

Communication-Efficient Federated Learning via Sketching with Sharp Rates

Anonymous Author(s)*

ABSTRACT

Communication complexity and data privacy are the two key challenges in Federated Learning (FL) where the goal is to perform a distributed learning through a large volume of devices. In this work, we introduce two new algorithms, namely FedSKETCH and FedSKETCHGATE, to address jointly both challenges and which are, respectively, intended to be used for homogeneous and heterogeneous data distribution settings. Our algorithms are based on a key and novel sketching technique, called HEAPRIX (HP) that is unbiased, compresses the accumulation of local gradients using count sketch, and exhibits communication-efficiency properties leveraging low-dimensional sketches. We provide sharp convergence guarantees of our algorithms and validate our theoretical findings with various sets of experiments.

ACM Reference Format:

Anonymous Author(s). 2022. Communication-Efficient Federated Learning via Sketching with Sharp Rates. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (SIGKDD'22)*. ACM, New York, NY, USA, 19 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Federated Learning (FL) is an emerging framework for distributed large scale machine learning problems. In FL, data is distributed across devices [23, 33] and users are only allowed to communicate with the parameter server. Formally, the optimization problem across p distributed devices is defined as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^d, \sum_{j=1}^p q_j = 1} f(\mathbf{x}) \triangleq \sum_{j=1}^p q_j f_j(\mathbf{x}), \quad (1)$$

where for device $j \in \{1, \dots, p\}$, $f_j(\mathbf{x}) = \mathbb{E}_{\xi \in \mathcal{D}_j} [L_j(\mathbf{x}, \xi)]$, L_j is a loss function that measures the performance of model \mathbf{x} , ξ is a random variable distributed according to probability distribution \mathcal{D}_j , $q_j \triangleq \frac{n_j}{n}$ indicates the portion of data samples, n_j is the number of data shards and $n = \sum_{j=1}^p n_j$ is the total number of data samples. Note that contrary to the homogeneous setting where we assume $\{\mathcal{D}_j\}_{j=1}^p$ have the same distribution across devices and $L_i = L_j$, $1 \leq (i, j) \leq p$, in the heterogeneous setting these distributions and loss functions L_j can vary from a device to another, which is a very practical scenario in real-world federated learning applications.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGKDD'22, August 14-18, 2022, Washington, DC

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

In general, there are several challenges that need to be addressed in FL in order to efficiently learn a global model that performs well in average for all devices:

- *Communication-efficiency*: There are often many devices communicating with the server, thus incurring immense communication overhead. One approach to reduce the number of communication rounds is using *local SGD with periodic averaging* [39, 43, 47, 48] which periodically averages local models after a few local updates, contrary to baseline SGD [6] where gradient averaging is performed at each iteration. Local SGD has been proposed in [23, 33] under the FL setting and its convergence analysis is studied in [39, 43, 47, 48], later on improved in the followup references [3, 13, 21, 41] for homogeneous setting. It is further extended to heterogeneous setting [13, 20, 30, 31, 38, 46]. The second approach dealing with communication cost aims at reducing the size of communicated message per communication round, such as gradient quantization [1, 4, 42, 44, 45] or sparsification [2, 32, 40, 41].

- *Data heterogeneity*: Since locally generated data in each device may come from different distributions, local computations involved in FL setting can lead to poor convergence error in practice [28, 31]. To mitigate the negative impact of data heterogeneity, [12, 15, 20, 31] suggest applying variance reduction or gradient tracking techniques along local computations.

- *Privacy* [11, 14]: Privacy has been widely addressed by injecting an additional layer of randomness to respect differential-privacy property [34] or using cryptography-based approaches under secure multi-party computation [5]. Further study related to FL setting can be found in recent surveys [28] and [18].

To jointly tackle the aforementioned challenges in FL, sketching based algorithms [7, 9, 22, 26] have been used as promising tools. For instance, to reduce the communication cost, [17] develops a distributed SGD algorithm using sketching, along with convergence analysis in the homogeneous setting, and establishes a communication complexity of order $O(\log(d))$ per round, where d is the dimension of the gradient vector compared to $O(d)$ complexity per round of baseline mini-batch SGD. Nonetheless, the proposed sketching scheme in [17], built from a communication-efficiency perspective, is based on a deterministic procedure requiring access to the exact gradients information, hence not meeting the privacy-preserving criteria. This systemic issue is partially addressed in [37].

Focusing on privacy, [27] derives a single framework in order to address these issues and introduces DiffSketch algorithm, based on the Count Sketch operator, yet does not provide its convergence analysis. Besides, the estimation error of DiffSketch is higher than the sketching scheme in [17] which could lead to poor convergence.

In this paper, we propose new methods to tackle the communication bottleneck of FL baselines. Our main contributions are summarized as follows:

- We provide a new algorithm – HEAPRIX (HP) – and theoretically show that it reduces the cost of communication, based

on unbiased sketching without requiring the broadcast of exact values of gradients to the server. Based on HP, we develop general algorithms for communication-efficient and sketch-based FL, namely FedSKETCH and FedSKETCHGATE for homogeneous and heterogeneous settings respectively.

- We establish non-asymptotic convergence bounds for Polyak-Łojasiewicz (PL), convex and non-convex functions in Theorems 1 and 2 in both homogeneous and heterogeneous cases, and highlight an improvement in the number of iterations to reach a stationary point. We also provide *sharper* analysis for the PRIVIX(PR)/DiffSketch¹ algorithm proposed in [27].
- We illustrate the benefits of FedSKETCH and FedSKETCHGATE over baseline methods through a number of experiments. The latter shows the advantages of the HP method achieving comparable test accuracy as Federated SGD (FedSGD) while compressing the exchanged information.

Notation: We denote the number of communication rounds and bits per round and per device by R and B respectively. The count sketch of vector \mathbf{x} is designated by $S(\mathbf{x})$. $[p]$ denotes the set $\{1, \dots, p\}$.

2 COMPRESSION USING COUNT SKETCH

Throughout the paper, we employ the commonly used Count Sketch [7], see Algorithm 1, as the building component of our algorithms. In this paper, we exploit the commonly used Count Sketch [7] which uses two sets of functions that encode any input vector \mathbf{x} into a hash table $S_{m \times t}(\mathbf{x})$. Pairwise independent hash functions $\{h_{j, 1 \leq j \leq t} : [d] \rightarrow [m]\}$ are used along with another set of pairwise independent sign hash functions $\{\text{sign}_{j, 1 \leq j \leq t} : [d] \rightarrow \{+1, -1\}\}$ to map entries of \mathbf{x} ($x_i, 1 \leq i \leq d$) into t different columns of $S_{m \times t}$, wherein, to lower the dimension of the input vector, we usually have $d \gg mt$. The final update reads $S[j][h_j(i)] = S[j][h_j(i)] + \text{sign}_j(i)x_i$ for any $1 \leq j \leq t$.

Algorithm 1 Count Sketch (CS) [7]

- 1: **Inputs:** $\mathbf{x} \in \mathbb{R}^d, t, k, S_{m \times t}, h_j (1 \leq i \leq t), \text{sign}_j (1 \leq i \leq t)$
 - 2: **Compress vector** $\mathbf{x} \in \mathbb{R}^d$ **into** $S(\mathbf{x})$:
 - 3: **for** $x_i \in \mathbf{x}$ **do**
 - 4: **for** $j = 1, \dots, t$ **do**
 - 5: $S[j][h_j(i)] = S[j-1][h_{j-1}(i)] + \text{sign}_j(i) \cdot x_i$
 - 6: **end for**
 - 7: **end for**
 - 8: **return** $S_{m \times t}(\mathbf{x})$
-

2.1 Sketching based Unbiased Compressor

We define an unbiased compressor as follows:

DEFINITION 1 (UNBIASED COMPRESSOR). We call the randomized function $C : \mathbb{R}^d \rightarrow \mathbb{R}^d$ an unbiased compression operator with $\Delta \geq 1$, if

$$\mathbb{E}[C(\mathbf{x})] = \mathbf{x} \quad \text{and} \quad \mathbb{E}[\|C(\mathbf{x})\|_2^2] \leq \Delta \|\mathbf{x}\|_2^2.$$

We denote this class of compressors by $\mathbb{U}(\Delta)$.

¹We use PRIVIX (PR) and DiffSketch [27] interchangeably throughout the paper.

This definition leads to the following property

$$\mathbb{E}[\|C(\mathbf{x}) - \mathbf{x}\|_2^2] \leq (\Delta - 1) \|\mathbf{x}\|_2^2.$$

This property allows us to control the noise of the compression. Note that if $\Delta = 1$, then our algorithm reduces to the case of no compression. For instance, PR is an unbiased compressor which obtains an estimate of input \mathbf{x} from a count sketch noted $S(\mathbf{x})$. See [27] for more details, or Algorithm 6 in the Appendix. We give below a useful property of Count Sketch for our theoretical analysis.

PROPERTY 1 ([27]). For any $\mathbf{x} \in \mathbb{R}^d$:

- 1) *Unbiased estimation:* As in [27], we have $\mathbb{E}_S[\text{PR}[S(\mathbf{x})]] = \mathbf{x}$.
- 2) *Bounded variance:* For the given $m < d, t = O\left(\ln\left(\frac{d}{\delta}\right)\right)$ with probability $1 - \delta$ we have:

$$\mathbb{E}_S[\|\text{PR}[S(\mathbf{x})] - \mathbf{x}\|_2^2] \leq c \frac{d}{m} \|\mathbf{x}\|_2^2,$$

where c ($e \leq c < m$) is a positive constant independent of the dimension of the input, d .

We note that this bounded variance assumption does not necessarily mean that compression is happening since dimension d may be relatively large. Thus, with probability $1 - \delta$, we obtain $\text{PR} \in \mathbb{U}(1 + c \frac{d}{m})$. $\Delta = 1 + c \frac{d}{m}$ implies that if $m \rightarrow d$, then $\Delta \rightarrow 1 + c$, indicates a noisy reconstruction. [27] shows that if the data is normally distributed, PR is differentially private [10], up to additional assumptions and algorithmic design choices.

2.2 Sketching based Biased Compressor

A biased compressor is defined as follows:

DEFINITION 2 (BIASED COMPRESSOR). A (randomized) function, $C : \mathbb{R}^d \rightarrow \mathbb{R}^d$ belongs to $\mathbb{C}(\Delta, \alpha)$, a class of compression operators with $\alpha > 0$ and $\Delta \geq 1$, if

$$\mathbb{E}[\|\alpha \mathbf{x} - C(\mathbf{x})\|_2^2] \leq \left(1 - \frac{1}{\Delta}\right) \|\mathbf{x}\|_2^2.$$

It is proven in [16] that $\mathbb{U}(\Delta) \subset \mathbb{C}(\Delta, \alpha)$. An example of biased compression using sketching methods and using a top_m operator is provided below:

Algorithm 2 HEAVYMIX (HX) (Modified [17])

- 1: **Inputs:** $S(\mathbf{g})$; parameter m
 - 2: Query the vector $\tilde{\mathbf{g}} \in \mathbb{R}^d$ from $S(\mathbf{g})$:
 - 3: Query $\tilde{\ell}_2^2 = (1 \pm 0.5) \|\mathbf{g}\|^2$ from sketch $S(\mathbf{g})$
 - 4: $\forall j$ query $\tilde{\mathbf{g}}_j^2 = \tilde{\mathbf{g}}_j^2 \pm \frac{1}{2m} \|\mathbf{g}\|^2$ from sketch $S(\mathbf{g})$
 - 5: $H = \{j | \tilde{\mathbf{g}}_j \geq \frac{\tilde{\ell}_2}{m}\}$ and $NH = \{j | \tilde{\mathbf{g}}_j < \frac{\tilde{\ell}_2}{m}\}$
 - 6: $\text{Top}_m = H \cup \text{rand}_\ell(NH)$, where $\ell = m - |H|$
 - 7: Get exact values of Top_m
 - 8: **Output:** $\tilde{\mathbf{g}} : \forall j \in \text{Top}_m : \tilde{\mathbf{g}}_i = \mathbf{g}_i$ else $\mathbf{g}_i = 0$
-

Following [17], HEAVYMIX (HX) with sketch size $\Theta\left(m \log\left(\frac{d}{\delta}\right)\right)$ is a biased compressor with $\alpha = 1$ and $\Delta = d/m$ with probability $\geq 1 - \delta$. In other words, with probability $1 - \delta$, $\text{HX} \in \mathbb{C}\left(\frac{d}{m}, 1\right)$. Note that Algorithm 2 is a variant of the sketching algorithm developed in [17] with the distinction that HX does not require any second

round of communication to obtain the exact values of top_m . This is mainly because in SKETCHED-SGD [17], the server receives the exact values of *the average of the sketches* while HX obtains exact local values. Additionally, while HX has a smaller estimation error compared to PR, in PR the central server does need to have access to the exact values of local gradient providing user privacy as underlined in [27]. In the following, we introduce HX which is built upon HX and PR methods.

2.3 Sketching based Induced Compressor

From Theorem 3 in [16], stating that a biased compressor can be converted into an unbiased one such that, for $C_1 \in \mathbb{C}(\Delta_1)$ with $\alpha = 1$, if one chooses $C_2 \in \mathbb{U}(\Delta_2)$, then the induced compressor $C : x \mapsto C_1(x) + C_2(x - C_1(x))$ belongs to $\mathbb{U}(\Delta)$ with $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1}$.

Algorithm 3 HEAPRIX (HP)

```

1: Inputs:  $x \in \mathbb{R}^d, S_{m \times t}, m < t$ 
2: Approximate  $S(x)$  using HX
3: Approximate  $S(x - \text{HX}[S(x)])$  with PR
4: Output:
    $\text{HP}(x) \triangleq \text{HX}[S(x)] + \text{PR}[S(x - \text{HX}[S(x)])]$ 

```

Based on this notion, Algorithm 3 proposes an induced sketching algorithm by utilizing HX and PR for C_1 and C_2 respectively where the reconstruction of input x is performed using hash table S and x , similar to PR and HX. Note that if $m \rightarrow d$, then $C(x) \rightarrow x$, implying that the convergence rate can be improved by decreasing the size of compression m .

COROLLARY 1. *Based on Theorem 3 of [16], HX in Algorithm 3 satisfies $C(x) \in \mathbb{U}(c \frac{d}{m})$.*

3 FEDSKETCH AND FEDSKETCHGATE

We introduce two new algorithms for both homogeneous and heterogeneous settings.

3.1 Homogeneous Setting

In FedSKETCH, we denote by τ the number of local updates, between two consecutive communication rounds, at device j . Unlike [12], the server does not store any global model, rather, device j has two models: $x^{(r)}$ and $x_j^{(\ell, r)}$, which respectively denote the global and local models. We develop FedSKETCH in Algorithm 4 with a variant of this algorithm implementing HEAPRIX. For this variant, we need to have an additional communication round between the server and worker j to aggregate $\delta_j^{(r)} \triangleq S_j[\text{HX}(S^{(r)})]$ (Lines 4 and 4) to compute $S^{(r)} = \frac{1}{k} \sum_{j \in \mathcal{K}} S_j^{(r)}$. The main difference between FedSKETCH and DiffSketch in [27] is that we use distinct local and global learning rates. Furthermore, unlike [27], we do not add local Gaussian noise as privacy is not the main focus of this paper.

Algorithmic comparison with [12] An important feature of our algorithm is that due to a lower dimension of the count sketch, the resulting averaged quantities ($S^{(r)}$ and $\tilde{S}^{(r)}$) received by the server are also of lower dimension. Therefore, our algorithms exploit a bidirectional compression during the communication phases between

Algorithm 4 FedSKETCH(R, τ, η, γ)

```

1: Inputs:  $x^{(0)}$ : initial model shared by local devices, global and
   local learning rates  $\gamma$  and  $\eta$ , respectively
2: for  $r = 0, \dots, R - 1$  do
3:   parallel for device  $j \in \mathcal{K}^{(r)}$  do
4:     if PRIVIX variant:
        $\text{HP}^{(r)} \triangleq \text{PR}[S^{(r-1)}]$ 
5:     if HEAPRIX variant:
        $\text{HP}^{(r)} \triangleq \text{HX}[S^{(r-1)}] + \text{PR}[S^{(r-1)} - \tilde{S}^{(r-1)}]$ 
6:     Set  $x^{(r)} = x^{(r-1)} - \gamma \text{HP}^{(r)}$  and  $x_j^{(0, r)} = x^{(r)}$ 
7:     for  $\ell = 0, \dots, \tau - 1$  do
8:       Sample a mini-batch  $\xi_j^{(\ell, r)}$  and compute  $\tilde{g}_j^{(\ell, r)}$ 
9:       Update  $x_j^{(\ell+1, r)} = x_j^{(\ell, r)} - \eta \tilde{g}_j^{(\ell, r)}$ 
10:    end for
11:    Device  $j$  broadcasts  $S_j^{(r)} \triangleq S_j(x_j^{(0, r)} - x_j^{(\tau, r)})$ .
12:    Server computes  $S^{(r)} = \frac{1}{k} \sum_{j \in \mathcal{K}} S_j^{(r)}$ .
13:    Server broadcasts  $S^{(r)}$  to devices in randomly drawn de-
       vices  $\mathcal{K}^{(r)}$ .
14:    if HEAPRIX variant:
15:      Second round of communication for computing  $\delta_j^{(r)} :=$ 
        $S_j[\text{HX}(S^{(r)})]$  and broadcasts  $\tilde{S}^{(r)} \triangleq \frac{1}{k} \sum_{j \in \mathcal{K}} \delta_j^{(r)}$  to devices in
       set  $\mathcal{K}^{(r)}$ 
16:    end parallel for
17:  end
18: Output:  $x^{(R-1)}$ 

```

server and devices. As a result, for the case of large quantization error $\omega = \theta(\frac{d}{m})$ as shown in [12], our algorithms can outperform those developed in [12] if sufficiently large hash tables are used and the uplink communication cost is high. Furthermore, while in [12], server stores a global model and aggregates the partial gradients from devices enabling it to extract some information regarding the device's local data, in contrast, in our algorithms server does not store the global model and only broadcasts the average sketches.

REMARK 1. *As discussed in [16], induced compressors transform a biased compressor into an unbiased one at the cost of doubling the communication cost since the devices need to send $C_1(x)$ and $C_2(x - C_1(x))$ separately. Here, we emphasize that with HEAPRIX, due to sketching techniques, the extra communication round cost is compensated with lower number of bits per round thanks to the lower dimension of the sketches.*

Benefits of HEAPRIX (HP) based algorithms: Corollary 1 states that, unlike PR, HP compression noise can be made as small as possible using larger hash size. In the distributed setting, unlike SKETCHED-SGD [17] where decompression is happening in the server, HP does not require access to exact top_m values of the input rather it can only have access to sketches of aggregated local gradients. This is because HP uses HX where decompression is performed at each device, thus not requiring exact values of gradients of each device.

Algorithm 5 FedSKETCHGATE(R, τ, η, γ)

```

1: Inputs:  $\mathbf{x}^{(0)} = \mathbf{x}_j^{(0)}$  shared by all local devices, global and local
   learning rates  $\gamma$  and  $\eta$ .
2: for  $r = 0, \dots, R-1$  do
3:   parallel for device  $j = 1, \dots, p$  do
4:     if PRIVIX variant:
5:        $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau} \left[ \text{PR}(\mathbf{S}^{(r-1)}) - \text{PR}(\mathbf{S}_j^{(r-1)}) \right]$ 
       where  $\text{HP}^{(r)} \triangleq \text{PR}(\mathbf{S}^{(r-1)})$ 
6:     if HEAPRIX variant:
7:        $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau} \left( \text{HP}^{(r)} - \text{HP}_j^{(r)} \right)$ 
8:     Set  $\mathbf{x}^{(r)} = \mathbf{x}^{(r-1)} - \gamma \text{HP}^{(r)}$  and  $\mathbf{x}_j^{(0,r)} = \mathbf{x}^{(r)}$ 
9:     for  $\ell = 0, \dots, \tau-1$  do
10:      Sample mini-batch  $\xi_j^{(\ell,r)}$  and compute  $\tilde{\mathbf{g}}_j^{(\ell,r)}$ 
11:       $\mathbf{x}_j^{(\ell+1,r)} = \mathbf{x}_j^{(\ell,r)} - \eta \left( \tilde{\mathbf{g}}_j^{(\ell,r)} - \mathbf{c}_j^{(r)} \right)$ 
12:    end for
13:    Device  $j$  broadcasts  $\mathbf{S}_j^{(r)} \triangleq \mathbf{S} \left( \mathbf{x}_j^{(0,r)} - \mathbf{x}_j^{(\tau,r)} \right)$ .
14:    Server computes  $\mathbf{S}^{(r)} = \frac{1}{p} \sum_{j=1}^p \mathbf{S}_j^{(r)}$  and broadcasts  $\mathbf{S}^{(r)}$ 
    to all devices.
15:    if HEAPRIX variant:
16:      Device  $j$  computes  $\text{HP}_j^{(r)} \triangleq \text{HP}[\mathbf{S}_j^{(r)}]$ .
17:      Second round of communication to obtain  $\delta_j^{(r)} := \mathbf{S}_j \left( \text{HX}[\mathbf{S}^{(r)}] \right)$ .
18:      Broadcasts  $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p} \sum_{j=1}^p \delta_j^{(r)}$  to devices.
19:    end parallel for
20:  end
21: Output:  $\mathbf{x}^{(R-1)}$ 

```

In other words, HP-based algorithm leverages the best of both: the unbiasedness of PR while using heavy hitters as in HX.

3.2 Heterogeneous Setting

In this section, we focus on the optimization problem of (1) where $q_1 = \dots = q_p = \frac{1}{p}$ with full device participation ($k = p$). These results can be extended to the scenario where a subset of devices is sampled. For non i.i.d. data, the FedSKETCH algorithm, designed for homogeneous setting, may fail to perform well in practice. The main reason is that in FL, devices are using local stochastic descent direction which could be different than global descent direction when the data distribution are non-identical. For that reason, to mitigate the negative impact of data heterogeneity, we introduce a new algorithm called FedSKETCHGATE described in Algorithm 5. The main idea, that our algorithm is based upon, is the use of an approximation of the global gradient, $\mathbf{c}_j^{(r)}$ allows to correct the local gradient direction, see gradient tracking technique applied in [12, 31]. Using PR variant, the correction vector $\mathbf{c}_j^{(r)}$ at device j and communication round r is computed in Line 5. Using HP variant, FedSKETCHGATE updates $\tilde{\mathbf{S}}^{(r)}$ via Line 5.

REMARK 2. Most of the existing communication-efficient algorithms with compression only consider gradient-compression from devices to server. However, Algorithms 4 and 5 improve the communication efficiency from server to devices as it exploits low-dimensional sketches in a bidirectional manner.

For both FedSKETCH and FedSKETCHGATE, HP variant requires a second round of communication, unlike PR. Therefore, in Cross-Device FL setting, where there could be millions of devices, HP variant may not be practical, and we note that it could be more suitable for Cross-Silo FL settings.

4 CONVERGENCE ANALYSIS

We first state commonly used assumptions required in the following convergence analysis (reminder of our notations can be found Table 3 of the Appendix).

ASSUMPTION 1 (SMOOTHNESS AND LOWER BOUNDEDNESS). The local objective function $f_j(\cdot)$ of device j is differentiable for $j \in [p]$ and L -smooth, i.e., $\|\nabla f_j(\mathbf{x}) - \nabla f_j(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Moreover, the optimal objective function $f(\cdot)$ is bounded below by $f^* := \min_{\mathbf{x}} f(\mathbf{x}) > -\infty$.

We present our results for PL, convex and general non-convex objectives. [19] show that strong convexity implies PL condition with same module (PL objectives can also be non-convex, hence PL condition does not imply the strong convexity necessarily).

4.1 Convergence of FEDSKETCH

We start with the homogeneous case where data is i.i.d. among local devices, and therefore, the stochastic local gradient of each worker is an unbiased estimator of the global gradient. We make the following assumption under that setting:

ASSUMPTION 2 (BOUNDED VARIANCE). For all $j \in [m]$, we sample an independent mini-batch ℓ_j of size $|\xi_j^{(\ell,r)}| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\mathbf{x}; \xi_j)$, $\mathbb{E}_{\xi_j}[\tilde{\mathbf{g}}_j] = \nabla f(\mathbf{x}) = \mathbf{g}$ with the variance bounded by a constant σ^2 : $\mathbb{E}_{\xi_j}[\|\tilde{\mathbf{g}}_j - \mathbf{g}\|^2] \leq \sigma^2$.

THEOREM 1. Suppose Assumptions 1-2 holds. Given $0 < m \leq d$ and considering Algorithm 4 with sketch size $B = O\left(m \log\left(\frac{dR}{\delta}\right)\right)$ and $\gamma \geq k$, with probability $1 - \delta$ we have:

In the **non-convex** case, $\{\mathbf{x}^{(r)}\}_{r=0}^R$ satisfies $\frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left[\left\| \nabla f(\mathbf{x}^{(r)}) \right\|_2^2 \right] \leq \epsilon$ if:

• FS-PRIVIX, for $\eta = \frac{1}{L\gamma} \sqrt{\frac{1}{R\tau\left(\frac{cd}{m} + \frac{1}{k}\right)}}$:

$$R = O(1/\epsilon) \quad \text{and} \quad \tau = O\left(\left(c\frac{d}{m} + \frac{1}{k}\right)/(\epsilon)\right).$$

• FS-HEAPRIX, for $\eta = \frac{1}{L\gamma} \sqrt{\frac{1}{R\tau\left(\frac{cd-m}{m} + \frac{1}{k}\right)}}$:

$$R = O(1/\epsilon) \quad \text{and} \quad \tau = O\left(\left(\frac{cd-m}{m} + \frac{1}{k}\right)/\epsilon\right).$$

In the **PL or strongly convex** case, $\{\mathbf{x}^{(r)}\}_{r=0}^R$ satisfies $\mathbb{E}[f(\mathbf{x}^{(R-1)}) - f(\mathbf{x}^{(*)})] \leq \epsilon$ if we set:

- FS-PRIVIX, for $\eta = \frac{1}{2L(cd/mk+1)\tau\gamma}$:

$$R = O\left(\left(cd/m + \frac{1}{k}\right) \kappa \log(1/\epsilon)\right)$$

$$\tau = O\left((cd/m + 1) \left/ (cd/m + 1/k) \right. \epsilon\right).$$

- FS-HEAPRIX, for $\eta = \frac{1}{2L((cd-m)/m+1/k)\tau\gamma}$:

$$R = O\left((cd-m)/m + 1/k\right) \kappa \log(1/\epsilon)$$

$$\tau = O\left(cd/m \left/ ((cd-m-1) + 1/k) \right. \epsilon\right).$$

In the **Convex** case, $\{\mathbf{x}^{(r)}\}_{r \geq 0}$ satisfies $\mathbb{E}[f(\mathbf{x}^{(R-1)}) - f(\mathbf{x}^{(*)})] \leq \epsilon$ if we set:

- FS-PRIVIX, for $\eta = \frac{1}{2L(cd/m+1/k)\tau\gamma}$:

$$R = O(L(1/k + d/m) / \epsilon \log(1/\epsilon)) \quad \text{and} \quad \tau = O(1/\epsilon^2).$$

- FS-HEAPRIX, for $\eta = \frac{1}{2L((cd-m)/mk+1)\tau\gamma}$:

$$R = O(L(1/k + (d-m)/m) / \epsilon \log(1/\epsilon)) \quad \text{and} \quad \tau = O(1/\epsilon^2).$$

The constant c comes from Property 1. The bounds in Theorem 1 suggest that, under the homogeneous setting, if we set $d = m$, i.e. no compression, the number of communication rounds to achieve the ϵ -error matches with the number of iterations required to achieve the same error under a centralized setting. Furthermore, for non-convex objective, FS-HEAPRIX improves the computational complexity over FS-PRIVIX, and for PL objectives FS-HEAPRIX improves communication complexity over FS-PRIVIX. Furthermore, this improvement is validated through our experiments in Section 5. Additionally, we can see that the computational complexity scales down with the number of sampled devices. To further stress on the impact of using sketching methods, we also compare our results with prior works in terms of total number of communicated bits per device.

Comparison with [17]. From a privacy aspect, we note that [17] requires for the central server to have access to exact values of top_m gradients, hence does not preserve privacy, whereas our schemes do not need those exact values. From a communication cost point of view, for strongly convex objectives and compared to [17], we improve the total communication per worker from $RB = O\left(\frac{d}{\epsilon} \log\left(\frac{d}{\delta\sqrt{\epsilon}} \max\left(\frac{d}{m}, \frac{1}{\sqrt{\epsilon}}\right)\right)\right)$ to

$$RB = O\left(\kappa(cd-m + \frac{m}{k}) \log \frac{1}{\epsilon} \log\left(\frac{\kappa d}{\delta} \left(\frac{cd-m}{m} + 1/k\right) \log \frac{1}{\epsilon}\right)\right).$$

We note that while reducing communication cost, our scheme requires $\tau = O(cd/m((\frac{cd-m}{m} + 1/k)\epsilon)) > 1$, which scales down with the number of sampled devices k . Moreover, unlike [17], we do not use the classical bounded gradient assumption and thus obtain stronger results with weaker assumptions. Regarding general non-convex objectives, our result improves the total communication cost per worker displayed in [17] from

$$RB = O\left(\max\left(\frac{1}{\epsilon^2}, \frac{d^2}{k^2\epsilon}\right) \log\left(\frac{d}{\delta} \max\left(\frac{1}{\epsilon^2}, \frac{d^2}{k^2\epsilon}\right)\right)\right)$$

for only one device to $RB = O(\frac{m}{\epsilon} \log(\frac{d}{\delta}))$. We also highlight that we can obtain similar rates for Algorithm 4 in heterogeneous environment with the uniform boundedness gradient assumption.

Note: Such an improved communication cost over prior works can be explained by – the joint exploitation of *sketching* – the reduction of the dimension of communicated messages – the use of *local updates* – the reduction of the number of communication rounds reaching a desired convergence error.

4.2 Convergence of FedSKETCHGATE

We start by assuming that the local variance is bounded:

ASSUMPTION 3 (BOUNDED LOCAL VARIANCE). For all $j \in [p]$, we can sample an independent mini-batch Ξ_j of size $|\xi_j| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\mathbf{x}; \xi_j)$ with $\mathbb{E}_{\xi}[\tilde{\mathbf{g}}_j] = \nabla f_j(\mathbf{x}) = \mathbf{g}_j$. Moreover, the variance of local stochastic gradients is bounded such that $\mathbb{E}_{\xi}[\|\tilde{\mathbf{g}}_j - \mathbf{g}_j\|^2] \leq \sigma^2$.

THEOREM 2. Suppose Assumptions 1 and 3 hold. Given $0 < m \leq d$, and considering FedSKETCHGATE, Algorithm 5, with sketch size $B = O\left(m \log\left(\frac{dR}{\delta}\right)\right)$ and $\gamma \geq p$ with probability $1 - \delta$ we have

In the **non-convex** case, $\eta = \frac{1}{L\gamma} \sqrt{\frac{mp}{R\tau(cd)}}$, $\{\mathbf{x}^{(r)}\}_{r \geq 0}$ satisfies

$$\frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left[\left\| \nabla f(\mathbf{x}^{(r)}) \right\|_2^2 \right] \leq \epsilon \text{ if:}$$

- FS-PRIVIX: $R = O((cd+m)/m\epsilon)$ and $\tau = O(1/(p\epsilon))$.
- FS-HEAPRIX: $R = O(d/m\epsilon)$ and $\tau = O(1/(p\epsilon))$.

In the **PL or Strongly convex** case, $\{\mathbf{x}^{(r)}\}_{r \geq 0}$ satisfies $\mathbb{E}[f(\mathbf{x}^{(R-1)}) - f(\mathbf{x}^{(*)})] \leq \epsilon$ if:

- FS-PRIVIX, for $\eta = 1/(2L(\frac{cd}{m} + 1)\tau\gamma)$:

$$R = O\left(\left(c\frac{d}{m} + 1\right) \kappa \log(1/\epsilon)\right) \quad \text{and} \quad \tau = O(1/(p\epsilon)).$$

- FS-HEAPRIX, for $\eta = m/(2cLd\tau\gamma)$:

$$R = O\left(\left(c\frac{d}{m}\right) \kappa \log(1/\epsilon)\right) \quad \text{and} \quad \tau = O(1/(p\epsilon)).$$

In the **convex** case, $\{\mathbf{x}^{(r)}\}_{r \geq 0}$ satisfies $\mathbb{E}[f(\mathbf{x}^{(R-1)}) - f(\mathbf{x}^{(*)})] \leq \epsilon$ if:

- FS-PRIVIX, for $\eta = 1/(2L(cd/m + 1)\tau\gamma)$:

$$R = O(L(d/m + 1)\epsilon \log(1/\epsilon)) \quad \text{and} \quad \tau = O(1/(p\epsilon^2)).$$

- FS-HEAPRIX, for $\eta = m/(2Lcd\tau\gamma)$:

$$R = O(L(d/m)\epsilon \log(1/\epsilon)) \quad \text{and} \quad \tau = O(1/(p\epsilon^2)).$$

Theorem 2 implies that the number of communication rounds and local updates are similar to the corresponding quantities in homogeneous setting except for the non-convex case where the number of rounds also depends on the compression rate (summarized Table 1-2 of the Appendix). We note that the convergence analysis of FS-PRIVIX provided in [27] for convex objectives is further tightened here. Moreover, FS-HEAPRIX improves the communication complexity of FS-PRIVIX for both PL and non-convex objectives which is empirically validated in Figures 1 and 2.

4.3 Comparison with Prior Methods

Main competing baselines of our methods are distributed algorithms based on sketching. Nonetheless, we also compare with prior non-sketching based distributed algorithms ([3, 12, 20, 36]) in the Supplementary material.

Li et al. [27]. Note that our convergence analysis does not rely on the bounded gradient assumption. We also improve both the number of communication rounds R and the size of transmitted bits B per communication round (please see Table 2 of Section A in appendix). Additionally, we highlight that, while [27] provides a convergence analysis for convex objectives, our analysis holds for PL (thus strongly convex case), general convex and general non-convex objectives.

Rothchild et al. [37]. Due to gradient tracking, our algorithm tackles data heterogeneity, while algorithms in [37] do not. Thereby, in FedSKETCHGATE each device has to store an additional state vector compared to [37]. Yet, as our method is built upon an unbiased compressor, server does not need to store any additional error correction vector. The convergence results for both FetchSGD variants in [37] rely on the uniform bounded gradient assumption which may not be applicable with L -smoothness assumption when data distribution is highly heterogeneous, as it is commonly the case in FL, see [21]. Besides, Theorem 1 [37] assumes that *Contraction Holds* for the sequence of gradients which may not hold in practice, yet based on this strong assumption, their total communication cost (RB) in order to achieve ϵ error is $RB = O\left(m \max\left(\frac{1}{\epsilon^2}, \frac{d^2-dm}{m^2\epsilon}\right) \log\left(\frac{d}{\delta} \max\left(\frac{1}{\epsilon^2}, \frac{d^2-dm}{m^2\epsilon}\right)\right)\right)$. For the sake of comparison, we let the compression ratio in [37] to be $\frac{m}{cd}$. In contrast, without any extra assumptions, our results in Theorem 2 for PRIVIX and HEAPRIX are respectively

$$RB = O\left(\frac{(cd+m)}{\epsilon} \log\left(\frac{(cd^2)}{\epsilon\delta} + d\right)\right) \quad \text{and} \quad RB = O\left(c\frac{d}{\epsilon} \log\left(\frac{cd^2}{\epsilon m\delta}\right)\right)$$

which improves the total communication cost of [37], Theorem 1, under regimes such that $\frac{1}{\epsilon} \geq d$ or $d \gg m$. Theorem 2 in [37] is based the *Sliding Window Heavy Hitters* assumption, which is similar to the gradient diversity assumption in [13, 29]. Under that assumption, the total communication cost is shown to be $RB = O\left(\frac{m \max(I^{2/3}, 2-\alpha)}{\epsilon^3\alpha} \log\left(\frac{d \max(I^{2/3}, 2-\alpha)}{\epsilon^3\delta}\right)\right)$ where I is a constant related to the window of gradients. We improve this bound under weaker assumptions in a regime where $\frac{I^{2/3}}{\epsilon^3} \geq d$. We also provide bounds for PL, convex and non-convex objectives contrary to [37].

5 NUMERICAL STUDY

In this section, we provide numerical experiments on benchmark datasets to illustrate the efficacy of the proposed methods.

5.1 Experiments on MNIST

MNIST [25] is a popular benchmark dataset in federated learning. It contains handwritten digits from ten classes 0 to 9, with 60000 samples for training and 10000 samples for testing, where each data point is a 28×28 image.

5.1.1 Setup. We train LeNet-5 Convolutional Neural Network (CNN) architecture introduced in [25], with 60 000 parameters. We compare Federated SGD (FedSGD) as the full-precision baseline, along with four sketching methods SketchSGD [17], FetchSGD [37], and two FedSketch variants FS-PRIVIX and FS-HEAPRIX. Note that in Algorithm 4, FS-PRIVIX with global learning rate $\gamma = 1$ is equivalent to the DiffSketch algorithm proposed in [29]. Thus, FS-PRIVIX itself can be regarded as a baseline method DiffSketch for comparison. Also, SketchSGD is slightly modified to compress the change in local weights (instead of local gradient in every iteration), and FetchSGD is implemented with second round of communication for fairness. (The original proposal does not include second round of communication, which performs worse with small sketch size.) As suggested in [37], the momentum factor of FetchSGD is set to 0.9. TopK compression is used to decode the sketches at the central server, and we apply the same implementation trick that when accumulating the errors, we only count the non-zero coordinates and leave other coordinates zero for the accumulator. This indeed greatly improves the empirical performance of FetchSGD.

In all experiments, the number of workers is set to 50 and we report the results for 1~5 local epochs. A local epoch is finished when all workers go through their local data samples once. The local mini-batch size is 30. In each round, we randomly choose half of the devices to be active. We tune the learning rates (η and γ , if applicable) over log-scale and report the best results, and all results are averaged over 3 independent runs.

5.1.2 Results. We now present the experimental results for both *homogeneous* and *heterogeneous* setting. In the former case, each device receives uniformly drawn data samples, and in the latter, it only receives samples from one or two classes among ten, which creates a highly non-iid client data distribution.

Homogeneous case. In Figure 1, we provide the training loss and test accuracy with different number of local epochs and sketch size, $(t, k) = (20, 40)$ and $(50, 100)$. Note that, these two choices of sketch size correspond to a $75\times$ and $12\times$ compression ratio, respectively. We conclude that

- In general, increasing the compression ratio sacrifices the learning performance. In all cases, FS-HEAPRIX performs the best in terms of both training objective and test accuracy, among all compressed methods.
- FS-HEAPRIX is better than FS-PRIVIX, especially with small sketches (high compression ratio). FS-HEAPRIX yields acceptable extra test error compared to full-precision FedSGD, particularly under high compression ratio (e.g., $75\times$).
- The training performance of FS-HEAPRIX improves when the number of local updates increases. *That is, the proposed method is able to further reduce the communication cost by reducing the number of rounds required for communication.* This is also consistent with our theoretical findings.

In general, FS-HEAPRIX outperforms all competing methods, and a sketch size of $(50, 100)$ is sufficient to approach the accuracy of full-precision FedSGD.

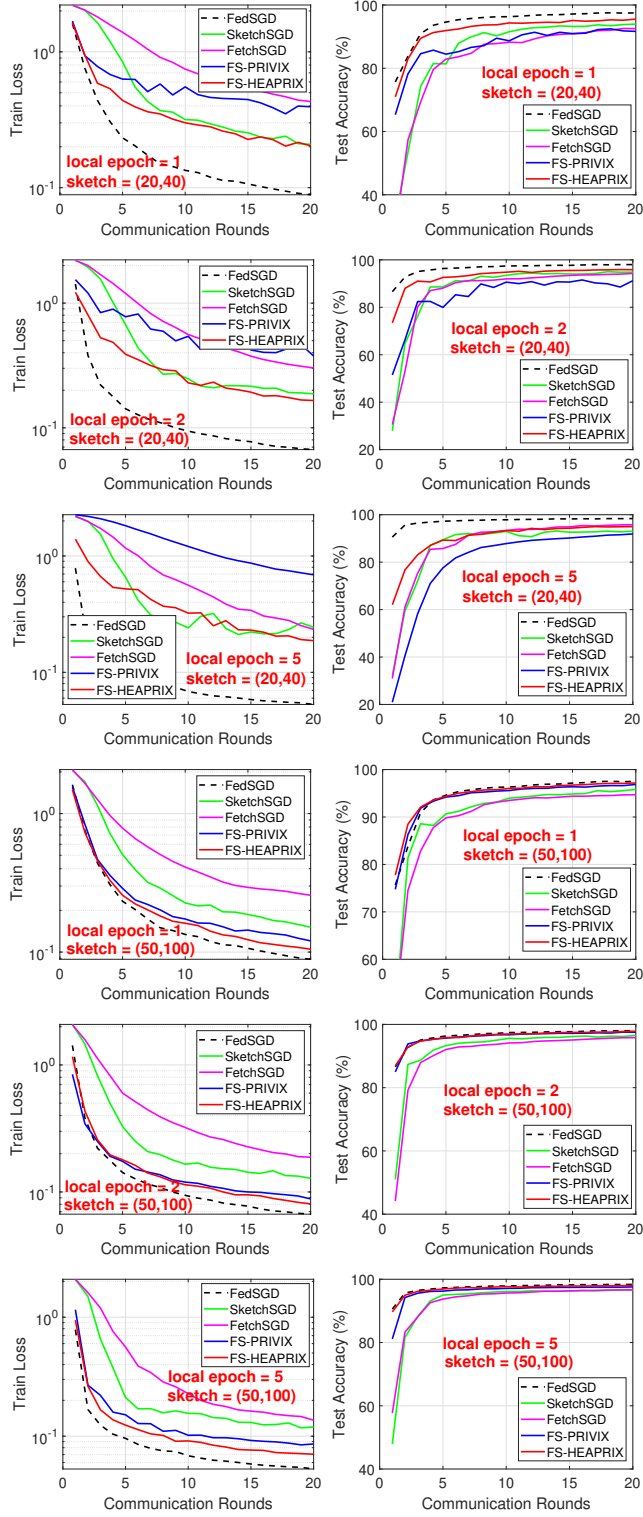


Figure 1: Homogeneous case on MNIST: Comparison of compressed FL methods on LeNet CNN.

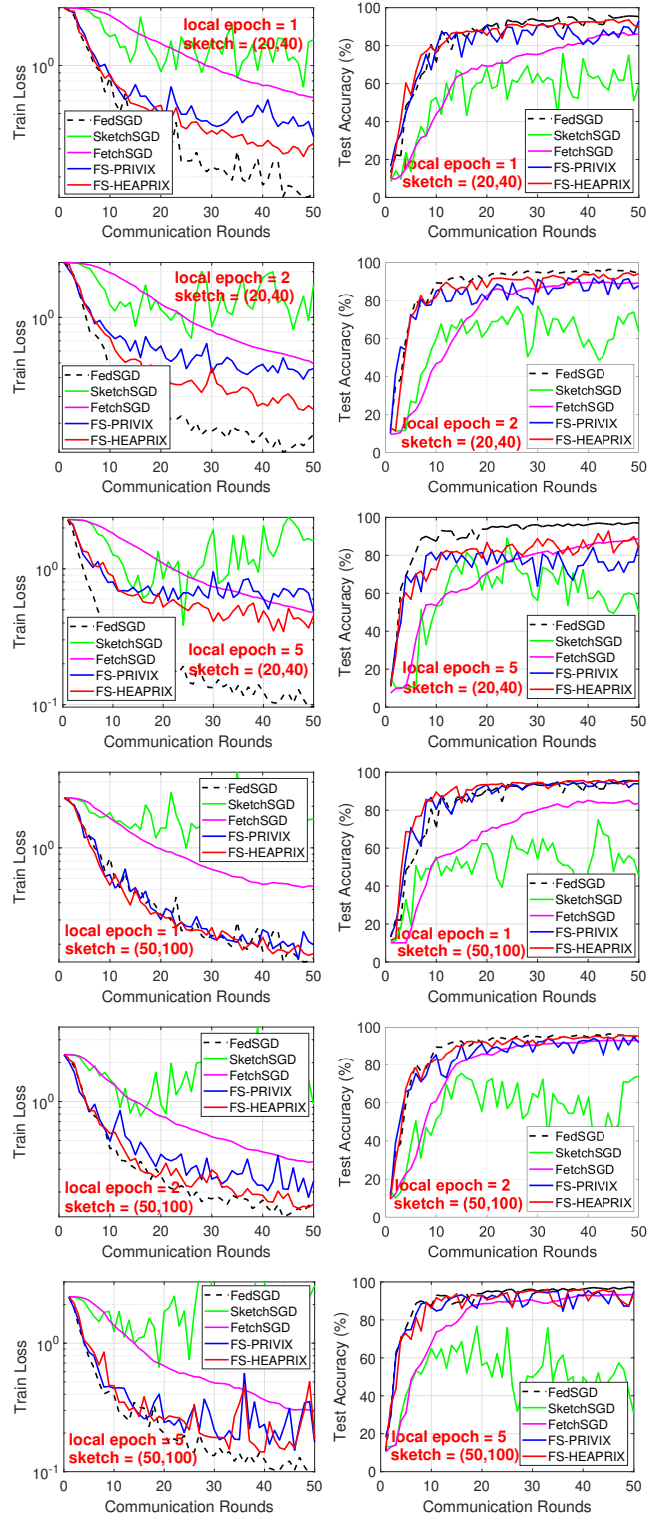


Figure 2: Heterogeneous case on MNIST: Comparison of compressed FL algorithms on LeNet CNN.

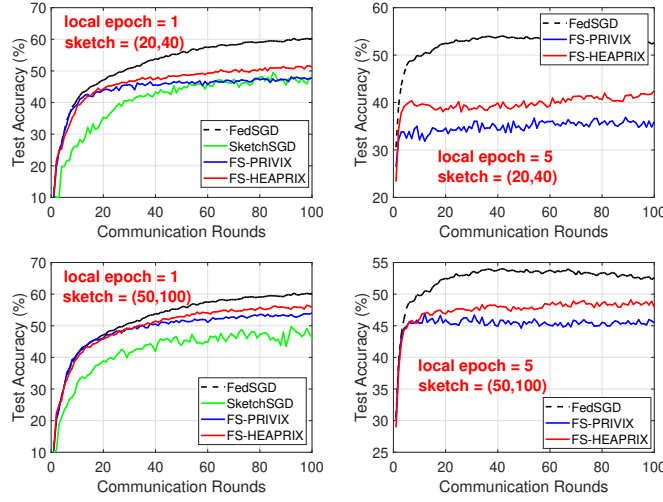


Figure 3: Homogeneous case on CIFAR10: Comparison of compressed FL methods on LeNet CNN.

Heterogeneous case. We plot similar set of results in Figure 2 for non-i.i.d. data distribution, which leads to more twists and turns in the training curves. We see that SketchSGD performs very poorly in the heterogeneous case, which is improved by error tracking and momentum in FetchSGD, as expected. However, both of these methods are worse than our proposed FedSketchGATE methods, which can achieve similar generalization accuracy as full-precision FedSGD, even with small sketch size (i.e., $75\times$ compression). Note that, slower convergence and worse generalization of FedSGD in non-i.i.d. data distribution case is also reported in e.g. [8, 33].

We also notice in Figure 2 the edge of FS-HEAPRIX over FS-PRIVIX in terms of training loss and test accuracy. However, we see that in the heterogeneous setting, more local updates tend to undermine the learning performance, especially with small sketch size. Nevertheless, when the sketch size is not too small, i.e., (50, 100), FS-HEAPRIX can still provide comparable test accuracy as FedSGD in both cases. Our empirical study demonstrates that FedSketch (and FedSketchGATE) frameworks are able to perform well in homogeneous (resp. heterogeneous) settings, with high compression rate. In particular, FedSketch methods are beneficial over SketchedSGD [17] and FetchSGD [37] in all cases. FS-HEAPRIX performs the best among all the tested compressed algorithms, which in many cases achieves similar generalization accuracy as full-precision FedSGD with small sketch size.

5.2 Experiments on CIFAR-10

We conduct similar sets of experiments on CIFAR10 [24] dataset, whose data samples are 3-channel 32×32 natural images split into 10 classes. We also use the simple LeNet CNN structure, as in practice small models are more favorable in federated learning due to the limitation of wireless client/mobile devices. The experimental setup is similar as that for MNIST. The test accuracy is presented in Figure 3 and Figure 4, for respectively homogeneous and heterogeneous data distribution. In general, we retrieve similar information

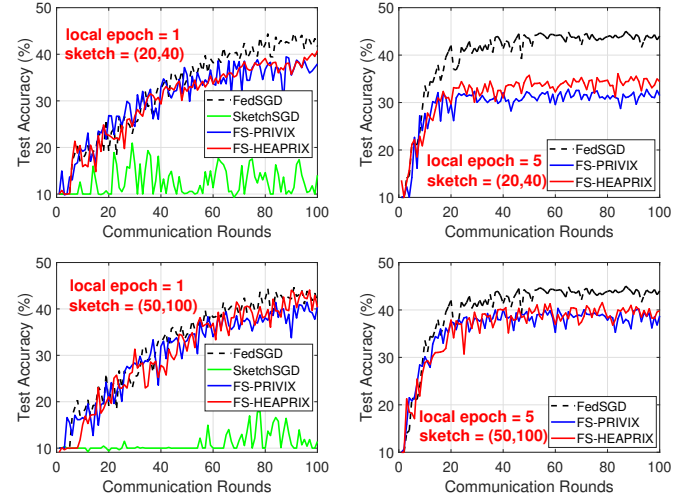


Figure 4: Heterogeneous case on CIFAR10: Comparison of compressed FL methods on LeNet CNN.

as from MNIST experiments: our proposed FS-HEAPRIX improves FS-PRIVIX/DiffSketch and SketchedSGD in all cases. In the heterogeneous case, FS-HEAPRIX is able to match the accuracy of full-precision FedSGD with sketch size (50, 100) (with 1 local epoch). In these experiments, we observe that too many local iterations would undermine the performance in general, as it may violate the communication bounds of the corresponding FL methods (both FedSGD and our algorithms). We note that although the test accuracy provided by LeNet cannot reach the state-of-the-art accuracy given by some huge models (which is less practical for FL), it is also informative in terms of comparing the relative performance of different FL methods.

6 CONCLUSION

In this paper, we introduced FedSKETCH and FedSKETCHGATE algorithms for homogeneous and heterogeneous data distribution setting respectively for Federated Learning wherein communication between server and devices is only performed using count sketch. Our algorithms, thus, provide communication-efficiency and privacy, through random hashes based sketches. We analyze the convergence error for *non-convex*, *PL* and *general convex* objective functions in the scope of Federated Optimization. We provide insightful numerical experiments showcasing the advantages of our FedSKETCH and FedSKETCHGATE methods over current federated optimization algorithm. The proposed algorithms outperform competing compression method and can achieve comparable test accuracy as Federated SGD, with high compression ratio.

REFERENCES

- [1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NIPS)*. Long Beach, 1709–1720.
- [2] Dan Alistarh, Torsten Hoefer, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. 2018. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*. Montréal,

- Canada, 5973–5983.
- [3] Debraj Basu, Deepesh Data, Can Karakus, and Suhas N. Diggavi. 2019. Qsparse-local-SGD: Distributed SGD with Quantization, Sparsification and Local Computations. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 14668–14679.
- [4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. SIGNSGD: Compressed Optimisation for Non-Convex Problems. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Stockholm, Sweden, 559–568.
- [5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Dallas, TX, 1175–1191.
- [6] Léon Bottou and Olivier Bousquet. 2008. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada, 161–168.
- [7] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. 2004. Finding frequent items in data streams. *Theoretical Computer Science* 312, 1 (2004), 3–15. [https://doi.org/10.1016/S0304-3975\(03\)00400-6](https://doi.org/10.1016/S0304-3975(03)00400-6)
- [8] Xiangyi Chen, Xiaoyun Li, and Ping Li. 2020. Toward Communication Efficient Adaptive Gradient Method. In *ACM-IMS Foundations of Data Science Conference (FODS)*. Seattle, WA.
- [9] Graham Cormode and Shan Muthukrishnan. 2005. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms* 55, 1 (2005), 58–75.
- [10] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II (Lecture Notes in Computer Science)*, Vol. 4052. Springer, 1–12.
- [11] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).
- [12] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. 2020. Federated Learning with Compression: Unified Analysis and Sharp Guarantees. *arXiv preprint arXiv:2007.01154* (2020).
- [13] Farzin Haddadpour and Mehrdad Mahdavi. 2019. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425* (2019).
- [14] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677* (2017).
- [15] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. 2019. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115* (2019).
- [16] Samuel Horváth and Peter Richtárik. 2020. A Better Alternative to Error Feedback for Communication-Efficient Distributed Learning. *arXiv preprint arXiv:2006.11077* (2020).
- [17] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. 2019. Communication-efficient Distributed SGD with Sketching. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada, 13144–13154.
- [18] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).
- [19] Hamed Karimi, Julie Nutini, and Mark Schmidt. 2016. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-Lojasiewicz Condition. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*. Riva del Garda, Italy, 795–811.
- [20] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2019. SCAFFOLD: Stochastic Controlled Averaging for On-Device Federated Learning. *arXiv preprint arXiv:1910.06378* (2019).
- [21] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. 2020. Tighter Theory for Local SGD on Identical and Heterogeneous Data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. Online [Palermo, Sicily, Italy], 4519–4529.
- [22] Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery* 7, 4 (2003), 373–397.
- [23] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
- [24] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
- [25] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
- [26] Ping Li, Kenneth Ward Church, and Trevor Hastie. 2008. One sketch for all: Theory and Application of Conditional Random Sampling. In *Advances in Neural Information Processing Systems (NIPS)*. Vancouver, Canada, 953–960.
- [27] Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. 2019. Privacy for Free: Communication-Efficient Learning with Differential Privacy Using Sketches. *arXiv preprint arXiv:1911.00972* (2019).
- [28] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* 37, 3 (2020), 50–60.
- [29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems (MLSys)*. Austin, TX.
- [30] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia.
- [31] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. 2019. Variance Reduced Local SGD with Lower Communication Complexity. *arXiv preprint arXiv:1912.12844* (2019).
- [32] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Vancouver, Canada.
- [33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Fort Lauderdale, FL, 1273–1282.
- [34] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. Vancouver, Canada.
- [35] Constantin Philippenko and Aymeric Dieuleveut. 2020. Artemis: tight convergence guarantees for bidirectional compression in Federated Learning. *arXiv preprint arXiv:2006.14591* (2020).
- [36] Amirhossein Reiszadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. Online [Palermo, Sicily, Italy], 2021–2031.
- [37] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. 2020. FetchSGD: Communication-Efficient Federated Learning with Sketching. *arXiv preprint arXiv:2007.07682* (2020).
- [38] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2018. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).
- [39] Sebastian Urban Stich. 2019. Local SGD Converges Fast and Communicates Little. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*. New Orleans, LA.
- [40] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*. Montréal, Canada, 4447–4458.
- [41] Sebastian U Stich and Sai Praneeth Karimireddy. 2019. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350* (2019).
- [42] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. 2018. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems (NeurIPS)*. Montréal, Canada, 7652–7662.
- [43] Jianyu Wang and Gauri Joshi. 2018. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576* (2018).
- [44] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2017. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems (NIPS)*. Long Beach, CA, 1509–1519.
- [45] Jiaxiang Wu, Weidong Huang, Junzhou Huang, and Tong Zhang. 2018. Error compensated quantized SGD and its applications to large-scale distributed optimization. *arXiv preprint arXiv:1806.08054* (2018).
- [46] Hao Yu, Rong Jin, and Sen Yang. 2019. On the Linear Speedup Analysis of Communication Efficient Momentum SGD for Distributed Non-Convex Optimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Long Beach, CA, 7184–7193.
- [47] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*. Honolulu, HI, 5693–5700.
- [48] Fan Zhou and Guojing Cong. 2018. On the Convergence Properties of a K-step Averaging Stochastic Gradient Descent Algorithm for Nonconvex Optimization. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*. Stockholm, Sweden, 3219–3227.

Appendix for Communication-Efficient Federated Learning via Sketching with Sharp Rates

A DEFINITIONS AND COMPARISON WITH PRIOR WORKS

Complete appendix can be found in Appendix.

A.1 PRIVIX method and compression error of HEAPRIX

For the sake of completeness we review PRIVIX algorithm that is also mentioned in [27] as follows:

Algorithm 6 PRIVIX/DiffSketch [27]: Unbiased compressor based on sketching.

```

1: Inputs:  $\mathbf{x} \in \mathbb{R}^d, t, m, S_{m \times t}, h_j (1 \leq i \leq t), \text{sign}_j (1 \leq i \leq t)$ 
2: Query  $\tilde{\mathbf{x}} \in \mathbb{R}^d$  from  $S(\mathbf{x})$ :
3: for  $i = 1, \dots, d$  do
4:    $\tilde{\mathbf{x}}[i] = \text{Median}\{\text{sign}_j(i).S[j][h_j(i)] : 1 \leq j \leq t\}$ 
5: end for
6: Output:  $\tilde{\mathbf{x}}$ 

```

For the purpose of further clarification, we summarize the comparison of our results with related works. We recall that p is the number of devices, d is the dimension of the model, κ is the condition number, ϵ is the target accuracy, R is the number of communication rounds, and τ is the number of local updates. We start with the homogeneous setting comparison. Comparison of our results and existing ones for homogeneous and heterogeneous setting are given respectively Table 1 and Table 2.

Comparison with [12] and [36] Convergence analysis of algorithms in [12] relies on unbiased compression, while in this paper our FL algorithm based on HEAPRIX enjoys from unbiased compression with equivalent biased compression variance. Moreover, we highlight that the convergence analysis of FedCOMGATE is based on the extra assumption of boundedness of the difference between the average of compressed vectors and compressed averages of vectors. However, we do not need this extra assumption as it is satisfied naturally due to linearity of sketching. Finally, as pointed out in Remark 2, our algorithms enjoy from a bidirectional compression property, unlike FedCOMGATE in general. Furthermore, since results in [12] improve the communication complexity of FedPAQ algorithm, developed in [36], hence FedSKETCH and FedSKETCHGATE improves the communication complexity obtained in [36].

[3]. We note that the algorithm in [3] uses a composed compression and quantization while our algorithm is solely based on compression. So, in order to compare with algorithms in [3] we only consider Qsparse-local-SGD with compression and we let compression factor $\gamma = \frac{m}{cd}$ (to compare with the same compression ratio induced with sketch size of mt). For strongly convex objective in Qsparse-local-SGD to achieve convergence error of ϵ they require $R = O\left(\kappa \frac{d}{m\sqrt{\epsilon}}\right)$ and $\tau = O\left(\frac{m}{pd\sqrt{\epsilon}}\right)$, which is improved to $R = O\left(\frac{ckd}{m} \log(1/\epsilon)\right)$ and $\tau = O\left(\frac{1}{p\epsilon}\right)$ for PL objectives. Similarly, for non-convex objective [3] requires $R = O\left(\frac{d}{m\epsilon^{1.5}}\right)$ and $\tau = O\left(\frac{m}{pd\sqrt{\epsilon}}\right)$, which is improved to $R = O\left(c \frac{d}{m\epsilon}\right)$ and $\tau = O\left(\frac{1}{p\epsilon}\right)$. We note that we reduce communication rounds at the cost of increasing number of local updates (which scales down with number of devices, p). Additionally, we highlight that our FedSKETCHGATE exploits the gradient tracking idea to deal with data heterogeneity, while algorithms in [3] does not develop such mechanism and may suffer from poor convergence in heterogeneous setting. We also note that setting $\tau = 1$ and using top_m compressor, the QSPARSE-local-SGD algorithm becomes similar to distributed SGD with sketching as they both use the error feedback framework to improve the compression variance. Finally, since the average of sparse vectors may not be sparse in general the number of transmitted bits from server to devices in QSPARSE-Local-SGD in [3] may not be sparse in general ($B = O(d)$), however our algorithms enjoy from bidirectional compression properly due to lower dimension and linearity properties of sketching ($B = O(m \log(\frac{Rd}{\delta}))$). Therefore, the total number of bits per device for strongly convex and non-convex objective is improved respectively from $RB = O\left(\kappa \frac{d^2}{m\sqrt{\epsilon}}\right)$ and $RB = O\left(\frac{d^2}{m\epsilon^{1.5}}\right)$ in [3] to $RB = O\left(\kappa d \log(\frac{ckd^2}{m\delta} \log(\frac{1}{\epsilon})) \log(1/\epsilon)\right) = O\left(\kappa d \max\left(\log(\frac{ckd^2}{m\delta}), \log^2(1/\epsilon)\right)\right)$ and $RB = O\left(\log(c \frac{d^2}{m\epsilon\delta}) \frac{d}{\epsilon}\right)$. Additionally, as we noted using sketching for transmission implies two way communication from master to devices and vice versa. Therefore, in order to show efficacy of our algorithm we compare our convergence analysis with the obtained rates in the following related work:

[35]. The reference [35] considers two-way compression from parameter server to devices and vice versa. They provide the convergence rate of $R = O\left(\frac{\omega^{\text{Up}} \omega^{\text{Down}}}{\epsilon^2}\right)$ for strongly-objective functions where ω^{Up} and ω^{Down} are uplink and downlink's compression noise (specializing to our case for the sake of comparison $\omega^{\text{Up}} = \omega^{\text{Down}} = \theta(d)$) for general heterogeneous data distribution. In contrast, while our algorithms are using bidirectional compression due to use of sketching for communication, our convergence rate for strongly-convex objective is $R = O(\kappa \mu^2 d \log(\frac{1}{\epsilon}))$ with probability $1 - \delta$.

Table 1: Comparison of results with compression and periodic averaging in the homogeneous setting. Here, p is the number of devices, μ is the PL constant, m is the number of bins of hash tables, d is the dimension of the model, κ is the condition number, ϵ is the target accuracy, R is the number of communication rounds, and τ is the number of local updates. UG and PP stand for Unbounded Gradient and Privacy Property respectively.

Reference	Non-Convex	UG	PP
[27]	–	–	$R = O\left(\frac{\mu^2 d}{\epsilon^2}\right), \tau = 1$ $B = O\left(k \log\left(\frac{dR}{\delta}\right)\right)$ $pRB = O\left(\frac{p\mu^2 d}{\epsilon^2} k \log\left(\frac{\mu^2 d^2}{\epsilon^2 \delta}\right)\right)$
Ivkin et al. [17]	$R = O\left(\max\left(\frac{d}{m\sqrt{\epsilon}}, \frac{1}{\epsilon}\right)\right), \tau = 1, B = O\left(m \log\left(\frac{dR}{\delta}\right)\right)$ $pRB = O\left(\frac{pd}{m\epsilon} \log\left(\frac{d}{\delta\sqrt{\epsilon}} \max\left(\frac{d}{m}, \frac{1}{\sqrt{\epsilon}}\right)\right)\right)$	✗	✗
Theorem 1	$R = O\left(\frac{1}{\epsilon}\right)$ $\tau = O\left((\mu^2(cd - m) + \frac{\mu^2}{k})\frac{1}{\epsilon}\right)$ $B = O(m \log(\frac{dR}{\delta}))$ $kBR = O(mk/\epsilon \log(\frac{d}{\epsilon\delta}))$	✓	✗

Table 2: Comparison of results with compression and periodic averaging in the heterogeneous setting. UG and PP stand for Unbounded Gradient and Privacy Property respectively.

Reference	non-convex	General Convex	UG	PP
Basu et al. [3] (with $\gamma = m/d$)	$R = O\left(\frac{d}{m\epsilon^{1.5}}\right) \quad \tau = O\left(\frac{m}{pd\sqrt{\epsilon}}\right)$ $B = O(d) \quad RB = O\left(\frac{d^2}{m\epsilon^{1.5}}\right)$	–	✗	✗
Li et al. [27]	–	$R = O\left(\frac{d}{m\epsilon^2}\right)$ $\tau = 1$ $B = O\left(m \log\left(\frac{d^2}{m\epsilon^2\delta}\right)\right)$	✗	✓
Rothchild et al. [37]	$R = O\left(\max\left(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2\epsilon}\right)\right) \quad \tau = 1$ $B = O\left(m \log\left(\frac{d}{\delta} \max\left(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2\epsilon}\right)\right)\right)$ $RB = O\left(m \max\left(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2\epsilon}\right) \log\left(\frac{d}{\delta} \max\left(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2\epsilon}\right)\right)\right)$	–	✗	✗
Rothchild et al. [37]	$R = O\left(\frac{\max(I^{2/3}, 2 - \alpha)}{\epsilon^3}\right) \quad \tau = 1$ $B = O\left(\frac{m}{\alpha} \log\left(\frac{d \max(I^{2/3}, 2 - \alpha)}{\epsilon^3 \delta}\right)\right)$ $RB = O\left(\frac{m \max(I^{2/3}, 2 - \alpha)}{\epsilon^3 \alpha} \log\left(\frac{d \max(I^{2/3}, 2 - \alpha)}{\epsilon^3 \delta}\right)\right)$	–	✗	✗
Theorem 2	$R = O\left(\frac{d}{m\epsilon}\right) \quad \tau = O\left(\frac{1}{p\epsilon}\right)$ $B = O(m \log(\frac{d^2}{m\epsilon\delta}))$ $RB = O(\frac{d}{\epsilon} \log(\frac{d^2}{m\epsilon\delta} \log(\frac{1}{\epsilon})))$	$R = O(\frac{d}{m\epsilon} \log(\frac{1}{\epsilon}))$ $\tau = O(\frac{1}{p\epsilon^2})$ $B = O(m \log(\frac{d^2}{m\epsilon\delta}))$	✓	✓