FedSKETCH: Communication-Efficient and Private Federated Learning via Sketching

Abstract

Communication complexity and privacy are the two key challenges in Federated Learning where the goal is to perform a distributed learning through a large volume of devices. In this work, we introduce `FedSKETCH` and `FedSKETCHGATE` algorithms to address both challenges in Federated learning jointly, where these algorithms are intended to be used for homogeneous and heterogeneous data distribution settings respectively. The key idea is to compress the accumulation of local gradients using count sketch, therefore, the server does not have access to the gradients themselves which provides privacy. Furthermore, due to the lower dimension of sketching used, our method exhibits communication-efficiency property as well. We provide, for the aforementioned schemes, sharp convergence guarantees. Finally, we back up our theory with various set of experiments.

1 Introduction

Increasing applications in machine learning include the learning of a complex model across a large amount of devices in a distributed manner. In the particular case of federated learning, the training data is stored across these multiple devices and can not be centralized. Two natural problems arise from this setting. First, communications bottlenecks appear when a central server and the multiple devices must exchange gradient-informed quantities. Then, privacy-related issues due to the protection of the sensitive individual data must be taken into account.

The former has extensively been tackled via quantization [1], sparsification [35] and compression [4] methods yielding to a drastic reduction of the number of bits required to communicate those gradient-related information. Solving the privacy issue has been widely executed injecting an additional layer of random noise in order to respect differential-privacy property of the method.

With the focus of communication-efficiency, [13] proposes a distributed SGD algorithm using sketching and they provide the convergence analysis in homogeneous data distribution setting.

Also with focus on privacy, in [18], the authors derive a single framework in order to tackle these issues jointly and introduce `DiffSketch` based on the Count Sketch operator. Compression and privacy is performed using random hash functions such that no third parties are able to access the original data. Yet, [18] does not provide the convergence analysis for the `DiffSketch` in Federated setting. In this work, we provide a thorough convergence analysis for the Federated Learning using sketching.

The main contributions of this paper are summarized as follows:

- Based on the current compression methods, we provide a new algorithm – `HEAPRIX` – that displays an unbiased estimator of the full gradient we ought to communicate to the central parameter server. We theoretically show that `HEAPRIX` jointly reduces the cost of communication between devices and server, preserves privacy and is unbiased.

- We develop a general algorithm for communication-efficient and privacy preserving federated learning based on this novel compression algorithm. Those methods, namely `FedSKETCH` and `FedSKETCHGATE`, are derived under *homogeneous* and *heterogeneous* data distribution settings.

- Non asymptotic analysis of our method is established for convex, Polyak-Łojasiewicz (generalization of strongly-convex) and nonconvex functions in Theorem 2 and Theorem 3 for respectively the i.i.d. and non i.i.d. case, and highlight an improvement in the number of iteration required to achieve a stationarity point.

2 Related Work

In this section, we provide a summary of the prior related research efforts as follows:

Local SGD with Periodic Averaging: Compared to baseline SGD where model averaging happens in every iteration, the main idea behind *Local SGD with periodic averaging* comes from the intuition of variance reduction by periodic model averaging [38] with purpose of saving communication rounds. While Local SGD has been proposed in [24, 16] under the title of Federated Learning Setting, the convergence analysis of Local SGD is studied in [39, 37, 31, 34]. The convergence analysis of Local SGD is improved in the follow up works [8, 2, 10, 3, 33] in majority for homogeneous data distribution setting. The convergence analysis is further extended to heterogeneous setting, wherein studied under the title of *Federated Learning*, with improved rates in [36, 20, 30, 21, 10, 14].

Additionally, a few recent Federated Learning/Local SGD with adaptive gradient methods can be found in [26, **?**].

ToDo: Revise this section!

Gradient Compression Based Algorithms for Distributed Setting: [13] develop a solution for leveraging sketches of full gradients in a distributed setting while training a global model using SGD [28, 5]. They introduce `Sketched-SGD` and establish a communication complexity of order $\mathcal{O}(\log(d))$ (per round) where d is the dimension of the parameters, i.e. the dimension of the gradient. Other recent solutions to reduce the communication cost include quantized gradient as developed in [1, 22, 32, 11]. Yet, their dependence on the number of devices p makes them harder to be used in some settings. Additionally, there are other research efforts such as [9, 27, 2, 11] that exploit compression in Federated Learning or distributed communication-efficient optimization. Finally, the recent work in [12] exploits variance reduction technique with compression jointly in distributed optimization.

Privacy-preserving Setting: Differentially private methods for federated learning have been extensively developed and studied in [18, 23] recently.

The remaining of the paper is organized as follows. Section 3 gives a formal presentation of the general problem. Section 4 describes the various compression algorithms used for communication efficiency and privacy preservation, and introduces our new compression method. The training algorithms are provided in Section 5 and their respective analysis in the strongly-convex or nonconvex cases are provided Section 6.

Notation: For the rest of the paper we indicate the number of communication rounds and number of bits per round per device with R and B respectively. For the rest of the paper we indicate the count sketch of any vector $\boldsymbol{x}$ with $\mathbf{S}(\boldsymbol{x})$.

3 Problem Setting

The federated learning optimization problem across p distributed devices is defined as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d,\,\sum_{j=1}^p q_j=1} f(\boldsymbol{x}) \triangleq \left[\sum_{j=1}^{p} q_j F_j(\boldsymbol{x})\right] \tag{1}$$

where $F_j(\boldsymbol{x}) = \mathbb{E}_{\xi\in\mathcal{D}_j}\left[L_j\left(\boldsymbol{x},\xi\right)\right]$ is the local cost function at device j, $q_j \triangleq \frac{n_j}{n}$ with n_j shows the number of data shards at device j and $n = \sum_{j=1}^p n_j$ is the total number of data samples. ξ is a random variable with probability distribution $\mathcal{D}_j$, and L_j is a loss function that measures the performance of model $\boldsymbol{x}$. We note that, while for the homogeneous data distribution, we assume $\mathcal{D}_j$ for $1\leq j\leq p$ have the same distribution and $L_1 = L_2 = \ldots = L_p$, in the heterogeneous setting these data distributions and loss functions L_j can be different from device to device.

We focus on solving optimization problem in Eq. (1) for the homogeneous data distribution but for the heterogeneous setting we consider the special case of $q_1 = \ldots = q_p = \frac{1}{p}$.

4 Count Sketch as a Compression Operation

A common sketching solution employed to tackle (1) called `Count Sketch` (for more detail see the seminal works [6, 7, 15]) is described Algorithm 1. The algorithm for generating count sketching is using two sets of functions that encode any input vector $\boldsymbol{x}$ **into a hash table** $\boldsymbol{S}_{t\times m}(\boldsymbol{x})$. We use hash functions $\{h_{j,1\leq j\leq t} : [d] \to m\}$ (which are pairwise independent) along with another set of pairwise independent sign hash functions $\{\text{sign}_j : [d] \to \{+1, -1\}\}$ to map every entry of $\boldsymbol{x}$ ($\boldsymbol{x}_i$, $1 \leq i \leq d$) into t different columns of hash table $\mathbf{S}_{t\times m}$. This steps are summarized in Algorithm 1.

4.1 Unbiased Compressor

Definition 1 (Unbiased compressor)**.** *A randomized function,* $C : \mathbb{R}^d \to \mathbb{R}^d$ *is called an unbiased compression operator with* $\Delta \geq 1$*, if we have*

$$\mathbb{E}\left[C(\boldsymbol{x})\right] = \boldsymbol{x} \quad \textit{and} \quad \mathbb{E}\left[\|C(\boldsymbol{x})\|_2^2\right] \leq \Delta \|\boldsymbol{x}\|_2^2 \,.$$

We indicate this class of compressor with $C \in \mathbb{U}(\Delta)$*.*

We note that this definition leads to the property

$$\mathbb{E}\left[\|\mathrm{C}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2\right] \leq (\Delta - 1) \|\boldsymbol{x}\|_2^2 \,.$$

Remark 1. *Note that in case of* $\Delta = 1$ *our algorithm reduces for the case of no compression. This property allows us to control the noise of the compression.*

4.2 An Example of Unbiased Compressor via Sketching

An instance of such unbiased compressor is `PRIVIX` which obtains an estimate of input $\boldsymbol{x}$ to a count sketching $\boldsymbol{S}(\boldsymbol{x})$. In this algorithm to query x_i, the $i-th$ element of the vector, we compute the median of t approximated value specified by the indices of $h_j(i)$ for $1 \leq j \leq t$. These steps are summarized in Algorithm 2.

Next we review a few properties of `PRIVIX` as follows:

Property 1 ([18])**.** *For our proof purpose we will need the following crucial properties of the count sketch described in Algorithm 1, for any real valued vector* $\mathbf{x} \in \mathbb{R}^d$*:*

1) Unbiased estimation*: As it is also mentioned in [18], we have:*

$$\mathbb{E}_{\mathbf{S}}\left[\mathtt{PRIVIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]\right] = \mathbf{x} \,.$$

2) Bounded variance*: With* $m = \mathcal{O}\left(\frac{e}{\mu^2}\right)$ *and* $t = \mathcal{O}\left(\ln\left(\frac{1}{\delta}\right)\right)$*, we have the following bound with probability* $1 - \delta$*:*

$$\mathbb{E}_{\mathbf{S}}\left[\|\mathtt{PRIVIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right] - \mathbf{x}\|_2^2\right] \leq \mu^2 d \|\mathbf{x}\|_2^2 \,.$$

Algorithm 1 `CS` [15]: Count Sketch to compress $\boldsymbol{x} \in \mathbb{R}^d$.

Inputs: $\boldsymbol{x} \in \mathbb{R}^d, t, k, \mathbf{S}_{t\times m}, h_j(1 \leq i \leq t), sign_j(1 \leq i \leq t)$
Compress vector $\boldsymbol{x} \in \mathbb{R}^d$ **into** $\mathbf{S}\left(\boldsymbol{x}\right)$**:**
for $\boldsymbol{x}_i \in \boldsymbol{x}$ **do**
  for $j = 1, \cdots, t$ **do**
    $\mathbf{S}[j][h_j(i)] = \mathbf{S}[j-1][h_{j-1}(i)] + \text{sign}_j(i).\boldsymbol{x}_i$
  end for
end for
return $\mathbf{S}_{t\times m}(\boldsymbol{x})$

Algorithm 2 PRIVIX[18]: Unbiased compressor based on sketching.

Inputs: $\boldsymbol{x} \in \mathbb{R}^d, t, m, \mathbf{S}_{t\times m}, h_j(1 \le i \le t), sign_j(1 \le i \le t)$
Query $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$ **from** $\mathbf{S}(\boldsymbol{x})$**:**
for $i = 1, \ldots, d$ **do**
    $\tilde{\boldsymbol{x}}[i] = \text{Median}\{\text{sign}_j(i).\mathbf{S}[j][h_j(i)] : 1 \le j \le t\}$
end for
Output: $\tilde{\boldsymbol{x}}$

Therefore, PRIVIX $\in \mathbb{U}(1 + \mu^2 d)$ with probability $1 - \delta$.

Remark 2. *We note that $\Delta = 1 + \mu^2 d$ implies that if $m \to d$, $\Delta \to 1 + 1 = 2$, which means that the case of no compression is not covered. Thus, the algorithms based on this may converges poorly.*

Definition 2. *A randomized mechanism $\mathcal{O}$ satisfies $\epsilon-$differential privacy, if for input data S_1 and S_2 differing by up to one element, and for any output D of $\mathcal{O}$,*

$$\Pr\left[\mathcal{O}(S_1) \in D\right] \le \exp\left(\epsilon\right) \Pr\left[\mathcal{O}(S_2) \in D\right] .$$

Assumption 1 (Input vector distribution)**.** *For the purpose of privacy analysis, similar to 3, we suppose that for any input vector S with length $|S| = l$, each element $s_i \in S$ is drawn i.i.d. from a Gaussian distribution: $s_i \sim \mathcal{N}(0, \sigma^2)$, and bounded by a large probability: $|s_i| \le C, 1 \le i \le p$ for some positive constant $C > 0$.*

Theorem 1 ($\epsilon-$ differential privacy of count sketch, [18])**.** *For a sketching algorithm $\mathcal{O}$ using Count Sketch $\mathbf{S}_{t\times m}$ with t arrays of m bins, for any input vector S with length l satisfying Assumption 1, $\mathcal{O}$ achieves $t.\ln\left(1 + \frac{\alpha C^2 m(m-1)}{\sigma^2(l-2)}(1 + \ln(l-m))\right)-$differential privacy with high probability, where α is a positive constant satisfying $\frac{\alpha C^2 m(m-1)}{\sigma^2(l-2)}(1 + \ln(l-m)) \le \frac{1}{2} - \frac{1}{\alpha}$.*

The proof of this theorem can be found in [18].

4.3 Biased compressor

Definition 3 (Biased compressor)**.** *A (randomized) function, $C : \mathbb{R}^d \to \mathbb{R}^d$ is called a compression operator with $\alpha > 0$ and $\Delta \ge 1$, if we have*

$$\mathbb{E}\left[\left\|\alpha \boldsymbol{x} - \bar{C}(\boldsymbol{x})\right\|_2^2\right] \le \left(1 - \frac{1}{\Delta}\right)\|\boldsymbol{x}\|_2^2 ,$$

then, any biased compression operator C is indicated by $C \in \mathbb{C}(\Delta, \alpha)$.

The following Lemma links these two definitions:

Lemma 1 ([12])**.** *We have $\mathbb{U}(\Delta) \subset \mathbb{C}(\Delta)$.*

An instance of biased compressor based on sketching is given in Algorithm 3.

Algorithm 3 HEAVYMIX

Inputs: $\mathbf{S}_\mathbf{g}$; parameter-m
Compress vector $\tilde{\mathbf{g}} \in \mathbb{R}^d$ **into** $\mathbf{S}\left(\tilde{\mathbf{g}}\right)$**:**
Query $\hat{\ell}_2^2 = (1 \pm 0.5)\left\|\mathbf{g}\right\|^2$ from sketch $\mathbf{S}_\mathbf{g}$
$\forall j$ query $\hat{\mathbf{g}}_j^2 = \hat{\mathbf{g}}_j^2 \pm \frac{1}{2m}\left\|\mathbf{g}\right\|^2$ from sketch $\mathbf{S}_\mathbf{g}$
$H = \{j | \hat{\mathbf{g}}_j \ge \frac{\hat{\ell}_2^2}{m}\}$ and $NH = \{j | \hat{\mathbf{g}}_j < \frac{\hat{\ell}_2^2}{m}\}$
$\text{Top}_m = H \cup rand_\ell(NH)$, where $\ell = m - |H|$
Get exact values of Top_m
Output: $\mathbf{g}_S : \forall j \in \text{Top}_m : \mathbf{g}_{Si} = \mathbf{g}_i$ and $\forall j \notin \text{Top}_m : \mathbf{g}_{Si} = 0$

Lemma 2 ([13]). *HEAVYMIX, 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 $1-\delta$, HEAVYMIX $\in C(\frac{d}{m}, 1)$.*

We note that Algorithm 3 is a variation of sketching algorithm in [13] with distinction that `HEAVYMIX` does not require extra second round of communcation to obtain the exact value of top_k.

4.4 Sketching Based on Induced Compressor

The following Lemma from [12] shows that how we can transfer biased compressor into an unbiased compressor:

Lemma 3 (Induced Compressor [12]). *For $C_1 \in \mathbb{C}(\Delta_1)$ with $\alpha = 1$, choose $C_2 \in \mathbb{U}(\Delta_2)$ and define the induced compressor with*

$$C(\mathbf{x}) = C_1(\mathbf{x}) + C_2\left(x - C_1\left(\mathbf{x}\right)\right),$$

then, the induced compressor C satisfies $C \in \mathbb{U}(\mathbf{x})$ with $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1}$.

Remark 3. *We note that if $\Delta_2 \geq 1$ and $\Delta_1 \leq 1$, we have $\Delta = \Delta_2 + \frac{1-\Delta_2}{\Delta_1} \leq \Delta_2$.*

Using this concept of the induced compressor we introduce `HEAPRIX`:

Algorithm 4 `HEAPRIX`

1: **Inputs:** $\boldsymbol{x} \in \mathbb{R}^d, t, m, \mathbf{S}_{t\times m}, h_j(1 \leq i \leq t), sign_j(1 \leq i \leq t)$, parameter-$m$
2: **Approximate** $\mathbf{S}(x)$ **using** `HEAVYMIX`
3: **Approximate** $\mathbf{S}\left(x - \texttt{HEAVYMIX}[\mathbf{S}(x)]\right)$ **using** `PRIVIX`
4: **Output:** $\texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right] + \texttt{PRIVIX}\left[\mathbf{S}\left(\mathbf{x} - \texttt{HEAVYMIX}\left[\mathbf{S}\left(\mathbf{x}\right)\right]\right)\right]$

Corollary 1. *Based on Lemma 3 and using Algorithm 4, we have $C(x) \in \mathbb{U}(\mu^2 d)$.*

Remark 4. *We highlight that in this case if $m \to d$, then $C(x) \to x$ which means that your convergence algorithm can be improved by decreasing the noise of compression (with choice of bigger m).*

In the following we define two general framework for different sketching algorithms for homogeneous and heterogeneous data distributions.

5 Algorithms for homogeneous and heterogeneous settings

In the following, first we present two algorithm for homogeneous setting. Then, we present two algorithms for heterogeneous algorithms to deal with data heterogeneity.

5.1 Homogeneous setting

In this section, we propose two algorithms for the setting where data at distributed devices is correlated. The proposed Federated Learning with averaging uses sketching to compress communication. The main difference between first algorithm and the algorithm in [18] is that we use distinct local and global learning rates. Additionally, unlike [18] we do not add add local Gaussian noise for the privacy purpose.

In `FedSKETCH`, we indicate the number of communication rounds between devices and server with R, and the number of local updates at device j is illustrated with τ, which happens between two consecutive communication rounds. Unlike [9], server node does not store any global model, instead device j has two models, $\boldsymbol{x}^{(r)}$ and $\boldsymbol{x}_j^{(\ell,r)}$. In communication round r device j, the local model $\boldsymbol{x}_j^{(\ell,r)}$ is updated using the rule

$$\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\tilde{\mathbf{g}}_j^{(\ell,r)} \qquad\qquad \text{for } \ell = 0,\ldots,\tau-1\,,$$

where $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \Xi_j^{(\ell,r)}) \triangleq \frac{1}{b}\sum_{\xi\in\Xi_j^{(\ell,r)}} \nabla L_j(\boldsymbol{x}_j^{(\ell,r)}, \xi)$ is a stochastic gradient of f_j evaluated using the mini-batch $\Xi_j^{(\ell,r)} = \{\xi_{j,1}^{(\ell,r)}, \ldots, \xi_{j,b_j}^{(\ell,r)}\}$ of size b_j. η is the local learning rate. After τ local updates locally, model at device j and communication round r is indicated by $\boldsymbol{x}_j^{(\tau,r)}$. The next step of our algorithm is that device j sends the count sketch $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j\left(\boldsymbol{x}_j^{(\tau,r)} - \boldsymbol{x}_j^{(0,r)}\right)$ back to the server. We highlight that

$$\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j\left(\boldsymbol{x}_j^{(\tau,r)} - \boldsymbol{x}_j^{(0,r)}\right) = \mathbf{S}_j\left(\eta\sum_{\ell=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(\ell,r)}\right) = \eta\mathbf{S}_j\left(\sum_{\ell=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(\ell,r)}\right),$$

which is the aggregation of the consecutive stochastic gradients multiplied with local updates η.

Upon receiving all $\mathbf{S}_j^{(r)}$ from sampled devices, the server computes

$$\mathbf{S}^{(r)} = \frac{1}{k}\sum_{j\in\mathcal{K}^{(r)}}\mathbf{S}_j^{(r)} \tag{2}$$

and broadcasts it to all devices. Devices after receiving $\mathbf{S}^{(r)}$ from server updates global model $\boldsymbol{x}^{(r)}$ using rule

$$\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\texttt{PRIVIX}\left[\mathbf{S}^{(r-1)}\right].$$

All these steps are summarized in `FedSKETCH` (Algorithm 5). A variant of this algorithm which uses a different compression scheme, called `HEAPRIX` is also described in Algorithm 5. We note that for this variant we need to have an additional communication round between server and worker j to aggregate $\delta_j^{(r)} \triangleq \mathbf{S}_j\left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)})\right]$. Then, server averages all $\delta_j^{(r)}$ and broadcasts to all devices the following quantity:

$$\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{k}\sum_{j\in\mathcal{K}^{(r)}}\delta_j^{(r)}. \tag{3}$$

Upon receiving $\tilde{\mathbf{S}}^{(r)}$ all devices compute

$$\boldsymbol{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r)} - \tilde{\mathbf{S}}^{(r)}\right] \tag{4}$$

and then updates his global model using $\boldsymbol{x}^{(r+!)} = \boldsymbol{x}^{(r)} - \gamma\boldsymbol{\Phi}^{(r)}$.

Remark 5 (Improvement over [9])**.** *An important feature of our algorithm is that due to lower dimension of the count sketch, the resulting averages ($\mathbf{S}^{(r)}$ and $\tilde{\mathbf{S}}^{(r)}$) taken by the server, are also of lower dimension. Therefore, these algorithms exploit bidirectional compression in communication from server to device back and forth. As a result, due to this bidirectional property of communicating sketching for the case of large quantiziation error shown by $q = \theta(\frac{d}{m})$ in [9], our algorithms outperform `FedCOM` and `FedCOMGATE` in [9]. Furthermore, sketching-based server-devices communication algorithm such as ours also provides privacy as a by-product.*

Algorithm 5 `FedSKETCH`(R, τ, η, γ): Private Federated Learning with Sketching.

Inputs: $\boldsymbol{x}^{(0)}$ as an initial model shared by all local devices, the number of communication rounds R, the number of local updates τ, and global and local learning rates γ and η, respectively
for $r = 0, \ldots, R-1$ **do**
  parallel for device $j \in \mathcal{K}^{(r)}$ **do**:
    if PRIVIX variant:
      Computes $\boldsymbol{\Phi}^{(r)} \triangleq \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)}\right]$
    if HEAPRIX variant:
      Computes $\boldsymbol{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \boldsymbol{\Phi}^{(r)}$
    Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    for $\ell = 0, \ldots, \tau - 1$ **do**
      Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\, \tilde{\mathbf{g}}_j^{(\ell,r)}$
    end for
      Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}_j\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$ back to the server.
  Server **computes**
      $\mathbf{S}^{(r)} = \frac{1}{k}\sum_{j \in \mathcal{K}} \mathbf{S}_j^{(r)}$.
  Server samples a subset of devices $\mathcal{K}^{(r)}$ randomly with replacement and **broadcasts** $\mathbf{S}^{(r)}$ to devices in set $\mathcal{K}^{(r)}$.
   if HEAPRIX variant:
    Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left[\texttt{HEAVYMIX}(\mathbf{S}^{(r)})\right]$
    Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{k}\sum_{j \in \mathcal{K}} \delta_j^{(r)}$ to devices in set $\mathcal{K}^{(r)}$
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

5.2 Heterogeneous setting

In this section, we focus on the optimization problem in Eq. (1) in special case of $q_1 = \ldots = q_p = \frac{1}{p}$ with full device participation ($k = p$). We also note that these results can be extended to the scenario where devices are sampled, but for simplicity we do not analyze it in this section. In the previous section, we discussed algorithm `FedSKETCH`, which is originally designed for homogeneous setting where data distribution available at devices are identical. However, in a heterogeneous setting where data distribution could be different, the aforementioned algorithms may fail to perform well in practice. The main reason to cause this issue is that in Federated learning devices are using local stochastic descent direction which could be different than global descent direction when the data distribution are non-identical.

Therefore, to mitigate the effect of data heterogeneity, we introduce new algorithm `FedSKETCHGATE` based on sketching. This algorithm uses the idea of gradient tracking introduced in [9] (with compression) and a variation in [21] (without compression). The main idea is that using an approximation of global gradient, $\mathbf{c}_j^{(r)}$, we correct the local gradient direction. For the `FedSKETCH GATE` with `PRIVIX` variant, the correction vector $\mathbf{c}_j^{(r)}$ at device j and communication round r is computed using the update rule $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\texttt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \texttt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$ where $\mathbf{S}_j^{(r-1)} \triangleq \mathbf{S}\left(\boldsymbol{x}_j^{(0,r-1)} - \boldsymbol{x}_j^{(\tau,r-1)}\right)$ is computed and stored at device j from previous communication round $r-1$. The term $\mathbf{S}^{(r-1)}$ is computed similar to `FedSKETCH` in (2). For `FedSKETCHGATE`, the server needs to compute $\tilde{\mathbf{S}}^{(r)}$ using (3). Then, device j computes $\boldsymbol{\Phi}_j \triangleq \texttt{HEAPRIX}[\mathbf{S}_j^{(r)}]$ and $\boldsymbol{\Phi} \triangleq \texttt{HEAPRIX}(\mathbf{S}^{(r-1)})$ and updates the correction vector $\mathbf{c}_j^{(r)}$ using the recursion $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\boldsymbol{\Phi} - \boldsymbol{\Phi}_j\right)$.

Algorithm 6 `FedSKETCHGATE`(R, τ, η, γ): Private Federated Learning with Sketching and gradient tracking.

Inputs: $\boldsymbol{x}^{(0)} = \boldsymbol{x}_j^{(0)}$ shared by all local devices, communication rounds R, local updates τ, global and local learning rates γ and η.
for $r = 0, \ldots, R-1$ **do**
  parallel for device $j = 1, \ldots, p$ **do**:
    if PRIVIX variant:
      Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\texttt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \texttt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$
      Computes $\mathbf{\Phi}^{(r)} \triangleq \texttt{PRIVIX}(\mathbf{S}^{(r-1)})$
    if HEAPRIX variant:
      Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\mathbf{\Phi}^{(r)} - \mathbf{\Phi}_j^{(r)}\right)$
      Computes $\mathbf{\Phi}^{(r)} \triangleq \texttt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \texttt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
    Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\mathbf{\Phi}^{(r)}$ and $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
    for $\ell = 0, \ldots, \tau-1$ **do**
      Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
      $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\left(\tilde{\mathbf{g}}_j^{(\ell,r)} - \mathbf{c}_j^{(r)}\right)$
    end for
    Device j sends $\mathbf{S}_j^{(r)} \triangleq \mathbf{S}\left(\boldsymbol{x}_j^{(0,r)} - \boldsymbol{x}_j^{(\tau,r)}\right)$ back to the server.
  Server **computes**
    $\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1}\mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
  if HEAPRIX variant:
      Device j computes $\mathbf{\Phi}_j^{(r)} \triangleq \texttt{HEAPRIX}[\mathbf{S}_j^{(r)}]$
      Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left(\texttt{HEAVYMIX}[\mathbf{S}^{(r)}]\right)$
      Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p}\delta_j^{(r)}$ to devices
  end parallel for
end
Output: $\boldsymbol{x}^{(R-1)}$

6 Convergence Analysis

6.1 Assumptions

Assumption 2 (Smoothness and Lower Boundedness). *The local objective function $f_j(\cdot)$ of jth device is differentiable for $j \in [p]$ and L-smooth, i.e., $\|\nabla f_j(\boldsymbol{u}) - \nabla f_j(\mathbf{v})\| \leq L\|\boldsymbol{u} - \mathbf{v}\|$, $\forall\, \boldsymbol{u}, \mathbf{v} \in \mathbb{R}^d$. Moreover, the optimal objective function $f(\cdot)$ is bounded below by $f^* = \min_{\boldsymbol{x}} f(\boldsymbol{x}) > -\infty$.*

Assumption 3 (Polyak-Łojasiewicz). *A function $f(\boldsymbol{x})$ satisfies the Polyak-Łojasiewicz condition with constant μ if $\frac{1}{2}\|\nabla f(\boldsymbol{x})\|_2^2 \geq \mu\big(f(\boldsymbol{x}) - f(\boldsymbol{x}^*)\big)$, $\forall \boldsymbol{x} \in \mathbb{R}^d$ with $\boldsymbol{x}^*$ is an optimal solution.*

6.2 Convergence of `FEDSKETCH` for homogeneous setting.

Now we focus on the homogeneous case in which the stochastic local gradient of each worker is an unbiased estimator of the global gradient.

Assumption 4 (Bounded Variance). *For all $j \in [m]$, we can sample an independent mini-batch ℓ_j of size $|\Xi_j^{(\ell,r)}| = b$ and compute an unbiased stochastic gradient $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w}; \Xi_j), \mathbb{E}_{\xi_j}[\tilde{\mathbf{g}}_j] = \nabla f(\boldsymbol{w}) = \mathbf{g}$ with the variance bounded is bounded by a constant σ^2, i.e., $\mathbb{E}_{\Xi_j}\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}\|^2\right] \leq \sigma^2$.*

Theorem 2. *Suppose that the conditions in Assumptions 2-4 hold. Given $0 < m = O\left(\frac{e}{\mu^2}\right) \leq d$, and Consider `FedSKETCH` in Algorithm 5 with sketch size $B = O\left(m \log\left(\frac{dR}{\delta}\right)\right)$. If the local data distributions of all users are identical (homogeneous setting), then with probability $1-\delta$ we have*

- ***Nonconvex:***
  1) *For the* `FedSKETCH-PRIVIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{L\gamma}\sqrt{\frac{k}{R\tau\left(\frac{\mu^2 d}{k}+1\right)}}$ *and* $\gamma\geq k$*, the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2\leq\epsilon$ *if we set* $R=O\left(\frac{1}{\epsilon}\right)$ *and* $\tau=O\left(\frac{\mu^2 d+1}{k\epsilon}\right)$.
  2) *For* `FedSKETCH-HEAPRIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{L\gamma}\sqrt{\frac{k}{R\tau\left(\frac{\mu^2 d-1}{k}+1\right)}}$ *and* $\gamma\geq k$*, the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2\leq\epsilon$ *if we set* $R=O\left(\frac{1}{\epsilon}\right)$ *and* $\tau=O\left(\frac{\mu^2 d}{k\epsilon}\right)$.
- ***PL or Strongly convex:***
  1) *For* `FedSKETCH-PRIVIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L\left(\frac{\mu^2 d}{k}+1\right)\tau\gamma}$ *and* $\gamma\geq k$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\left(\frac{\mu^2 d}{k}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{\mu^2 d+1}{k\left(\frac{\mu^2 d}{k}+1\right)\epsilon}\right)$.
  2) *For* `FedSKETCH-HEAPRIX` *algorithm by choosing stepsizes as* $\eta=\frac{1}{2L\left(\frac{\mu^2 d-1}{k}+1\right)\tau\gamma}$ *and* $\gamma\geq k$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\left(\frac{\mu^2 d-1}{k}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{\mu^2 d}{k\left(\frac{\mu^2 d-1}{k}+1\right)\epsilon}\right)$.
- ***Convex:***
  1) *For the* `FedSKETCH-PRIVIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L\left(\frac{\mu^2 d}{k}+1\right)\tau\gamma}$ *and* $\gamma\geq k$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\frac{L\left(1+\frac{\mu^2 d}{k}\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{\left(\mu^2 d+1\right)^2}{k\left(\frac{\mu^2 d}{k}+1\right)^2\epsilon^2}\right)$.
  2) *For the* `FedSKETCH-HEAPRIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L\left(\frac{\mu^2 d-1}{k}+1\right)\tau\gamma}$ *and* $\gamma\geq k$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\frac{L\left(\frac{\mu^2 d-1}{k}+1\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{\left(\mu^2 d\right)^2}{k\left(\frac{\mu^2 d-1}{k}+1\right)^2\epsilon^2}\right)$.

Corollary 2 (Total communication cost)**.** *As a consequence of Remark 7, the total communication cost per-worker becomes*

$$O\left(RB\right)=O\left(Rm\log\left(\frac{dR}{\delta}\right)\right)=O\left(\frac{m}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)\tag{5}$$

We note that this result in addition to improving over the communication complexity of federated learning of the state-of-the-art from $O\left(\frac{d}{\epsilon}\right)$ *in [14, 34, 21] to* $O\left(\frac{mk}{\epsilon}\log\left(\frac{dk}{\epsilon\delta}\right)\right)$*, it also implies differential privacy. As a result, total communication cost is*

$$BkR=O\left(\frac{mk}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right).$$

We note that the state-of-the-art in [14] the total communication cost is

$$BkR=O\left(kd\left(\frac{1}{\epsilon}\right)\frac{P^{2/3}}{k^{2/3}}\right)=O\left(\frac{kd}{\epsilon}\frac{P^{2/3}}{k^{2/3}}\right)\tag{6}$$

We improve this result, in terms of dependency to d, to

$$BkR = O\left(\frac{mk}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right) \tag{7}$$

In comparison to [13], we improve the total communication per worker from $RB = O\left(\frac{m}{\epsilon^2}\log\left(\frac{d}{\epsilon^2\delta}\right)\right)$ *to* $RB = O\left(\frac{m}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)$.

Remark 6. *It is worthy to note that most of the available communication-efficient algorithm with quantization or compression only consider communication-efficiency from devices to server. However, Algorithm 5 also improves the communication efficiency from server to devices as well.*

Corollary 3 (Total communication cost for PL or strongly convex)**.** *To achieve the convergence error of* ϵ, *we need to have* $R = O\left(\kappa(\frac{\mu^2 d}{k}+1)\log\frac{1}{\epsilon}\right)$ *and* $\tau = \left(\frac{(\mu^2 d+1)}{(\frac{\mu^2 d}{k}+1)k\epsilon}\right)$. *This leads to the total communication cost per worker of*

$$BR = O\left(m\kappa(\frac{\mu^2 d}{k}+1)\log\left(\frac{\kappa(\frac{\mu^2 d^2}{k}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right) \tag{8}$$

As a consequence, the total communication cost becomes:

$$BkR = O\left(m\kappa(\mu^2 d+k)\log\left(\frac{\kappa(\frac{\mu^2 d^2}{k}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right) \tag{9}$$

We note that the state-of-the-art in [14] the total communication cost is

$$BkR = O\left(\kappa kd\log\left(\frac{1}{\epsilon}\right)\right) = O\left(\kappa kd\log\left(\frac{1}{\epsilon}\right)\right) \tag{10}$$

We improve this result, in terms of dependency to d, to

$$BkR = O\left(m\kappa(\mu^2 d+k)\log\left(\frac{\kappa(\frac{\mu^2 d}{k}+d)\log\frac{1}{\epsilon}}{\delta}\right)\log\frac{1}{\epsilon}\right) \tag{11}$$

Improving from kd *to* $k+d$.

6.3 Convergence of FedSKETCHGATE in data heterogeneous setting.

Assumption 5 (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* $\tilde{\mathbf{g}}_j = \nabla f_j(\boldsymbol{w};\Xi_j), \mathbb{E}_\xi[\tilde{\mathbf{g}}_j] = \nabla f_j(\boldsymbol{w}) = \mathbf{g}_j$. *Moreover, the variance of local stochastic gradients is bounded above by a constant* σ^2, *i.e.,* $\mathbb{E}_\Xi\left[\|\tilde{\mathbf{g}}_j - \mathbf{g}_j\|^2\right] \le \sigma^2$.

Theorem 3. *Suppose that the conditions in Assumptions 2 and 5 hold. Given* $0 < m = O\left(\frac{e}{\mu^2}\right) \le d$, *and Consider* ***FedSKETCHGATE*** *in Algorithm 6 with sketch size* $B = O\left(m\log\left(\frac{dR}{\delta}\right)\right)$. *If the local data distributions of all users are identical (homogeneous setting), then with probability* $1-\delta$ *we have*

- ***Nonconvex:***
  1) *For the* ***FedSKETCHGATE-PRIVIX*** *algorithm, by choosing stepsizes as* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\mu^2 d)}}$ *and* $\gamma \ge p$, *the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \le \epsilon$ *if we set* $R = O\left(\frac{\mu^2 d+1}{\epsilon}\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.
  2) *For* ***FedSKETCHGATE-HEAPRIX*** *algorithm, by choosing stepsizes as* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\mu^2 d)}}$ *and* $\gamma \ge p$, *the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \le \epsilon$ *if we set* $R = O\left(\frac{\mu^2 d}{\epsilon}\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.

	Objective function			
Reference	Nonconvex	PL/Strongly Convex	UG	PP
[13]	–	$R=O\left(\frac{\mu^2 d}{\epsilon}\right)$ $\tau=1$ $B=O\left(m\log\left(\frac{dR}{\delta}\right)\right)$ $pRB=O\left(\frac{p\mu^2 d}{\epsilon}m\log\left(\frac{\mu^2 d^2}{\epsilon\delta}\right)\right)$	✗	✗
Theorem 2	$\boldsymbol{R=O\left(\frac{1}{\epsilon}\right)}$ $\boldsymbol{\tau=O\left(\frac{\mu^2 d+1}{k\epsilon}\right)}$ $\boldsymbol{B=O\left(m\log\left(\frac{dR}{\delta}\right)\right)}$ $\boldsymbol{kBR=O\left(\frac{mk}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)}$	$\boldsymbol{R=O\left(\kappa\left(\frac{\mu^2 d}{k}+1\right)\log\left(\frac{1}{\epsilon}\right)\right)}$ $\boldsymbol{\tau=O\left(\frac{(\mu^2 d+1)}{k\left(\frac{\mu^2 d}{k}+1\right)\epsilon}\right)}$ $\boldsymbol{B=O\left(m\log\left(\frac{dR}{\delta}\right)\right)}$ $\boldsymbol{kBR=O\left(m\kappa(\mu^2 d+k)\log\frac{1}{\epsilon}\log\left(\frac{\kappa(\frac{\mu^2 d^2}{k}+d)\log\frac{1}{\epsilon}}{\delta}\right)\right)}$	✔	✔
Theorem 2	$\boldsymbol{R=O\left(\frac{1}{\epsilon}\right)}$ $\boldsymbol{\tau=O\left(\frac{\mu^2 d}{k\epsilon}\right)}$ $\boldsymbol{B=O\left(m\log\left(\frac{dR}{\delta}\right)\right)}$ $\boldsymbol{kBR=O\left(\frac{mk}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right)}$	$\boldsymbol{R=O\left(\kappa\left(\frac{\mu^2 d-1}{k}+1\right)\log\left(\frac{1}{\epsilon}\right)\right)}$ $\boldsymbol{\tau=O\left(\frac{(\mu^2 d)}{k\left(\frac{\mu^2 d}{k}+1\right)\epsilon}\right)}$ $\boldsymbol{B=O\left(m\log\left(\frac{dR}{\delta}\right)\right)}$ $\boldsymbol{kBR=O\left(m\kappa(\mu^2 d-1+k)\log\frac{1}{\epsilon}\log\left(\frac{\kappa(d\frac{\mu^2 d-1}{k}+d)\log\frac{1}{\epsilon}}{\delta}\right)\right)}$	✔	✔

Table 1 Comparison of results with compression and periodic averaging in the homogeneous setting. Here, m is the number of devices, μ is compression of hash table, d is the dimension of the model, κ is condition number, ϵ is 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 properly.

- ***PL or Strongly convex:***
  1) *For the* `FedSKETCHGATE-PRIVIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L(\mu^2 d+1)\tau\gamma}$ *and* $\gamma\geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\left(\mu^2 d+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{1}{p\epsilon}\right)$.
  2) *For the case of* `FedSKETCHGATE-HEAPRIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L(\mu^2 d)\tau\gamma}$ *and* $\gamma\geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\left(\mu^2 d\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{1}{p\epsilon}\right)$.
- ***Convex:***
  1) *For the* `FedSKETCHGATE-PRIVIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L(\mu^2 d+1)\tau\gamma}$ *and* $\gamma\geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\frac{L\left(1+\mu^2 d\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{1}{p\epsilon^2}\right)$.
  2) *For the* `FedSKETCHGATE-HEAPRIX` *algorithm, by choosing stepsizes as* $\eta=\frac{1}{2L(\mu^2 d)\tau\gamma}$ *and* $\gamma\geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)})-f(\boldsymbol{w}^{(*)})\Big]\leq\epsilon$ *if we set* $R=O\left(\frac{L\left(\mu^2 d\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau=O\left(\frac{1}{p\epsilon^2}\right)$.

Comparison with [18], [29] and [25]:

Comparison to[18]. We note that our convergence analysis does not rely on bounded gradient assumption, it can be seen that we both improve the number of communication rounds R and the size of vector B per communication round while preserving privacy property. Additionally, we highlight that our while [18] provides convergence result for convex objectives, we provide the convergence analysis for PL (thus strongly convex case), general convex and general non-convex objectives.

	Objective function			
Reference	Nonconvex	General Convex	UG	PP
[18]	–	$R = O\left(\frac{\mu^2 d}{\epsilon^2}\right)$ $\tau = 1$ $B = O\left(m \log\left(\frac{\mu^2 d^2}{\epsilon^2 \delta}\right)\right)$	✗	✔
[29]	$R = O\left(\max(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2 \epsilon})\right)$ $\tau = 1$ $B = O\left(m \log\left(\frac{d}{\epsilon^2 \delta}\right)\right)$ $BR = O\left(\frac{m}{\epsilon^2} \max(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2 \epsilon}) \log\left(\frac{d}{\delta} \max(\frac{1}{\epsilon^2}, \frac{d^2 - md}{m^2 \epsilon})\right)\right)$	–	✗	✗
[29]	$R = O\left(\frac{\max(I^{2/3}, 2-\alpha)}{\epsilon^3}\right)$ $\tau = 1$ $B = O\left(\frac{m}{\alpha} \log\left(\frac{d \max(I^{2/3}, 2-\alpha)}{\epsilon^3 \delta}\right)\right)$ $BR = O\left(\frac{m \max(I^{2/3}, 2-\alpha)}{\epsilon^3 \alpha} \log\left(\frac{d \max(I^{2/3}, 2-\alpha)}{\epsilon^3 \delta}\right)\right)$	–	✗	✗
Theorem 3	$\boldsymbol{R = O\left(\frac{\mu^2 d + 1}{\epsilon}\right)}$ $\boldsymbol{\tau = O\left(\frac{1}{p\epsilon}\right)}$ $\boldsymbol{B = O\left(m \log\left(\frac{\mu^2 d^2 + d}{\epsilon \delta}\right)\right)}$ $\boldsymbol{BR = O\left(\frac{m(\mu^2 d + 1)}{\epsilon} \log\left(\frac{\mu^2 d^2 + d}{\epsilon \delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)}$	$\boldsymbol{R = O\left(\frac{1 + \mu^2 d}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right)}$ $\boldsymbol{\tau = O\left(\frac{1}{p\epsilon^2}\right)}$ $\boldsymbol{B = O\left(m \log\left(\frac{\mu^2 d^2 + d}{\epsilon \delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)}$	✔	✔
Theorem 3	$\boldsymbol{R = O\left(\frac{\mu^2 d}{\epsilon}\right)}$ $\boldsymbol{\tau = O\left(\frac{1}{p\epsilon}\right)}$ $\boldsymbol{B = O\left(m \log\left(\frac{\mu^2 d^2}{\epsilon \delta}\right)\right)}$ $\boldsymbol{BR = O\left(\frac{m(\mu^2 d)}{\epsilon} \log\left(\frac{\mu^2 d^2}{\epsilon \delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)}$	$\boldsymbol{R = O\left(\frac{\mu^2 d}{\epsilon} \log\left(\frac{1}{\epsilon}\right)\right)}$ $\boldsymbol{\tau = O\left(\frac{1}{p\epsilon^2}\right)}$ $\boldsymbol{B = O\left(m \log\left(\frac{\mu^2 d^2}{\epsilon \delta}\right)\right)}$	✔	✔

Table 2 Comparison of results with compression and periodic averaging in the heterogeneous setting. Here, p is the number of devices, μ is compression of hash table, d is the dimension of the model, κ is condition number, ϵ is 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 properly respectively.

Comparison with [29]. Consider two versions of `FetchSGD` in this reference. First while in our schemes we do not to have access to the exact entries of gradients, since the approaches in [29] is based on top_m queries, both of the proposed algorithms (in [29]) require to have access to the exact value of top_k gradients, hence they do not preserve privacy. Second, both of the convergence results in [29] rely on the bounded gradient assumption and it is known that this assumption is not in consistent with L-smoothness when data distribution is heterogeneous which is the case in Federated Learning (see [3] for more detail). However, our convergence results do not need any bounded gradient assumption. Third, Theorem 1 [29] is based on an Assumption that *Contraction Holds* for the sequence of gradients encountered during the optimization which may not hold necessarily in practice, yet based on this strong assumption their total communication cost (RB) to achieve ϵ error is $BR = O\left(m \max(\frac{1}{\epsilon^2}, \frac{d^2 - dm}{m^2 \epsilon}) \log\left(\frac{d}{\delta} \max(\frac{1}{\epsilon^2}, \frac{d^2 - dm}{m^2 \epsilon})\right)\right)$ (Note for the sake of comparison we let the compression ration in [29] to be $\frac{m}{d}$). In contrast, without any extra assumptions, our results in Theorem 3 for `PRIVIX` and `HEAPRIX` are respectively $BR = O\left(\frac{m\left(\mu^2 d + 1\right)}{\epsilon} \log\left(\frac{\mu^2 d^2 + d}{\epsilon \delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)$ and $BR = O\left(\frac{m\left(\mu^2 d\right)}{\epsilon} \log\left(\frac{\mu^2 d^2}{\epsilon \delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)$ which improves total communication cost in Theorem 1 in [29] in regimes where $\frac{1}{\epsilon} \geq d$ or $d >> m$. Theorem 2 in [29] is based on another assumption of Sliding Window Heavy Hitters, which is similar to gradient diversity assumption in [19, 10] (but it is weaker assumption of contraction in Theorem 1 in [29]), and they showed that the total communication cost is $BR = O\left(\frac{m \max(I^{2/3}, 2-\alpha)}{\epsilon^3 \alpha} \log\left(\frac{d \max(I^{2/3}, 2-\alpha)}{\epsilon^3 \delta}\right)\right)$ (I is constant comes from the extra assumption over the window of gradients which similar to bounded gradient diversity) which is again worse than obtained result in this paper with weaker assumptions in a regime where $\frac{I^{2/3}}{\epsilon^2} \geq d$. Next, unlike [29] which only focuses on non-convex objectives, in this work we provide the convergence analysis for PL (thus strongly convex case), general convex and general non-convex objectives. Finally, although the algorithm in [29] requires additional memory for the server to store the compression error correction vector, our algorithm does not need such additional storage.

Comparison with [25]. The reference [25] considers two-way compression from parameter server to devices and vice versa. They provide the convergence rate of $R = O\left(\frac{\omega^{\mathrm{Up}}\omega^{\mathrm{Down}}}{\epsilon^2}\right)$ for strongly-objective functions where ω^{Up} and ω^{Down} are uplink and downlink's compression noise (specilalizing to our case for the sake of comparision $\omega^{\mathrm{Up}} = \omega^{\mathrm{Down}} = \theta\left(d\right)$) for general heterogeneous data distribution. In contrast, while as pointed out in Remark 5 that our algorithms are using bidirectional compression due to use of sketching for communication, our convergence rate for strongly-convex objective is $R = O(\kappa\mu^2 d \log\left(\frac{1}{\epsilon}\right))$ with probability $1-\delta$.

7 Experiments

In this section, we provide empirical results on MNIST dataset to demonstrate the effectiveness of our proposed algorithms. The model we use is the LeNet-5 Convolutional Neural Network (CNN) architecture introduced in [17], with 60 000 model parameters in total.

Four methods are compared in our experiments: Federated SGD (FedSGD), SketchSGD [13], FedSketch-PRIVIX (FS-PRIVIX) and FedSketch-HEAPRIX (FS-HEAPRIX). We implement the algorithms by simulating the distributed and federated environment. Note that in Algorithm 5, FS-PRIVIX with global learning rate $\gamma = 1$ is equivalent to the DiffSketch algorithm proposed in [19]. In the following experiments, we set the number of workers to 50. For federated learning algorithms, we use different number of local updates τ. For SketchedSGD which is under synchronous distributed learning framework, τ is fixed and equal to 1. For all methods, we tune the learning rates (both local, i.e. η and global, i.e. γ, if applicable) over the log-scale and report the best results.

In each round of local update, we randomly choose half of the local devices to be active, which is the common practice in real-world applications. For the data distribution on each device, we test both *homogeneous* and *heterogeneous* setting. In the former case, each device receives uniformly drawn data samples (each class has equal probability to be selected). In the latter case, each device only receives samples from one or two classes among ten digits in the MNIST dataset. Since data is not distributed i.i.d. among local devices, training is expected to be harder in the heterogeneous case.

Homogeneous case. In Figure 1, we provide the training loss and test accuracy for the four algorithms mentioned above, with $\tau = 1$ (since SketchSGD requires single local update per round). We also test different sizes of sketching matrix, $(t, k) = (20, 40)$ and $(50, 100)$. Note that these two choices of sketch size correspond to a $75\times$ and $12\times$ compression ratio, respectively. In general, as one would expect, higher compression ratio leads to worse learning performance. In both cases, FS-HEAPRIX performs the best in terms of both training objective and test accuracy. FS-PRIVIX is better when sketch size is large (i.e. when the estimation from sketches are more accurate), while SketchSGD performs better with small sketch size.

The results for multiple local updates are given in Figure 2, where we set $\tau = 2, 5$. We see that FS-HEAPRIX is significantly better than FS-PRIVIX, either with small or large sketching matrix. In both cases, FS-HEAPRIX yields acceptable extra test error compared to FedSGD, especially when considering the high compression ratio (e.g. $75\times$). However, FS-PRIVIX performs poorly with small sketch size $(20, 40)$, and even diverges with $\tau = 5$. We also observe that the performances of FS-HEAPRIX improve 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 claims established in this paper. For $\tau = 1, 2, 5$, we see that a sketch size of $(50, 100)$ is sufficient to give similar test accuracy as the federated SGD (FedSGD).

Heterogeneous case. We plot similar sets of results in Figure 3 and Figure 4 for non-i.i.d. data distribution (heterogeneous setting). This setting leads to more twists and turns in the training curves. From Figure 3 ($\tau = 1$), we see that SketchSGD performs very poorly in the heterogeneous case, while both our proposed FedSketchGATE methods, see Algorithm 6, achieve similar generalization accuracy as the federated SGD (FedSGD) algorithm, even with fairly small sketch size (i.e. $75\times$ compression ratio). In addition, FS-HEAPRIX is again better than FS-PRIVIX in terms of both training loss and test accuracy.

Furthermore, we notice in Figure 4 the advantage of FS-HEAPRIX over FS-PRIVIX. However, empirically we see that in the heterogeneous setting, more local updates τ tend to undermine the learning performance, especially with small sketch size. This is because in this scenario, each local device only receives samples with a

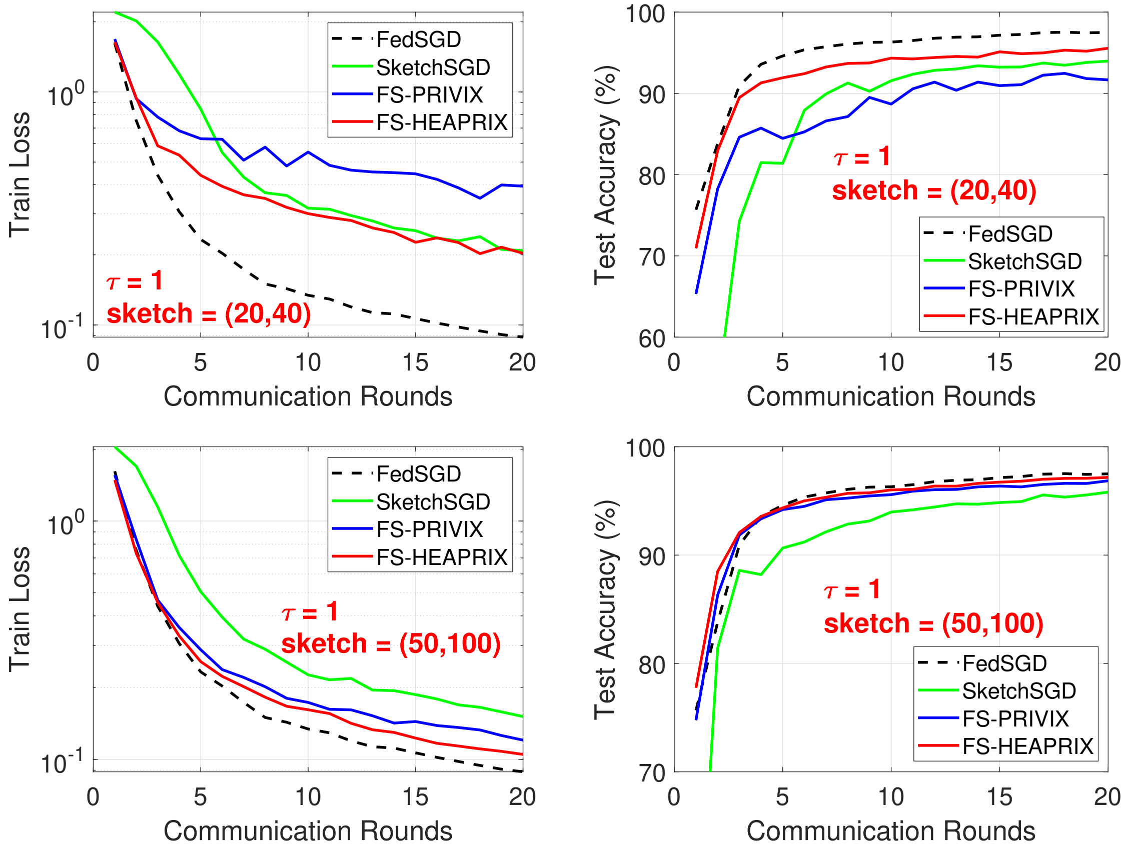

Figure 1 Homogeneous case: Comparison of four algorithms on LeNet CNN architecture.

few classes, so each local model is actually trained with biased stochastic gradients. More local updates would result in more bias in the local models, which makes federated averaging less effective. Nevertheless, we see that when sketch size is large, i.e. $(50, 100)$, FS-HEAPRIX can still provide comparable test accuracy as FedSGD.

Our empirical study demonstrates that our proposed FedSketch (and FedSketchGATE) frameworks are able to perform well in homogeneous (resp. heterogeneous) learning setting, with high compression rate. In particular, FedSketch methods are advantageous over prior SketchedSGD [13] method in both cases. FS-HEAPRIX performs the best among all the tested compressed optimization algorithms, which in many cases achieves similar generalization accuracy as federated SGD with small sketch size. In general, in any tested case, we can at least achieve $12\times$ compression ratio with very little loss in test accuracy.

8 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. We analyze the convergence error for *non-convex*, *Polyak-Łojasiewicz* 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.

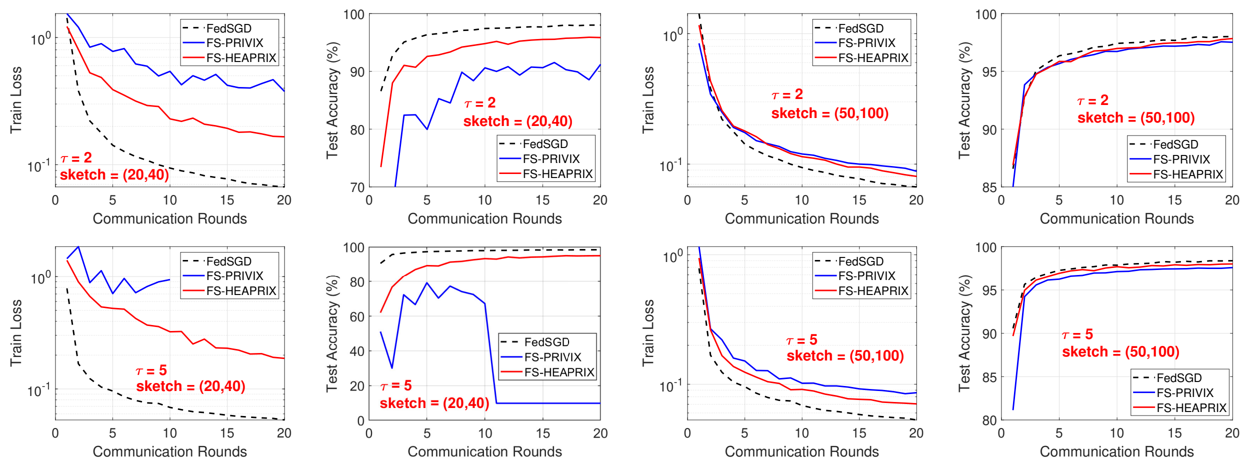

Figure 2 Homogeneous case: comparison of FedSGD, FS-PRIVIX and FS-HEAPRIX on LeNet CNN architecture, with number of local updates being 2 and 5.

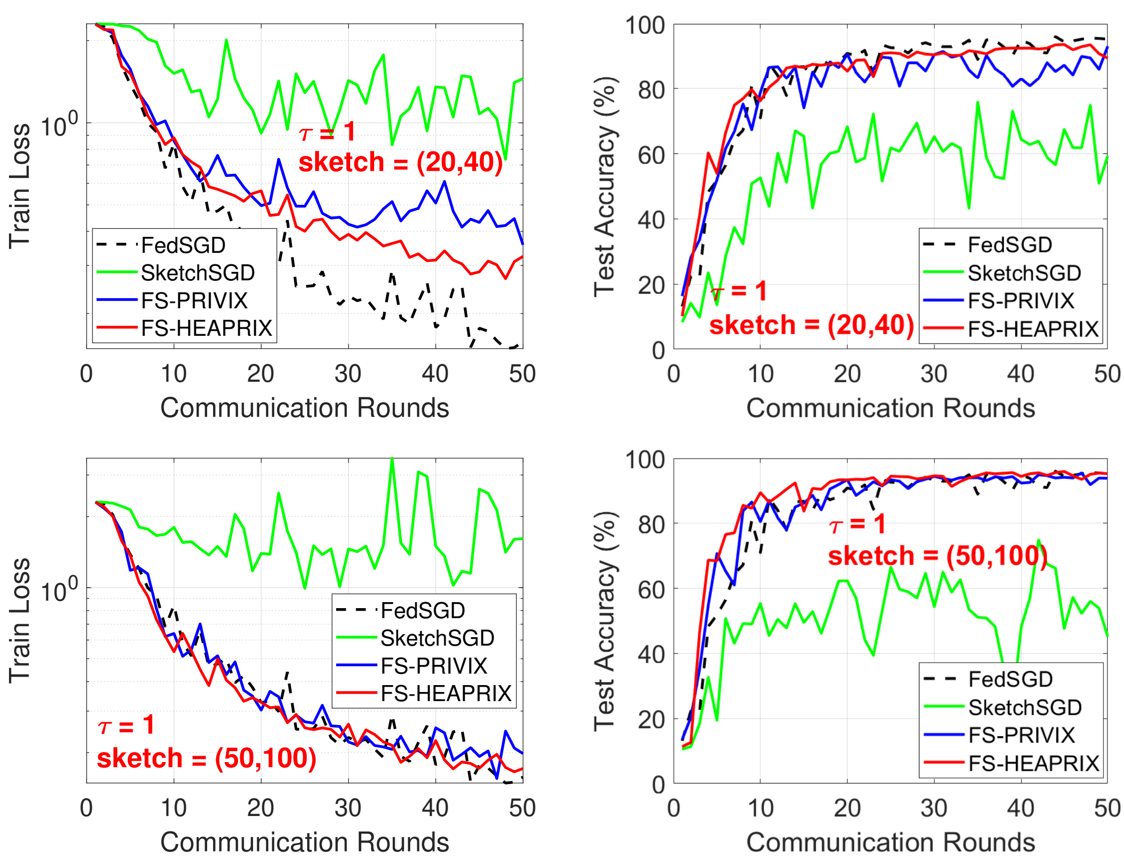

Figure 3 Heterogeneous case: the comparison of four algorithms on LeNet CNN architecture.

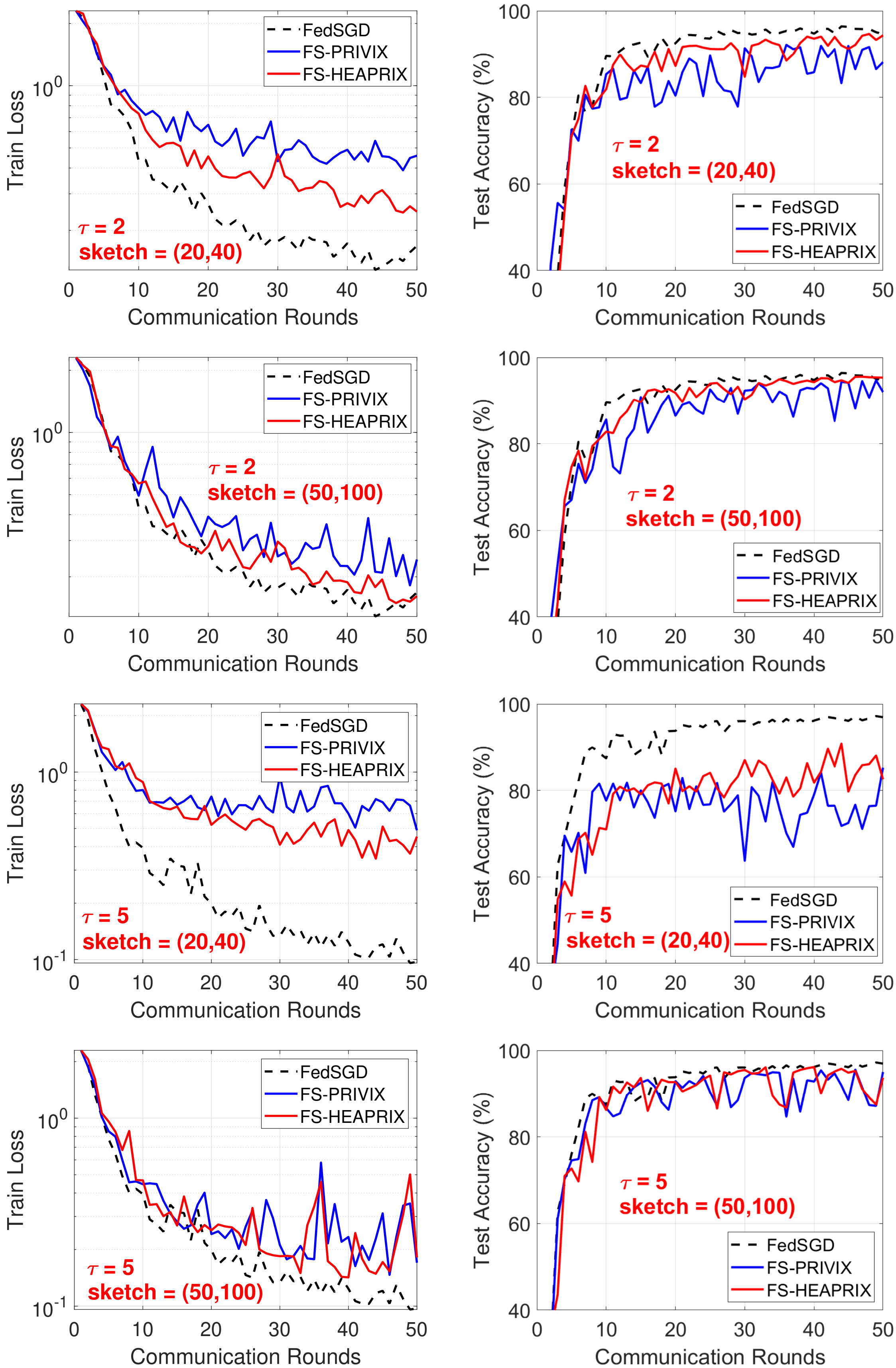

Figure 4 Heterogeneous case comparison of FedSGD, FS-PRIVIX and FS-HEAPRIX on LeNet CNN architecture, with number of local updates being 2 and 5.

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, pages 14695–14706, 2019.

[3] Ahmed Khaled Ragab Bayoumi, Konstantin Mishchenko, and Peter Richtarik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529, 2020.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

[5] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[6] Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

[7] Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[8] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization. In *ICML*, pages 2545–2554, 2019.

[9] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. *arXiv preprint arXiv:2007.01154*, 2020.

[10] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[11] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[12] Samuel Horváth and Peter Richtárik. A better alternative to error feedback for communication-efficient distributed learning. *arXiv preprint arXiv:2006.11077*, 2020.

[13] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Ion Stoica, Raman Arora, et al. Communication-efficient distributed sgd with sketching. In *Advances in Neural Information Processing Systems*, pages 13144–13154, 2019.

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[15] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

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

[18] Tian Li, Zaoxing Liu, Vyas Sekar, and Virginia Smith. Privacy for free: Communication-efficient learning with differential privacy using sketches. *arXiv preprint arXiv:1911.00972*, 2019.

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[20] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[21] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[22] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

[23] Zaoxing Liu, Tian Li, Virginia Smith, and Vyas Sekar. Enhancing the privacy of federated learning with sketching. *arXiv preprint arXiv:1911.01812*, 2019.

[24] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[25] Constantin Philippenko and Aymeric Dieuleveut. Artemis: tight convergence guarantees for bidirectional compression in federated learning. *arXiv preprint arXiv:2006.14591*, 2020.

[26] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[27] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. *arXiv preprint arXiv:1909.13014*, 2019.

[28] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[29] Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching, 2020.

[30] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

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

[32] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

[33] Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

[34] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[35] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages 1299–1309, 2018.

[36] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817*, 2019.

[37] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2018.

[38] Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Parallel sgd: When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016.

[39] Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017.

A Appendix

Notation. Here we indicate the count sketch of the vector $\boldsymbol{x}$ with $\mathbf{S}(\boldsymbol{x})$ and with abuse of notation we indicate the expectation over the randomness of count sketch with $\mathbb{E}_{\mathbf{S}}[.]$. We illustrate the the random subset of the devices selected by server with $\mathcal{K}$ with size $|\mathcal{K}| = k \leq p$, and we represent the expectation over the device sampling with $\mathbb{E}_{\mathcal{K}}[.]$.

We will use the following fact (which is also used in [20, 10]) in proving results.

Fact 4 ([20, 10])**.** *Let $\{x_i\}_{i=1}^p$ denote any fixed deterministic sequence. We sample a multiset $\mathcal{P}$ (with size K) uniformly at random where x_j is sampled with probability q_j for $1 \leq j \leq p$ with replacement. Let $\mathcal{P} = \{i_1, \ldots, i_K\} \subset [p]$ (some i_j's may have the same value). Then*

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{i\in\mathcal{P}} x_i\right] = \mathbb{E}_{\mathcal{P}}\left[\sum_{k=1}^{K} x_{i_k}\right] = K\mathbb{E}_{\mathcal{P}}\left[x_{i_k}\right] = K\left[\sum_{j=1}^{p} q_j x_j\right] \tag{12}$$

B Results for the Homogeneous Setting

In this section, we study the convergence properties of our `FedSKETCH` method presented in Algorithm 5. Before stating the proofs for `FedSKETCH` in the homogeneous setting, we first mention the following intermediate lemmas.

Lemma 4. *Using unbiased compression and under Assumption 4, we have the following bound:*

$$\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S},\xi^{(r)}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]\right] = \mathbb{E}_{\xi^{(r)}}\mathbb{E}_{\mathbf{S}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right] \leq \tau(\frac{\omega}{k}+1)\sum_{j=1}^{m} q_j\left[\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2 + \sigma^2\right] \tag{13}$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{\xi^{(r)}|\boldsymbol{w}^{(r)}}\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\|^2\right]\right]\\
&= \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\underbrace{\mathbf{S}\left(\overbrace{\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}}^{\tilde{\mathbf{g}}_j^{(r)}}\right)}_{\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}}\|^2\right]\right]\right]\\
&\overset{①}{=} \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)} - \frac{1}{k}\sum_{j\in\mathcal{K}}\mathbb{E}_{\mathbf{S}}\left[\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right]\|^2\right] + \|\mathbb{E}_{\mathbf{S}}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S},j}^{(r)}\right]\|^2\right]\right]\\
&\overset{②}{=} \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\left[\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)} - \sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\right]\|^2\right] + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]\\
&= \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\left[\mathrm{Var}_{\mathbf{S}}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right]\right] + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]\\
&= \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\mathrm{Var}_{\mathbf{S}_j}\left[\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right] + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]\\
&\leq \mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\omega\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2 + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]
\end{aligned}$$

$$
\begin{aligned}
&= \left[\mathbb{E}_{\xi}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\omega\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2 + \mathbb{E}_{\mathcal{K}}\mathbb{E}_{\xi^{(r)}}\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right] \\
&= \left[\mathbb{E}_{\xi}\left[\frac{\omega}{k}\sum_{j=1}^{p}q_j\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2 + \mathbb{E}_{\mathcal{K}}\left[\mathrm{Var}\left(\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\right) + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{g}_j^{(r)}\|^2\right]\right]\right] \\
&= \frac{\omega}{k}\sum_{j=1}^{p}q_j\mathbb{E}_{\xi}\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2 + \mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\mathrm{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right) + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{g}_j^{(r)}\|^2\right] \\
&\leq \frac{\omega}{k}\sum_{j=1}^{p}q_j\mathbb{E}_{\xi}\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2 + \mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\tau\sigma^2 + \frac{1}{k}\sum_{j\in\mathcal{K}}\|\mathbf{g}_j^{(r)}\|^2\right] \\
&= \frac{\omega}{k}\sum_{j=1}^{p}q_j\left[\mathrm{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right) + \left\|\mathbf{g}_j^{(r)}\right\|^2\right] + \left[\frac{\tau\sigma^2}{k} + \sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right] \\
&\leq \frac{\omega}{k}\sum_{j=1}^{p}q_j\left[\tau\sigma^2 + \left\|\mathbf{g}_j^{(r)}\right\|^2\right] + \left[\frac{\tau\sigma^2}{k} + \sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right] \\
&= (\omega+1)\frac{\tau\sigma^2}{k} + (\frac{\omega}{k}+1)\left[\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right] \qquad (14)
\end{aligned}
$$

where ① holds due to $\mathbb{E}\left[\|\mathbf{x}\|^2\right] = \mathrm{Var}[\mathbf{x}] + \|\mathbb{E}[\mathbf{x}]\|^2$, ② is due to $\mathbb{E}_{\mathbf{S}}\left[\frac{1}{p}\sum_{j=1}^{p}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right] = \frac{1}{p}\sum_{j=1}^{m}\tilde{\mathbf{g}}_j^{(r)}$.

Next we show that from Assumptions 5, we have

$$
\mathbb{E}_{\xi^{(r)}}\left[\left[\|\tilde{\mathbf{g}}_j^{(r)} - \mathbf{g}_j^{(r)}\|^2\right]\right] \leq \tau\sigma^2 \qquad (15)
$$

To do so, note that

$$
\begin{aligned}
\mathrm{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right) &= \mathbb{E}_{\xi^{(r)}}\left[\left\|\tilde{\mathbf{g}}_j^{(r)} - \mathbf{g}_j^{(r)}\right\|^2\right] \\
&\overset{①}{=} \mathbb{E}_{\xi^{(r)}}\left[\left\|\sum_{c=0}^{\tau-1}\left[\tilde{\mathbf{g}}_j^{(c,r)} - \mathbf{g}_j^{(c,r)}\right]\right\|^2\right] \\
&= \mathrm{Var}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right) \\
&\overset{②}{=} \sum_{c=0}^{\tau-1}\mathrm{Var}\left(\tilde{\mathbf{g}}_j^{(c,r)}\right) \\
&= \sum_{c=0}^{\tau-1}\mathbb{E}\left[\left\|\tilde{\mathbf{g}}_j^{(c,r)} - \mathbf{g}_j^{(c,r)}\right\|^2\right] \\
&\overset{③}{\leq} \tau\sigma^2 \qquad (16)
\end{aligned}
$$

where in ① we use the definition of $\tilde{\mathbf{g}}_j^{(r)}$ and $\mathbf{g}_j^{(r)}$, in ② we use the fact that mini-batches are chosen in i.i.d. manner at each local machine, and ③ immediately follows from Assumptions 4.

Replacing $\mathbb{E}_{\xi^{(r)}}\left[\|\tilde{\mathbf{g}}_j^{(r)} - \mathbf{g}_j^{(r)}\|^2\right]$ in (14) by its upper bound in (15) implies that

$$
\mathbb{E}_{\xi^{(r)}|\boldsymbol{w}^{(r)}}\mathbb{E}_{\mathbf{S},\mathcal{K}}\Big[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\|^2\Big] \leq (\omega+1)\frac{\tau\sigma^2}{k} + (\frac{\omega}{k}+1)\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2 \qquad (17)
$$

Further note that we have

$$\left\|\mathbf{g}_j^{(r)}\right\|^2 = \|\sum_{c=0}^{\tau-1}\mathbf{g}_j^{(c,r)}\|^2 \le \tau\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2 \tag{18}$$

where the last inequality is due to $\left\|\sum_{j=1}^{n}\mathbf{a}_i\right\|^2 \le n\sum_{j=1}^{n}\|\mathbf{a}_i\|^2$, which together with (17) leads to the following bound:

$$\mathbb{E}_{\xi^{(r)}|\boldsymbol{w}^{(r)}}\mathbb{E}_{\mathbf{S}}\Big[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\|^2\Big] \le (\omega+1)\frac{\tau\sigma^2}{k} + \tau(\frac{\omega}{k}+1)\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(c,r)}\|^2, \tag{19}$$

and the proof is complete. □

Lemma 5. *Under Assumption 2, and according to the* `FedCOM` *algorithm the expected inner product between stochastic gradient and full batch gradient can be bounded with:*

$$-\mathbb{E}_{\xi,\mathbf{S},\mathcal{K}}\left[\left\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}^{(r)}\right\rangle\right] \le \frac{1}{2}\eta\frac{1}{m}\sum_{j=1}^{m}\sum_{c=0}^{\tau-1}\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2 - \|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2 + L^2\|\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(c,r)}\|_2^2\right] \tag{20}$$

Proof. We have:

$$\begin{aligned}
&-\mathbb{E}_{\{\xi_1^{(t)},\ldots,\xi_m^{(t)}|\boldsymbol{w}_1^{(t)},\ldots,\boldsymbol{w}_m^{(t)}\}}\mathbb{E}_{\mathbf{S},\mathcal{K}}\left[\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}_{\mathbf{S},\mathcal{K}}^{(r)}\rangle\right]\\
&= -\mathbb{E}_{\{\xi_1^{(t)},\ldots,\xi_m^{(t)}|\boldsymbol{w}_1^{(t)},\ldots,\boldsymbol{w}_m^{(t)}\}}\left[\left\langle\nabla f(\boldsymbol{w}^{(r)}),\eta\sum_{j\in\mathcal{K}}q_j\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right\rangle\right]\\
&= -\left\langle\nabla f(\boldsymbol{w}^{(r)}),\eta\sum_{j=1}^{m}q_j\sum_{c=0}^{\tau-1}\mathbb{E}_{\xi,\mathbf{S}}\left[\tilde{\mathbf{g}}_{j,\mathbf{S}}^{(c,r)}\right]\right\rangle\\
&= -\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left\langle\nabla f(\boldsymbol{w}^{(r)}),\mathbf{g}_j^{(c,r)}\right\rangle\\
&\overset{①}{=}\frac{1}{2}\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2 - \|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2 + \|\nabla f(\boldsymbol{w}^{(r)}) - \nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2\right]\\
&\overset{②}{\le}\frac{1}{2}\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2 - \|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2 + L^2\|\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(c,r)}\|_2^2\right]
\end{aligned} \tag{21}$$

where ① is due to $2\langle\mathbf{a},\mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}-\mathbf{b}\|^2$, and ② follows from Assumption 2. □

The following lemma bounds the distance of local solutions from global solution at rth communication round.

Lemma 6. *Under Assumptions 4 we have:*

$$\mathbb{E}\left[\|\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(c,r)}\|_2^2\right] \le \eta^2\tau\sum_{c=0}^{\tau-1}\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2 + \eta^2\tau\sigma^2 \tag{22}$$

Proof. Note that

$$\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(c,r)}\right\|_2^2\right] &= \mathbb{E}\left[\left\|\boldsymbol{w}^{(r)} - \left(\boldsymbol{w}^{(r)} - \eta\sum_{k=0}^{c}\tilde{\mathbf{g}}_j^{(k,r)}\right)\right\|_2^2\right]\\
&= \mathbb{E}\left[\left\|\eta\sum_{k=0}^{c}\tilde{\mathbf{g}}_j^{(k,r)}\right\|_2^2\right]
\end{aligned}$$

$$
\begin{aligned}
&\overset{①}{=} \mathbb{E}\left[\left\|\eta\sum_{k=0}^{c}\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+\left[\left\|\eta\sum_{k=0}^{c}\mathbf{g}_j^{(k,r)}\right\|_2^2\right]\\
&\overset{②}{=} \eta^2\sum_{k=0}^{c}\mathbb{E}\left[\left\|\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+(c+1)\,\eta^2\sum_{k=0}^{c}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]\\
&\leq \eta^2\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+\tau\eta^2\sum_{k=0}^{\tau-1}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]\\
&\overset{③}{\leq} \eta^2\sum_{k=0}^{\tau-1}\sigma^2+\tau\eta^2\sum_{k=0}^{\tau-1}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]\\
&= \eta^2\tau\sigma^2+\eta^2\sum_{k=0}^{\tau-1}\tau\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2
\end{aligned} \tag{23}
$$

where ① comes from $\mathbb{E}\left[\mathbf{x}^2\right]=\mathrm{Var}\left[\mathbf{x}\right]+\left[\mathbb{E}\left[\mathbf{x}\right]\right]^2$ and ② holds because $\mathrm{Var}\left(\sum_{j=1}^{n}\mathbf{x}_j\right)=\sum_{j=1}^{n}\mathrm{Var}\left(\mathbf{x}_j\right)$ for i.i.d. vectors $\mathbf{x}_i$ (and i.i.d. assumption comes from i.i.d. sampling), and finally ③ follows from Assumption 4. ☐

B.1 Main result for the non-convex setting

Now we are ready to present our result for the homogeneous setting. We first state and prove the result for the general nonconvex objectives.

Theorem 5 (Non-convex)**.** *For* `FedSKETCH`(τ,η,γ)*, for all* $0\leq t\leq R\tau-1$*, under Assumptions 2 to 4, if the learning rate satisfies*

$$
1\geq\tau^2L^2\eta^2+\left(\frac{\omega}{k}+1\right)\eta\gamma L\tau \tag{24}
$$

and all local model parameters are initialized at the same point $\boldsymbol{w}^{(0)}$*, then the average-squared gradient after* τ *iterations is bounded as follows:*

$$
\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2\leq\frac{2\left(f(\boldsymbol{w}^{(0)})-f(\boldsymbol{w}^{(*)})\right)}{\eta\gamma\tau R}+\frac{L\eta\gamma(\omega+1)}{k}\sigma^2+L^2\eta^2\tau\sigma^2 \tag{25}
$$

where $\boldsymbol{w}^{(*)}$ *is the global optimal solution with function value* $f(\boldsymbol{w}^{(*)})$*.*

Proof. Before proceeding to the proof of Theorem 5, we would like to highlight that

$$
\boldsymbol{w}^{(r)}-\ \boldsymbol{w}_j^{(\tau,r)}=\eta\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}. \tag{26}
$$

From the updating rule of Algorithm 5 we have

$$
\boldsymbol{w}^{(r+1)}=\boldsymbol{w}^{(r)}-\gamma\eta\left(\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\Big(\sum_{c=0,r}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\Big)\right)=\boldsymbol{w}^{(r)}-\gamma\left[\frac{\eta}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\right] \tag{27}
$$

In what follows, we use the following notation to denote the stochastic gradient used to update the global model at rth communication round

$$
\tilde{\mathbf{g}}_{\mathbf{S},\mathcal{K}}^{(r)}\triangleq\frac{\eta}{p}\sum_{j=1}^{p}\mathbf{S}\left(\frac{\boldsymbol{w}^{(r)}-\ \boldsymbol{w}_j^{(\tau,r)}}{\eta}\right)=\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right).
$$

and notice that $\boldsymbol{w}^{(r)}=\boldsymbol{w}^{(r-1)}-\gamma\tilde{\mathbf{g}}^{(r)}$.

Then using the Assumption ?? we have:

$$\mathbb{E}_{\mathbf{S}}\left[\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\right]=\frac{1}{k}\sum_{j\in\mathcal{K}}\left[-\eta\mathbb{E}_{\mathbf{S}}\left[\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\right]\right]=\frac{1}{k}\sum_{j\in\mathcal{K}}\left[-\eta\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\right]\triangleq\tilde{\mathbf{g}}_{\mathbf{S},\mathcal{K}}^{(r)} \tag{28}$$

From the L-smoothness gradient assumption on global objective, by using $\tilde{\mathbf{g}}^{(r)}$ in inequality (26) we have:

$$f(\boldsymbol{w}^{(r+1)})-f(\boldsymbol{w}^{(r)})\leq-\gamma\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}^{(r)}\rangle+\frac{\gamma^2 L}{2}\|\tilde{\mathbf{g}}^{(r)}\|^2 \tag{29}$$

By taking expectation on both sides of above inequality over sampling, we get:

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\Big[f(\boldsymbol{w}^{(r+1)})-f(\boldsymbol{w}^{(r)})\Big]\right]&\leq-\gamma\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\rangle\right]\right]+\frac{\gamma^2 L}{2}\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]\\&\overset{(a)}{=}-\gamma\underbrace{\mathbb{E}\left[\left[\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}^{(r)}\rangle\right]\right]}_{\text{(I)}}+\frac{\gamma^2 L}{2}\underbrace{\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]\right]}_{\text{(II)}}\end{aligned} \tag{30}$$

We proceed to use Lemma 4, Lemma 5, and Lemma 6, to bound terms (I) and (II) in right hand side of (30), which gives

$$\begin{aligned}&\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\Big[f(\boldsymbol{w}^{(r+1)})-f(\boldsymbol{w}^{(r)})\Big]\right]\\&\leq\gamma\frac{1}{2}\eta\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\left[-\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2-\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2+L^2\eta^2\sum_{c=0}^{\tau-1}\left[\tau\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2+\sigma^2\right]\right]\\&\quad+\frac{\gamma^2L(\frac{\omega}{k}+1)}{2}\left[\eta^2\tau\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2\right]+\frac{\gamma^2\eta^2L(\omega+1)}{2}\frac{\tau\sigma^2}{k}\\&\overset{①}{\leq}\frac{\gamma\eta}{2}\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\left[-\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2-\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2+\tau L^2\eta^2\left[\tau\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2+\sigma^2\right]\right]\\&\quad+\frac{\gamma^2L(\frac{\omega}{k}+1)}{2}\left[\eta^2\tau\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2\right]+\frac{\gamma^2\eta^2L(\omega+1)}{2}\frac{\tau\sigma^2}{k}\\&=-\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2\\&\quad-\left(1-\tau L^2\eta^2\tau-(\frac{\omega}{k}+1)\eta\gamma L\tau\right)\frac{\eta\gamma}{2}\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2+\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta+\gamma(\omega+1)\right)\sigma^2\\&\overset{②}{\leq}-\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2+\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta+\gamma(\omega+1)\right)\sigma^2\end{aligned} \tag{31}$$

where in ① we incorporate outer summation $\sum_{c=0}^{\tau-1}$, and ② follows from condition

$$1\geq\tau L^2\eta^2\tau+(\frac{\omega}{k}+1)\eta\gamma L\tau. \tag{32}$$

Summing up for all R communication rounds and rearranging the terms gives:

$$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2\leq\frac{2\left(f(\boldsymbol{w}^{(0)})-f(\boldsymbol{w}^{(*)})\right)}{\eta\gamma\tau R}+\frac{L\eta\gamma(\omega+1)}{k}\sigma^2+L^2\eta^2\tau\sigma^2 \tag{33}$$

From above inequality, is it easy to see that in order to achieve a linear speed up, we need to have $\eta\gamma=O\left(\frac{\sqrt{k}}{\sqrt{R\tau}}\right)$. □

Corollary 4 (Linear speed up)**.** *In Eq. (25) for the choice of* $\eta\gamma = O\left(\frac{1}{L}\sqrt{\frac{k}{R\tau(\omega+1)}}\right)$*, and* $\gamma \geq k$ *the convergence rate reduces to:*

$$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq O\left(\frac{L\sqrt{(\omega+1)}\left(f(\boldsymbol{w}^{(0)})-f(\boldsymbol{w}^*)\right)}{\sqrt{kR\tau}} + \frac{\left(\sqrt{(\omega+1)}\right)\sigma^2}{\sqrt{kR\tau}} + \frac{k\sigma^2}{R\gamma^2}\right). \tag{34}$$

Note that according to Eq. (34), if we pick a fixed constant value for γ*, in order to achieve an* ϵ*-accurate solution,* $R = O\left(\frac{1}{\epsilon}\right)$ *communication rounds and* $\tau = O\left(\frac{\omega+1}{k\epsilon}\right)$ *local updates are necessary. We also highlight that Eq. (34) also allows us to choose* $R = O\left(\frac{\omega+1}{\epsilon}\right)$ *and* $\tau = O\left(\frac{1}{k\epsilon}\right)$ *to get the same convergence rate.*

Remark 7. *Condition in Eq. (24) can be rewritten as*

$$\begin{aligned}\eta &\leq \frac{-\gamma L\tau\left(\frac{\omega}{k}+1\right)+\sqrt{\gamma^2\left(L\tau\left(\frac{\omega}{k}+1\right)\right)^2+4L^2\tau^2}}{2L^2\tau^2}\\ &= \frac{-\gamma L\tau\left(\frac{\omega}{k}+1\right)+L\tau\sqrt{\left(\frac{\omega}{k}+1\right)^2\gamma^2+4}}{2L^2\tau^2}\\ &= \frac{\sqrt{\left(\frac{\omega}{k}+1\right)^2\gamma^2+4}-\left(\frac{\omega}{k}+1\right)\gamma}{2L\tau}\end{aligned} \tag{35}$$

So based on Eq. (35), if we set $\eta = O\left(\frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\omega+1)}}\right)$*, it implies that:*

$$R \geq \frac{\tau k}{(\omega+1)\gamma^2\left(\sqrt{\left(\frac{\omega}{k}+1\right)^2\gamma^2+4}-\left(\frac{\omega}{k}+1\right)\gamma\right)^2} \tag{36}$$

We note that $\gamma^2\left(\sqrt{\left(\frac{\omega}{k}+1\right)^2\gamma^2+4}-\left(\frac{\omega}{k}+1\right)\gamma\right)^2 = \Theta(1) \leq 5$ *therefore even for* $\gamma \geq m$ *we need to have*

$$R \geq \frac{\tau k}{5(\omega+1)} = O\left(\frac{\tau k}{(\omega+1)}\right) \tag{37}$$

Therefore, for the choice of $\tau = O\left(\frac{\omega+1}{k\epsilon}\right)$*, due to condition in Eq. (37), we need to have* $R = O\left(\frac{1}{\epsilon}\right)$*. Similarly, we can have* $R = O\left(\frac{\omega+1}{\epsilon}\right)$ *and* $\tau = O\left(\frac{1}{k\epsilon}\right)$*.*

Corollary 5 (Special case, $\gamma = 1$)**.** *By letting* $\gamma = 1$*,* $\omega = 0$ *and* $k = p$ *the convergence rate in Eq. (25) reduces to*

$$\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \frac{2\left(f(\boldsymbol{w}^{(0)})-f(\boldsymbol{w}^{(*)})\right)}{\eta R\tau} + \frac{L\eta}{p}\sigma^2 + L^2\eta^2\tau\sigma^2 \tag{38}$$

which matches the rate obtained in [34]. In this case the communication complexity and the number of local updates become

$$R = O\left(\frac{p}{\epsilon}\right), \quad \tau = O\left(\frac{1}{\epsilon}\right). \tag{39}$$

This simply implies that in this special case the convergence rate of our algorithm reduces to the rate obtained in [34], which indicates the tightness of our analysis.

B.2 Main result for the PL/Strongly convex setting

We now turn to stating the convergence rate for the homogeneous setting under PL condition which naturally leads to the same rate for strongly convex functions.

Theorem 6 (PL or strongly convex)**.** *For* `FedSKETCH`(τ, η, γ)*, for all* $0 \leq t \leq R\tau - 1$*, under Assumptions 2 to 4 and 3,if the learning rate satisfies*

$$1 \geq \tau^2 L^2 \eta^2 + \left(\frac{\omega}{k} + 1\right) \eta\gamma L\tau \tag{40}$$

and if the all the models are initialized with $\boldsymbol{w}^{(0)}$ *we obtain:*

$$\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq (1 - \eta\gamma\mu\tau)^R \left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\mu}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] \tag{41}$$

Proof. From Eq. (31) under condition:

$$1 \geq \tau L^2\eta^2\tau + (\frac{\omega}{k} + 1)\eta\gamma L\tau \tag{42}$$

we obtain:

$$\begin{aligned}\mathbb{E}\Big[f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(r)})\Big] &\leq -\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2 \\ &\leq -\eta\mu\gamma\tau\left(f(\boldsymbol{w}^{(r)}) - f(\boldsymbol{w}^{(r)})\right) + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2\end{aligned} \tag{43}$$

which leads to the following bound:

$$\mathbb{E}\Big[f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(*)})\Big] \leq (1 - \eta\mu\gamma\tau)\Big[f(\boldsymbol{w}^{(r)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2 \tag{44}$$

By setting $\Delta = 1 - \eta\mu\gamma\tau$ we obtain the following bound:

$$\begin{aligned}&\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \\ &\leq \Delta^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1-\Delta^R}{1-\Delta}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2 \\ &\leq \Delta^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1}{1-\Delta}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2 \\ &= (1 - \eta\mu\gamma\tau)^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1}{\eta\mu\gamma\tau}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2\end{aligned} \tag{45}$$

□

Corollary 6. *If we let* $\eta\gamma\mu\tau \leq \frac{1}{2}$*,* $\eta = \frac{1}{2L\left(\frac{\omega}{k}+1\right)\tau\gamma}$ *and* $\kappa = \frac{L}{\mu}$ *the convergence error in Theorem 6, with* $\gamma \geq k$ *results in:*

$$\begin{aligned}&\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \\ &\leq e^{-\eta\gamma\mu\tau R}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\mu}\left[\frac{1}{2}\tau L^2\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] \\ &\leq e^{-\frac{R}{2\left(\frac{\omega}{k}+1\right)\kappa}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\mu}\left[\frac{1}{2}L^2\frac{\tau\sigma^2}{L^2\left(\frac{\omega}{k}+1\right)^2\gamma^2\tau^2} + \frac{(1+\omega)L\sigma^2}{2\left(\frac{\omega}{k}+1\right)L\tau k}\right] \\ &= O\left(e^{-\frac{R}{2\left(\frac{\omega}{k}+1\right)\kappa}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{\sigma^2}{\left(\frac{\omega}{k}+1\right)^2\gamma^2\mu\tau} + \frac{(\omega+1)\sigma^2}{\mu\left(\frac{\omega}{k}+1\right)\tau k}\right) \\ &= O\left(e^{-\frac{R}{2\left(\frac{\omega}{k}+1\right)\kappa}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{\sigma^2}{\gamma^2\mu\tau} + \frac{(\omega+1)\sigma^2}{\mu\left(\frac{\omega}{k}+1\right)\tau k}\right)\end{aligned} \tag{46}$$

which indicates that to achieve an error of ϵ*, we need to have* $R = O\left(\left(\frac{\omega}{k}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \frac{(\omega+1)}{k\left(\frac{\omega}{k}+1\right)\epsilon}$*. Additionally, we note that if* $\gamma \to \infty$*, yet* $R = O\left(\left(\frac{\omega}{k}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = \frac{(\omega+1)}{k\left(\frac{\omega}{k}+1\right)\epsilon}$ *will be necessary.*

B.3 Main result for the general convex setting

Theorem 7 (Convex)**.** *For a general convex function $f(\boldsymbol{w})$ with optimal solution $\boldsymbol{w}^{(*)}$, using* `FedSKETCH`(τ, η, γ) *to optimize $\tilde{f}(\boldsymbol{w}, \phi) = f(\boldsymbol{w}) + \frac{\phi}{2}\|\boldsymbol{w}\|^2$, for all $0 \leq t \leq R\tau - 1$, under Assumptions 2 to 4, if the learning rate satisfies*

$$1 \geq \tau^2 L^2 \eta^2 + \left(\frac{\omega}{k} + 1\right)\eta\gamma L\tau \tag{47}$$

and if the all the models initiate with $\boldsymbol{w}^{(0)}$, with $\phi = \frac{1}{\sqrt{k\tau}}$ and $\eta = \frac{1}{2L\gamma\tau\left(1+\frac{\omega}{k}\right)}$ we obtain:

$$\begin{aligned}\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] &\leq e^{-\frac{R}{2L\left(1+\frac{\omega}{k}\right)\sqrt{m\tau}}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) \\ &\quad + \left[\frac{\sqrt{k}\sigma^2}{8\sqrt{\tau}\gamma^2\left(1+\frac{\omega}{k}\right)^2} + \frac{(\omega+1)\,\sigma^2}{4\left(\frac{\omega}{k}+1\right)\sqrt{k\tau}}\right] + \frac{1}{2\sqrt{k\tau}}\left\|\boldsymbol{w}^{(*)}\right\|^2 \end{aligned} \tag{48}$$

We note that above theorem implies that to achieve a convergence error of ϵ we need to have $R = O\left(L\left(1+\frac{\omega}{k}\right)\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ and $\tau = O\left(\frac{(\omega+1)^2}{k\left(\frac{\omega}{k}+1\right)^2\epsilon}\right)$.

Proof. Since $\tilde{f}(\boldsymbol{w}^{(r)}, \phi) = f(\boldsymbol{w}^{(r)}) + \frac{\phi}{2}\left\|\boldsymbol{w}^{(r)}\right\|^2$ is ϕ-PL, according to Theorem 6, we have:

$$\begin{aligned}&\tilde{f}(\boldsymbol{w}^{(R)}, \phi) - \tilde{f}(\boldsymbol{w}^{(*)}, \phi) \\ &= f(\boldsymbol{w}^{(r)}) + \frac{\phi}{2}\left\|\boldsymbol{w}^{(r)}\right\|^2 - \left(f(\boldsymbol{w}^{(*)}) + \frac{\phi}{2}\left\|\boldsymbol{w}^{(*)}\right\|^2\right) \\ &\leq (1-\eta\gamma\phi\tau)^R\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right]\end{aligned} \tag{49}$$

Next rearranging Eq. (49) and replacing μ with ϕ leads to the following error bound:

$$\begin{aligned}&f(\boldsymbol{w}^{(R)}) - f^* \\ &\leq (1-\eta\gamma\phi\tau)^R\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] \\ &\quad + \frac{\phi}{2}\left(\|\boldsymbol{w}^*\|^2 - \left\|\boldsymbol{w}^{(r)}\right\|^2\right) \\ &\leq e^{-(\eta\gamma\phi\tau)R}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] + \frac{\phi}{2}\left\|\boldsymbol{w}^{(*)}\right\|^2\end{aligned} \tag{50}$$

Next, if we set $\phi = \frac{1}{\sqrt{k\tau}}$ and $\eta = \frac{1}{2\left(1+\frac{\omega}{k}\right)L\gamma\tau}$, we obtain that

$$\begin{aligned}&f(\boldsymbol{w}^{(R)}) - f^* \\ &\leq e^{-\frac{R}{2\left(1+\frac{\omega}{k}\right)L\sqrt{m\tau}}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \sqrt{k\tau}\left[\frac{\sigma^2}{8\tau\gamma^2\left(1+\frac{\omega}{k}\right)^2} + \frac{(\omega+1)\,\sigma^2}{4\left(\frac{\omega}{k}+1\right)\tau k}\right] + \frac{1}{2\sqrt{k\tau}}\left\|\boldsymbol{w}^{(*)}\right\|^2,\end{aligned} \tag{51}$$

thus the proof is complete. $\square$

C Proof of main Theorems

The proof of Theorem 2 follows directly from the results in [9]. For the sake of the completeness we review an assumptions from this reference for the quantiziation with their notation.

Assumption 6 ([9])**.** *The output of the compression operator $Q(\mathbf{x})$ is an unbiased estimator of its input $\mathbf{x}$, and its variance grows with the squared of the squared of ℓ_2-norm of its argument, i.e.,* $\mathbb{E}[Q(\mathbf{x})] = \mathbf{x}$ *and* $\mathbb{E}\left[\|Q(\mathbf{x})-\mathbf{x}\|^2\right] \leq \omega \|\mathbf{x}\|^2$.

C.1 Proof of Theorem 2

Based on Assumption 6 we have:

Theorem 8 ([9])**.** *Consider* `FedCOM` *in [9]. Suppose that the conditions in Assumptions 2, 4 and 6 hold. If the local data distributions of all users are identical (homogeneous setting), then we have*

- ***Nonconvex:*** *By choosing stepsizes as* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau\left(\frac{\omega}{p}+1\right)}}$ *and* $\gamma \geq p$*, the sequence of iterates satisfies* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *if we set* $R = O\left(\frac{1}{\epsilon}\right)$ *and* $\tau = O\left(\frac{\frac{\omega}{p}+1}{p\epsilon}\right)$.
- ***Strongly convex or PL:*** *By choosing stepsizes as* $\eta = \frac{1}{2L\left(\frac{\omega}{p}+1\right)\tau\gamma}$ *and* $\gamma \geq m$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = O\left(\left(\frac{\omega}{p}+1\right)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.
- ***Convex:*** *By choosing stepsizes as* $\eta = \frac{1}{2L\left(\frac{\omega}{p}+1\right)\tau\gamma}$ *and* $\gamma \geq p$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = O\left(\frac{L\left(1+\frac{\omega}{p}\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon^2}\right)$.

Proof. Since the sketching `PRIVIX` and `HEAPRIX`, satisfy the Assumption 6 with $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively with probablity $1-\delta$. Therefore, all the results in Theorem 2, conclude from Theorem 8 with probability $1-\delta$ and plugging $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively into the corresponding convergence bounds. □

C.2 Proof of Theorem 3

For the heterogeneous setting, the results in [9] requires the following extra assumption that naturally holds for the sketching:

Assumption 7 ([9])**.** *The compression scheme Q for the heterogeneous data distribution setting satisfies the following condition* $\mathbb{E}_Q[\|\frac{1}{m}\sum_{j=1}^m Q(\boldsymbol{x}_j)\|^2 - \|Q(\frac{1}{m}\sum_{j=1}^m \boldsymbol{x}_j)\|^2] \leq G_q$.

We note that since sketching is a linear compressor, in the case of our algorithms for heterogeneous setting we have $G_q = 0$.

Next, we restate the Theorem in [9] here as follows:

Theorem 9. *Consider* `FedCOMGATE` *in [9]. If Assumptions 2, 5, 6 and 7 hold, then even for the case the local data distribution of users are different (heterogeneous setting) we have*

- ***Non-convex:*** *By choosing stepsizes as* $\eta = \frac{1}{L\gamma}\sqrt{\frac{p}{R\tau(\omega+1)}}$ *and* $\gamma \geq p$*, we obtain that the iterates satsify* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *if we set* $R = O\left(\frac{\omega+1}{\epsilon}\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.
- ***Strongly convex or PL:*** *By choosing stepsizes as* $\eta = \frac{1}{2L\left(\frac{\omega}{p}+1\right)\tau\gamma}$ *and* $\gamma \geq \sqrt{p\tau}$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = O\left((\omega+1)\kappa\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon}\right)$.
- ***Convex:*** *By choosing stepsizes as* $\eta = \frac{1}{2L(\omega+1)\tau\gamma}$ *and* $\gamma \geq \sqrt{p\tau}$*, we obtain that the iterates satisfy* $\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big] \leq \epsilon$ *if we set* $R = O\left(\frac{L(1+\omega)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon^2}\right)$.

Proof. Since the sketching PRIVIX and HEAPRIX, satisfy the Assumption 6 with $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively with probablity $1 - \delta$. Therefore, all the results in Theorem 3, conclude from Theorem 9 with probability $1 - \delta$ and plugging $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively into the convergence bounds. □