γ-FedHT: Stepsize-Aware Hard-Threshold Gradient Compression in Federated Learning

Rongwei Lu[†], Yutong Jiang[†], Jinrui Zhang[†], Chunyang Li[‡*], Yifei Zhu[§] Bin Chen[‡], Zhi Wang[†**]
[†]Tsinghua Shenzhen International Graduate School, Tsinghua University
[‡]Harbin Institute of Technology, Shenzhen
[§]UM-SJTU Joint Institute, Shanghai Jiao Tong University
{lurw24, jiang-yt24, zhangjr23} @mails.tsinghua.edu.cn, 210110812@stu.hit.edu.cn,
yifei.zhu@sjtu.edu.cn, chenbin2021@hit.edu.cn, wangzhi@sz.tsinghua.edu.cn

***Abstract*—Gradient compression can effectively alleviate communication bottlenecks in Federated Learning (FL). Contemporary state-of-the-art sparse compressors, such as Top-k, exhibit high computational complexity, up to $\mathcal{O}(d \log_2 k)$, where d is the number of model parameters. The hard-threshold compressor, which simply transmits elements with absolute values higher than a fixed threshold, is thus proposed to reduce the complexity to $\mathcal{O}(d)$. However, the hard-threshold compression causes accuracy degradation in FL, where the datasets are non-IID and the stepsize γ is decreasing for model convergence. The decaying stepsize reduces the updates and causes the compression ratio of the hard-threshold compression to drop rapidly to an aggressive ratio. At or below this ratio, the model accuracy has been observed to degrade severely. To address this, we propose γ-FedHT, a stepsize-aware low-cost compressor with Error-Feedback to guarantee convergence. Given that the traditional theoretical framework of FL does not consider Error-Feedback, we introduce the fundamental conversation of Error-Feedback. We prove that γ-FedHT has the convergence rate of $\mathcal{O}(\frac{1}{T})$ (T representing total training iterations) under μ-strongly convex cases and $\mathcal{O}(\frac{1}{\sqrt{T}})$ under non-convex cases, *same as FedAVG*. Extensive experiments demonstrate that γ-FedHT improves accuracy by up to 7.42% over Top-k under equal communication traffic on various non-IID image datasets.**

***Index Terms*—Federated Learning, Adaptive Gradient Compression, Convergence Analysis, Hard-Threshold Sparsification**

I. Introduction

FL is an increasingly important Distributed Machine Learning (DML) framework that addresses the critical need for data privacy in model training across multiple edge nodes [1], [2]. FL requires a decaying stepsize[1], not a fixed one, to ensure model convergence in non-IID scenarios [4], which are common in FL [5]. In FL, gradient compression has been widely adopted to alleviate communication bottlenecks. The classic process of FL training with gradient compression involves three main steps: (1) clients train the local model for several iterations to obtain updates; (2) clients compress the updates and send them to the central server; (3) the server decompresses the updates, aggregates them (*e.g.*, averaging them in FedAVG [3]), and updates the global model. Gradient compression methods can be classified into three categories: (1) sparsification, transmitting a part of the gradients; (2) quantization, mapping high-precision elements into low-precision ones; and (3) low-rank, decomposing the gradient into two low-rank matrices. Sparsification compression is often preferred due to its superior efficiency in reducing redundant gradient information. The gradient sparsification usually comes with Error-Feedback (EF) [6], [7], a popular mechanism that collects and reuses the errors from the gradient compression to mitigate the compression bias and guarantee convergence. The two popular sparsification compressors are the Top-k compressor [8] and the hard-threshold compressor [9]. The Top-k compressor transmits elements with the top k absolute values, while the hard-threshold compressor transmits elements with absolute values larger than a threshold.

Top-k is recognized as the state-of-the-art (SOTA) sparsification compressor in FL[2], but its counterpart, the hard-threshold compressor, is not suitable for FL. *We are the first to demonstrate that the hard-threshold compressor shows inferior convergence rates compared to Top-k in FL* (as shown in Fig. 1 in Sec. III-A) by the control variable method. This is because the hard-threshold compression is sensitive to the combination of the decaying stepsize and non-IID cases. In particular, *we examine the sensitivity of the hard-threshold compression to such cases* (as shown in Fig. 2 in Sec. III-B) through the full factorial experimental design [13], an approach involving systematically varying all experimental factors and their combinations to understand their effects comprehensively. The compression ratio of the hard-threshold compressor drops rapidly to an extremely aggressive value (like 0.1% for a CNN model with CIFAR-10 datasets [14]) in the combination of the decaying stepsize and non-IID cases. Such aggressive compression stops the model from converging in non-IID scenarios, thereby reducing the model accuracy.

Although the hard-threshold compression degrades the accuracy, the lightweight compression is extremely appealing in FL. In fact, the computation cost of Top-k is up to hundreds

* Chunyang Li has been pre-admitted to Tsinghua Shenzhen International Graduate School when doing this work.

**Corresponding author.

[1]We focus on the original FedAVG [3] with a single learning rate on the client side instead of two on both the server and client sides.

[2]Many well-performing hybrid gradient compressors in FL use Top-k for sparsification [10]–[12].

of times of the hard-threshold compression[3], primarily due to two reasons: (1) Top-k selection has a time complexity of $O(d \log_2 k)$, and $\log_2 k$ depends on the model scale (*e.g.*, it can be up to nearly 30 for GPT2 [16]), while the hard-threshold compression requires traversal of parameters with the time complexity of $O(d)$; and (2) Top-k selection does not perform well on accelerators such as GPUs [15]. Clearly, the ideal sparsification compressor in FL is one that has both the low-cost computation of the hard-threshold compression and the superior performance of Top-k. *Sadly, no sparsification compressor in FL has been developed to have a time complexity of $O(d)$ and the same theoretical convergence rate as vanilla FedAVG so far.*

We propose γ-FedHT (as shown in Algo. 1 in Sec. IV), an ideal sparsification compressor in FL satisfying the above properties. γ-FedHT is a stepsize-aware hard-threshold compressor with vanilla EF, avoiding the accelerator-unfriendly operations like Top-k selection, and inheriting the low-cost property. To improve the performance, the threshold should satisfy the increasing and then decreasing monotonicity with a limit of zero. Combining two simple functions, the inverse proportional function and the logarithmic function, the adaptive threshold can satisfy the two mathematical properties without introducing more hyperparameters. Although there have been efforts to theoretically validate gradient compression algorithms in FL [10], [17], these works have not considered EF, which is important and necessary for sparsification compression. To derive the convergence rate of our design, we solve the problem of *how to integrate gradient compression with EF into the theoretical framework of FL.* We fuse the mathematical description of EF into the framework and establish an iterative equation. Based on this, we derive the convergence rates. The convergence rates of γ-FedHT are $\mathcal{O}(\frac{1}{T})$ under μ-strongly convex functions and $\mathcal{O}(\frac{1}{\sqrt{T}})$ under non-convex functions, *the same rate as FedAVG without compression*.

Our contributions are as follows:

- We are the first to reveal that the model trained with the hard-threshold compression converges less effectively than the one trained with Top-k compressor by the controlled variable method. We use a full factorial experimental design to demonstrate that it is the combination of the decaying stepsize and non-IID scenarios that contributes to the failure of the hard-threshold compression in FL.
- We propose γ-FedHT, the first sparsification compressor in FL with a time complexity of $\mathcal{O}(d)$ and the same convergence rate as vanilla FedAVG. We expand the application of the traditional FL theoretical framework and derive the convergence rate of FedAVG with gradient compression and EF, based on introducing the iterative equation of EF.
- We apply γ-FedHT to both real-world non-IID and artificially partitioned non-IID datasets, including convex cases (*e.g.* Logistic) and non-convex cases (*e.g.* VGG, CNN and GPT2). The experimental results validate the great compression-accuracy trade-offs of our design. Under equal traffic communication, γ-FedHT can improve accuracy by up to 7.42% over Top-k on the CNN model with non-IID datasets.

[3]According to Fig. 15(d) in the appendix of the previous work [15], the compression time of Top-k is hundreds of times that of SIDCo. Furthermore, the absolute compressor is more effective than SIDCo.

II. Preliminaries

Our research focuses on the synchronous FedAVG algorithm with the sparsification compressor. We aim to provide a succinct overview of the optimization problem of FedAVG, the differences between FL and traditional DML, gradient compression, and Error-Feedback, with a particular emphasis on the hard-threshold compressor.

The optimization problem of FL: the optimization problem of FL minimizes a loss function f as follows:

$$\min_{\mathbf{x}} \left[f(\mathbf{x}) := \sum_{i=1}^{n} p_i f_i(\mathbf{x}) \right],$$

where n represents the number of clients. The i-th client possesses a mutually disjoint partition D^i of the overall training dataset D and the training weight $p_i = \frac{|D^i|}{\sum_{i'=1}^{n} |D^{i'}|}$. The local training target $f_i(\mathbf{x})$ is the loss function evaluated on D^i.

FL vs. Traditional DML: FL distinguishes itself from traditional DML in the following four fundamental aspects:

- *Non-IID:* In FL, to safeguard data privacy, datasets cannot be exchanged between nodes, resulting in unbalanced data distributions and quantities across nodes [4]. This contrasts with traditional DML, where datasets are uniformly partitioned across nodes, typically yielding IID datasets.
- *Decaying-γ:* FL necessitates the stepsize γ that decays to zero to guarantee model convergence [5]. Fixed-γ in FL can cause significant deviation of the global model from the optimal, with the L2 norm of the difference being proportionate to γ^2. However, in traditional DML, decaying-γ is not necessary.
- *Infrequent communication:* FL is characterized by low bandwidth and high latency due to the training across WAN [18]. This necessitates infrequent communication, where aggregation occurs after several training iterations, not after each one. This study adheres to a fixed communication frequency E, consistent with vanilla FedAVG, and does not explore adaptive strategies for E.
- *Partial node participation:* Due to device heterogeneity and unreliability in FL, aggregation rounds typically involve only the fastest-responding nodes, avoiding delays from slower ones. In contrast, DML benefits from homogeneous and reliable nodes, allowing consistent participation in all aggregation rounds. This paper focuses on the strategy of uniform random node selection [4].

Gradient compression: According to whether the mathematical expectation of the gradient changes before and after compression [8], compressors can be classified into biased and unbiased compressors. The biased compressors with EF [19] are widely used in FL because they can apply more aggressive compression than unbiased compressors.

The hard-threshold compressor: The hard-threshold algorithm, also known as the threshold-λ sparsification compressor, is a variant of Top-k, with which it shares a certain conversion relationship: a given k corresponds to a specific λ. With the objective of minimizing the sum of compression errors throughout the entire training, the hard-threshold achieves a more favorable compression-accuracy trade-off than Top-k [9]. The hard-threshold compressor is also called the absolute compressor due to its key property that the error term possesses an upper bound independent of $\mathbf{x}$. We denote the threshold as λ and the absolute compressor as $\mathbf{C}_\lambda(\cdot)$. $\mathbf{C}_\lambda(\cdot)$ represents a mapping: $\mathbb{R}^d \to \mathbb{R}^d$, characterized by the following property:

$$\mathbb{E}_{\mathbf{C}_\lambda}\|\mathbf{C}_\lambda(\mathbf{x}) - \mathbf{x}\|^2 \leq d\gamma^2\lambda^2.$$

We focus on FedAVG with the hard-threshold compression, which has not been investigated in FL.

Error-Feedback: The error-feedback mechanism [7] in gradient compression involves storing and accumulating compression errors over iterations, which are then added to the gradient in subsequent steps to ensure accurate gradient updates and mitigate the loss of information due to compression [20]–[23]. Some works propose new EF mechanisms to guarantee sharper convergence rate, like EF21 [24] and EControl [25], by introducing momentum terms or other compensation terms. We focus on vanilla EF for two reasons: (1) EF is orthogonal to sparsification compressors (*e.g.*, EF and EF21 are orthogonal to sparsification); and (2) New EF mechanisms tend to introduce hyperparameters or require additional storage space, complicating the optimization problem (*e.g.*, the performance of EControl is sensitive to the hyperparameter).

III. Motivation

This measurement answers the following questions:

- Does the hard-threshold compressor induce non-convergence in FL? If so, is the degree of non-convergence correlated with λ? (Fig. 1 in Sec. III-A)
- What are the underlying causes for the non-convergence exhibited by the hard-threshold compressor in FL? (Fig. 2 in Sec. III-B)

Specifically, we employ two measurements to validate our motivation thoroughly. 1) Logistic Regression [12] on Fashion-MNIST [26] dataset (denoted as Logistic@FMNIST), which is the most classical convex case [17]. 2) CNN on CIFAR-10 [27] dataset (denoted as CNN@CIFAR-10), a widely-used non-convex scenario [17]. To align with previous works [4], [28], we set the number of clients as 10, the communication frequency $E = 5$ (*i.e.*, global communication after every 5 local iterations). We denote the stepsize at the t-th iteration as γ_t and set $\gamma_t = \frac{100}{t+1000}$. The non-IID partition strategy for Logistic@FMNIST (as well as CNN@CIFAR-10) is $\#C = 2$ ($\#C = 5$) quantity-based label imbalance [29], where $\#C = 2$ (as well as $\#C = 5$) means that each node owns data samples of 2 (5) labels and there is no overlap between the samples of each partition. We set vanilla FedAVG as the baseline.

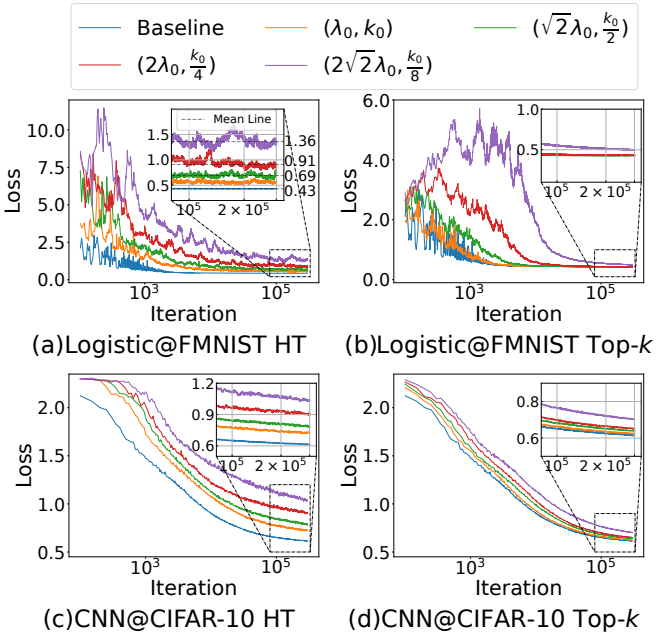

Fig. 1: The global loss curves (Loss vs. Iterations) for FedAVG with hard-threshold compression (denoted as HT, left) and Top-k (right) on different tasks (top to bottom). λ_0 is $\frac{\sqrt{2}}{2}$ (as well as $\frac{\sqrt{2}}{10}$) in Logistic@FMNIST (CNN@CIFAR-10). k_0 is 1% for two tasks. The loss curves in (a, c) do not converge to the baseline, while those in (b, d) converge.

A. Poor Convergence of the Hard-Threshold Compression

In Fig. 1, the comparative analysis of Top-k and the hard-threshold compression reveals that the hard-threshold compression makes the global model far from the optimal model. Specifically, the loss curves of Top-k (b, d) converge to the baseline, while ones of hard-threshold compression (a, c) does not converge to the baseline. This discrepancy indicates that the *hard-threshold compressor introduces the accuracy degradation and the convergence rate of FedAVG with hard-threshold compression cannot reach that of vanilla FedAVG*, both for the convex case and the non-convex one.

The l2-norm between the global model parameters and the optimal model is positively correlated with λ. In Fig. 1(a) and 1(c), we observe that the larger the value of λ, the greater the divergence between the loss of FedAVG with the hard-threshold compressor and the baseline loss. This indicates an increasingly wider gap between the performance of the global model and optimal model[4]. Moreover, in convex scenarios, we

[4]This is referring to Appendix C-3 in [4]. We consider the global model to have converged to the optimal model when its loss matches the baseline loss. Otherwise, there is a distance between the global and optimal models.

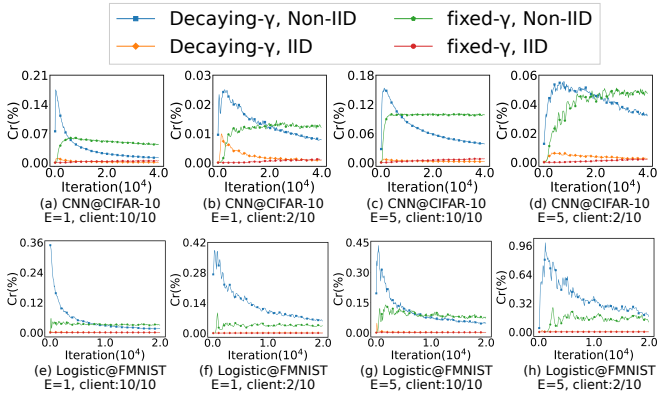

Fig. 2: Compression curves (Compression ratio vs. Iterations) for HT with $\lambda = 1$ under different settings in the convex (a-d) and non-convex (e-h) cases. Here, $E = 1$ and $E = 5$ denote frequent and infrequent communication, respectively, while Client: $10/10$ and $2/10$ indicate partial and full partition scenarios. Each subplot consistently demonstrates that in non-IID scenarios, the decaying-γ prompts the hard-threshold compressor to engage in increasingly aggressive compression strategies as training progresses into the late stage.

observed a positive correlation between the distance of losses and λ^2.

B. Peeking Behind the Curtains of Poor Convergence

Referring to the part of *FL vs. traditional DML* detailed in Sec. II, we identify four key differences between FL and traditional DML. Our objective is to explore which of these factors (or a combination of factors) leads to the failure of the hard-threshold compression. For this purpose, we adopt a full factorial experimental design, an approach that systematically tests all possible combinations of the factors under consideration. We conduct experiments using both the inclusion and exclusion of these four factors, resulting in 16 experiments in Fig. 2.

Decaying-γ in non-IID scenarios leading to overly aggressive compression during the late training. As the convex and non-convex ones come up with the same conclusion, we take the convex case as an example in the following content. In Fig. 2, we observe a unique phenomenon where the relative compression ratio initially increases and then decreases during the whole training process, occurring under the simultaneous conditions of the decaying-γ and non-IID scenarios. In IID settings, the model converges rapidly in the early stage, allowing the hard-threshold compression to transmit a substantial amount of parameters before convergence. Once convergence is achieved, the algorithm automatically inhibits the transmission of extraneous gradients, thereby striking an efficient balance. However, in non-IID scenarios, the model converges more slowly. The decaying-γ leads to smaller updates, which, when faced with a fixed threshold, result in increasingly aggressive compression strategies. This leads to a stagnation in model convergence during the late training. The settings of infrequent communication and partial node participation do not affect the trend of the compression ratio curve, but only alter the relative magnitude of the compression ratio.

IV. The Stepsize-Aware Hard-Threshold Compression in FL

We aim to develop a low-cost adaptive hard-threshold compressor less sensitive to decaying-γ. We denote the threshold at the t-th iteration as λ_t, and the technical challenge is: *How to determine λ_t as a function of γ_t, i.e.*, let $\lambda_t^2 = \lambda_0^2 F(\gamma_t)$, how do we determine $F(\gamma_t)$[5]? It is difficult to carve $F(\gamma_t)$ out of simple functions because λ_t needs to satisfy the following mathematical properties: λ_t should initially increase and subsequently diminish to 0. This can be described as: (1) $\lim_{t->+\infty} F(\gamma_t) = 0$; (2) $\exists 0 < t_1 \leq t_2 < T$, $\forall t \in (0, t_1)$, $\frac{\mathrm{d}F}{\mathrm{d}t} \geq 0$ and $\forall t \in (t_2, T)$, $\frac{\mathrm{d}F}{\mathrm{d}t} \leq 0$. The initial increase aligns with the recommendation of employing conservative compression during the initial stages of training [30], and the subsequent decrease aims to slow down the decline of the compression ratio.

In order to reduce the construction space to get $F(\cdot)$, we let $t_1 = t_2$ and start with simple functions. Since a single class of simple functions cannot satisfy both properties, we consider a combination of two functions. We choose the inverse proportional function to fulfill the limit of 0. For the increasing and then decreasing monotonicity, we can select the quadratic function or the logarithmic function. We note that the combination of the inverse proportional function and the logarithmic function, *i.e.*, $F(\gamma_t) = (\gamma_t^{\alpha} + \gamma_t^{-\alpha})^{-1}$ (α is a constant and $\alpha \geq 1$), can satisfy the mathematical properties while introducing only one hyperparameter. Additionally, we normalize this function with the geometric mean of γ_t and determine $\lambda_t^2 = \lambda_0^2 \cdot \frac{\gamma_t^{\alpha}(\gamma_0\gamma_T)^{\frac{\alpha}{2}}}{\gamma_t^{2\alpha}+(\gamma_0\gamma_T)^{\alpha}}$, where γ_0 (as well as γ_T) is the start (end) value of γ.

Based on this, we propose γ-FedHT, with a time complexity of $\mathcal{O}(1)$ for calculating λ_t and $\mathcal{O}(d)$ for compressing gradients. Compared to the compression cost $\mathcal{O}(d)$, the time required to compute λ_t is negligible, so *the compression cost of γ-FedHT is as low as the hard-threshold compressor*. We show the convergence analysis in Sec. V. *We reveal that the convergence rate is fastest when $\alpha = 1$.* In other words, there is only one hyperparameter λ_0 in the γ-FedHT.

The pseudo-code of γ-FedHT is shown in Algo. 1. $\mathbf{C}_{\lambda_t}(\mathbf{x})$ means compressing the tensor $\mathbf{x}$ with the absolute compressor and the compression threshold is λ_t.

V. Theoretical Analysis

A. Regularity Assumptions

We follow assumptions, which are standard and widely accepted in the theoretical framework of DML [4], [17], [31].

Assumption 1 (L-smoothness). We assume L-smoothness of $f_i, i \in [n]$, that is for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\| \leq L\|\mathbf{y} - \mathbf{x}\|. \tag{1}$$

[5]The reason for using λ_t^2 instead of λ_t is that the compression error introduced by the hard-threshold compression is linearly related to λ_t^2 rather than to λ_t.

Algorithm 1: γ-FedHT

Input: number of workers n, training weight p_i, initial parameters $\mathbf{x}_0$, stepsize γ_t, absolute compressor $\mathbf{C}_\lambda(\cdot)$, initial threshold λ_0, α, communication frequency E, initial local error $\mathbf{e}_0^i = \mathbf{0}_d$

Output: $\mathbf{x}_T$

1 **for** $t = 0, \ldots, T-1$ **do**
2 **if** $t \mod E = 0$ **then**
3 Server picks nodes $\mathcal{S}_t$ uniformly at random;
4 **end**
  /* Worker side */
5 **for** $i \in \mathcal{S}_t$ **do**
6 **if** $t \mod E = 0$ **then**
7 Download $\mathbf{x}_t$ from the server;
8 **end**
9 $\mathbf{x}_{t+1}^i := \mathbf{x}_t^i - \gamma_t \mathbf{g}^i(\mathbf{x}_t)$, $\mathbf{e}_{t+1}^i := \mathbf{e}_t^i$;
10 **if** $t \mod E = E-1$ **then**
11 $\lambda_{t+1} = \lambda_0 \sqrt{\frac{\gamma_{t+1}^\alpha (\gamma_0\gamma_T)^{\alpha/2}}{\gamma_{t+1}^{2\alpha} + (\gamma_0\gamma_T)^\alpha}}$;
12 $\hat{\Delta}_t^i := \mathbf{C}_{\lambda_{t+1}}(\mathbf{e}_t^i + \mathbf{x}_{t+1}^i - \mathbf{x}_{t+1-E}^i)$;
13 $\mathbf{e}_{t+1}^i := \mathbf{e}_t^i + \mathbf{x}_{t+1}^i - \mathbf{x}_{t+1-E}^i - \hat{\Delta}_t^i$;
14 Upload $\hat{\Delta}_t^i$;
15 **end**
16 **end**
  /* Server side */
17 **if** $t \mod E = E-1$ **then**
18 Gather $\hat{\Delta}_t^i$ from nodes in $\mathcal{S}_t$;
19 $\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{n}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} p_i \hat{\Delta}_t^i$;
20 Broadcast $\mathbf{x}_{t+1}$;
21 **else**
22 $\mathbf{x}_{t+1} := \mathbf{x}_t$;
23 **end**
24 **end**
25 **Return** $\mathbf{x}_T$;

Assumption 2 (Bounded gradient noise). We assume that stochastic gradient oracles $\mathbf{g}^i(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^d$ are available for each $f_i, i \in [n]$. For simplicity, we only consider the instructive case where the noise is uniformly bounded for all $\mathbf{x}, \in \mathbb{R}^d, i \in [n]$:

$$\mathbf{g}^i(\mathbf{x}) = \nabla f_i(\mathbf{x}) + \boldsymbol{\xi}^i, \quad \mathbb{E}_{\boldsymbol{\xi}^i} \boldsymbol{\xi}^i = \mathbf{0}_d, \quad \mathbb{E}_{\boldsymbol{\xi}^i} \|\boldsymbol{\xi}^i\|^2 \leq \sigma^2. \quad (2)$$

Assumption 3 (Bounded gradient norm). We assume the expected squared norm of stochastic gradients is uniformly bounded:

$$\mathbb{E}_{\boldsymbol{\xi}^i} \|\mathbf{g}^i(\mathbf{x})\|^2 \leq G^2, \quad (3)$$

where G stands for the upper bound of the gradient norm.

Assumption 4 (μ-strongly convexity). We assume μ-strong convexity of $f_i, i \in [n]$, that is for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:

$$f(\mathbf{x}) - (\mathbf{y}) \geq \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2} \|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|^2. \quad (4)$$

For the convex cases In convex cases (Theorem 1), we apply Assumptions 1-4 and use $\Gamma_c = f^* - \sum_{i=1}^n p_i f_i^*$ to measure the data heterogeneity. f^* (as well as f_i^*) is the optimal value of f (f_i) referring to previous works [10], [17]. In non-convex cases, we apply Assumptions 1-3 and use $\Gamma_n \leq \mathbb{E}\|\nabla f_i(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2$ to quantize the non-IID degree of nodes referring to the work [32].

B. Convergence Rate of γ-FedHT

The technical challenge in proving this theorem lies in *integrating gradient compression algorithms with EF into the theoretical framework of FL* [4]. To tackle this problem, *it is crucial to establish relationship between $\boldsymbol{x}_{t+1}^i$ and $\boldsymbol{x}_t^i$*. When aggregation is not performed (*i.e.*, for $t \mod E \neq E-1$), we have $\mathbf{x}_{t+1}^i := \mathbf{x}_t^i - \gamma_t \mathbf{g}^i(\mathbf{x}_t)$, consistent with vanilla FedAVG. In aggregation rounds (*i.e.*, for $t \mod E = E-1$), we have:

$$\mathbf{x}_{t+1}^i + \sum_{i \in \mathcal{S}_{t+1}} \frac{1}{S} \mathbf{e}_{t+1}^i = \sum_{i \in \mathcal{S}_t} \frac{1}{S} [\mathbf{x}_t^i + \mathbf{e}_t^i - \gamma_t \mathbf{g}^i(\mathbf{x}_t^i)],$$

which can be derived by lines 13 and 19 in Algo. 1. By the above conversation equation and the virtual sequence method (also known as perturbed iterate analysis), we derive the convergence rate of γ-FedHT.

Theorem 1 (μ-strongly convex Convergence rate of γ-FedHT)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and μ-convex. Choose $\kappa = \frac{L}{\mu}$, $b = \max\{12\frac{L}{\mu}, E\} - 1$ and the stepsize $\gamma_t = \frac{3}{\mu(b+t)}$. Then γ-FedHT satisfies

$$\mathbb{E}[f(\mathbf{x}_T)] - f^* \leq \frac{\kappa}{t+b} \{ \frac{9}{\mu} [(1 + \frac{\gamma_0 \mu}{2}) B + (\frac{1}{2} + \frac{2(\gamma_0/\gamma_T)^{\alpha/2}}{\mu\gamma_0}) D] + \frac{\mu(1+b)}{2} \Delta_1 \}, \quad (5)$$

where $B = \sum_{i=1}^n p_i^2 \sigma^2 + 6L\Gamma + 8(E-1)^2 G^2 + \frac{4}{S} E^2 G^2$ and $D = 4d\lambda_0^2$.

Remark 1. The convergence rate of γ-FedHT is $\mathcal{O}(\frac{1}{T})$ under the μ-strongly convex cases, same as vanilla FedAVG [4].

Remark 2. The term $(\frac{1}{2} + \frac{2(\gamma_0/\gamma_T)^{\alpha/2}}{\mu\gamma_0}) D$ is the bound of the compression error, which is linearly correlated with λ_0^2. When $\lambda_0 = 0$, γ-FedHT degrades to vanilla FedAVG.

Remark 3. The larger the α, the more iterations are needed for the convergence. So *we take $\alpha = 1$ by default* for $\alpha \geq 1$ in the μ-strongly convex cases.

Theorem 2 (Non-Convex Convergence rate of γ-FedHT)**.** Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth. Choose $c > 0$, and $\gamma_t \leq \frac{1}{8LE}$. γ_t satisfies $\gamma_t EL \leq \frac{2S}{S-1}$ and $30nE^2\gamma_t^2 L^2 \sum_{i=1}^n p_i^2 + \frac{2L\gamma_t}{S}(90E^3L^2\gamma_t^2 + 3E) < 1$. The convergence rate of γ-FedHT satisfies

$$\min_{t \in [T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{f_0 - f^*}{c\gamma_{T-1}TE} + (LE^2\sigma^2 + \frac{3E^2L\Gamma_n}{S}) \frac{\sum_{t=0}^{T-1} \gamma_t^2}{c\gamma_{T-1}TE} + \frac{2Ld\lambda_0^2}{\gamma_0} (\frac{\gamma_0}{\gamma_{T-1}})^{\frac{\alpha}{2}} \frac{\sum_{t=0}^{T-1} \gamma_t^3}{c\gamma_{T-1}TE} + \frac{\sigma^2 + 6E\Gamma_n}{c\gamma_{T-1}TE} [\frac{5nE^2L^2 \sum_{i=1}^n p_i^2 \sum_{t=0}^{T-1} \gamma_t^3}{2} + \frac{15E^3L^3 \sum_{t=0}^{T-1} \gamma_t^4}{S}].$$

Remark 4. The convergence rate of γ-FedHT is $\mathcal{O}(\frac{1}{\sqrt{T}})$ under non-convex cases, also same as FedAVG [32].

Remark 5. The term $\frac{LD}{2\gamma_0}(\frac{\gamma_0}{\gamma_{T-1}})^{\alpha/2}\frac{\sum_{t=0}^{T-1}\gamma_t^3}{c\gamma_{T-1}TE}$ represents the compression error and is bounded by λ_0^2. When $\lambda_0 = 0$, the convergence of γ-FedHT degrades to that of FedAVG.

Remark 6. Due to $\frac{\gamma_0}{\gamma_{T-1}} > 1$, γ-FedHT converges the fastest when $\alpha = 1$. *We take $\alpha = 1$ by default in the non-convex cases too.*

C. Proof of Theorem 1

Let $\mathcal{I}_E$ be the set of communication iterations. We have $\mathcal{I}_E = \{iE | i = 1, 2, \dots\}$. Here we introduce a variable $\mathbf{v}_{t+1}^i$ to represent the result of local SGD from $\mathbf{x}_{t+1}^i$. Then the training of Algo. 1 can be described as

$$\mathbf{v}_{t+1}^i = \mathbf{x}_t^i - \gamma_t \mathbf{g}^i(\mathbf{x}_t^i), \tag{6}$$

$$\mathbf{x}_{t+1}^i = \begin{cases} \mathbf{v}_{t+1}^i & \text{if } t+1 \notin \mathcal{I}_E, \\ \sum_{i\in\mathcal{S}_t}\frac{1}{S}(\mathbf{v}_{t+1}^i + \mathbf{e}_t^i) - \sum_{i\in\mathcal{S}_{t+1}}\frac{1}{S}\mathbf{e}_{t+1}^i & \text{if } t+1 \in \mathcal{I}_E. \end{cases} \tag{7}$$

Motivated by previous works, we introduce two virtual sequences $\bar{\mathbf{v}}_t = \sum_{i=1}^n p_i \mathbf{v}_t^i$ and $\bar{\mathbf{x}}_t = \sum_{i=1}^n p_i \mathbf{x}_t^i$. We use the notations $\mathbf{E}_t = \mathbb{E}\|\frac{1}{S}\sum_{i\in\mathcal{S}_t}\mathbf{e}_{t+1}^i\|^2$ and $\Delta_t = \mathbb{E}\|\bar{\mathbf{x}}_t - \mathbf{x}^*\|^2$.

According to Lemma 1-5 from the work [4], we have:

Lemma 1. Following the Assumption 1-4. If $\gamma_t \le \frac{1}{4L}$, γ_t is non-increasing and $\gamma_t \le 2\gamma_{t+E}$ for all $t \ge 0$, we have

$$\mathbb{E}\|\bar{\mathbf{v}}_{t+1} - \mathbf{x}^*\|^2 \le (1-\gamma_t\mu)\Delta_t + \gamma_t^2 B, \tag{8}$$

where $B = \sum_{i=1}^n p_i^2\sigma^2 + 6L\Gamma + 8(E-1)^2G^2 + \frac{4}{S}E^2G^2$.

Before the proof, we compare Eq. 6 and Eq. 7 with the Sec. A-3 in [4]. We can find that $\bar{\mathbf{v}}_{t+1} = \bar{\mathbf{x}}_t - \gamma_t\mathbf{g}_t$ no matter $t+1 \notin \mathcal{I}_E$ or $t+1 \in \mathcal{I}_E$. Then we categorize and discuss two cases (*i.e.*, $t+1 \notin \mathcal{I}_E$ and $t+1 \in \mathcal{I}_E$) because Eq. 7 is different from [4].

• $t+1 \notin \mathcal{I}_E$: Due to $\bar{\mathbf{v}}_{t+1} = \bar{\mathbf{x}}_{t+1}$ and Lemma 1, we have

$$\Delta_{t+1} \le (1-\gamma_t\mu)\Delta_t + \gamma_t^2 B. \tag{9}$$

• $t+1 \in \mathcal{I}_E$: Due to $\bar{\mathbf{x}}_{t+1} = \bar{\mathbf{v}}_{t+1} - \frac{1}{S}\sum_{i\in\mathcal{S}_t}\mathbf{e}_{t+1}^i$, we have

$$\begin{aligned}\mathbb{E}\|\bar{\mathbf{x}}_{t+1} - \mathbf{x}^*\|^2 &\le (1-\frac{\gamma_t\mu}{2})\Delta_t + (1+\frac{\gamma_t\mu}{2})\gamma_t^2 B \\ &\quad +2(1+\frac{2}{\gamma_t\mu})(\mathbf{E}_{t+1} + \mathbf{E}_t).\end{aligned} \tag{10}$$

The inequality is followed by Jensen inequality and Lemma 1.

We next focus on $\mathbf{E}_t$ and have

$$\mathbf{E}_t \le \frac{1}{S}\sum_{i\in\mathcal{S}_t}\|\mathbf{e}_t^i\|^2 \le \gamma_t^2 d\lambda_t^2 \le \gamma_t^2 d\lambda_0^2 F(\gamma_{t-1}), \tag{11}$$

where the first inequality is due to $\|\sum_{i=1}^k a_i\|^2 \le k\sum_{i=1}^k\|a_i\|^2$. The second inequality follows the property of the absolute compressor.

We combine Eq. 10 and Eq. 11 and have

$$\begin{aligned}&\mathbb{E}\|\bar{\mathbf{x}}_{t+1} - \mathbf{x}^*\|^2 \le (1-\frac{\gamma_t\mu}{2})\Delta_t + (1+\frac{\gamma_t\mu}{2})\gamma_t^2 B \\ &+(1+\frac{2}{\gamma_t\mu})\gamma_{t+1}^2 d\lambda_0^2(F(\gamma_t) + F(\gamma_{t+1})) \\ &\le (1-\frac{\gamma_t\mu}{2})\Delta_t + (1+\frac{\gamma_t\mu}{2})\gamma_t^2 B + [(1+\frac{2}{\gamma_t\mu})\gamma_t^2 F(\gamma_t) \\ &+(1+\frac{2}{\gamma_{t+1}\mu})\gamma_{t+1}^2 F(\gamma_{t+1})]D,\end{aligned} \tag{12}$$

where $B = \sum_{i=1}^n p_i^2\sigma^2 + 6L\Gamma + 8(E-1)^2G^2 + \frac{4}{S}E^2G^2$ and $D = 2d\lambda_0^2$.

The second inequality is due to the non-increasing γ_t.

Since the RHS of Eq. 9 is smaller than the RHS of Eq. 12, the RHS of Eq. 12 applies for all $t \le 0$. We have

$$\begin{aligned}\Delta_{t+1} &\le (1-\frac{\gamma_t\mu}{2})\Delta_t + (1+\frac{\gamma_t\mu}{2})\gamma_t^2 B \\ &\quad + \left[(1+\frac{2}{\gamma_t\mu})\gamma_t^2 F(\gamma_t) + (1+\frac{2}{\gamma_{t+1}\mu})\gamma_{t+1}^2 F(\gamma_{t+1})\right] D.\end{aligned} \tag{13}$$

For a decaying learning rate, $\gamma_t = \frac{\beta}{t+b}$ for some $\beta > \frac{2}{\mu}$ and $b > 0$ such that $\gamma_0 \le \frac{1}{4L}$ and $\gamma_t \le 2\gamma_{t+E}$. We utilize the mathematical induction to prove $\Delta_t \le \frac{v}{b+t}$, where $v = \max\{\frac{2\beta^2}{\beta\mu-2}[(1+\frac{\gamma_0\mu}{2})B + (\frac{1}{2} + \frac{2\gamma_0^{\alpha-1}}{\mu(\gamma_0\gamma_T)^{\alpha/2}})4d\lambda_0^2], (1+b)\Delta_1\}$.

When $t = 1$, $\Delta_t \le \frac{v}{b+t}$ clearly holds.

When $t > 1$, it follows that

$$\begin{aligned}\Delta_{t+1} &\le (1-\frac{\gamma_t\mu}{2})\frac{v}{t+b} + (1+\frac{\gamma_0\mu}{2})\gamma_t^2 B \\ &\quad + \left[(1+\frac{2}{\gamma_t\mu})\gamma_t^2 F(\gamma_t) + (1+\frac{2}{\gamma_{t+1}\mu})\gamma_{t+1}^2 F(\gamma_{t+1})\right] D \\ &\le \frac{v}{t+b+1} + \left[(1+\frac{\gamma_0\mu}{2})\gamma_t^2 B - \frac{\beta\mu-2}{2(t+b)^2}v\right] \\ &\quad + \underbrace{\left[(1+\frac{2}{\gamma_t\mu})\gamma_t^2 F(\gamma_t) + (1+\frac{2}{\gamma_{t+1}\mu})\gamma_{t+1}^2 F(\gamma_{t+1})\right] D}_{A_1}.\end{aligned} \tag{14}$$

The first inequality holds by the inductive conclusion $\Delta_t \le \frac{v}{b+t}$ and $\gamma_t \le \gamma_0$.

We next aim to bound A_1. According to γ-FedHT, we have

$$\begin{aligned}A_1 &\le D(\frac{\gamma_t^2}{2} + \frac{2\gamma_t^{\alpha+1}}{\mu(\gamma_0\gamma_T)^{\alpha/2}}) + D(\frac{\gamma_{t+1}^2}{2} + \frac{2\gamma_{t+1}^{\alpha+1}}{\mu\sqrt{\gamma_0\gamma_T}^{\alpha}}) \\ &\le 2D(\frac{\gamma_t^2}{2} + \frac{2\gamma_t^2\gamma_0^{\alpha-1}}{\mu(\gamma_0\gamma_T)^{\alpha/2}}).\end{aligned} \tag{15}$$

The first inequality is due to the arithmetic-geometric mean inequality (the first part) and $\gamma_t > 0$ (the second part). The second inequality is due to the decaying-γ.

Combining Eq. 15 and Eq. 14, we have

$$\begin{aligned}\Delta_{t+1} &\le \frac{v}{t+b+1} + \gamma_t^2\{\frac{2\beta^2}{\beta\mu-2}[(1+\frac{\gamma_0\mu}{2})B \\ &\quad +(\frac{1}{2} + \frac{2\gamma_0^{\alpha-1}}{\mu(\gamma_0\gamma_T)^{\alpha/2}})D'] - v\} \le \frac{v}{t+b+1},\end{aligned}$$

which completes the proof of $\Delta \le \frac{v}{t+b}$ and $D' = 2D = 4d\lambda_0^2$.

According to the L-smoothness of $f(\cdot)$,

$$\mathbb{E}[f(\mathbf{x}_T)] - f^* \leq \frac{L}{2}\Delta_t \leq \frac{L}{2}\frac{v}{t+b}.$$

We let $\beta = \frac{3}{\mu}$, $b = \max\{12\kappa, E\} - 1$ ($\kappa = \frac{L}{\mu}$) and have $v \leq \frac{18}{\mu^2}[(1+\frac{\gamma_0\mu}{2})B + (\frac{1}{2} + \frac{2\gamma_0^{\alpha-1}}{\mu(\gamma_0\gamma_T)^{\alpha/2}})D'] + (1+b)\Delta_1$.

D. Proof of Theorem 2

We let $t' = \lfloor \frac{t}{E} \rfloor$, $\Delta^i_{t'} = \sum_{j=0}^{E-1} \nabla f_i(\mathbf{x}^i_{t'+j})$. t' represents the communication iteration, and $\Delta^i_{t'}$ represents the gradients accumulated between the t'-th and the $t'+1$-th global iterations at node i.

We define a virtual sequence:

$$\bar{\mathbf{x}}_0 = \mathbf{x}_0, \quad \bar{\mathbf{x}}_{t'+1} := \bar{\mathbf{x}}_{t'} - \frac{\gamma_{t'}}{S}\sum_{i\in\mathcal{S}_t}\Delta^i_{t'}.$$

The error term that represents the deviation between the virtual sequence and the actual sequence is

$$\bar{\mathbf{x}}_{t'} - \mathbf{x}_{t'} = \frac{\gamma_{t'}}{S}\sum_{i=1}^{n}\mathbf{e}^i_{t'}.$$

Given L-smoothness of f, we have

$$\begin{aligned}\mathbb{E}f(\bar{\mathbf{x}}_{t'+1}) \leq f(\bar{\mathbf{x}}_{t'}) &- < \nabla f(\mathbf{x}_{t'}), \frac{\gamma_{t'}}{S}\sum_{i\in\mathcal{S}_t}\Delta^i_{t'} > \\ &+ \underbrace{< \nabla f(\mathbf{x}_{t'}) - \nabla f(\bar{\mathbf{x}}_{t'}), \frac{\gamma_{t'}}{S}\sum_{i\in\mathcal{S}_t}\Delta^i_{t'} >}_{A_2} + \frac{L}{2}\mathbb{E}\|\frac{\gamma_{t'}}{S}\sum_{i=1}^{n}\Delta^i_{t'}\|^2.\end{aligned} \tag{16}$$

We next aim to bound A_2, where

$$\begin{aligned}&< \nabla f(\mathbf{x}_{t'}) - \nabla f(\bar{\mathbf{x}}_{t'}), \frac{\gamma_{t'}}{S}\sum_{i\in\mathcal{S}_t}\Delta^i_{t'} > \\ &\leq \frac{1}{2L}\mathbb{E}\|f(\mathbf{x}_{t'}) - \nabla f(\bar{\mathbf{x}}_{t'})\|^2 + \frac{L}{2}\mathbb{E}\|\frac{\gamma_{t'}}{S}\sum_{i=1}^{n}\Delta^i_{t'}\|^2 \\ &\leq \frac{L}{2}\mathbf{E}_{t'} + \frac{L}{2}\mathbb{E}\|\frac{\gamma_{t'}}{S}\sum_{i=1}^{n}\Delta^i_{t'}\|^2.\end{aligned} \tag{17}$$

The first inequality is followed by Jensen inequality, and the last inequality is held by L-smooth functions.

According to the Appendix B in the work [17], Eq. 11 and 17, we can convert Eq. 16 into:

$$\begin{aligned}&\mathbb{E}f(\bar{\mathbf{x}}_{t'+1}) \leq f(\bar{\mathbf{x}}_{t'}) - \gamma_{t'}E\|\nabla f(\mathbf{x}_{t'})\|^2 \\ &[\frac{1}{2} - 15nE^2\gamma^2_{t'}L^2\sum_{i=1}^{n}p_i^2 - \frac{L\gamma_{t'}}{S}(90E^3L^2\gamma^2_{t'} + 3E)] \\ &+(\frac{5nE^2\gamma^3_{t'}L^2\sum_{i=1}^{n}p_i^2}{2} + \frac{15E^3L^3\gamma^4_{t'}}{S})(\sigma^2 + 6E\Gamma_n) \\ &+LE^2\gamma^2_{t'}\sigma^2 + \frac{3E^2L\gamma^2_{t'}\Gamma_n}{S} + \frac{LD}{2\gamma_0}(\frac{\gamma_0}{\gamma_{T-1}})^{\alpha/2}\gamma^3_{t'} \\ &+(\frac{L\gamma^2_{t'}(S-1)}{S} - \frac{\gamma_{t'}}{2E})\mathbb{E}\|\sum_{i=1}^{n}p_i\sum_{j=0}^{E-1}\nabla f_i(\mathbf{x}^i_{t'+j})\|^2 \\ &\leq f(\bar{\mathbf{x}}_{t'}) - c\gamma_{t'}E\|\nabla f(\mathbf{x}_{t'})\|^2 \\ &+(\frac{5nE^2\gamma^3_{t'}L^2\sum_{i=1}^{n}p_i^2}{2} + \frac{15E^3L^3\gamma^4_{t'}}{S})(\sigma^2 + 6E\Gamma_n) \\ &+LE^2\gamma^2_{t'}\sigma^2 + \frac{3E^2L\gamma^2_{t'}\Gamma_n}{S} + \frac{LD}{2\gamma_0}(\frac{\gamma_0}{\gamma_{T-1}})^{\alpha/2}\gamma^3_{t'}.\end{aligned}$$

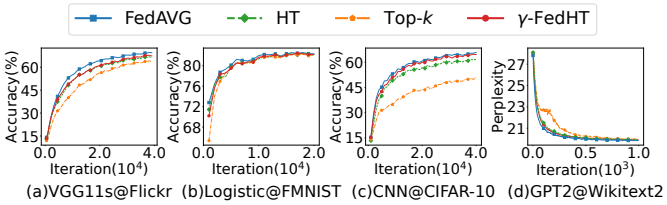

Fig. 3: Training curves (Accuracy vs. Iterations). The artificially non-IID partition strategy is $\#C = 2$. On all benchmarks, γ-FedHT outperforms hard-threshold compression (HT) and Top-k.

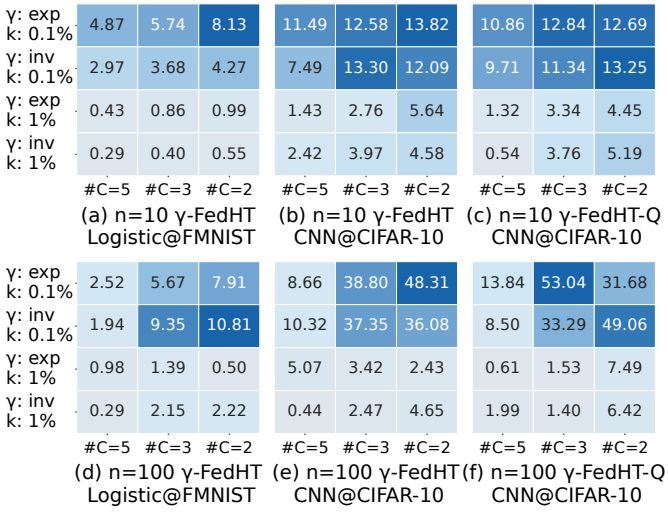

Fig. 4: Heatmaps of the accuracy difference (%) for different types of stepsizes (denoted as γ), compression ratios (k), worker size (n) on different tasks. The accuracy difference in (a, b, d, e) is the final accuracy of γ-FedHT minus that of Top-k, and the difference in (c, f) is γ-FedHT-Q minus STC. In all combinations, γ-FedHT (as well as γ-FedHT-Q) is superior to Top-k (STC).

The last inequality follows from $\frac{L\gamma^2_{t'}(S-1)}{S} - \frac{\gamma_{t'}}{2E} \leq 0$ if $\gamma_{t'}EL \leq \frac{S}{2}$ and $\frac{1}{2} - 15nE^2\gamma^2_{t'}L^2\sum_{i=1}^{n}p_i^2 - \frac{L\gamma_{t'}}{S}(90E^3L^2\gamma^2_{t'} + 3E) > c > 0$ if $15nE^2\gamma^2_{t'}L^2\sum_{i=1}^{n}p_i^2 + \frac{L\gamma_{t'}}{S}(90E^3L^2\gamma^2_{t'} + 3E) < \frac{1}{2}$. We complete the proof.

VI. Evaluation Experiments

A. Experimental Settings

Experiment tasks: We conduct experiments on the four tasks and the detailed setting is shown in Table I.

Non-IID partition strategy: For the artificially non-IID partition, we adopt 3 distinct non-IID partition strategies, namely $\#C = 2$, $\#C = 3$, $\#C = 5$. For Flickr, we divide workers according to the subcontinent they belong to, with 15 clients in total.

Baselines: We compare (1) γ-FedHT with Top-k, the hard-threshold compressor with fixed-λ (denoted as HT), and vanilla FedAVG; (2) γ-FedHT-Q (γ-FedHT followed by the quantizer used in STC) with STC. Top-k is the SOTA sparsification gradient compressor in FL and serves as a key component of nowadays hybrid gradient compressors [10]–[12].

TABLE I: Summary of the experiment settings used in this work.

Task	Model	Model parameters	Dataset	Non-IID type	Loss convexity	Batch size	n	Metric	Iterations
CV	VGG11s [33]	$865,482$	Flickr [5]	Real-world	Non-convex loss	8	15	Accuracy	$40,000$
	Logistic [28]	$10,250$	FMNIST [26]	Artificially	Non-convex loss	50	10 & 100		$20,000$
	CNN [12]	$235,690$	CIFAR-10 [27]	Artificially	Convex loss	8	10 & 100		$40,000$
NLP	GPT2 [16]	$124,000,000$	Wikitext2 [34]	Artificially	Non-convex loss	1	10	Perplexity	$1,000$

TABLE II: Values of compression-related hyperparameters.

Model@Dataset	Top-k k	HT λ	γ-FedHT λ_0 (inv γ)	γ-FedHT λ_0 (exp γ)
VGG11s@Flickr	0.1%	1.70×10^{-2}	3.35×10^{-2}	6.28×10^{-2}
Logistic@FMNIST	1%	4.94×10^{-2}	8.70×10^{-2}	9.41×10^{-2}
CNN@CIFAR-10	0.1%	3.26×10^{-2}	6.42×10^{-2}	1.21×10^{-1}
GPT2@Wikitext2	0.1%	1.42×10^{-3}	2.29×10^{-3}	9.02×10^{-3}

HT is the SOTA sparsifier in traditional DML [9]. FedAVG without compression is used as the benchmark for evaluation. STC is the SOTA hybrid compressor in FL [12]. *We take $\alpha = 1$ in γ-FedHT due to Remark 3 and Remark 6.*

Hyperparameters: For each communication round, we randomly select half of the clients to participate. We configure the communication frequency $E = 5$, the inverse-proportional decay stepsize $\gamma_t = \frac{100}{t+1000}$ and exponential decay stepsize $\gamma_t = 0.1 \times 0.999^{t/E}$ for $n = 10$, 15. For $n = 100$, we reduce γ_t by a multiple of 10.

B. Comparison of Model Accuracy

Our experimental results show that the training results of γ-FedHT outperform HT and Top-k on all tasks. The compression-related parameters are shown in Table II. We calculate λ and λ_0 referring to Appendix D of the work[6] [9].

In Fig. 3, we find that γ-FedHT always converges better than other sparsifiers. Let us take CNN@CIFAR-10 as an example. To converge to 50%, 55%, 60% accuracy, γ-FedHT is faster than HT by about 3.75%, 4.25%, 8.15% iterations, and the accuracy of Top-k cannot reach 60%. Vanilla HT and Top-k introduce severe accuracy degradation in this case, but γ-FedHT does not.

In Fig. 4, we conduct a sensitivity analysis for both Logistic@FMNIST and CNN@CIFAR-10 on five factors. Below is a detailed discussion of these factors.

Compression ratio and non-IID partition strategy: The more aggressive the compression (k from 1% to 0.1%) and the more severe the non-IID problem (from $\#C = 5$ to $\#C = 2$), the larger the accuracy difference is. This shows that *γ-FedHT greatly alleviates accuracy degradation when faced with extremely aggressive compression and severe non-IID problems*.

Worker size: The variance of the accuracy differences at $n = 100$ is larger than that at $n = 10$. This is interesting because it illustrates that a larger worker size can have two effects at the same time. Firstly, when confronted with accuracy degradation, a larger n tends to enlarge this degradation. This can be seen from outliers ($> 30\%$) in (e, f), which indicate that Top-k does not converge. In contrast, *γ-FedHT is surprisingly robust to large-scale FL training*. Secondly, a larger n also accelerates the model convergence (by training more data in one iteration), thus reducing the accuracy differences.

[6]We use $\lambda = \frac{1}{2\sqrt{dk}}$ and $\int_0^T \frac{1}{\lambda^2} dt = \int_0^T \frac{1}{\lambda_t^2} dt$ to work out λ, λ_0 respectively.

TABLE III: Accuracy and communication traffic of different gradient compression algorithms under different non-IID partition strategies. The results show that γ-FedHT outperforms other sparsifiers under both non-convex and convex cases especially when the communication is restricted and the non-IID problem is extremely severe.

Model @Dataset	Non-IID Partition Strategy	Method	Accuracy	Comm. Traffic	Comm. Traffic Reduced to
Logistic @FMNIST	$\#C = 2$	Top-(k_{mean})	81.97%	3.44MB	2.20%
		HT	81.99%	3.78MB	2.42%
		γ-FedHT	**82.23%**	**3.44MB**	**2.20%**
		FedAVG	82.34%	156.40MB	100%
	$\#C = 3$	Top-(k_{mean})	82.84%	3.19MB	2.04%
		HT	82.82%	3.50MB	2.24%
		γ-FedHT	**83.05%**	**3.19MB**	**2.04%**
		FedAVG	83.11%	156.4MB	100%
	$\#C = 5$	**Top-(k_{mean})**	**83.56%**	**2.56MB**	**1.64%**
		HT	83.43%	2.82MB	1.80%
		γ-FedHT	83.51%	2.56MB	1.64%
		FedAVG	83.57%	156.40 MB	100%
CNN @CIFAR-10	$\#C = 2$	Top-(k_{mean})	57.57%	23.74MB	0.33%
		HT	61.56%	21.58MB	0.30%
		γ-FedHT	**64.51%**	**23.74MB**	**0.33%**
		FedAVG	65.75%	7192.69MB	100%
	$\#C = 3$	Top-(k_{mean})	64.88%	25.89MB	0.36%
		HT	71.12%	30.21MB	0.42%
		γ-FedHT	**72.30%**	**25.89MB**	**0.36%**
		FedAVG	72.42%	7192.69MB	100%
	$\#C = 5$	Top-(k_{mean})	70.33%	20.14MB	0.28%
		HT	73.35%	23.74MB	0.33%
		γ-FedHT	**74.58%**	**20.14MB**	**0.28%**
		FedAVG	75.36%	7192.69MB	100%

Decaying type of the stepsize and whether to bring the quantizer or not: Carrying a quantizer essentially makes the compression more aggressive, whereas γ-FedHT is robust to aggressive compression environments, so the difference is further enlarged in γ-FedHT-Q in (c, f). The decaying type of γ does not affect the excellent compression-accuracy trade-off of γ-FedHT.

C. Comparison of Communication Traffic

Experimental results show that γ-FedHT performs better than HT and Top-k under equal communication traffic. In this part, we focus on two metrics, accuracy and communication traffic. We compare γ-FedHT with HT and Top-k (under the same total communication traffic) in Table III. We compare γ-FedHT-Q with STC in Table IV. A detailed discussion follows.

γ-FedHT with vanilla HT: γ-FedHT achieves higher accuracy with less communication traffic than HT under nearly all

TABLE IV: Accuracy and communication traffic of STC and γ-FedHT-Q under different non-IID partition strategies. The results show that γ-FedHT-Q outperforms STC under both non-convex and convex cases.

Model @Dataset	Non-IID Partition Strategy	Method	Accuracy	Comm. Traffic	Comm. Traffic Reduced to
Logistic @FMNIST	$\#C=2$	STC	81.75%	0.21MB	0.14%
		γ-FedHT-Q	**82.19%**		
	$\#C=3$	STC	82.73%	0.20MB	0.13%
		γ-FedHT-Q	**82.92%**		
	$\#C=5$	STC	83.36%	0.16MB	0.10%
		γ-FedHT-Q	**83.46%**		
CNN @CIFAR-10	$\#C=2$	STC	58.83%	1.42MB	0.020%
		γ-FedHT-Q	**63.90%**		
	$\#C=3$	STC	64.59%	1.56MB	0.022%
		γ-FedHT-Q	**71.26%**		
	$\#C=5$	STC	70.19%	1.22MB	0.017%
		γ-FedHT-Q	**74.10%**		

cases. This shows that the adaptive mechanism of our design effectively optimizes the training process of HT and avoids the waste of communication traffic.

γ-FedHT with Top-k: In Logistic@FMNIST, γ-FedHT exhibits a higher accuracy by 0.26% compared to Top-k under $\#C = 2$. In CNN@CIFAR-10, this accuracy difference expands to 7.42% under $\#C = 3$. This suggests that *γ-FedHT can achieve better communication-accuracy trade-off than the SOTA sparsifier in FL, especially under non-convex and communication-constrained cases*.

γ-FedHT-Q with STC: Similar to Fig. 4, the performance of γ-FedHT is not affected whether it carries a quantizer. Even with an extremely aggressive compression strategy, γ-FedHT does not introduce serious accuracy degradation, which validates the conclusion that γ-FedHT converges at the same rate as FedAVG.

VII. Related Works

FedAVG with gradient sparsification: Research in this area has achieved impressive compression ratios as low as 1% or less. However, many studies lack theoretical analysis, and the computational complexity cannot achieve $\mathcal{O}(d)$. One such study proposes STC [12], which manages to achieve a compression ratio of nearly 0.1% without significant accuracy degradation. This is accomplished by implementing the downstream compression and encoding on Top-k combined with ternary quantization. Other studies with similar approaches include FedZIP [11] and B-MUSTC [10]. The work [35] jointly considers adaptive node selection and sparsification compression, but does not derive the number of iterations required to converge to a specified error. The work [36] analyzes the convergence rate of FedAVG when using the sparsification compression, which is $\mathcal{O}(\frac{1}{\sqrt{T}})$ (for both the convex and non-convex scenarios), slower than vanilla FedAVG. γ-FedHT, however, can achieve the same asymptotic convergence rate as FedAVG and keep the low-cost feature.

Low-cost compression in FedAVG: Most works only use the quantization compression to keep the time complexity of $\mathcal{O}(d)$ or even less. They provide the convergence analysis, but typically only achieve a compression ratio of nearly 10%. One such work proposes [10] MUSTC, an unbiased version of B-MUSTC which converges at the rate of $\mathcal{O}(\frac{1}{T})$ under convex scenarios. Similar works combine the quantization compression with mechanisms such as periodic aggregation [37], downstream compression [38] and local gradient tracking [39]. The work [17] proposes an adaptation framework for robust dynamic networks by strategically adjusting the compression ratio, but not considers EF. The work [40] proposes Cepe-FL, propose a two-way adaptive compressive sensing scheme in FL and reduce the computational complexity from $\mathcal{O}(n)$ to $\mathcal{O}(1)$, but not guarantee the model convergence. The work [41] reduces the computational cost of Top-k by compressing the indexes of compressed parameters and proposes FedComp, but still has the GPU-unfriendly operation.

Theoretical analysis of sparsification compression in non-IID scenarios: Most works ignore the node selection, infrequent communication, and the decaying learning rate, thus not applicable for FL. The work [8] compares distributed quantized SGD with unbiased quantizers and distributed SGD with Error-Feedback and biased compressors in non-IID scenarios. The work [28] proposes DAGC, which assigns compression ratios according to the training weight. The work [24] proposes EF21, which refines the traditional error-feedback mechanism. The work [42] introduces a compression-based FL algorithm equipped with EF and achieves the same convergence rate as vanilla FedAVG in the non-convex cases, but lacks the analysis in the convex cases and does not consider the absolute compressor. Our theoretical analysis considers both the infrequent communication and the partial node participation, making it suitable for FL.

VIII. Conclusion

In this paper, we propose an ideal sparsifier for FL with a time complexity of $\mathcal{O}(d)$, named γ-FedHT. We first reveal that the hard-threshold compressor induces accuracy degradation in FL and the decaying-γ in non-IID scenarios leads to the failure of this compressor in FL. Then, we propose γ-FedHT, a stepsize-aware low-cost hard-threshold compressor in FL, with the time complexity of $\mathcal{O}(d)$ and the same convergence rate as FedAVG. Experimental results show that γ-FedHT can improve accuracy by up to 7.42% over Top-k under the equal communication amount in non-IID scenarios. γ-FedHT is expected to replace Top-k as the SOTA sparsifier in FL due to its excellent performance.

Acknowledgment

We would like to thank Chen Tang, Mingzhou Wu and Shuzhao Xie for their help in making this work possible. This work is supported in part by National Key Research and Development Project of China under Grant 2023YFF0905502, National Natural Science Foundation of China under Grant 62472249, and Shenzhen Science and Technology Program under Grant JCYJ20220818101014030. The work of Bin Chen is supported by the National Natural Science Foundation of China under Grant 62301189.

REFERENCES

[1] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–44, 2023.

[2] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1273–1282.

[4] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2019.

[5] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *International Conference on Machine Learning*, PMLR, 2020, pp. 4387–4398.

[6] R. Dorfman, S. Vargaftik, Y. Ben-Itzhak, and K. Y. Levy, "Docofl: Downlink compression for cross-device federated learning," in *International Conference on Machine Learning*, PMLR, 2023, pp. 8356–8388.

[7] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed updates," *Journal of Machine Learning Research*, vol. 21, pp. 1–36, 2020.

[8] S. U. Stich, "On communication compression for distributed optimization on heterogeneous data," *arXiv preprint arXiv:2009.02388*, 2020.

[9] A. Sahu, A. Dutta, A. M Abdelmoniem, T. Banerjee, M. Canini, and P. Kalnis, "Rethinking gradient sparsification as total error minimization," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[10] L. Cui, X. Su, Y. Zhou, and Y. Pan, "Slashing communication traffic in federated learning by transmitting clustered model updates," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2572–2589, 2021.

[11] A. Malekijoo, M. J. Fadaeieslam, H. Malekijou, M. Homayounfar, F. Alizadeh-Shabdiz, and R. Rawassizadeh, "Fedzip: A compression framework for communication-efficient federated learning," *arXiv preprint arXiv:2102.01593*, 2021.

[12] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[13] S. Oimoen, "Classical designs: Full factorial designs," *STAT Center of Excellence: Dayton, OH, USA*, 2019.

[14] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *arXiv preprint arXiv:1704.05021*, 2017.

[15] A. M Abdelmoniem, A. Elzanaty, M.-S. Alouini, and M. Canini, "An efficient statistical-based gradient compression technique for distributed training systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 297–322, 2021.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[17] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, IEEE, 2022, pp. 1459–1468.

[18] L. Zhu, H. Lin, Y. Lu, Y. Lin, and S. Han, "Delayed gradient averaging: Tolerate the communication latency for federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 995–30 007, 2021.

[19] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized sgd and its applications to large-scale distributed optimization," in *International Conference on Machine Learning*, PMLR, 2018, pp. 5325–5333.

[20] 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*, vol. 31, 2018.

[21] 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*, Singapore, vol. 2014, 2014, pp. 1058–1062.

[22] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized sgd and its applications to large-scale distributed optimization," in *International conference on machine learning*, PMLR, 2018, pp. 5325–5333.

[23] L. Nguyen, P. H. Nguyen, M. Dijk, P. Richtárik, K. Scheinberg, and M. Takác, "Sgd and hogwild! convergence without the bounded gradients assumption," in *International Conference on Machine Learning*, PMLR, 2018, pp. 3750–3758.

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

[25] Y. Gao, R. Islamov, and S. U. Stich, "Econtrol: Fast distributed optimization with compression and error control," in *The Twelfth International Conference on Learning Representations*, 2024.

[26] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

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

[28] R. Lu, J. Song, B. Chen, L. Cui, and Z. Wang, "Dagc: Data-aware adaptive gradient compression," in *INFOCOM*, 2023.

[29] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *IEEE International Conference on Data Engineering*, 2022.

[30] S. Agarwal, H. Wang, K. Lee, S. Venkataraman, and D. Papailiopoulos, "Adaptive gradient communication via critical learning regime identification," *Machine Learning and Systems*, vol. 3, pp. 55–80, 2021.

[31] A. Koloskova, S. U. Stich, and M. Jaggi, "Sharper convergence guarantees for asynchronous sgd for distributed and federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 202–17 215, 2022.

[32] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," in *International Conference on Learning Representations*, 2021.

[33] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *International Joint Conference on Neural Networks*, IEEE, 2019, pp. 1–8.

[34] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *International Conference on Learning Representations*, 2022.

[35] Z. Jiang, Y. Xu, H. Xu, Z. Wang, and C. Qian, "Heterogeneity-aware federated learning with adaptive client selection and gradient compression," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, IEEE, 2023, pp. 1–10.

[36] X. Li and P. Li, "Analysis of error feedback in compressed federated non-convex optimization," 2022.

[37] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 2021–2031.

[38] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Federated Learning With Quantized Global Model Updates," *arXiv e-prints*, arXiv:2006.10672, arXiv:2006.10672, Jun. 2020. arXiv: 2006.10672 [cs.IT].

[39] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 2350–2358.

[40] Y. Liu, S. Chang, and Y. Liu, "Cepe-fl: Communication-efficient and privacy-enhanced federated learning via adaptive compressive sensing," *IEEE Transactions on Big Data*, 2024.

[41] D. Wu, W. Yang, H. Jin, X. Zou, W. Xia, and B. Fang, "Fedcomp: A federated learning compression framework for resource-constrained edge computing devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 230–243, 2024. DOI: 10.1109/TCAD.2023.3307459.

[42] H. Yang, J. Liu, and E. S. Bentley, "Cfedavg: Achieving efficient communication and fast convergence in non-iid federated learning," in *19th WiOpt*, 2021. DOI: 10.23919/WiOpt52861.2021.9589061.