

---

# Communication-Compressed Adaptive Gradient Method for Distributed Nonconvex Optimization

---

**Yujia Wang**

Pennsylvania State University

**Lu Lin**

University of Virginia

**Jinghui Chen**

Pennsylvania State University

## Abstract

Due to the explosion in the size of the training datasets, distributed learning has received growing interest in recent years. One of the major bottlenecks is the large communication cost between the central server and the local workers. While error feedback compression has been proven to be successful in reducing communication costs with stochastic gradient descent (SGD), there are much fewer attempts in building communication-efficient adaptive gradient methods with provable guarantees, which are widely used in training large-scale machine learning models. In this paper, we propose a new communication-compressed AMSGrad for distributed nonconvex optimization problem, which is provably efficient. Our proposed distributed learning framework features an effective gradient compression strategy and a worker-side model update design. We prove that the proposed communication-efficient distributed adaptive gradient method converges to the first-order stationary point with the same iteration complexity as uncompressed vanilla AMSGrad in the stochastic nonconvex optimization setting. Experiments on various benchmarks back up our theory.

## 1 Introduction

The recent success of deep learning and large-scale training made it possible to use machine learning to solve complicated real-world tasks, such as computer vision (He et al., 2016), speech recognition (Hin-

ton et al., 2012), natural language processing (Devlin et al., 2018), etc. Due to the explosion in the size of training datasets in recent years, single GPU training on such models can easily take up to weeks or even months to finish. As a consequence, distributed training algorithms have attracted growing interest over the years. In standard distributed settings with one central server and  $n$  workers, local workers parallelly compute local gradients, while the server aggregates the gradients from the workers, updates the model, and sends the new model back to the workers. However, data transmissions across the machines can be quite expansive in terms of both communication costs (especially for cellular networks) and latency. Therefore, various methods were studied in order to achieve communication-efficient distributed learning by either reducing communication bits (Stich et al., 2018; Basu et al., 2019; Karimireddy et al., 2019) or saving the number of communication rounds (Zheng et al., 2017; Chen et al., 2021b). One of the principled approaches for effectively reducing the communication cost is to perform gradient compression before transmissions. It directly compresses the local fresh gradient on each worker before uploading the gradient to the server. However, the compression would slow down the convergence or even diverge (Beznosikov et al., 2020) due to the loss of information at each compression step. Later on, error feedback (Stich et al., 2018; Karimireddy et al., 2019) was proposed to alleviate this problem and reduce the information loss by proposing a compensating error sequence. Each worker compresses and uploads the combination of the last step compensating error and the local fresh gradient instead of compressing the gradient directly. Recent studies (Seide et al., 2014; Karimireddy et al., 2019; Stich et al., 2018) show that error feedback has been widely used in distributed SGD to communicate efficiently and ensure the same convergence rate as vanilla SGD.

While communication-efficient distributed SGD has been widely studied, there are much fewer attempts in building communication-efficient distributed adaptive gradient methods (Kingma and Ba, 2014; Reddi et al., 2018). It has been shown that SGD works less

efficiently compared with adaptive gradient methods when training large-scale models such as BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020) and GAN (Goodfellow et al., 2014). One of the major challenges is that the traditional error feedback mechanism is not compatible with adaptive gradient methods since the variance term (i.e., second-order moments of the historical gradients) in the adapted gradient method can be unstable due to the accumulated compression error (Tang et al., 2021). 1-bit Adam (Tang et al., 2021) partially solve this problem by first using vanilla Adam at the beginning of training to get an estimate of variance term, then freezing this variance term and performing distributed Adam with the fixed variance. While this solution indeed gets rid of the unstable variance issue, the adaptivity is no longer changing and the major part of the algorithm is actually more similar to SGD with momentum. Due to the same reason, the convergence guarantee provided in Tang et al. (2021) is not related to distributed Adam but is an extension to distributed SGD with momentum.

In this paper, we develop a new compression method which is provably efficient, namely **Communication-compressed Distributed Adaptive gradient Method (CD-Adam)**. We revisit the unstable variance issue in distributed AMSGrad (Reddi et al., 2018) and identify the need for an improved gradient compression strategy. Specifically, we adopt the Markov compression sequence recently proposed by Richtárik et al. (2021), which ideally could lead to contractive compression error for gradient descent if the gradient sequence (to be compressed) is convergent. Note that this is not simply applying the conclusion of Richtárik et al. (2021) to the distributed AMSGrad setting, as Richtárik et al. (2021) only deals with worker-to-server compression. While extending Richtárik et al. (2021) to both worker-to-server and server-to-worker compression may seem straightforward for standard gradient descent, it is particularly tricky for the adaptive gradient method (see Section 5 for details). Therefore, it requires carefully designed algorithms and analyses to build a provably efficient distributed adaptive gradient method.

We summarize our contribution as follows:

- We propose a new communication-compressed distributed AMSGrad approach which solves the bottleneck of applying communication compression strategies for fully-functional<sup>1</sup> adaptive gradient methods in the distributed setting. The proposed method largely reduces the communication cost by enforcing both worker-to-server

and server-to-worker communication compression without any warm-up stages.

- We theoretically prove the convergence of our proposed algorithm in the nonconvex stochastic optimization setting. Specifically, we show that our proposed fully compressed CD-Adam can reach its first-order  $\epsilon$ -stationary point within  $O(1/\epsilon^2)$  iterations, which is the same iteration complexity as the uncompressed vanilla AMSGrad. This suggests that without applying any variance-freezing tricks, the fully compressed distributed adaptive gradient method can still provably converge to its  $\epsilon$ -stationary point.
- Thorough experiments on various real-world benchmarks show that our proposed CD-Adam reduces the communication cost by around  $32\times$  over the original AMSGrad and around  $5\times$  over 1-bit Adam.

## 2 Related Work

**Stochastic gradient descent and adaptive gradient methods:** Stochastic gradient descent (SGD) (Robbins and Monro, 1951) is broadly used in training large-scale machine learning problems. Despite being simple to implement, SGD can be sensitive to parameters such as learning rate and slow to converge. Adaptive gradient methods were proposed to further improve over SGD. Adam (Kingma and Ba, 2014), one of the most popular adaptive gradient methods, has shown to be fast convergent and also robust to hyperparameters like learning rate. It designs an adaptive learning rate for each different coordinate using the past gradient. Aside from Adam, AdaGrad (Duchi et al., 2011) applied the second moment of the gradient to adaptive the learning rate, RMSProp (Tieleman et al., 2012) further used the second moment of the gradient with a decay rate, AdaDelta (Zeiler, 2012) is an extension of AdaGrad with a non-increasing learning rate. Later on, Reddi et al. (2018) pointed out a non-convergence issue of Adam, and proposed a new AMSGrad algorithm for ensuring the convergence. Zaheer et al. (2018) studied the effect of adaptive denominator constant  $\nu$  and mini-batch size in the convergence of adaptive gradient methods. Adabound (Luo et al., 2019) proposed with both upper and lower bound for the variance term of Adam. AdamW (Loshchilov and Hutter, 2019) proposed to fix the weight decay regularization in Adam by decoupling the weight decay from the gradient update. Chen et al. (2020) proposed a partially adaptive gradient method and proved its convergence in nonconvex settings. Chen et al. (2019); Zhou et al. (2018) showed the convergence rate of a class of adaptive gradient

<sup>1</sup>Here we use “fully-functional” to differentiate with the variance-freezed Adam used in Tang et al. (2021).

methods under the nonconvex stochastic optimization setting. Alacaoglu et al. (2020) proposed a new framework to derive data-dependent regret bounds with a constant momentum parameter in various settings.

**Distributed SGD and error feedback:** For communication-efficient distributed SGD, one of the most common strategies is to compress the gradients before uploading. Alistarh et al. (2017) provided a theoretical analysis of the centralized compressed distributed SGD. Seide et al. (2014) compressed the coordinates of the gradient into  $\pm 1$  by its sign. Bernstein et al. (2018) proposed signSGD and proved its convergence in the nonconvex setting. Variant works has applied kinds of compression methods such as sparsification (Stich et al., 2018; Basu et al., 2019), quantization (Karimireddy et al., 2019), and sketching (Ivkin et al., 2019). Mishchenko et al. (2019) proposed DIANA, which adopts the compression of gradient difference in distributed settings. Horváth et al. (2019) introduced several variants of DIANA including the variance reduction version. Philippenko and Dieuleveut (2020) introduced a bi-directional compression schemes with memory in distributed or federated setting. Error feedback (or error compensation) largely improves the compression error bound and has been shown to be critical for ensuring fast convergence of the compression mentioned above. Seide et al. (2014) showed that with error feedback, even 1-bit gradient communication under SGD still obtains the convergence rate of vanilla SGD. Karimireddy et al. (2019) applied error feedback to signSGD (Bernstein et al., 2018) under nonconvex setting, Stich et al. (2018) performed error feedback in sparsity strong convex settings, and Stich and Karimireddy (2019) proposed a error-feedback framework also works on nonconvex but single node setting. Tang et al. (2019) proposed bi-directional SGD method combining with error compensation. Gorbunov et al. (2020) proposed a framework with variants of quantized SGD including EC-SGD and EC-SGD-DIANA. Liu et al. (2020); Zheng et al. (2019); Philippenko and Dieuleveut (2021) also studied combining bi-directional compression and error feedback strategy. There are also more works on the distributed SGD under nonconvex optimization (Koloskova et al., 2019; Basu et al., 2019).

**Communication-efficient distributed adaptive gradient method:** There are only fewer attempts in developing communication-efficient distributed adaptive gradient methods. 1-bit Adam (Tang et al., 2021) adopted a variance-freezed Adam by pointing out that the variance term of Adam becomes stable in later training stages. Combined with error feedback, 1-bit Adam achieves the same convergence rate as distributed SGD. Zhong et al. (2021) proposed a gradi-

ent compressed version of LANS (Zheng et al., 2020). Chen et al. (2021a) developed a distributed quantized Adam with error feedback. However, the proposed algorithm in Chen et al. (2021a) can only converge to its  $\epsilon$ -stationary point with worker-to-server compression alone but not fully compressed Adam with both worker-to-server and server-to-worker compression. Another related work is CADA (Chen et al., 2021b), which reduced the communication rounds instead of performing communication compression. CADA adaptively reused the stale update parameters, and it achieved the same convergence rate as vanilla AMSGrad for nonconvex optimization. Note that CADA’s strategy of reducing communication rounds is orthogonal to ours can possibly be combined with ours for further improvements.

### 3 Problem Formulation

In this paper, we aim to solve the following distributed optimization problem under nonconvex stochastic optimization setting:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^n \underbrace{\mathbb{E}_{\xi^{(i)}} f_i(\mathbf{x}; \xi^{(i)})}_{:= f_i(\mathbf{x})}, \quad (3.1)$$

where  $d$  denotes the dimension of the model,  $n$  denotes the total amount of workers,  $\xi^{(i)}$  denotes the stochastic noise variable, and  $f_i(\mathbf{x})$  denotes the loss function on the  $i$ -th worker. In the stochastic setting, we cannot obtain the full gradient of loss function  $f_i(\mathbf{x})$ . Instead, we can only obtain the unbiased estimators of  $\nabla f_i(\mathbf{x})$ , i.e.,  $\nabla f_i(\mathbf{x}; \xi^{(i)})$ .

First, let us revisit the vanilla distributed setting of AMSGrad (Reddi et al., 2018). Consider a distributed learning system that contains a parameter server and  $n$  workers, with each worker  $i$  owns its local data from distribution  $\mathcal{D}_i$ . Let  $\mathbf{x}_t$  denotes the current model at  $t$ -th iteration. The server will first broadcast  $\mathbf{x}_t$  to all the workers in each iteration. Each worker  $i$  computes the stochastic gradient  $\mathbf{g}_t^{(i)} = \nabla f_i(\mathbf{x}_t; \xi_t^{(i)})$  using the local samples, and then uploads  $\mathbf{g}_t^{(i)}$  to the server. The server then aggregates the stochastic gradients and obtains  $\mathbf{g}_t = \frac{1}{n} \sum_{i=1}^n \mathbf{g}_t^{(i)}$ . The model uses  $\mathbf{g}_t$  to update parameter via vanilla single node AMSGrad:

$$\begin{aligned} \mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t, \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2, \\ \widehat{\mathbf{v}}_t &= \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t), \mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t, \end{aligned}$$

where  $\widehat{\mathbf{V}}_t = \text{diag}(\widehat{\mathbf{v}}_t + \nu)$ ,  $\alpha_t > 0$  is the step size,  $\nu > 0$  is a small number for numerical stability, and  $\beta_1, \beta_2 \in [0, 1]$  are adaptive hyperparameters. This vanilla distributed AMSGrad achieves the same convergence rate as its centralized version (Reddi et al.,

2018). However, the worker-to-server and server-to-worker communications in each iteration can be extremely expansive especially for cellular networks, and the communication latency also makes the overall system less efficient. Therefore, attempts have been made to find novel approaches that can further reduce the communication cost while maintaining a similar convergence rate as its uncompressed counterpart. In the following, let's first revisit some traditional communication compression strategies for distributed SGD and drawbacks of applying them on distributed adaptive gradient methods.

## 4 Existing Solutions and Drawbacks

**Naive compression for SGD:** The simplest strategy to reduce communication cost is to directly compress the local gradient  $\mathbf{g}_t^{(i)}$  with a compressor  $\mathcal{C}(\cdot)$  before sending to the server. The server aggregates the compressed gradient  $\widehat{\mathbf{g}}_t^{(i)} = \mathcal{C}(\mathbf{g}_t^{(i)})$  and updates model by

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\alpha}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_t^{(i)},$$

where  $\alpha$  denotes the learning rate. The common choice of  $\mathcal{C}(\cdot)$  can be top- $k$  (Basu et al., 2019) or sign operation<sup>2</sup> (leads to signSGD (Bernstein et al., 2018)). Although this naive compression method is intuitive, it can diverge in practice, even in simple quadratic problems (Beznosikov et al., 2020) or constraint linear problems (Karimireddy et al., 2019). Intuitively speaking, one of the major drawbacks of naive compression is that the compression error is accumulating during the training process. Each step will introduce new errors that cannot be canceled later, and the accumulation of compression error leads the divergence.

**Error feedback for SGD:** Error feedback (EF), or error compensation (Karimireddy et al., 2019; Stich et al., 2018) is widely used for correcting the bias generated by compression errors. Distributed SGD with error feedback effectively reduces the communication bits by introducing a compensating error sequence to cancel the compression error in previous iterations and obtains the same convergence rate as vanilla SGD. Specifically, error feedback introduces a new sequence  $\delta_t^{(i)}$ , which denotes the accumulated compression error at iteration  $t$ . At  $t$ -th iteration, the  $i$ -th worker computes the compressed gradient  $\widehat{\mathbf{g}}_t^{(i)}$  based on the previous iteration's error  $\delta_{t-1}^{(i)}$  and the current local gradient  $\mathbf{g}_t^{(i)}$ , i.e.,  $\widehat{\mathbf{g}}_t^{(i)} = \mathcal{C}(\mathbf{g}_t^{(i)} + \delta_{t-1}^{(i)})$ . And the new compensating error is updated by  $\delta_t^{(i)} = \mathbf{g}_t^{(i)} + \delta_{t-1}^{(i)} - \widehat{\mathbf{g}}_t^{(i)}$ .

<sup>2</sup>See the Supplemental A for more details about the compressors.

Upon compression, the server collects the compressed gradients  $\widehat{\mathbf{g}}_t^{(i)}$  from all workers and update model parameters via vanilla SGD. Karimireddy et al. (2019) showed that the compression error of error feedback is bounded by constant if the following assumption hold for the biased compressor.

**Assumption 4.1** (Biased compressor). Consider a biased operator  $\mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ , there exists a constant  $0 < \pi \leq 1$  such that

$$\mathbb{E}_{\mathcal{C}}[\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|_2^2] \leq \pi \|\mathbf{x}\|_2^2, \quad \forall \mathbf{x} \in \mathbb{R}^d. \quad (4.1)$$

Note that  $\pi = 0$  leads to  $\mathcal{C}(\mathbf{x}) = \mathbf{x}$ . Assumption 4.1 is a common assumption for biased compressor (Richtárik et al., 2021; Karimireddy et al., 2019; Stich et al., 2018). Canonical examples of the compressor satisfying Assumption 4.1 include top- $k$  or random- $k$  as well as scaled sign compressor<sup>3</sup>. With Assumption 4.1, the distributed SGD with error feedback (Karimireddy et al., 2019) in nonconvex setting achieves the same convergence rate as vanilla SGD.

**Drawbacks of error feedback on adaptive gradient methods:** While error feedback guarantees bounded accumulated gradient compression error, in adaptive gradient methods (Kingma and Ba, 2014; Reddi et al., 2018), the variance term  $\mathbf{v}_t$ , which is the moving average of the quadratic of gradient, can be unstable due to accumulating gradient compression error (Tang et al., 2021). Specifically, let's denote  $\mathbf{g}_t$  as the averaged fresh gradient without compression (average of all  $\mathbf{g}_t^{(i)}$ ), denote  $\widehat{\mathbf{g}}_t$  as the averaged compressed gradient (average of all  $\widehat{\mathbf{g}}_t^{(i)}$ ). The updating rule for  $\mathbf{v}_{t+1}$  follows:

$$\begin{aligned} \mathbf{v}_{t+1} &= \beta_2 \mathbf{v}_t + (1 - \beta_2) \widehat{\mathbf{g}}_t^2 \\ &= \beta_2 \mathbf{v}_t + (1 - \beta_2) [\widehat{\mathbf{g}}_t - \mathbf{g}_t + \mathbf{g}_t]^2 \\ &= \beta_2 \mathbf{v}_t + (1 - \beta_2) \mathbf{g}_t^2 + \underbrace{(1 - \beta_2)(\widehat{\mathbf{g}}_t - \mathbf{g}_t)^2}_{\text{accumulating error term}} \\ &\quad + 2(1 - \beta_2) \langle \mathbf{g}_t, \widehat{\mathbf{g}}_t - \mathbf{g}_t \rangle. \end{aligned} \quad (4.2)$$

Tang et al. (2021) claims that the inner product term in (4.2) can possibly be canceled out during training, while the quadratic term will certainly accumulate. Since the traditional error feedback can only guarantee constant compression error, the accumulation of the quadratic error will make the variance diverge. Therefore, the communication-efficient adaptive gradient method actually requires a stronger gradient compression error bound to obtain a stable variance term.

<sup>3</sup>See the supplemental materials for more details about the above three compressors as well as other compressors satisfying Assumption 4.1.

## 5 Proposed Method

In this section, we formally develop our proposed **Communication-compressed Distributed Adaptive gradient Method** (CD-Adam). Our proposed method consists of two key components: Markov compression sequence and worker-side model update design, which jointly provide a better compression error bound. We first investigate the definition and property of Markov compression sequence (Richtárik et al., 2021).

**Markov compression sequence:** Markov compression sequence is introduced<sup>4</sup> in Richtárik et al. (2021). Given a biased compressor  $\mathcal{C}(\cdot)$  and a sequence of vectors  $\{\mathbf{w}_t\}$ , Markov compression sequence  $\{\hat{\mathbf{w}}_t\}$  can be recursively defined as:  $\hat{\mathbf{w}}_0 = \mathcal{C}(\mathbf{w}_0)$ ,  $\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t + \mathcal{C}(\mathbf{w}_{t+1} - \hat{\mathbf{w}}_t)$ . The main advantage of Markov compression sequence lies in that the compression error can be largely reduced if the underlying sequence  $\{\mathbf{w}_t\}$  is convergent:

$$\begin{aligned}\|\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1}\|^2 &= \|\hat{\mathbf{w}}_t + \mathcal{C}(\mathbf{w}_{t+1} - \hat{\mathbf{w}}_t) - \mathbf{w}_{t+1}\|^2 \\ &\leq \pi \|\mathbf{w}_{t+1} - \hat{\mathbf{w}}_t\|^2 \\ &\leq \pi(1 + \gamma) \|\hat{\mathbf{w}}_t - \mathbf{w}_t\|^2 \\ &\quad + \pi(1 + \gamma^{-1}) \|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2,\end{aligned}\tag{5.1}$$

where the last inequality holds due to Young’s inequality. As can be seen from (5.1), the compression error  $\|\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1}\|$  is directly controlled by  $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|$ . If the sequence of  $\{\mathbf{w}_t\}$  is convergent, Markov compression error at  $t$ -step will be much smaller than the constant error bound obtained by plain error feedback. In particular, Richtárik et al. (2021) showed that if the sequence of  $\{\mathbf{w}_t\}$  achieves a linear rate of convergence, the compression error of the Markov compression can converge to 0 as  $t \rightarrow \infty$ .

**Worker-side model update:** Richtárik et al. (2021) only considered a one-way compression, i.e., workers compress the gradient difference  $\mathbf{g}_t^{(i)} - \hat{\mathbf{g}}_{t-1}^{(i)}$  and update  $\mathcal{C}(\mathbf{g}_t^{(i)} - \hat{\mathbf{g}}_{t-1}^{(i)})$  to the server, while the server only updates model  $\mathbf{x}_{t+1}$  and broadcasts it to workers. In this case, the server broadcast still results in high communication overhead. Therefore, to further reduce the communication cost, we change the model update style from server-side update to worker-side update, i.e., the model  $\mathbf{x}_t$  is stored and updated on each worker instead of on the server. The reason is that, in server-side update style, the server will need to compress the model update parameter  $\hat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t$  for reducing the communication cost, however, it is difficult to obtain the convergence of the underlying sequence  $\hat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t$ , which

<sup>4</sup>Note that Künstner (2017) also proposed QDGD-F algorithm which is similar to the Markov compression sequence.

is essential for the proof of the Markov compression sequence. Thus, it is difficult to get a better compression error bound for the model update. In our design, the server simply aggregates the compressed gradients and compressed it again before sending back to the workers. The compression errors via Markov compressor are directly related to fresh stochastic gradients or aggregated gradients, whose convergences are much easier to obtain. This two-way compression together with worker-side model update take the full advantage of the Markov compressor for establishing the theoretical guarantees of our proposed method.

**Algorithm overview:** Combining Markov compression sequence and worker-side model update design, we propose a new communication-compressed distributive gradient method, which is summarized in Algorithm 1. Specifically, at  $t$ -th iteration, each worker first computes the stochastic gradient  $\mathbf{g}_t^{(i)}$  with a mini-batch size  $\tau$ , then builds the Markov compression sequence  $\hat{\mathbf{g}}_t^{(i)}$  with  $\mathbf{c}_t^{(i)} = \mathcal{C}(\mathbf{g}_t^{(i)} - \hat{\mathbf{g}}_{t-1}^{(i)})$  and  $\hat{\mathbf{g}}_{t-1}^{(i)}$ . Note that the Markov compression sequence  $\hat{\mathbf{g}}_t^{(i)}$  is no longer bits-compressed, therefore, only  $\mathbf{c}_t^{(i)}$  is being uploaded to the server. On the other hand, the server updates the aggregated compressed gradient by  $\hat{\mathbf{g}}_t = \hat{\mathbf{g}}_{t-1} + \frac{1}{n} \sum_{i=1}^n \mathbf{c}_t^{(i)}$  as the last iterate  $\hat{\mathbf{g}}_{t-1}$  has been stored locally in the server. The server then builds another Markov compression sequence  $\tilde{\mathbf{g}}_t$  based on  $\hat{\mathbf{g}}_{t-1}$  and  $\mathbf{c}_t = \mathcal{C}(\hat{\mathbf{g}}_t - \tilde{\mathbf{g}}_{t-1})$  and sends  $\mathbf{c}_t$  to all workers. Similarly, each worker can recover the Markov compression sequence  $\tilde{\mathbf{g}}_t$  upon receiving  $\mathbf{c}_t$ , since the last iterate  $\tilde{\mathbf{g}}_{t-1}$  has been locally stored in each worker. Then each worker uses the double-compressed gradient  $\tilde{\mathbf{g}}_t$  for updating model  $\mathbf{x}_{t+1}$  as in standard AMSGrad.

Algorithm 1 is indeed communication-efficient for distributed learning. It improves the communication efficiency by reducing the communication bits during transmission. Only the compressed vectors instead of the full precision vectors are transferred for both server-to-worker and worker-to-server communications. Specifically, if we adopt the scaled sign compressor, CD-Adam only takes 1-bit<sup>5</sup> instead of 32-bits per dimension in each communication round. This greatly reduces the communication costs for distributed implementation of adaptive methods. Note that compared with 1-bit Adam (Tang et al., 2021), which performs a few epochs of uncompressed Adam at the beginning of training, our proposed method is much more communication-efficient (see Figure 1) as we start the compression from the very first iteration.

<sup>5</sup>Rigorously, the scaling number will also take 32-bits for communication, so the overall cost for compressing a  $d$ -dimensional vector should be  $32 + d$  bits.

**Algorithm 1** Communication-Compressed Distributed Adaptive Gradient Method (CD-Adam)

**Input:** initial point  $\mathbf{x}_1$ , non-increasing step size  $\{\alpha_t\}_{t=1}^T$ ,  $\beta_1, \beta_2, \nu$ , batch size  $\tau$ , compressor  $\mathcal{C}(\cdot)$ .

- 1:  $\mathbf{g}_0^{(i)} \leftarrow 0, \mathbf{m}_0 \leftarrow 0, \mathbf{v}_0 \leftarrow 0, \hat{\mathbf{g}}_0^{(i)} = \mathcal{C}(\mathbf{g}_0^{(i)}), \tilde{\mathbf{g}}_0 = \mathcal{C}\left(\frac{1}{n} \sum_{i=1}^n \hat{\mathbf{g}}_0^{(i)}\right)$
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:   **(On  $i$ -th worker)**
- 4:     Compute local stochastic gradient  $\mathbf{g}_t^{(i)}$  with batch size  $\tau$
- 5:     Compress  $\mathbf{c}_t^{(i)} = \mathcal{C}(\mathbf{g}_t^{(i)} - \hat{\mathbf{g}}_{t-1}^{(i)})$
- 6:     Send  $\mathbf{c}_t^{(i)}$  to the server and update local state  $\hat{\mathbf{g}}_t^{(i)} = \hat{\mathbf{g}}_{t-1}^{(i)} + \mathbf{c}_t^{(i)}$
- 7:   **(On Server)**
- 8:     Update  $\hat{\mathbf{g}}_t = \hat{\mathbf{g}}_{t-1} + \frac{1}{n} \sum_{i=1}^n \mathbf{c}_t^{(i)}$
- 9:     Compress  $\mathbf{c}_t = \mathcal{C}(\hat{\mathbf{g}}_t - \tilde{\mathbf{g}}_{t-1})$
- 10:    Send  $\mathbf{c}_t$  to all the workers and update local state  $\tilde{\mathbf{g}}_t : \tilde{\mathbf{g}}_t = \tilde{\mathbf{g}}_{t-1} + \mathbf{c}_t$
- 11:   **(On  $i$ -th worker)**
- 12:     Update  $\tilde{\mathbf{g}}_t = \tilde{\mathbf{g}}_{t-1} + \mathbf{c}_t$
- 13:      $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \tilde{\mathbf{g}}_t$
- 14:      $\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \tilde{\mathbf{g}}_t^2$
- 15:      $\hat{\mathbf{v}}_t = \max(\hat{\mathbf{v}}_{t-1}, \mathbf{v}_t)$
- 16:     Update  $\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \hat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t$  with  $\hat{\mathbf{V}}_t = \text{diag}(\hat{\mathbf{v}}_t + \nu)$
- 17: **end for**

## 6 Convergence Analysis

In this section, we present the convergence results of our proposed Communication-Compressed Distributed Adaptive Gradient Method (CD-Adam). Before we jump into the main theorem, let us first introduce the additional assumptions needed.

**Assumption 6.1** (Smoothness). Each component loss function on the  $i$ -th worker  $f_i(\mathbf{x}) = \mathbb{E}[f(\mathbf{x}; \xi^{(i)})]$  is  $L$ -smooth, i.e.,  $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ ,

$$|f_i(\mathbf{x}) - f_i(\mathbf{y}) - \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle| \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|_2^2.$$

Assumption 6.1 is a standard assumption for nonconvex stochastic optimization (Koloskova et al., 2019; Basu et al., 2019; Richtárik et al., 2021). Note that the  $L$ -smoothness assumption on each worker's loss  $f_i(\cdot)$  implies the  $L$ -smoothness condition on  $f(\cdot)$ . Assumption 6.1 also implies the  $L$ -gradient Lipschitz condition, i.e.,  $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$ .

**Assumption 6.2** (Bounded gradient). Each component loss function on the  $i$ -th worker  $f_i(\mathbf{x})$  has  $G$ -bounded stochastic gradient on  $\ell_2$  and has  $G_\infty$ -bounded stochastic gradient on  $\ell_\infty$ , i.e., for all  $\xi$ ,

$$\|\nabla f_i(\mathbf{x}; \xi)\|_2 \leq G, \quad \|\nabla f_i(\mathbf{x}; \xi)\|_\infty \leq G_\infty.$$

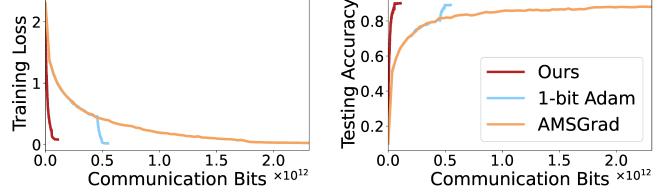


Figure 1: Comparison of training loss and testing accuracy against communication bits of training ResNet-18 on CIFAR-10. Our proposed method achieves around  $32\times$  communication cost improvement over the original AMSGrad and around  $5\times$  over 1-bit Adam.

Assumption 6.2 is a common assumption in adaptive gradient literature Kingma and Ba (2014); Chen et al. (2019); Zhou et al. (2018); Chen et al. (2018). It also implies the bounded gradient for  $f(\mathbf{x})$ , i.e.,  $\|\nabla f(\mathbf{x}; \xi)\|_2 \leq G$ ,  $\|\nabla f(\mathbf{x}; \xi)\|_\infty \leq G_\infty$ .

**Assumption 6.3** (Bounded local variance). Each stochastic gradient on the  $i$ -th worker  $\mathbf{g}^i = \nabla f_i(\mathbf{x}, \xi^i)$  has a bounded variance, i.e.,  $\forall \mathbf{x}$ ,

$$\mathbb{E}_{\xi^i \sim \mathcal{D}_i} \|\nabla f_i(\mathbf{x}, \xi^i) - \nabla f_i(\mathbf{x})\|^2 \leq \sigma^2.$$

Assumption 6.3 implies that the variance of stochastic gradients is bounded on each worker. The assumption of bounded variance for stochastic gradient has also been used in (Basu et al., 2019; Karimireddy et al., 2019; Horváth et al., 2019).

Now we are ready to present our main theorem.

**Theorem 6.4.** Under Assumptions 4.1, 6.1, 6.2 and 6.3, suppose  $\alpha_t = \alpha$ , denote  $N$  as the number of data samples on each worker, the iterates of Algorithm 1 achieves an  $\epsilon$ -stationary point of (3.1), i.e.,  $\min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \leq \epsilon$ , if

$$\begin{aligned} \alpha &\leq \frac{n\epsilon}{6nM_3 + 6M_4\sigma^2}, \tau \geq \left\lceil \frac{N(3M_5\sigma)^2}{(N-1)\epsilon^2 + (3M_5\sigma)^2} \right\rceil, \\ T &\geq \left\lceil \frac{36M_1M_3}{\epsilon^2} + \frac{36M_1M_4\sigma^2}{n\epsilon^2} + \frac{3M_2}{\epsilon} \right\rceil, \end{aligned} \quad (6.1)$$

where

$$\begin{aligned} M_1 &= C \cdot \Delta f, \quad M_2 = \frac{CG\tilde{G}}{(1 - \beta_1)\sqrt{\nu}}, \\ M_3 &= \frac{32CC_1\tilde{G}^2}{\nu} + \frac{2\sqrt{\pi}CLG\tilde{G}\sqrt{d}}{\nu(1 - \sqrt{\pi})^2} \\ M_4 &= \frac{4CC_1}{\nu}, \quad M_5 = \frac{4\sqrt{\pi}CG}{\sqrt{\nu}(1 - \sqrt{\pi})^2}, \end{aligned}$$

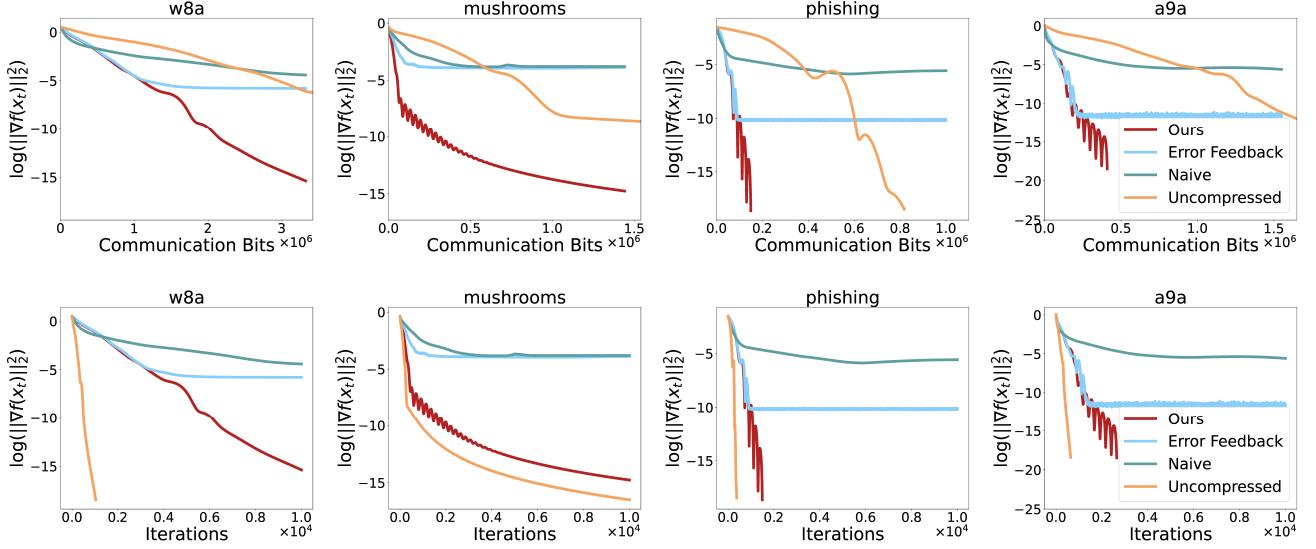


Figure 2: Gradient norm comparison of different compressing strategies on nonconvex logistic regression trained by AMSGrad with scaled sign compressor. The upper row shows the norm convergence with respect to the communication cost, and the lower row is with respect to the training iteration.

and

$$\begin{aligned} \Delta f &= f(\mathbf{x}_1) - \inf_x f(\mathbf{x}), \tilde{G} = \frac{(1 + \sqrt{\pi})^2}{(1 - \sqrt{\pi})^2} G, \\ C &= 2(\tilde{G}_\infty^2 + \nu)^{1/2}, C_1 = 2L + 3L \left( \frac{\beta_1}{1 - \beta_1} \right)^2, \\ \tilde{G}_\infty &= \frac{(1 + \sqrt{\pi})^2}{(1 - \sqrt{\pi})^2} G_\infty. \end{aligned}$$

**Remark 6.5.** Theorem 6.4 suggests that Algorithm 1 reaches the  $\epsilon$ -stationary point of (3.1) with  $O(1/\epsilon^2)$  iterations, which matches the iteration complexity of uncompressed vanilla AMSGrad. This suggests that our proposed method is indeed provably efficient, and the fully compressed distributed adaptive gradient method can reach  $\epsilon$ -stationary point with the same iteration complexity as the uncompressed counterpart without applying any variance-freezing tricks.

**Remark 6.6.** Theorem 6.4 also suggests that a larger number of workers  $n$  can lead to a better iteration complexity as shown in (6.1).

In addition, the condition on the mini-batch size  $\tau$  is needed for controlling the stochastic variance in a mini-batch. Note that  $\tau$  is always smaller than the number of data samples on each worker  $N$  as the full batch gradient does not incur stochastic variance.

## 7 Experiments

In this section, we present empirical results of our proposed communication-compressed distributed

adaptive gradient method and compare it with other communication-compressed distributed baselines. Specifically, we first present the experimental results on a nonconvex logistic regression problem as an illustrative case study. Then we present the deep learning experiments of image classification on standard benchmarks.

### 7.1 Illustrative case study: nonconvex logistic regression

We consider minimizing the following logistic regression problem with a nonconvex regularizer similar to Richtárik et al. (2021),

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \log \left( 1 + \exp(-y_i a_i^\top x) \right) + \lambda \sum_{j=1}^d \frac{x_j^2}{1 + x_j^2}, \quad (7.1)$$

where  $a_i \in \mathbb{R}^d$ ,  $y_i$  are training data with value  $\pm 1$ , and  $\lambda$  is the regularizer parameter. We set  $\lambda = 0.1$  in the following nonconvex logistic experiments.

We use datasets `phishing`, `mushrooms`, `a9a` and `w8a` from LibSVM (Chang and Lin, 2011). We equally separate each dataset to  $n = 20$  parts, and we use full batch gradients in this experiment. For each method, we choose the best step size starting from 0.001 and increase it by adding 0.002 till achieving 0.01.

Let's first study the effect of using different compression strategies on AMSGrad, though the other strategies do not enjoy any theoretical guarantees. Figure 2 compares the gradient norm convergence of our

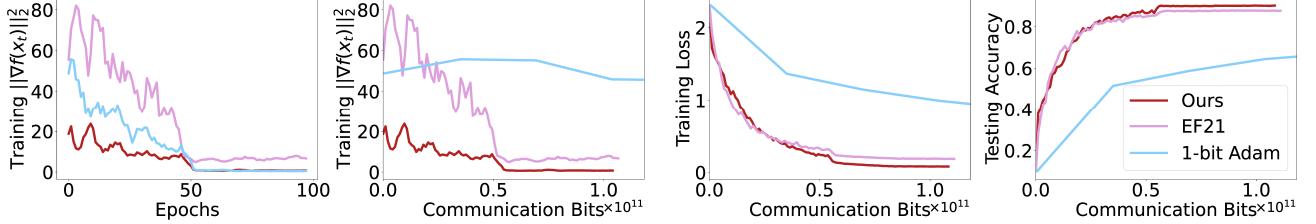


Figure 3: Comparison on gradient norm, training loss and testing accuracy among our proposed method and the baselines when training ResNet-18 model on CIFAR-10.

proposed CD-Adam with AMSGrad using error feedback, naive compression and without compression at all. For all compressed methods, we use the scaled sign as a canonical example of biased compressor<sup>6</sup>  $\mathcal{C}(\cdot)$ . We observe that our proposed CD-Adam achieves the best performances on all four datasets against the other three compression strategies. Specifically, if we compare the communication cost, CD-Adam achieves much better communication efficiency against uncompressed AMSGrad. Compared with the error feedback and the naive compression, CD-Adam still achieves a much smaller gradient norm under the same communication budgets. If we compare the performance of different compression strategies under the same iteration complexity, our proposed algorithm achieves roughly the same final gradient norm as the uncompressed AMSGrad across all four datasets. In fact, it achieves a much better convergence result comparing with the error feedback and the naive compression strategies, whose gradient norm stops decreasing at the early stages.

## 7.2 Deep learning experiments on image classification

We compare CD-Adam with several state-of-the-art communication-compressed distributed learning algorithms that are provably efficient, including: (1) EF21 (Richtárik et al., 2021)<sup>7</sup> (2) 1-bit Adam (Tang et al., 2021). We adopt CD-Adam with scaled sign compressor in this experiment. Note that the original EF21 paper (Richtárik et al., 2021) adopts the top-K compressor as the base compressor  $\mathcal{C}(\cdot)$  with only worker-to-server compression applied. For a fair comparison, we further extend EF21 to allow both worker-to-server and server-to-worker compression and choose  $K = 0.016d$  such that the communication compression ratio keeps roughly the same as the scaled-sign based compressor.

We train CIFAR10 (Krizhevsky et al., 2009) us-

<sup>6</sup>Additional experiments based on the Top- $k$  biased compressor can be found in the supplemental materials.

<sup>7</sup>The comparison to SGD with Error Feedback has been presented in Richtárik et al. (2021).

ing three popular models: ResNet-18 (He et al., 2016), VGG-16 (Simonyan and Zisserman, 2014) and WideResNet-16-4 (Zagoruyko and Komodakis, 2016). The image classification dataset CIFAR10 includes a training set of 50000 images and a test set of 10000 images, where each image is assigned one of the 10 labels. The dataset is split into  $n = 8$  equal parts, which are distributed to 8 workers. The mini-batch size for each worker is set to be 128. We set the learning rate as  $1 \times 10^{-1}$  for SGD, which is used in EF21, and  $1 \times 10^{-4}$  for 1-bit Adam and our proposed CD-Adam. We also set  $\beta_1 = 0.9$  and  $\beta_2 = 0.99$  for 1-bit Adam and our CD-Adam. For 1-bit Adam, its warm-up epochs are set as 13 according to its original paper (Tang et al., 2021). The normalized weight decay is set to  $5 \times 10^{-4}$  for all methods. We test all methods for 100 epochs and decay the learning rate by 0.1 at the 50th and 75th epoch. Due to the space limit, we leave the experimental results on VGG-16 and WideResNet-16-4 models in the supplemental materials.

We show that our proposed algorithm enjoys a fast convergence speed with high accuracy and less communication cost comparing with EF21 and 1-bit Adam. The first and second plots in Figure 3 demonstrate the training gradient norm in terms of epochs and communication bits respectively. We can observe that our proposed CD-Adam obtains a smaller gradient norm than EF21 and 1-bit Adam under the same epoch or communication budget. Note that since 1-bit Adam will need to run uncompressed Adam for a few epochs as a warm-up, its per communication bits performance is actually much worse (has been shown in Figure 1). The third plot in Figure 3 shows the training loss against communication bits on CIFAR-10 dataset. At the early stage of the training process, CD-Adam and EF21 obtain a similar speed of reducing the training loss, while 1-bit Adam is much slower due to the warm-up process. At later stages, CD-Adam shows a clear advantage compared with EF21. The last plot in Figure 3 shows the test accuracy against communication bits. Similar phenomenon can be observed that CD-Adam achieves the overall best test accuracy compared with EF21 and 1-bit Adam, under the same communication budget. These results suggest that our pro-

posed CD-Adam is indeed much more communication-efficient while maintaining a high accuracy compared with other communication compression baselines.

## 8 Conclusion

In this paper, we propose a communication-compressed distributed adaptive gradient method which solves the bottleneck of applying communication compression strategies for adaptive gradient methods in distributed settings. We provide theoretical convergence analysis in the nonconvex stochastic optimization setting and show that our proposed algorithm converges to an  $\epsilon$ -stationary point with the same iteration complexity as the uncompressed vanilla AMSGrad. Furthermore, compared with prior work which adopts variance-freezed Adam, our proposed algorithm is fully adaptive during the entire training process. It does not require any warm-up procedure with uncompressed communication, leading to better empirical results in terms of both performance and communication cost.

## References

- Alacaoglu, A., Malitsky, Y., Mertikopoulos, P., and Cevher, V. (2020). A new regret analysis for adam-type algorithms. In *International Conference on Machine Learning*, pages 202–210. PMLR.
- Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. (2017). Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720.
- Basu, D., Data, D., Karakus, C., and Diggavi, S. (2019). Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations. *arXiv preprint arXiv:1906.02367*.
- Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. (2018). signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR.
- Beznosikov, A., Horváth, S., Richtárik, P., and Safranyan, M. (2020). On biased compression for distributed learning. *arXiv preprint arXiv:2002.12410*.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3).
- Chen, C., Shen, L., Huang, H., and Liu, W. (2021a). Quantized adam with error feedback. *ACM Trans. Intell. Syst. Technol.*, 12(5).
- Chen, J., Zhou, D., Tang, Y., Yang, Z., Cao, Y., and Gu, Q. (2020). Closing the generalization gap of adaptive gradient methods in training deep neural networks. In *International Joint Conferences on Artificial Intelligence*.
- Chen, T., Guo, Z., Sun, Y., and Yin, W. (2021b). Cada: Communication-adaptive distributed adam. In *International Conference on Artificial Intelligence and Statistics*, pages 613–621. PMLR.
- Chen, X., Liu, S., Sun, R., and Hong, M. (2018). On the convergence of a class of adam-type algorithms for non-convex optimization.
- Chen, X., Liu, S., Sun, R., and Hong, M. (2019). On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. In *J. Mach. Learn. Res.*
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Gorbunov, E., Kovalev, D., Makarenko, D., and Richtárik, P. (2020). Linearly converging error compensated sgd. *Advances in Neural Information Processing Systems*, 33:20889–20900.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29.
- Horváth, S., Kovalev, D., Mishchenko, K., Stich, S., and Richtárik, P. (2019). Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*.
- Ivkin, N., Rothchild, D., Ullah, E., Braverman, V., Stoica, I., and Arora, R. (2019). Communication-efficient distributed sgd with sketching. *arXiv preprint arXiv:1903.04488*.

- Karimireddy, S. P., Rebjock, Q., Stich, S., and Jaggi, M. (2019). Error feedback fixes signsgd and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261. PMLR.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Koloskova, A., Lin, T., Stich, S. U., and Jaggi, M. (2019). Decentralized deep learning with arbitrary communication compression. *arXiv preprint arXiv:1907.09356*.
- Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.
- Künstner, F. (2017). Fully quantized distributed gradient descent. Technical report.
- Liu, X., Li, Y., Tang, J., and Yan, M. (2020). A double residual compression algorithm for efficient distributed learning. In *International Conference on Artificial Intelligence and Statistics*, pages 133–143. PMLR.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Luo, L., Xiong, Y., Liu, Y., and Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*.
- Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. (2019). Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*.
- Philippenko, C. and Dieuleveut, A. (2020). Artemis: tight convergence guarantees for bidirectional compression in federated learning. *arXiv preprint arXiv:2006.14591*.
- Philippenko, C. and Dieuleveut, A. (2021). Preserved central model for faster bidirectional compression in distributed settings. *Advances in Neural Information Processing Systems*, 34.
- Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations*.
- Richtárik, P., Sokolov, I., and Fatkhullin, I. (2021). Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34.
- Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407.
- Seide, F., Fu, H., Droppo, J., Li, G., and Yu, D. (2014). 1-bit stochastic gradient descent and application to data-parallel distributed training of speech dnns. In *Interspeech 2014*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Stich, S. U., Cordonnier, J.-B., and Jaggi, M. (2018). Sparsified sgd with memory. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 4447–4458. Curran Associates, Inc.
- Stich, S. U. and Karimireddy, S. P. (2019). The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*.
- Tang, H., Gan, S., Awan, A. A., Rajbhandari, S., Li, C., Lian, X., Liu, J., Zhang, C., and He, Y. (2021). 1-bit adam: Communication efficient large-scale training with adam’s convergence speed. In *International Conference on Machine Learning*, pages 10118–10129. PMLR.
- Tang, H., Yu, C., Lian, X., Zhang, T., and Liu, J. (2019). DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6155–6165, Long Beach, California, USA. PMLR.
- Tieleman, T., Hinton, G., et al. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.
- Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- Zaheer, M., Reddi, S., Sachan, D., Kale, S., and Kumar, S. (2018). Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pages 9793–9803.
- Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Zheng, S., Huang, Z., and Kwok, J. (2019). Communication-efficient distributed blockwise momentum sgd with error-feedback. *Advances in Neural Information Processing Systems*, 32.
- Zheng, S., Lin, H., Zha, S., and Li, M. (2020). Accelerated large batch optimization of bert pretraining in 54 minutes. *arXiv preprint arXiv:2006.13484*.
- Zheng, S., Meng, Q., Wang, T., Chen, W., Yu, N., Ma, Z.-M., and Liu, T.-Y. (2017). Asynchronous stochastic gradient descent with delay compensation. In *In-*

*t*ernational Conference on Machine Learning, pages  
4120–4129. PMLR.

Zhong, Y., Xie, C., Zheng, S., and Lin, H. (2021).  
Compressed communication for distributed training:  
Adaptive methods and system. *arXiv preprint  
arXiv:2105.07829*.

Zhou, D., Chen, J., Cao, Y., Tang, Y., Yang, Z., and  
Gu, Q. (2018). On the convergence of adaptive  
gradient methods for nonconvex optimization.

## A Compression Schemes

Here we provide several compressors satisfy Assumption 4.1.

**Rand- $k$**  (Stich et al., 2018): For  $1 \leq k \leq d$ , denotes  $S$  as chosen  $k$  elements randomly in  $[d] = \{1, 2, \dots, d\}$ , i.e.  $S \subseteq [d] = \{1, 2, \dots, d\}$ ,  $|S| = k$ . Denotes  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_d$  as standard unit basis vectors in  $\mathbb{R}^d$ , for  $\mathbf{x} \in \mathbb{R}^d$ , the compressor  $\mathcal{C}_{\text{rand}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is defined as

$$\mathcal{C}_{\text{rand}}(\mathbf{x}) = \sum_{i \in S} \mathbf{x}_i \mathbf{e}_i.$$

For Rand- $k$  compressor, we have

$$\mathbb{E}[\|\mathcal{C}_{\text{Rand}}(\mathbf{x}) - \mathbf{x}\|_2^2] = \mathbb{E}\left[\left\|\sum_{i \notin S} \mathbf{x}_i \mathbf{e}_i\right\|_2^2\right] = \left(1 - \frac{k}{d}\right)\|\mathbf{x}\|_2^2. \quad (\text{A.1})$$

**Top- $k$**  (Stich et al., 2018): For  $1 \leq k \leq d$  and  $\mathbf{x} \in \mathbb{R}^d$ , the order of  $\mathbf{x}$  is ordered by the magnitude of coordinates with  $|x_{(1)}| \leq |x_{(2)}| \leq \dots \leq |x_{(d)}|$ . The compressor  $\mathcal{C}_{\text{Top}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is defined as

$$\mathcal{C}_{\text{Top}}(\mathbf{x}) = \sum_{i=d-k+1}^d \mathbf{x}_{(i)} \mathbf{e}_{(i)}.$$

For Top- $k$  compressor, considering the defition of rand- $k$  compressor, we have

$$\|\mathcal{C}_{\text{Top}}(\mathbf{x}) - \mathbf{x}\|_2 \leq \|\mathcal{C}_{\text{Rand}}(\mathbf{x}) - \mathbf{x}\|_2,$$

thus we conclude

$$\|\mathcal{C}_{\text{Top}}(\mathbf{x}) - \mathbf{x}\|_2 \leq \left(1 - \frac{k}{d}\right)\|\mathbf{x}\|_2^2.$$

**Scaled sign** (Karimireddy et al., 2019): For  $1 \leq k \leq d$  and  $\mathbf{x} \in \mathbb{R}^d$ , the scaled sign compressor  $\mathcal{C}_{\text{sign}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is defined as

$$\mathcal{C}_{\text{sign}}(\mathbf{x}) = \|\mathbf{x}\|_1 \cdot \text{sign}(\mathbf{x})/d,$$

where the  $\text{sign}(\mathbf{x})$  takes sign of each coordinates. For scaled sign compressor, we have

$$\begin{aligned} \|\mathcal{C}_{\text{sign}}(\mathbf{x}) - \mathbf{x}\|_2^2 &= \sum_{i=1}^d \left( \frac{1}{d} \sum_{j=1}^d |x_j| - |x_i| \right)^2 \\ &= \sum_{i=1}^d \left( \frac{1}{d} \sum_{j=1}^d |x_j| \right)^2 - 2 \left( \sum_{i=1}^d |x_i| \right) \left( \frac{1}{d} \sum_{j=1}^d |x_j| \right) + \sum_{i=1}^d |x_i|^2 \\ &= d \cdot \frac{1}{d} \|\mathbf{x}\|_2^2 - \frac{2}{d} \|\mathbf{x}\|_1^2 + \|\mathbf{x}\|_2^2 \\ &= \|\mathbf{x}\|_2^2 - \frac{1}{d} \|\mathbf{x}\|_1^2 \\ &= \left(1 - \frac{\|\mathbf{x}\|_1^2}{d\|\mathbf{x}\|_2^2}\right) \|\mathbf{x}\|_2^2. \end{aligned} \quad (\text{A.2})$$

Thus we conclude that

$$\|\mathcal{C}_{\text{sign}}(\mathbf{x}) - \mathbf{x}\|_2^2 = \left(1 - \frac{\|\mathbf{x}\|_1^2}{d\|\mathbf{x}\|_2^2}\right) \|\mathbf{x}\|_2^2.$$

## B Main Theorem for Algorithm 1

We provide more following lemmas to support the proof of the main theorem.

**Lemma B.1.** For compressors  $\mathcal{C}(\cdot)$  satisfying Assumption 4.1, we have  $\mathbb{E}[\|\mathcal{C}(\mathbf{x})\|_2] \leq (1 + \sqrt{\pi})\|\mathbf{x}\|_2$ .

Lemma B.1 can be easily verified by applying triangle inequality to Assumption 4.1.

**Lemma B.2.** Under Assumptions 4.1 and 6.2, we have  $\|\nabla f(\mathbf{x})\|_2 \leq G$ ,  $\|\widehat{\mathbf{g}}_t\|_2 \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G = \widehat{G}$ ,  $\|\widetilde{\mathbf{g}}_t\|_2 \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}\widehat{G} = \widetilde{G}$ ,  $\|\mathbf{v}_t\|_2 \leq \widetilde{G}^2$  and  $\|\mathbf{m}_t\|_2 \leq \widetilde{G}$ . Additionally, we have  $\|\nabla f(\mathbf{x})\|_\infty \leq G_\infty$ ,  $\|\widehat{\mathbf{g}}_t\|_\infty \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G_\infty = \widehat{G}_\infty$ ,  $\|\widetilde{\mathbf{g}}_t\|_\infty \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}\widehat{G}_\infty = \widetilde{G}_\infty$ ,  $\|\mathbf{v}_t\|_\infty \leq \widetilde{G}_\infty^2$  and  $\|\mathbf{m}_t\|_\infty \leq \widetilde{G}_\infty$ .

Lemma B.2 shows that the compressed gradients and the update parameters can be bounded by constants, it is a quite standard result for the adam-type convergence analysis.

**Lemma B.3.** For the mini-batch stochastic gradient on each worker, its variance satisfies

$$\mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \leq \sigma^2 \frac{N - \tau}{\tau(N - 1)},$$

where  $\tau$  denotes the batch size and  $N$  denotes the total amount of data on each worker.

Lemma B.3 shows the bounded variance of local stochastic gradient with batch size  $\tau$ .

**Lemma B.4.** Let  $\beta_1, \beta_2$  be the weight parameters such that

$$\begin{aligned}\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \widetilde{\mathbf{g}}_t, \\ \mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \widetilde{\mathbf{g}}_t^2,\end{aligned}$$

set  $\alpha_t = \alpha$ ,  $t = 1, \dots, T$  be the step sizes. Under Assumption 6.2 and 6.3, we have the following two results:

$$\sum_{t=1}^T \alpha_t^2 \|\widehat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t\|_2^2 \leq \frac{32T\alpha^2 \widetilde{G}^2}{\nu} + \frac{4T\alpha^2 \sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],$$

and

$$\sum_{t=1}^T \alpha_t^2 \|\widehat{\mathbf{V}}_t^{-1/2} \widetilde{\mathbf{g}}_t\|_2^2 \leq \frac{32T\alpha^2 \widetilde{G}^2}{\nu} + \frac{4T\alpha^2 \sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],$$

Lemma B.4 provides upper bounds for the summation of  $\alpha_t^2 \|\widehat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t\|_2$  and  $\alpha_t^2 \|\widehat{\mathbf{V}}_t^{-1/2} \widetilde{\mathbf{g}}_t\|_2$ .

**Lemma B.5.** Denote  $\mathbf{g}_t = \frac{1}{n} \sum_{i=1}^n \mathbf{g}_t^{(i)}$  as the aggregated fresh gradient and denote  $\widehat{\mathbf{g}}_t = \frac{1}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_t^{(i)}$  as the aggregated worker-side compressed gradient. The compression error of the worker-side compression  $\|\widehat{\mathbf{g}}_t - \mathbf{g}_t\|_2$  satisfies

$$\mathbb{E}[\|\widehat{\mathbf{g}}_t - \mathbf{g}_t\|_2] \leq \frac{L\widetilde{G}\sqrt{\pi}}{1 - \sqrt{\pi}} \cdot \alpha_1 \sqrt{d\nu^{-1/2}} + \frac{2\sqrt{\pi}\sigma}{1 - \sqrt{\pi}} \sqrt{\frac{N - \tau}{\tau(N - 1)}}, \quad (\text{B.1})$$

where  $N$  is the total number of data samples on each worker and  $\tau$  is the batch-size.

**Lemma B.6.** The compression error of the server-side compression  $\|\widetilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\|_2$  satisfies

$$\mathbb{E}[\|\widetilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\|_2] \leq \frac{L\widetilde{G}\sqrt{\pi}(1 + \sqrt{\pi})}{(1 - \sqrt{\pi})^2} \cdot \alpha_1 \sqrt{d\nu^{-1/2}} + \frac{2\sqrt{\pi}(1 + \sqrt{\pi})\sigma}{(1 - \sqrt{\pi})^2} \sqrt{\frac{N - \tau}{\tau(N - 1)}}, \quad (\text{B.2})$$

where  $N$  is the total number of data samples on each worker and  $\tau$  is the batch-size.

Lemma B.5 and B.6 provide upper bounds for the worker-side and the server-side compression errors respectively.

Following Zhou et al. (2018), we define an auxiliary sequence  $\mathbf{z}_t$  as follows: let  $\mathbf{x}_0 = \mathbf{x}_1$ , and for each  $t \geq 1$ ,

$$\mathbf{z}_t = \mathbf{x}_t + \frac{\beta_1}{1 - \beta_1} (\mathbf{x}_t - \mathbf{x}_{t-1}). \quad (\text{B.3})$$

The following three lemmas directly follows Lemma 7.1, 7.2, and 7.3 from Zhou et al. (2018).

**Lemma B.7.** Let  $\mathbf{z}_t$  be defined in (B.3). For  $t \geq 2$ , we have

$$\mathbf{z}_{t+1} - \mathbf{z}_t = \frac{\beta_1}{1 - \beta_1} \left[ \mathbf{I} - (\alpha_t \widehat{\mathbf{V}}_t^{-1/2}) (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2})^{-1} \right] (\mathbf{x}_{t-1} - \mathbf{x}_t) - \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t, \quad (\text{B.4})$$

and

$$\mathbf{z}_{t+1} - \mathbf{z}_t = \frac{\beta_1}{1 - \beta_1} (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}) \mathbf{m}_{t-1} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t, \quad (\text{B.5})$$

Specifically, for  $t = 1$ , we have

$$\mathbf{z}_2 - \mathbf{z}_1 = -\alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1. \quad (\text{B.6})$$

**Lemma B.8.** Let  $\mathbf{z}_t$  be defined in (B.3). For  $t \geq 2$ , we have

$$\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2 \leq \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2 + \frac{\beta_1}{1 - \beta_1} \|\mathbf{x}_{t-1} - \mathbf{x}_t\|_2.$$

**Lemma B.9.** Let  $\mathbf{z}_t$  be defined in (B.3). For  $t \geq 2$ , we have

$$\|\nabla f(\mathbf{z}_t) - \nabla f(\mathbf{x}_t)\|_2 \leq L \left( \frac{\beta_1}{1 - \beta_1} \right) \cdot \|\mathbf{x}_{t-1} - \mathbf{x}_t\|_2.$$

Together with previous Lemmas, now we are ready to prove Theorem 6.4.

*Proof of Theorem 6.4.* Since  $f$  is  $L$ -smooth, we have:

$$\begin{aligned} f(\mathbf{z}_{t+1}) &\leq f(\mathbf{z}_t) + \nabla f(\mathbf{z}_t)^\top (\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{L}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 \\ &= f(\mathbf{z}_t) + \nabla f(\mathbf{x}_t)^\top (\mathbf{z}_{t+1} - \mathbf{z}_t) + (\nabla f(\mathbf{z}_t) - \nabla f(\mathbf{x}_t))^\top (\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{L}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2. \end{aligned} \quad (\text{B.7})$$

When  $t = 1$ , for (B.7) we have

$$\begin{aligned} f(\mathbf{z}_2) &\leq f(\mathbf{z}_1) + \nabla f(\mathbf{x}_1)^\top (\mathbf{z}_2 - \mathbf{z}_1) + (\nabla f(\mathbf{z}_1) - \nabla f(\mathbf{x}_1))^\top (\mathbf{z}_2 - \mathbf{z}_1) + \frac{L}{2} \|\mathbf{z}_2 - \mathbf{z}_1\|_2^2 \\ &= f(\mathbf{z}_1) - \nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1 + 2L \|\alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1\|_2^2. \end{aligned} \quad (\text{B.8})$$

When  $t \geq 2$ , we have

$$\begin{aligned} \nabla f(\mathbf{x}_t)^\top (\mathbf{z}_{t+1} - \mathbf{z}_t) &= \nabla f(\mathbf{x}_t)^\top \left[ \frac{\beta_1}{1 - \beta_1} (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}) \mathbf{m}_{t-1} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t \right] \\ &= \frac{\beta_1}{1 - \beta_1} \nabla f(\mathbf{x}_t)^\top (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}) \mathbf{m}_{t-1} - \nabla f(\mathbf{x}_t)^\top \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t \\ &= \frac{\beta_1}{1 - \beta_1} \nabla f(\mathbf{x}_t)^\top (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}) \mathbf{m}_{t-1} + \nabla f(\mathbf{x}_t)^\top (\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}) \tilde{\mathbf{g}}_t \\ &\quad - \nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \tilde{\mathbf{g}}_t \\ &\leq \frac{1}{1 - \beta_1} \|\nabla f(\mathbf{x}_t)\|_2 \cdot \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}\|_{\infty, \infty} \cdot \|\mathbf{m}_{t-1}\|_2 - \nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \tilde{\mathbf{g}}_t \\ &\leq \frac{1}{1 - \beta_1} G \widetilde{G} \left[ \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2}\|_{\infty, \infty} - \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2}\|_{\infty, \infty} \right] - \nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \tilde{\mathbf{g}}_t, \end{aligned} \quad (\text{B.9})$$

where the first equality holds due to (B.5) in Lemma B.7. The first inequality holds because for vector  $\mathbf{x}$ ,  $\mathbf{y}$  and positive diagonal matrix  $\mathbf{A}$ , we have  $\mathbf{x}^\top \mathbf{A} \mathbf{y} \leq \|\mathbf{x}\|_2 \cdot \|\mathbf{A}\|_{\infty, \infty} \cdot \|\mathbf{y}\|_2$ . The second inequality holds due to Lemma B.7 and  $\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \succeq \alpha_t \widehat{\mathbf{V}}_t^{-1/2} \succeq 0$ . Thus for diagonal matrices  $\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2}$  and  $\alpha_t \widehat{\mathbf{V}}_t^{-1/2}$ , we directly conclude  $\|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} - \alpha_t \widehat{\mathbf{V}}_t^{-1/2}\|_{\infty, \infty} = \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2}\|_{\infty, \infty} - \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2}\|_{\infty, \infty}$ .

Next we bound the last two terms in (B.7).

$$\begin{aligned}
 & (\nabla f(\mathbf{z}_t) - \nabla f(\mathbf{x}_t))^\top (\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{L}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 \\
 & \leq \|\nabla f(\mathbf{z}_t) - \nabla f(\mathbf{x}_t)\|_2 \cdot \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2 + \frac{L}{2} \|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 \\
 & \leq L \left( \frac{\beta_1}{1 - \beta_1} \right) \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2 \cdot \left[ \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2 + \frac{\beta_1}{1 - \beta_1} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2 \right] + \frac{L}{2} \left[ \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2 + \frac{\beta_1}{1 - \beta_1} \|\mathbf{x}_{t-1} - \mathbf{x}_t\|_2 \right]^2 \\
 & \leq \frac{L}{2} \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + \frac{3L}{2} \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 + L \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\mathbf{x}_{t-1} - \mathbf{x}_t\|_2^2 \\
 & \leq 2L \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + 3L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2,
 \end{aligned} \tag{B.10}$$

where the second inequality holds because of Lemma B.8 and B.9, the third inequality holds by applying Young's inequality to the first term and applying Cauchy inequality to the second term.

For  $t = 1$ , recalling (B.8), taking expectation and rearranging terms, we have

$$\begin{aligned}
 & \mathbb{E}[f(\mathbf{z}_2) - f(\mathbf{z}_1)] \\
 & \leq \mathbb{E} \left[ -\nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1 + 2L \|\alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1\|_2^2 \right] \\
 & = \mathbb{E} \left[ -\nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \mathbf{g}_1 - \nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} (\tilde{\mathbf{g}}_1 - \mathbf{g}_1) - \nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} (\tilde{\mathbf{g}}_1 - \widehat{\mathbf{g}}_1) + 2L \|\alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1\|_2^2 \right] \\
 & \leq \mathbb{E} \left[ -\alpha_1 \|\nabla f(\mathbf{x}_1)\|_2^2 \cdot (\tilde{G}_\infty^2 + \nu)^{-1/2} - \nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} (\tilde{\mathbf{g}}_1 - \mathbf{g}_1) - \nabla f(\mathbf{x}_1)^\top \alpha_1 \widehat{\mathbf{V}}_1^{-1/2} (\tilde{\mathbf{g}}_1 - \widehat{\mathbf{g}}_1) \right. \\
 & \quad \left. + 2L \|\alpha_1 \widehat{\mathbf{V}}_1^{-1/2} \tilde{\mathbf{g}}_1\|_2^2 \right],
 \end{aligned} \tag{B.11}$$

where the first term of the last inequality holds because

$$\begin{aligned}
 \mathbb{E}[-\alpha_1 \nabla f(\mathbf{x}_1)^\top \widehat{\mathbf{V}}_1^{-1/2} \mathbf{g}_1] &= \mathbb{E}[-\alpha_1 \nabla f(\mathbf{x}_1)^\top \widehat{\mathbf{V}}_1^{-1/2} \nabla f(\mathbf{x}_1)] \\
 &\leq -\alpha_1 (\tilde{G}_\infty^2 + \nu)^{-1/2} \mathbb{E}[\|\nabla f(\mathbf{x}_1)\|_2^2].
 \end{aligned}$$

For  $t \geq 2$ , substituting (B.9) and (B.10) into (B.7), taking expectation and rearranging terms, we have

$$\begin{aligned}
 & \mathbb{E} \left[ f(\mathbf{z}_{t+1}) + \frac{G \tilde{G} \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2}\|_{\infty, \infty}}{1 - \beta_1} - \left( f(\mathbf{z}_t) + \frac{G \tilde{G} \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2}\|_{\infty, \infty}}{1 - \beta_1} \right) \right] \\
 & \leq \mathbb{E} \left[ -\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \tilde{\mathbf{g}}_t + 2L \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + 3L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 \right] \\
 & = \mathbb{E} \left[ -\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{g}_t - \nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} (\tilde{\mathbf{g}}_t - \mathbf{g}_t) - \nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} (\tilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t) \right. \\
 & \quad \left. + 2L \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + 3L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{m}_{t-1}\|_2^2 \right] \\
 & \leq \mathbb{E} \left[ -\alpha_{t-1} \|\nabla f(\mathbf{x}_t)\|_2^2 (\tilde{G}_\infty^2 + \nu)^{-1/2} + 2L \|\alpha_t \widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2 + 3L \left( \frac{\beta_1}{1 - \beta_1} \right)^2 \|\alpha_{t-1} \widehat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{m}_{t-1}\|_2^2 \right. \\
 & \quad \left. + \alpha_{t-1} \nu^{-1/2} G \|\widehat{\mathbf{g}}_t - \mathbf{g}_t\|_2 + \alpha_{t-1} \nu^{-1/2} G \|\tilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\|_2 \right],
 \end{aligned} \tag{B.12}$$

where the equality holds because  $\tilde{\mathbf{g}}_t = \mathbf{g}_t + (\hat{\mathbf{g}}_t - \mathbf{g}_t) + (\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t)$ . The second inequality holds because of

$$\begin{aligned}\mathbb{E}[-\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{g}_t] &= \mathbb{E}[-\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} \mathbb{E}[\mathbf{g}_t]] \\ &= \mathbb{E}[-\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} \nabla f(\mathbf{x}_t)] \\ &\leq -\alpha_{t-1} (\tilde{G}_\infty^2 + \nu)^{-1/2} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],\end{aligned}\tag{B.13}$$

$$\begin{aligned}\mathbb{E}[-\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} (\hat{\mathbf{g}}_t - \mathbf{g}_t)] &\leq \alpha_{t-1} \nu^{-1/2} \|\nabla f(\mathbf{x}_t)\|_2 \cdot \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] \\ &\leq \alpha_{t-1} \nu^{-1/2} G \cdot \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2],\end{aligned}\tag{B.14}$$

$$\begin{aligned}\mathbb{E}[-\nabla f(\mathbf{x}_t)^\top \alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} (\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t)] &\leq \alpha_{t-1} \nu^{-1/2} \|\nabla f(\mathbf{x}_t)\|_2 \cdot \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2] \\ &\leq \alpha_{t-1} \nu^{-1/2} G \cdot \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2].\end{aligned}\tag{B.15}$$

Summing up (B.12) for  $t = 2$  to  $T$ , rearranging and adding with (B.11), we have

$$\begin{aligned}&(\tilde{G}_\infty^2 + \nu)^{-1/2} \sum_{t=1}^T \alpha_t \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\ &\leq \mathbb{E}\left[f(\mathbf{z}_1) + \frac{G\tilde{G}\|\alpha_1 \hat{\mathbf{V}}_1^{-1/2}\|_{\infty,\infty}}{1-\beta_1} - \left(f(\mathbf{z}_{T+1}) + \frac{G\tilde{G}\|\alpha_T \hat{\mathbf{V}}_T^{-1/2}\|_{\infty,\infty}}{1-\beta_1}\right)\right] \\ &\quad + 2L \sum_{t=1}^T \mathbb{E}[\|\alpha_t \hat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2] + 3L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \sum_{t=1}^T \mathbb{E}[\|\alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{m}_{t-1}\|_2^2] \\ &\quad + \alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] + \alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2] \\ &\leq \mathbb{E}\left[\Delta f + \frac{\alpha_1 G\tilde{G}\nu^{-1/2}}{1-\beta_1}\right] + 2L \sum_{t=1}^T \mathbb{E}[\|\alpha_t \hat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2] + 3L \left(\frac{\beta_1}{1-\beta_1}\right)^2 \sum_{t=1}^T \mathbb{E}[\|\alpha_{t-1} \hat{\mathbf{V}}_{t-1}^{-1/2} \mathbf{m}_{t-1}\|_2^2] \\ &\quad + \alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] + \alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2],\end{aligned}\tag{B.16}$$

where  $\Delta f = f(\mathbf{x}_1) - \inf_x f(\mathbf{x})$ .

By Lemma B.4, with the constant stepsize  $\alpha_t = \alpha$ , we have

$$\sum_{t=1}^T \alpha_t^2 \mathbb{E}[\|\hat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t\|_2^2] \leq \frac{32T\alpha^2\tilde{G}^2}{\nu} + \frac{4T\alpha^2\sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],\tag{B.17}$$

and

$$\sum_{t=1}^T \alpha_t^2 \mathbb{E}[\|\hat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2] \leq \frac{32T\alpha^2\tilde{G}^2}{\nu} + \frac{4T\alpha^2\sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2].\tag{B.18}$$

By Lemma B.5 and B.6, we conclude the last two terms of (B.16) as

$$\begin{aligned}&\alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] + \alpha_{t-1} \nu^{-1/2} G \cdot \sum_{t=1}^T \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2] \\ &\leq \frac{\sqrt{\pi}}{1-\sqrt{\pi}} LTG\tilde{G}\alpha_1\alpha_{t-1}\sqrt{d}\nu^{-1} + \frac{\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} LTG\tilde{G}\alpha_1\alpha_{t-1}\sqrt{d}\nu^{-1} \\ &\quad + \frac{2\sigma\sqrt{\pi}}{1-\sqrt{\pi}} TG\alpha_{t-1}\nu^{-1/2} \sqrt{\frac{N-\tau}{\tau(N-1)}} + \frac{2\sigma\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} TG\alpha_{t-1}\nu^{-1/2} \sqrt{\frac{N-\tau}{\tau(N-1)}} \\ &\leq \frac{2\sqrt{\pi}}{\nu(1-\sqrt{\pi})^2} LG\tilde{G}T\alpha_1\alpha_{t-1}\sqrt{d} + \frac{4\sqrt{\pi}}{\sqrt{\nu}(1-\sqrt{\pi})^2} GT\alpha_{t-1} \cdot \sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}.\end{aligned}\tag{B.19}$$

Substituting (B.17), (B.18) and (B.19) into (B.16), set the constant stepsize  $\alpha_t = \alpha$  and rearranging, we have

$$\begin{aligned}
 & (\tilde{G}_\infty^2 + \nu)^{-1/2} \sum_{t=1}^T \alpha \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\
 & \leq \mathbb{E}\left[\Delta f + \frac{\alpha G \tilde{G} \nu^{-1/2}}{1 - \beta_1}\right] + \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{32T\alpha^2 \tilde{G}^2}{\nu} \\
 & \quad + \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{4T\alpha^2 \sigma^2}{\nu n} + \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\
 & \quad + \frac{2\sqrt{\pi}}{\nu(1 - \sqrt{\pi})^2} L G \tilde{G} T \alpha^2 \sqrt{d} + \frac{4\sqrt{\pi}}{\sqrt{\nu}(1 - \sqrt{\pi})^2} G T \alpha \cdot \sigma \sqrt{\frac{N - \tau}{\tau(N - 1)}}. \tag{B.20}
 \end{aligned}$$

Hence we obtain

$$\begin{aligned}
 & \left[(\alpha(\tilde{G}_\infty^2 + \nu)^{-1/2} - \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{4\alpha^2}{\nu}\right] \sum_{t=1}^T \alpha \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\
 & \leq \mathbb{E}\left[\Delta f + \frac{\alpha G \tilde{G} \nu^{-1/2}}{1 - \beta_1}\right] + \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{32T\alpha^2 \tilde{G}^2}{\nu} + \left[2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2\right] \frac{4T\alpha^2 \sigma^2}{\nu n} \\
 & \quad + \frac{2\sqrt{\pi}}{\nu(1 - \sqrt{\pi})^2} L G \tilde{G} T \alpha^2 \sqrt{d} + \frac{4\sqrt{\pi}}{\sqrt{\nu}(1 - \sqrt{\pi})^2} G T \alpha \cdot \sigma \sqrt{\frac{N - \tau}{\tau(N - 1)}}. \tag{B.21}
 \end{aligned}$$

Let  $[2L + 3L(\frac{\beta_1}{1 - \beta_1})^2] \frac{4\alpha^2}{\nu} \leq \frac{\alpha}{2(\tilde{G}_\infty^2 + \nu)^{1/2}}$ , we obtain the condition on learning rate  $\alpha \leq \frac{\nu}{8(\tilde{G}_\infty^2 + \nu)^{1/2}[2L + 3L(\frac{\beta_1}{1 - \beta_1})^2]} = \frac{\nu}{4CC_1}$ . Then we get

$$\min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \leq \frac{M_1}{T\alpha} + \frac{M_2}{T} + \alpha M_3 + \frac{\alpha M_4 \sigma^2}{n} + M_5 \sigma \sqrt{\frac{N - \tau}{\tau(N - 1)}}, \tag{B.22}$$

where

$$\begin{aligned}
 M_1 &= C \cdot \Delta f, \quad M_2 = \frac{CG\tilde{G}}{(1 - \beta_1)\sqrt{\nu}}, \quad M_3 = \frac{32CC_1\tilde{G}^2}{\nu} + \frac{2\sqrt{\pi}CLG\tilde{G}\sqrt{d}}{\nu(1 - \sqrt{\pi})^2}, \\
 M_4 &= \frac{4CC_1}{\nu}, \quad M_5 = \frac{4\sqrt{\pi}CG}{\sqrt{\nu}(1 - \sqrt{\pi})^2},
 \end{aligned}$$

and

$$\begin{aligned}
 \Delta f &= f(\mathbf{x}_1) - \inf_x f(\mathbf{x}), \quad \tilde{G} = C_2 G, \quad \tilde{G}_\infty = C_2 G_\infty, \\
 C &= 2(\tilde{G}_\infty^2 + \nu)^{1/2}, \quad C_1 = 2L + 3L\left(\frac{\beta_1}{1 - \beta_1}\right)^2, \quad C_2 = \frac{(1 + \sqrt{\pi})^2}{(1 - \sqrt{\pi})^2}.
 \end{aligned}$$

After  $T$  iterations, it requires

$$\frac{M_1}{T\alpha} + \alpha \left(M_3 + \frac{M_4 \sigma^2}{n}\right) \leq \frac{\epsilon}{3}, \quad \frac{M_2}{T} \leq \frac{\epsilon}{3}, \quad M_5 \sigma \sqrt{\frac{N - \tau}{\tau(N - 1)}} \leq \frac{\epsilon}{3}.$$

Note that  $\frac{n\epsilon}{6nM_3 + 6M_4\sigma^2} \leq \frac{\epsilon\nu}{32CC_1\tilde{G}^2} \leq \frac{\nu}{4CC_1}$ , therefore, we get

$$\alpha \leq \frac{n\epsilon}{6nM_3 + 6M_4\sigma^2}, \quad \tau \geq \left\lceil \frac{N(3M_5\sigma)^2}{(N - 1)\epsilon^2 + (3M_5\sigma)^2} \right\rceil, \quad T \geq \left\lceil \frac{36M_1M_3}{\epsilon^2} + \frac{36M_1M_4\sigma^2}{n\epsilon^2} + \frac{3M_2}{\epsilon} \right\rceil. \tag{B.23}$$

This concludes the proof. □

## C Supporting Lemmas for Algorithm 1

*Proof of Lemma B.2.* Since by Assumption 6.1,  $f$  has  $G$ -bounded stochastic gradient, for any  $\mathbf{x}$  and  $\xi$ , there is  $\|\nabla f(\mathbf{x}; \xi)\|_2 \leq G$ . We have

$$\|\nabla f(\mathbf{x})\|_2 = \|\mathbb{E}_\xi \nabla f(\mathbf{x}; \xi)\|_2 \leq \mathbb{E}_\xi \|\nabla f(\mathbf{x}; \xi)\|_2 \leq G.$$

First we prove the upper bound for compressed gradient on  $i$ -th worker by induction. For  $t = 1$ , it is obvious that  $\|\widehat{\mathbf{g}}_1^{(i)}\|_2 \leq \|\mathcal{C}(\mathbf{g}_1^{(i)})\|_2 \leq \|\mathbf{g}_1^{(i)}\|_2 \leq G$ . For  $t > 1$ , suppose we have  $\|\widehat{\mathbf{g}}_t^{(i)}\|_2 \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G = \widehat{G}$ , our goal is to show that  $\|\widehat{\mathbf{g}}_{t+1}^{(i)}\|_2 \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G$ :

$$\begin{aligned} \|\widehat{\mathbf{g}}_{t+1}^{(i)}\|_2 &= \|\widehat{\mathbf{g}}_t^{(i)} + \mathcal{C}(\mathbf{g}_{t+1}^{(i)} - \widehat{\mathbf{g}}_t^{(i)})\|_2 \\ &= \|\widehat{\mathbf{g}}_t^{(i)} + \mathcal{C}(\mathbf{g}_{t+1}^{(i)} - \widehat{\mathbf{g}}_t^{(i)}) - \mathbf{g}_{t+1}^{(i)} + \mathbf{g}_{t+1}^{(i)}\|_2 \\ &\leq \|\widehat{\mathbf{g}}_t^{(i)} + \mathcal{C}(\mathbf{g}_{t+1}^{(i)} - \widehat{\mathbf{g}}_t^{(i)}) - \mathbf{g}_{t+1}^{(i)}\|_2 + \|\mathbf{g}_{t+1}^{(i)}\|_2 \\ &\leq \sqrt{\pi}\|\mathbf{g}_{t+1}^{(i)} - \widehat{\mathbf{g}}_t^{(i)}\|_2 + \|\mathbf{g}_{t+1}^{(i)}\|_2 \\ &\leq \sqrt{\pi}(G + \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G) + G \\ &\leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G := \widehat{G}, \end{aligned} \tag{C.1}$$

where the first inequality holds due to triangle inequality and the second inequality holds by Assumption 4.1.

Then we bound the aggregated compressed gradient  $\widehat{\mathbf{g}}_t$ :

$$\|\widehat{\mathbf{g}}_t\|_2 = \left\| \frac{1}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_t^{(i)} \right\|_2 \leq \frac{1}{n} \sum_{i=1}^n \|\widehat{\mathbf{g}}_t^{(i)}\|_2 \leq \widehat{G}. \tag{C.2}$$

Then we prove the upper bound for compressed aggregated gradient  $\widetilde{\mathbf{g}}$  on the server by induction. Similarly, for  $t = 1$ , we have  $\|\widetilde{\mathbf{g}}_1\|_2 \leq \|\mathcal{C}(\widehat{\mathbf{g}}_1)\|_2 \leq \|\widehat{\mathbf{g}}_1\|_2 \leq \widehat{G}$ . For  $t > 1$ , suppose we have  $\|\widetilde{\mathbf{g}}_t\|_2 \leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}\widehat{G}$ , then we have

$$\begin{aligned} \|\widetilde{\mathbf{g}}_{t+1}\|_2 &= \|\widetilde{\mathbf{g}}_t + \mathcal{C}(\widehat{\mathbf{g}}_{t+1} - \widetilde{\mathbf{g}}_t)\|_2 \\ &\leq \|\widetilde{\mathbf{g}}_t + \mathcal{C}(\widehat{\mathbf{g}}_{t+1} - \widetilde{\mathbf{g}}_t) - \widehat{\mathbf{g}}_{t+1}\|_2 + \|\widehat{\mathbf{g}}_{t+1}\|_2 \\ &\leq \sqrt{\pi}\|\widehat{\mathbf{g}}_{t+1} - \widetilde{\mathbf{g}}_t\|_2 + \|\widehat{\mathbf{g}}_{t+1}\|_2 \\ &\leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}\widehat{G} := \widetilde{G}. \end{aligned} \tag{C.3}$$

Next we bound  $\|\mathbf{m}_t\|_2$  by induction. We prove this by induction. For  $t = 0$ , we have  $\|\mathbf{m}_0\|_2 = 0 \leq \widetilde{G}$ . For  $t \geq 1$ , suppose that  $\|\mathbf{m}_t\|_2 \leq \widetilde{G}$ , then for  $\mathbf{m}_{t+1}$ , we have

$$\begin{aligned} \|\mathbf{m}_{t+1}\|_2 &= \|\beta_1 \mathbf{m}_t + (1 - \beta_1) \widetilde{\mathbf{g}}_{t+1}\|_2 \\ &\leq \beta_1 \|\mathbf{m}_t\|_2 + (1 - \beta_1) \|\widetilde{\mathbf{g}}_{t+1}\|_2 \\ &\leq \beta_1 \widetilde{G} + (1 - \beta_1) \widetilde{G} \\ &= \widetilde{G}. \end{aligned}$$

Thus, for any  $t \geq 0$ , we have  $\|\mathbf{m}_t\|_2 \leq \widetilde{G}$ .

Finally we bound  $\|\widehat{\mathbf{v}}_t\|_2$  again by induction. For  $t = 1$ , we have  $\|\mathbf{v}_0\|_2 = \|\widehat{\mathbf{v}}_0\|_2 = 0 \leq \widetilde{G}^2$ . For  $t > 1$ , suppose we

have  $\|\hat{\mathbf{v}}_t\|_2 \leq \tilde{G}^2$ , then for  $\mathbf{v}_{t+1}$ ,

$$\begin{aligned}\|\mathbf{v}_{t+1}\|_2 &= \|\beta_2 \mathbf{v}_t + (1 - \beta_2) \tilde{\mathbf{g}}_{t+1}^2\|_2 \\ &\leq \beta_2 \|\mathbf{v}_t\|_2 + (1 - \beta_2) \|\tilde{\mathbf{g}}_{t+1}^2\|_2 \\ &\leq \beta_2 \tilde{G}^2 + (1 - \beta_2) \tilde{G}^2 \\ &= \tilde{G}^2,\end{aligned}$$

and by definition, we have  $\|\hat{\mathbf{v}}_{t+1}\|_2 = \max\{\|\hat{\mathbf{v}}_t\|_2, \|\mathbf{v}_{t+1}\|_2\} \leq \tilde{G}^2$ . Thus, for any  $t \geq 0$ , we have  $\|\hat{\mathbf{v}}_t\|_2 \leq \tilde{G}^2$ . The proof of  $\ell_\infty$ -boundedness is similar to the proof of  $\ell_2$ -boundedness. This concludes the proof.  $\square$

*Proof of Lemma B.3.* By definition, we have

$$\mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2] = \mathbb{E}\left[\left\|\frac{1}{\tau} \sum_{j \in \mathcal{S}_\tau^i} \mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\right\|_2^2\right],$$

where  $\mathcal{S}_\tau^i$  is a set of batch index with  $\tau$  samples on the  $i$ -th worker, then for  $\forall j \in \mathcal{S}_\tau^i$ , we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\frac{1}{\tau} \sum_{j \in \mathcal{S}_\tau^i} \mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\right\|_2^2\right] \\ &= \frac{1}{\tau^2} \mathbb{E}\left[\sum_{j,j' \in \mathcal{S}_\tau^i} [\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)]^\top [\mathbf{g}_t^{(i,j')} - \nabla f_i(\mathbf{x}_t)]\right] \\ &= \frac{1}{\tau^2} \mathbb{E}\left[\sum_{j \neq j' \in \mathcal{S}_\tau^i} [\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)]^\top [\mathbf{g}_t^{(i,j')} - \nabla f_i(\mathbf{x}_t)]\right] + \frac{1}{\tau} \mathbb{E}[\|\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\ &= \frac{\tau - 1}{\tau N(N - 1)} \left[ \sum_{j \neq j' \in [N]} [\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)]^\top [\mathbf{g}_t^{(i,j')} - \nabla f_i(\mathbf{x}_t)] \right] + \frac{1}{\tau} \mathbb{E}[\|\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\ &= \frac{\tau - 1}{\tau N(N - 1)} \left[ \sum_{j,j' \in [N]} [\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)]^\top [\mathbf{g}_t^{(i,j')} - \nabla f_i(\mathbf{x}_t)] \right] \\ &\quad - \frac{\tau - 1}{\tau(N - 1)} \mathbb{E}[\|\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\|_2^2] + \frac{1}{\tau} \mathbb{E}[\|\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\ &= \frac{N - \tau}{\tau(N - 1)} \mathbb{E}[\|\mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\ &\leq \frac{N - \tau}{\tau(N - 1)} \sigma^2,\end{aligned}\tag{C.4}$$

where the second equality holds by the sampling strategy, the third one follows by  $\mathbb{P}\{j, j' \in \mathcal{S}_\tau^i\} = \frac{\tau(\tau-1)}{N(N-1)}$ . The inequality holds due to Assumption 6.3. Therefore, we have

$$\mathbb{E}\left[\left\|\frac{1}{\tau} \sum_{j \in \mathcal{S}_\tau^i} \mathbf{g}_t^{(i,j)} - \nabla f_i(\mathbf{x}_t)\right\|_2^2\right] \leq \frac{N - \tau}{\tau(N - 1)} \sigma^2.$$

This concludes the proof.  $\square$

*Proof of Lemma B.4.* For the squared norm of compressed gradient, we have

$$\begin{aligned}\mathbb{E}[\|\tilde{\mathbf{g}}_t\|_2^2] &= \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t + \hat{\mathbf{g}}_t - \mathbf{g}_t + \mathbf{g}_t - \nabla f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)\|_2^2] \\ &\leq 4\mathbb{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2] + 4\mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2^2] + 4\mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2^2] + 4\mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2].\end{aligned}\tag{C.5}$$

For the stochastic variance, we have

$$\begin{aligned}
 \mathbb{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|_2^2] &= \mathbb{E}\left[\left\|\frac{1}{n} \sum_{i=1}^n [\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)]\right\|_2^2\right] \\
 &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\
 &= \frac{1}{n} \mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2] \\
 &\leq \frac{\sigma^2(N-\tau)}{n\tau(N-1)} \\
 &\leq \frac{\sigma^2}{n},
 \end{aligned} \tag{C.6}$$

where the first inequality holds by Lemma B.3, and the second one holds by  $1 \leq \tau \leq N$ . For the squared norm of gradient compression error, we have

$$\begin{aligned}
 \mathbb{E}[\|\widehat{\mathbf{g}}_t - \mathbf{g}_t\|_2^2] &\leq \mathbb{E}[\|\widehat{\mathbf{g}}_t\|_2^2] + \mathbb{E}[\|\mathbf{g}_t\|_2^2] \\
 &\leq 2\widehat{G}^2 + 2G^2,
 \end{aligned} \tag{C.7}$$

where the second inequality follows by Lemma B.2. Similarly, for the squared norm of gradient compression error on the server side, we have

$$\begin{aligned}
 \mathbb{E}[\|\widetilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\|_2^2] &\leq \mathbb{E}[\|\widetilde{\mathbf{g}}_t\|_2^2] + \mathbb{E}[\|\widehat{\mathbf{g}}_t\|_2^2] \\
 &\leq 2\widetilde{G}^2 + 2\widehat{G}^2,
 \end{aligned} \tag{C.8}$$

where the second inequality follows by Lemma B.2. Summing over  $t = 1, \dots, T$ , we have

$$\begin{aligned}
 \sum_{t=1}^T \mathbb{E}[\|\widetilde{\mathbf{g}}_t\|_2^2] &\leq 4T(2\widehat{G}^2 + 2\widetilde{G}^2) + 4T(2\widehat{G}^2 + 2G^2) + \frac{4T\sigma^2}{n} + 4 \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\
 &\leq 32T\widetilde{G}^2 + \frac{4T\sigma^2}{n} + 4 \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],
 \end{aligned} \tag{C.9}$$

where the second inequality follows by  $G \leq \widehat{G} \leq \widetilde{G}$ . By the updating rule, we have

$$\begin{aligned}
 \mathbb{E}[\|\mathbf{m}_t\|^2] &= \mathbb{E}\left[\|(1-\beta_1) \sum_{\tau=1}^t \beta_1^{t-\tau} \widetilde{\mathbf{g}}_\tau\|^2\right] \\
 &\leq (1-\beta_1)^2 \sum_{i=1}^d \mathbb{E}\left[\left(\sum_{\tau=1}^t \beta_1^{t-\tau} \widetilde{\mathbf{g}}_{\tau,i}\right)^2\right] \\
 &\leq (1-\beta_1)^2 \sum_{i=1}^d \mathbb{E}\left[\left(\sum_{\tau=1}^t \beta_1^{t-\tau}\right) \left(\sum_{\tau=1}^t \beta_1^{t-\tau} \widetilde{\mathbf{g}}_{\tau,i}^2\right)\right] \\
 &\leq (1-\beta_1) \sum_{\tau=1}^t \beta_1^{t-\tau} \mathbb{E}[\|\widetilde{\mathbf{g}}_\tau\|^2] \\
 &\leq 32\widetilde{G}^2 + \frac{4\sigma^2}{n} + 4(1-\beta_1) \sum_{\tau=1}^t \beta_1^{t-\tau} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2],
 \end{aligned} \tag{C.10}$$

where the second inequality holds by applying Cauchy-Schwarz inequality, and the third inequality holds by summation of series. Hence summing over  $t = 1, \dots, T$ , we have

$$\sum_{t=1}^T \mathbb{E}[\|\mathbf{m}_t\|^2] \leq 32T\widetilde{G}^2 + \frac{4T\sigma^2}{n} + 4 \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2]. \tag{C.11}$$

Therefore, with the condition  $\alpha_t = \alpha$  in Lemma B.4, we have

$$\sum_{t=1}^T \alpha_t^2 \mathbb{E}[\|\widehat{\mathbf{V}}_t^{-1/2} \mathbf{m}_t\|_2^2] \leq \frac{\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\mathbf{m}_t\|_2^2] \leq \frac{32T\alpha^2\widetilde{G}^2}{\nu} + \frac{4T\alpha^2\sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2]. \quad (\text{C.12})$$

We also have

$$\sum_{t=1}^T \alpha_t^2 \mathbb{E}[\|\widehat{\mathbf{V}}_t^{-1/2} \tilde{\mathbf{g}}_t\|_2^2] \leq \frac{\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\tilde{\mathbf{g}}_t\|_2^2] \leq \frac{32T\alpha^2\widetilde{G}^2}{\nu} + \frac{4T\alpha^2\sigma^2}{\nu n} + \frac{4\alpha^2}{\nu} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2]. \quad (\text{C.13})$$

□

*Proof of Lemma B.5.* Considering on the  $i$ -th worker, we have

$$\begin{aligned} \mathbb{E}[\|\widehat{\mathbf{g}}_t^{(i)} - \mathbf{g}_t^{(i)}\|_2] &= \mathbb{E}[\|\widehat{\mathbf{g}}_{t-1}^{(i)} + \mathcal{C}(\mathbf{g}_t^{(i)} - \widehat{\mathbf{g}}_{t-1}^{(i)}) - \mathbf{g}_t^{(i)}\|_2] \\ &\leq \sqrt{\pi} \mathbb{E}[\|\mathbf{g}_t^{(i)} - \widehat{\mathbf{g}}_{t-1}^{(i)}\|_2] \\ &\leq \sqrt{\pi} \mathbb{E}[\|\widehat{\mathbf{g}}_{t-1}^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] + \sqrt{\pi} \mathbb{E}[\|\mathbf{g}_t^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2], \end{aligned} \quad (\text{C.14})$$

where the first inequality holds because of Assumption 4.1. For the compression error between the local stochastic gradient and the gradient of loss function, by Lemma B.3 we have

$$\mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2] = \frac{N-\tau}{\tau(N-1)} \sigma^2.$$

Since for all variables  $\mathbf{X}$ , there is  $[\mathbb{E}\mathbf{X}]^2 \leq \mathbb{E}\mathbf{X}^2$ , we have

$$\mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2] \leq [\mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2^2]]^{\frac{1}{2}} \leq \sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}.$$

Then the expectation of gradient compression error on all workers is

$$\begin{aligned} \mathbb{E}[\|\widehat{\mathbf{g}}_t - \mathbf{g}_t\|_2] &= \mathbb{E}\left[\left\|\frac{1}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_t^{(i)} - \frac{1}{n} \sum_{i=1}^n \mathbf{g}_t^{(i)}\right\|_2\right] \\ &\leq \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\xi^{(i)}}[\|\widehat{\mathbf{g}}_t^{(i)} - \mathbf{g}_t^{(i)}\|_2] \\ &\leq \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\xi^{(i)}}[\|\widehat{\mathbf{g}}_{t-1}^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] + \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\xi^{(i)}}[\|\mathbf{g}_t^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] \\ &\leq \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\widehat{\mathbf{g}}_{t-1}^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] + \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\nabla f_i(\mathbf{x}_t) - \nabla f_i(\mathbf{x}_{t-1})\|_2] \\ &\quad + \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_t^{(i)} - \nabla f_i(\mathbf{x}_t)\|_2] + \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_{t-1}^{(i)} - \nabla f_i(\mathbf{x}_{t-1})\|_2] \\ &\leq \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\widehat{\mathbf{g}}_{t-1}^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] + \sqrt{\pi} L \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2 + 2\sqrt{\pi} \sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}, \end{aligned} \quad (\text{C.15})$$

where the first inequality holds by extended triangle inequality, the second inequality holds by applying (C.14)

and the last inequality holds by  $L$ -smoothness. Telescoping the right hand side of (C.15), we have

$$\begin{aligned}
 \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] &\leq \sqrt{\pi} \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\hat{\mathbf{g}}_{t-1}^{(i)} - \mathbf{g}_{t-1}^{(i)}\|_2] + \sqrt{\pi} L \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2 + 2\sqrt{\pi} \sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \\
 &\leq L \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \|\mathbf{x}_j - \mathbf{x}_{j-1}\|_2 + 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &= L \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \|\alpha_{j-1} \hat{\mathbf{V}}_{j-1}^{-1/2} \mathbf{m}_{j-1}\|_2 + 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &\leq L \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \|\alpha_{j-1} \hat{\mathbf{V}}_{j-1}^{-1/2}\|_2 \|\mathbf{m}_{j-1}\|_2 + 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &\leq L \tilde{G} \sum_{j=2}^t (\sqrt{\pi})^{t+1-j} \|\alpha_{j-1} \hat{\mathbf{V}}_{j-1}^{-1/2}\|_2 + 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j}, \tag{C.16}
 \end{aligned}$$

where the second inequality holds by  $L$ -smooth and the recursion (C.15). The third inequality holds by taking iteration of  $\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|$ . The forth inequality holds by the update rule of  $\mathbf{x}$  and for matrix  $\mathbf{A}$  and vector  $\mathbf{x}$ , we have  $\|\mathbf{Ax}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2$ . The last inequality holds because we set the initial momentum  $\mathbf{m}_0 = 0$  and by Lemma B.2, there is an upper bound  $\|\mathbf{m}_t\|_2 \leq \tilde{G}$ .

Since  $\alpha_{j-1} \hat{\mathbf{V}}_{j-1}^{-1/2} \succeq \alpha_j \hat{\mathbf{V}}_j^{-1/2} \succeq 0$ , we have

$$\begin{aligned}
 \mathbb{E}[\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_2] &\leq L \tilde{G} \sum_{j=2}^t (\sqrt{\pi})^{t+1-j} \|\alpha_1 \hat{\mathbf{V}}_1^{-1/2}\|_2 + 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &\leq \frac{\sqrt{\pi}}{1-\sqrt{\pi}} L \tilde{G} \alpha_1 \sqrt{d} \nu^{-1/2} + \frac{2\sigma \sqrt{\pi}}{1-\sqrt{\pi}} \sqrt{\frac{N-\tau}{n\tau(N-1)}}, \tag{C.17}
 \end{aligned}$$

where the last inequality holds because  $\|\hat{\mathbf{V}}_1^{-1/2}\|_2 \leq \sqrt{d} \nu^{-1/2}$ . This completes the proof.  $\square$

*Proof of Lemma B.6.* By Assumption 4.1, we have

$$\begin{aligned}
 \mathbb{E}[\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\|_2] &\leq \mathbb{E}[\|\tilde{\mathbf{g}}_{t-1} + \mathcal{C}(\hat{\mathbf{g}}_t - \tilde{\mathbf{g}}_{t-1}) - \hat{\mathbf{g}}_t\|_2] \\
 &\leq \sqrt{\pi} \mathbb{E}[\|\hat{\mathbf{g}}_t - \tilde{\mathbf{g}}_{t-1}\|_2] \\
 &\leq \sqrt{\pi} \mathbb{E}[\|\hat{\mathbf{g}}_{t-1} - \tilde{\mathbf{g}}_{t-1}\|_2] + \sqrt{\pi} \mathbb{E}[\|\hat{\mathbf{g}}_t - \hat{\mathbf{g}}_{t-1}\|_2] \\
 &\leq \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \cdot \mathbb{E}[\|\hat{\mathbf{g}}_j - \hat{\mathbf{g}}_{j-1}\|_2], \tag{C.18}
 \end{aligned}$$

the last inequality holds by telescoping the right hand side of the previous inequality, with the similar approach

in (C.16), we have

$$\begin{aligned}
 \mathbb{E}[\|\widehat{\mathbf{g}}_j - \widehat{\mathbf{g}}_{j-1}\|_2] &= \mathbb{E}\left[\left\|\frac{1}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_j^{(i)} - \frac{1}{n} \sum_{i=1}^n \widehat{\mathbf{g}}_{j-1}^{(i)}\right\|_2\right] \\
 &= \mathbb{E}\left[\left\|\frac{1}{n} \sum_{i=1}^n \mathcal{C}(\mathbf{g}_j^{(i)} - \widehat{\mathbf{g}}_{j-1}^{(i)})\right\|_2\right] \\
 &\leq \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\|\mathcal{C}(\mathbf{g}_j^{(i)} - \widehat{\mathbf{g}}_{j-1}^{(i)})\|_2] \\
 &\leq \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_j^{(i)} - \widehat{\mathbf{g}}_{j-1}^{(i)}\|_2] \\
 &\stackrel{(a)}{\leq} \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\widehat{\mathbf{g}}_{j-1}^{(i)} - \mathbf{g}_{j-1}^{(i)}\|_2] + \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_{j-1}^{(i)} - \mathbf{g}_j^{(i)}\|_2], \tag{C.19}
 \end{aligned}$$

where we apply Jensen's inequality to get the first inequality. The second inequality holds due to Lemma B.1, the third inequality holds by triangle inequality, then similar to previous proof, we have

$$\begin{aligned}
 \mathbb{E}[\|\widehat{\mathbf{g}}_j - \widehat{\mathbf{g}}_{j-1}\|_2] &\leq \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\widehat{\mathbf{g}}_{j-1}^{(i)} - \mathbf{g}_{j-1}^{(i)}\|_2] + \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\nabla f_i(\mathbf{x}_{j-1}) - \nabla f_i(\mathbf{x}_j)\|_2] \\
 &\quad + \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_j^{(i)} - \nabla f_i(\mathbf{x}_j)\|_2] + \frac{1+\sqrt{\pi}}{n} \sum_{i=1}^n \mathbb{E}[\|\mathbf{g}_{j-1}^{(i)} - \nabla f_i(\mathbf{x}_{j-1})\|_2] \\
 &\leq \frac{(1+\sqrt{\pi})(1-\sqrt{\pi}^j)}{1-\sqrt{\pi}} L\tilde{G}\alpha_1 \|\widehat{\mathbf{V}}_1^{-1/2}\|_2 + \frac{(1+\sqrt{\pi})(1-\sqrt{\pi}^j)}{1-\sqrt{\pi}} \cdot 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \\
 &\leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}} L\tilde{G}\alpha_1 \|\widehat{\mathbf{V}}_1^{-1/2}\|_2 + \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}} \cdot 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}, \tag{C.20}
 \end{aligned}$$

where inequalities hold by proof of Lemma B.5. Substituting (C.19) into (C.18)

$$\begin{aligned}
 \mathbb{E}[\|\widetilde{\mathbf{g}}_t - \widehat{\mathbf{g}}_t\|_2] &\leq \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}} L\tilde{G}\alpha_1 \cdot \|\widehat{\mathbf{V}}_1^{-1/2}\|_2 \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} + \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}} \cdot 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &\leq \frac{\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} L\tilde{G}\alpha_1 \|\widehat{\mathbf{V}}_1^{-1/2}\|_2 + \frac{\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} \cdot 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}} \sum_{j=1}^t (\sqrt{\pi})^{t+1-j} \\
 &\leq \frac{\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} L\tilde{G}\alpha_1 \sqrt{d\nu^{-1/2}} + \frac{\sqrt{\pi}(1+\sqrt{\pi})}{(1-\sqrt{\pi})^2} \cdot 2\sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}. \tag{C.21}
 \end{aligned}$$

where the last inequality holds because  $\|\widehat{\mathbf{V}}_1^{-1/2}\|_2 \leq \sqrt{d}\nu^{-1/2}$ . This concludes the proof.  $\square$

## D Dependency on the Compression Constant

We track the impact of the compressor related parameter  $\pi$ . Recall in Assumption 4.1, we assume we have compressors satisfy  $\mathbb{E}_{\mathcal{C}}[\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|_2^2] \leq \pi\mathbf{x}^2, \forall \mathbf{x} \in \mathbb{R}^d$ . Note that  $\pi = 0$  leads to  $\mathcal{C}(\mathbf{x}) = \mathbf{x}$ . In fact, a stronger compression means a larger  $\pi$ , i.e.,  $\pi \rightarrow 1$ , may leads to worse convergence. For our convergence bound

$$\begin{aligned}
 \min_{t \in [T]} \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] &\leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|_2^2] \\
 &\leq \frac{M_1}{T\alpha} + \frac{M_2}{T} + \alpha M_3 + \frac{\alpha M_4 \sigma^2}{n} + M_5 \sigma \sqrt{\frac{N-\tau}{\tau(N-1)}}, \tag{D.1}
 \end{aligned}$$

where

$$\begin{aligned} M_1 &= C \cdot \Delta f, \quad M_2 = \frac{CG\tilde{G}}{(1-\beta_1)\sqrt{\nu}}, \quad M_3 = \frac{32CC_1\tilde{G}^2}{\nu} + \frac{2\sqrt{\pi}CLG\tilde{G}\sqrt{d}}{\nu(1-\sqrt{\pi})^2}, \\ M_4 &= \frac{4CC_1}{\nu}, \quad M_5 = \frac{4\sqrt{\pi}CG}{\sqrt{\nu}(1-\sqrt{\pi})^2}, \end{aligned}$$

and

$$\begin{aligned} \Delta f &= f(\mathbf{x}_1) - \inf_x f(\mathbf{x}), \quad \tilde{G} = C_2 G, \quad \tilde{G}_\infty = C_2 G_\infty, \\ C &= 2(\tilde{G}_\infty^2 + \nu)^{1/2}, \quad C_1 = 2L + 3L\left(\frac{\beta_1}{1-\beta_1}\right)^2, \quad C_2 = \frac{(1+\sqrt{\pi})^2}{(1-\sqrt{\pi})^2}. \end{aligned}$$

Following by Lemma B.2, as  $\pi \rightarrow 1$ , we conclude the series of compressed gradient bounds  $\hat{G}$ ,  $\tilde{G}$ ,  $\hat{G}_\infty$ ,  $\tilde{G}_\infty$  as follows

$$\begin{aligned} \hat{G} &= \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G = O\left(\frac{G}{1-\pi}\right), \quad \hat{G}_\infty = \frac{1+\sqrt{\pi}}{1-\sqrt{\pi}}G_\infty = O\left(\frac{G_\infty}{1-\pi}\right), \\ \tilde{G} &= C_2 G = \frac{(1+\sqrt{\pi})^2}{(1-\sqrt{\pi})^2}G = O\left(\frac{G}{(1-\pi)^2}\right), \quad \tilde{G}_\infty = C_2 G_\infty = \frac{(1+\sqrt{\pi})^2}{(1-\sqrt{\pi})^2}G_\infty = O\left(\frac{G_\infty}{(1-\pi)^2}\right), \\ C &= 2(\tilde{G}_\infty^2 + \nu)^{1/2} = O(\tilde{G}_\infty) = O\left(\frac{G_\infty}{(1-\pi)^2}\right). \end{aligned}$$

Const	Order of $\pi$
$M_1$	$O(\Delta f G_\infty (1-\pi)^{-2})$
$M_2$	$O(G^2 G_\infty (1-\pi)^{-4})$
$M_3$	$O(G^2 G_\infty (1-\pi)^{-6} + L\sqrt{d}G^2 G_\infty (1-\pi)^{-6})$
$M_4$	$O(G_\infty (1-\pi)^{-2})$
$M_5$	$O(GG_\infty (1-\pi)^{-4})$
$T$	$O(M_1 M_3 \epsilon^{-2} + M_1 M_4 \sigma^2 n^{-1} \epsilon^{-2} + M_2 \epsilon^{-1})$

Table 1: Theorem constants with dependency on  $\pi$ .

Table 1 summarize the dependency of the theorem constant on  $\pi$ . Note that here the total number of iterates  $T$  scales as  $(1-\pi)^{-8}$ , which is higher than that of distributed SGD (Philippenko and Dieuleveut, 2020, 2021). This is mainly due to the fact that the convergence of adaptive gradient methods and the theoretical proof of adaptive gradient methods rely heavily on the bounded gradient assumption. While for communication-compressed adaptive methods, the gradient bound for compressed gradient  $\hat{\mathbf{g}}$  and  $\tilde{\mathbf{g}}$  is already dependent on  $\pi$  as shown in the above derivations. In fact, our dependency on gradient bounds matches the existing works studying vanilla adaptive gradient methods (Zhou et al., 2018; Chen et al., 2019), which suggests that our bound is indeed tight.

Note that in practice, the high dependency on  $\pi$  does not make much difference. In fact, the actual  $\pi$  computed using real compressor is in a constant order. For example, in our ResNet-18 experiment with scaled sign compressor, the actual  $\pi$  ranges from [0.597, 0.713] during the entire training procedure, which is indeed a reasonable constant.

## E Additional Experiments

In this section, we present the additional experiments of our proposed method. In Section E.1, we provide experiments using the nonconvex logistic regression with Top- $k$  compressor. In Section E.2, we present additional deep learning experiments on other popular state-of-the-art model architectures.

### E.1 Nonconvex Logistic Regression Based on Top-k Compressor

The nonconvex logistic regression problem setting follows exactly the same as in Section 7.1. Here we apply the proposed Markov compression sequence based on the Top- $k$  compressor, where  $k = 1$  for the nonconvex logistic regression with  $d = 300$  parameters.

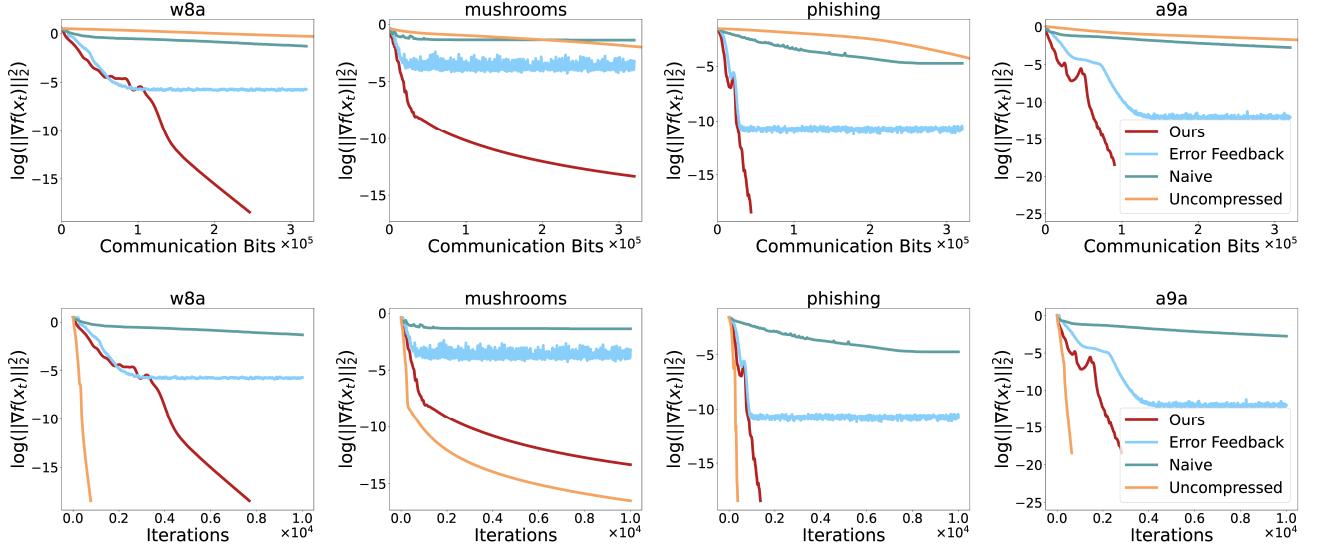


Figure 4: Gradient norm comparison of different compression strategies on nonconvex logistic regression trained by AMSGrad with Top-1 based Markov compression sequence. The upper row is the norm convergence with respect to the communication cost, and the lower row is with respect to the training iteration.

Figure 4 shows that our proposed CD-Adam achieved the best performances for all four datasets against the other three compression strategies. Specifically, we can observe that CD-Adam achieves a much better result than AMSGrad with error feedback and naive compression in terms of both communication efficiency and the final performances. Furthermore, compared with uncompressed AMSGrad, our communication efficiency is greatly improved while maintaining similar final performances.

### E.2 Deep Learning Experiments on Image Classification

In this part, we provide additional deep learning experiments on other state-of-the-art model architectures. The problem setting follows exactly the same as in Section 7.2. In the following, we show that our proposed algorithm enjoys a fast convergence speed with high accuracy and less communication cost compared with EF21 and 1-bit Adam.

Figure 5, 7 and 9 compare the performance with respect to the communication cost using different model architectures. The experiments show that our proposed method obtains better training/testing loss and accuracy, especially much better training/testing gradient norm on various CNN model architectures, including ResNet, VGG, and WideResNet. Notice that under this comparison, EF21, which is also equipped with Markov compression sequences on SGD, performs worse than our method, especially in later training stages. While for 1-bit Adam, since it needs an uncompressed warm-up procedure for a few epochs, the communication efficiency is much worse compared with CD-Adam and EF21.

Figure 6, 8 and 10 compare the performance with respect to the iteration epochs using different model architectures. Compared with EF21, our proposed method obtains a much better convergence speed in terms of training/testing loss, accuracy, and the gradient norm. Our method also shows a better performance than 1-bit Adam in terms of the final testing accuracy and gradient norm.

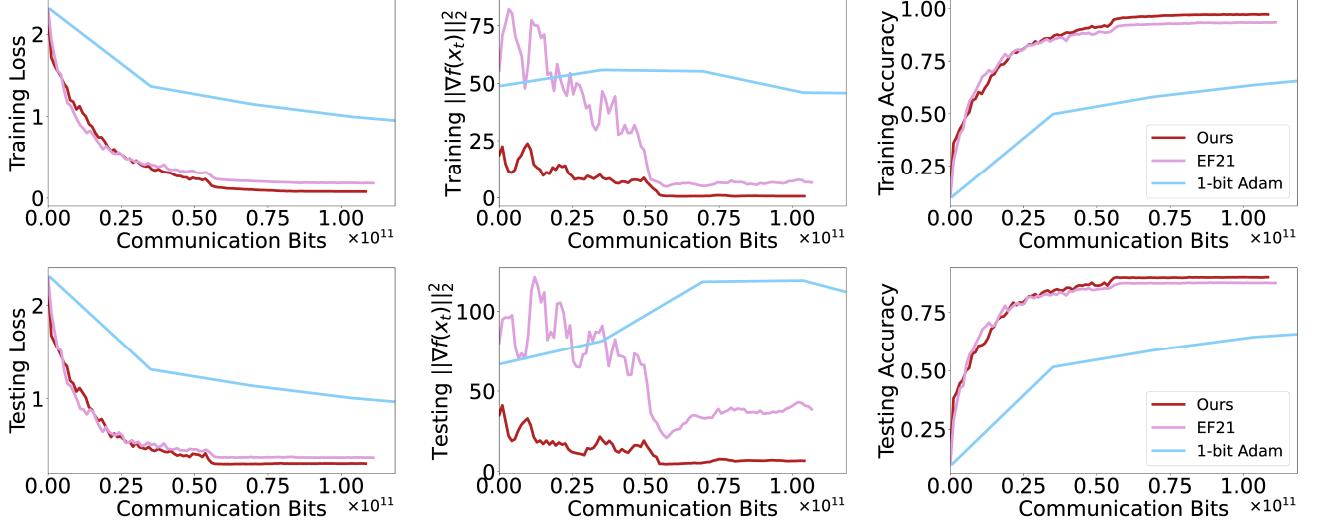


Figure 5: Comparison on training/testing loss, gradient norm and accuracy with respect to communication cost among our proposed method and the baselines when training ResNet-18 model on CIFAR10.

Figure 5 is the continuous of Figure 3. It shows that our proposed CD-Adam achieves better performance than EF21 and 1-bit Adam under the same communication cost.

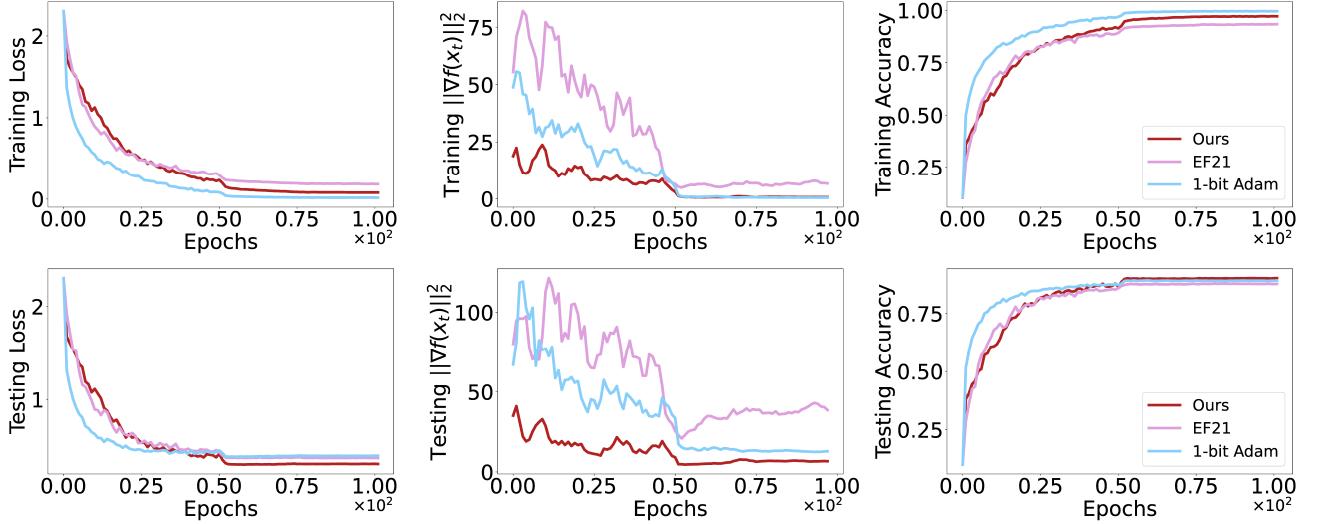


Figure 6: Comparison on training/testing loss, gradient norm and accuracy with respect to epochs among our proposed method and the baselines when training ResNet-18 model on CIFAR10.

In Figure 6, the left two plots show that at the early stage of the training process, CD-Adam and EF21 obtain a similar speed of reducing the training loss, while 1-bit Adam is quite faster due to the warm-up process. At later stages, CD-Adam shows a clear advantage compared with EF21. Our method also shows better performance on testing loss than 1-bit Adam. From the middle two plots in Figure 6, we can observe that our proposed CD-Adam obtains a smaller gradient norm than EF21 and 1-bit Adam under the same epochs when training

and testing. The right two plots show that our proposed method achieves high accuracy on both the training and testing sets. At the early stage of the training process, CD-Adam and EF21 obtain a similar speed of increasing the accuracy, while 1-bit Adam is faster due to the warm-up process. At later stages, CD-Adam shows a clear advantage compared with EF21. Our method shows better performance on testing accuracy compared with 1-bit Adam.

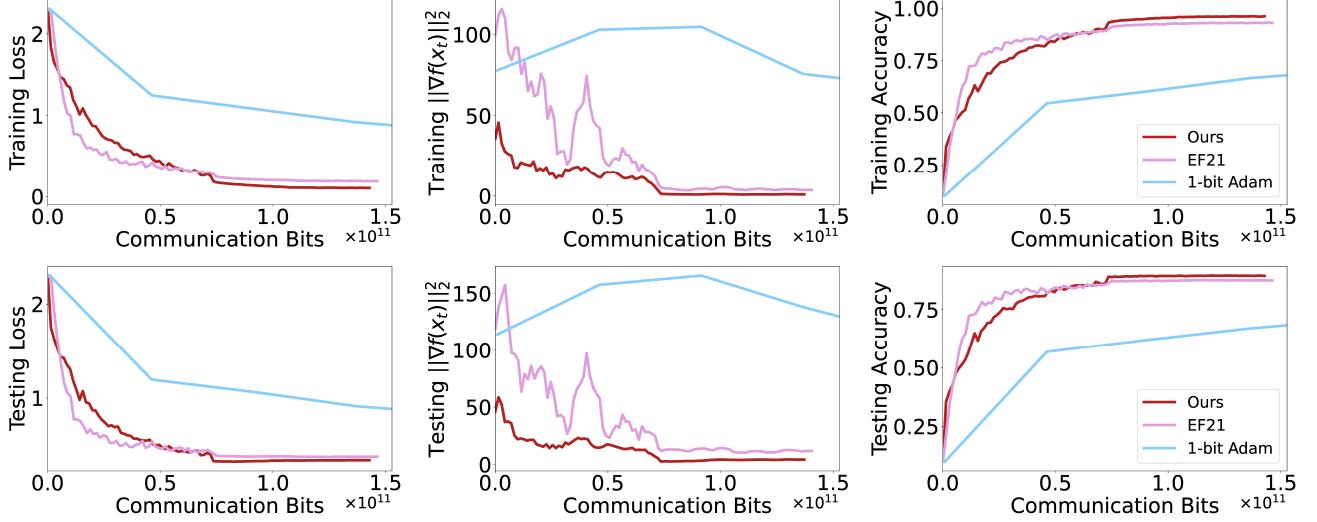


Figure 7: Comparison on training/testing loss, gradient norm and accuracy with respect to communication cost among our proposed method and the baselines when training VGG-16 model on CIFAR10.

In Figure 7, the left two plots show that our proposed method obtains better training/testing loss than EF21 and 1-bit Adam, though EF21 intermediately shows a faster decrease in the loss function. From the middle two plots, we can observe that our proposed CD-Adam obtains smaller training/testing gradient norms than EF21 and 1-bit Adam under the same communication budget. The right two plots show that CD-Adam obtains better training/testing accuracy than EF21 and 1-bit Adam, though EF21 intermediately shows a faster increase in accuracy as well.

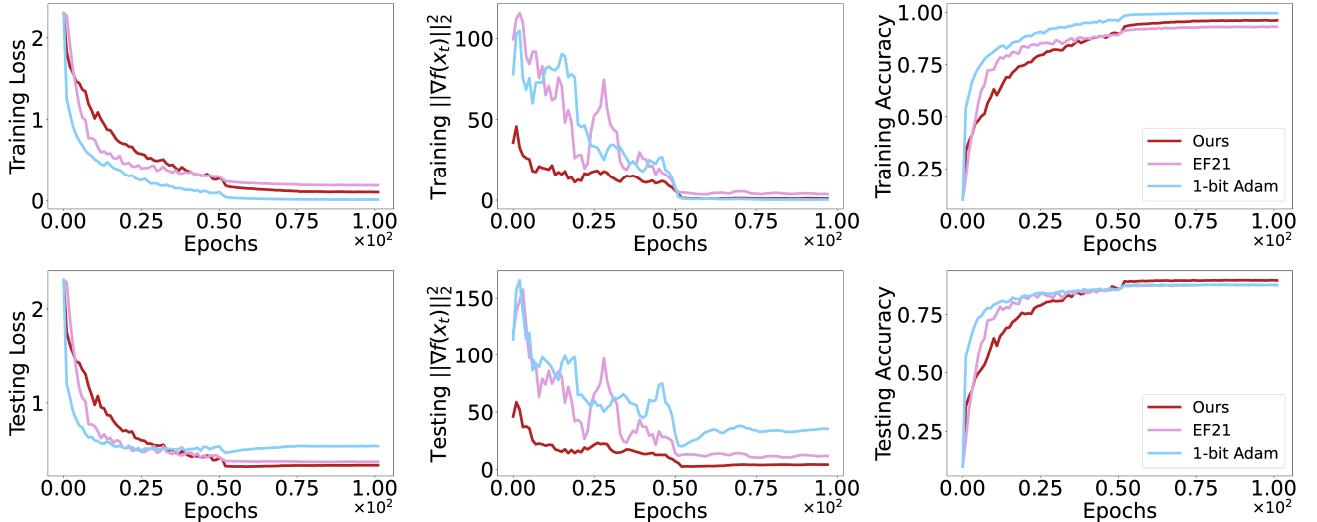


Figure 8: Comparison on training/testing loss, gradient norm and accuracy with respect to epochs among our proposed method and the baselines when training VGG-16 model on CIFAR10.

In Figure 8, the left two plots show that our proposed method achieves better training/testing loss than EF21, though EF21 intermediately shows a faster decrease in the loss function. Our method shows better performance on testing loss compared with 1-bit Adam. The middle two plots in Figure 8 show that our proposed CD-Adam obtains a smaller gradient norm than EF21 and 1-bit Adam under the same epochs. The right two plots show that our proposed method achieves high accuracy on both the training and testing sets. It shows that our proposed method achieves better training/testing loss than EF21, though EF21 intermediately shows a faster increase in accuracy. Our CD-Adam also shows a better testing accuracy than 1-bit Adam.

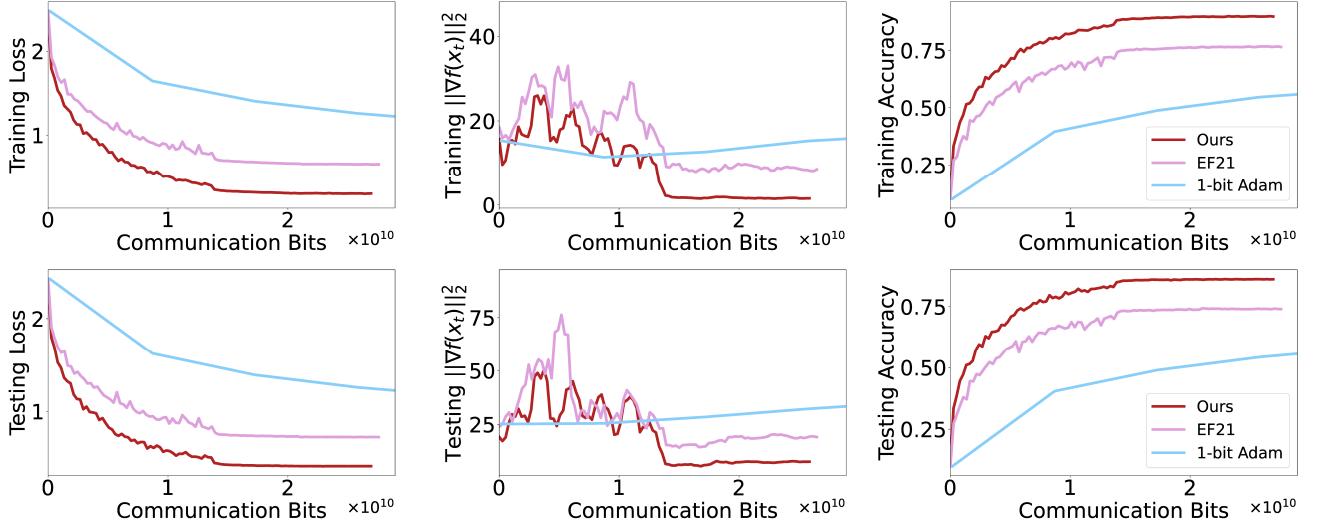


Figure 9: Comparison on training/testing loss, gradient norm and accuracy with respect to communication cost among our proposed method and the baselines when training WideResNet-16-4 model on CIFAR10.

In Figure 9, the left two plots and the right two plots show that our proposed method maintains a significant advantage on training/testing loss and accuracy compared with EF21 and 1-bit Adam. The middle two plots show that our proposed CD-Adam achieves smaller gradient norms than EF21 and 1-bit Adam. In addition, 1-bit Adam initially shows a lower gradient norm while its gradient norm diverges later.

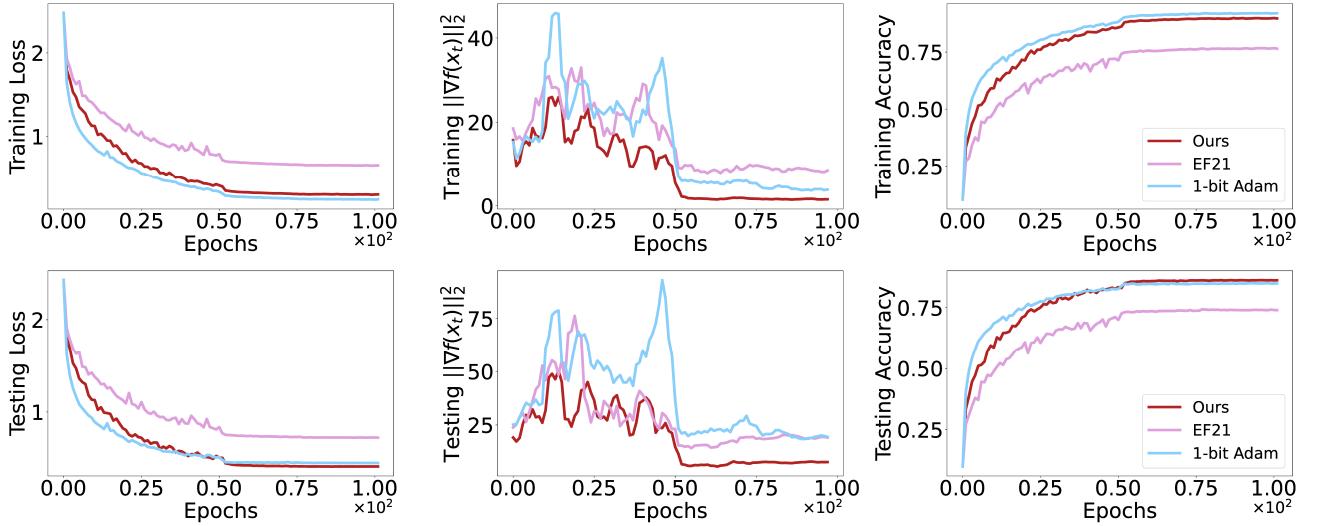


Figure 10: Comparison on training/testing loss, gradient norm and accuracy with respect to epochs among our proposed method and the baselines when training WideResNet-16-4 model on CIFAR10.

In Figure 10, the left two plots and right two plots show that our proposed method achieves better training/testing loss and accuracy than EF21. Furthermore, CD-Adam still achieves better performance on testing loss and testing accuracy than 1-bit Adam, while 1-bit Adam keeps a slighter better performance on training loss and training accuracy. The middle two plots in Figure 9 show that CD-Adam achieves better training/testing gradient norms compared with EF21 and 1-bit Adam.

### E.3 Additional results

**Average runtime and total bits:** The overhead on compression as well as the worker-side update is quite small, as shown from Table 2. Note that EF21 takes more time since it adopts top- $k$  which is harder to compute compared to scaled sign compressor.

Method	Avg Time (s/iter)	Total Bits
Uncompressed	1.015	$32d \times 2T$
EF21	1.402	$\approx (32k \times 2) \times 2T$
1-bit Adam	1.041	$32d \times 2T_1 + (32+d) \times 2(T-T_1)$
<b>CD-Adam</b>	<b>1.134</b>	<b><math>(32+d) \times 2T</math></b>

Table 2: Average runtime and total bits

**Ablation on  $n$  and  $\tau$ :** The left plot in Figure 11 shows the effect of the total amount of workers  $n$  on the convergence of training loss. It shows that a larger  $n$  leads to a faster decrease on training loss but not absolutely leads to a better convergence rate. The right plot in Figure 11 shows the effect of batch size  $\tau$  on the convergence of training loss. We observe that a larger number of batch size achieves a faster convergence rate.

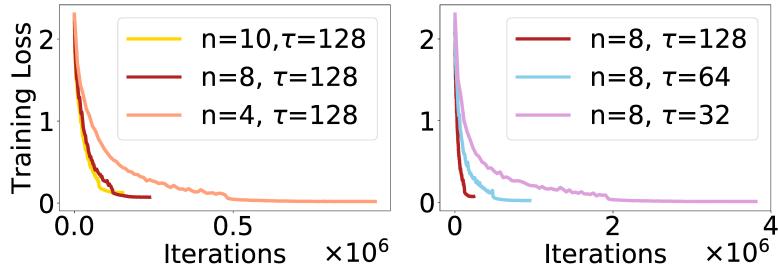


Figure 11: Ablation on  $n$  and  $\tau$