectory approaches the stationary point. Without ACTION, compression errors may disrupt alignment with the global gradient, causing the red trajectory to move away from the stationary point. This example illustrates how directional constraints help preserve descent under biased compression.

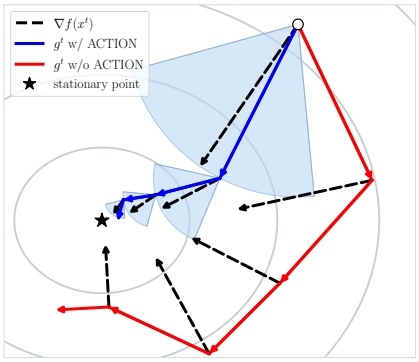

Figure 1: Optimization trajectories with and without ACTION.

Lemma 2 (Global Gradient Error under Generalized Biased Compression). *Under Assumption 2, suppose that at iteration t, each node compresses its local gradient using a generalized biased compressor $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$, and let g^t denote the resulting global compressed gradient. Then, $\|g^t - \nabla f(x^t)\|^2 \leq \omega \left(\|\nabla f(x^t)\|^2 + \zeta^2 \right)$.*

We establish the convergence that satisfies ACTION under generalized biased compressors.

Theorem 1 (Convergence of ACTION). *Let Assumption 1-2 hold, and let the learning rate satisfies $\eta < \min(1, \frac{1}{1+\omega}) \cdot \frac{1}{L}$. The communication round is $T \geq 1$ and let $\hat{x}^T$ be chosen from the iterates $x^0, x^1, ..., x^{T-1}$ uniformly at random. Then, the convergence of ACTION under generalized biased compression is,*

$$\mathbb{E}\left[\left\| \nabla f(\hat{x}^T) \right\|^2 \right] \leq \frac{2(f(x^0) - f^{\inf})}{\eta T} + \omega L \eta \zeta^2. \tag{4}$$

By choosing the learning rate $\eta = \Theta(1/\sqrt{T})$, the gradient converges to zero at the rate $\mathcal{O}(1/\sqrt{T})$. To enforce ACTION, we propose direction-constrained biased compressors that constrain compression errors relative to a reference direction.

Definition 3 (Direction-Constrained Compressors). *We say that a generalized biased compressor $\mathcal{C}$ is a direction-constrained compressor with respect to $y \in \mathbb{R}^d$ if $[y]_j [\mathcal{C}(x) - x]_j \geq 0$ for all $x \in \mathbb{R}^d$ and all $j \in [d]$. The family of such compressors is denoted by $\mathbb{B}(\omega, y)$.*

At iteration t, each node applies the direction-constrained biased compressor $\mathcal{C}^t \in \mathbb{B}(\omega, \nabla f(x^t))$ to its local gradient, yielding $g_i^t = \mathcal{C}^t(\nabla f_i(x^t))$. By the definition of the direction-constrained compressor, we have $\langle \nabla f(x^t), g_i^t - \nabla f_i(x^t) \rangle \geq 0$. By the fact that $\nabla f(x^t) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^t)$, it follows that $\langle \nabla f(x^t), g^t - \nabla f(x^t) \rangle \geq 0$, which guarantees that ACTION is satisfied.

2.3 L-ACTION: ACTION WITH LAGGED GLOBAL GRADIENT

Although ACTION provides a theoretical mechanism to guarantee convergence, it is not directly implementable in distributed optimization, since the current global gradient is unavailable before local gradients are compressed. To address this limitation, we use the aggregated gradient from the previous iteration as the constraint vector, leading to L-ACTION (Lagged ACTION).

Definition 4 (L-ACTION). *Initialize $g^{-1} = 0$. For any iteration t, L-ACTION requires the following condition for the inexact gradient used in the parameter update $\left\langle g^{t-1}, g^t - \nabla f(x^t) \right\rangle \geq 0$.*

At iteration t, each node applies a direction-constrained compressor $\mathcal{C}^t \in \mathbb{B}(\omega, g^{t-1})$ to its gradient, yielding $g_i^t = \mathcal{C}^t(\nabla f_i(x^t))$. The compressor ensures that each local compression error has a non-negative inner product with the shared reference g^{t-1}. Since the aggregated compression error is the average of the local compression errors, it inherits this property and thus satisfies L-ACTION.

L-ACTION uses g^{t-1} as the available reference direction, while the exact gradient $\nabla f(x^{t-1})$ provides a connection to the current global gradient through smoothness. Therefore, it is important to characterize the coordinatewise sign consistency between the compressed gradient and the global gradient. Although many compressors preserve the signs of individual local gradient coordinates or set some coordinates to zero, compression combined with aggregation over heterogeneous local gradients may change the relative contributions of positive and negative components. Consequently, the aggregated compressed gradient g^{t-1} may become sign-inconsistent with the global gradient $\nabla f(x^{t-1})$ on a subset of coordinates. To quantify the impact of these sign-inconsistent coordinates, we define $R_t := \frac{\sum_{j \in S_t^-} [\nabla f(x^t)]_j^2}{\|\nabla f(x^t)\|^2}$, where $S_t^- = \{j \mid [g^t]_j [\nabla f(x^t)]_j < 0\}$. The quantity R_t measures the fraction of the gradient energy contained in sign-inconsistent coordinates at iteration t. We further define $r := \sup_t R_t$ to characterize the maximum level of sign inconsistency throughout the optimization process, the range of r is $[0, 1]$. Then we can establish the following convergence theorem for L-ACTION.

Theorem 2 (Convergence of L-ACTION). *Suppose that Assumptions 1-2 hold. At every iteration t, suppose that each node uses a compressor $\mathcal{C}^t \in \mathbb{B}(\omega, g^{t-1})$. If $r < \frac{20\omega+10-\sqrt{(20\omega+10)^2-4}}{2}$ and the learning rate satisfies $\eta < \frac{(1+r)+\sqrt{(1+r)^2-4(5\omega+3)r}}{(10\omega+6)L}$. The communication round is $T \geq 1$ and let $\hat{x}^T$ be chosen from the iterates $x^0, x^1, ..., x^{T-1}$ uniformly at random. Then, the convergence of L-ACTION under generalized biased compression is,*

$$\mathbb{E}\left[\left\|\nabla f(\hat{x}^T)\right\|^2\right] \leq \frac{2(f(x^0) - f^{\inf})}{\eta T} + 5\omega L\eta\zeta^2 + \frac{2(1+\omega)L\eta\|f(x^0)\|^2}{T} + \frac{(1-L\eta)r\|f(x^0)\|^2}{L\eta T} + \frac{\left\|g^0 - \nabla f(x^0)\right\|^2}{T} \tag{5}$$

With the choice of learning rate $\eta = \Theta(1/\sqrt{T})$, the first two terms in the convergence bound decay at the rate of $\mathcal{O}(1/\sqrt{T})$. The third term, which corresponds to the initialization gradient magnitude, decays faster at the rate of $\mathcal{O}(1/T^{3/2})$. The fourth term, which captures the effect of sign-inconsistent coordinates through the energy ratio r, converges to zero at the rate of $\mathcal{O}(1/\sqrt{T})$. The last term, corresponding to the initial compression error, decreases at the rate of $\mathcal{O}(1/T)$. Therefore, under an appropriate learning rate choice, all terms in the convergence bound vanish as T increases, yielding the standard nonconvex convergence rate of $\mathcal{O}(1/\sqrt{T})$. Since this convergence guarantee requires r to satisfy the condition stated in the theorem, we evaluate its magnitude across compression methods, heterogeneity levels, gradient dimensions, and numbers of nodes in Appendix F. The results show that r is zero or negligible in most evaluated settings, with appreciable values arising only under strong heterogeneity combined with low gradient dimensionality and aggressive compression.

3 Direction-constrained Compressors

This section provides examples showing that the rounding quantization (Ren & Luo, 2026) and the top-k (Bereyhi et al., 2025) sparsification can be modified with a minor adjustment to become direction-constrained compressors. For completeness, detailed descriptions of rounding quantization and top-k sparsification are provided in the Appendix G.

3.1 Direction-constrained Quantization

Quantization does not satisfy the directional constraint, as we show next. For example, when the reference direction y_j is positive, rounding x_j/s to the lower quantization level may result in a

quantization error with the opposite sign, such that $y_j(\hat{x}_j - x_j) < 0$. Therefore, the rounding quantization does not generally satisfy the directional constraint. This observation suggests that only a minor modification of the rounding operation is needed to construct a direction-constrained biased quantization. Specifically, we replace the rounding operation with the direction-constrained rounding function, denoted by $\mathrm{dc\text{-}round}$. For any coordinate index $i \in [d]$, any input vector $v \in \mathbb{R}^d$, and any reference direction $y \in \mathbb{R}^d$, we define:

$$\mathrm{dc\text{-}round}(v_i, y_i) = \begin{cases} \lceil v_i \rceil, & \text{if } y_i > 0, \\ \lfloor v_i \rfloor, & \text{if } y_i < 0, \\ \mathrm{round}(v_i), & \text{otherwise.} \end{cases} \tag{6}$$

Therefore, by replacing the rounding operation in the quantization with $\mathrm{dc\text{-}round}$, the quantization becomes a direction-constrained biased compressor (see Appendix H.1). The detailed encoding and decoding procedures are presented in Algorithm 4.

3.2 Direction-constrained Sparsification

Sparsification does not satisfy the directional constraint, as we show next. When a discarded element has the same sign as the corresponding component of the constraint direction, the resulting compression error may violate the directional constraint. Specifically, a discarded element produces an error of $-x_i$. If both x_i and y_i are nonzero and have the same sign, then $-x_i y_i < 0$, which violates the directional constraint. Therefore, we modify the sparsification procedure as follows.

Given an input vector $x \in \mathbb{R}^d$ to be compressed and a constraint vector $y \in \mathbb{R}^d$, we first partition the indices into a safe index set $S^{\text{safe}} = \{i \mid i \in [d], x_i y_i \leq 0\}$ and an unsafe index set $S^{\text{unsafe}} = \{i \mid i \in [d], x_i y_i > 0\}$. For an index in S^{safe}, discarding x_i produces an error of $-x_i$, which satisfies $-x_i y_i \geq 0$. Thus, discarding such an element does not violate the directional constraint. In contrast, discarding any index in S^{unsafe} would violate the directional constraint.

We consider two cases depending on whether the communication budget k is sufficient to retain all unsafe indices, i.e., $k \geq |S^{\text{unsafe}}|$ or $k < |S^{\text{unsafe}}|$. First, suppose that $k \geq |S^{\text{unsafe}}|$. In this case, we retain all indices in S^{unsafe} and select the $k - |S^{\text{unsafe}}|$ elements with the largest magnitudes from S^{safe}. The resulting compression error is then generated only at the unselected indices in S^{safe}. Therefore, the directional constraint is satisfied. Second, suppose that $k < |S^{\text{unsafe}}|$. In this case, the communication budget is insufficient to retain all unsafe indices. Since directly discarding any index in S^{unsafe} violates the directional constraint, we introduce an additional d-dimensional mask to record the unsafe indices that cannot be transmitted explicitly. If the budget k is sufficiently large, we reduce the number of transmitted elements from k to $k' := \lfloor k - d/32 \rceil$ to offset the communication cost of the mask and maintain the same overall communication budget. Otherwise, we set $k' = k$. We then select the k' elements with the largest magnitudes from S^{unsafe} for transmission. For the remaining $|S^{\text{unsafe}}| - k'$ indices, the mask records their locations. Their values are recovered according to the sign of the reference vector, i.e., an index i is assigned M if $y_i > 0$ and m if $y_i < 0$, where M and m denote the maximum and minimum values among the corresponding elements, respectively.

Through the aforementioned modifications, the sparsification becomes a direction-constrained biased compressor (see Appendix H.2). The detailed encoding and decoding procedures are presented in Algorithm 5.

4 Experiments

4.1 Experimental Setup

Models and Datasets We evaluate our method on image classification and natural language processing tasks. For image classification, we consider MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), CIFAR10 and CIFAR100 (Krizhevsky, 2009), using CNN, VGG16 (Simonyan & Zisserman, 2015), and ResNet18 (He et al., 2016) models. For natural language processing, we evaluate on AG_News (Zhang et al., 2015) and IMDB (Maas et al., 2011) using bidirectional LSTM (Yu et al., 2019) and Transformer (Vaswani et al., 2017) models. Detailed model architectures and hyperparameters are provided in Appendix I.

Baselines For quantization, we compare the proposed direction-constrained quantization (DC-Q) with several representative quantization methods, including standard quantization (Q) (Vargaftik et al., 2022) and quantization combined with the EF21 framework (EF21-Q) (Richtárik et al., 2021; Fatkhullin et al., 2025). For sparsification, we compare the proposed direction-constrained sparsification (DC-S) with representative sparsification methods, including standard top-k sparsification (S) (Alistarh et al., 2018) and top-k sparsification combined with the EF21 framework (EF21-S) (Richtárik et al., 2021; Fatkhullin et al., 2025). When DC-Q and DC-S are applied to ACTION and L-ACTION, respectively, we abbreviate the four resulting combinations as ACTION+DC-Q (**A-Q**), L-ACTION+DC-Q (**LA-Q**), ACTION+DC-S (**A-S**), and L-ACTION+DC-S (**LA-S**).

4.2 Experimental Results

Validation of ACTION To validate the effectiveness of ACTION, we conduct an idealized experiment using a CNN model on MNIST and Fashion-MNIST. Specifically, we aggregate the uncompressed local gradients to obtain the current global gradient and use it as the constraint information for gradient compression. Although this setting is not directly applicable to practical distributed training due to the unavailability of the current global gradient before compression, it allows us to independently evaluate the theoretical properties of ACTION under ideal information. We record the average training loss, test accuracy, and ℓ_2 norm of the global gradient after each communication round. Results on Fashion-MNIST are provided in Appendix J.1.

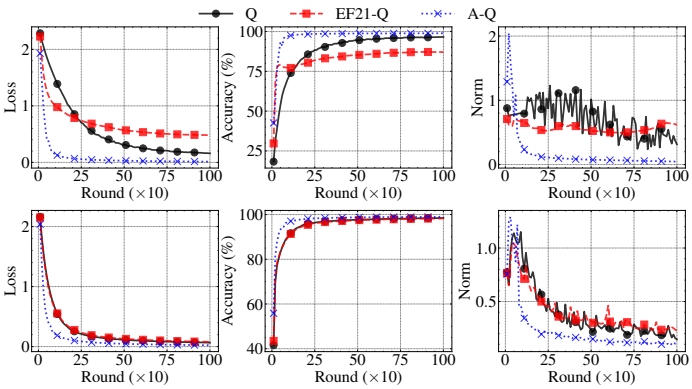

Figure 2: Comparison of Quantizations for CNN

The results of the CNN model on MNIST are shown in Fig. 2 and Fig. 3. In both figures, the first, second, and third columns correspond to the average training loss, test accuracy, and global gradient ℓ_2 norm, respectively. For quantization, the first and second rows correspond to 2bit and 4bit quantization, respectively, while for sparsification, the first and second rows correspond to $k = 1000$ and $k = 2000$, respectively.

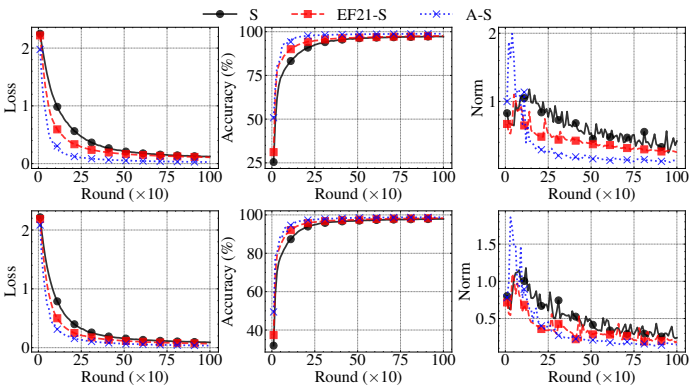

Figure 3: Comparison of Sparsifications for CNN

For quantization, the results show that A-Q consistently achieves the best performance in terms of both training loss and test accuracy under the idealized setting. In the 2bit quantization setting, A-Q converges to a lower training loss and achieves higher test accuracy than the other methods, while also exhibiting a faster convergence rate. With 4bit quantization, the performance gap becomes less pronounced as the quantization error is reduced with a larger bit-width. Nevertheless, A-Q still exhibits faster convergence than the other methods. Moreover, the global gradient ℓ_2 norm provides direct evidence of the convergence behavior predicted by ACTION. The global gradient norm under A-Q decreases more rapidly and exhibits a more stable convergence than those of the non-ACTION methods, indicating that the direction constraint introduced by ACTION effectively guides the compressed gradient toward a descent direction under biased compression.

For sparsification, A-S achieves the best performance in terms of both training loss and test accuracy, while also exhibiting faster convergence. Under the $k = 1000$ communication budget, the performance improvement of A-S is more pronounced. When the budget is increased to $k = 2000$, the performance gap becomes smaller because more gradient information is retained and the compression error is reduced. Nevertheless, A-S still provides faster convergence in terms of both training loss and test accuracy. In terms of the global gradient ℓ_2 norm, A-S exhibits a clear convergence. This difference is mainly due to the communication-preserving design of DC-S. Under the same communication budget, DC-S needs to reduce k to accommodate the additional communication cost of the mask, resulting in fewer explicitly transmitted gradient coordinates. This can slightly weaken the convergence of the global gradient.

Performance of L-ACTION This section further evaluates the proposed method on larger-scale models and datasets, covering both image classification and natural language processing tasks. To provide a concise presentation in the main text, we report the results of ResNet18 under different quantization bit-widths and Transformer under different sparsification budgets, while the results of the remaining experiments are provided in the Appendix J.2. The results for ResNet18 on CIFAR100 are shown in Fig. 4. The first, second, and third columns correspond to 4bit, 6bit, and 8bit quantization, respectively, while the first and second rows show the average training loss and test accuracy. The results for Transformer on IMDB are shown in Fig. 5. The first, second, and third columns correspond to 0.2M, 0.3M, and 0.4M in k, respectively, while the first and second rows show the average training loss and test accuracy, respectively.

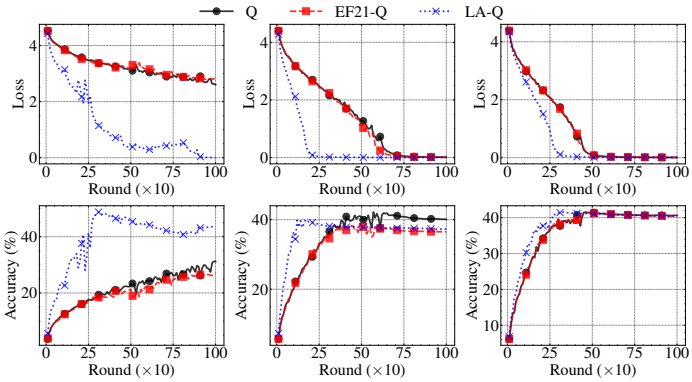

Figure 4: Comparison of Quantizations for ResNet18

For the quantization results, LA-Q demonstrates a particularly clear advantage under the 4bit setting. It reaches a substantially lower training loss and a higher test accuracy than the other methods within approximately 500 communication rounds, while the other methods remain at considerably higher training loss and lower test accuracy throughout the training process. Under the 6bit setting, LA-Q still exhibits a faster convergence rate, reaching a near-converged loss level much earlier than the other methods. Although the final test accuracy gap becomes smaller, LA-Q maintains competitive model performance throughout training. With 8bit quantization, the performance differences among the methods become less pronounced, as the quantization error is substantially reduced. Nevertheless, LA-Q still exhibits a fast convergence trend and reaches a competitive final accuracy.

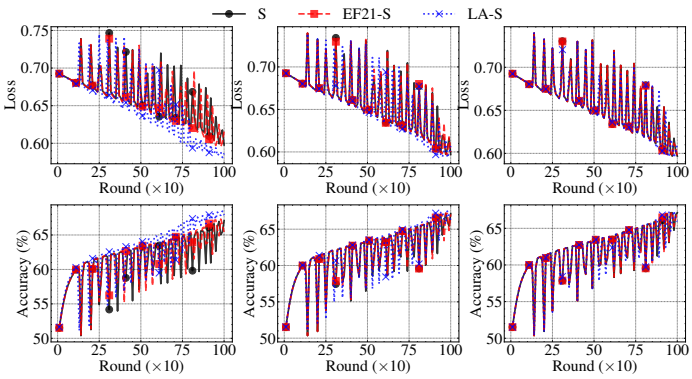

Figure 5: Comparison of Sparsifications for Transformer

For the sparsification results, LA-S achieves its most noticeable improvement under the $k = 0.2$M communication budget, LA-S achieves the lowest training loss and the highest test accuracy among the evaluated methods in the later stage of training. When the sparsification budget is increased to $k = 0.3$M, LA-S continues to exhibit competitive convergence and model performance, although the

performance gap becomes smaller. With $k = 0.4$M, the three methods exhibit increasingly similar training trends and performance, indicating that the benefit of the direction constraint becomes less pronounced as more gradient coordinates are retained.

Communication Overhead Comparison We further compare the number of communication rounds required to reach specified target accuracies. Since all methods use the same communication cost per round under each quantization setting, the reduction in the number of communication rounds directly corresponds to a proportional reduction in the total communication cost. In the main text, we report the number of communication rounds for ResNet18 (Table 1) and Transformer (Table 2) under different quantization settings, while other results are provided in the Appendix J.3.

Table 1: Communication Rounds for ResNet18 under Quantizations

Target Acc.	Quant. Bits	Communication Round		
		Q	EF21-Q	LA-Q
30%	4bit	<u>856</u>	–	**132**
	6bit	185	<u>181</u>	**67**
	8bit	<u>144</u>	147	**106**
40%	4bit	–	–	**208**
	6bit	<u>333</u>	438	**114**
	8bit	292	<u>291</u>	**258**

As shown in Table 1, LA-Q consistently requires the fewest communication rounds to reach all considered target accuracies for ResNet18 across different quantizations. The advantage is particularly pronounced under aggressive 4bit quantization. For example, to reach 30% accuracy, LA-Q requires only 132 communication rounds, compared with 856 rounds for Q, while EF21-Q fails to reach the target within the maximum number of communication rounds.

The results on Transformer further demonstrate the communication efficiency of LA-Q. As shown in Table 2, LA-Q achieves the fewest communication rounds for all considered target accuracies and quantization bit-widths for which the target accuracy is attainable. For the 60% target accuracy, LA-Q requires only 54, 27, and 47 communication rounds under 4bit, 6bit, and 8bit quantization, respectively, which are substantially fewer than the corresponding baselines.

Table 2: Communication Rounds for Transformer under Quantizations

Target Acc.	Quant. Bits	Communication Round		
		Q	EF21-Q	LA-Q
60%	4bit	194	<u>107</u>	**54**
	6bit	123	<u>107</u>	**27**
	8bit	107	<u>105</u>	**47**
70%	4bit	–	–	**287**
	6bit	–	–	**292**
	8bit	–	–	**453**

Overall, these results demonstrate that L-ACTION improves communication efficiency across different architectures. This highlights the effectiveness of L-ACTION in reducing communication overhead while maintaining reliable convergence performance in distributed optimization.

5 Conclusion

We presented ACTION and its practical lagged variant, L-ACTION, for distributed optimization with generalized biased compression, both of which are supported by theoretical convergence guarantees. ACTION controls compression errors through directional constraints, while L-ACTION uses the previous aggregated compressed gradient as a reference and retains only its signs, reducing auxiliary storage compared with numerical error accumulators. We derived convergence bounds that accommodate non-contractive compressors and developed direction-constrained quantization and sparsification schemes. The empirical results demonstrate the effectiveness of the proposed methods across different settings. Direction-constrained quantization reduces communication rounds to target accuracies in several evaluated settings, while sparsification achieves competitive gains under different communication budgets.

Future work will focus on mitigating the impact of compression-induced sign reversals in L-ACTION and improving global gradient sign prediction for a more accurate approximation of ACTION.

AI USE STATEMENT

In this work, we used generative AI tools to assist with checking the correctness of theoretical analysis. We have not used generative AI tools for generating synthetic datasets. Additionally, we used generative AI tools for language polishing. We have reviewed all AI-assisted work. In particular, we checked the AI feedback on our theoretical analysis by examining the mathematical derivations and verifying their consistency with the stated assumptions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ali Bereyhi, Ben Liang, Gary Boudreau, and Ali Afana. Regularized top-k: A bayesian framework for gradient sparsification. *IEEE Transactions on Signal Processing*, 73:4463–4478, 2025.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

Chuyan Chen, Yutong He, Pengrui Li, Weichen Jia, and Kun Yuan. Greedy low-rank gradient compression for distributed learning with convergence guarantees. *IEEE Transactions on Signal Processing*, 74:937–951, 2026a.

Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. LAG: Lazily aggregated gradient for communication-efficient distributed learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Wei Chen, Zidong Wang, Jimmy Chih-Hsien Peng, and Guo-Ping Liu. A novel distributed optimization algorithm with linear convergence for economic dispatch under compressed communication. *IEEE Transactions on Industrial Informatics*, (9), 2026b.

Laurent Condat, Kai Yi, and Peter Richtárik. EF-BV: A unified theory of error feedback and variance reduction mechanisms for biased and unbiased compression in distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ilyas Fatkhullin, Alexander Tyurin, and Peter Richtárik. Momentum provably improves error feedback! In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ilyas Fatkhullin, Igor Sokolov, Eduard Gorbunov, Zhize Li, and Peter Richtárik. EF21 with bells & whistles: Six algorithmic extensions of modern error feedback. *Journal of Machine Learning Research*, 26(189):1–50, 2025.

Yuan Gao, Rustem Islamov, and Sebastian Stich. Econtrol: Fast distributed optimization with compression and error control. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Yuan Gao, Anton Rodomanov, Jeremy Rack, and Sebastian Stich. Composite optimization with error feedback: the dual averaging approach. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Xinmeng Huang, Yiming Chen, Wotao Yin, and Kun Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Rustem Islamov, Yarden As, and Ilyas Fatkhullin. Safe-EF: Error feedback for nonsmooth constrained optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes signSGD and other gradient compression schemes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, Canada, April 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Yiwei Liao, Zhuorui Li, Kun Huang, and Shi Pu. A compressed gradient tracking method for decentralized optimization with linear convergence. *IEEE Transactions on Automatic Control*, 67 (10):5622–5629, 2022.

Shuai Liu, Youqing Hua, Qing-Long Han, and Lihua Xie. Distributed optimization under information constraints: A survey. *IEEE Transactions on Industrial Informatics*, 22(4):2653–2666, 2026.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *Optimization Methods and Software*, 40(5):1181–1196, 2025.

Xiaoxing Ren, Nicola Bastianello, Karl H. Johansson, and Thomas Parisini. Communication-efficient stochastic distributed learning. *IEEE Transactions on Automatic Control*, 71(9):5741–5756, 2026.

Xiaozhe Ren and Qiong Luo. BitDP: Ultra-low-bit communication for data parallelism in llm training. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Peter Richtárik, Elnur Gasanov, and Konstantin Burlachenko. Error feedback reloaded: From quadratic to arithmetic mean of smoothness constants. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Interspeech*, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Daniel Berg Thomsen, Adrien Taylor, and Aymeric Dieuleveut. Tight analyses of first-order methods with error feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Daniel Berg Thomsen, Adrien Taylor, and Aymeric Dieuleveut. A tight theory of error feedback algorithms in distributed optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Shay Vargaftik, Ran Ben Basat, Amit Portnoy, Gal Mendelson, Yaniv Ben Itzhak, and Michael Mitzenmacher. EDEN: Communication-efficient and robust distributed mean estimation for federated learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Lei Xu, Xinlei Yi, Guanghui Wen, Yang Shi, Karl H Johansson, and Tao Yang. Compressed gradient tracking algorithms for distributed nonconvex optimization. *Automatica*, 177:112286, 2025.

Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

Xinlei Yi, Shengjun Zhang, Tao Yang, Tianyou Chai, and Karl Henrik Johansson. Communication compression for distributed nonconvex optimization. *IEEE Transactions on Automatic Control*, 68(9):5477–5492, 2022.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270, 2019.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

A NOTATIONS

The main notations used in this paper are summarized in Table 3.

Table 3: Summary of notations.

Notation	Description
d	Dimension of the model parameter vector
n	Number of computing nodes
x^t	Model parameter at iteration t
$f_i(x)$	Local objective function of node i
$f(x)$	Global objective function, $\frac{1}{n}\sum_{i=1}^n f_i(x)$
$\nabla f_i(x)$	Local gradient of node i
$\nabla f(x)$	Global gradient, $\frac{1}{n}\sum_{i=1}^n \nabla f_i(x)$
$\mathcal{C}(\cdot)$	Gradient compression operator
g_i^t	Compressed local gradient, $\mathcal{C}(\nabla f_i(x^t))$
g^t	Aggregated compressed gradient, $\frac{1}{n}\sum_{i=1}^n g_i^t$
η	Learning rate
L	Smoothness constant of f
ζ	Gradient heterogeneity parameter
ω	Compression error parameter in generalized biased compression
$f_{\inf}$	Lower bound of the global objective function
R_t	Gradient energy fraction in sign-inconsistent coordinates
$\langle x, y\rangle$	Inner product between vectors x and y
$\|x\|$	Norm of vector x
$[x]_j$	The j-th coordinate of vector x
$[d]$	Index set $\{1, \ldots, d\}$

B ALGORITHMS

We provide the detailed implementations of the optimization algorithms and the proposed direction-constrained compressors in this appendix.

The detailed implementation of Distributed Gradient Descent (DGD) is provided in Algorithm 1. The standard DGD procedure computes and aggregates full-precision local gradients before updating the model parameters. The implementation of Distributed Compressed Gradient Descent (DCGD) is presented in Algorithm 2. Each node compresses its local gradient before communication, and the master node aggregates the compressed gradients for model updating.

DCGD under L-ACTION is illustrated in Algorithm 3. Although L-ACTION requires the global gradient g^{t-1} from the previous iteration, in practice, it is not necessary to store the exact values of the previous global gradient. Only the sign information of the reference direction, $r^{t-1} := \text{sign}(g^{t-1})$, needs to be retained to construct a compressor satisfying L-ACTION. This substantially reduces the storage requirement. At initialization, since no historical global gradient information is available, we use an all-zero vector as the initial reference direction, $r^{-1} = \mathbf{0} \in \mathbb{R}^d$. At iteration t, each computing node first computes its local gradient $\nabla f_i(x^t)$ based on the current model parameters and its local data. Each node then constructs a direction-constrained biased compressor $\mathcal{C}^t \in \mathbb{B}(\omega, r^{t-1})$ using the stored sign information of the previous global gradient and compresses its local gradient, $g_i^t := \mathcal{C}^t(\nabla f_i(x^t))$. The compressed gradients are then transmitted to the master node, which aggregates them as $g^t := \frac{1}{n}\sum_{i=1}^n g_i^t$, and broadcasts the aggregated gradient to all computing nodes. Each node then updates its model parameters according to $x^{t+1} := x^t - \eta g^t$. Meanwhile, all nodes compute the sign information of the aggregated global gradient, $r^t := \text{sign}(g^t)$, and store it as the reference direction for the next iteration.

Algorithm 4 describes the encoding and decoding procedures of direction-constrained quantization (DC-Q). The proposed direction-constrained rounding operation modifies the conventional quantization procedure to satisfy the required direction constraints. Algorithm 5 provides the implementation

of direction-constrained sparsification (DC-S). The algorithm modifies the top-k sparsification procedure by partitioning coordinates according to the constraint direction and reconstructing discarded elements accordingly.

Algorithm 1 Distributed Gradient Descent(DGD)

Require: starting point $x^0 \in \mathbb{R}^d$, learning rate $\eta > 0$
1: **for** $t = 0, 1, ..., T-1$ **do**
2: **for all** nodes $i = 1, ..., n$ **in parallel do**
3: Update gradient $\nabla f_i(x^t)$ to master node
4: **end for**
5: The master node aggregates gradients via $\nabla f(x^t) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^t)$
6: The master node computes $x^{t+1} = x^t - \eta \nabla f(x^t)$ and broadcasts x^{t+1} to all clients
7: **end for**

Algorithm 2 Distributed Compressed Gradients Descent (DCGD)

Require: Initial parameter $x^0 \in \mathbb{R}^d$, learning rate $\eta > 0$, compressor $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$
1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: **for all** nodes $i = 1, \ldots, n$ **in parallel do**
3: Compute local gradient $\nabla f_i(x^t)$
4: Send compressed $g_i^t := \mathcal{C}(\nabla f_i(x^t))$ to the master node
5: **end for**
6: The master node aggregates gradients via $g^t := \frac{1}{n}\sum_{i=1}^n g_i^t$
7: The master node computes $x^{t+1} = x^t - \eta g^t$ and broadcasts x^{t+1} to all clients
8: **end for**

Algorithm 3 DCGD under L-ACTION

Require: Initial parameters $x^0 \in \mathbb{R}^d$, learning rate $\eta > 0$
1: Initialize the reference direction $r^{-1} = \mathbf{0} \in \mathbb{R}^d$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: **for all** nodes $i = 1, \ldots, n$ **in parallel do**
4: Compute the local gradient $\nabla f_i(x^t)$
5: Construct the compressor $\mathcal{C}^t \in \mathbb{B}(\omega, r^{t-1})$
6: Compress $g_i^t := \mathcal{C}^t(\nabla f_i(x^t))$
7: If $t > 0$, set $[g_i^t]_j := [\nabla f_i(x^t)]_j$ for all j with $r_j^{t-1} = 0$
8: Upload g_i^t to the master node
9: **end for**
10: The master node aggregates the local gradients: $g^t := \frac{1}{n}\sum_{i=1}^n g_i^t$
11: The master node broadcasts g^t; all nodes update: $x^{t+1} := x^t - \eta g^t$
12: All nodes record $r^t := \text{sign}(g^t)$
13: **end for**

C Proof of Lemma 2

Under generalized biased compression, since the compression operation is applied to the local gradient at each node, we have the following lemma to characterize the resulting global gradient error.

Proof. By Jensen's inequality, the left-hand side can be bounded as

$$\left\|g^t - \nabla f(x^t)\right\|^2 \leq \frac{1}{n}\sum_{i=1}^n \left\|g_i^t - \nabla f_i(x^t)\right\|^2, \tag{7}$$

Algorithm 4 DC-Q

Require: Input vector $x \in \mathbb{R}^d$, constraint vector $y \in \mathbb{R}^d$, integer quantization bit-width $b \geq 2$

Encoding:
Set $K = 2^{b-1} - 1$ and $a = \|x\|_\infty$
if $a > 0$ **then**
  Set the quantization scale $s = a/K$
else
  Set $s = 1$
end if
Compute $q_i = \text{dc-round}(x_i/s, y_i) + K$ for all $i \in [d]$
Transmit the integer codes q and the scale s

Decoding:
Reconstruct $\hat{x}_i = (q_i - K)s$ for all $i \in [d]$

Algorithm 5 DC-S

Require: Input vector $x \in \mathbb{R}^d$, constraint vector $y \in \mathbb{R}^d$, sparsification parameter k

Encoding:
Partition the index set into the safe index set S^{safe} and the unsafe index set S^{unsafe}
if $k \geq |S^{\text{unsafe}}|$ **then**
  Select the $k - |S^{\text{unsafe}}|$ indices with the largest magnitudes from S^{safe}, denoted by $\mathcal{I}_1$
  Set the transmitted index set as $\mathcal{I} = \mathcal{I}_1 \cup S^{\text{unsafe}}$
  Transmit the index set $\mathcal{I}$ and the corresponding values V
else
  Initialize a d-dimensional all-false mask mask and set $k' = \lfloor k - d/32 \rceil$; if $k' \leq 0$, set $k' = k$
  Select the k' indices with the largest magnitudes from S^{unsafe}, denoted by $\mathcal{I}$
  Set the unselected indices in S^{unsafe} to true in mask, and record their maximum and minimum values as M and m
  Transmit $\mathcal{I}$, the corresponding values V, mask, M, and m
end if

Decoding:
Initialize a d-dimensional all-zero vector $\hat{x}$
Fill the entries of $\hat{x}$ at the indices in $\mathcal{I}$ with the corresponding values in V
if mask exists **then**
  For each index i such that $\text{mask}_i = \text{true}$:
  if $y_i > 0$ **then**
    Set $\hat{x}_i = M$
  else if $y_i < 0$ **then**
    Set $\hat{x}_i = m$
  else
    Set $\hat{x}_i = 0$
  end if
end if

where $g_i^t = \mathcal{C}(\nabla f_i(x^t))$ denotes the compressed local gradient. By the generalized biased compression property, we have

$$\left\|g^t - \nabla f(x^t)\right\|^2 \leq \frac{\omega}{n} \sum_{i=1}^{n} \left\|\nabla f_i(x^t)\right\|^2. \tag{8}$$

Expanding the term on the right-hand side gives

$$\frac{\omega}{n} \sum_{i=1}^{n} \left\|\nabla f_i(x^t)\right\|^2 = \frac{\omega}{n} \sum_{i=1}^{n} \Big[\left\|\nabla f(x^t)\right\|^2 + \left\|\nabla f_i(x^t) - \nabla f(x^t)\right\|^2 + 2 \left\langle \nabla f(x^t), \nabla f_i(x^t) - \nabla f(x^t) \right\rangle \Big]. \tag{9}$$

Since

$$\sum_{i=1}^{n} \left(\nabla f_i(x^t) - \nabla f(x^t)\right) = 0,$$

the inner-product terms vanish after summation. Applying Assumption 2 then yields

$$\left\|g^t - \nabla f(x^t)\right\|^2 \leq \omega \left(\left\|\nabla f(x^t)\right\|^2 + \zeta^2 \right). \tag{10}$$

□

D Proof of Theorems

D.1 Proof of Theorem 1

Proof. Under Assumption 1, recall Eq. (3). Since the learning rate satisfies $\eta < 1/L$ and the ACTION condition holds, we have:

$$f(x^{t+1}) \leq f(x^t) - \left(\eta - \frac{L\eta^2}{2}\right) \left\|\nabla f(x^t)\right\|^2 + \frac{L\eta^2}{2} \left\|g^t - \nabla f(x^t)\right\|^2. \tag{11}$$

Applying Lemma 2 under Assumption 2, we obtain

$$f(x^{t+1}) \leq f(x^t) - \left(\eta - \frac{L\eta^2}{2}\right) \left\|\nabla f(x^t)\right\|^2 + \frac{L\eta^2\omega}{2} \left(\left\|\nabla f(x^t)\right\|^2 + \zeta^2 \right). \tag{12}$$

Rearranging the terms gives

$$f(x^{t+1}) \leq f(x^t) - \left(\eta - \frac{1+\omega}{2} L\eta^2\right) \left\|\nabla f(x^t)\right\|^2 + \frac{\omega}{2} L\eta^2\zeta^2. \tag{13}$$

Subtracting $f^{\inf}$ from both sides and defining $\delta^t := f(x^t) - f^{\inf}$, we obtain

$$\delta^{t+1} \leq \delta^t - \left(\eta - \frac{1+\omega}{2} L\eta^2\right) \left\|\nabla f(x^t)\right\|^2 + \frac{\omega}{2} L\eta^2\zeta^2. \tag{14}$$

Summing Eq. (14) over $t = 0, \ldots, T-1$ yields

$$0 \leq \delta^T \leq \delta^0 - \left(\eta - \frac{1+\omega}{2} L\eta^2\right) \sum_{t=0}^{T-1} \left\|\nabla f(x^t)\right\|^2 + \frac{\omega}{2} LT\eta^2\zeta^2. \tag{15}$$

Since the learning rate satisfies $\eta < \frac{1}{(1+\omega)L}$, we have $\eta - \frac{1+\omega}{2} L\eta^2 > \frac{\eta}{2}$. Therefore,

$$\frac{\eta}{2} \sum_{t=0}^{T-1} \left\|\nabla f(x^t)\right\|^2 \leq \delta^0 + \frac{\omega}{2} LT\eta^2\zeta^2. \tag{16}$$

Multiplying both sides by $\frac{2}{\eta T}$ and rearranging yields the convergence result

$$\frac{1}{T} \sum_{t=0}^{T-1} \left\|\nabla f(x^t)\right\|^2 \leq \frac{2(f(x^0) - f^{\inf})}{\eta T} + \omega L\eta\zeta^2. \tag{17}$$

It remains to note that the left-hand side can be equivalently written as $\mathbb{E}\left[\left\|\nabla f(\hat{x}^T)\right\|^2\right]$, where $\hat{x}^T$ is uniformly sampled from $x^0, x^1, \ldots, x^{T-1}$. □

D.2 Proof of Theorem 2

Proof. At iteration $t = 0$, each node compresses its local gradient using the zero reference vector $g^{-1} = 0$. Applying Lemma 1 gives

$$f(x^1) \le f(x^0) - \frac{\eta}{2}\left\|\nabla f(x^0)\right\|^2 - \left(\frac{1}{2\eta} - \frac{L}{2}\right)\left\|x^1 - x^0\right\|^2 + \frac{\eta}{2}\left\|g^0 - \nabla f(x^0)\right\|^2 \tag{18}$$

$$\le f(x^0) - \frac{\eta}{2}\left\|\nabla f(x^0)\right\|^2 + \frac{\eta}{2}\left\|g^0 - \nabla f(x^0)\right\|^2. \tag{19}$$

Then, we have

$$\frac{\eta}{2}\left\|\nabla f(x^0)\right\|^2 \le f(x^0) - f(x^1) + \frac{\eta}{2}\left\|g^0 - \nabla f(x^0)\right\|^2 \tag{20}$$

We next consider iterations $t \ge 1$. Under Assumption 1, recalling Eq. (3), we have

$$f(x^{t+1}) \le f(x^t) - \left(\eta - \frac{L\eta^2}{2}\right)\left\|\nabla f(x^t)\right\|^2 + \frac{L\eta^2}{2}\left\|g^t - \nabla f(x^t)\right\|^2 - (\eta - L\eta^2)\left\langle \nabla f(x^t), g^t - \nabla f(x^t)\right\rangle. \tag{21}$$

For $t \ge 1$, we first derive a lower bound on $\left\langle \nabla f(x^{t-1}), g^t - \nabla f(x^t)\right\rangle$. Define

$$S_t^+ = \left\{j \mid j \in [d], [g^{t-1}]_j[\nabla f(x^{t-1})]_j \ge 0\right\},\ S_t^- = \left\{j \mid j \in [d], [g^{t-1}]_j[\nabla f(x^{t-1})]_j < 0\right\}. \tag{22}$$

Under Assumption 2, every $j \in S_t^-$ satisfies $|[\nabla f(x^{t-1})]_j| \le \zeta$: otherwise, all local gradients at that coordinate would have the same sign as the global gradient, and compression that preserves local signs or maps coordinates to zero could not reverse the aggregate sign. Meanwhile, the coordinatewise implementation of L-ACTION ensures $[g^{t-1}]_j[g^t - \nabla f(x^t)]_j \ge 0$ for every coordinate j. For $j \in S_t^+$ with $[g^{t-1}]_j \ne 0$, the previous global gradient is either zero or has the same sign as $[g^{t-1}]_j$, implying $[\nabla f(x^{t-1})]_j[g^t - \nabla f(x^t)]_j \ge 0$. If $[g^{t-1}]_j = 0$, each node transmits its local gradient at that coordinate without compression, so $[g^t - \nabla f(x^t)]_j = 0$ and the same inequality holds. Consequently,

$$\left\langle \nabla f(x^{t-1}), g^t - \nabla f(x^t)\right\rangle$$

$$= \sum_{j\in S_t^+}[\nabla f(x^{t-1})]_j[g^t - \nabla f(x^t)]_j + \sum_{j\in S_t^-}[\nabla f(x^{t-1})]_j[g^t - \nabla f(x^t)]_j \tag{23}$$

$$\ge -\sum_{j\in S_t^-}\left|[\nabla f(x^{t-1})]_j\right|\left|[g^t - \nabla f(x^t)]_j\right| \tag{24}$$

$$\ge -\frac{1}{2L\eta}\sum_{j\in S_t^-}[\nabla f(x^{t-1})]_j^2 - \frac{L\eta}{2}\sum_{j\in S_t^-}[g^t - \nabla f(x^t)]_j^2, \tag{25}$$

where the last inequality is Young' inequality.

Since

$$R_t = \frac{\sum_{j\in S_t^-}[\nabla f(x^t)]_j^2}{\|\nabla f(x^t)\|^2}, \tag{26}$$

The range of R_t is $[0, 1]$, since $r = \sup_t R_t$. Then

$$\begin{aligned}
\left\langle \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle &\geq -\frac{1}{2L\eta} \sum_{j \in S_t^-} [\nabla f(x^{t-1})]_j^2 - \frac{L\eta}{2} \sum_{j \in S_t^-} [g^t - \nabla f(x^t)]_j^2 && (27)\\
&= -\frac{R_{t-1}}{2L\eta} \sum_{j \in S_t^-} [\nabla f(x^{t-1})]_j^2 - \frac{L\eta}{2} \sum_{j \in S_t^-} [g^t - \nabla f(x^t)]_j^2 && (28)\\
&\geq -\frac{r}{2L\eta} \sum_{j \in S_t^-} [\nabla f(x^{t-1})]_j^2 - \frac{L\eta}{2} \sum_{j \in S_t^-} [g^t - \nabla f(x^t)]_j^2 && (29)\\
&\geq -\frac{r}{2L\eta} \|\nabla f(x^{t-1})\|^2 - \frac{L\eta}{2} \left\| g^t - \nabla f(x^t) \right\|^2 && (30)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\left\langle \nabla f(x^t), g^t - \nabla f(x^t) \right\rangle \\
&= \left\langle \nabla f(x^t) - \nabla f(x^{t-1}) + \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle && (31)\\
&\geq \left\langle \nabla f(x^t) - \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle - \frac{r}{2L\eta} \|\nabla f(x^{t-1})\|^2 - \frac{L\eta}{2} \left\| g^t - \nabla f(x^t) \right\|^2 && (32)
\end{aligned}$$

Substituting Eq. (32) into Eq. (21), then

$$\begin{aligned}
f(x^{t+1}) \leq f(x^t) &- \left(\eta - \frac{L\eta^2}{2} \right) \|\nabla f(x^t)\|^2 + \frac{L\eta^2}{2} \|g^t - \nabla f(x^t)\|^2 \\
&- (\eta - L\eta^2) \langle \nabla f(x^t) - \nabla f(x^{t-1}), g^t - \nabla f(x^t) \rangle \\
&+ \frac{(1 - L\eta) r}{2L} \left\| \nabla f(x^{t-1}) \right\|^2 + \left(\frac{L\eta^2}{2} - \frac{L^2\eta^3}{2} \right) \|g^t - \nabla f(x^t)\|^2 && (33)
\end{aligned}$$

For the inner-product term on the right-hand side, by the Cauchy–Schwarz inequality and Assumption 1, we have

$$\begin{aligned}
&- (\eta - L\eta^2) \left\langle \nabla f(x^t) - \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle \\
&\quad \leq (\eta - L\eta^2) \left\| \nabla f(x^t) - \nabla f(x^{t-1}) \right\| \left\| g^t - \nabla f(x^t) \right\| \\
&\quad \leq (\eta - L\eta^2) L \left\| x^t - x^{t-1} \right\| \left\| g^t - \nabla f(x^t) \right\|. && (34)
\end{aligned}$$

Using the parameter update rule and Young's inequality, we further obtain

$$\begin{aligned}
&- (\eta - L\eta^2) \left\langle \nabla f(x^t) - \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle \\
&\quad \leq (\eta - L\eta^2) L\eta \left\| g^{t-1} \right\| \left\| g^t - \nabla f(x^t) \right\| \\
&\quad \leq (\eta - L\eta^2) L\eta \left(\frac{1}{2} \left\| g^{t-1} \right\|^2 + \frac{1}{2} \left\| g^t - \nabla f(x^t) \right\|^2 \right). && (35)
\end{aligned}$$

Furthermore, using

$$g^{t-1} = \nabla f(x^{t-1}) + \big(g^{t-1} - \nabla f(x^{t-1}) \big), \tag{36}$$

we have

$$\begin{aligned}
&- (\eta - L\eta^2) \left\langle \nabla f(x^t) - \nabla f(x^{t-1}), g^t - \nabla f(x^t) \right\rangle \\
&\quad \leq (\eta - L\eta^2) L\eta \left(\left\| \nabla f(x^{t-1}) \right\|^2 + \left\| g^{t-1} - \nabla f(x^{t-1}) \right\|^2 + \frac{1}{2} \left\| g^t - \nabla f(x^t) \right\|^2 \right). && (37)
\end{aligned}$$

Substituting Eq. (37) into Eq. (33) and collecting terms yields

$$\begin{aligned} f(x^{t+1}) \leq f(x^t) &- \left(\eta - \frac{L\eta^2}{2}\right)\|\nabla f(x^t)\|^2 + \frac{L\eta^2}{2}\|g^t - \nabla f(x^t)\|^2 \\ &+ (L\eta^2 - L^2\eta^3)\|\nabla f(x^{t-1})\|^2 + (L\eta^2 - L^2\eta^3)\|\nabla g^{t-1} - f(x^{t-1})\|^2 \\ &+ \left(\frac{L\eta^2}{2} - \frac{L^2\eta^3}{2}\right)\|g^t - \nabla f(x^t)\|^2 \\ &+ \frac{(1-L\eta)r}{2L}\left\|\nabla f(x^{t-1})\right\|^2 + \left(\frac{L\eta^2}{2} - \frac{L^2\eta^3}{2}\right)\|g^t - \nabla f(x^t)\|^2 \qquad (38) \\ = f(x^t) &- \left(\eta - \frac{L\eta^2}{2}\right)\|\nabla f(x^t)\|^2 + \left(\frac{3L\eta^2}{2} - L^2\eta^3\right)\|g^t - \nabla f(x^t)\|^2 \\ &+ \left(L\eta^2 - L^2\eta^3 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^{t-1})\|^2 + (L\eta^2 - L^2\eta^3)\|\nabla g^{t-1} - f(x^{t-1})\|^2 \qquad (39) \end{aligned}$$

Applying Lemma 2, we obtain

$$\begin{aligned} f(x^{t+1}) \leq f(x^t) &- \left(\eta - \frac{(3\omega+1)L\eta^2}{2}\right)\|\nabla f(x^t)\|^2 \\ &+ \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^{t-1})\|^2 + \frac{5L\omega\eta^2}{2}\zeta^2 \qquad (40) \end{aligned}$$

This inequality also follows by dropping the nonpositive terms.

Subtracting $f^{\inf}$ from both sides and defining $\delta^t := f(x^t) - f^{\inf}$, we obtain

$$\begin{aligned} \delta^{t+1} \leq \delta^t &- \left(\eta - \frac{(3\omega+1)L\eta^2}{2}\right)\|\nabla f(x^t)\|^2 \\ &+ \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^{t-1})\|^2 + \frac{5L\omega\eta^2}{2}\zeta^2 \qquad (41) \end{aligned}$$

Summing Eq. (41) over $t = 1, \ldots, T-1$ and using $\delta^T \geq 0$, we obtain

$$\begin{aligned} 0 \leq \delta^1 &- \left(\eta - \frac{(3\omega+1)L\eta^2}{2}\right)\sum_{t=1}^{T-1}\|\nabla f(x^t)\|^2 + \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\sum_{t=1}^{T-2}\|\nabla f(x^t)\|^2 \\ &+ \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^0)\|^2 + \frac{5}{2}L\omega\eta^2\zeta^2(T-1) \qquad (42) \end{aligned}$$

Combining the two summation terms gives

$$\begin{aligned} &\left(\frac{2+r}{2}\eta - \frac{(5\omega+3)L\eta^2}{2} - \frac{r}{2L}\right)\sum_{t=1}^{T-1}\|\nabla f(x^t)\|^2 \\ &\leq \delta^1 + \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^0)\|^2 + \frac{5}{2}L\omega\eta^2\zeta^2(T-1) \qquad (43) \end{aligned}$$

Since $r < \frac{20\omega+10-\sqrt{(20\omega+10)^2-4}}{2}$ and the learning rate satisfices $\eta < \frac{(1+r)+\sqrt{(1+r)^2-4(5\omega+3)r}}{(10\omega+6)L}$ Therefore,

$$\frac{\eta}{2}\sum_{t=1}^{T-1}\left\|\nabla f(x^t)\right\|^2 \leq \delta^1 + \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^0)\|^2 + \frac{5}{2}L\omega\eta^2\zeta^2(T-1) \qquad (44)$$

Combine with Eq. (20), then

$$\begin{aligned} \frac{\eta}{2}\sum_{t=0}^{T-1}\left\|\nabla f(x^t)\right\|^2 \leq \delta^0 &+ \left((1+\omega)L\eta^2 + \frac{(1-L\eta)r}{2L}\right)\|\nabla f(x^0)\|^2 \\ &+ \frac{5}{2}L\omega\eta^2\zeta^2(T-1) + \frac{\eta}{2}\left\|g^0 - \nabla f(x^0)\right\|^2 \qquad (45) \end{aligned}$$

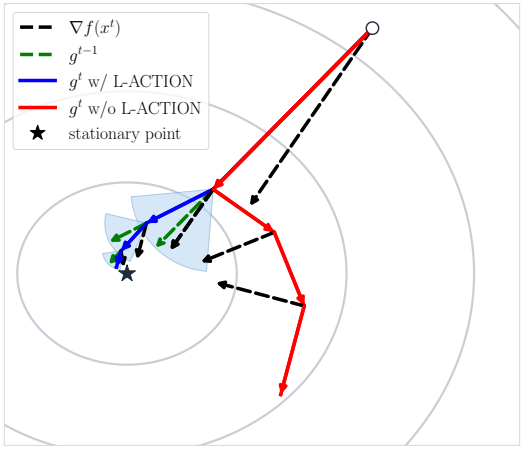

Figure 6: Optimization trajectories with and without L-ACTION.

Multiplying both sides by $\frac{2}{\eta T}$ gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\left\|\nabla f(x^t)\right\|^2 \le \frac{2(f(x^0)-f^{\inf})}{\eta T} + 5\omega L\eta\zeta^2 + \frac{2(1+\omega)L\eta\|f(x^0)\|^2}{T} + \frac{(1-L\eta)r\|f(x^0)\|^2}{L\eta T} + \frac{\left\|g^0-\nabla f(x^0)\right\|^2}{T} \quad (46)$$

It remains to note that the left-hand side can be equivalently written as $\mathbb{E}\left[\left\|\nabla f(\hat{x}^T)\right\|^2\right]$, where $\hat{x}^T$ is uniformly sampled from $x^0, x^1, \ldots, x^{T-1}$. □

E Optimization Trajectories of L-ACTION

Fig. 6 illustrates L-ACTION, which constrains compression errors relative to the previous aggregated compressed gradient g^{t-1}. Since $g^{-1}=0$, the first iteration has no directional restriction, as shown by the initial red step. Subsequent iterations use the previous compressed gradient as the reference, indicated by the purple dashed arrows. The blue trajectory approaches the stationary point, while the red trajectory illustrates possible failure without the constraints.

F Analysis of r

We investigate r using synthetic gradients. We sample a reference vector $v \sim \mathcal{N}(0, I_d)$ and independent perturbations $u_i \sim \mathcal{N}(0, I_d)$ for $i \in [n]$. Writing $\bar{u} = \frac{1}{n}\sum_{i=1}^n u_i$, we construct the local gradients as

$$g_i = v + \zeta\frac{u_i - \bar{u}}{\max_{\ell\in[n]}|u_\ell - \bar{u}|}. \quad (47)$$

The global gradient $g = \frac{1}{n}\sum_{i=1}^n g_i$. We compress each local gradient and compute the aggregate $\hat{g} = \frac{1}{n}\sum_{i=1}^n \mathcal{C}(g_i)$.

We evaluate quantization with $b \in \{2,4,8\}$ bits and top-k sparsification with $k \in \{100, 1000, 10000\}$. We consider all combinations of heterogeneity levels $\zeta \in \{10, 100, 1000\}$, node counts $n \in \{10, 100, 1000\}$, and dimensions $d \in \{10^5, 10^6, 10^7\}$.

Fig. 7 shows that r is zero or close to zero in most quantization configurations and remains below 0.01 in most evaluated settings. For $\zeta \in \{10, 100\}$, all values are below 0.007 or equal to it, while 8bit quantization keeps r below 0.0005 across all tested configurations. Larger values occur primarily under strong heterogeneity and low-bit quantization, reaching 0.243 when $\zeta = 1000$, $d =$

10^5, $n = 10$, and $b = 2$. Under strong heterogeneity, increasing the number of clients consistently reduces r, while larger gradient dimensions generally yield smaller values.

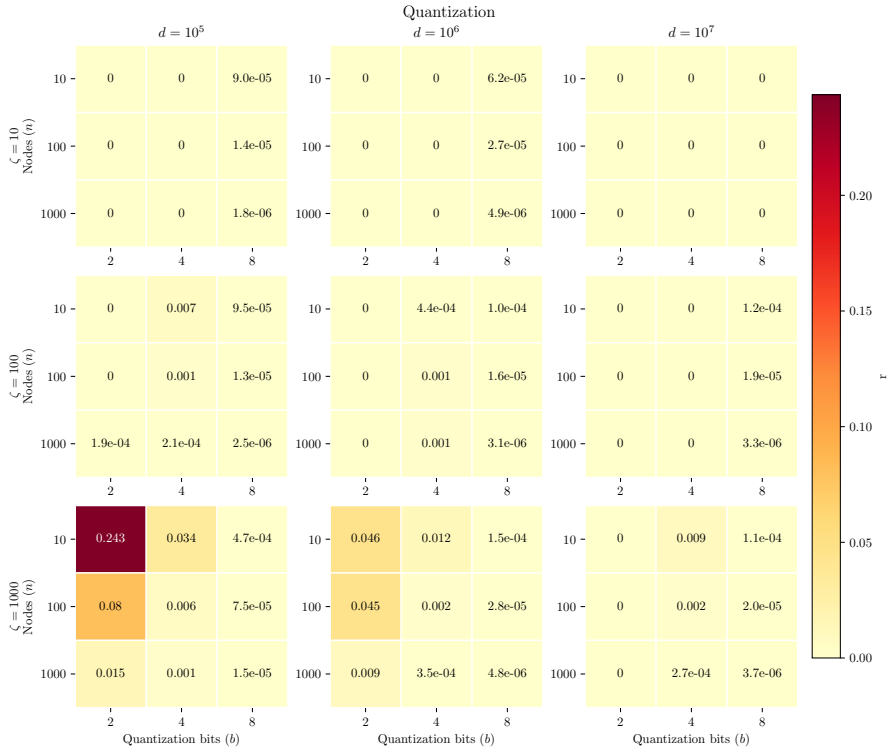

Figure 7: r under quantization. Rows correspond to different heterogeneity levels ζ, and columns correspond to gradient dimensions d. Each heatmap reports reversal rates across quantization bit-widths b and client counts n.

Fig. 8 shows that r is zero in most evaluated configurations under top-k sparsification. All reported values are zero when $\zeta \in \{10, 100\}$ or $d = 10^7$. Nonzero values occur only under strong heterogeneity ($\zeta = 1000$) at smaller dimensions. For $d = 10^5$, r ranges from 0.02 to 0.218, with the maximum attained at $n = 10$ and $k = 10^4$. For $d = 10^6$, all values remain at or below 0.032. These results indicate that larger gradient dimensions are associated with smaller r in the tested settings, whereas the effects of k and n are not consistently monotonic.

Overall, r is zero or negligible in most evaluated settings. Appreciable values arise only under strong data heterogeneity combined with low model dimensionality and aggressive compression.

G Details of Quantization and Sparsification

G.1 Rounding Quantization

Given an input vector $x \in \mathbb{R}^d$ and an integer bit width $b \geq 2$, define

$$K = 2^{b-1} - 1, \qquad a = \|x\|_\infty, \qquad s = \begin{cases} a/K, & a > 0, \\ 1, & a = 0. \end{cases} \tag{48}$$

Here, s is the quantization scale; the convention $s = 1$ for an all-zero input avoids division by zero and yields exact reconstruction.

For encoding, we compute the unsigned integer codes coordinatewise as

$$q_i = \mathrm{round}(x_i/s) + K, \qquad i \in [d], \tag{49}$$

where round denotes nearest-integer rounding. Since $x_i/s \in [-K, K]$, each code satisfies $q_i \in \{0, \ldots, 2K\}$ and can be stored using b bits. The encoder transmits the codes q together with the

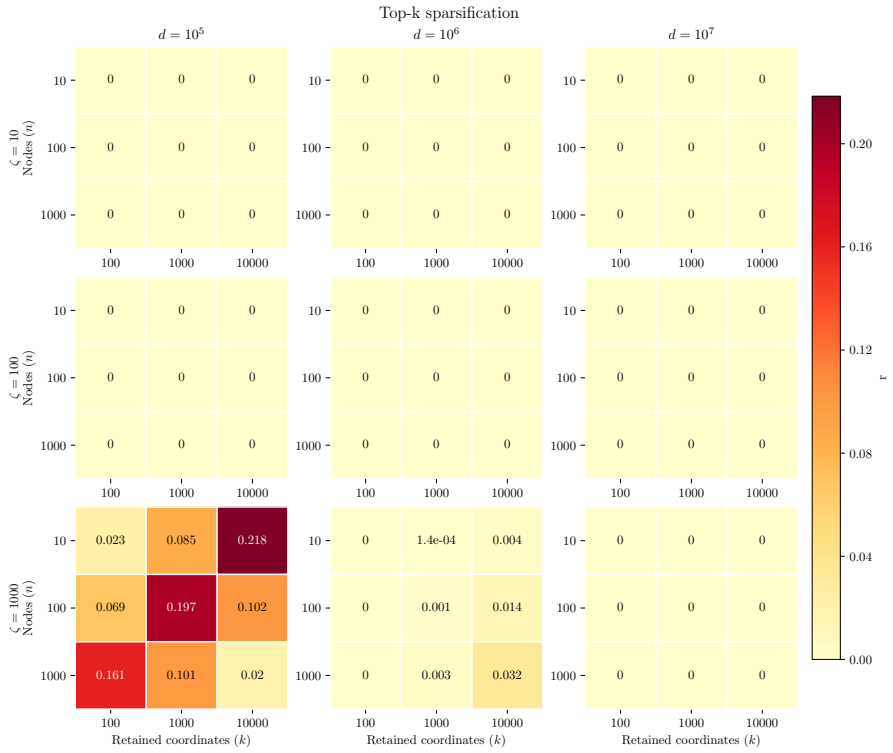

Figure 8: r under sparsification. Rows correspond to different heterogeneity levels ζ, and columns correspond to gradient dimensions d. Each heatmap reports reversal rates across sparsification k and client counts n.

scale s. The decoder reconstructs the vector as

$$\hat{x}_i = Q(x)_i = (q_i - K)s = s \cdot \mathrm{round}(x_i/s), \qquad i \in [d]. \tag{50}$$

G.2 Top-k Sparsification

For sparsification, the encoding and decoding procedures are described as follows. Let $x \in \mathbb{R}^d$ denote the vector to be sparsified, and let k denote the number of retained elements. Define an index set $\mathcal{I} \subseteq \{1, 2, \cdots, d\}$ satisfying $|\mathcal{I}| = k$. The sparsification operator $\mathcal{S} : \mathbb{R}^d \to \mathbb{R}^d$ is defined as

$$\mathcal{S}(x) = \begin{cases} x_i, & i \in \mathcal{I}, \\ 0, & i \notin \mathcal{I}, \end{cases} \tag{51}$$

where $\mathcal{I}$ denotes the set of indices corresponding to the retained elements. The receiving node can then reconstruct the sparse vector from the received values and their corresponding indices, $\hat{x} = S(x)$, where $\hat{x}$ denotes the reconstructed sparse vector, whose error arises from Eq. (51).

H Verification of the Directional Constraint

H.1 DC-Q

We show that DC-Q satisfies the directional constraint. For any coordinate i, if $y_i > 0$, we round v_i upward, so that $\text{dc-round}(v_i, y_i) - v_i \geq 0$. If $y_i < 0$, we round v_i downward, so that $\text{dc-round}(v_i, y_i) - v_i \leq 0$. In both cases,

$$y_i\big(\text{dc-round}(v_i, y_i) - v_i\big) \geq 0. \tag{52}$$

When $y_i = 0$, conventional rounding is used to minimize the quantization error, and the above product equals zero regardless of the rounding outcome. Thus, the inequality holds for every coordinate i, DC-Q is a direction-constrained compressor.

H.2 DC-S

We show that DC-S satisfies the directional constraint. The selected elements are retained exactly and therefore contribute zero to the directional constraint. For unselected indices in S^{safe}, the corresponding elements are reconstructed as zero. By the definition of S^{safe}, their reconstruction errors satisfy

$$y_i(0 - x_i) = -x_i y_i \geq 0. \tag{53}$$

For unselected indices in S^{unsafe}, reconstruction is determined by the sign of y_i. If $y_i \geq 0$, the corresponding element is reconstructed as M. Since $M \geq x_i$, we have

$$y_i(M - x_i) \geq 0. \tag{54}$$

If $y_i < 0$, the corresponding element is reconstructed as m. Since $m \leq x_i$, we similarly obtain

$$y_i(m - x_i) \geq 0. \tag{55}$$

Thus, for every coordinate, the product of y_i and the reconstruction error is nonnegative, DC-S is a direction-constrained compressor.

I Experimental Details

I.1 Model Details

For the MNIST and Fashion-MNIST datasets, the CNN model consists of three convolutional blocks, which progressively map the input channels to 16, 32, and 64 channels, respectively. All convolutional layers use 3×3 kernels with a stride and padding of 1. Each convolutional layer is followed by a batch normalization (BN) layer, a ReLU activation function, and a max-pooling layer with a kernel size of 2. After feature extraction through the three convolutional blocks, the feature maps have 64 channels and a spatial size of 3×3. Average pooling is then applied to obtain a 64-dimensional feature vector. The feature vector is flattened and passed through a linear layer that maps it to a 10-dimensional output. The model contains 24,170 trainable parameters.

For CIFAR10, the VGG16 architecture is modified to accommodate the input resolution and classification task. The network consists of five convolutional blocks with 13 convolutional layers in total, followed by three fully connected layers. The five convolutional blocks contain 2, 2, 3, 3, and 3 convolutional layers, respectively, with 64, 128, 256, 512, and 512 output channels. All convolutional layers use 3×3 kernels with a stride and padding of 1, and each convolutional layer is followed by a BN layer and a ReLU activation function. Max-pooling layers with a kernel size and stride of 2 are inserted between consecutive convolutional blocks for downsampling. The convolutional feature extractor produces a 512-dimensional feature vector, which is then fed into three fully connected layers with output dimensions of 4096, 4096, and 10, respectively. Each fully connected layer is followed by a ReLU activation function and a Dropout layer with a dropout rate of 0.5. The model contains 33,646,666 trainable parameters.

For CIFAR100, the ResNet18 architecture is modified accordingly. The model consists of an initial convolutional layer, four convolutional stages, and a fully connected classification layer. The initial convolutional layer uses a 3×3 kernel with a stride and padding of 1 and maps the input image to 64 channels. The four convolutional stages have 64, 128, 256, and 512 output channels, respectively. Each stage contains two residual units, with each unit consisting of two 3×3 convolutional layers and a residual connection. Except for the first stage, the first residual unit of each stage performs downsampling using a stride-2 convolution, while a 1×1 convolution is used to match the feature dimensions in the residual connection. All convolutional layers are followed by BN and ReLU. An adaptive average pooling layer is applied after the final convolutional stage to obtain a 512-dimensional feature vector, which is then mapped to a 100-dimensional output through the fully connected layer. The model contains 11,220,132 trainable parameters.

For AG_NEWS, a bidirectional LSTM is employed for text classification. The model consists of a word embedding layer, a bidirectional LSTM layer, and a fully connected classification layer. The

embedding layer maps the input text sequences to 128-dimensional word embeddings, with a vocabulary size of 30,000. The resulting embeddings are fed into the bidirectional LSTM for sequential feature extraction, with a hidden dimension of 128. The bidirectional structure captures contextual information from both forward and backward directions. The final LSTM hidden states are used as the text representation and passed through a fully connected layer to obtain a 4-dimensional output for AG_NEWS classification. The model contains 4,104,708 trainable parameters.

For IMDB, a Transformer model is employed for text classification. The model consists of a word embedding layer, a positional encoding layer, a Transformer encoder, and a fully connected classification layer. The embedding layer maps the input text sequences to 128-dimensional word embeddings, with a vocabulary size of 30,000. Positional encoding is applied to incorporate positional information into the input sequence. The resulting representations are fed into a Transformer encoder with two layers and four attention heads. The hidden dimension of the feed-forward network is 512, the dropout rate is 0.1, and the maximum input sequence length is 512. Average pooling is applied to the encoder output to obtain a text representation, which is then mapped to a 2-dimensional output through the fully connected layer. The model contains 4,104,708 trainable parameters.

I.2 HYPERPARAMETERS

For the compression hyperparameters, we evaluate the quantization methods using different quantization bit-widths. Specifically, 2bit and 4bit quantization are used for the CNN model, while 4bit, 6bit, and 8bit quantization are used for the other models. The sparsification parameters are selected according to the model size, with the detailed configurations listed in Table 4.

Each client computes its local gradient using all local training samples at every communication round, no mini-batch sampling is used.

Table 4: Experimental models and algorithm parameter settings

Models	**Quantization Bits**	**Sparsification k**	**Number of Parameters**
CNN	2, 4	1000, 2000	24,170
VGG16	4, 6, 8	2M, 3M, 4M	33,646,666
ResNet18	4, 6, 8	0.5M, 1M, 1.5M	11,220,132
LSTM	4, 6, 8	0.2M, 0.3M, 0.4M	4,104,708
Transformer	4, 6, 8	0.2M, 0.3M, 0.4M	4,104,708

For the training hyperparameters, we use 10 clients for distributed training and partition the datasets according to a Dirichlet distribution with concentration parameter 5, representing a mildly heterogeneous data distribution. The learning rate is set to 3 for LSTM on the AG_NEWS dataset and to 0.1 for all other experiments. All experiments are conducted for 1,000 communication rounds. We use full-batch gradient descent, which introduces no randomness through mini-batch sampling, and fix the random seed to 42 in all experiments. For clearer visualization, the curves in the experimental results are obtained by averaging the results over every 10 communication rounds.

J EXPERIMENTAL RESULTS

J.1 VALIDATION OF ACTION

We further validate the effectiveness of ACTION on Fashion-MNIST using a CNN model. Fig. 9 and Fig. 10 show the comparison results for quantization and sparsification based gradient compression methods, respectively.

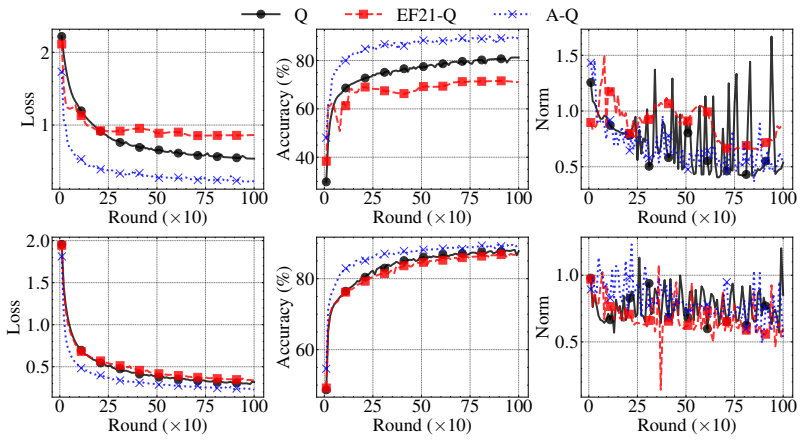

Figure 9: Comparison of CNN with different quantization gradient compression methods on Fashion-MNIST

For quantization (Fig. 9), the first and second rows correspond to 2bit and 4bit settings, respectively. Under 2bit quantization, A-Q reduces the training loss faster and achieves a lower final loss and higher test accuracy than Q and EF21-Q. Under 4bit quantization, A-Q retains an advantage in training loss and early-stage test accuracy, while the final accuracy differences are smaller. The gradient norms fluctuate throughout training. Although A-Q exhibits fewer large late-stage spikes than Q in the 2bit setting, it does not consistently attain the smallest gradient norm across both settings.

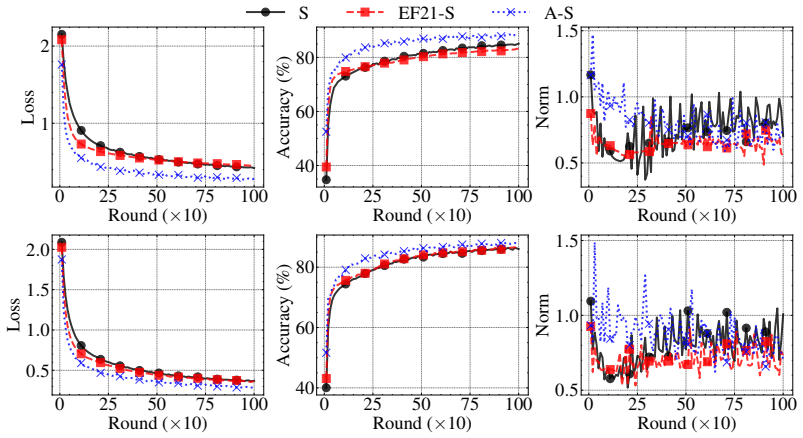

Figure 10: Comparison of CNN with different sparsification gradient compression methods on Fashion-MNIST

For sparsification (Fig. 10), the first and second rows correspond to $k = 1000$ and $k = 2000$, respectively. A-S achieves lower training loss and higher test accuracy than S and EF21-S, with a more pronounced advantage at $k = 1000$. Increasing the budget to $k = 2000$ narrows the performance differences, particularly in final test accuracy. The gradient-norm curves remain noisy, and A-S does not show a consistent advantage over EF21-S on this metric. Overall, these results support the benefits of ACTION for training loss and test accuracy in the evaluated settings, particularly under more aggressive compression.

J.2 Performance of L-ACTION

We further evaluate L-ACTION on different models and datasets. Fig. 11, Fig. 12, and Fig. 13 present quantization results for VGG16 on CIFAR10, Transformer on IMDB, and LSTM on

AG_NEWS, respectively. The three columns correspond to 4bit, 6bit, and 8bit quantization, while the two rows show training loss and test accuracy.

For VGG16 (Fig. 11), LA-Q reduces training loss and improves test accuracy faster during the early stages of 4bit training. However, it exhibits a pronounced loss spike and accuracy drop later in training, ending with lower test accuracy than Q. Under 6bit quantization, LA-Q achieves faster early progress and slightly higher final test accuracy. At 8 bits, the three methods attain similar final losses and accuracies. One possible explanation for the 4bit instability is that lagged reference signs become less reliable as gradient coordinates approach or cross zero.

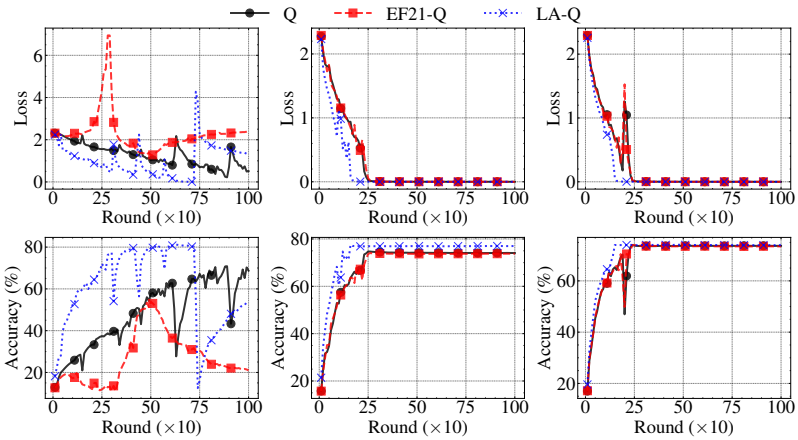

Figure 11: Comparison of VGG16 with different quantization gradient compression methods on CIFAR10

For Transformer (Fig. 12), LA-Q achieves lower final training loss and higher final test accuracy than Q and EF21-Q at all three bit-widths. Its loss continues to decrease while the baselines improve more slowly and exhibit recurrent fluctuations. Under 4bit quantization, EF21-Q also shows a substantial late-stage deterioration. These results demonstrate a clear advantage for LA-Q on this task within the evaluated training budget.

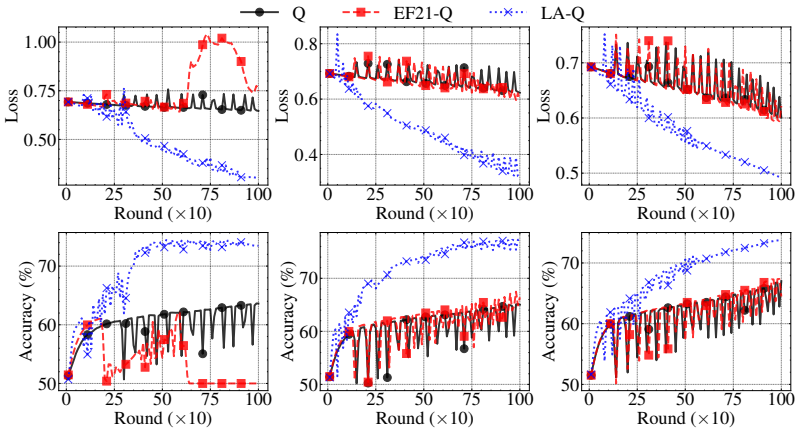

Figure 12: Comparison of Transformer with different quantization gradient compression methods on IMDB

For LSTM (Fig. 13), LA-Q again achieves faster early progress at 4bit, but subsequently experiences a sharp loss spike and accuracy drop, followed by only partial recovery. Consequently, its early advantage does not translate into better final performance. At 6bit, LA-Q reaches a lower training loss and higher test accuracy than the two baselines. At 8bit, it retains a lower training loss, while the final test accuracies are similar.

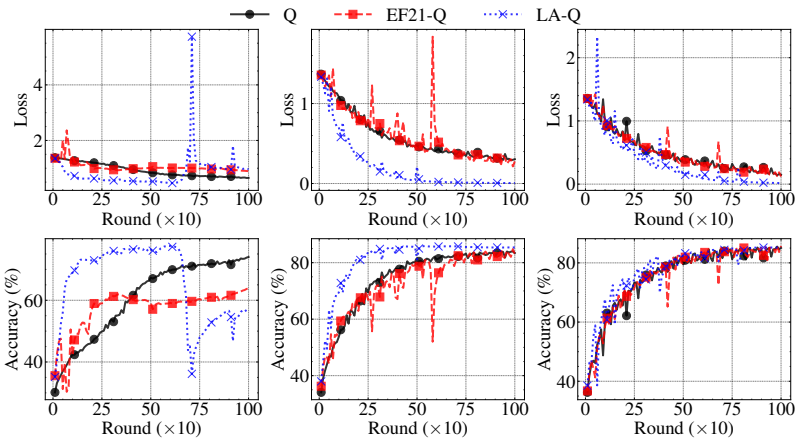

Figure 13: Comparison of LSTM with different quantization gradient compression methods on AG_NEWS

Fig. 14, Fig. 15, and Fig. 16 present sparsification results for VGG16 on CIFAR10, ResNet18 on CIFAR100, and LSTM on AG_NEWS, respectively. The columns correspond to increasing sparsification budgets, while the rows show training loss and test accuracy.

For VGG16 (Fig. 14), S, EF21-S, and LA-S reach similar final training losses and test accuracies across all three budgets. Differences mainly occur during the initial training phase. LA-S exhibits comparable overall performance, without a consistent advantage in training speed or final accuracy.

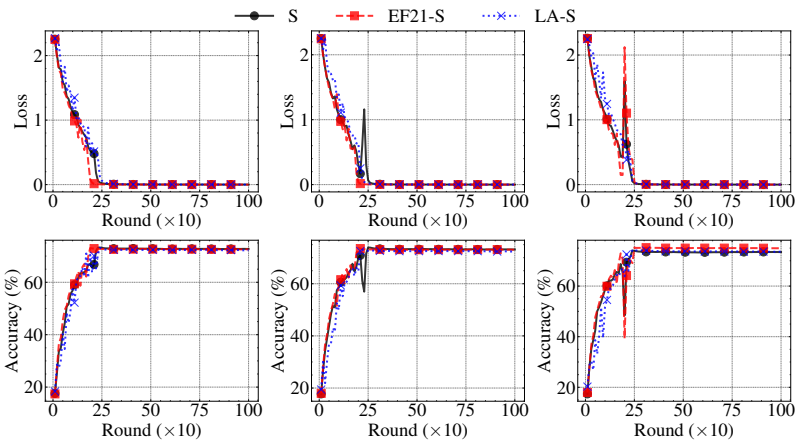

Figure 14: Comparison of VGG16 with different sparsification gradient compression methods on CIFAR10

For ResNet18 (Fig. 15), LA-S generally improves more slowly during early training than S and EF21-S, but reaches final test accuracies comparable to S and higher than EF21-S. Although EF21-S reduces training loss faster over substantial portions of training, this does not translate into higher final test accuracy. Increasing the sparsification budget allows all three methods to reach lower training losses within the same training horizon. LA-S also exhibits occasional loss spikes and accuracy drops, particularly at the intermediate and largest budgets.

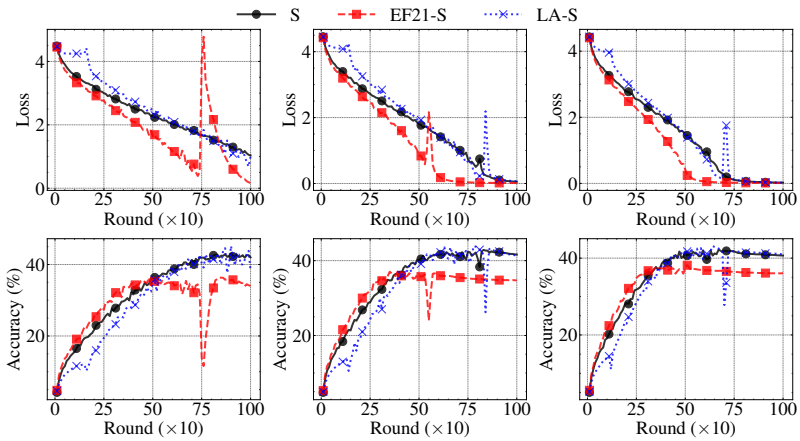

Figure 15: Comparison of ResNet18 with different sparsification gradient compression methods on CIFAR100

For LSTM (Fig. 16), the three methods follow similar training-loss and test-accuracy trajectories across the evaluated budgets. LA-S exhibits an early loss spike in each setting, after which its performance closely tracks the baselines. The final losses and accuracies are similar, indicating no consistent advantage for LA-S on this task.

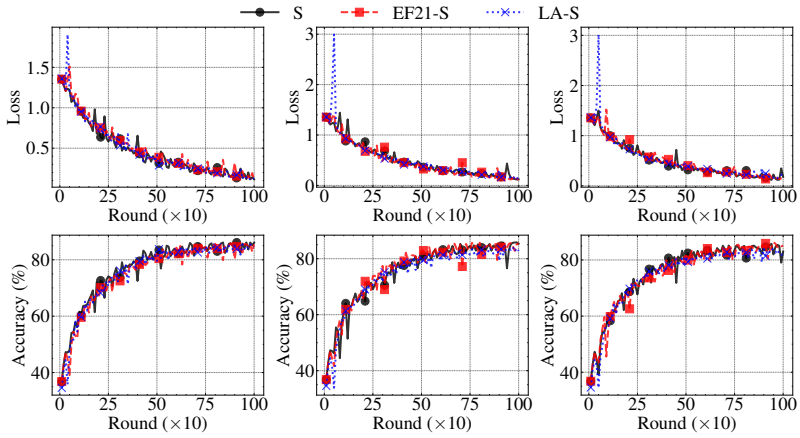

Figure 16: Comparison of LSTM with different sparsification gradient compression methods on AG_NEWS

J.3 Communication Overhead Comparison

We further compare the communication efficiency of different compression methods in terms of the number of communication rounds required to achieve target accuracies. Tables 5 and 6 report the results for quantization-based gradient compression on VGG16 and LSTM models, respectively. Tables 7, 8, 9, and 10 present the corresponding results for sparsification-based compression on different models and datasets.

For quantization, LA-Q requires fewer communication rounds than Q and EF21-Q for every target accuracy attained by at least one method in Tables 5 and 6.

Table 5: Comparison of communication rounds required to achieve target accuracy for VGG16 on CIFAR10 with different quantization

Target Acc.	Quant. Bits	Communication Round		
		Q	EF21-Q	LA-Q
60%	4bit	498	–	**128**
	6bit	111	111	**57**
	8bit	90	89	**58**
70%	4bit	798	–	**225**
	6bit	161	224	**85**
	8bit	177	174	**126**

The results on VGG16 (Table 5) demonstrate the communication efficiency of LA-Q under various quantization precisions. When targeting 70% accuracy, LA-Q requires only 225 communication rounds with 4bit quantization, achieving a substantial reduction compared with Q (798 rounds), while EF21-Q fails to attain the target accuracy within the same budget. Similar improvements are observed for 6bit and 8bit quantization, where LA-Q reaches the target accuracy in 85 and 126 rounds, respectively, consistently outperforming the competing methods. Notably, the gain becomes more significant with lower-bit quantization, indicating that LA-Q is particularly effective in addressing the convergence degradation caused by aggressive quantization. Overall, these results verify that LA-Q achieves a better trade-off between communication efficiency and model accuracy.

Table 6: Comparison of communication rounds required to achieve target accuracy for LSTM on AG_NEWS with different quantization

Target Acc.	Quant. Bits	Communication Round		
		Q	EF21-Q	LA-Q
70%	4bit	563	–	**106**
	6bit	214	204	**88**
	8bit	175	166	**101**
80%	4bit	–	–	–
	6bit	415	464	**175**
	8bit	377	371	**320**

The results on LSTM (Table 6) further verify the communication efficiency of LA-Q under different quantization. For the 70% accuracy target, LA-Q requires only 106 rounds with 4bit quantization, achieving a significant reduction compared with Q (563 rounds), while EF21-Q fails to reach the target within the training budget. Similar improvements are observed with 6-bit and 8-bit quantization, where LA-Q consistently converges faster than both baselines. For the more demanding 80% accuracy target, all methods fail under 4bit quantization, indicating the difficulty of training LSTM models with extremely aggressive quantization. Nevertheless, LA-Q achieves the target accuracy within 175 and 320 rounds for 6bit and 8bit quantization, respectively, substantially reducing the required communication rounds compared with Q and EF21-Q. These results highlight the robustness of LA-Q in maintaining efficient convergence under distribution optimization.

For sparsification, the results depend on the model and target accuracy. It is worth noting that LA-S introduces an additional communication overhead for transmitting the sparsification mask, which effectively reduces the available budget for transmitting update values under the same communication constraint. Despite this extra cost, LA-S still achieves competitive convergence performance compared with the baseline methods, demonstrating its ability to preserve useful update information and maintain efficient training under sparse communication.

Table 7: Comparison of communication rounds required to achieve target accuracy for VGG16 on CIFAR10 with different sparsification

Target Acc.	k	Communication Round		
		S	EF21-S	LA-S
60%	2M	103	**91**	95
	3M	**87**	90	103
	4M	93	**77**	115
70%	2M	210	173	**162**
	3M	196	**160**	201
	4M	174	**144**	178

The results on VGG16 (Table 7) demonstrate that LA-S achieves competitive communication efficiency under different sparsification levels. For the 70% accuracy target, LA-S requires only 162 communication rounds with 2M sparsification, reducing the communication cost compared with S (210 rounds) and EF21-S (173 rounds). Although EF21-S achieves slightly fewer rounds under 3M and 4M sparsification, LA-S maintains comparable convergence performance. For the 60% accuracy target, all methods exhibit similar communication efficiency, suggesting that the advantage of sparsification methods becomes more apparent when a higher accuracy target is required.

Table 8: Comparison of communication rounds required to achieve target accuracy for ResNet18 on CIFAR100 with different sparsification

Target Acc.	k	Communication Round		
		S	EF21-S	LA-S
30%	0.5M	334	**255**	397
	1M	240	**193**	293
	1.5M	215	**164**	247
40%	0.5M	613	–	**593**
	1M	**462**	–	486
	1.5M	**378**	–	394

The results on ResNet18 (Table 8) further demonstrate the robustness of LA-S under sparse communication. For the 30% accuracy target, EF21-S achieves the fastest convergence across different sparsification levels, while LA-S remains competitive with S. However, when targeting a higher accuracy of 40%, EF21-S fails to reach the target under all settings, whereas LA-S successfully achieves the target accuracy with 0.5M, 1M, and 1.5M retained parameters. Compared with S, LA-S obtains comparable communication efficiency while providing more stable convergence under challenging sparsification conditions.

Table 9: Comparison of communication rounds required to achieve target accuracy for LSTM on AG_NEWS with different sparsification

Target Acc.	k	Communication Round		
		S	EF21-S	LA-S
70%	0.2M	**159**	186	178
	0.3M	190	**179**	200
	0.4M	**159**	197	173
80%	0.2M	**319**	368	363
	0.3M	383	**357**	415
	0.4M	377	**367**	418

The results on LSTM (Table 9) further verify the effectiveness of LA-S under sparse communication. For the 70% accuracy target, LA-S achieves competitive convergence performance across different sparsification levels. When targeting 80% accuracy, all methods require more communication rounds due to the increased optimization difficulty. Nevertheless, LA-S successfully reaches the target accuracy under all sparsification settings, demonstrating stable convergence comparable to the baseline methods. These results show that LA-S can effectively preserve optimization performance under sparse updates, particularly for LSTM models where training convergence is more sensitive to communication compression.

Table 10: Comparison of communication rounds required to achieve target accuracy for Transformer on IMDB with different sparsification

Target Acc.	k	Communication Round		
		S	EF21-S	LA-S
60%	0.2M	107	<u>105</u>	**95**
	0.3M	<u>105</u>	<u>105</u>	**103**
	0.4M	<u>105</u>	<u>105</u>	**104**
65%	0.2M	761	<u>748</u>	**607**
	0.3M	<u>737</u>	739	**709**
	0.4M	**737**	**737**	**737**

The results on Transformer (Table 10) further demonstrate the communication efficiency of LA-S under sparse communication. For the 60% accuracy target, all methods achieve convergence within a similar number of rounds, with LA-S obtaining slightly better or comparable performance under different sparsification levels. When targeting a higher accuracy of 65%, LA-S achieves more significant communication savings. Specifically, LA-S reduces the required communication rounds to 607 and 709 with 0.2M and 0.3M retained parameters, respectively, outperforming both S and EF21-S. Under 0.4M sparsification, all methods achieve the target accuracy in the same number of rounds. These results indicate that LA-S is particularly effective under more aggressive sparsification settings, where preserving important update information becomes critical for maintaining convergence performance.