Clapping: Removing Per-sample Storage for Pipeline Parallel Learning with Communication Compression

Boao Kong[1,*], Xu Huang[1,*], Yuqi Xu[1,*]
Yixuan Liang[2], Bin Wang[3], Kun Yuan[1,†]

kongboao@stu.pku.edu.cn, ydove1031@gmail.com, xuyq@stu.pku.edu.cn
liangyx25@mails.tsinghua.edu.cn, wangbin_bme@zju.edu.cn, kunyuan@pku.edu.cn

[1] Peking University [2] Tsinghua University [3] MetaCarbon

[*] Equal contribution. [†] Corresponding author.

Abstract

Pipeline-parallel distributed optimization is essential for large-scale machine learning but is challenged by significant communication overhead from transmitting high-dimensional activations and gradients between workers. Existing approaches often depend on impractical unbiased gradient assumptions or incur sample-size memory overhead. This paper introduces **Clapping**, a **C**ommunication compression algorithm with **LA**zy sam**P**ling for **P**ipeline-parallel learn**ING**. Clapping adopts a lazy sampling strategy that reuses data samples across steps, breaking sample-wise memory barrier and supporting convergence in few-epoch or online regimes. Clapping comprises two variants including Clapping-**FC** and Clapping-**FU**, both of which achieve convergence without unbiased assumption for compressed gradient, effectively addressing compression error propagation in multi-worker settings. Numerical experiments validate the performance of Clapping across different tasks.

1 Introduction

Large-scale optimization and learning have become essential tools in numerous applications. Addressing these complex and large problems presents a substantial challenge, frequently necessitating extensive computation over many days or even months. Consequently, distributed algorithms are crucial for accelerating large-scale optimization and learning processes. In distributed optimization, multiple workers collaborate to solve a global problem with the help of communication between workers. Most existing research focuses on data-parallel distributed optimization [32, 43, 5, 73, 27, 1]. In this paradigm, each worker maintains a complete replica of the model, independently samples training data, and exchanges models or gradients at each iteration, thereby achieving significant acceleration of large-scale optimization and learning tasks.

As model parameters in contemporary optimization and learning problems have grown to hundreds of billions [46, 30, 47, 74, 34], these models have exceeded single-worker memory capacity, necessitating model-parallel distributed optimization. This paradigm partitions the model across multiple workers, with each worker managing a subset of parameters, enabling the training of massive models that would be intractable on a single worker. Model-parallel distributed optimization is particularly prevalent in the pre-training and fine-tuning of Large Language Models (LLMs) [61, 39]. For instance, consider an LLM architecture comprising 24 transformer layers that exceeds the memory capacity of a single GPU cluster. The model can be efficiently segmented, with the initial 12 layers allocated to one GPU cluster and the remaining layers to another. This partitioning ensures each GPU maintains a fraction of the model's parameters, substantially reducing per-device memory requirements. Another

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

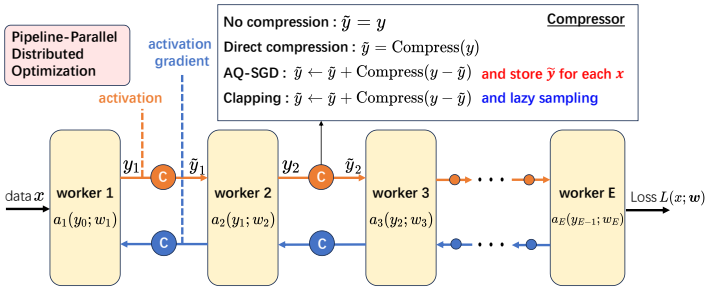

Figure 1: Illustration of the pipeline-parallel distributed optimization with communication compression.

prominent application of model-parallel distributed optimization is split learning [60, 33, 66], where the entire machine learning model is divided into smaller network segments and trained independently across multiple edge computing devices.

Problem statement. This paper examines pipeline-parallel distributed optimization, a specific form of model-parallel optimization that partitions model parameters in a pipeline-like manner [41, 21, 42, 52, 63]. Consider a computing cluster consisting of $E \geq 2$ workers. Model parameter $\boldsymbol{w}$ is partitioned into E parts, denoted as $\boldsymbol{w} = (w_1, \cdots, w_E) \in \mathbb{R}^{d_{w_1} + \cdots + d_{w_E}}$, where each block component $w_e \in \mathbb{R}^{d_{w_e}}$ is maintained by worker e. Pipeline-parallel optimization can be formulated as follows:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad \mathbb{E}_{x \in \mathcal{D}}\Big[L(x; \boldsymbol{w}) := y_E\Big], \tag{1a}$$

$$\text{s.t.} \quad y_e = a_e(y_{e-1}, w_e), \quad \forall e \in \{1, \cdots E\}. \tag{1b}$$

In the above problem, the random variable x denotes the data sample following distribution $\mathcal{D}$. The loss $L(x, \boldsymbol{w})$ is a composite function consisting of E operators $a_e(y_{e-1}, w_e) : \mathbb{R}^{d_{e-1}} \times \mathbb{R}^{d_{w_e}} \to \mathbb{R}^{d_e}$ with $d_E = 1$, where y_e, following the terminology in LLMs, refers to the activation. For initialization, we let $y_0 = x$. An illustration for problem (1) is shown in Fig. 1, where each worker participates in the computation in a pipeline fashion. In LLMs, each operator $a_e(y_{e-1}, w_e)$ corresponds to a transformer layer. In split learning, there are two operators: $a_1(y_0, w_1)$, which represents the client-side sub-network, and $a_2(y_1, w_2)$, which corresponds to the server-side sub-network.

Pipeline-parallel distributed algorithms are effective for large-scale problems with multiple devices but incur significant communication overhead from transmitting high-dimensional activation and gradient vectors over low-bandwidth links [9, 52, 67], as illustrated in Figure 2. While current LLM pre-training is typically run on tightly coupled GPU clusters with 100–400 Gbps interconnects [54], such infrastructure is accessible only to a few industrial labs. In contrast, emerging decentralized and over-the-internet training frameworks aim to scale model parallelism across commodity GPUs connected via consumer-grade networks at only *tens to hundreds of Mbps*, yet still match datacenter-level convergence [49, 67]. This regime is fundamentally communication-bound: even a single forward–backward pass can be dominated by transmitting activations and gradients. These trends motivate studying pipeline-parallel *communication compression* methods that remain stable under aggressive activation/gradient compression without assuming high-bandwidth interconnects, and thus are applicable to both datacenter and slow-network deployments.

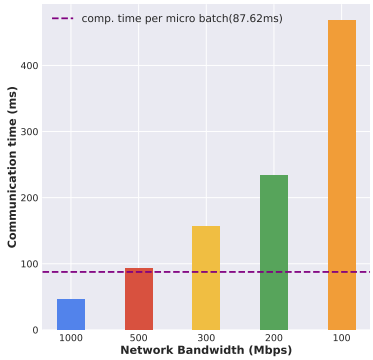

Figure 2: Communication time for fine-tuning GPT2-xl with different bandwidth and average computation time per micro batch on 4 Nvidia A100 GPUs.

Fundamental challenges. While communication compression has been extensively studied for data-parallel distributed optimization [1, 58, 62, 50, 20, 12], it remains largely unexplored for pipeline-parallel distributed optimization. Compressing activations and gradients, as illustrated in Fig.1, presents two fundamental challenges. First, each stochastic gradient computed through

forward-backward propagation requires $2E-2$ rounds of compression ($E-1$ in the forward pass and $E-1$ in the backward pass), introducing substantial errors in gradient estimation that may lead to non-convergence. Second, the composite structure of activations (see Eq. (1b)) results in error propagation during pipeline communication, causing a non-separable entanglement of gradient information and compression errors (detailed in SectionA.1). These challenges render most effective techniques from data-parallel optimization infeasible for pipeline-parallel optimization.

1.1 Limitations in existing works

Several studies have emerged to address these challenges [10, 13, 67], but they exhibit critical limitations.

(L1) Impractical unbiased assumption for compressed gradient. The unbiased assumption for compressed gradient is crucial for ensuring the convergence of optimization algorithms. In line with this, existing works [10, 13] assume that unbiased activation compression leads to unbiased gradient errors, but this fails due to the composite and non-linear structure of the operator $a_e(y_{e-1}, w_e)$. For example, even if $\mathbb{E}[\tilde{y}_{e-1}] = y_{e-1}$, it is not generally true that $\mathbb{E}[a_e(\tilde{y}_{e-1}, w_e)] = a_e(y_{e-1}, w_e)$.

(L2) Sample-size memory overhead. [67] proposes AQ-SGD, which achieves convergence without requiring the unbiased assumption for compressed gradient. Unlike approaches in [10, 13] that directly compress activations, AQ-SGD compresses the change in activations for identical training samples across epochs. However, AQ-SGD requires storing activations for **all training samples**, resulting in memory overhead proportional to the sample size.

(L3) Multiple-epoch training requirement. Another technique behind the convergence of AQ-SGD is error compensation [53, 58, 50], which can progressively mitigate activation compression errors. However, this approach requires extensive training epochs to eliminate errors, leading to non-convergence in few-epoch training regimes. This limitation is particularly significant for large-scale LLM training/fine-tuning, where practical workloads typically involve only a few epochs [47, 74, 39, 56, 40, 55, 15].

(L4) Limited scalability beyond two-worker setup. The convergence analysis in [10, 13] is designed for two-worker configurations and does not extend to larger pipeline systems. While AQ-SGD offers convergence guarantees for multi-worker setups, it assumes **no error accumulation** in the multi-worker pipeline scenario, which is generally **NOT** true in practice.

1.2 Contributions

This paper develops novel algorithms to address the aforementioned limitations. Our contributions are as follows:

(C1). We propose `Clapping`, a **C**ommunication compression framework with **LA**zy sam**P**ling for **P**ipeline-parallel Learn**ING**. The `Clapping` framework is versatile and can be implemented in two variants: `Clapping`-**FC** and `Clapping`-**FU**, which differ in their approaches to the compression strategy in forward- and backward-propagation. A core technique underpinning `Clapping` is its novel lazy sampling strategy. By retaining the same data sample across multiple steps, this strategy enables `Clapping` to overcome the sample-size memory overhead associated with AQ-SGD and achieve convergence under few-epoch or even single-epoch training regimes, effectively addressing **Limitations (L2)** and **(L3)**.

(C2). We establish convergence guarantees for two variants, `Clapping`-**FC** and `Clapping`-**FU**, for pipeline-parallel optimization with inter-stage communication compression. Under standard smoothness assumptions and a contractive compressor model, `Clapping`-**FC** achieves an $O(T^{-1/3})$ stationarity rate and `Clapping`-**FU** achieves an $O(T^{-1/2})$ rate, without relying on the additional *unbiased compressed-gradient* assumption and multi-epoch training, addressing **Limitations (L1)** and **(L4)**. Moreover, our analysis explicitly characterizes how compression errors accumulate with the pipeline depth E in multi-stage settings.

(C3). We conduct extensive experiments across split learning, LLMs fine-tuning and pre-training. `Clapping` is comparable to AQ-SGD in several fine-tuning with far less memory cost and has a wider applicability than the other communication compression algorithms.

Table 1: Comparison between different communication compression algorithms for pipeline-parallel optimization. Notation T denotes the number of iterations, x denotes the training sample, N is the size of training data, B is the batch size, q denotes the dimension of the communicated parameters. We also list the result of Momentum SGD without compression in the bottom line for reference.

Algorithms	**C. Rate**$^{\diamond}$	**No Unbiased Comp-Grad. Assu.**†	**C. Mem.**‡	**Few-epo. Rate**$^{\triangleright}$	**Multi. Workers**$^{\triangleleft}$
AC-GC [10]	$T^{-1/2}$	✗	0	N.A.	✗
TinyScript [13]	$T^{-1/2}$	✗	0	N.A.	✗
AQ-SGD [67]	$NT^{-1/2}$	✔	$\mathcal{O}(Nq)$	✗	☹
Clapping-FC (Ours)	$\boldsymbol{T^{-1/3}}$	✔	$\boldsymbol{\mathcal{O}(Bq)}$	✔	✔
Clapping-FU (Ours)	$\boldsymbol{T^{-1/2}}$	✔	$\boldsymbol{\mathcal{O}(Bq)}$	✔	✔
Momentum SGD [44]	$T^{-1/2}$	✔	N.A.	✔	✔

$^{\diamond}$ The convergence rate with respect to total steps $T \to \infty$ (smaller is better).
† No additional assumptions regarding the unbiased compression-induced gradient error.
‡ Extra memory overhead incurred during a single communication compression operation. N.A. indicates no compression.
$^{\triangleright}$ The algorithm can achieve convergence in the few-epoch learning tasks, i.e., $N = \mathcal{O}(T)$.
$^{\triangleleft}$ Convergence analysis can be extended to the multiple-worker setup. ☹ indicates the analysis is with impractical assumptions.

All theoretical results and existing algorithms for communication compression in pipeline-parallel optimization are summarized in Table 1. Notably, `Clapping` achieves convergence without assuming an unbiased compression-induced gradient estimator, avoids sample-wise activation storage, and applies to few-epoch workloads. Moreover, our theory characterizes how compression errors propagate and accumulate with the pipeline depth E, rather than implicitly assuming the depth effect away.

Notations. In this paper, we define $\ell(\boldsymbol{w}) := \mathbb{E}_{x\sim\mathcal{D}}[L(x;\boldsymbol{w})]$ throughout this paper. We further denote its partial gradient with respect to the parameter w_e as $\nabla_e \ell(\boldsymbol{w}) := \partial\ell(\boldsymbol{w})/\partial w_e \in \mathbb{R}^{d_{w_e}}$ for $e = 1, 2, \ldots, E$. For the operator $a_e(y_{e-1}, w_e) : \mathbb{R}^{d_{e-1}} \times \mathbb{R}^{d_{w_e}} \to \mathbb{R}^{d_e}$, we define $\nabla_1 a_e(y_{e-1}, w_e) \in \mathbb{R}^{d_e \times d_{e-1}}$ and $\nabla_2 a_e(y_{e-1}, w_e) \in \mathbb{R}^{d_e \times d_{w_e}}$ as the Jacobian matrix with respect to y_{e-1} and w_e, respectively. The notation $\|\cdot\|$ represents the ℓ_2-norm of vectors, and $\mathbf{1}_B \in \mathbb{R}^B$ denotes a vector with all elements equal to 1. For variables $\{y_e^{(t)}\}_{e=1,\ldots,E}^{t=0,\ldots,T+1}$, the subscript e (resp. superscript t) denotes the index of the worker (resp. iteration). We use $a \lesssim b$ to indicate that there exists a constant $C \geq 0$ such that $a \leq Cb$, and $a \lesssim_d b$ indicates that there exists a $C \geq 0$ that is independent with d such that $a \leq Cb$.

2 Prior Arts

Here we display some prior works of pipeline-parallel distributed optimization. Additional related works can be found in Appendix A.2.

2.1 Pipeline-parallel SGD

To solve pipeline-parallel optimization problem (1), we define $v_e := \nabla_{y_e} L(x;\boldsymbol{w}) \in \mathbb{R}^{d_e}$ as the gradient of the loss function with respect to the activation y_e, referred to as the activation gradient. Similarly, we define $u_e := \nabla_{w_e} L(x;\boldsymbol{w}) \in \mathbb{R}^{d_{w_e}}$ as the gradient with respect to the weight w_e, referred to as the weight gradient. Pipeline-parallel SGD performs the following steps at iteration t:

- **Forward**: Each worker e computes activation $y_e^{(t)} = a_e(y_{e-1}^{(t)}, w_e^{(t)})$ following the forward order $e = 1, \cdots, E$. We let $y_0^{(t)}$ be the random sample $x^{(t)}$ for initialization.
- **Backward**: Each worker e computes activation gradient $v_{e-1}^{(t)} = \nabla_1 a_e(y_{e-1}^{(t)}, w_e^{(t)})^\top v_e^{(t)}$ following the backward order $e = E, \cdots, 2$. We initialize $v_E^{(t)} = 1$.
- **Update**: Each worker e computes weight gradient $u_e^{(t)} = \nabla_2 a_e(y_{e-1}^{(t)}, w_e^{(t)})^\top v_e^{(t)}$ following the backward order $e = E, \cdots, 1$. We update parameter $w_e^{(t+1)} = w_e^{(t)} - \gamma u_e^{(t)}$.

Pipeline-parallel SGD iteratively repeats the aforementioned steps until convergence (see Fig. 1). This requires transmitting all activations and their corresponding gradients between workers. To reduce communication overhead, we can apply direct compression of activations and gradients [11, 13]. However, such a compression may cause non-convergence due to biased gradient estimation and error propagation, even with unbiased compressors (see Appendix A.1).

2.2 AQ-SGD

To eliminate convergence bias in direct compression, [67] proposes AQ-SGD, an algorithm designed to ensure exact convergence to the stationary solution without relying on the impractical assumption of unbiased compressed gradients. The key mechanism employed by AQ-SGD is error feedback [50]. For a specific random sample x, we let $y_{x,e}$ represent the activation associated with that sample on each worker e, where $y_{x,0} = x$ serves as the initialization. Rather than compressing each activation $y_{x,e}$ directly, AQ-SGD compresses the changes in activations as follows:

$$y_{x,e}^{(t)} = a_e(\tilde{y}_{x,e-1}^{(t)}, w_e^{(t)}), \tag{2}$$

$$\tilde{y}_{x,e}^{(t)} = \tilde{y}_{x,e}^{(t-1)} + \mathcal{C}(y_{x,e}^{(t)} - \tilde{y}_{x,e}^{(t-1)}). \tag{3}$$

Suppose the compressor $\mathcal{C}(\cdot)$ is contractive, i.e., $\|\mathcal{C}(y) - y\| \leq \omega\|y\|$ for some $\omega \in (0, 1)$ (see Assumption 3 for details), it then holds that

$$\|\tilde{y}_{x,e}^{(t)} - y_{x,e}^{(t)}\| = \|\mathcal{C}(y_{x,e}^{(t)} - \tilde{y}_{x,e}^{(t-1)}) - (y_{x,e}^{(t)} - \tilde{y}_{x,e}^{(t-1)})\| \leq \omega\|\tilde{y}_{x,e}^{(t-1)} - y_{x,e}^{(t)}\|.$$

If we further assume that $y_{x,e}^{(t)} \to y_{x,e}^{\star} := a_e(y_{x,e-1}^{\star}, w_e^{\star})$, it follows that $\|\tilde{y}_{x,e}^{(t)} - y_{x,e}^{\star}\| \leq \omega\|\tilde{y}_{x,e}^{(t-1)} - y_{x,e}^{\star}\|$, implying that $\tilde{y}_{x,e}^{(t)}$ converges asymptotically to $y_{x,e}^{\star}$.

While error feedback progressively eliminates the compression bias in the activations, it requires repeatedly executing (2) with the same sample x over multiple iterations. To enable the error feedback update, AQ-SGD passes all data samples across multiple epochs and stores each $\tilde{y}_{x,e}$ for every data sample x and worker e. When a data sample x is selected at iteration t, AQ-SGD updates $\tilde{y}_{x,e}^{(t)}$ according to (2); otherwise, it retains the previous value, ensuring $\tilde{y}_{x,e}^{(t)} = \tilde{y}_{x,e}^{(t-1)}$. Specifically, AQ-SGD operates as follows.

- **Forward**: Suppose sample x is selected at the current iteration t. Each worker e computes activation $y_{x,e}^{(t)}$ and compresses $\tilde{y}_{x,e}^{(t)}$ according to (2), following the forward order $e = 1, \cdots, E$. We let $y_0^{(t)} = x^{(t)}$. Finally, we update $\tilde{y}_{x',e}^{(t)} = \tilde{y}_{x',e}^{(t-1)}$ for any $x' \neq x$.
- **Backward**: Each worker e computes activation gradient $v_{e-1}^{(t)} = \nabla_1 a_e(\tilde{y}_{x,e-1}^{(t)}, w_e^{(t)})^{\mathsf{T}} \tilde{v}_e^{(t)}$ and compresses $\tilde{v}_{e-1}^{(t)} = \mathcal{C}(v_{e-1}^{(t)})$ following the order $e = E, \cdots, 2$.
- **Update**: Each worker e computes weight gradient $u_e^{(t)} = \nabla_2 a_e(\tilde{y}_{x,e-1}^{(t)}, w_e^{(t)})^{\mathsf{T}} \tilde{v}_e^{(t)}$ following the order $e = E, \cdots, 1$. We update parameter $w_e^{(t+1)} = w_e^{(t)} - \gamma u_e^{(t)}$.

AQ-SGD operations are illustrated in Fig. 1, with error feedback applied only during the forward pass (see details in [66]). Each worker e must store activations $\tilde{y}_{x,e-1}$ for every sample x, limiting the method to tiny datasets and making it unsuitable for online learning with infinite data streams. This requires memory proportional to sample size, resulting in significant overhead and posing a substantial challenge in large-scale optimization, especially with extensive datasets and large models.

3 `Clapping` Algorithm

Here, we introduce `Clapping` , a novel approach designed to overcome the limitations of AQ-SGD.

Lazy sampling. As discussed in Section 2.2, error feedback is critical in communication compression for pipeline-parallel distributed optimization. The primary challenge lies in the fact that the update (2) must be repeated a sufficient number of times to progressively remove the compression bias. To facilitate error feedback, AQ-SGD stores $\tilde{y}_{x,e}$ for every data sample x on each worker e. This design

Algorithm 1 LazySampling$(\mathcal{D}, t, p)$

if $t = 1$ **then**
  Sample $x^{(1)} \sim \mathcal{D}$; $f_{\text{FU}}^{(1)} \leftarrow$ `True`.
else
  With prob. $1-p$: set $x^{(t)} \leftarrow x^{(t-1)}$ and $f_{\text{FU}}^{(t)} \leftarrow$ `False`;
  With prob. p: sample $x^{(t)} \sim \mathcal{D}$ and $f_{\text{FU}}^{(t)} \leftarrow$ `True`.
end if
Return: $x^{(t)}$, $f_{\text{FU}}^{(t)}$.

Algorithm 2 Forward$_e(\tilde{y}_e^{(t-1)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$

$y_e^{(t)} \leftarrow a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})$.
if `Clapping`-**FU and** $f_{\text{FU}}^{(t)} =$ `True` **then**
  Send $y_e^{(t)}$ from worker e to e+1;
  $\tilde{y}_e^{(t)} \leftarrow y_e^{(t)}$.
else
  Send $\mathcal{C}(y_e^{(t)} - \tilde{y}_e^{(t-1)})$ from worker e to e+1;
  $\tilde{y}_e^{(t)} \leftarrow \tilde{y}_e^{(t-1)} + \mathcal{C}(y_e^{(t)} - \tilde{y}_e^{(t-1)})$.
end if
Return: $\tilde{y}_e^{(t)}$.

Algorithm 3 Backward$_e(\tilde{v}_{e-1}^{(t-1)}, \tilde{v}_e^{(t)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$

$v_{e-1}^{(t)} \leftarrow \nabla_1 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\top \tilde{v}_e^{(t)}$.
if `Clapping`-**FU and** $f_{\text{FU}}^{(t)} =$ `True` **then**
  Send $v_{e-1}^{(t)}$ from worker e to $e-1$;
  $\tilde{v}_{e-1}^{(t)} \leftarrow v_{e-1}^{(t)}$.
else
  Send $\mathcal{C}(v_{e-1}^{(t)} - \tilde{v}_{e-1}^{(t-1)})$ from worker e to $e-1$;
  $\tilde{v}_{e-1}^{(t)} \leftarrow \tilde{v}_{e-1}^{(t-1)} + \mathcal{C}(v_{e-1}^{(t)} - \tilde{v}_{e-1}^{(t-1)})$.
end if
Return: $\tilde{v}_{e-1}^{(t)}$.

is the key reason for its limitations, which include a sample-size memory overhead and the inability to handle infinite datasets.

We propose a lazy sampling strategy to address this challenge. Unlike AQ-SGD, which stores $\tilde{y}_{e,x}$ and updates it only when data x is re-sampled during multiple-epoch training, our approach employs a fixed sample x for gradient evaluation across multiple consecutive steps, combined with error compensation. Once the compression bias is sufficiently reduced, we proceed to sample the next data point. As outlined in Algorithm 1, we retain the previous sample with probability $1-p$; otherwise, a new sample is drawn from $\mathcal{D}$. This strategy enables sample reuse across iterations without storing $y_{e,x}$, thereby eliminating sample-size memory overhead. Moreover, lazy sampling facilitates error feedback updates even in settings with infinitely many data samples.

Error feedback. `Clapping` employs error feedback in both forward and backward processes, as detailed in Algorithms 2 and 3. In contrast, AQ-SGD applies error feedback exclusively to the forward process. Furthermore, leveraging the lazy sampling strategy, our approach eliminates the need to maintain $\{\tilde{y}_{e,x}\}$ and $\{\tilde{v}_{e,x}\}$ for each sample $x \in \mathcal{D}$. As shown in Algorithms 2 and 3, we only maintain $\tilde{y}_e$ and $\tilde{v}_e$ across all samples and iterations.

Incorporation of momentum. Momentum is a widely used technique to accelerate SGD convergence, acting as a surrogate for large-batch gradients and reducing gradient variance. Recent studies underscore its theoretical benefits: [7] highlights its role in mitigating data heterogeneity, while [12] demonstrates its effectiveness in enhancing error feedback. As algorithm 4 illustrates, we can adopt the following momentum update to mitigate the gradient bias from inaccurate activation gradients:

$$\tilde{u}_e^{(t)} = (1-m_t)\tilde{u}_e^{(t-1)} + m_t \nabla_2 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\top \tilde{v}_e^{(t)}, \quad w_e^{(t+1)} = w_e^{(t)} - \gamma \tilde{u}_e^{(t)}. \tag{4}$$

where the coefficient $m_t \in (0, 1)$.

`Clapping` algorithm. Combining lazy sampling, error feedback, and momentum updates, the complete `Clapping` framework is presented in Algorithm 4. While `Clapping` is formulated with momentum SGD and a batch size of 1, it can be extended to Adam-style optimizers [25, 36]. We present `Clapping` with Adam optimizer as well as the extended theoretical convergence analysis in Appendix D, and we evaluate `Clapping` with Adam-based optimizers in Section 5. For large-batch scenarios, lazy sampling can be adapted batch-wise, simplifying implementation. The detailed lazy sampling strategy and algorithmic formulation for large batches are provided in Appendix E.1.

`Clapping`-FC vs. `Clapping`-FU. `Clapping`-**FC** and `Clapping`-**FU** differ only on refresh iterations, i.e., iterations where a fresh sample or batch is drawn. In Algorithms 2 and 3, `Clapping`-**FC** is

Algorithm 4 `Clapping`

Require: Initialize $\tilde{y}_e^{(0)} = 0, \tilde{v}_e^{(0)} = 0$ for $e = 1, \cdots, E-1$, and $\tilde{u}_e^{(0)} = 0$ for $e = 1, \cdots, E$. Initialize dataset $\mathcal{D}$, learning rate γ_t, compressor $\mathcal{C}$, and lazy sampling rate $\{p_t\}_{t=1}^T$.

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad x^{(t)}, f_{\text{FU}}^{(t)} \leftarrow \mathsf{LazySampling}(\mathcal{D}, t, p_t)$, $\tilde{y}_0^{(t)} \leftarrow x^{(t)}$, $\tilde{v}_E^{(t)} \leftarrow 1$.
3: $\quad$ **for** $e = 1, 2, \cdots, E-1$ **do**
4: $\quad\quad \tilde{y}_e^{(t)} \leftarrow \mathsf{Forward}_e(\tilde{y}_e^{(t-1)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$.
5: $\quad$ **end for**
6: $\quad$ **for** $e = E, E-1, \cdots, 1$ **do**
7: $\quad\quad \tilde{v}_{e-1}^{(t)} \leftarrow \mathsf{Backward}_e(\tilde{v}_{e-1}^{(t-1)}, \tilde{v}_e^{(t)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$ if $e \neq 1$. Update $\tilde{u}_e^{(t)}$ and $w_e^{(t+1)}$ by (4).
8: $\quad$ **end for**
9: **end for**

the always-compress variant: the condition in line 2 is never activated, so both the forward and backward routines always execute the compressed error-feedback branch in lines 6–7. By contrast, `Clapping`-**FU** executes the raw, uncompressed branch in lines 3–4 on refresh iterations, and otherwise uses the same compressed branch as `Clapping`-**FC**. Thus, at a fixed iteration budget, `Clapping`-**FC** sends fewer bytes than `Clapping`-**FU**. For example, at $p_t = 0.4$, the fixed-budget payload reduction of `Clapping`-**FC** is about $1.67\times$ that of `Clapping`-**FU**. The role of `Clapping`-**FU** is different: raw refreshes act as resets that can improve optimization stability or time-to-target performance, rather than reducing per-iteration payload more than `Clapping`-**FC**.

Memory Overhead. With the error feedback technique, `Clapping` caches the current batch's activations and gradients, resulting in $\mathcal{O}(B)$ memory overhead for batch size B. In contrast, [67]'s sample-wise error compensation incurs $\mathcal{O}(N)$ memory requirement for sample size N. The memory cost reduction is significant: while [67] theoretically requires TBs for a single communication compression, `Clapping` only needs several GBs for pre-training models with billions of parameters like LLaMA-2 7B or LLaMA-3 8B, acceptable in practice (See Appendix G for details). This advantage is more pronounced with larger models and more extensive datasets.

4 Theoretical Analysis

This section presents theoretical analysis for `Clapping`-**FC** and `Clapping`-**FU**.

4.1 Assumptions

We first introduce assumptions used throughout this paper.

Assumption 1. *There exist constants $L_{\nabla\ell}, C_a, L_{\nabla a}, L_a$ such that:*

1. *$\nabla\ell$ is $L_{\nabla\ell}$-Lipschitz continuous;*
2. *For $e = 1, 2, \cdots, E$, the gradient of a_e can be bounded by C_a, i.e. $\|\nabla a_e(y, \boldsymbol{w})\| \leq C_a$;*
3. *For $e = 1, 2, \cdots, E-1$, $a_e(y, \boldsymbol{w}), \nabla a_{e+1}(y, \boldsymbol{w})$ are $L_a, L_{\nabla a}$ Lipschitz continuous with respect to y and $\boldsymbol{w}$, respectively.*

We remark that Assumption 1 is comparable to the smoothness assumption used in [67].

Assumption 2. *The stochastic gradient with no compression error $\nabla L(x; \boldsymbol{w})$ is an unbiased estimate of $\nabla\ell(\boldsymbol{w})$ with bounded variances σ^2.*

Remark 1. *Assumption 2 is the standard unbiased stochastic-gradient assumption due to data sampling, which is orthogonal to the unbiased compressed-gradient assumption criticized in limitation **(L1)**.* `Clapping` *removes the latter, while retaining the former as in most SGD analyses.*

Assumption 3. *For the compressor $\mathcal{C}$, there exist constants $\omega_F, \omega_B \in [0, 1)$ such that:*

$$\mathbb{E}\left[\|x - \mathcal{C}(x)\|^2 \middle| x\right] \leq \begin{cases} \omega_F^2 \|x\|^2, \textit{ forward propagation}, \\ \omega_B^2 \|x\|^2, \textit{ backward propagation}. \end{cases}$$

Table 2: The score (↑) of all the tasks for GLUE benchmark with communication compression algorithms.

Algorithms	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg
No comp.	**90.01**	**96.41**	88.25	**64.69**	**94.59**	**92.03**	82.35	**92.52**	87.61
EF21	89.28	94.79	87.00	62.97	93.87	92.01	81.62	91.30	86.61
Clapping-**FC**	89.94	96.06	**90.50**	63.98	94.04	91.89	**84.19**	91.71	**87.79**

Compressors satisfying the above assumption are referred to as contractive compressors. In general, more aggressive compression leads to greater information distortion, corresponding to a larger ω. This assumption applies to numerous compressors, including top-K and low-rank projection [1, 2, 62, 4, 58], and is widely adopted in communication-efficient algorithms [26, 50, 12, 65, 20].

The assumption below is critical for lazy sampling:

Assumption 4. *For each $x_1, x_2 \sim \mathcal{D}$, there exists $\varphi > 0$ such that $\mathbb{E}_{x_1,x_2\in\mathcal{D}}\left[\|x_1 - x_2\|^2\right] \le \varphi^2$.*

It is important to note that Assumption 4 is not overly restrictive for most optimization and learning tasks. Indeed, Assumption 4 holds for **all finite datasets**. Moreover, it is likely to be satisfied even for infinite datasets, particularly **when normalization techniques are applied**.

4.2 Clapping convergence

Firstly, we present the convergence result of Clapping-**FC** is as follows.

Lemma 1. *Suppose $x^{(t)}$ is the sampled data at iteration t, if we let $p_2 = 1$ and $p_3 = \cdots = p_T = p$ as a constant, then for* Clapping-***FC****, under Assumptions 1–3 the following holds.*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim_{T,p,m} \frac{1}{\gamma T} + \frac{1}{mT} + \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|x^{(t+1)} - x^{(t)}\right\|^2\right] + \sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2} + \frac{1}{T}\left(\frac{1}{m^2}-\frac{1}{\gamma^2}\right)\sum_{e=1}^{E}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right]. \tag{5}$$

In inequality (5), the term $\sum_{t=1}^{T}\mathbb{E}[\|x^{(t+1)} - x^{(t)}\|^2]$ arises from error feedback across iterations. If different data samples are selected at iterations t and $t+1$, the term $y_e^{(t+1)} - \tilde{y}_e^{(t)}$ in Algorithm 2 introduces an error due to the discrepancy between $x^{(t+1)}$ and $x^{(t)}$. When $\sum_{t=1}^{T}\mathbb{E}[\|x^{(t+1)} - x^{(t)}\|^2] = \mathcal{O}(T)$, Algorithm 4 converges to an $\mathcal{O}(1)$ bias, as indicated by (5). This highlights the necessity to introduce lazy sampling to mitigate this term.

As outlined in Algorithm 1, under a constant lazy-sampling rate $p_t \equiv p$ we have $x^{(t+1)} = x^{(t)}$ w.p. $1-p$, and hence $\sum_{t=1}^{T}\mathbb{E}\left[\|x^{(t+1)} - x^{(t)}\|^2\right] = O(Tp)$. With this, we can present the convergence of Clapping-**FC** as follow:

Theorem 1. *Suppose $x^{(t)}$ is the sampled data at iteration t, there exist properly chosen constant step sizes γ, a momentum coefficient m, and a lazy-sampling coefficient p such that, for* Clapping-***FC****, under Assumptions 1–4 the following holds (see formal version in Theorem 4).*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim \frac{\sigma^{\frac{4}{3}}}{T^{\frac{1}{3}}(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}} + \frac{\delta}{T}, \tag{6}$$

where δ is a constant only depends on ω_B, ω_F, E as

$$\delta \lesssim \frac{1}{(1-\omega_F)^{E-1}(1-\omega_B)^{E-1}} + \frac{\omega_F^2 + \omega_B}{(1-\omega_F)^2(1-\omega_B)^{2(E-1)}} + \frac{1}{(1-\omega_F)^{2(E-2)-1}}. \tag{7}$$

Meanwhile, the convergence rate of Clapping-**FU** is as follows.

Theorem 2. *Suppose $x^{(t)}$ is the sampled data at iteration t, there exist properly chosen constant step sizes γ, a momentum coefficient m, and lazy sampling coefficient p such that, for* Clapping-***FU****, under Assumptions 1–3 the following holds (see formal version in Theorem 3).*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim \frac{\sigma}{\sqrt{T}} + \frac{1}{T(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}. \tag{8}$$

Table 3: The evaluation accuracy (↑) of different communication compression algorithms when fine-tuning different models by Wikitext with Top-5% compressor.

Model	No comp.	Direct comp.	EF21	AQ-SGD	Clapping-FC		
					$p = 0.3$	$p = 0.4$	$p = 0.5$
LLaMA-2 7B	0.5948	0.5473	0.5678	0.5696	**0.5960**	0.5920	0.5877
LLaMA-3 8B	0.5991	0.5671	0.5677	0.5683	0.5865	**0.5969**	0.5887

As illustrated in inequality (6) and (8), Clapping-**FC** and Clapping-**FU** can achieve an asymptotic convergence rate of $\mathcal{O}(1/\sqrt[3]{T})$ and $\mathcal{O}(1/\sqrt{T})$, respectively. This complements prior pipeline-parallel compression analyses that either require sample-wise activation storage (e.g., AQ-SGD) or impose extra unbiasedness assumptions on the compression-induced gradient estimator.

Trade-off in the selection of p_t. The lazy-sampling coefficient p_t controls the trade-off between sample refresh and sample reuse. A smaller p_t gives error feedback more consecutive updates on the same sample and, for Clapping-**FU**, also reduces the frequency of raw refresh communications. However, p_t cannot be arbitrarily small. For Clapping-**FU**, the proof uses a fixed positive refresh probability $p = p_0 \in (0, 1]$ independent of T; the degenerate case $p = 0$ is not covered. When $p \to 0$, the repeated-sample variance term remains of order $\Theta(\sigma^2)$, so the proof no longer yields the same stationarity guarantee. For Clapping-**FC**, the σ^2 term in (5) equals $\mathcal{O}(1)$ when $p \to 0$, leading to non-convergence. Moreover, an excessively small p_t may hurt generalization, since some samples can be over-used while others are rarely seen.

In practice, p_t should be treated as a refresh-frequency parameter rather than a delicate constant. A simple rule is to start from a moderate value such as $p_t \in [0.4, 0.5]$. If training becomes unstable or validation quality degrades, one can increase p_t to refresh data more often; if training is stable but communication remains the bottleneck, one can decrease p_t to allow more sample reuse and stronger error feedback. Existing sweeps in Appendix H support that moderate values are usually preferable to very small or very large refresh probabilities; see Sec. 5 for further details.

Impact of compression error accumulation. Evidently, more compression entails more accumulated error and results in slower convergence. It is also noteworthy that Clapping-**FU** requires $\mathcal{O}\left(1/\varepsilon^2 + 1/\left(\varepsilon(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}\right)\right)$ iterations to approach an ε-stationary point. Hence, the impact of compression in our proposed method can be asymptotically neglected, as this term dominates the convergence rate—a finding consistent with the result in [12]. Moreover, the exponential accumulation of compression error with the increase in the number of workers E has also been documented in prior analyses [49, 29]. However, the adverse effect of communication compression persists in the algorithms presented by [67]. Ablation studies on the training performance and the number of workers E can be found in Appendix H.3.3.

Convergence with Adam. We provide the convergence analysis for both Clapping-**FC** and Clapping-**FU** with the Adam optimizer in Appendix D.2. It can be shown that Clapping with Adam shares the same convergence rate as that with momentum SGD. Such a convergence guarantee illustrates that Clapping is suitable for LLM pre-training and fine-tuning tasks, which significantly extends the applicability of our proposed algorithm.

Convergence in large batch scenario. We provide the convergence analysis of Clapping with the batch size $B > 1$ in Appendix E.2. It can be shown that Clapping-**FC** achieves a convergence rate of $\mathcal{O}(1/\sqrt[3]{BT})$ and Clapping-**FU** achieves an convergence rate of $\mathcal{O}(1/\sqrt{BT})$, with the impact of compression error remaining similar to the case when $B = 1$. Additionally, the selection of the lazy sampling coefficient can also extended to the large-batch scenario.

Convergence and error propagation in multi-worker scenario. Eq. (6) and (8) illustrate the convergence result of Algorithm 4 with multiple workers with the analysis of the error accumulation in the compression. And we also present a detailed analysis of error propagation in Appendix C. Such a result remedies the shortage of [67] in multi-worker compression. See Appendix F.2 for more detail. Furthermore, we provide an empirical analysis of error propagation across multiple workers; refer to Appendix H.3.3 for experimental details and results.

5 Experiments

We present experiments to validate the performance of Clapping. Unless otherwise specified, Clapping refers to Clapping-**FC** in this section, since FC is the harsher always-compress endpoint and gives the lower fixed-budget payload. The FU variant is complementary: it uses raw refreshes to reset error-feedback states and can improve optimization stability or time-to-target performance; we therefore report FU in the communication-bound pre-training study where this distinction is most relevant. Additional experimental details, extended results, and supplementary experiments on synthetic logistic regression, ResNet-18/CIFAR-10, and LLM fine-tuning/pre-training are provided in Appendix H.

Fine-tuning on GLUE benchmark. We fine-tune pre-trained RoBERTa-large [35] on the GLUE benchmark [64] for 10 epochs using communication compression algorithms, including Clapping and direct compression with EF21 [50], and compare them with uncompressed fine-tuning on two NVIDIA A800 GPUs. A TopK compressor retains 30% of elements at the network midpoint. As shown in Table 2, Clapping outperforms EF21 in most tasks and achieves the highest average score.

Fine-tuning on Wikitext. We fine-tune pre-trained LLaMA-2-7B [61] and LLaMA-3-8B [14] models on the Wikitext-2 dataset [38]. The model is split at the midpoint, and a Top-K compressor [69] retains 5% of elements. Table 3 shows that Clapping outperforms other algorithms, including direct compression, EF21, and AQ-SGD. By tuning p, Clapping-**FC** achieves $\approx 95\%$ communication reduction with no more than a 0.5 percentage-point absolute drop in evaluation accuracy in these fine-tuning tasks.

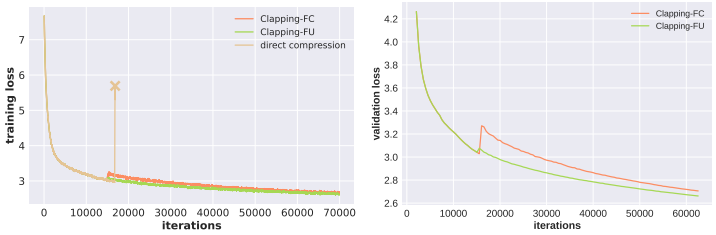

Figure 3: The training loss (left) and evaluation loss (right) with various communication compression algorithms for pre-training LLaMA-2 1B. Here the pre-training task under direct compression fails to converge.

Pre-training LLaMA-2 1B under a communication-bound pipeline. We pre-train a LLaMA-2 1B model [61] on C4-en [48] with pipeline parallelism under a fixed bandwidth cap (100 MB/s) and activation/gradient compression (Top-K sparsification with quantization). Following [75, 28]'s setup (see Appendix H.4.2 for more details), Figure 3 reports the training loss and evaluation loss *versus training iterations*: direct compression fails to converge, while both Clapping-**FU** and Clapping-**FC** remain stable and track the uncompressed reference trajectory. This highlights Clapping as an *optimization-stability* mechanism for pipeline-parallel communication compression in communication-bound regimes; additional implementation details are provided in Appendix H.4.2.

6 Conclusions

We propose Clapping, a communication-compression framework for pipeline-parallel distributed optimization that combines error feedback with lazy sampling. Under standard smoothness assumptions and a contractive compressor model, we establish sublinear stationarity rates for two variants: Clapping-**FC** achieves an $O(T^{-1/3})$ rate and Clapping-**FU** achieves an $O(T^{-1/2})$ rate, up to an additional term that explicitly captures compression-error accumulation with the pipeline depth E. Compared with prior pipeline-parallel compression methods, Clapping removes the need for sample-wise activation storage and does not rely on the unbiased compressed-gradient assumption.

Acknowledgments

This work is funded by the National Natural Science Foundation of China (No. 92370121, 12288101), and the National Key Research and Development Program of China (No. 2024YFA1012902). This research is also supported by the AI for Science Institute, Beijing, China. Part of the computational resources were supported by the Centerfor Intelligent Computing and Song-Shan Lake HPC Cen-ter (SSL-HPC) in Great Bay University, Dongguan, China.

References

[1] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[2] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

[3] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[4] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

[5] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[6] T. Chen, B. Xu, C. Zhang, and C. Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[7] Z. Cheng, X. Huang, P. Wu, and K. Yuan. Momentum benefits non-iid federated learning simply and provably. *arXiv preprint arXiv:2306.16504*, 2023.

[8] N. Ding, Y. Chen, B. Xu, Y. Qin, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051. Association for Computational Linguistics, 2023.

[9] M. Diskin, A. Bukhtiyarov, M. Ryabinin, L. Saulnier, A. Sinitsin, D. Popov, D. V. Pyrkin, M. Kashirin, A. Borzunov, A. Villanova del Moral, et al. Distributed deep learning in open collaborations. *Advances in Neural Information Processing Systems*, 34:7879–7897, 2021.

[10] R. D. Evans and T. Aamodt. Ac-gc: Lossy activation compression with guaranteed convergence. *Advances in Neural Information Processing Systems*, 34:27434–27448, 2021.

[11] R. D. Evans, L. Liu, and T. M. Aamodt. Jpeg-act: accelerating deep learning via transform-based lossy compression. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 860–873. IEEE, 2020.

[12] I. Fatkhullin, A. Tyurin, and P. Richtárik. Momentum provably improves error feedback! *Advances in Neural Information Processing Systems*, 36, 2024.

[13] F. Fu, Y. Hu, Y. He, J. Jiang, Y. Shao, C. Zhang, and B. Cui. Don't waste your bits! squeeze activations and gradients for deep neural networks via tinyscript. In *International Conference on Machine Learning*, pages 3304–3314. PMLR, 2020.

[14] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[15] S. Guo, B. Zhang, T. Liu, T. Liu, M. Khalman, F. Llinares, A. Rame, T. Mesnard, Y. Zhao, B. Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

[16] G. He, Y. Cao, Y. He, T. Bai, K. Yuan, and B. Yuan. Tah-quant: Effective activation quantization in pipeline parallelism over slow network. *arXiv preprint arXiv:2506.01352*, 2025.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Y. He, X. Huang, Y. Chen, W. Yin, and K. Yuan. Lower bounds and accelerated algorithms in distributed stochastic optimization with communication compression. *arXiv preprint arXiv:2305.07612*, 2023.

[19] S. Horvóth, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtárik. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pages 129–141. PMLR, 2022.

[20] X. Huang, Y. Chen, W. Yin, and K. Yuan. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. *Advances in Neural Information Processing Systems*, 35:18955–18969, 2022.

[21] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

[22] J. Jiang, F. Fu, T. Yang, and B. Cui. Sketchml: Accelerating distributed machine learning with data sketches. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1269–1284, 2018.

[23] Z. Jiang, X. Chen, X. Huang, X. Du, D. Zhou, and Z. Wang. Back razor: Memory-efficient transfer learning by self-sparsified backpropagation. *Advances in neural information processing systems*, 35:29248–29261, 2022.

[24] S. Jin, G. Li, S. L. Song, and D. Tao. A novel memory-efficient deep learning training framework via error-bounded lossy compression. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 485–487, 2021.

[25] D. P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi. Decentralized deep learning with arbitrary communication compression. *arXiv preprint arXiv:1907.09356*, 2019.

[27] J. Konečnỳ, B. McMahan, and D. Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[28] B. Kong, J. Liang, Y. Liu, R. Deng, and K. Yuan. Cr-net: Scaling parameter-efficient training with cross-layer low-rank structure. *arXiv preprint arXiv:2509.18993*, 2025.

[29] B. Kong, H. Zhang, and K. Yuan. On the convergence of stochastic gradient descent with perturbed forward-backward passes. *arXiv preprint arXiv:2602.20646*, 2026.

[30] M. V. Koroteev. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

[31] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on operating systems design and implementation (OSDI 14)*, pages 583–598, 2014.

[33] Z. Lin, G. Qu, X. Chen, and K. Huang. Split learning in 6g edge networks. *IEEE Wireless Communications*, 2024.

[34] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[36] I. Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[37] I. Markov, A. Vladu, Q. Guo, and D. Alistarh. Quantized distributed training of large models with convergence guarantees. In *International Conference on Machine Learning*, pages 24020–24044. PMLR, 2023.

[38] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[39] A. Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.

[40] M. Nakamoto, S. Zhai, A. Singh, M. Sobol Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[41] D. Narayanan, A. Harlap, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, P. B. Gibbons, and M. Zaharia. Pipedream: Generalized pipeline parallelism for dnn training. In *Proceedings of the 27th ACM symposium on operating systems principles*, pages 1–15, 2019.

[42] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15, 2021.

[43] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[44] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[45] J. Peterson, S. Meylan, and D. Bourgin. Open clone of openai's unreleased webtext dataset scraper, 2019.

[46] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[47] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[48] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[49] S. Ramasinghe, T. Ajanthan, G. Avraham, Y. Zuo, and A. Long. Protocol models: Scaling decentralized training with communication-efficient model parallelism. *arXiv preprint arXiv:2506.01260*, 2025.

[50] P. Richtárik, I. Sokolov, and I. Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34:4384–4396, 2021.

[51] M. I. Rudakov, A. N. Beznosikov, Y. A. Kholodov, and A. V. Gasnikov. Activations and gradients compression for model-parallel training. In *Doklady Mathematics*, volume 108, pages S272–S281. Springer, 2023.

[52] M. Ryabinin, T. Dettmers, M. Diskin, and A. Borzunov. Swarm parallelism: Training large models can be surprisingly communication-efficient. In *International Conference on Machine Learning*, pages 29416–29440. PMLR, 2023.

[53] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, volume 2014, pages 1058–1062. Singapore, 2014.

[54] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

[55] Y. Song, G. Swamy, A. Singh, D. Bagnell, and W. Sun. The importance of online data: Understanding preference fine-tuning via coverage. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[56] Y. Song, Y. Zhou, A. Sekhari, J. A. Bagnell, A. Krishnamurthy, and W. Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

[57] S. U. Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

[58] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified sgd with memory. *Advances in neural information processing systems*, 31, 2018.

[59] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. In *International Conference on Machine Learning*, pages 6155–6165. PMLR, 2019.

[60] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8485–8493, 2022.

[61] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[62] T. Vogels, S. P. Karimireddy, and M. Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[63] X. Wan, P. Qi, G. Huang, M. Lin, and J. Li. Pipeoffload: Improving scalability of pipeline parallelism with memory optimization. *arXiv preprint arXiv:2503.01328*, 2025.

[64] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[65] J. Wang, Y. Lu, B. Yuan, B. Chen, P. Liang, C. De Sa, C. Re, and C. Zhang. Cocktailsgd: Fine-tuning foundation models over 500mbps networks. In *International Conference on Machine Learning*, pages 36058–36076. PMLR, 2023.

[66] J. Wang, H. Qi, A. S. Rawat, S. Reddi, S. Waghmare, F. X. Yu, and G. Joshi. Fedlite: A scalable approach for federated learning on resource-constrained clients. *arXiv preprint arXiv:2201.11865*, 2022.

[67] J. Wang, B. Yuan, L. Rimanic, Y. He, T. Dao, B. Chen, C. Ré, and C. Zhang. Fine-tuning language models over slow networks using activation quantization with guarantees. *Advances in Neural Information Processing Systems*, 35:19215–19230, 2022.

[68] Z. Wang, M. Wen, Y. Xu, Y. Zhou, J. H. Wang, and L. Zhang. Communication compression techniques in distributed deep learning: A survey. *Journal of Systems Architecture*, 142:102927, 2023.

[69] J. Wangni, J. Wang, J. Liu, and T. Zhang. Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[70] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *Advances in neural information processing systems*, 30, 2017.

[71] H. Xu, C.-Y. Ho, A. M. Abdelmoniem, A. Dutta, E. H. Bergou, K. Karatsenidis, M. Canini, and P. Kalnis. Compressed communication for distributed deep learning: Survey and quantitative evaluation. 2020.

[72] H. Yu and J. Wu. Compressing transformers: features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11007–11015, 2023.

[73] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[74] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[75] J. Zhao, Z. Zhang, B. Chen, Z. Wang, A. Anandkumar, and Y. Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Appendix

Contents

1 Introduction **1**

1.1 Limitations in existing works . . . 3

1.2 Contributions . . . 3

2 Prior Arts **4**

2.1 Pipeline-parallel SGD . . . 4

2.2 AQ-SGD . . . 5

3 `Clapping` Algorithm **5**

4 Theoretical Analysis **7**

4.1 Assumptions . . . 7

4.2 `Clapping` convergence . . . 8

5 Experiments **10**

6 Conclusions **10**

A More related works **17**

A.1 Direct compression . . . 17

A.2 Other related works . . . 17

B Proof of the convergence rate **18**

B.1 Notations and proof roadmap . . . 18

B.2 A Master Descent Bound . . . 19

B.2.1 Evaluated-gradient error . . . 20

B.3 Forward-side technical lemmas . . . 27

B.3.1 One-step compression error . . . 27

B.3.2 Forward activation error . . . 28

B.3.3 Forward temporal variation . . . 29

B.4 Variation Bounds . . . 30

B.4.1 Forward variation . . . 30

B.4.2 Exact-adjoint variation . . . 31

B.4.3 Aggregated backward error bound . . . 31

B.4.4 Forward variation to parameter/sample variation . . . 34

B.4.5 Exact-adjoint variation to parameter/sample variation . . . 36

B.5 Master Convergence Bound . . . 38

B.5.1 Convergence rate of `Clapping`-**FU** . . . 38

B.5.2 Convergence rate of `Clapping`-**FC** . . . 39

B.6 A simplified case: convergence rate of the two-worker case 41

C Error propagation analysis for a single forward-backward step **43**

D `Clapping` with Adam optimizer **44**

D.1 Algorithm design 44

D.2 Convergence of `Clapping` with Adam optimizer 45

E `Clapping` with large batch **47**

E.1 Algorithm development 47

E.2 Convergence of `Clapping` with large batch 47

F Additional discussions **51**

F.1 When should communication compression be enabled? 51

F.2 Additional details on multi-worker scenarios 52

F.3 On the relation to vanilla SGD and the $m = 0$ case 52

F.4 Lazy sampling, convergence and generalization 53

F.4.1 Adaptive refresh rates and scope of interpretation 53

G Memory overhead analysis for LLMs pre-training with `Clapping` **53**

H Experimental details **54**

H.1 Synthetic Logistic Regression 54

H.2 Training ResNet-18 on CIFAR-10 55

H.3 Fine-tuning LLMs 55

H.3.1 Fine-tuning on GLUE benchmark 55

H.3.2 Additional Clapping-FU results on GLUE 56

H.3.3 Fine-tuning LLaMA models with TopK compressor 56

H.3.4 Independent runs and WikiText evaluation 57

H.3.5 FC/FU comparison across refresh rates and compressor strengths 57

H.3.6 Pipeline-depth sensitivity on LLaMA-2 13B 58

H.3.7 Sequential non-IID data-order stress test 58

H.3.8 Generated-output quality: instruction tuning and LLM-as-a-Judge 58

H.3.9 Fine-tuning with multiple compression. 59

H.4 Pre-training LLMs 63

H.4.1 Pre-training GPT-2 small with multiple compression. 63

H.4.2 Pre-training LLaMA-2 1B 64

H.4.3 Long-sequence perplexity comparison 65

H.4.4 Gradient fidelity and optimization stability 65

H.4.5 Bandwidth sensitivity and larger-scale pre-training task for LLaMA-2 3B . . 65

H.4.6 Measured LLaMA-2 13B pipeline-parallel step time 66

H.4.7 Ablations across compressor families 66

A More related works

A.1 Direct compression

To reduce communication overhead in pipeline-parallel optimization, we can directly compress the activations and their corresponding gradients, significantly reducing their size [11, 13]. Let $\mathcal{C}(\cdot)$ denote a compressor. The compressed pipeline-parallel SGD follows the same forward-backward procedure as described in Section 2.1, with several core operations slightly modified:

- **Forward**: Each worker e computes activation $y_e^{(t)} = a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})$ and compresses $\tilde{y}_e^{(t)} = \mathcal{C}(y_e^{(t)})$ following the forward order $e = 1, \cdots, E$. We let $y_0^{(t)} = x^{(t)}$.
- **Backward**: Each worker e computes activation gradient $v_{e-1}^{(t)} = \nabla_1 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\top \tilde{v}_e^{(t)}$ and compresses $\tilde{v}_{e-1}^{(t)} = \mathcal{C}(v_{e-1}^{(t)})$ following the backward order $e = E, \cdots, 2$.
- **Update**: Each worker e computes weight gradient $u_e^{(t)} = \nabla_2 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\top \tilde{v}_e^{(t)}$ following the backward order $e = E, \cdots, 1$. We update parameter $w_e^{(t+1)} = w_e^{(t)} - \gamma u_e^{(t)}$.

Pipeline-parallel SGD with direct compression is also illustrated in Fig. 1. Compression errors introduce unique challenges to pipeline-parallel SGD, which differ significantly from those encountered in data-parallel SGD.

Unbiased compressor leads to biased gradient. Direct compression in pipeline-parallel optimization naturally results in biased gradient estimates. Suppose $\mathcal{C}(\cdot)$ is a unbiased compressor such that $\mathbb{E}[\tilde{y}_e] = y_e$, it cannot be guaranteed that $\mathbb{E}[\nabla_j a_e(\tilde{y}_{e-1}, w_e)] = \nabla_j a_e(y_{e-1}, w_e)\,(j = 1, 2)$ due to the composite and non-linear structure of the activation operator $a_e(y_{e-1}, w_e)$, leading to biased activation gradient and weight gradient estimates. However, [11] and [13] impose the impractical assumption of an unbiased compression-induced gradient error to establish convergence guarantees. Without this assumption, direct compression cannot achieve convergence to stationary solutions.

Error propagation. Compression errors in pipeline-parallel SGD propagate in both the forward and backward processes. To illustrate this, we consider a linear neural network mapping $a_e(y_{e-1}, w_e) = W_e y_{e-1}$, where W_e is the weight matrix reshaped from w_e. Let $\tilde{v}_e = v_e + \epsilon_e$, where ϵ_e represents the error incurred during the compression $\tilde{v}_e = \mathcal{C}(v_e)$. From the backward step in direct compression, it holds that

$$v_1 = W_1^\top \left(W_2^\top \left(\cdots \left(W_E^\top v_E + \epsilon_E \right) + \cdots \right) + \epsilon_2 \right) + \epsilon_1.$$

It is evident that the innermost error ϵ_E propagates through layers and can be significantly amplified by $W_1^\top \cdots W_{E-1}^\top$. This error propagation leads to a complex entanglement between the true gradient and the compressed one, severely impairing the performance and stability of the optimization.

A.2 Other related works

Communication compression in data parallelism. Communication compression has demonstrated significant efficacy in data-parallel distributed optimization [71, 68], with two core strategies underpinning its success: *sparsification* and *quantization*. Classical sparsification methods include Top-K [69, 2] and Rand-K [57, 4], while quantization techniques encompass Sign-SGD [53, 3], TurnGrad [70], and natural compression [19]. However, compression inevitably introduces information distortion, which can hinder convergence rates or even lead to non-convergence. To mitigate these challenges, a variety of advanced techniques have been developed, including error feedback [58, 59, 50, 12], hybrid compression [65], and multiple-step compression [20, 18]. Furthermore, [37] introduced weight compression as a complementary approach. Despite these significant advancements, none of these results have been directly extended to pipeline-parallel distributed optimization.

Activation compression. A closely related technique is activation compression, which aims to reduce memory costs during LLM pre-training and fine-tuning [22, 23, 24, 10, 11, 13, 72]. In communication compression for pipeline-parallel optimization, activations must also be compressed to minimize communication overhead. The key distinction lies in the fact that, in memory-efficient settings,

activations are computed precisely during forward propagation, with their compressed copies stored for backward propagation [23], thereby introducing no additional errors during forward propagation. However, in pipeline-parallel communication compression, forward propagation is inherently error-prone, as compressed activations are immediately used for subsequent computations [67], leading to error accumulation across both forward and backward propagation. This poses a significant challenge for algorithm design and theoretical analysis. Recently, [16] also presented an effective activation compression algorithm in pipeline parallelism. While favorable compression performance is achieved on the LLMs fine-tuning tasks, it still lacks theoretical convergence guarantees.

B Proof of the convergence rate

In this section, we prove the convergence results for Algorithm 4. The proof separates optimization error from communication error. We first reduce convergence to the error of the evaluated gradient, then control the forward and backward compression errors, and finally combine these bounds for `Clapping`-**FU** and `Clapping`-**FC** .

B.1 Notations and proof roadmap

To begin with, we use different superscripts to represent variables with/without compression. Suppose the communication compression is taken in the t-th iteration, then we have:

- **Variables with a hat** (like $\hat{y}_e^{(t)}, \hat{v}_e^{(t)}$): The variable is obtained from standard momentum SGD algorithm **without any communication compression**.
- **Variables with a tilde** (like $\tilde{y}_e^{(t)}, \tilde{v}_e^{(t)}$): The variable which is used for the gradient evaluation during the t-th iteration. If the communication compression is taken during the t-th iteration, it denotes variabl obtained from Algorithm 1 - 4 and **has already been compressed**.
- **Variables without any additional superscript** (like $y_e^{(t)}, v_e^{(t)}$): The variable is obtained from Algorithm 1 - 4 and **has not been compressed** during the t-th iteration if the communication compression is taken.

If the communication compression is not taken in this iteration, same variables with different superscripts are all denote the variable without communication, such as $y_e^{(t)} = \hat{y}_e^{(t)} = \tilde{y}_e^{(t)}$.

Moreover, in the t-th iteration, we firstly denote $\mathcal{F}_e^{(t)}$ and $\mathcal{G}_e^{(t)}$ for $e = 1, 2, \cdots, E-1$ as follows:

- $\mathcal{F}_e^{(t)}$: the filtration before the communication from machine e to machine $e+1$ in **forward propagation**,
- $\mathcal{G}_e^{(t)}$: the filtration before the communication from machine $e+1$ to machine e in **backward propagation**.

We also define $\mathcal{F}^{(t)}$ as the σ-algebra generated by all randomness *up to (but excluding)* the fresh sampling decision at iteration t. In particular, $x^{(t)}$ is independent of $\mathcal{F}^{(t)}$ whenever a fresh sample is drawn.

Proof roadmap. The proof has four steps.

Step 1. Lemma 2 gives a descent bound and reduces the problem to controlling the evaluated-gradient error.

Step 2. Lemma 3 decomposes this error into stochastic noise, forward activation error, and backward adjoint error.

Step 3. The technical lemmas in the next two subsections bound these forward/backward errors by parameter changes and, for `Clapping`-**FC**, the sample-change term $\|x^{(t+1)} - x^{(t)}\|^2$.

Step 4. The results are combined in the master convergence bound to prove the rates for `Clapping`-**FU** and `Clapping`-**FC**.

The key difference between the two variants is how they handle refresh iterations. `Clapping`-**FU** sends exact activations/adjoints on refresh iterations, so the corresponding error-feedback states are

reset. Clapping-**FC** compresses every iteration, so a sample refresh can leave an additional term depending on $x^{(t+1)} - x^{(t)}$.

B.2 A Master Descent Bound

We start with the optimization-level part of the proof. The next lemma is a standard smoothness descent bound: it shows that convergence follows once we can control the evaluated-gradient error produced by compression and lazy sampling.

Lemma 2 (Descent Lemma)**.** *Suppose Assumption 1-2 holds and $\gamma \le \frac{1}{2L_{\nabla\ell}}$, then in Algorithm 4 we have:*

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\ \le&\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}\ell(W)\right]-\frac{1}{2\gamma^2 T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+\frac{1}{T}\sum_{t=1}^{T}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right].\end{aligned} \tag{9}$$

Proof. According to the result in [12], we can get from Assumption 2 that:

$$\begin{aligned}\ell(\boldsymbol{w}^{(t+1)}) \le&\ell(\boldsymbol{w}^{(t)})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\\ &-\left(\frac{1}{2\gamma}-\frac{L_{\nabla\ell}}{2}\right)\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\frac{\gamma}{2}\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2.\end{aligned} \tag{10}$$

Then, as $\gamma \le \frac{1}{2L_{\nabla\ell}}$, we have:

$$\begin{aligned}\ell(\boldsymbol{w}^{(t+1)})-\inf_{\boldsymbol{w}}l(\boldsymbol{w}) \le&\ell(\boldsymbol{w}^{(t)})-\inf_{\boldsymbol{w}}\ell(\boldsymbol{w})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\\ &-\frac{1}{4\gamma}\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\frac{\gamma}{2}\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2.\end{aligned} \tag{11}$$

Taking expectation and summation on both sides over $t = 1, 2, \cdots, T$, we can get:

$$\begin{aligned}\mathbb{E}\left[\ell(\boldsymbol{w}^{(T+1)})-\inf_{\boldsymbol{w}}\ell(\boldsymbol{w})\right] \le&\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}l(\boldsymbol{w})\right]-\frac{\gamma}{2}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\ &-\frac{1}{4\gamma}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+\frac{\gamma}{2}\sum_{t=1}^{T}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right].\end{aligned} \tag{12}$$

Finally, we have:

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\ \le&\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}\ell(W)\right]-\frac{1}{2\gamma^2 T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+\frac{1}{T}\sum_{t=1}^{T}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right].\end{aligned}$$

□

Remark 2. *It is noteworthy that one can get $\|\hat{v}_e^{(t)}\|$ is bounded for $e = 1, \cdots, E-1$ from Assumption 1 and the definition of $\hat{v}_e^{(t)}$. If we let $a_e^\circ(v, y, w_e) = \nabla_2 a_e(y, w_e)^\intercal v$, and $L'_{\nabla a} = \mathcal{O}\left(\max\{L_{\nabla a}, L_{\nabla a}\|\hat{v}_e^{(t)}\|\}\right)$, then there exist $L^\circ_{\nabla a} = \mathcal{O}(C_a)$ such that*

$$\begin{aligned}&\left\|a_e^\circ(\hat{v}_e^{(t)}, y, w_e)-a_e^\circ(v', y', w'_e)\right\|^2=\left\|\nabla_2 a_e(y, w_e)^\intercal\hat{v}_e^{(t)}-\nabla_2 a_e(y', w'_e)^\intercal v'\right\|^2\\ \le&2\left\|\nabla_2 a_e(y, w_e)^\intercal\hat{v}_e^{(t)}-\nabla_2 a_e(y', w'_e)^\intercal\hat{v}_e^{(t)}\right\|^2+2\left\|\nabla_2 a_e(y', w'_e)^\intercal\hat{v}_e^{(t)}-\nabla_2 a_e(y', w'_e)^\intercal v'\right\|^2\\ \le&(L'_{\nabla a})^2\left(\|y-y'\|^2+\|w_e-w'_e\|^2\right)+(L^\circ_{\nabla a})^2\left\|\hat{v}_e^{(t)}-v'\right\|^2,\end{aligned} \tag{13}$$

where the last inequality is due to the Lipschitz continuous of $\nabla_2 a_e$, the boundness of $\hat{v}_e^{(t)}$ and the boundness of $\nabla_2 a_e$.

Lemma 2 shows that it remains to control the evaluated-gradient error. All subsequent lemmas are devoted to bounding this term under FC and FU.

B.2.1 Evaluated-gradient error

The descent bound leaves the term $\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\|^2$. The next lemma decomposes this term while keeping the effects of momentum and lazy sampling explicit. After this step, the remaining work is to bound the forward activation error and the backward adjoint error.

Lemma 3. *Suppose Assumption 1 and 2 hold, and let $m_1 = \cdots = m_T = m_{T+1} = m$ as well as $p_3 = \cdots = p_T = p_{T+1} = p$. Moreover, we set $p_2 = 1$. Then, for all $t = 2, \cdots, T+1$ we have:*

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\leq & 32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))} + \frac{1}{m^2}\right)\sum_{e=1}^{E}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\
&+ 8(L_{\nabla a}^{\circ})^2\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2\right] + 8(L'_{\nabla a})^2\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2\right] \\
&+ 4T\sigma^2\frac{(2-p)m - (1-p)m^2}{1-(1-p)(1-m)^2} + \frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)} - \nabla_e \ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned} \tag{14}$$

Proof. In the beginning, we need to claim that we set $p_1 = p_2 = 1$ so that $\psi(t) \geq 2$ and the term $\tilde{u}^{(\psi(t)-1)}$ is always well-defined. This only affects $O(1)$ initial iterations and does not change the asymptotic rates.

For $t = 2, 3, \cdots, T+1$, we denote $\psi(t)$ as the last moment in which the sample is randomly obtained with $\mathcal{D}$ as of the t-th iteration. Specially,

$$\psi(t) \coloneqq \max_{\tau \in \mathbb{S}_t} \tau, \quad \text{where } \mathbb{S}_t \coloneqq \{\tau = 2, 3, \cdots, t | \text{sampling randomly at iteration } \tau\}.$$

Then, with the fact that the $p_2 = 1$, it holds for $\tau = 2, \cdots, t$ that $\Pr(\psi(t) = \tau) = \begin{cases}(1-p)^{t-2}, & \text{if } \tau = 2\\ p(1-p)^{t-\tau}, & \text{else.}\end{cases}$

For $e = 1, 2, \cdots, E-1$ and $t = 2, 3, \cdots, T+1$, the error between the evaluated gradient and the true gradient satisfies:

$$\begin{aligned}
&\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)}) \\
=& \sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\tilde{v}_e^{(\tau)} + (1-m)^{t+1-\psi(t)}\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(t)}) \\
=& \underbrace{\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)}\right)}_{\coloneqq \Xi_{e,1}} \\
&+ \sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{(\tau)})\right) \\
&+ \underbrace{\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)}_{\coloneqq \Xi_{e,2}} + \underbrace{(1-m)^{t+1-\psi(t)}\left(\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)}_{\coloneqq \Xi_{e,3}} \\
&+ (1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right),
\end{aligned} \tag{15}$$

where the first equation is from the momentum update rule. Moreover, we use $\Xi_{e,1}, \Xi_{e,2}, \Xi_{e,3}$ to denote some complex terms, which have been shown in Eq. (15).

Additionally, we denote $\mathcal{F}^{(t)}$ as the filtration before the t-th iteration. Thus, $\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})$ is measureable with respect to $\mathcal{F}^{(\psi(t))}$ for any $e = 1, 2, \cdots, E$. Moreover, the sampling process at the iteration $\psi(t)$ is **independent** with respect to $\mathcal{F}^{(\psi(t))}$. Thus, the non-compressed term $\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})\hat{v}_e^{(\tau)}$ is an unbiased estimation of the gradient $\nabla_e \ell(\boldsymbol{w}^{(\tau)})$ with bounded variance according to Assumption 2. Thus, taking the ℓ_2-norm and conditional expectation with respect to $\mathcal{F}^{(\psi(t))}$ on both sides of Eq. (15), we can obtain:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
=&\mathbb{E}\left[\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\mathbb{E}\Bigg[\Bigg\langle \sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right), \\
&\qquad (1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\Bigg\rangle \Bigg| \mathcal{F}^{(\psi(t))}\Bigg] \\
\leq&2\mathbb{E}\left[\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|\Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right],
\end{aligned}
\tag{16}
$$

where the inequality is due to Cauchy-Schwarz inequality and Assumption 2.

For the second term of the right-hand-side of Eq. (16), it holds that:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq&2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ (1-m)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right],
\end{aligned}
\tag{17}
$$

where the inequality holds is due to the convexity of the ℓ_2-norm.

Moreover, for the last term, it also holds that:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq&2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_e^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ (1-m)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \left(\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned}
\tag{18}
$$

Form Young's inequality, for any $u > 0$ the last term of (18) holds that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \left(\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq &(1+u)\mathbb{E}\left[\left\|\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \left(1+\frac{1}{u}\right)\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned} \tag{19}$$

If we take $u = \dfrac{\left(1-\frac{m}{2}\right)^{t+1-\psi(t)} - (1-m)^{t+1-\psi(t)}}{(1-m)^{t+1-\psi(t)}}$, then $1+u = \dfrac{\left(1-\frac{m}{2}\right)^{t+1-\psi(t)}}{(1-m)^{t+1-\psi(t)}}$ and

$$\begin{aligned}
1+\frac{1}{u} =& 1 + \frac{(1-m)^{t+1-\psi(t)}}{\left(1-\frac{m}{2}\right)^{t+1-\psi(t)} - (1-m)^{t+1-\psi(t)}} = 1 + \frac{(1-m)^{t+1-\psi(t)}}{\frac{m}{2}\left(\sum_{j=0}^{t-\psi(t)}\left(1-\frac{m}{2}\right)^j (1-m)^{t-\psi(t)-j}\right)} \\
\leq& 1 + \frac{2}{m(t+1-\psi(t))}.
\end{aligned}$$

Substituting (19) into Eq. (18) and taking the value of u, we can obtain that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq& 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_e^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \left(1-\frac{m}{2}\right)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ (1-m)^{t+1-\psi(t)}\left(1+\frac{2}{m(t+1-\psi(t))}\right)\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned} \tag{20}$$

Substituting Eq. (17) and Eq. (20) into Eq.(16), we can obtain that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq& 2\mathbb{E}\left[\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_e^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \left(1-\frac{m}{2}\right)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2(1-m)^{t+1-\psi(t)}\left(1+\frac{1}{m(t+1-\psi(t))}\right)\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned} \tag{21}$$

With Assumption 2, we can get:

$$\mathbb{E}\left[\sum_{e=1}^{E}\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_e^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \leq \left(\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\right)^2 \sigma^2. \tag{22}$$

We denote $\Lambda^{(\tau)}$ as the vector resulting from connecting $\nabla_2 a_e(\tilde{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \tilde{v}_e^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_e^{(\tau)}$ for all $e = 1, 2, \cdots, E$ end-to-end. Then, taking the summation on both sides of Eq. (21), it holds that:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq& 2\left(\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\right)^2 \sigma^2 + 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\Lambda^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\tau)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \left(1-\frac{m}{2}\right)^{t+1-\psi(t)} \mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2(1-m)^{t+1-\psi(t)}\left(1+\frac{1}{m(t+1-\psi(t))}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned} \tag{23}
$$

Then, taking the conditional expectation with respect to $\psi(t)$ on both sides of (23), it holds that:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \psi(t)\right] \\
\leq& 2\mathbb{E}\left[\left(\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\right)^2 \sigma^2 \middle| \psi(t)\right] + 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\Lambda^{(\tau)}\right\|^2 \middle| \psi(t)\right] \\
&+ 4\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\tau)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \psi(t)\right] \\
&+ \left(1-\frac{m}{2}\right)^{t+1-\psi(t)} \mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right\|^2 \middle| \psi(t)\right] \\
&+ 2(1-m)^{t+1-\psi(t)}\left(1+\frac{1}{m(t+1-\psi(t))}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \psi(t)\right].
\end{aligned} \tag{24}
$$

Furthermore, taking the expectation over $\psi(t)$, it holds that:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\leq& \sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-\frac{m}{2}\right)^{t+1-\kappa}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\kappa-1)} - \nabla_e \ell(\boldsymbol{w}^{(\kappa-1)})\right\|^2\right] \\
&+ 4\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\sum_{\tau=\kappa}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\tau)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
&+ 2\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-(1-m)^{t-\kappa+1}\right)^2\sigma^2 + 4\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\sum_{\tau=\kappa}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\Lambda^{(\tau)}\right\|^2\right] \\
&+ 2\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)(1-m)^{t+1-\kappa}\left(1+\frac{1}{m(t+1-\kappa)}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)}) - \nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right].
\end{aligned} \tag{25}
$$

Thus,

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&\left(1-\frac{m}{2}\right)\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-\frac{m}{2}\right)^{t-\kappa}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\kappa-1)}-\nabla_e\ell(\boldsymbol{w}^{(\kappa-1)})\right\|^2\right]\\
&+4\sum_{\kappa=2}^{t}\left(\Pr(\psi(t)=\kappa)(1-m)^{t+1-\kappa}\left(1+\frac{1}{m(t+1-\kappa)}\right)\right.\\
&\qquad\left.+\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)m(1-m)^{t+1-\kappa}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)})-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
&+2\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-(1-m)^{t-\kappa+1}\right)^2\sigma^2+4\sum_{\kappa=2}^{t}\left(\sum_{\tau=2}^{\kappa}\Pr(\psi(t)=\tau)\right)m(1-m)^{t-\kappa}\mathbb{E}\left[\left\|\Lambda^{(\kappa)}\right\|^2\right].
\end{aligned}
\tag{26}
$$

Taking summatation over $t=2,\cdots,T+1$, we can obtain that:

$$
\begin{aligned}
&\sum_{t=2}^{T+1}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&\left(1-\frac{m}{2}\right)\sum_{t=2}^{T+1}\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-\frac{m}{2}\right)^{t-\kappa}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\kappa-1)}-\nabla_e\ell(\boldsymbol{w}^{(\kappa-1)})\right\|^2\right]\\
&+4\sum_{t=2}^{T+1}\sum_{\kappa=2}^{t}\left(\Pr(\psi(t)=\kappa)(1-m)^{t+1-\kappa}\left(1+\frac{1}{m(t+1-\kappa)}\right)\right.\\
&\qquad\left.+\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)m(1-m)^{t+1-\kappa}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)})-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
&+2\sum_{t=2}^{T+1}\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)\left(1-(1-m)^{t-\kappa+1}\right)^2\sigma^2\\
&+4\sum_{t=2}^{T+1}\sum_{\kappa=2}^{t}\left(\sum_{\tau=2}^{\kappa}\Pr(\psi(t)=\tau)\right)m(1-m)^{t-\kappa}\mathbb{E}\left[\left\|\Lambda^{(\kappa)}\right\|^2\right].
\end{aligned}
\tag{27}
$$

Here we consider each term of Eq. (27). Firstly, we can obtain that:

$$
\begin{aligned}
&\sum_{\kappa=2}^{t}(1-m)^{t+1-\kappa}\left(\Pr(\psi(t)=\kappa)\left(1+\frac{1}{m(t+1-\kappa)}\right)+m\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)})-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&\sum_{\kappa=2}^{t}(t+1-\kappa)(1-m)^{t+1-\kappa}\left(\Pr(\psi(t)=\kappa)\left(1+\frac{1}{m(t+1-\kappa)}\right)+m\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)\right)\\
&\qquad\cdot\sum_{\tau=\kappa}^{t}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\tau-1)})-\nabla\ell(\boldsymbol{w}^{(\tau)})\right\|^2\right]\\
=&\sum_{\kappa=1}^{t-1}(1-m)^{t-\kappa}\left(\Pr(\psi(t)=\kappa+1)(t-\kappa+\frac{1}{m})+m(t-\kappa)\sum_{\tau=2}^{\kappa}\Pr(\psi(t)=\tau)\right)\\
&\qquad\cdot\sum_{\tau=\kappa}^{t-1}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\tau)})-\nabla\ell(\boldsymbol{w}^{(\tau+1)})\right\|^2\right]\\
=&\sum_{\kappa=1}^{t-1}\left[\sum_{\tau=1}^{\kappa}(1-m)^{t-\tau}\left(\Pr(\psi(t)=\tau+1)(t-\tau+\frac{1}{m})+m(t-\tau)\sum_{\iota=2}^{\tau}\Pr(\psi(t)=\iota)\right)\right]\\
&\qquad\cdot\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)})-\nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right].
\end{aligned}
\tag{28}
$$

let $s := t-\tau$, then the coefficient of the term $\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)}) - \nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right]$ in Eq. (28) holds that:

$$
\begin{aligned}
&\sum_{\tau=1}^{\kappa}(1-m)^{t-\tau}\left(\Pr(\psi(t)=\tau+1)(t-\tau+\frac{1}{m})+m(t-\tau)\sum_{\iota=2}^{\tau}\Pr(\psi(t)=\iota)\right)\\
&=\sum_{\tau=2}^{\kappa}(1-m)^{t-\tau}\left(\Pr(\psi(t)=\tau+1)(t-\tau+\frac{1}{m})+m(t-\tau)\sum_{\iota=2}^{\tau}\Pr(\psi(t)=\iota)\right)+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m})\\
&=\sum_{\tau=2}^{\kappa}(1-m)^{t-\tau}\left(p(1-p)^{t-\tau-1}(t-\tau+\frac{1}{m})+m(t-\tau)(1-p)^{t-\tau}\right)+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m})\\
&=\sum_{s=t-\kappa}^{t-2}(1-m)^{s}\left(p(1-p)^{s-1}(s+\frac{1}{m})+ms(1-p)^{s}\right)+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m})\\
&=\sum_{s=t-\kappa}^{t-2}\left[(1-m)^{s}(1-p)^{s-1}s(p+m(1-p))+\frac{p}{m}(1-m)^{s}(1-p)^{s-1}\right]+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m}),
\end{aligned}
\tag{29}
$$

where the first equation is due to the fact that $\sum_{\iota=2}^{\tau}\Pr(\psi(t)=\iota)=(1-p)^{t-\tau}$ for $\tau=2,\cdots,T$. Substituting Eq. (29) into Eq. (28), it holds that: Thus, we can obtain that:

$$
\begin{aligned}
&\sum_{\kappa=2}^{t}(1-m)^{t+1-\kappa}\left(\Pr(\psi(t)=\kappa)\left(1+\frac{1}{m(t+1-\kappa)}\right)+m\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)})-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\le&\sum_{\kappa=1}^{t-1}\left[\sum_{s=t-\kappa}^{t-2}\left[(1-m)^{s}(1-p)^{s-1}s(p+m(1-p))+\frac{p}{m}(1-m)^{s}(1-p)^{s-1}\right]+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m})\right]\\
&\qquad\cdot\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)})-\nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right]\\
\le&(p+m(1-p))\sum_{\kappa=1}^{t-1}\left[\left(\sum_{\tau=t-\kappa}^{t-2}(1-m)^{\tau}(1-p)^{\tau-1}\tau\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)})-\nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right]\right]\\
&+\frac{p}{m}\sum_{\kappa=1}^{t-1}\left[\left(\sum_{\tau=t-\kappa}^{t-2}(1-m)^{\tau}(1-p)^{\tau-1}\right)\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)})-\nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right]\right]\\
&+(1-m)^{t-1}(1-p)^{t-2}(t-1+\frac{1}{m})\sum_{\kappa=1}^{t-1}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa)})-\nabla\ell(\boldsymbol{w}^{(\kappa+1)})\right\|^2\right].
\end{aligned}
\tag{30}
$$

Taking summation on both sides over $t=1,2,\cdots,T$ and use the fact that $p\le m\le 1$, it holds that:

$$
\begin{aligned}
&\sum_{t=1}^{T}\left[\sum_{\kappa=2}^{t}(1-m)^{t+1-\kappa}\left(\Pr(\psi(t)=\kappa)\left(1+\frac{1}{m(t+1-\kappa)}\right)+m\sum_{\tau=2}^{\kappa-1}\Pr(\psi(t)=\tau)\right)\right.\\
&\qquad\left.\cdot\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(\kappa-1)})-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\right]\\
\le&\sum_{t=1}^{T}\left((p+m(1-p))\sum_{\tau=1}^{+\infty}(1-m)^{\tau}(1-p)^{\tau-1}\tau^2+\frac{p+m}{m}\sum_{\tau=1}^{+\infty}(1-m)^{\tau}(1-p)^{\tau-1}\tau+\frac{1}{m}\sum_{\tau=1}^{+\infty}(1-m)^{\tau}(1-p)^{\tau-1}\right)\\
&\qquad\cdot\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right]\\
\le&\sum_{t=1}^{T}\left(\frac{(1-m)(1+(1-p)(1-m)}{(1-(1-p)(1-m))^2}+\frac{p+m}{m}\cdot\frac{1-m}{(1-(1-p)(1-m))^2}+\frac{1}{m}\cdot\frac{1-m}{1-(1-p)(1-m)}\right)\\
&\qquad\cdot\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right]\\
\le&\left(\frac{4(p+m)}{m(1-(1-p)(1-m))^2}+\frac{4}{m(1-(1-p)(1-m))}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right].
\end{aligned}
\tag{31}
$$

Moreover, it also holds that:

$$\begin{aligned}
&\sum_{\kappa=2}^{t} \Pr(\psi(t)=\kappa)\left(1-(1-m)^{t-\kappa+1}\right)^2 \\
=&(1-p)^{t-2}\left(1-(1-m)^{t-1}\right)^2+\sum_{\kappa=3}^{t} p(1-p)^{t-\kappa}\left(1-(1-m)^{t-\kappa+1}\right)^2 \\
\le&\sum_{s=t-2}^{+\infty} p(1-p)^{s}\left(1-(1-m)^{s+1}\right)^2+\sum_{\kappa=3}^{t} p(1-p)^{t-\kappa}\left(1-(1-m)^{t-\kappa+1}\right)^2 \\
=&\sum_{s=0}^{+\infty} p(1-p)^{s}\left(1-(1-m)^{s+1}\right)^2=\frac{(2-p)m^2-(1-p)m^3}{(1-(1-p)(1-m))(1-(1-p)(1-m)^2)}.
\end{aligned} \tag{32}$$

Finally, we can obtain that:

$$\begin{aligned}
&(1-\frac{m}{2})\sum_{t=2}^{T+1}\left(\sum_{\kappa=2}^{t}\Pr(\psi(t)=\kappa)(1-\frac{m}{2})^{t-\kappa}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(\kappa-1)}-\nabla_e\ell(\boldsymbol{w}^{(\kappa-1)})\right\|^2\right]\right) \\
=&(1-\frac{m}{2})\sum_{t=2}^{T+1}\left(\sum_{\kappa=t}^{T+1}\Pr(\psi(\kappa)=t)(1-\frac{m}{2})^{\kappa-t}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t-1)}-\nabla_e\ell(\boldsymbol{w}^{(t-1)})\right\|^2\right]\right).
\end{aligned} \tag{33}$$

We note that

$$\sum_{\kappa=t}^{T+1}(1-m)^{\kappa-t}(1-p)^{\kappa-t}\le\frac{1}{1-(1-m)(1-p)}.$$

For $t\ge 3$, it holds that

$$\sum_{\kappa=t}^{T+1}\Pr(\psi(\kappa)=t)(1-\frac{m}{2})^{\kappa-t}=p\sum_{\kappa=t}^{T+1}(1-p)^{\kappa-t}(1-\frac{m}{2})^{\kappa-t}\le\frac{p}{1-(1-p)(1-\frac{m}{2})}.$$

For $t=2$, it holds that

$$\sum_{\kappa=2}^{T+1}\Pr(\psi(\kappa)=2)(1-\frac{m}{2})^{\kappa-2}=\sum_{\kappa=2}^{T+1}(1-p)^{\kappa-2}(1-\frac{m}{2})^{\kappa-2}\le\frac{1}{1-(1-p)(1-\frac{m}{2})}.$$

Combining Eq. (27), (31), (32), (33) together, we can obtain that:

$$\begin{aligned}
&\sum_{t=2}^{T+1}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\le&2T\sigma^2\frac{(2-p)m^2-(1-p)m^3}{(1-(1-p)(1-m))(1-(1-p)(1-m)^2)}+\frac{4m}{1-(1-p)(1-\frac{m}{2})}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\Lambda^{(t)}\right\|^2\right] \\
&+\frac{(1-\frac{m}{2})p}{1-(1-p)(1-\frac{m}{2})}\sum_{t=2}^{T}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]+\frac{1}{1-(1-p)(1-\frac{m}{2})}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right] \\
&+16\left(\frac{1}{m(1-(1-p)(1-\frac{m}{2}))}+\frac{p+m}{m(1-(1-p)(1-\frac{m}{2}))^2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right].
\end{aligned} \tag{34}$$

Thus, we can obtain that:

$$\begin{aligned}
&\frac{\frac{m}{2}}{1-(1-p)(1-\frac{m}{2})}\sum_{t=1}^{T+1}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\le&2T\sigma^2\frac{(2-p)m^2-(1-p)m^3}{(1-(1-p)(1-\frac{m}{2}))(1-(1-p)(1-m)^2)}+\frac{4m}{1-(1-p)(1-\frac{m}{2})}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\Lambda^{(t)}\right\|^2\right] \\
&+\frac{\frac{m}{2}+1}{1-(1-p)(1-\frac{m}{2})}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right] \\
&+16\left(\frac{1}{m(1-(1-p)(1-\frac{m}{2}))}+\frac{p+m}{m(1-(1-p)(1-\frac{m}{2}))^2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right].
\end{aligned} \tag{35}$$

Then, we can get that:

$$\begin{aligned}
&\sum_{t=1}^{T+1}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\mathbb{E}\left[\sum_{e=1}^{E}\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right]+8\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\Lambda^{(t)}\right\|^2\right]\\
&+32\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})-\nabla\ell(\boldsymbol{w}^{(t+1)})\right\|^2\right].
\end{aligned}\tag{36}$$

Finally, with Eq. (13), it holds that

$$\begin{aligned}
\mathbb{E}\left[\left\|\Lambda^{(t)}\right\|^2\right]=&\mathbb{E}\left[\sum_{e=1}^{E}\left\|\nabla_2 a_e(\tilde{y}_{e-1}^{(t)},w_e^{(t)})^{\mathsf{T}}\tilde{v}_e^{(t)}-\nabla_2 a_e(\hat{y}_{e-1}^{(t)},w_e^{(t)})^{\mathsf{T}}\hat{v}_e^{(t)}\right\|^2\right]\\
\leq&(L_{\nabla a}^{\circ})^2\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]+(L'_{\nabla a})^2\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{y}_{e-1}^{(t)}-\hat{y}_{e-1}^{(t)}\right\|^2\right]\\
=&(L_{\nabla a}^{\circ})^2\sum_{e=1}^{E-1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]+(L'_{\nabla a})^2\sum_{e=1}^{E-1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right],
\end{aligned}\tag{37}$$

where the last equation is from the fact that $\hat{y}_0=\tilde{y}_0=x_0$ and $\hat{v}_E=\tilde{v}_E=1$.

Thus, with Assumption 1, it holds that:

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)\sum_{e=1}^{E}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+8(L_{\nabla a}^{\circ})^2\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]+8(L'_{\nabla a})^2\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]\\
&+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}$$

Thus, we finish the proof of this lemma. □

Remark 3. *Throughout our analysis, we set $p_2=1$, i.e., the second iteration always draws a fresh sample from $\mathcal{D}$. This acts as a benign initialization that keeps the subsequent recursion clean: in particular, it ensures that the "last refresh time" $\psi(t)$ used in the proofs is well-defined for all $t\geq 2$ and avoids carrying arbitrary initial error-feedback states into the main induction. Since this affects only a constant number of iterations, it does not change the asymptotic stationarity rates and incurs negligible additional communication compared with T.*

B.3 Forward-side technical lemmas

We next bound the errors introduced by error-feedback compression. This subsection focuses on the forward pass and on temporal changes of compressed activations. These estimates are also used later when bounding the backward recursion.

B.3.1 One-step compression error

We first isolate one compression operation at one pipeline boundary. The following lemma uses the contractive compressor assumption to relate the current compression error to the change of the uncompressed variable between two consecutive iterations.

Lemma 4 (Compress error of forward and backward propagation)**.** *Suppose Assumption 3 holds, then for $e=1,2,\cdots,E-1$ and $T_1>T_0\geq 1$ we have:*

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2\right]\leq\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|v_e^{(t+1)}-v_e^{(t)}\right\|^2\right]+\frac{\omega_B}{1-\omega_B}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-v_e^{(T_0)}\right\|^2\right],\tag{38a}$$

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2\right]\le\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right]. \tag{38b}$$

Proof. Firstly we consider the term $\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2$ for $e=1,2,\cdots,E-1$. According to Algorithm 3, we have:

$$\begin{aligned}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2\middle|\mathcal{G}_e^{(t)}\right]=&\mathbb{E}\left[\left\|\tilde{v}_e^{(t-1)}+\mathcal{C}(v_e^{(t)}-\tilde{v}_e^{(t-1)})-v_e^{(t)}\right\|^2\middle|\mathcal{G}_e^{(t)}\right]\le\omega_B^2\left\|\tilde{v}_e^{(t-1)}-v_e^{(t)}\right\|^2\\ \le&\omega_B\left\|\tilde{v}_e^{(t-1)}-v_e^{(t-1)}\right\|^2+\frac{\omega_B^2}{1-\omega_B}\left\|v_e^{(t)}-v_e^{(t-1)}\right\|^2,\end{aligned} \tag{39}$$

where the first inequality is due to Assumption 3 and the second inequality uses Young's inequality. Then, taking expectation on both sides and then taking summation over $t=T_0+1,\cdots,T_1+1$, we have:

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2\right]\le\omega_B\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2\right]+\frac{\omega_B^2}{1-\omega_B}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|v_e^{(t+1)}-v_e^{(t)}\right\|^2\right]. \tag{40}$$

Then, we can get:

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-v_e^{(t)}\right\|^2\right]\le\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|v_e^{(t+1)}-v_e^{(t)}\right\|^2\right]+\frac{\omega_B}{1-\omega_B}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-v_e^{(T_0)}\right\|^2\right].$$

Thus, Eq. (38a) holds.

Next, we consider the term $\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2$ for $e=1,2,\cdots,E-1$. According to Algorithm 2, we have:

$$\begin{aligned}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2\middle|\mathcal{F}_e^{(t)}\right]=&\mathbb{E}\left[\left\|\tilde{y}_e^{(t-1)}+\mathcal{C}(y_e^{(t)}-\tilde{y}_e^{(t-1)})-y_e^{(t)}\right\|^2\middle|\mathcal{F}_e^{(t)}\right]\le\omega_F^2\left\|\tilde{y}_e^{(t-1)}-y_e^{(t)}\right\|^2\\ \le&\omega_F\left\|\tilde{y}_e^{(t-1)}-y_e^{(t-1)}\right\|^2+\frac{\omega_F^2}{1-\omega_F}\left\|y_e^{(t)}-y_e^{(t-1)}\right\|^2,\end{aligned} \tag{41}$$

where the first inequality is due to Assumption 3 and the second inequality uses Young's inequality. Then, taking expectation on both sides and then taking summation over $t=T_0+1,\cdots,T_1+1$, we have:

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2\right]\le\omega_F\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2\right]+\frac{\omega_F^2}{1-\omega_F}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]. \tag{42}$$

Then, we can get:

$$\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-y_e^{(t)}\right\|^2\right]\le\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right].$$

Thus, Eq. (38b) holds. □

B.3.2 Forward activation error

We now propagate the one-step forward compression error across pipeline stages. The goal is to compare the activation used by `Clapping` with the activation that would have been obtained by the uncompressed pipeline.

Lemma 5. *Suppose Assumption 1 and 3 holds, then for any $T_1>T_0\ge1$ we have:*

$$\begin{aligned}\sum_{e=1}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]\le&\sum_{e=1}^{E-1}\sum_{\iota=e}^{E-1}2(2L_a^2)^{\iota-e}\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\ &+\sum_{e=1}^{E-1}\sum_{\iota=e}^{E-1}2(2L_a^2)^{\iota-e}\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right].\end{aligned} \tag{43}$$

Proof. For $1 \le e \le E-1$, we have:

$$\begin{aligned}\left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2 \le& 2\left\|\tilde{y}_e^{(t)} - y_e^{(t)}\right\|^2 + 2\left\|y_e^{(t)} - \hat{y}_e^{(t)}\right\|^2 \\ =& 2\left\|\tilde{y}_e^{(t)} - y_e^{(t)}\right\|^2 + 2\left\|a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)}) - a_e(\hat{y}_{e-1}^{(t)}, w_e^{(t)})\right\|^2 \\ \le& 2\left\|\tilde{y}_e^{(t)} - y_e^{(t)}\right\|^2 + 2L_a^2\left\|\tilde{y}_{e-1}^{(t)} - \hat{y}_{e-1}^{(t)}\right\|^2 \le \cdots \le \sum_{\iota=1}^{e} 2(2L_a^2)^{e-\iota}\left\|\tilde{y}_\iota^{(t)} - y_\iota^{(t)}\right\|^2,\end{aligned} \tag{44}$$

where the second inequality is due to Assumption 1.

Taking expectation and then taking summation on both sides of (44) over $t = T_0+1, \cdots, T_1+1$, then we have:

$$\begin{aligned}&\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2\right] \le \sum_{\iota=1}^{e} 2(2L_a^2)^{e-\iota}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_\iota^{(t)} - y_\iota^{(t)}\right\|^2\right] \\ \le& \sum_{\iota=1}^{e} 2(2L_a^2)^{e-\iota}\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_\iota^{(t+1)} - y_\iota^{(t)}\right\|^2\right] + \sum_{\iota=1}^{e} 2(2L_a^2)^{e-\iota}\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_\iota^{(T_0)} - y_\iota^{(T_0)}\right\|^2\right],\end{aligned} \tag{45}$$

where the last inequality is due to the result of (38b).

Taking summation on both sides of (45) over e, we can get:

$$\begin{aligned}\sum_{e=1}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2\right] \le& \sum_{e=1}^{E-1}\sum_{\iota=e}^{E-1} 2(2L_a^2)^{\iota-e}\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] \\ &+ \sum_{e=1}^{E-1}\sum_{\iota=e}^{E-1} 2(2L_a^2)^{\iota-e}\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)} - y_e^{(T_0)}\right\|^2\right].\end{aligned}$$

□

B.3.3 Forward temporal variation

The backward analysis needs to compare compressed activations from two consecutive iterations. The next lemma converts this temporal variation into the corresponding change of the uncompressed activation, plus the initial compression error.

Lemma 6 (Temporal variation of compressed activations)**.** *Suppose Assumption 3 and Eq.* (38b) *holds, then for any $T_1 > T_0 \ge 1$ we have:*

$$\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t+1)} - \tilde{y}_e^{(t)}\right\|^2\right] \le \frac{8}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] + \frac{8}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)} - y_e^{(T_0)}\right\|^2\right]. \tag{46}$$

Proof. For $t = T_0+1, \cdots, T_1+1$, we have:

$$\begin{aligned}&\mathbb{E}\left[\left\|\tilde{y}_e^{(t)} - \tilde{y}_e^{(t-1)}\right\|^2\middle|\mathcal{F}^{(t)}\right] = \mathbb{E}\left[\left\|\mathcal{C}\left(y_e^{(t)} - \tilde{y}_e^{(t-1)}\right)\right\|^2\middle|\mathcal{F}^{(t)}\right] \\ \le& \left(1+\frac{1}{\omega_F}\right)\mathbb{E}\left[\left\|\mathcal{C}\left(y_e^{(t)} - \tilde{y}_e^{(t-1)}\right) - \left(y_e^{(t)} - \tilde{y}_e^{(t-1)}\right)\right\|^2\middle|\mathcal{F}^{(t)}\right] + (1+\omega_F)\left\|y_e^{(t)} - \tilde{y}_e^{(t-1)}\right\|^2 \\ \le& (1+\omega_F)^2\left\|y_e^{(t)} - \tilde{y}_e^{(t-1)}\right\|^2 \le 8\left\|y_e^{(t)} - y_e^{(t-1)}\right\|^2 + 8\left\|y_e^{(t-1)} - \tilde{y}_e^{(t-1)}\right\|^2,\end{aligned} \tag{47}$$

where the first inequality is due to Young's inequality, the second inequality is due to Assumption 3, and the last inequality is due to $\omega_F \in [0,1)$ in Assumption 3.

Then, taking expectation and taking summation on both sides of (47) over $t = T_0+1, \cdots, T_1+1$, we have:

$$\begin{aligned}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t+1)} - \tilde{y}_e^{(t)}\right\|^2\right] \le& 8\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] + 8\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t)} - \tilde{y}_e^{(t)}\right\|^2\right] \\ \le& \frac{8}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] + \frac{8}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)} - y_e^{(T_0)}\right\|^2\right],\end{aligned}$$

where the last inequality is due to Eq. (38b) and the fact that $1 + \frac{\omega_F^2}{(1-\omega_F)^2} \leq \frac{1}{(1-\omega_F)^2}$ as $\omega_F \in [0,1)$. □

For intuition, Appendix B.6 provides a simplified proof for $E = 2$. The general proof below does not rely on the $E = 2$ case.

B.4 Variation Bounds

We now turn to the backward pass and to the variation terms created by the pipeline recursion. The following bounds translate activation and adjoint changes into quantities that can later be absorbed by the descent inequality.

B.4.1 Forward variation

We first relate the backward adjoint error $\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\|^2$ to two simpler sources: local backward compression error and the forward activation error bounded above.

Lemma 7. *Suppose Assumption 1 and 3 holds, then for any $T_1 > T_0 \geq 1$ and $e = 1, 2, \cdots, E-1$ we have:*

$$\begin{aligned}&\sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2\right] \leq 2\frac{\omega_B^2}{(1-\omega_B)^2} \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|v_\iota^{(t+1)} - v_\iota^{(t)}\right\|^2\right] \\ &+(2(L'_{\nabla a})^2) \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2\right] + 2\frac{\omega_B}{1-\omega_B} \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \mathbb{E}\left[\left\|\tilde{v}_\iota^{(T_0)} - v_\iota^{(T_0)}\right\|^2\right].\end{aligned} \tag{48}$$

Proof. For $1 \leq e \leq E-1$, we have:

$$\begin{aligned}\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \leq& 2\left\|\tilde{v}_e^{(t)} - v_e^{(t)}\right\|^2 + 2\left\|v_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \\ =& 2\left\|\tilde{v}_e^{(t)} - v_e^{(t)}\right\|^2 + 2\left\|\nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, w_{e+1}^{(t)})^\top \tilde{v}_{e+1}^{(t)} - \nabla_1 a_{e+1}(\hat{y}_e^{(t)}, w_{e+1}^{(t)})^\top \hat{v}_{e+1}^{(t)}\right\|^2 \\ \leq& 2\left\|\tilde{v}_e^{(t)} - v_e^{(t)}\right\|^2 + 2(L'_{\nabla a})^2 \left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2 + 2(L_{\nabla a}^{\circ})^2 \left\|\tilde{v}_{e+1}^{(t)} - \hat{v}_{e+1}^{(t)}\right\|^2,\end{aligned} \tag{49}$$

where the second inequality is due to (13). Then, we can get:

$$\begin{aligned}\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \leq& \sum_{\iota=e}^{E-1} 2(2(L_{\nabla a}^{\circ})^2)^{\iota-e} \left[\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 + (L'_{\nabla a})^2 \left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2\right] \\ \leq& 2\sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 + (2(L'_{\nabla a})^2) \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2.\end{aligned} \tag{50}$$

Taking expectation and then taking summation on both sides of (50) over $t = T_0 + 1, \cdots, T_1 + 1$, then we gan get

$$\begin{aligned}&\sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2\right] \\ \leq& 2\sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2\right] + (2(L'_{\nabla a})^2) \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2\right] \\ \leq& 2\frac{\omega_B^2}{(1-\omega_B)^2} \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|v_\iota^{(t+1)} - v_\iota^{(t)}\right\|^2\right] + 2\frac{\omega_B}{1-\omega_B} \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \mathbb{E}\left[\left\|\tilde{v}_\iota^{(T_0)} - v_\iota^{(T_0)}\right\|^2\right] \\ &+ (2(L'_{\nabla a})^2) \sum_{\iota=e}^{E-1} (2(L_{\nabla a}^{\circ})^2)^{\iota-e} \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2\right],\end{aligned} \tag{51}$$

where the last inequality is due to Eq. (38a). □

B.4.2 Exact-adjoint variation

The previous bound contains the temporal difference $\|v_e^{(t+1)} - v_e^{(t)}\|^2$. We next control this term by changes in the next-stage adjoint, the compressed activation, and the local model block.

Lemma 8. *Suppose Assumption 1 and* (46) *hold, then for any $T_1 > T_0 \geq 1$ and $e = 1, 2, \cdots, E-1$ we have:*

$$\begin{aligned}
&\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|v_e^{(t+1)} - v_e^{(t)}\right\|^2\right] \leq 5(L_{\nabla a}^{\circ})^2 \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(t)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] + 5(L_{\nabla a}^{\circ})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\hat{v}_{e+1}^{(t+1)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] \\
&+5(L'_{\nabla a})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|w_{e+1}^{(t+1)} - w_{e+1}^{(t)}\right\|^2\right] + \frac{40(L'_{\nabla a})^2}{(1-\omega_F)^2} \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] \\
&+\frac{40(L'_{\nabla a})^2}{1-\omega_F} \mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)} - y_e^{(T_0)}\right\|^2\right] + \frac{5}{2}(L_{\nabla a}^{\circ})^2 \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(T_0)} - \hat{v}_{e+1}^{(T_0)}\right\|^2\right].
\end{aligned} \tag{52}$$

Proof. Firstly, we consider the case of $e = 1, \cdots, E-1$, we have:

$$\begin{aligned}
&\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|v_e^{(t+1)} - v_e^{(t)}\right\|^2\right] \\
=&\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\nabla_1 a_{e+1}(\tilde{y}_e^{(t+1)}, w_{e+1}^{(t+1)})^\top \tilde{v}_{e+1}^{(t+1)} - \nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, w_{e+1}^{(t)})^\top \tilde{v}_{e+1}^{(t)}\right\|^2\right] \\
\leq&\frac{5}{2}\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\nabla_1 a_{e+1}(\tilde{y}_e^{(t+1)}, w_{e+1}^{(t+1)})^\top \tilde{v}_{e+1}^{(t+1)} - \nabla_1 a_{e+1}(\tilde{y}_e^{(t+1)}, w_{e+1}^{(t+1)})^\top \hat{v}_{e+1}^{(t+1)}\right\|^2\right] \\
&+\frac{5}{2}\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, w_{e+1}^{(t)})^\top \tilde{v}_{e+1}^{(t)} - \nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, w_{e+1}^{(t)})^\top \hat{v}_{e+1}^{(t)}\right\|^2\right] \\
&+5\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\nabla_1 a_{e+1}(\tilde{y}_e^{(t+1)}, w_{e+1}^{(t+1)})^\top \hat{v}_{e+1}^{(t+1)} - \nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, w_{e+1}^{(t)})^\top \hat{v}_{e+1}^{(t)}\right\|^2\right].
\end{aligned} \tag{53}$$

Then, we have:

$$\begin{aligned}
&\sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|v_e^{(t+1)} - v_e^{(t)}\right\|^2\right] \\
\leq&\frac{5}{2}(L_{\nabla a}^{\circ})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(t+1)} - \hat{v}_{e+1}^{(t+1)}\right\|^2\right] + \frac{5}{2}(L_{\nabla a}^{\circ})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(t)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] \\
&+5(L_{\nabla a}^{\circ})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\hat{v}_{e+1}^{(t+1)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] + 5(L'_{\nabla a})^2 \sum_{t=T_0}^{T_1} \left(\mathbb{E}\left[\left\|\tilde{y}_e^{(t+1)} - \tilde{y}_e^{(t)}\right\|^2\right] + \mathbb{E}\left[\left\|w_{e+1}^{(t+1)} - w_{e+1}^{(t)}\right\|^2\right]\right) \\
\leq&5(L_{\nabla a}^{\circ})^2 \sum_{t=T_0+1}^{T_1+1} \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(t)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] + 5(L_{\nabla a}^{\circ})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|\hat{v}_{e+1}^{(t+1)} - \hat{v}_{e+1}^{(t)}\right\|^2\right] \\
&+5(L'_{\nabla a})^2 \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|w_{e+1}^{(t+1)} - w_{e+1}^{(t)}\right\|^2\right] + \frac{40(L'_{\nabla a})^2}{(1-\omega_F)^2} \sum_{t=T_0}^{T_1} \mathbb{E}\left[\left\|y_e^{(t+1)} - y_e^{(t)}\right\|^2\right] \\
&+\frac{40(L'_{\nabla a})^2}{1-\omega_F} \mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)} - y_e^{(T_0)}\right\|^2\right] + \frac{5}{2}(L_{\nabla a}^{\circ})^2 \mathbb{E}\left[\left\|\tilde{v}_{e+1}^{(T_0)} - \hat{v}_{e+1}^{(T_0)}\right\|^2\right],
\end{aligned} \tag{54}$$

where the first inequality is due to Eq. (13) and the last inequality is due to Eq. (46). □

B.4.3 Aggregated backward error bound

Combining the two previous recursions gives a pipeline-level bound for all backward adjoint errors. This is where the compression constants begin to accumulate with the number of pipeline stages.

Lemma 9. *Suppose Lemma 7 and Lemma 8 are all satisfied, then there exist coefficients* $C^{\circ}_{y,e}, C^{\circ}_{W,e}, C^{\circ}_{v,e}, C^{1}_{v,e}, C^{2}_{v,e}, C^{\circ}_{\theta,e}, C^{1}_{\theta,e} \geq 0$ *(which has been defined by Eq.* (59)*) for each* $e = 1, 2, \cdots, E-1$ *and* $T_1 > T_0 \geq 1$ *such that:*

$$\begin{aligned}
&\sum_{e=1}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\leq&\sum_{e=1}^{E-1}C^{\circ}_{y,e}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C^{\circ}_{W,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C^{\circ}_{v,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C^{\circ}_{\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C^{1}_{\theta,e}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C^{1}_{v,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-\hat{v}_e^{(T_0)}\right\|^2\right]+\sum_{e=1}^{E-1}C^{2}_{v,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-v_e^{(T_0)}\right\|^2\right].
\end{aligned}\tag{55}$$

Proof. Combing (48) and (52) together, we have:

$$\begin{aligned}
&\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\leq&\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\sum_{\iota=e}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_{\iota+1}^{(t)}-\hat{v}_{\iota+1}^{(t)}\right\|^2\right]+\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\sum_{\iota=e}^{E-1}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\hat{v}_{\iota+1}^{(t+1)}-\hat{v}_{\iota+1}^{(t)}\right\|^2\right]\\
&+(L^*_{\nabla a})^2(2(L'_{\nabla a})^2)\sum_{\iota=e}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_{\iota}^{(t)}-\hat{y}_{\iota}^{(t)}\right\|^2\right]+\frac{20(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\sum_{\iota=e}^{E-1}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_{\iota+1}^{(t+1)}-w_{\iota+1}^{(t)}\right\|^2\right]\\
&+\frac{80(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}\sum_{\iota=e}^{E-1}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_{\iota}^{(t+1)}-y_{\iota}^{(t)}\right\|^2\right]+\frac{80(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)}\sum_{\iota=e}^{E-1}\mathbb{E}\left[\left\|\tilde{y}_{\iota}^{(T_0)}-y_{\iota}^{(T_0)}\right\|^2\right]\\
&+\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\sum_{\iota=e}^{E-2}\mathbb{E}\left[\left\|\tilde{v}_{\iota+1}^{(T_0)}-\hat{v}_{\iota+1}^{(T_0)}\right\|^2\right]+2(L^*_{\nabla a})^2\frac{\omega_B}{1-\omega_B}\sum_{\iota=e}^{E-1}\mathbb{E}\left[\left\|\tilde{v}_{\iota}^{(T_0)}-v_{\iota}^{(T_0)}\right\|^2\right]
\end{aligned}\tag{56}$$

holds for $e = 1, 2, \cdots, E-1$, where $(L^*_{\nabla a})^2 := \max\{1, (2(L^{\circ}_{\nabla a})^2)^E\} \geq 0$.

For $e = 1, 2, \cdots, E-1$, define $\{C_{v,e}\}$ as:

$$C_{v,1} = 1, \quad C_{v,e} = 1 + \sum_{\iota=1}^{e-1} C_{v,\iota}\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2} \quad (e = 2, \cdots, E-1). \tag{57}$$

Then taking summation on both sides of (56) over $e = 1, 2, \cdots, E-1$, we have:

$$\begin{aligned}
\sum_{e=1}^{E-1}C_{v,e}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\leq&\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e}C_{v,\iota}(L^*_{\nabla a})^2(2(L'_{\nabla a})^2)\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E}\left(\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{20(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e}C_{v,\iota}\frac{80(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{5(L^*_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2}\right)\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-\hat{v}_e^{(T_0)}\right\|^2\right]+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e}C_{v,\iota}2(L^*_{\nabla a})^2\frac{\omega_B}{1-\omega_B}\right)\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-v_e^{(T_0)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(\sum_{\iota=1}^{e}C_{v,\iota}\frac{80(L^*_{\nabla a})^2(L'_{\nabla a})^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)}\right)\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right].
\end{aligned}\tag{58}$$

Then, with the definition of $C_{v,e}$ in Eq. (57), we have:

$$\begin{aligned}
&\sum_{e=1}^{E-1}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}C_{y,e}^{\circ}\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C_{W,e}^{\circ}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C_{v,e}^{\circ}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{\circ}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{1}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C_{v,e}^{1}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-\hat{v}_e^{(T_0)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{v,e}^{2}\mathbb{E}\left[\left\|\tilde{v}_e^{(T_0)}-v_e^{(T_0)}\right\|^2\right],
\end{aligned}$$

where the coefficients are defined as follows:

$$C_{v,e}^{\circ}=\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{5(L_{\nabla a}^{*})^2\omega_B^2}{(1-\omega_B)^2}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2(e-1)}}\right),\tag{59a}$$

$$C_{v,e}^{1}=\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{5(L_{\nabla a}^{*})^2\omega_B^2}{(1-\omega_B)^2}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2(e-1)}}\right),\tag{59b}$$

$$C_{v,e}^{2}=\sum_{\iota=1}^{e}C_{v,\iota}\frac{2(L_{\nabla a}^{*})^2\omega_B}{1-\omega_B}\le\mathcal{O}\left(\frac{\omega_B}{(1-\omega_B)^{2e-1}}\right),\tag{59c}$$

$$C_{W,e}^{\circ}=\sum_{\iota=1}^{e-1}C_{v,\iota}\frac{20(L_{\nabla a}^{*})^2(L_{\nabla a}')^2\omega_B^2}{(1-\omega_B)^2}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2(e-1)}}\right),\tag{59d}$$

$$C_{y,e}^{\circ}=\sum_{\iota=1}^{e}C_{v,\iota}(L_{\nabla a}^{*})^2(2(L_{\nabla a}')^2)\le\mathcal{O}\left(\frac{1}{(1-\omega_B)^{2(e-1)}}\right),\tag{59e}$$

$$C_{\theta,e}^{\circ}=\sum_{\iota=1}^{e}C_{v,\iota}\frac{80(L_{\nabla a}^{*})^2(L_{\nabla a}')^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2e}(1-\omega_F)^2}\right),\tag{59f}$$

$$C_{\theta,e}^{1}=\sum_{\iota=1}^{e}C_{v,\iota}\frac{80(L_{\nabla a}^{*})^2(L_{\nabla a}')^2\omega_B^2}{(1-\omega_B)^2(1-\omega_F)}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2e}(1-\omega_F)}\right).\tag{59g}$$

□

Now, we try to combine the result of Eq. (55) with Eq. (14). For `Clapping`-**FU**, it holds for any $r=1,2,\cdots,r_0$ that:

$$\begin{aligned}
&\sum_{e=1}^{E-1}\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]=\sum_{e=1}^{E-1}\sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}C_{y,e}^{\circ}\sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C_{W,e}^{\circ}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C_{v,e}^{\circ}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{\circ}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right],
\end{aligned}\tag{60}$$

where it holds due to the fact that in `Clapping`-**FU** at Q_r-iteration it satisfies that

$$\tilde{v}_e^{(Q_r)}=v_e^{(Q_r)}=\hat{v}_e^{(Q_r)},\quad \tilde{y}_e^{(Q_r)}=y_e^{(Q_r)}=\hat{y}_e^{(Q_r)}.$$

Taking summation over r, it holds that:

$$\begin{aligned}
&\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}C_{y,e}^{\circ}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C_{W,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}C_{v,e}^{\circ}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{\circ}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right].
\end{aligned}\tag{61}$$

Plugging Eq. (61) into (14), we can find that:

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
&\le\sum_{e=1}^{E}\left(32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)+8(L_{\nabla a}^{\circ})^2C_{W,e}^{\circ}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L_{\nabla a}^{\circ})^2C_{y,e}^{\circ}\right)\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{v,e}^{\circ}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{\theta,e}^{\circ}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
&\quad+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}\tag{62}$$

Moreover for `Clapping`-**FC**, it holds from Eq. (55) that:

$$\begin{aligned}
&\sum_{e=1}^{E-1}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_e^{(t)}-\hat{v}_e^{(t)}\right\|^2\right]\\
&\le\sum_{e=1}^{E-1}C_{y,e}^{\circ}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C_{W,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}C_{v,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{\theta,e}^{1}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}C_{v,e}^{1}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{v,e}^{2}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right].
\end{aligned}\tag{63}$$

Plugging Eq. (63) into (14), we can find that:

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
&\le\sum_{e=1}^{E}\left(32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)+8(L_{\nabla a}^{\circ})^2C_{W,e}^{\circ}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L_{\nabla a}^{\circ})^2C_{y,e}^{\circ}\right)\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{v,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{\theta,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
&\quad+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right]+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{\theta,e}^{1}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]\\
&\quad+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{v,e}^{1}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{v,e}^{2}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right].
\end{aligned}\tag{64}$$

B.4.4 Forward variation to parameter/sample variation

The bounds above still contain activation-variation terms. To close the proof, we convert these terms into parameter increments and, when samples change, the lazy-sampling difference $\|x^{(t+1)}-x^{(t)}\|^2$. The precise combination that needs to be bounded is

$$\begin{aligned}
&\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L_{\nabla a}^{\circ})^2C_{y,e}^{\circ}\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{\theta,e}^{\circ}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}8(L_{\nabla a}^{\circ})^2C_{v,e}^{\circ}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right].
\end{aligned}$$

The next lemma provides this conversion for any $T_1 > T_0 \ge 1$.

Lemma 10. *Suppose Assumption 1 and 3 holds, then there exist coefficients $C^{\circ}_{y,\theta,e}$ and $C^{1}_{y,\theta,e}$ defined by* (72) *for each $e = 1, 2, \cdots, E-1$ and $T_1 > T_0 \geq 1$ such that:*

$$
\begin{aligned}
&\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L^{\circ}_{\nabla a})^2 C^{\circ}_{y,e}\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L^{\circ}_{\nabla a})^2 C^{\circ}_{\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\leq&\sum_{e=1}^{E-1}L_a^2\left(\sum_{\iota=e}^{E-1}C^{\circ}_{y,\theta,\iota}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\left(\sum_{\iota=1}^{E-1}C^{\circ}_{y,\theta,\iota}L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-1}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(C^{1}_{y,\theta,e}+\sum_{\iota=e+1}^{E-1}\frac{8L_a^2}{1-\omega_F}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e-1}\right)\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right].
\end{aligned}
\tag{65}
$$

Proof. For $e = 2, \cdots, E-1$ and $t = T_0, \cdots, T_1$, we have:

$$
\begin{aligned}
\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2=&\left\|a_e(\tilde{y}_{e-1}^{(t+1)},w_e^{(t+1)})-a_e(\tilde{y}_{e-1}^{(t)},w_e^{(t)})\right\|^2\\
\leq&L_a^2\left\|\tilde{y}_{e-1}^{(t+1)}-\tilde{y}_{e-1}^{(t)}\right\|^2+L_a^2\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2.
\end{aligned}
\tag{66}
$$

Taking expectation and then taking summation on boths over t, we can get:

$$
\begin{aligned}
&\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\leq L_a^2\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\tilde{y}_{e-1}^{(t+1)}-\tilde{y}_{e-1}^{(t)}\right\|^2\right]+L_a^2\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
\leq&\frac{8L_a^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_{e-1}^{(t+1)}-y_{e-1}^{(t)}\right\|^2\right]+\frac{8L_a^2}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_{e-1}^{(T_0)}-y_{e-1}^{(T_0)}\right\|^2\right]\\
&+L_a^2\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right],
\end{aligned}
\tag{67}
$$

where the first inequality is due to Assumption 1 and the second inequality is due to (46).
Then, we have:

$$
\begin{aligned}
&\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\leq&\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-1}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]+\sum_{\iota=1}^{e-1}\frac{8L_a^2}{1-\omega_F}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-1-\iota}\mathbb{E}\left[\left\|\tilde{y}_\iota^{(T_0)}-y_\iota^{(T_0)}\right\|^2\right]\\
&+\sum_{\iota=2}^{e}L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-\iota}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_\iota^{(t+1)}-w_\iota^{(t)}\right\|^2\right].
\end{aligned}
\tag{68}
$$

Moreover, from Assumption 1 we get:

$$
\begin{aligned}
\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2=&\left\|a_1(x^{(t+1)},w_1^{(t+1)})-a_1(x^{(t)},w_1^{(t)})\right\|^2\\
\leq&L_a^2\left\|x^{(t+1)}-x^{(t)}\right\|^2+L_a^2\left\|w_1^{(t+1)}-w_1^{(t)}\right\|^2.
\end{aligned}
\tag{69}
$$

Plugging (69) into (68), then we have:

$$
\begin{aligned}
&\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\leq&\sum_{\iota=1}^{e}L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-\iota}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_\iota^{(t+1)}-w_\iota^{(t)}\right\|^2\right]+L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-1}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]\\
&+\sum_{\iota=1}^{e-1}\frac{8L_a^2}{1-\omega_F}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{e-1-\iota}\mathbb{E}\left[\left\|\tilde{y}_\iota^{(T_0)}-y_\iota^{(T_0)}\right\|^2\right].
\end{aligned}
\tag{70}
$$

Then, consider Eq. (45), we have:

$$
\begin{aligned}
&\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,e}\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L^\circ_{\nabla a})^2C^\circ_{\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,e}\right)\sum_{\iota=1}^{e}2(2L_a^2)^{e-\iota}\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_\iota^{(t+1)}-y_\iota^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,e}\right)\sum_{\iota=1}^{e}2(2L_a^2)^{e-\iota}\frac{\omega_F}{1-\omega_F}\mathbb{E}\left[\left\|\tilde{y}_\iota^{(T_0)}-y_\iota^{(T_0)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}8(L^\circ_{\nabla a})^2C^\circ_{\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}C^\circ_{y,\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C^1_{y,\theta,e}\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right],
\end{aligned}
\tag{71}
$$

where

$$
C^\circ_{y,\theta,e}=8(L^\circ_{\nabla a})^2C^\circ_{\theta,e}+\frac{2\omega_F^2\sum_{\iota=e}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,\iota}\right)(2L_a^2)^{\iota-e}}{(1-\omega_F)^2},
\tag{72a}
$$

$$
C^1_{y,\theta,e}=\frac{2\omega_F\sum_{\iota=e}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,\iota}\right)(2L_a^2)^{\iota-e}}{1-\omega_F}.
\tag{72b}
$$

From (59) we know that:

$$
\begin{aligned}
&C^\circ_{y,e}\le\mathcal{O}\left(\frac{1}{(1-\omega_B)^{2(e-1)}}\right),\quad C^\circ_{\theta,e}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2e}(1-\omega_F)^2}\right).\\
&C^\circ_{y,\theta,e}\le\mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}+\frac{\omega_F^2}{(1-\omega_B)^{2(E-2)}(1-\omega_F)^2}\right),\\
&C^1_{y,\theta,e}\le\mathcal{O}\left(\frac{\omega_F^2}{(1-\omega_B)^{2(E-2)}(1-\omega_F)^2}\right).
\end{aligned}
$$

Then, plugging (70) into (71), we can get:

$$
\begin{aligned}
&\sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2+8(L^\circ_{\nabla a})^2C^\circ_{y,e}\right)\sum_{t=T_0+1}^{T_1+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}8(L^\circ_{\nabla a})^2C^\circ_{\theta,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right]\\
\le&\sum_{e=1}^{E-1}L_a^2\left(\sum_{\iota=e}^{E-1}C^\circ_{y,\theta,\iota}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\left(\sum_{\iota=1}^{E-1}C^\circ_{y,\theta,\iota}L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-1}\right)\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\left(C^1_{y,\theta,e}+\sum_{\iota=e+1}^{E-1}\frac{8L_a^2}{1-\omega_F}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e-1}\right)\mathbb{E}\left[\left\|\tilde{y}_e^{(T_0)}-y_e^{(T_0)}\right\|^2\right].
\end{aligned}
$$

□

B.4.5 Exact-adjoint variation to parameter/sample variation

The last remaining variation term is the change of the no-compression adjoint. Since it follows the exact pipeline recursion, it can be bounded directly by sample changes and parameter increments.

Lemma 11. *Suppose Assumption 1 and 3 holds, then there exist coefficients $C^\circ_{v,x}$ and $C^\circ_{v,w,e}$ defined by* (79) *for each $e=1,2,\cdots,E$ and $T_1>T_0\ge1$ such that:*

$$
\begin{aligned}
&\sum_{e=1}^{E-1}8(L^\circ_{\nabla a})^2C^\circ_{v,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right]\\
\le&C^\circ_{v,x}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]+\sum_{e=1}^{E}C^\circ_{v,w,e}\sum_{t=T_0}^{T_1}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right].
\end{aligned}
\tag{73}
$$

Proof. For every $e = 1, 2, \cdots, E-2$ and $t = 1, 2, \cdots, T$ and $T = T_0, \cdots, T_1$, we have:

$$\begin{aligned}\left\|\hat{v}_e^{(t+1)} - \hat{v}_e^{(t)}\right\|^2 =& \left\|\nabla_1 a_{e+1}(\hat{y}_e^{(t+1)}, W_{e+1}^{(t+1)})^\intercal \hat{v}_{e+1}^{(t+1)} - \nabla_1 a_{e+1}(\hat{y}_e^{(t)}, W_{e+1}^{(t)})^\intercal \hat{v}_{e+1}^{(t)}\right\|^2 \\ \leq& (L'_{\nabla a})^2 \left(\left\|\hat{y}_e^{(t+1)} - \hat{y}_e^{(t)}\right\|^2 + \left\|w_{e+1}^{(t+1)} - w_{e+1}^{(t)}\right\|^2\right) + (L^\circ_{\nabla a})^2 \left\|\hat{v}_{e+1}^{(t+1)} - \hat{v}_{e+1}^{(t)}\right\|^2,\end{aligned} \tag{74}$$

where the first inequality is due to (13).

Then we can get:

$$\begin{aligned}\left\|\hat{v}_e^{(t+1)} - \hat{v}_e^{(t)}\right\|^2 \leq& \sum_{\iota=e}^{E-1} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{\iota-e} \left\|\hat{y}_\iota^{(t+1)} - \hat{y}_\iota^{(t)}\right\|^2 \\ &+ \sum_{\iota=e+1}^{E} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{\iota-e-1} \left\|w_\iota^{(t+1)} - w_\iota^{(t)}\right\|^2,\end{aligned} \tag{75}$$

where the second inequality is due to the definition of $\hat{v}_{E-1}^{(t)}$ and Assumption 1. And we can know that (75) also holds in the case of $e = E-1$.

Then we consider the term $\left\|\hat{y}_e^{(t+1)} - \hat{y}_e^{(t)}\right\|^2$. For $e = 1, 2, \cdots, E-1$, we obtain from Assumption 1 that:

$$\begin{aligned}\left\|\hat{y}_e^{(t+1)} - \hat{y}_e^{(t)}\right\|^2 =& \left\|a_e(\hat{y}_{e-1}^{(t+1)}, w_e^{(t+1)}) - a_e(\hat{y}_{e-1}^{(t)}, w_e^{(t)})\right\|^2 \\ \leq& L_a^2 \left\|\hat{y}_{e-1}^{(t+1)} - \hat{y}_{e-1}^{(t)}\right\|^2 + L_a^2 \left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2.\end{aligned} \tag{76}$$

Then we have:

$$\left\|\hat{y}_e^{(t+1)} - \hat{y}_e^{(t)}\right\|^2 \leq \sum_{\iota=1}^{e} (L_a^2)^{e-\iota+1} \left\|w_\iota^{(t+1)} - w_\iota^{(t)}\right\|^2 + (L_a^2)^e \left\|x^{(t+1)} - x^{(t)}\right\|^2. \tag{77}$$

Combining (75) and (77) together, then we can get:

$$\begin{aligned}&\sum_{e=1}^{E-1} 8(L^\circ_{\nabla a})^2 C^\circ_{v,e} \sum_{t=1}^{T} \mathbb{E}\left[\left\|\hat{v}_e^{(t+1)} - \hat{v}_e^{(t)}\right\|^2\right] \\ \leq& \sum_{e=1}^{E-1} \left(\sum_{\iota=1}^{e} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota+1}\right) \sum_{t=1}^{T} \mathbb{E}\left[\left\|\hat{y}_e^{(t+1)} - \hat{y}_e^{(t)}\right\|^2\right] \\ &+ \sum_{e=2}^{E} \left(\sum_{\iota=1}^{e-1} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota}\right) \sum_{t=1}^{T} \mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\ \leq& \sum_{e=1}^{E-1} \left(\sum_{\iota=1}^{e} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota+1}\right) \\ &\qquad \cdot \sum_{t=1}^{T} \mathbb{E}\left[\sum_{\eta=1}^{e} (L_a^2)^{e-\eta+1} \left\|w_\eta^{(t+1)} - w_\eta^{(t)}\right\|^2 + (L_a^2)^e \left\|x^{(t+1)} - x^{(t)}\right\|^2\right] \\ &+ \sum_{e=2}^{E} \left(\sum_{\iota=1}^{e-1} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota}\right) \sum_{t=1}^{T} \mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\ \leq& C^\circ_{v,x} \sum_{t=1}^{T} \mathbb{E}\left[\left\|x^{(t+1)} - x^{(t)}\right\|^2\right] + \sum_{e=1}^{E} C^\circ_{v,w,e} \sum_{t=1}^{T} \mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right],\end{aligned} \tag{78}$$

where

$$C^\circ_{v,x} = \sum_{e=1}^{E-1} \left(\sum_{\iota=1}^{e} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota+1}\right) (L_a^2)^e, \tag{79a}$$

$$C^\circ_{v,w,e} = \sum_{\iota=1}^{e-1} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{e-\iota} + \sum_{\eta=e}^{E-1} \left(\sum_{\iota=1}^{\eta} 8C^\circ_{v,\iota} (L'_{\nabla a})^2 ((L^\circ_{\nabla a})^2)^{\eta-\iota+1}\right) (L_a^2)^{\eta-e+1}. \tag{79b}$$

And from (59) we know that:

$$C^\circ_{v,x} \leq \mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2(E-2)}}\right), \quad C^\circ_{v,w,e} \leq \mathcal{O}\left(\frac{\omega_B^2}{(1-\omega_B)^{2(E-2)}}\right).$$

□

B.5 Master Convergence Bound

We now combine the descent bound, the evaluated-gradient decomposition, and the compression-error bounds. The two variants use the same ingredients, but refresh iterations affect their error-feedback states differently.

B.5.1 Convergence rate of `Clapping-FU`

For `Clapping`-**FU** , a refresh iteration sends the exact activation/adjoint and resets the corresponding error-feedback state. Thus the initial compression errors vanish at the beginning of each refresh interval, and the sample-change term does not appear inside such intervals.

Theorem 3 (Formal convergence rate of **`Clapping-FU`**)**.** *Suppose Assumption 1-4 hold. Then for Algorithm 4 there exist $\gamma, m, p > 0$ such that:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim \frac{\sigma}{\sqrt{T}} + \frac{1}{T(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}. \tag{80}$$

Proof. Combining (65), and (73) together, it holds that:

$$\begin{aligned}
&\sum_{e=1}^{E-1} 8(L_{\nabla a}^{\circ})^2 C_{v,e}^{\circ} \sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\hat{v}_e^{(t+1)}-\hat{v}_e^{(t)}\right\|^2\right] + \sum_{e=1}^{E-1} 8(L_{\nabla a}^{\circ})^2 C_{\theta,e}^{\circ} \sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_e^{(t+1)}-y_e^{(t)}\right\|^2\right] \\
&+ \sum_{e=1}^{E-1}\left(8(L'_{\nabla a})^2 + 8(L_{\nabla a}^{\circ})^2 C_{y,e}^{\circ}\right)\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_e^{(t)}-\hat{y}_e^{(t)}\right\|^2\right] \qquad (81)\\
\le &\sum_{e=1}^{E}\left(L_a^2\left(\sum_{\iota=e}^{E-1} C_{y,\theta,\iota}^{\circ}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e}\right) + C_{v,w,e}^{\circ}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right].
\end{aligned}$$

Eq. (81) holds is also due to the fact that $x^{(t+1)} = x^{(t)}$ for any $t = Q_r, \cdots, Q_{r+1} - 2$, $y_e^{(Q_r)} = \hat{y}_e^{(Q_r)} = \tilde{y}_e^{(Q_r)}$, and $v_e^{(Q_r)} = \hat{v}_e^{(Q_r)} = \tilde{v}_e^{(Q_r)}$ for any $r = 1, 2, \cdots, r_0$.

Combining (62), (65), and (73) together, we can obtain:

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\le &\sum_{e=1}^{E}\left(32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right) + C_{w,e}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right] \qquad (82)\\
&+ 4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2} + \frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)} - \nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right],
\end{aligned}$$

where

$$\begin{aligned}
C_{w,e} =& 8(L_{\nabla a}^{\circ})^2 C_{W,e}^{\circ} + L_a^2\left(\sum_{\iota=e}^{E-1} C_{y,\theta,\iota}^{\circ}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e}\right) + C_{v,w,e}^{\circ} \\
\lesssim& \frac{1}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^{2(E-1)}}.
\end{aligned} \tag{83}$$

Let $p = p_0 = \mathcal{O}(1)$ as a constant with respect to $\sigma, T, \omega_F, \omega_B$. Then let

$$\begin{aligned}
m &\sim \left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}} + \sigma\sqrt{T}\right)^{-1},\ m \le 1, \\
\gamma &\sim \left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}} + \sigma\sqrt{T}\right)^{-1},\ \gamma \le 1.
\end{aligned}$$

At this time, if γ/m is sufficiently small, we can obtain that

$$32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right) + C_{w,e} - \frac{1}{2\gamma^2} \le 0.$$

Then we have from Eq. (9):

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\lesssim\frac{\sigma}{\sqrt{T}}+\frac{1}{T(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}.$$

□

B.5.2 Convergence rate of `Clapping`-FC

For `Clapping`-**FC** , compression is applied at every iteration, including refresh iterations. Therefore the proof keeps an additional sample-change term $\sum_t \mathbb{E}\|x^{(t+1)}-x^{(t)}\|^2$, which is then controlled by lazy sampling and Assumption 4.

Theorem 4 (Formal convergence rate of `Clapping`-**FC**)**.** *Suppose Assumption 1-4 hold. Then for Algorithm 4 there exist $\gamma, m>0$ such that:*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\lesssim&\frac{\sigma^{\frac{4}{3}}}{T^{\frac{1}{3}}(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}}\\&+\frac{1}{T}\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}+\frac{\omega_F^2+\omega_B}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^2}+\frac{1}{(1-\omega_F)^{2(E-2)-1}}\right).\end{aligned}\tag{84}$$

Proof. Combining (64), (65), and (73) together, we can obtain:

$$\begin{aligned}&\sum_{e=1}^{E}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\\le&\sum_{e=1}^{E}\left(32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)+C_{w,e}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\&+C_x\sum_{t=1}^{T}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{y,\theta,e}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}C_{v,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]\\&+\sum_{e=1}^{E-1}C_{v,\chi,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right]+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}\\&+\frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right],\end{aligned}\tag{85}$$

where

$$\begin{aligned}C_{w,e}=&8(L_{\nabla a}^{\circ})^2C_{W,e}^{\circ}+L_a^2\left(\sum_{\iota=e}^{E-1}C_{y,\theta,\iota}^{\circ}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e}\right)+C_{v,w,e}^{\circ}\\\lesssim&\frac{1}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^{2(E-1)}},&\text{(86a)}\\C_x=&C_{v,x}^{\circ}+\sum_{\iota=1}^{E-1}C_{y,\theta,\iota}^{\circ}L_a^2\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-1}\\\lesssim&\frac{\omega_F^2+\omega_B^2}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^{2(E-1)}},&\text{(86b)}\\C_{y,\theta,e}=&8(L_{\nabla a}^{\circ})^2C_{\theta,e}^{1}+C_{y,\theta,e}^{1}+\sum_{\iota=e+1}^{E-1}\frac{8L_a^2}{1-\omega_F}\left(\frac{8L_a^2}{(1-\omega_F)^2}\right)^{\iota-e-1}\\\lesssim&\frac{\omega_B^2}{(1-\omega_B)^{2e}(1-\omega_F)^2}+\frac{\omega_F^2}{(1-\omega_B)^{2(E-2)}(1-\omega_F)^2}+\frac{1}{(1-\omega_F)^{2(E-e-2)+1}},&\text{(86c)}\\C_{v,e}=&8(L_{\nabla a}^{\circ})^2C_{v,e}^{1},&\text{(86d)}\\C_{v,\chi,e}=&8(L_{\nabla a}^{\circ})^2C_{v,e}^{2}.&\text{(86e)}\end{aligned}$$

Thus, we know that:

$$\begin{aligned}
&\sum_{e=1}^{E-1} C_{y,\theta,e}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1} C_{v,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1} C_{v,\chi,e}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right]\\
\lesssim&\frac{\omega_B}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^2}+\frac{\omega_F^2}{(1-\omega_B)^{2(E-2)}(1-\omega_F)^2}+\frac{1}{(1-\omega_F)^{2(E-2)-1}}.
\end{aligned}\tag{87}$$

Plugging (82) into (9), and use the fact that $\frac{p+m}{1-(1-p)(1-\frac{m}{2})}\le 2$ when $p,m\le 1$, we can get:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\le&\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}} l(\boldsymbol{w})\right]+\frac{C_x}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|x^{(t+1)}-x^{(t)}\right\|^2\right]\\
&+\frac{1}{T}\sum_{e=1}^{E}\left(96L_{\nabla\ell}^2\frac{1}{m^2}+C_{w,e}-\frac{1}{2\gamma^2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\frac{C_{y,\theta,e}}{T}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}\frac{C_{v,e}}{T}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}\frac{C_{v,\chi,e}}{T}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right]\\
&+4\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{mT}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right]+\frac{1}{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}\tag{88}$$

Taking $p=\sqrt{m}$ where h is an undetermined function with respectively to ω_F,ω_B, then the term of stochastic noise satisfies:

$$4\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}\le 4\sigma^2\frac{(2-\sqrt{m})m-(1-\sqrt{m})m^2}{1-(1-\sqrt{m})(1-m)^2}\le 4\sigma^2\cdot 2\sqrt{m}.\tag{89}$$

Let

$$\begin{aligned}
m\sim&\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}+\frac{\sigma^{\frac{4}{3}}T^{\frac{2}{3}}}{(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}}\right)^{-1},\ m\le 1,\\
\gamma\sim&\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}+\frac{\sigma^{\frac{4}{3}}T^{\frac{2}{3}}}{(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}}\right)^{-1},\ \gamma\le 1.
\end{aligned}$$

At this time, if γ/m is sufficiently small, we can obtain that

$$96L_{\nabla\ell}^2\frac{1}{m^2}+C_{w,e}-\frac{1}{2\gamma^2}\le 0.$$

Then with Assumption 4, we have:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla l(W^{(t)})\right\|^2\right]\\
\le&\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}} l(\boldsymbol{w})\right]+C_x\sqrt{m}\varphi^2+8\sigma^2\sqrt{m}+\frac{3}{mT}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right]\\
&+\sum_{e=1}^{E-1}\frac{C_{y,\theta,e}}{T}\mathbb{E}\left[\left\|\tilde{y}_e^{(1)}-y_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}\frac{C_{v,e}}{T}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-\hat{v}_e^{(1)}\right\|^2\right]+\sum_{e=1}^{E-1}\frac{C_{v,\chi,e}}{T}\mathbb{E}\left[\left\|\tilde{v}_e^{(1)}-v_e^{(1)}\right\|^2\right]\\
&+\frac{1}{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(1)})\right\|^2\right]\\
\lesssim&\frac{\sigma^{\frac{4}{3}}}{T^{\frac{1}{3}}(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}}\\
&+\frac{1}{T}\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}+\frac{\omega_F^2+\omega_B}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^2}+\frac{1}{(1-\omega_F)^{2(E-2)-1}}\right).
\end{aligned}$$

□

Remark 4. *It can be observed that the conclusion in Lemma 1 can be directly obtained from* (88)*.*

B.6 A simplified case: convergence rate of the two-worker case

The proof above covers the general E-worker case. For intuition, we also include the two-worker case, where there is only one forward communication and one backward communication. This simplified proof makes the role of error feedback and lazy sampling easier to see.

Lemma 12 (Convergence rate of `Clapping`-**FU** in the case $E = 2$)**.** *Suppose Assumption 1, 2, and 3 hold. Then for* `Clapping`-***FU*** *, there exist $\gamma, m > 0$ such that:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim \frac{\sigma}{\sqrt{T}} + \frac{1}{T(1-\omega_B)(1-\omega_F)}. \tag{90}$$

Proof. Substituting $E = 2$ into (14), we have:

$$\begin{aligned}
&\sum_{t=1}^{T+1}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\leq& 32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))} + \frac{1}{m^2}\right)\sum_{t=1}^{T}\sum_{e=1}^{2}\mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\
&+ 8(L_{\nabla a}^{\circ})^2\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_1^{(t)} - \hat{v}_1^{(t)}\right\|^2\right] + 8(L'_{\nabla a})^2\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right] \\
&+ 4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2} + \frac{3}{m}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)} - \nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned} \tag{91}$$

For the first $T-1$ iterations, suppose $1 = Q_1 < Q_2 < \cdots < Q_{r_0} \leq T$ are the all moments at which the sample is randomly obtained from $\mathcal{D}$. We also denote $Q_{r_0+1} = T+1$.

For any $1 \leq r \leq r_0$, we note that $\tilde{y}_1^{(Q_r)} = y_1^{(Q_r)} = \hat{y}_1^{(Q_r)}$ as we do not compress the activation and gradients in `Clapping`-**FU**. Thus, substituting $E = 2$ into (43), we can get:

$$\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right] = \sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right] \leq C_y^2\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)} - y_1^{(t)}\right\|^2\right]. \tag{92}$$

Taking summation over $r = 1, 2, \cdots, r_0$, it holds that:

$$\begin{aligned}
\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right] &= \sum_{r=1}^{r_0}\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right] \\
&\leq C_y^2\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)} - y_1^{(t)}\right\|^2\right].
\end{aligned} \tag{93}$$

Then, use the fact that ∇a_2 is Lipschitz continous, we have:

$$\left\|v_1^{(t)} - \hat{v}_1^{(t)}\right\|^2 = \left\|\nabla a_2(\tilde{y}_1^{(t)}, w_2^{(t)}) - \nabla a_2(\hat{y}_1^{(t)}, w_2^{(t)})\right\|^2 \leq L_{\nabla a}^2\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2. \tag{94}$$

From $\tilde{v}_1^{(Q_r)} = v_1^{(Q_r)} = \hat{v}_1^{(Q_r)}$, we can obtain that:

$$\begin{aligned}
&\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|\tilde{v}_1^{(t)} - \hat{v}_1^{(t)}\right\|^2\right] \leq 2\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|\tilde{v}_1^{(t)} - v_1^{(t)}\right\|^2\right] + 2\sum_{t=Q_r+1}^{Q_{r+1}}\mathbb{E}\left[\left\|v_1^{(t)} - \hat{v}_1^{(t)}\right\|^2\right] \\
=&2\sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|\tilde{v}_1^{(t)} - v_1^{(t)}\right\|^2\right] + 2\sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|v_1^{(t)} - \hat{v}_1^{(t)}\right\|^2\right] \\
\leq&2\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|v_1^{(t+1)} - v_1^{(t)}\right\|^2\right] + 2L_{\nabla a}^2\sum_{t=Q_r+1}^{Q_{r+1}-1}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)} - \hat{y}_1^{(t)}\right\|^2\right],
\end{aligned} \tag{95}$$

where the second inequality is due to (38a) and (94). Taking summation over r, it holds that:

$$\begin{aligned}&\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_1^{(t)}-\hat{v}_1^{(t)}\right\|^2\right]\\ \leq&2\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|v_1^{(t+1)}-v_1^{(t)}\right\|^2\right]+2L_{\nabla a}^2\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_1^{(t)}-\hat{y}_1^{(t)}\right\|^2\right],\end{aligned}\tag{96}$$

Plugging Eq. (93) and Eq. (96) into Eq. (91), it holds that:

$$\begin{aligned}&\sum_{t=1}^{T+1}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\ \leq&32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)\sum_{t=1}^{T}\sum_{e=1}^{2}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\ &+16(L_{\nabla a}^{\circ})^2\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|v_1^{(t+1)}-v_1^{(t)}\right\|^2\right]\\ &+\left(8(L_{\nabla a}')^2+16(L_{\nabla a}^{\circ})^2L_{\nabla a}^2\right)C_y^2\frac{\omega_F^2}{(1-\omega_F)^2}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]\\ &+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].\end{aligned}\tag{97}$$

Note that $E=2$, thus for any $1\leq r\leq r_0$ we have:

$$\begin{aligned}&\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|v_1^{(t+1)}-v_1^{(t)}\right\|^2\right]=\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\nabla_1 a_2(\tilde{y}_1^{(t+1)},w_2^{(t+1)})-\nabla_1 a_2(\tilde{y}_1^{(t)},w_2^{(t)})\right\|^2\right]\\ \leq&L_{\nabla a}^2\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\tilde{y}_1^{(t+1)}-\tilde{y}_1^{(t)}\right\|^2+\left\|w_2^{(t+1)}-w_2^{(t)}\right\|^2\right]\\ \leq&L_{\nabla a}^2\frac{8}{(1-\omega_F)^2}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]+L_{\nabla a}^2\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|w_2^{(t+1)}-w_2^{(t)}\right\|^2\right],\end{aligned}\tag{98}$$

where the first inequality is due to Assumption 1 and the second inequality is due to (46).

Taking summation over r, it holds that:

$$\begin{aligned}&\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|v_1^{(t+1)}-v_1^{(t)}\right\|^2\right]\\ \leq&L_{\nabla a}^2\frac{8}{(1-\omega_F)^2}\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]+L_{\nabla a}^2\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_2^{(t+1)}-w_2^{(t)}\right\|^2\right],\end{aligned}\tag{99}$$

Then as $x^{(t+1)}=x^{(t)}$ for any $t=Q_r,\cdots,Q_{r+1}-2$, we also have

$$\begin{aligned}&\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]=\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|\hat{y}_1^{(t+1)}-\hat{y}_1^{(t)}\right\|^2\right]\\ =&\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|a_1(x^{(t+1)},w_1^{(t+1)})-a_1(x^{(t)},w_1^{(t)})\right\|^2\right]\leq L_a^2\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|w_1^{(t+1)}-w_1^{(t)}\right\|^2\right],\end{aligned}\tag{100}$$

where the inequality is due to Assumption 1.

Taking summation over r, it holds that:

$$\sum_{r=1}^{r_0}\sum_{t=Q_r}^{Q_{r+1}-2}\mathbb{E}\left[\left\|y_1^{(t+1)}-y_1^{(t)}\right\|^2\right]\leq L_a^2\sum_{t=1}^{T}\mathbb{E}\left[\left\|w^{(t+1)}-w^{(t)}\right\|^2\right],\tag{101}$$

Plugging (99) and (101) into (97), we can get:

$$
\begin{aligned}
&\sum_{t=1}^{T+1}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)\sum_{t=1}^{T}\sum_{e=1}^{2}\mathbb{E}\left[\left\|w_e^{(t+1)}-w_e^{(t)}\right\|^2\right]\\
&+16L_{\nabla_a}^2(L_{\nabla a}^{\circ})^2\frac{\omega_B^2}{(1-\omega_B)^2}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_2^{(t+1)}-w_2^{(t)}\right\|^2\right]\\
&+\left(\left(8(L'_{\nabla a})^2+16(L_{\nabla a}^{\circ})^2L_{\nabla a}^2\right)C_y^2\frac{\omega_F^2}{(1-\omega_F)^2}+128L_{\nabla_a}^2(L_{\nabla a}^{\circ})^2\frac{\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}\right)\\
&\qquad\qquad\cdot L_a^2\sum_{t=1}^{T}\mathbb{E}\left[\left\|w^{(t+1)}-w^{(t)}\right\|^2\right]\\
&+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right]\\
\leq&C_w\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+4T\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{m}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}
\tag{102}
$$

where:

$$
\begin{aligned}
C_w=&32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))}+\frac{1}{m^2}\right)+16L_{\nabla_a}^2(L_{\nabla a}^{\circ})^2\frac{\omega_B^2}{(1-\omega_B)^2}\\
&+\left(8(L'_{\nabla a})^2+16(L_{\nabla a}^{\circ})^2L_{\nabla a}^2\right)L_a^2C_y^2\frac{\omega_F^2}{(1-\omega_F)^2}+128L_{\nabla_a}^2(L_{\nabla a}^{\circ})^2L_a^2\frac{\omega_B^2}{(1-\omega_B)^2(1-\omega_F)^2}.
\end{aligned}
$$

Plugging $E=2$ into (9), combining it with (102), and then taken $p=m$, we can get

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\\
\leq&\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}\ell(\boldsymbol{w})\right]+\frac{1}{T}\left(C_w-\frac{1}{2\gamma^2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]\\
&+4\sigma^2\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2}+\frac{3}{mT}\sum_{e=1}^{2}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)}-\nabla_e\ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}
\tag{103}
$$

Let $p=p_0$ as a constant with the order of $\mathcal{O}(1)$, and let:

$$
m\sim\left(\frac{1}{(1-\omega_B)(1-\omega_F)}+\sigma\sqrt{T}\right)^{-1},\quad \gamma\sim\left(\frac{1}{(1-\omega_B)(1-\omega_F)}+\sigma\sqrt{T}\right)^{-1}\text{ and }m,\gamma\leq 1.
$$

Then, as γ has the same order as m with respect to $\omega_B,\omega_F,\sigma,T$, it holds that $C_w-\frac{1}{2\gamma^2}\leq 0$ if γ/m is sufficiently small.

Thus, we have:

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\lesssim\frac{\sigma}{\sqrt{T}}+\frac{1}{T(1-\omega_B)(1-\omega_F)}.
$$

□

C Error propagation analysis for a single forward-backward step

The following lemma states how the compressed error propagates for one forward and backward step.

Lemma 13. *Suppose Assumptions 1 and 3 hold. Then, for $e = 1, \ldots, E-1$, the error of the forward activation in Algorithm 4 can be bounded above as follows:*

$$\left\|\tilde{y}_e^{(t)} - \hat{y}_e^{(t)}\right\|^2 \leq \sum_{\iota=1}^{e} 2(2L_a^2)^{e-\iota} \left\|\tilde{y}_\iota^{(t)} - y_\iota^{(t)}\right\|^2. \tag{104}$$

For the error of backward gradients in Algorithm 4, there exist constants $L_{\nabla a}^{\circ}, L'_{\nabla a} > 0$ such that:

$$\begin{aligned}\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \leq & 2\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 \\ & + 4(L'_{\nabla a})^2 \sum_{\iota=1}^{E-1}\sum_{s=\max\{e,\iota\}}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{s-e}(2L_a^2)^{s-\iota}\left\|\tilde{y}_\iota^{(t)} - y_\iota^{(t)}\right\|^2.\end{aligned} \tag{105}$$

Proof. Eq. (104) is actually the same as Eq. (44). Then from (50) we can get:

$$\begin{aligned}&\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \\ \leq & 2\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 + (2(L'_{\nabla a})^2)\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\left\|\tilde{y}_\iota^{(t)} - \hat{y}_\iota^{(t)}\right\|^2,\end{aligned} \tag{106}$$

where the last inequality uses the fact that $L'_{\nabla a} \geq L_{\nabla a}$ in Remark 2.

Then, plugging (104) into (106), we can obtain:

$$\begin{aligned}&\left\|\tilde{v}_e^{(t)} - \hat{v}_e^{(t)}\right\|^2 \\ \leq & 2\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 + (2(L'_{\nabla a})^2)\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\sum_{s=1}^{\iota}2(2L_a^2)^{\iota-s}\left\|\tilde{y}_s^{(t)} - y_s^{(t)}\right\|^2 \\ \leq & 2\sum_{\iota=e}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{\iota-e}\left\|\tilde{v}_\iota^{(t)} - v_\iota^{(t)}\right\|^2 + 4(L'_{\nabla a})^2\sum_{\iota=1}^{E-1}\sum_{s=\max\{e,\iota\}}^{E-1}(2(L_{\nabla a}^{\circ})^2)^{s-e}(2L_a^2)^{s-\iota}\left\|\tilde{y}_\iota^{(t)} - y_\iota^{(t)}\right\|^2.\end{aligned} \tag{107}$$

Then we know that Eq. (105) holds for $e = 1, \cdots, E-1$. □

As indicated by Eq. (104), the error in forward activation is directly accumulated due to the compression operations of the preceding machines. Meanwhile, according to Eq. (105), the error in backward gradient evaluation, denoted as $\tilde{v}_e^{(t)} = \tilde{v}_{e+1}^{(t)} \nabla_1 a_{e+1}(\tilde{y}_e^{(t)}, W_e^{(t)})$, arises from two aspects. One aspect pertains to the accumulated errors during the backward compression in preceding machines, while the other relates to the error of the forward activation $\tilde{y}_e^{(t)}$. Moreover, the mathematical formulations presented in Eq. (104) and Eq. (105) jointly demonstrate that compression errors exhibit exponential amplification across distributed computing nodes. Furthermore, this analysis reveals a positive correlation between error magnitude and system complexity: as model dimensionality and parallelism scale increase, communication-induced compression errors emerge as a critical bottleneck in distributed training architectures.

D `Clapping` with Adam optimizer

In this section, we present `Clapping` equipped with Adam [25] optimizer, which can especially fit the need of pre-training and fine-tuning tasks for LLMs.

D.1 Algorithm design

Similar to the standard back-propagation algorithms with Adam optimizer, we introuduce two coefficients $\beta_1, \beta_2 \in (0, 1)$ and use $\nu_e^{(t)}$ and $\upsilon_e^{(t)}$ to record the first- and second-order optimizer states for the parameter $w_e^{(t)}$ for $e = 1, 2, \cdots, E$, respectively. Then we use the following update rules to optimize $w_e^{(t)}$:

$$\tilde{u}_e^{(t)} = (1-\beta_1)\tilde{u}_e^{(t-1)} + \beta_1 \nabla_2 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\intercal \tilde{v}_e^{(t)}, \tag{108a}$$

Algorithm 5 `Clapping` with Adam optimizer

Require: Initialize $\tilde{y}_e^{(0)} = 0, \tilde{v}_e^{(0)} = 0$ for $e = 1, \cdots, E-1$, and $v_e^{(0)} = 0, \tilde{u}_e^{(0)} = 0$ for $e = 1, \cdots, E$. Initialize dataset $\mathcal{D}$, learning rate γ_t, compressor $\mathcal{C}$, and lazy sampling rate $\{p_t\}_{t=1}^T$.

for $t = 1, \cdots, T$ **do**
  $x^{(t)}, f_{\text{FU}}^{(t)} \leftarrow \mathsf{LazySampling}(\mathcal{D}, t, p_t)$;
  $\tilde{y}_0^{(t)} \leftarrow x^{(t)}, \quad \tilde{v}_E^{(t)} \leftarrow 1.$
  for $e = 1, 2, \cdots, E-1$ **do**
    $\tilde{y}_e^{(t)} \leftarrow \mathsf{Forward}_e(\tilde{y}_e^{(t-1)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$.
  end for
  for $e = E, E-1, \cdots, 1$ **do**
    if $e \neq 1$ **then**
      $\tilde{v}_{e-1}^{(t)} \leftarrow \mathsf{Backward}_e(\tilde{v}_{e-1}^{(t-1)}, \tilde{v}_e^{(t)}, \tilde{y}_{e-1}^{(t)}, w_e^{(t)}, f_{\text{FU}}^{(t)})$.
    end if
    Update $v_e^{(t)}$, $\tilde{u}_e^{(t)}$ and $w_e^{(t+1)}$ by (108).
  end for
end for

$$v_e^{(t)} = (1-\beta_2) v_e^{(t-1)} + \beta_2 \left(\nabla_2 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\mathsf{T} \tilde{v}_e^{(t)} \right)^{\odot 2}, \tag{108b}$$

$$\nu_e^{(t)} = \frac{\tilde{u}_e^{(t)}}{\sqrt{v_e^{(t)} + \varepsilon}}, \tag{108c}$$

$$w_e^{(t+1)} = w_e^{(t)} - \gamma \nu_e^{(t)}, \tag{108d}$$

where $\odot 2$ denotes the second moment and $\varepsilon > 0$ is a fixed constant. The division and addition operate in (108c) are all sample-wised. Then we can present `Clapping` with Adam optimizer as Algorithm 5.

D.2 Convergence of `Clapping` with Adam optimizer

In this subsection we present the convergence analysis of `Clapping` with Adam optimizer. Firstly, we need to present an additional assumption for the bounded gradient estimation:

Assumption 5. *There exist $M_1 \geq 0$ such that for each $e = 1, 2, \cdots, E$, the gradient estimation can be bonuded as:*

$$\left\| \nabla_2 a_e(\tilde{y}_{e-1}^{(t)}, w_e^{(t)})^\mathsf{T} \tilde{v}_e^{(t)} \right\| \leq M_1.$$

Remark 5. *We assume the gradient of $a_e(y, \boldsymbol{w})$ is bounded according to Assumption 1. Thus, if there exists a constant ω_0 such that the compressor $\mathcal{C}$ satisfies $\|x - \mathcal{C}(x)\|^2 \leq \omega_0 \|x\|^2$ for any input vector x, then Assumption 5 can be satisfied. Such a bounded property is common for traditional compressors like TopK and quantization.*

With Assumption 5, one can obtain that the second-order optimizer state $v_e^{(t)}$ satisfy that $\left\| v_e^{(t)} \right\| \leq M_1^2$. Then we can directly obtain the following lemma

Lemma 14. *There exists $M > 0$ such that for $t = 1, 2, \cdots, T$ and $e = 1, 2, \cdots, E$ it holds that:*

$$\varepsilon \leq \left\| \sqrt{v_e^{(t)} + \varepsilon} \right\|_1 \leq M,$$

where $\|\cdot\|_1$ denote the ℓ_1-norm.

Lemma 15. *Suppose Assumption 2 and Assumption 5 hold. And the step-size γ satisfies that $\gamma \leq \min\left\{ \frac{1}{4L_{\nabla \ell}}, \frac{1}{4C_\nu} \right\}$, where $C_\nu^2 \leq \max\left\{ (M-1)^2, (\varepsilon - 1)^2 \right\}$ is a constant. Then in Algorithm 5*

we have:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]$$
$$\leq\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}\ell(W)\right]-\frac{1}{2\gamma^2 T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+\frac{2}{T}\sum_{t=1}^{T}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right]. \tag{109}$$

Proof. As ℓ is $L_{\nabla\ell}$-smooth, then it holds that:

$$\begin{aligned}
&\ell(\boldsymbol{w}^{(t+1)})\\
\leq&\ell(\boldsymbol{w}^{(t)})+\langle\nabla\ell(\boldsymbol{w}^{(t)}),\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\rangle+\frac{L_{\nabla\ell}}{2}\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\\
\leq&\ell(\boldsymbol{w}^{(t)})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2-\left(\frac{\gamma}{2}-\frac{L_{\nabla\ell}}{2}\right)\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\frac{\gamma}{2}\sum_{e=1}^{E}\left\|\nu_e^{(t)}-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\\
\leq&\ell(\boldsymbol{w}^{(t)})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2-\left(\frac{\gamma}{2}-\frac{L_{\nabla\ell}}{2}\right)\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\gamma\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\\
&+\gamma\sum_{e=1}^{E}\left\|\left(\sqrt{v_e^{(t)}+\varepsilon}-1\right)\odot\nu_e^{(t)}\right\|^2\\
\leq&\ell(\boldsymbol{w}^{(t)})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2-\left(\frac{\gamma}{2}-\frac{L_{\nabla\ell}}{2}-\frac{C_\nu^2}{\gamma}\right)\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\gamma\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2,
\end{aligned} \tag{110}$$

where $C_\nu^2\leq\max\left\{(M-1)^2,(\varepsilon-1)^2\right\}$.

Then as $\gamma\leq\min\left\{\frac{1}{4L_{\nabla\ell}},\frac{1}{4C_\nu}\right\}$, we obtain that:

$$\begin{aligned}
&\ell(\boldsymbol{w}^{(t+1)})\\
\leq&\ell(\boldsymbol{w}^{(t)})-\frac{\gamma}{2}\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2-\frac{\gamma}{4}\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2+\gamma\sum_{e=1}^{E}\left\|\tilde{u}_e^{(t)}-\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2.
\end{aligned} \tag{111}$$

Similar to the proof of Lemma 2, we can obtain from Eq. (111) that:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]$$
$$\leq\frac{2}{\gamma T}\mathbb{E}\left[\ell(\boldsymbol{w}^{(1)})-\inf_{\boldsymbol{w}}\ell(W)\right]-\frac{1}{2\gamma^2 T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right]+\frac{2}{T}\sum_{t=1}^{T}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)}-\nabla_e\ell(\boldsymbol{w}^{(t)})\right\|^2\right].$$

Then we finish the proof of Eq. (109). □

As the analysis of compress error and error accumulation process are independent with the optimizer, we can directly use the other lemmas in Appendix B. Thus, (82) and (85) also hold for `Clapping`-**FU** and `Clapping`-**FC**, respectively. Finally, we can present the convergence rate of `Clapping` with Adam optimizer as follows:

Lemma 16 (Convergence rate of `Clapping`-**FU** with Adam optimizer)**.** *Suppose Assumption 1-5 hold. Then for Algorithm 5 there exist $\gamma,\beta_1,p>0$ such that:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\lesssim\frac{\sigma}{\sqrt{T}}+\frac{1}{T(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}. \tag{112}$$

Lemma 17 (Convergence rate of `Clapping`-**FC** with Adam optimizer)**.** *Suppose Assumption 1-5 hold. Then for Algorithm 5 there exist $\gamma,\beta_1,p>0$ such that:*

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right]\lesssim&\frac{\sigma^{\frac{4}{3}}}{T^{\frac{1}{3}}(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}}\\
&+\frac{1}{T}\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}+\frac{\omega_F^2+\omega_B}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^2}+\frac{1}{(1-\omega_F)^{2(E-2)-1}}\right).
\end{aligned} \tag{113}$$

Algorithm 6 LazySampling_LargeBatch($\mathcal{D}, t, p$) with Sample-wise rule / Batch-wise rule

if $t = 1$ **then**
  Get samples $x_1^{(1)}, \cdots, x_B^{(1)}$ / the first batch $\boldsymbol{x}^{(1)}$ from $\mathcal{D}$.
  Set $f_{\mathrm{FU},b}^{(1)} \leftarrow \texttt{True}$ for all $b = 1, \cdots, B$.
else
  for $b = 1, \cdots, B$ independently **do**
    With prob. $1-p$: $x_b^{(t)} \leftarrow x_b^{(t-1)}$, $f_{\mathrm{FU},b}^{(t)} \leftarrow \texttt{False}$.
    With prob. p: $x_b^{(t)} \sim \mathcal{D}$, $f_{\mathrm{FU},b}^{(t)} \leftarrow \texttt{True}$.
  end for
  With prob. $1-p$: $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)}$ and set $f_{\mathrm{FU},b}^{(t)} \leftarrow \texttt{False}$ for all b.
  With prob. p: $\boldsymbol{x}^{(t)} \sim \mathcal{D}$ and set $f_{\mathrm{FU},b}^{(t)} \leftarrow \texttt{True}$ for all b.
end if
Return: $\boldsymbol{x}^{(t)} = (x_1^{(t)}, \cdots, x_B^{(t)})^\top$, $f_{\mathrm{FU}}^{(t)} = \left(f_{\mathrm{FU},1}^{(t)}, \cdots, f_{\mathrm{FU},B}^{(t)}\right)$.

E `Clapping` with large batch

E.1 Algorithm development

Here we establish the convergence of `Clapping` with large-batch gradients. Before presenting the theoretical analysis, we firstly present the detailed algorithm of `Clapping` with large-batch gradients.

Notations. Suppose the batch size is $B \geq 1$, and we use bold type like $\boldsymbol{y}, \boldsymbol{v}$ to represent a high-dimensional matrix formed by variables computed by the current batch. And we use standard type with subscribe $b = 1, 2, \cdots, B$ denote the activation and gradient of activation in the corresponding batch. For example, we can denote $\boldsymbol{y}_e^{(t)} := (y_{e,1}^{(t)}, y_{e,2}^{(t)}, \cdots, y_{e,B}^{(t)})^\top$ as the activation of the e-th layer.

Lazy sampling strategies of `Clapping` with large batch. The lazy sampling strategies of `Clapping` with large batch are shown as Algorithm 6. The strategies in the large-batch are including Sample-wise rule and Batch-wise rule. The sample-wise rule means taking lazy sampling process **sample by sample**. Meanwhile, the batch-wise rule means one can keep the whole batch with probability $1-p$ and use a new batch with probability p. We will show that both strategies can achieve the convergence later in Appendix E.2.

Algorithm formulation of `Clapping` with large batch. In large batch scenario, the forward and backward function for large-bach scenario are easy to obtain as they are all sample-wise. The detail is summarized as Algorithm 7 and 8. With the lazy sampling strategy as well as the forward/backward process, the `Clapping` algorithm in with large batch can be summarized as Algorithm 9.

Practical implementation of sample-wise FU. The sample-wise rule in Algorithm 6 does not require literal per-sample branching inside a communication kernel. In practice, one maintains a boolean refresh mask $f_{\mathrm{FU},b}$ over batch slots. Before each boundary transmission, the batch is stably partitioned into two contiguous sub-batches: refreshed samples with $f_{\mathrm{FU},b} = \texttt{True}$, and reused samples with $f_{\mathrm{FU},b} = \texttt{False}$. The refreshed group takes the raw path, while the reused group takes the compressed error-feedback path. The outputs are then scattered back to the original batch order. Thus the sample-wise rule can be implemented using standard batched gather/compute/send and scatter operations. The batch-wise rule is the simplest default, while the sample-wise rule is the more flexible fine-grained version.

E.2 Convergence of `Clapping` with large batch

In this subsection, we present the convergence analysis of `Clapping` with large batch. Firstly, it is worth noting that the Descent Lemma (Lemma 2) remains applicable. Moreover, since the forward

Algorithm 7 ForwardEF_LargeBatch$_e(\tilde{y}_{e,b}^{(t-1)}, \tilde{y}_{e-1,b}^{(t)}, w_e^{(t)}, f_{\mathrm{FU},b}^{(t)})$

In worker e: $y_{e,b}^{(t)} = a_e(\tilde{y}_{e-1,b}^{(t)}, w_e^{(t)})$.
if `Clapping`-**FU and** $f_{\mathrm{FU},b}^{(t)}$ = `True` **then**
  Send $y_{e,b}^{(t)}$ from worker e to worker $e+1$.
  $\tilde{y}_{e,b}^{(t)} \leftarrow y_{e,b}^{(t)}$.
else
  Send $\mathcal{C}\Big(y_{e,b}^{(t)} - \tilde{y}_{e,b}^{(t-1)}\Big)$ from worker e to worker $e+1$.
  $\tilde{y}_{e,b}^{(t)} \leftarrow \tilde{y}_{e,b}^{(t-1)} + \mathcal{C}\Big(y_{e,b}^{(t)} - \tilde{y}_{e,b}^{(t-1)}\Big)$.
end if
Return: $\tilde{y}_{e,b}^{(t)}$.

Algorithm 8 BackwardEF_LargeBatch$_e(\tilde{v}_{e-1,b}^{(t-1)}, \tilde{v}_{e,b}^{(t)}, \tilde{y}_{e-1,b}^{(t)}, w_e^{(t)}, f_{\mathrm{FU},b}^{(t)})$

In worker e: $v_{e-1,b}^{(t)} = \nabla_1 a_e(\tilde{y}_{e-1,b}^{(t)}, w_e^{(t)})^\top \tilde{v}_{e,b}^{(t)}$.
if `Clapping`-**FU and** $f_{\mathrm{FU},b}^{(t)}$ = `True` **then**
  Send $v_{e-1,b}^{(t)}$ from worker e to worker $e-1$.
  $\tilde{v}_{e-1,b}^{(t)} \leftarrow v_{e-1,b}^{(t)}$.
else
  Send $\mathcal{C}\Big(v_{e-1,b}^{(t)} - \tilde{v}_{e-1,b}^{(t-1)}\Big)$ from worker e to worker $e-1$.
  $\tilde{v}_{e-1,b}^{(t)} \leftarrow \tilde{v}_{e-1,b}^{(t-1)} + \mathcal{C}\Big(v_{e-1,b}^{(t)} - \tilde{v}_{e-1,b}^{(t-1)}\Big)$.
end if
Return: $\tilde{v}_{e-1,b}^{(t)}$.

Algorithm 9 `Clapping` with large-batch gradients.

Require: Initialize $\{\tilde{y}_{e,b}^{(0)} = 0\}_{e=1\ b=1}^{E-1\ B}$, $\{\tilde{v}_{e,b}^{(0)} = 0\}_{e=1\ b=1}^{E-1\ B}$, and $\{\tilde{u}_e^{(0)} = 0\}_{e=1}^{E}$. Dataset $\mathcal{D}$, stepsize $\{\gamma_t\}$, compressor $\mathcal{C}$, lazy-sampling $\{p_t\}$.
for $t = 1, \cdots, T$ **do**
  $\boldsymbol{x}^{(t)}, \boldsymbol{f}_{\mathrm{FU}}^{(t)} \leftarrow$ LazySampling_LargeBatch$(\mathcal{D}, t, p_t)$.
  Set $\tilde{y}_{0,b}^{(t)} \leftarrow x_b^{(t)}$ for all $b = 1, \cdots, B$, and set $\tilde{v}_{E,b}^{(t)} \leftarrow 1$.
  for $e = 1, 2, \cdots, E-1$ **do**
    for $b = 1, 2, \cdots, B$ **do**
      $\tilde{y}_{e,b}^{(t)} \leftarrow$ ForwardEF_LargeBatch$_e(\tilde{y}_{e,b}^{(t-1)}, \tilde{y}_{e-1,b}^{(t)}, w_e^{(t)}, f_{\mathrm{FU},b}^{(t)})$.
    end for
  end for
  for $e = E, E-1, \cdots, 1$ **do**
    $g_e^{(t)} \leftarrow \frac{1}{B}\sum_{b=1}^{B} \nabla_2 a_e(\tilde{y}_{e-1,b}^{(t)}, w_e^{(t)})^\top \tilde{v}_{e,b}^{(t)}$.
    $\tilde{u}_e^{(t)} \leftarrow (1-m_t)\tilde{u}_e^{(t-1)} + m_t g_e^{(t)}$.
    $w_e^{(t+1)} \leftarrow w_e^{(t)} - \gamma_t \tilde{u}_e^{(t)}$.
    if $e \neq 1$ **then**
      for $b = 1, 2, \cdots, B$ **do**
        $\tilde{v}_{e-1,b}^{(t)} \leftarrow$ BackwardEF_LargeBatch$_e(\tilde{v}_{e-1,b}^{(t-1)}, \tilde{v}_{e,b}^{(t)}, \tilde{y}_{e-1,b}^{(t)}, w_e^{(t)}, f_{\mathrm{FU},b}^{(t)})$.
      end for
    end if
  end for
end for

and backward propagations are, in fact, sample-wise, the propagations of activations and the gradients of activations during the forward and backward passes of Algorithm 9 are identical to those of

Algorithm 4. The sole distinction between Algorithm 9 and Algorithm 4 lies in the acquisition of $\tilde{u}$. Consequently, the lemmas regarding error accumulation, specifically Lemmas 4 - 6 and Lemmas 7 - 11, can be directly employed in the analysis of Algorithm 9.

The following lemma is the large-batch version of Lemma 3:

Lemma 18 (Large-batch version of Lemma 3)**.** *Suppose Assumption 1 and 2 holds, and let* $m_1 = \cdots = m_T = m_{T+1} = m$ *as well as* $p_3 = \cdots = p_T = p_{T+1} = p$*. Moreover, we set* $p_2 = 1$*. Then, for all* $t = 2, \cdots, T+1$ *we have:*

$$
\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\leq & 32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))} + \frac{1}{m^2}\right)\sum_{e=1}^{E}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\
&+ \frac{8(L_{\nabla a}^{\circ})^2}{B}\sum_{e=1}^{E-1}\sum_{b=1}^{B}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_{e,b}^{(t)} - \hat{v}_{e,b}^{(t)}\right\|^2\right] + \frac{8(L'_{\nabla a})^2}{B}\sum_{e=1}^{E-1}\sum_{b=1}^{B}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_{e,b}^{(t)} - \hat{y}_{e,b}^{(t)}\right\|^2\right] \\
&+ 4T\frac{\sigma^2}{B}\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2} + \frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)} - \nabla_e \ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}
\tag{114}
$$

Proof. For $t = 2, 3, \cdots, T$, we denote $\psi(t)$ as the last moment in which the sample is randomly obtained with $\mathcal{D}$ as of the t-th iteration. Specially,

$$
\psi(t) \coloneqq \max_{\tau \in \mathbb{S}_t} \tau, \quad \text{where } \mathbb{S}_t \coloneqq \{\tau = 2, 3, \cdots, t | \text{sampling randomly at iteration } \tau\}.
$$

Then, with the fact that the $p_2 = 1$, it holds for $\tau = 2, \cdots, t$ that $\Pr(\psi(t) = \tau) = \begin{cases} (1-p)^{t-2}, & \text{if } \tau = 2 \\ p(1-p)^{t-\tau}, & \text{else.} \end{cases}$

For $e = 1, 2, \cdots, E-1$ and $t = 2, 3, \cdots, T+1$, the error between the evaluated gradient and the true gradient satisfies:

$$
\begin{aligned}
&\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)}) \\
=& \underbrace{\sum_{\tau=\psi(t)}^{t}\frac{1}{B}\sum_{b=1}^{B} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\tilde{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\tilde{v}_{e,b}^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_{e,b}^{(\tau)}\right)}_{\coloneqq \Xi_{e,1}} \\
&+ \sum_{\tau=\psi(t)}^{t}\frac{1}{B}\sum_{b=1}^{B} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^{\mathsf{T}}\hat{v}_{e,b}^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right) \\
&+ \underbrace{\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)}_{\coloneqq \Xi_{e,2}} + \underbrace{(1-m)^{t+1-\psi(t)}\left(\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)}_{\coloneqq \Xi_{e,3}} \\
&+ (1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right),
\end{aligned}
\tag{115}
$$

where the first equation is from the momentum update rule. Moreover, we use $\Xi_{e,1}, \Xi_{e,2}, \Xi_{e,3}$ to denote some complex terms, which have been shown in Eq. (115).

Taking the ℓ_2-norm and conditional expectation with respect to $\mathcal{F}^{(\psi(t))}$ on both sides of Eq. (115), we can obtain:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
=&\mathbb{E}\left[\left\|\sum_{\tau=\psi(t)}^{t} \frac{1}{B}\sum_{b=1}^{B} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_{e,b}^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\mathbb{E}\Bigg[\Bigg\langle \sum_{\tau=\psi(t)}^{t} \frac{1}{B}\sum_{b=1}^{B} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_{e,b}^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right), \\
&\qquad (1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\Bigg\rangle \Bigg| \mathcal{F}^{(\psi(t))}\Bigg].
\end{aligned} \tag{116}$$

Thus, we can get:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq&\frac{2}{B^2}\sum_{b=1}^{B}\mathbb{E}\left[\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_{e,b}^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|\Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ \mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right],
\end{aligned} \tag{117}$$

where the inequality is due to Cauchy-Schwarz inequality, Assumption 2 and the fact that samples in the batch are obtained independently.

For the second term of the right-hand-side of Eq. (117), it holds that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|\Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq&\frac{2}{B}\sum_{\tau=\psi(t)}^{t}\sum_{b=1}^{B} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \tilde{v}_{e,b}^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_{e,b}^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ (1-m)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right],
\end{aligned} \tag{118}$$

where the inequality holds is due to the convexity of the ℓ_2-norm.

Moreover, for the last term, it also holds that:

$$\begin{aligned}
&\mathbb{E}\left[\left\|(1-m)^{t+1-\psi(t)}\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \Xi_{e,1} + \Xi_{e,2} + \Xi_{e,3}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
\leq&\frac{2}{B}\sum_{\tau=\psi(t)}^{t}\sum_{b=1}^{B} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_2 a_e(\tilde{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \tilde{v}_{e,b}^{(\tau)} - \nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^\top \hat{v}_{e,b}^{(\tau)}\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ 2\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\mathbb{E}\left[\left\|\nabla_e \ell(\boldsymbol{w}^{(\tau)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right] \\
&+ (1-m)^{t+1-\psi(t)}\mathbb{E}\left[\left\|\left(\tilde{u}_e^{(\psi(t)-1)} - \nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)})\right) + \left(\nabla_e \ell(\boldsymbol{w}^{(\psi(t)-1)}) - \nabla_e \ell(\boldsymbol{w}^{(t)})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right].
\end{aligned} \tag{119}$$

With Assumption 2, we can get:

$$\frac{1}{B^2}\mathbb{E}\left[\sum_{b=1}^{B}\sum_{e=1}^{E}\left\|\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\left(\nabla_2 a_e(\hat{y}_{e-1,b}^{(\tau)}, w_e^{(\tau)})^{\intercal}\hat{v}_{e,b}^{(\tau)} - \nabla_e \ell(\boldsymbol{w}^{\tau})\right)\right\|^2 \middle| \mathcal{F}^{(\psi(t))}\right]$$
$$\leq \left(\sum_{\tau=\psi(t)}^{t} m(1-m)^{t-\tau}\right)^2 \frac{\sigma^2}{B}.$$

Finally, with the same process of Lemma 3, we can present the result that:

$$\begin{aligned}
&\sum_{e=1}^{E}\sum_{t=1}^{T+1}\mathbb{E}\left[\left\|\tilde{u}_e^{(t)} - \nabla_e \ell(\boldsymbol{w}^{(t)})\right\|^2\right] \\
\leq& 32L_{\nabla\ell}^2\left(\frac{p+m}{m^2(1-(1-p)(1-\frac{m}{2}))} + \frac{1}{m^2}\right)\sum_{e=1}^{E}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_e^{(t+1)} - w_e^{(t)}\right\|^2\right] \\
&+ \frac{8(L_{\nabla a}^{\circ})^2}{B}\sum_{e=1}^{E-1}\sum_{b=1}^{B}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{v}_{e,b}^{(t)} - \hat{v}_{e,b}^{(t)}\right\|^2\right] + \frac{8(L'_{\nabla a})^2}{B}\sum_{e=1}^{E-1}\sum_{b=1}^{B}\sum_{t=2}^{T+1}\mathbb{E}\left[\left\|\tilde{y}_{e,b}^{(t)} - \hat{y}_{e,b}^{(t)}\right\|^2\right] \\
&+ 4T\frac{\sigma^2}{B}\frac{(2-p)m-(1-p)m^2}{1-(1-p)(1-m)^2} + \frac{3}{m}\sum_{e=1}^{E}\mathbb{E}\left[\left\|\tilde{u}_e^{(1)} - \nabla_e \ell(\boldsymbol{w}^{(1)})\right\|^2\right].
\end{aligned}$$

□

Compared with Eq. (14), the most difference of Eq. (114) is that the noise term σ^2 has been replaced to $\frac{\sigma^2}{B}$. And thus we can simply obtain the convergence of `Clapping` in the large batch scenario.

Lemma 19 (Convergence rate of `Clapping`-**FU** with large batch)**.** *Suppose Assumption 1, 2, and 3 hold. Then for Algorithm 4 there exist $\gamma, m, p > 0$ such that:*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim \frac{\sigma}{\sqrt{BT}} + \frac{1}{T(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}}. \tag{120}$$

Lemma 20 (Convergence rate of `Clapping`-**FC** with large batch)**.** *Suppose Assumption 1, 2, and 3 hold. Then for Algorithm 4 there exist $\gamma, m > 0$ such that:*

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\ell(\boldsymbol{w}^{(t)})\right\|^2\right] \lesssim& \frac{\sigma^{\frac{4}{3}}}{(BT)^{\frac{1}{3}}(1-\omega_B)^{\frac{4(E-1)}{3}}(1-\omega_F)^{\frac{4(E-1)}{3}}} \\
&+ \frac{1}{T}\left(\frac{1}{(1-\omega_B)^{E-1}(1-\omega_F)^{E-1}} + \frac{\omega_F^2+\omega_B}{(1-\omega_B)^{2(E-1)}(1-\omega_F)^2} + \frac{1}{(1-\omega_F)^{2(E-2)-1}}\right).
\end{aligned} \tag{121}$$

F Additional discussions

In this section, we present a series of additional discussions to the proposed algorithm, theoretical analysis and emprical efficiency.

F.1 When should communication compression be enabled?

`Clapping` is designed for *communication-bound* pipeline parallelism, where inter-stage transfers of activations/adjoints dominate iteration time under limited bandwidth. In high-bandwidth intra-node deployments (e.g., NVLink/NVSwitch) where communication is not the bottleneck, compression may offer little benefit and can introduce overhead; in such settings it is reasonable to disable compression. Our focus is therefore on regimes with constrained cross-node/over-the-internet links, where compression is often necessary for efficiency and `Clapping` provides an optimization-stability mechanism under aggressive inter-stage compression (see Appendix H.4.5 for a bandwidth-sensitivity study).

F.2 Additional details on multi-worker scenarios

In Section 2.2, we present the forward and backward process of AQ-SGD [67]. However, it is noteworthy that the convergence analysis of [67, Appendix A.1.] assume that the machine e computes the gradient of activation by

$$v_{e-1}^{(t)} = \nabla_1 a_e(\tilde{y}_{x,e-1}^{(t)}, w_e^{(t)})^\mathsf{T} \mathcal{C}\left[\frac{\partial a_{e+1} \circ \cdots \circ a_E}{\partial \tilde{y}_{x,e}^{(t)}}(\tilde{y}_{x,e}^{(t)}, w_{e+1}^{(t)}, \cdots, w_E^{(t)})\right], \tag{122}$$

where $a_{e+1} \circ \cdots \circ a_E$ denotes the composition of $a_{e+1}, \cdots, a_E$. However, the parameters $w_{e+1}^{(t)}, \cdots, w_E^{(t)}$ are held in different machines. Consequently, the computation of the partial gradient in the last term of (122) cannot avoids the communication between workers. Thus, it cannot be obtained losslessly through the communication compression during the propagation process. Thus, there causes a mismatch between the analysis in [67] and the reality. In fact, the gradient of activation $v_{e-1}^{(t)}$ can be obtained by the back-propagation as:

$$v_{e-1}^{(t)} = \nabla_1 a_e(\tilde{y}_{x,e-1}^{(t)}, w_e^{(t)})^\mathsf{T} \mathcal{C}\left[\nabla_1 a_{e+1}(\tilde{y}_{x,e}^{(t)}, w_{e+1}^{(t)})^\mathsf{T} \mathcal{C}\left[\cdots \mathcal{C}\left(\nabla_1 a_E(\tilde{y}_{x,E-1}^{(t)}, w_E^{(t)})\right)^\mathsf{T} \cdots\right]\right], \tag{123}$$

which calls for the analysis of error accumulation because of the multiple compression.

F.3 On the relation to vanilla SGD and the $m = 0$ case

The convergence bounds of `Clapping`-**FU** in Lemma 12 and Lemma 3 are derived under the standing assumption that the lazy–sampling probability is a fixed positive constant, i.e., $p_t \equiv p \in (0, 1]$, and that the stepsize γ and the momentum parameter m are chosen accordingly. In this regime we obtain the nonconvex rate

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\|\nabla \ell(w^{(t)})\right\|^2 \lesssim \frac{\sigma}{\sqrt{T}} + \frac{1}{T(1-\omega_F)(1-\omega_B)}$$

up to logarithmic factors, where the variance term σ^2 comes from the stochasticity of the activation gradients and the compression errors.

(i) Behaviour as $p \to 0$. The proofs of Lemma 12 and Lemma 3 crucially require that p is bounded away from zero. Indeed, the coefficient in front of the variance term,

$$\sigma^2 \cdot \frac{(2-p)m - (1-p)m^2}{1-(1-p)(1-m)^2},$$

remains $\Theta(\sigma^2)$ as $p \to 0$. In this limiting regime the algorithm keeps reusing the same sample over a long horizon, and the bound only guarantees convergence to a variance–dominated neighbourhood, rather than $\|\nabla \ell(w^{(t)})\| \to 0$. This behaviour is consistent with classical SGD intuition when the data is not refreshed: the iterates converge to a noise–dominated region determined by the fixed sample, instead of an exact stationary point. A similar phenomenon appears in the large–batch analysis (see Eq. (114)), where setting $p = 0$ reduces the relevant ratio to 1 and leaves a residual term of order $4\sigma^2/B$ that does not vanish with T. Therefore our rate statements should be interpreted for fixed $p > 0$ and are not intended to cover the degenerate case $p = 0$.

(ii) Why $m = 0$ does not reduce to vanilla SGD. Even when $m = 0$, Algorithm 4 still performs forward and backward passes with *compressed* activations and adjoints, together with *error feedback* across pipeline stages. The update direction used at stage e is

$$\tilde{u}_e^{(t)} = \nabla_2 a_e\big(\tilde{y}_{e-1}^{(t)}, w_e^{(t)}\big)^\mathsf{T} \tilde{v}_e^{(t)},$$

where $(\tilde{y}, \tilde{v})$ are generated by the compression and error–feedback recursions. Hence, the case $m = 0$ in our framework corresponds to SGD with a *biased, compression–affected* gradient estimator, rather than standard vanilla SGD on fresh, uncompressed samples.

The residual σ^2 term in Lemma 12 and Lemma 3 exactly captures the fact that, in this setting, compression induces a persistent gradient bias that is not completely averaged out when $m = 0$.

This residual bias arises from the pipeline compression structure and is present even without lazy sampling; it is not a contradiction with the classical convergence guarantees for vanilla SGD.

(iii) Recovery of the standard nonconvex SGD rate. If we additionally disable compression ($\omega_F = \omega_B = 0$) and turn off lazy sampling ($p_t \equiv 1$), Algorithm 4 reduces to pipeline–parallel vanilla SGD. In this special case the error–feedback recursions collapse and our analysis recovers the standard nonconvex rate

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\nabla \ell(w^{(t)})\right\|^2 = \mathcal{O}\left(\frac{\sigma}{\sqrt{T}}\right)$$

without additive residual terms. Thus there is no inconsistency between our bounds and the known behaviour of vanilla SGD; the additive σ^2 term only appears in the genuinely compressed-and-lazy–sampled regime we target.

F.4 Lazy sampling, convergence and generalization

Lazy sampling keeps the marginal data distribution unchanged: at each iteration we either reuse the previous sample $x^{(t-1)}$ with probability $1 - p_t$ or draw a fresh sample $x^{(t)} \sim \mathcal{D}$ with probability p_t. Hence the sequence $\{x^{(t)}\}_{t\geq 1}$ is no longer i.i.d., but every marginal law is still $\mathcal{D}$. Lazy sampling therefore introduces temporal correlations without systematically excluding any part of the data distribution.

The convergence bounds of `Clapping` make the role of $\{p_t\}$ explicit. For the FC variant, the variance term

$$\sigma^2 \cdot \frac{(2-p)m - (1-p)m^2}{1-(1-p)(1-m)^2}$$

remains of order $\Theta(\sigma^2)$ when p_t is chosen too small, which prevents the averaged gradient norm from vanishing. Intuitively, as $p_t \to 0$ the algorithm repeatedly optimizes on almost the same sample, so the optimization dynamics are dominated by the noise induced by that sample and the method can only converge to a noise-dominated neighborhood of a stationary point.

On the other hand, if p_t is taken very close to 1, a new sample is seen at almost every step and the temporal correlation disappears, but the error–feedback iterates have fewer opportunities to cancel the compression bias. In this regime `Clapping`-**FC** may converge more slowly, whereas `Clapping`-**FU** effectively approaches standard (uncompressed) SGD and thus loses its communication advantages.

These observations suggest that the lazy–sampling schedule $\{p_t\}$ should be chosen in an *intermediate* regime. It must be bounded away from zero to ensure that the effective number of distinct samples grows linearly with T, while it need not (and in general should not) be exactly one. In our analysis p_t thus plays the role of a bias–variance trade-off parameter: moderate values of p_t guarantee both sufficient exploration of the data distribution and enough sample reuse for error–feedback to attenuate compression bias.

F.4.1 Adaptive refresh rates and scope of interpretation

Only fixed refresh probabilities p were evaluated in the additional experiments. Although a time-varying or adaptive p is compatible with the algorithmic update, neither an adaptive-policy experiment nor an adaptive-schedule convergence guarantee is available. Compression-induced implicit regularization remains a possible explanation for occasional differences in held-out scores, rather than a causally established mechanism.

G Memory overhead analysis for LLMs pre-training with `Clapping`

In this section, we present the analysis for the memory overhead introduced by `Clapping` in the pre-training tasks for LLMs.

Basic setup. We consider a pre-training task on LLaMA-2-based models with SwiGLU activation [61] running on Nvidia A100 80G GPUs. The dataset is C4-en [48], which is primarily intended for pre-training language models and word representations on a large scale. We use the T5-base tokenizer with a sequence length of $s = 4096$, resulting in a total of approximately $|\mathcal{D}| \approx 45.6M$ training

Table 4: Hyperparameter configurations for LLaMA-2 models of different scales. 'Num_workers' denotes the total workers for a training pipeline.

Parameters	Hidden size (h)	Heads (a)	Layers (L)	Vocabulary size (V)	Num_workers
7B	4096	32	32	32000	2
13B	5120	40	40	32000	4
70B	8192	64	80	32000	16

Table 5: The memory overhead for pre-training LLaMA-2 models with different model size. 'Basic M.' means the memory overhead for parameters, gradients, optimizer states, and activation memorys. '`Clapping` M.' means the additional memory introduced by `Clapping` . 'AQ-SGD M.' means the additional memory introduced by AQ-SGD. 'Ratio' denotes the ration between the additional memory of `Clapping` and the basic memory.

Parameters	Basic M. (GB)	`Clapping` M. (GB)	Ratio	AQ-SGD M. (GB)
7B	65.0	2.0	3.07%	2850.0
13B	118.1	7.5	6.35%	10687.5
70B	562.0	60.0	10.68%	85500.0

samples. The micro-batch size is $B = 16$, and the total batch size is set to 256. The optimizer used is Adam [25], and we employ BF16 precision, where each parameter requires 2 bytes for storage.

If we assume the intermediate size $h_{\text{ff}} = \frac{8h}{3}$, we use vanilla gradient checkpointing (GCP) [6] to save activations and do not use GQA for 70B models. The basic memory overhead for parameters, gradients, optimizer states, and activations is:

$$\underbrace{4Vh + 48Lh^2}_{\text{parameters, gradients, optimizer states}} + \underbrace{LBsh}_{\text{Activation memory with GCP}}$$

For `Clapping` , it should cache the term $\tilde{y}_e$ in workers e and $e+1$, as well as the term $\tilde{v}_{e-1}$ in workers $e-1$ and e. Since we often split the model at the end of some transformer block, both $\tilde{y}_e$ and $\tilde{v}_{e-1}$ have a size of Bsh. Thus, the memory overhead of `Clapping` is:

$$4 \times (\text{num_workers} - 1)Bsh.$$

Similarly, the memory overhead of AQ-SGD can be expressed as:

$$2 \times (\text{num_workers} - 1)|\mathcal{D}|sh.$$

We present the memory overhead analysis for pre-training LLaMA-2 models with different communication compression algorithms under pipeline parallelism in Table 5. It can be observed that the memory overhead introduced by `Clapping` is less than 11% of the basic memory even when the model size scales to 70B, which is acceptable for practical pre-training tasks. Meanwhile, AQ-SGD requires thousands of GBs to store the sample-wise cache, making it unsuitable for pre-training tasks.

H Experimental details

In this section, we present the details of our numerical experiments, which were discussed in Section 5. Additionally, we provide additional experimental results that were not included in the main text due to space limitations.

H.1 Synthetic Logistic Regression

Herein, we consider the following logistic regression task with the objective function given by:

$$f(w,b) = \mathbb{E}_{(\xi,\zeta)\sim\mathcal{D}}\left[\ln(1+\exp(-\zeta\cdot(\xi^\top w+b))) + C_r\|w\|^2 + C_r b^2\right], \tag{124}$$

where w and b denote the regression parameters, and C_r is a regularization parameter. $\mathcal{D}$ represents the finite sample set. Here, we the sample size $|\mathcal{D}|$. For each stochastic sample (ξ,ζ) in $\mathcal{D}$, ξ is a 200-dimensional vector generated as the standard $\xi^* + \varepsilon$, where each element of ξ^* is independently

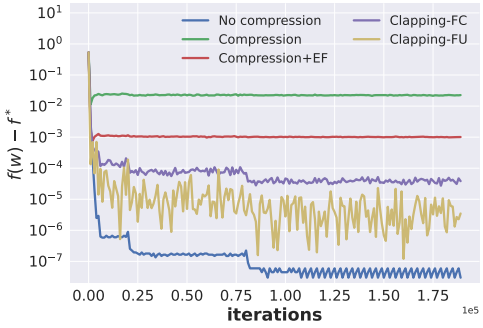

Figure 4: The gap between the current loss and the optimal loss for different algorithms for the logistic regression problem.

normal distribution drawn from n $\mathcal{N}(0, 0.5)$ and each element of ε is independently drawn from $\mathcal{N}(0, 0.3)$. Let $C_r = 0.005$. Then, Equation (124) is strongly convex. We employ gradient descent to find the minimum of Equation (124), denote as f^*. Subsequently, we split the model into two parts according to the formulation in (1), where

$$y_1 = -\zeta \cdot (\xi^\top w + b), \quad y_2 = \ln(1 + \exp(y_1)) + C_r\|w\|^2 + C_r b^2.$$

After obtaining y_1 during the forward-propagation process, we introduce an error to y_1 as δ_1 to simulate activation compression and use $\tilde{y}_1 = y_1 + \delta_1$ to compute y_2, where δ_1 follows the uniform distribution $U(-0.2, 0.2)$. We set the batch size to 128. Then, we run the moving-average Stochastic Gradient Descent (SGD) algorithm for 200,000 iterations with different error compensation strategies, including direct compression without error feedback **(Compression)**, compression with error feedback but without lazy sampling **(Compression + EF)**, `Clapping`-**FC**, and `Clapping`-**FU**. We compare these strategies with the case of no compression **(No Compression)**. The moving-average term is set to 0.9, and the step-size is initialized to 0.1. When the iteration number , the step-size is multiplied by 0.5, and the historical gradient is cleared. Figure 4 depicts the gap between the current loss and the optimal loss for different algorithms. Both direct compression and compression with error feedback but without lazy sampling fail to converge as the step-size decreases. However, both `Clapping`-**FC** and `Clapping`-**FU** can converge as the step-size decreases, and `Clapping`-**FU** convergence faster than `Clapping`-**FC** to a smaller minimum. The error introduced by activation compression only slows down the convergence process, which validates our theoretical findings in Section 4.2.

H.2 Training ResNet-18 on CIFAR-10

We trained a ResNet-18 [17] model on CIFAR-10 [31] dataset on a NVIDIA A100 80G GPU with communication compression algorithms for pipeline-parallel learning including directly compression and `Clapping`-**FC**. The basic setting is similar to that of [51]. We split the model into 4 parts and used 3 direct quantization [1] with different bits to simulate the communication compression. We set the batch size to 128. We trained 100 epochs with directly compression and trained with `Clapping`-**FC** for the same number of iterations. We used the SGD optimizer with momentum 0.9 and weight decay 5e-4. The learning rate is initialized to 0.01 and scheduled by a cosine annealing scheduler with $T_{\max} = 200$. During inference, we obtain the test accuracy with the sample compression as training and without any compression, respectively.

Table 6 shows the average best test accuracy of directly compression and `Clapping` with different lazy sampling coefficient p_t with different compression strategies over 3 independent runs. We can observe that the error feedback technique and lazy sampling strategy can improve the prediction accuracy in the communication compression of pipeline parallelism.

H.3 Fine-tuning LLMs

H.3.1 Fine-tuning on GLUE benchmark

We fine-tune pre-trained RoBERTa-large [35] dataset on GLUE benchmark [64] with communication compression algorithms including `Clapping` and direct compression with EF21 [50] and compare then

Table 6: The best test accuracy in training ResNet-18 on CIFAR-10.

	Strategy	Direct comp.	Clapping with p_t				
			0.2	0.4	0.6	0.8	1.0
C. on	f5-b5	0.8604 (0.00170)	0.8719 (0.01041)	0.8889 (0.00257)	**0.9034 (0.00117)**	0.9016 (0.00066)	0.8963 (0.00209)
	f6-b6	0.9024 (0.00709)	0.8985 (0.00997)	0.9119 (0.00271)	**0.9185 (0.00466)**	0.9150 (0.00373)	0.9133 (0.00397)
	f6-b8	0.9056 (0.00558)	0.9014 (0.00199)	0.9109 (0.00496)	0.9140 (0.00805)	**0.9184 (0.00129)**	0.9160 (0.00535)
C. off	f5-b5	0.8623 (0.00271)	0.8733 (0.01010)	0.8907 (0.00187)	**0.9045 (0.00104)**	0.9022 (0.00039)	0.8982 (0.00197)
	f6-b6	0.9025 (0.00643)	0.8986 (0.00978)	0.9123 (0.00282)	**0.9186 (0.00438)**	0.9150 (0.00412)	0.9137 (0.00373)
	f6-b8	0.9058 (0.00551)	0.9015 (0.00179)	0.9109 (0.00491)	0.9139 (0.00798)	**0.9184 (0.00121)**	0.9161 (0.00533)

'f[A]-b[B]' means compressing activation to A bits in forward propagation and compressing gradient to B bits in backward propagation. *'C. on'* means taking the same compression during inference and *'C. off'* means taking no compression.

Table 7: RoBERTa-large GLUE task scores for Clapping-FC and Clapping-FU (higher is better).

Method	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
Clapping-FC	89.94	96.06	90.50	63.98	94.04	91.89	84.19	91.71	87.79
Clapping-FU	89.92	96.18	90.25	64.20	94.17	91.97	83.82	92.27	87.85

with fine-tuning without compression on two Nvidia A800 GPUs. We use the TopK compression with 30% elements at the middle of the networks. The batch size are all set 64 and the learning rate is 1e-5. For each dataset, we fine-tune the model for 10 epochs using both fine-tuning without compression and the EF21 algorithm. We also tune fine-tune the model for the same number of iterations as in 10 epochs by `Clapping` with lazy sampling coefficient $p = 0.5$. As Table 2 illustrates, `Clapping` outperforms EF21 in majority of tasks and even outperforms fine-tuning without communication compression in tasks including MRPC and RTE. And `Clapping` achieves the highest average score among all the algorithms.

H.3.2 Additional Clapping-FU results on GLUE

The RoBERTa-large GLUE results in the main text report FC (Table 2). We additionally record the corresponding FU scores for all eight tasks in Table 7. The FC column is repeated to facilitate direct comparison. Scores and their average follow the reporting scale used in the original GLUE experiment.

H.3.3 Fine-tuning LLaMA models with TopK compressor

We fine-tuned a pretrained LLaMA-2 7B [61] model and a LLaMA-3 8B [14] model on Wikitext dataset (wikitext-2-raw-v1 version) [38] on two NVIDIA A100 80G GPUs with communication compression algorithms for pipeline-parallel learning including directly compression, AQ-SGD, EF21 and `Clapping`-**FC**. The block size was set to 1024. We used batch size 8, and we fine-tuned the model for 4 epochs for the competitive algorithms and we use the same iterations for `Clapping` , respectively. We use the SGD optimizer and FP16 for fine-tuning. The learning rate is initialized from 2×10^{-5} and scheduled by a linear scheduler. We compare different compression algorithms with the compressor of Top-5% [69]. For each model, we independently repeat `Clapping` with different lazy sampling coefficient p_t including $\{0.3, 0.4, 0.5\}$.

Table 3 has present the evaluation accuracy of different approaches under Top-5% compressor. It can be observed that `Clapping` outperforms other algorithms, including direct compression, EF21, and

Table 8: WikiText-2 validation accuracy over three seeds (higher is better). FC and FU use $p = 0.5$ and Top-5% compression.

Model	No comp.	Direct	EF21	AQ-SGD	Clapping-FC	Clapping-FU
LLaMA-2 7B	0.5954 (0.0015)	0.5526 (0.0055)	0.5567 (0.0102)	0.5684 (0.0026)	0.5855 (0.0038)	0.5960 (0.0015)
LLaMA-3 8B	0.5953 (0.0034)	0.5648 (0.0025)	0.5649 (0.0033)	0.5703 (0.0028)	0.5893 (0.0045)	0.5919 (0.0008)

Table 9: Paired two-sided t-test p-values for WikiText-2 accuracy compared with direct compression; matched seeds 42, 123, and 555.

Model	FC vs. Direct	FU vs. Direct
LLaMA-2 7B	0.0242	0.0050
LLaMA-3 8B	0.0230	0.0037

Table 10: WikiText-2 validation accuracy with Top-5% compression for different refresh probabilities (higher is better).

Model	Clapping-FC			Clapping-FU		
	$p = 0.3$	$p = 0.4$	$p = 0.5$	$p = 0.3$	$p = 0.4$	$p = 0.5$
LLaMA-2 7B	0.5960	0.5920	0.5877	0.5923	0.5944	0.5947
LLaMA-3 8B	0.5865	0.5969	0.5887	0.5862	0.5910	0.5912

AQ-SGD. By tuning the lazy-sampling coefficient p, **`Clapping-FC` achieves 95% communication saving with less than 0.5% error in practical fine-tuning tasks**.

H.3.4 Independent runs and WikiText evaluation

For the LLaMA-2 7B and LLaMA-3 8B WikiText-2 fine-tuning experiments, the original run used seed 555. Two independent repetitions with seeds 42 and 123 were added. The model, data, training budget, Top-5% compressor, and, for FC and FU, the refresh probability $p = 0.5$ were held fixed across these runs. Table 8 reports the mean and standard deviation of validation accuracy over the three seeds.

The results were compared with direct compression using paired, two-sided t-tests on the three matched seeds. Table 9 reports the stated unadjusted p-values. All four comparisons pass a Holm correction at family-wise significance level 0.05; no significance claim against NoComp is made.

H.3.5 FC/FU comparison across refresh rates and compressor strengths

Table 10 gives the single-seed (seed 555) WikiText-2 validation accuracies when p varies over $\{0.3, 0.4, 0.5\}$, using the Top-5% compressor. The FC values also appear in the main experimental table (Table 3); the FU values are additional results. These single-seed observations are distinct from the three-seed summary in Table 8.

A separate, stronger-compression setting retains only the largest 1% of elements, with $p = 0.4$ for FC and FU. The corresponding WikiText-2 validation accuracies are given in Table 11.

Table 11: WikiText-2 validation accuracy with Top-1% compression ($p = 0.4$ for FC/FU; higher is better).

Model	Direct	Clapping-FC	Clapping-FU
LLaMA-2 7B	0.5339	0.5705	0.5741
LLaMA-3 8B	0.4627	0.5745	0.5761

Table 12: The evaluation perplexity (↓) and accuracy (↑) for the fine-tuning task for LLaMA-2 13B under Wikitext with `Clapping`-**FC** with different number of workers (E).

Number of workers (E)	Validation loss ↓	evaluation perplexity	Validation accuracy
1	1.813602	6.1325	0.5823
2	1.915392	6.7896	0.5654
4	2.077440	7.9840	0.5422
9	2.603682	13.5134	0.4606
16	2.842107	17.1519	—
32	3.227581	25.2186	—

Table 13: LLaMA-2 13B WikiText sequential non-IID stress test.

Method	Evaluation loss ↓	Accuracy ↑
NoComp	1.76885	0.59511
Lazy-only	1.76169	0.59598
Clapping-FU	1.76027	0.59614

FC performs compressed error-feedback communication at every step, including refresh steps. FU uses uncompressed boundary transfers on refresh steps and compressed error-feedback transfers on reuse steps. For a constant refresh probability p, FC has $1/(1-p)$ times as many compressed-step opportunities as FU in expectation; this factor equals approximately 1.67 at $p = 0.4$. This counts compressed steps rather than specifying an end-to-end byte ratio, which additionally depends on message size and the compressor.

H.3.6 Pipeline-depth sensitivity on LLaMA-2 13B

Ablations on the impact of the number of workers on performance. In Section 4.2, we showed that communication compression errors can accumulate exponentially with the number of workers E. To empirically examine this effect, we fine-tuned a pre-trained LLaMA-2 13B model [61] on WikiText for 4 epochs using `Clapping`-**FC** with a Top-20% compressor and a fixed refresh probability of $p = 0.5$. We varied the number of workers over $E \in \{1, 2, 4, 9, 16, 32\}$ while keeping the other configurations unchanged, except for increasing the initial learning rate to 5×10^{-5} to amplify the effect of error accumulation. Table 12 reports validation loss and perplexity across all six configurations, together with validation accuracy for $E \leq 9$. The results show that validation loss and perplexity increase as the number of workers grows, consistent with the accumulation of compression errors across pipeline stages.

H.3.7 Sequential non-IID data-order stress test

We consider a single sequential-order WikiText experiment on LLaMA-2 13B. Examples are traversed in their stored order with random shuffling disabled. The sequence length is 1,024, micro-batch size is one per GPU, global batch size is eight, and training lasts four epochs in FP16 using SGD with learning rate 5×10^{-5}. Lazy-only and Clapping-FU use $p = 0.5$; the recorded methods are NoComp, Lazy-only (sample reuse without compression), and Clapping-FU (sample reuse with compression). All other settings are held fixed. Table 13 reports the held-out results.

This experiment tests one sequential data stream and one random seed; it does not establish generalization guarantees for arbitrary non-IID orderings.

H.3.8 Generated-output quality: instruction tuning and LLM-as-a-Judge

For an output-quality stress test, LLaMA-2 7B is instruction-tuned for 500 steps on UltraChat 200K [8] using NoComp, Direct, FC, and FU. The methods share the initialization, random seed, token budget, batch size, and decoding configuration. The learning rate is fixed to the intentionally aggressive value 5×10^{-5} for all four methods; this is a matched stress test rather than a separate learning-rate search for each method.

Table 14: Position-balanced pairwise LLM-as-a-Judge judgments on 200 UltraChat prompts after 500-step LLaMA-2 7B instruction tuning at learning rate 5×10^{-5}. Each method pair is judged in both orderings.

Comparison (X vs. Y)	X wins	Tie	Y wins	Abstain
FC vs. Direct	51.0%	7.0%	36.5%	5.5%
FU vs. Direct	59.8%	5.0%	29.0%	6.3%
Direct vs. NoComp	22.5%	5.0%	65.0%	7.5%
FC vs. NoComp	25.0%	3.3%	63.5%	8.3%
FU vs. NoComp	31.5%	2.8%	54.5%	11.3%
FU vs. FC	56.3%	5.3%	31.5%	7.0%

The evaluation uses 200 shared held-out prompts and a separately fine-tuned LLaMA-2 7B judge. Each response pair is assessed in both A/B orientations. In Table 14, "X wins" means that the judge favors the first method named in the comparison; "Y wins" favors the second method; "Tie" denotes equal preference; "Abstain" means no unambiguous judgment was extracted. Values are reported to one decimal place, with minor deviations from 100% possible due to rounding.

The results demonstrate that Clapping consistently outperforms direct compression in generated-output quality, with Clapping-FU achieving the strongest performance among the communication-compressed methods. In particular, Clapping-FU achieves a win rate of 59.8% against direct compression, compared with a loss rate of 29.0%, highlighting its effectiveness in preserving model quality under aggressive communication compression.

H.3.9 Fine-tuning with multiple compression.

Here we present the experimental result in fine-tuning tasks with multiple compression. Different from the fine-tuning tasks introduced in Appendix H.3.1 and H.3.3, the experimental setup we take here is more **STRICT** than the practical fine-tuning tasks to present a comparsion for different algorithms.

Fine-tuning GPT-2 on Wikitext. We fine-tuned a pretrained GPT-2 [46] model on Wikitext dataset (wikitext-2-raw-v1 version) [38] on a NVIDIA A100 80G GPU with communication compression algorithms for pipeline-parallel learning including directly compression, AQ-SGD, `Clapping` without lazy sampling and `Clapping`-**FC**. As the experiment is taken in a single GPU, we simulate the communication compression by adding the corresponding error to the activation and gradient. The block size was set to 1024. We used batch size 8, and we fine-tuned the model for 8 epochs for the competitive algorithms and we use the same iterations for `Clapping` , respectively. We use the AdamW optimizer [36] and FP16 for fine-tuning. The learning rate is initialized from 2×10^{-5} and scheduled by a linear scheduler. We compare different compression algorithms under three compression strategies. These strategies integrated different compressor including TopK [69], direct quantization [1], and Natural compression [19] on GPT-2 with different model size, which can be summarized as follow.

Strategy S1 Basic model: GPT-2 small;
Compression position: At the end of layer 2, 5, 8;
Forward compressor: Top 40%;
Backward compressor: Top 40%.

Strategy S2 Basic model: GPT-2 medium;
Compression position: At the end of layer 4, 10, 16;
Forward compressor: Direct taking Natural Compression;
Backward compressor: Quantizing to 8 bits, then taking Natural Compression.

Strategy S3 Basic model: GPT-2 medium;
Compression position: At the end of layer 4, 10, 16;
Compressor in layer 4: Direct taking Natural Compression;
Compressor in layer 10 and 16: Quantizing to 8 bits, then taking Natural Compression.

Table 15: Evaluation perplexity of `Clapping` in GPT-2 fine-tuning with different lazy coefficient p_t.

Strategy	p_t						
	0.1	0.2	0.3	0.4	0.6	0.8	1.0
S1	16.118	15.597	15.523	**15.482**	15.675	15.999	16.582
	(0.095)	(0.070)	(0.013)	**(0.014)**	(0.015)	(0.014)	(0.417)
S2	13.432	12.992	12.713	**12.558**	12.630	12.645	12.756
	(0.118)	(0.239)	(0.343)	**(0.352)**	(0.306)	(0.250)	(0.182)
S3	15.337	14.330	**12.815**	14.106	14.261	14.466	14.712
	(0.187)	(0.372)	**(0.222)**	(0.208)	(0.218)	(0.231)	(0.246)

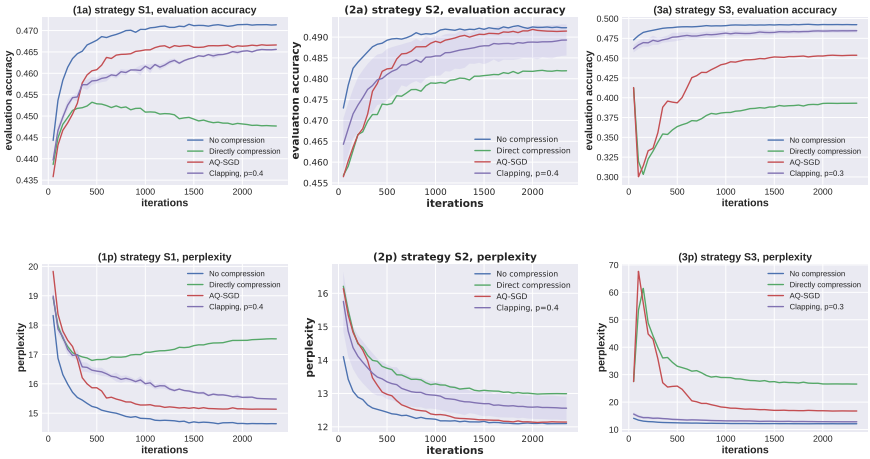

Figure 5: The evaluation accuracy and perplexity of GPT-2 fine-tuning with different compression strategies and difference compression algorithms. (Left: Strategy S1, Middle: Strategy S2, Right: Strategy S3.)

For each compression strategies, we independently repeat `Clapping` with different lazy sampling coefficient p_t including $\{0.1, 0.2, 0.3, 0.4, 0.6, 0.8\}$ for 3 times and select the p_t with the best average evaluation accuracy.

Figure 5 illustrates the evaluation accuracy and perplexity of `Clapping` with best p_t as well as the other algorithms. It can be observed that `Clapping` outperforms direct compression. Moreover, lazy sampling can improve the evaluation accuracy. In the case, `Clapping` achieves a better accuracy than AQ-SGD because AQ-SGD suffers from a high compression error in the beginning.

And we also present the best evaluation accuracy and perplexity of `Clapping` , which is shown in Table 15. From Table 15, we can observe that a suitable p_t like 0.3 or 0.4 can benefit the convergence and generalization in the fine-tuning tasks. And a large p_t may harm the generalization, which matches our discussion in Section 4.2.

Fine-tuning GPT-2 on arXiv abstracts. We also respectively fine-tuned pretrained GPT-2 small models and GPT-2 medium models [46] on a dataset with 30K arXiv abstracts [67] on 8 NVIDIA RTX 4090 GPUs with communication compression algorithms for pipeline-parallel learning including directly compression, `Clapping`-**FC** without lazy sampling and `Clapping`-**FC** with lazy sampling. We also compared those approaches to the case with no compression. The block size was set to 1024. We set data parallel degree 8 with the macro-batch size 16 and the micro-batch size 2. As each micro-batch is computed in a single GPU, we simulate the communication compression by adding the corresponding error to the activation and gradient. And we fine-tuned the model for 8 epochs with other competitive algorithms and the same iterations with `Clapping` , respectively. We simulated the communication compression for each GPU by adding the compression hook with Top-K [69] compressor that keep 50% of the elements. For GPT-2 small, we add the hook at the end of the 2-nd, 5-th, and 8-th transformer layer. For GPT-2 medium, we add the hook at the end of the 4-th, 10-th, and 16-th transformer layer.

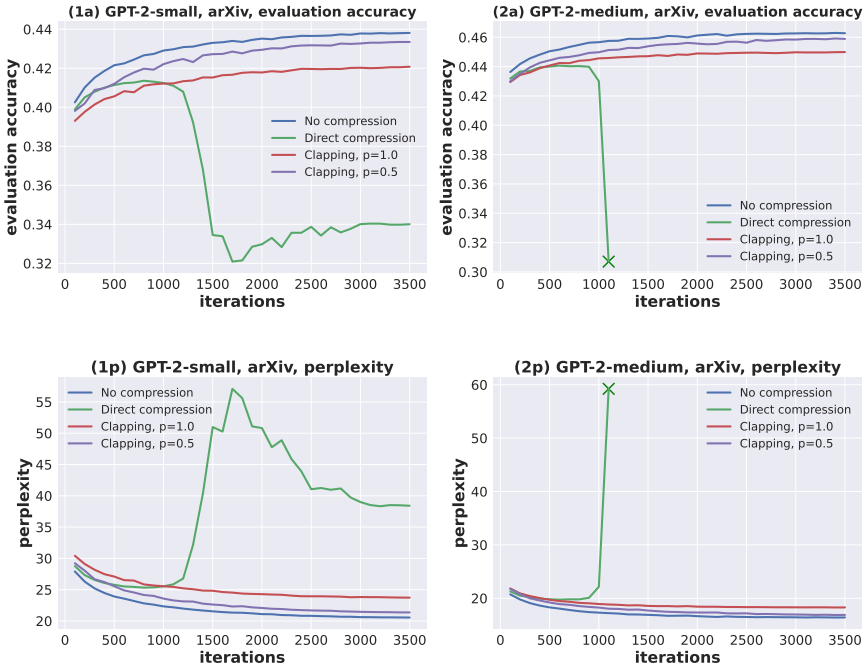

Figure 6: The evaluation accuracy and perplexity of GPT-2 small and GPT-2 medium fine-tuning by arXiv with Top50%. (Left: GPT-2 small, Right: GPT-2 medium.)

We use the AdamW optimizer [36] and FP16 for fine-tuning. The learning rate is initialized from 5×10^{-5} and scheduled by a cosine scheduler. The lazy sampling coefficient p_t for `Clapping` with lazy sampling was set to 0.5. Figure 6 illustrates the evaluation accuracy and perplexity. It can be observed that both `Clapping` with lazy sampling and `Clapping` without lazy sampling outperforms direct compression. Moreover, lazy sampling can make the evaluation accuracy and perplexity comparable to those of non-compressed case. Specifically, we can find that direct compression in GPT-2 medium suffers from the non-convergence but `Clapping` not, which can illustrates the benefit of error feedback technique.

Fine-tuning LLaMA2-7B on Wikitext. We fine-tuned a pre-trained LLaMA2-7B model [61] on Wikitext dataset (wikitext-2-raw-v1 version) [38] on 4 NVIDIA A100 40G GPUs with communication compression algorithms for pipeline-parallel learning including directly compression, `Clapping`-**FC** without lazy sampling and `Clapping`-**FC** with $p = 0.3, 0.5$. We also compared those approaches to the case with no compression. We set data parallel degree 4 with the macro-batch size 8 and the micro-batch size 2. As each micro-batch is computed in a single GPU, we simulate the communication compression by adding the corresponding error to the activation and gradient. We simulated the communication compression for each GPU by adding the compression hook at the end of 8-th, 16-th, and 24-th transformer layers. The block size was set to 1024. We fine-tuned the model for 6 epoch with the other competitive algorithms and the same iterations with `Clapping` , respectively. We used the SGD optimizer with momentum 0.9 and FP16 for fine-tuning. The learning rate is 5×10^{-5}. And we used TopK [69] that keeps 50% of elements and natural compression [19] as the compressor, respectively. According to [19], FP16 training with natural compression can compression the activations and gradients to 6-bit, thus the communication overhead is 37.5% compared to the non-compressed scenario.

Figure 7 shows the evaluation accuracy and perplexity of different algorithms in LLaMA-2 fine-tuning tasks. It can be observed that `Clapping` outperforms directly compression in the evaluation scalability. It is also noteworthy that for both compressor, `Clapping` can achieve nearly the **SAME** evaluation accuracy and perplexity as the non-compressed case. Thus one can tune a suitable p_t to let `Clapping` totally eliminate the negative impact of communication compression with more than $2\times$ communication saving. Figure 8 illustrate the evaluation accuracy and perplexity of different

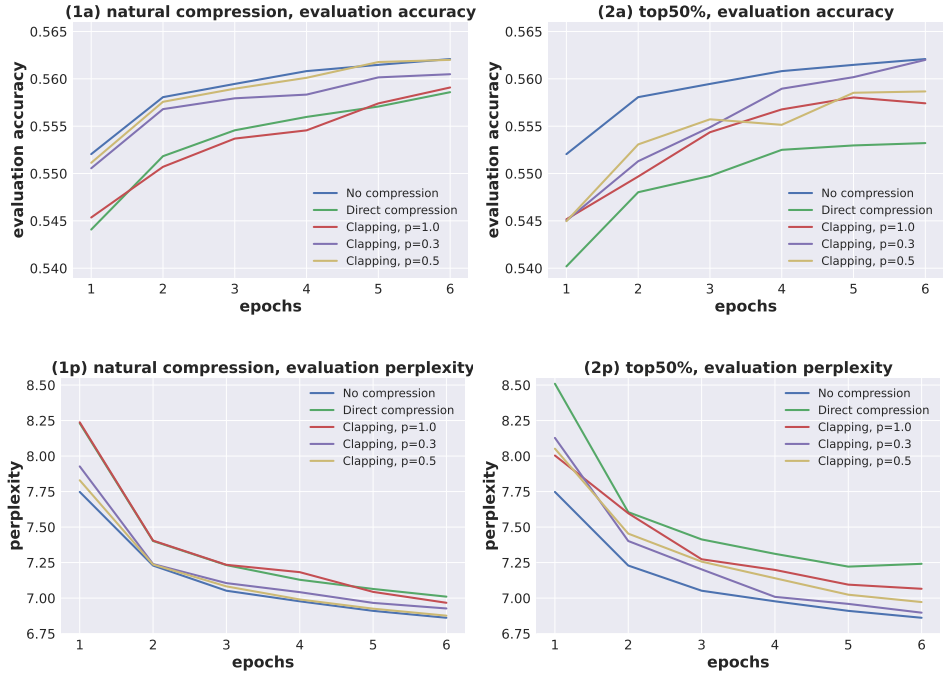

Figure 7: The evaluation accuracy and perplexity of LLaMA-2 fine-tuning with Wikitext-2 with different compressors and different compression algorithms. (Left: natural compression. Right: Top50%)

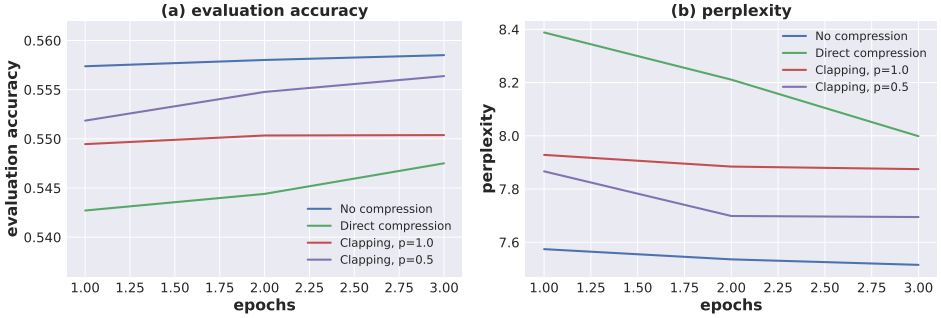

Figure 8: The evaluation accuracy of various approaches in LLaMA-2 fine-tuning task by arXiv abstracts.

compression algorithms. It can be observed the benefit of error feedback technique and lazy sampling strategy.

Fine-tuning LLaMA2-7B on arXiv abstracts. We fine-tuned a pre-trained LLaMA2-7B model [61] on a dataset with 30K arXiv abstracts [38] on 4 NVIDIA A100 40G GPUs with communication compression algorithms for pipeline-parallel learning including directly compression, `Clapping`-**FC** without lazy sampling and `Clapping`-**FC** with $p = 0.5$. We also compared those approaches to the case with no compression. We set data parallel degree 4 with the macro-batch size 2 and the micro-batch size 8. We simulated the communication compression for each GPU by adding the corresponding error to the activation and gradient at the end of 8-th, 16-th, and 24-th transformer layers. The block size was set to 1024. We fine-tuned the model for 3 epochs with the other competitive algorithms and the same iterations for `Clapping` , respectively. We used the SGD optimizer with momentum 0.9 and FP16 for fine-tuning. The learning rate is 5×10^{-5}. And we used TopK [69] that keeps 50% of elements as the compressor.

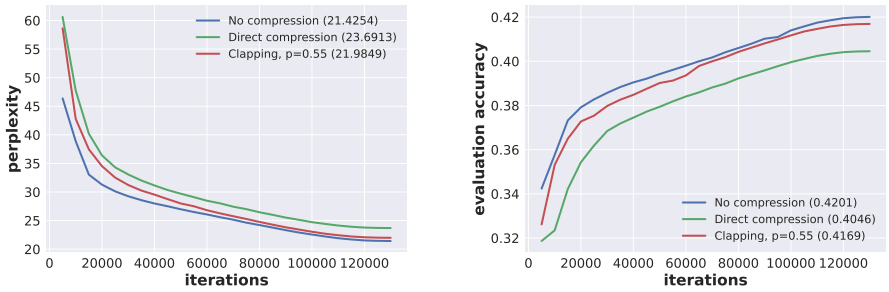

Figure 9: The evaluation perplexity (left) and accuracy (right) with various communication compression algorithms for pre-training GPT-2.

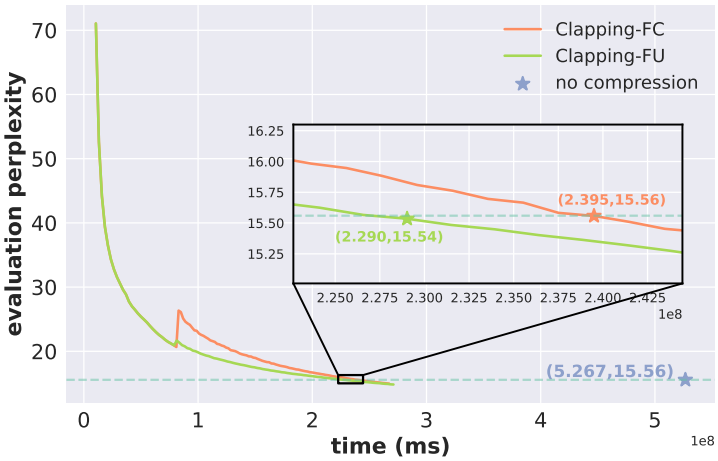

Figure 10: The evaluation perplexity with various communication compression algorithms for pre-training LLaMA-2 1B.

H.4 Pre-training LLMs

H.4.1 Pre-training GPT-2 small with multiple compression.

We pre-trained a GPT-2 small [46] model on openwebtext [45] on 8 NVIDIA RTX 4090 GPUs with 24GB of memory using communication compression algorithms for pipeline-parallel learning including directly compression and `Clapping` with lazy sampling and batch rule. Specifically, we cleared the cache for error feedback unless the former samples are kept. The block size was set to 1024. We set data parallel degree 8 with the macro-batch size 64 and the micro-batch size 8. We simulated the communication compression for each GPU by adding the corresponding error to the activation and gradient. We use the AdamW optimizer [36] and FP16 for training. We used natural compression [19] as compressor and added the compressor at the end of the 4-th and 8-th transformer layer. According to [19], FP16 training with natural compression can compression the activations and gradients to 6-bit, thus the communication overhead is 37.5% compared to the non-compressed scenario.

We trained the model for 130800 iterations for each algorithms as the total sample complexity is nearly equal to one epoch. The learning is initialized from 6×10^{-4} with 2000 warm-up steps and scheduled by a cosine scheduler. For `Clapping` , we obtain the evaluation perplexity after 5000 steps with different lazy sampling coefficient $p_t \in \{0.2, 0.4, 0.45, 0.5, 0.55, 0.6, 0.8\}$ and finally selected the best one $p_t = 0.55$.

Table 16: Single-iteration time (ms) and speedup of Clapping-FC and non-compression across different network bandwidths

Method	**100 MB/s**	**200 MB/s**	**300 MB/s**	**400 MB/s**	**500 MB/s**
Clapping-FC	572.68	460.18	422.68	403.93	392.68
No Compression	947.68	647.61	547.28	497.63	467.68
Speedup	1.65×	1.41×	1.30×	1.23×	1.19

Table 17: Comparison of training time ($\times 10^5$ s) to reach target evaluation perplexity for LLaMA-2 3B pre-training under bandwidth constraint

Target evaluation perplexity	**17.5**	**17.0**
No compression	2.34036	2.69330
Clapping-FC	2.08600	2.42791

Figure 9 has shown the evaluation perplexity, and evaluation accuracy of the pre-training task with different compression algorithms. They illustrates `Clapping` outperforms directly compression in both training and evaluation.

H.4.2 Pre-training LLaMA-2 1B

The 1B C4 pre-training results use random seed 42 and correspond to one training run, not a mean over multiple random seeds.

We pre-trained a LLaMA-2 1B model [61] on the C4 dataset [48] using 8 NVIDIA A800 GPUs (80GB memory) with pipeline-parallel learning. Three communication compression strategies were compared: direct compression, `Clapping`-**FU**, `Clapping`-**FC**, and a baseline without compression. The model was split after the 16th transformer layer with a data parallel degree of 4. For communication compression, activations/gradients were quantized to 6 bits followed by Top-30% sparsification (also quantized to 6 bits). Compression was disabled during the initial 15,000 iterations. Network bandwidth was constrained to 100 MB/s throughout training. We employed the AdamW optimizer for 100,000 iterations, with coefficient p_t initialized at 0.5 and progressively increased to 1.0 via cosine scheduling. Other hyperparameters followed [75]. The baseline perplexity and total time were derived from [75], where total time was extrapolated from the first 20,000 training steps.

Figure 3 demonstrates the training dynamics. Direct compression fails to converge, while both Clapping-FU and Clapping-FC remain stable and reach an evaluation perplexity comparable to the uncompressed reference trajectory.

The wall-clock time comparison under the vanilla F-then-B strategy is shown in Figure 10. It can be observed that both `Clapping`-**FU** and `Clapping`-**FC** reach the same target perplexity in substantially less time than the no-compression baseline in this communication-bound setting, which is primarily attributable to reduced inter-stage communication. We emphasize that CLAPPING is proposed as an optimization-stability mechanism under aggressive compression, rather than a guarantee of universal end-to-end speedups across different hardware interconnect regimes.

Notably, `Clapping` achieves evaluation perplexity comparable to that of the uncompressed scenario. This indicates that even though the lazy sampling technique in `Clapping` reduces the number of samples input to the model, an appropriate choice of the lazy sampling coefficient p can preserve the model's expressive power with negligible degradation. This result demonstrates that `Clapping` , combined with lazy sampling, enables an effective trade-off between communication compression and model convergence and expressive capability.

In this experiment, the compressor achieves approximately 50% reduction in communication overhead. However, existing empirical results illustrate that communication compression can improve evaluation performance [49]. One possible explanation is that the noise introduced by the compressor implicitly helps avoid overfitting and leads to convergence to a minimum that exhibits better generalization ability. Thus, training with `Clapping` can achieve acceleration beyond mere communication overhead reduction. We plan to investigate how communication compression improves evaluation performance in future work.

Table 18: Comparison of evaluation loss across different compression methods

Top-50%	**8-bit Quantization**	**Natural Compression**	**Hybrid Compressor**
5.0260	3.1425	3.0990	2.8944

Table 19: Reported validation perplexity for a matched LLaMA-2 1B long-sequence experiment (lower is better).

Metric	No compression	Clapping-FU	Clapping-FC
Validation PPL ↓	14.62	14.68	14.84

Table 20: Gradient fidelity and inter-stage boundary errors during 2,000 LLaMA-2 1B C4 steps, starting from a common step-20,000 checkpoint. Lower deviation/error and higher cosine are better.

Method	Rel. deviation ↓	Cosine ↑	Forward err. ↓	Backward err. ↓
Direct	0.327640	0.946658	—	—
Clapping-FC	0.079157	0.976070	0.003331	0.012016
Clapping-FU	0.010587	0.999915	0.001501	0.005313

Table 21: Global gradient-norm statistics for the 2,000-step LLaMA-2 1B C4 diagnostic.

Statistic	Direct	Clapping-FC	Clapping-FU
Mean	0.103631	0.134602	0.115916
Variance	0.000047	0.002516	0.000231

H.4.3 Long-sequence perplexity comparison

In a matched LLaMA-2 1B long-sequence experiment, the reported validation perplexities are shown in Table 19. The available record does not specify the exact sequence length or the full hyperparameter configuration; we therefore report only the available metric and do not equate this setting with the 1B C4 training configuration in Appendix H.4.2.

H.4.4 Gradient fidelity and optimization stability

We measure the effective gradient generated by each communication method against the uncompressed stochastic gradient on the same batch and at the same model state. The experiment begins from a shared step-20,000 NoComp checkpoint of the LLaMA-2 1B C4 pre-training task. Direct, FC, and FU are then trained separately for 2,000 steps. At each step, the uncompressed gradient is computed as a diagnostic reference and discarded; only the method-specific gradient estimate is used to update model parameters. The statistics below are aggregated over this 2,000-step window.

Table 20 reports the relative deviation from the uncompressed stochastic gradient, cosine similarity to that gradient, and forward/backward boundary errors. Boundary-error values are unavailable for Direct and are therefore indicated by dashes. These are *realized-gradient* diagnostics; a high cosine similarity does not establish that the compression-induced gradient error has zero conditional expectation.

Table 21 reports the mean and variance of the global gradient norm of the *effective gradient used for optimization*, computed over the same 2,000-step window.

These results demonstrate that Clapping effectively mitigates compression-induced gradient distortions, achieving more accurate update directions and stable gradient norms. Together with the convergence analysis, these findings highlight the effectiveness of error feedback and lazy sampling in controlling gradient errors and maintaining optimization stability under communication compression.

H.4.5 Bandwidth sensitivity and larger-scale pre-training task for LLaMA-2 3B

When is inter-stage compression beneficial? `Clapping` targets *communication-bound* pipeline parallelism, where transmitting high-dimensional activations/adjoints dominates per-step latency (cf.

Table 22: Measured LLaMA-2 13B, PP=8, 1F1B steady-state system performance; 50 timed steps after five warm-up steps.

Method	Step time ↓	Speedup ↑	Time reduction
Direct	5.5826 s	2.0888×	52.13%
Clapping-FC	5.5961 s	2.0838×	52.01%
Clapping-FU	6.7694 s	1.7226×	41.95%
NoComp	11.6610 s	1.0000×	—

Figure 2). In tightly coupled intra-node settings with high-speed interconnects (e.g., NVLink/NVSwitch), communication is often not the bottleneck and may already be hidden by compute and micro-batching; in such regimes, enabling compression can introduce unnecessary overhead and should be selectively disabled. In contrast, in cross-node or geo-distributed deployments over commodity links, bandwidth can be the dominant bottleneck, motivating communication-efficient pipeline-parallel optimization that remains stable under aggressive activation/gradient compression.

Pareto-style per-iteration trade-off under bandwidth caps. To characterize the practical trade-off between communication bandwidth and iteration time, we run a controlled per-iteration microbenchmark using LLaMA-2 3B with micro-batch size 8 on 8 NVIDIA A100 GPUs. We apply a hybrid 6-bit compression strategy to both activations and gradients on three pipeline boundaries, and sweep the effective inter-stage bandwidth cap. Table 16 reports the total single-iteration time (forward + backward + update + communication) and the resulting speedup over the no-compression baseline. As expected, the advantage is most pronounced in communication-bound regimes and diminishes as bandwidth increases.

Time-to-target perplexity on LLaMA-2 3B. To validate `Clapping` on a larger-scale pre-training task, we further pre-train LLaMA-2 3B under a fixed bandwidth cap (50 MB/s) using the same compressor configuration as in the LLaMA-2 1B pre-training setup. We disable the compressor during the first 8,000 steps and train for 50,000 steps total. Table 17 compares the wall-clock time required for the no-compression baseline and `Clapping`-**FC** to reach the same target validation perplexity, showing consistent efficiency gains in the targeted communication-bound regime.

H.4.6 Measured LLaMA-2 13B pipeline-parallel step time

We evaluate the system-level efficiency of Clapping on LLaMA-2 13B using eight GPUs with pipeline parallelism ($\mathrm{PP} = 8$) and a Megatron-style 1F1B schedule. Each GPU hosts one pipeline stage, with activation and backward-gradient tensors exchanged through actual point-to-point communication. All methods share identical model partitions, micro-batch configurations, and execution schedules to ensure a fair comparison. After five warm-up steps, we measure the average step time over 50 steady-state iterations. As shown in Table 22, Clapping-FC and Clapping-FU achieve **2.08× and 1.72× end-to-end speedups**, respectively, reducing step time by **52.01% and 41.95%** compared with uncompressed training. These results demonstrate that Clapping translates communication compression into substantial practical training acceleration on a large-scale pipeline-parallel system.

H.4.7 Ablations across compressor families

To probe robustness across compressor designs commonly used in communication-efficient training, we implement four representative compressor families spanning sparsification, quantization, and hybrid pipelines: (i) Top-k sparsification, (ii) direct low-bit quantization, (iii) Natural Compression, and (iv) a hybrid compressor that combines quantization and sparsification (matching the configuration used in Appendix H.4.2).

While `Clapping` is compatible with all of the above compressors, the choice of compressor can materially affect the final evaluation loss in pre-training. Table 18 summarizes the comparison on a controlled LLaMA-2 250M pre-training setup, highlighting that selecting an appropriate compressor can yield better optimization outcomes under the same communication budget.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:Refer to Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Refer to Section Conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Refer to Section Theoretical Analysis for our assumptions, and Appendix for detailed proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer:[Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets we used are all open-accessed. We present the core codes in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We show the variance of experimental results in experiments where we consider them essential.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no such risk in the paper.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We comply with the licenses of existing assets used in the paper and provide necessary references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The proposed algorithm can be used for LLMs pre-training and fine-tuning. Thus we present some experiments about LLMs pre-training and fine-tuning. Meanwhile, we use LLMs for editing.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.