`FedSKETCH`: Communication-Efficient and Private Federated Learning via Sketching

Farzin Haddadpour, Belhal Karimi, Ping Li, Xiaoyun Li

Cognitive Computing Lab

Baidu Research

10900 NE 8th St. Bellevue, WA 98004

{farzin.haddadpour, belhal.karimi, lixiaoyun996, pingli98}@gmail.com

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

Federated Learning is a recently emerging setting for distributed large scale machine learning problems. In Federated Learning, data is distributed across devices (which could be any smartphone or IOT edge device) [?, ?] and due to privacy concerns, users are only allowed to communicate with parameter server. The parameter server orchestrates optimization among devices by aggregating gradient-related information of devices and broadcasts the average of received vectors. Additionally, moving data across the devices for the purpose of learning a global model can be impractical and could violate the privacy of users/devices [?, ?].

There are a number of challenges to be addressed in Federated Learning to efficiently learn a global model that performs well in average for all devices. The first challenge is the *communication-efficiency* as there could be a million of devices communicating iteratively among them which can incur huge communication overhead. The second challenge is *data heterogeneity*. Since the data in smartphones or devices are generated locally in Federated Learning, generated data may come from various probability distributions. Thus it is supposed that data distribution is non-iid. It is known that non-iid data distribution can lead to poor convergence error in practice [?, ?]. The last, yet important, issue is *device privacy* [?, ?]. It is important to make sure that the privacy of the sensitive information on each device is preserved during the training.

Almost all of the previous studies consider addressing the aforementioned challenges separately. One approach to deal with communication cost is the idea of *local SGD with periodic averaging* [?, ?, ?, ?] which asserts that instead of taking the average within each iteration, like baseline SGD [?], one may take the average periodically and performs local update, see local SGD [?]. It is shown

that local SGD with periodic averaging benefits from the same convergence rate as baseline SGD, while requiring less communication rounds. The second approach to deal with communication cost is aiming at reducing the size of communicated message per each communication round. Available methods reduce the size of the message by communicating compressed local gradients or models to parameter server via quantization [?, ?, ?, ?, ?], sparsification [?, ?, ?, ?].

There are a number of research efforts such as [?, ?, ?, ?] aiming at mitigating the effect of data heterogeneity by exploiting variance reduction or gradient tracking techniques in distributed optimization settings where data distribution is non-iid.

Solving the privacy issue has been widely performed by injecting an additional layer of random noise in order to respect differential-privacy property of the method [?] or using cryptography based approaches under secure multi-party computation [?] framework.

Another promising recent approach with a potential to tackle all major issues in Federated Learning setting is based on sketching algorithms [?, ?, ?, ?]. Sketches are built from independent hash tables (functions), needed to compress a high dimensional vector into a lower dimensional one and the corresponding estimation error of sketching are well studied. With the focus of communication-efficiency, [?] proposes a distributed SGD algorithm using sketching and they provide the convergence analysis in homogeneous data distribution setting. Also with focus on privacy, in [?], 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 are performed using random hash functions such that no third parties are able to access the original data. Yet, [?] does not provide the convergence analysis for the `DiffSketch` in Federated setting, and additionally the estimation error of the `DiffSketch` is relatively higher than the sketching scheme in [?] which might end up in poor convergence error. Finally, [?] considers using sketching technique for Federated Learning in heterogeneous setting from a communication-efficiency perspective. The proposed sketching schemes in [?, ?] are based on a deterministic scheme which requires having access to the exact values of the gradient-related information, thus are not privacy-preserving.

In this work, we provide a thorough convergence analysis for the Federated Learning using sketching for both homogeneous and heterogeneous settings. Additionally, all of our sketching algorithms including a novel scheme, do not need to obtain exact values of gradient, hence are privacy preserving. Therefore, our proposed algorithms based on sketching addresses all the aforementioned three main challenges jointly.

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 ?? and Theorem ?? 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 stationary point.

- We illustrate the benefits of `FedSKETCH` and `FedSKETCHGATE` over baseline methods through a set of experiments. In particular, we plot training loss and accuracy curves depending on the

method used for training, the size of the sketches employed and the number of local updates performed at each round of communication. Numerical experiments show the advantages of, in particular, `FedSKETCH-HEAPRIX` algorithm that achieves comparable test accuracy as Federated SGD (`FedSGD`) while compressing the information exchanged between devices and server.

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 [**?**] with purpose of saving communication rounds. While Local SGD has been proposed in [**?**, **?**] under the title of Federated Learning Setting, the convergence analysis of Local SGD is studied in [**?**, **?**, **?**, **?**]. The convergence analysis of Local SGD is improved in the follow up works [**?**, **?**, **?**, **?**, **?**, **?**] 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 [**?**, **?**, **?**, **?**, **?**, **?**]. Additionally, a few recent Federated Learning/Local SGD with adaptive gradient methods can be found in [**?**, **?**].

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

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

The remaining of the paper is organized as follows. Section **??** gives a formal presentation of the general problem. Section **??** 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 **??** and their respective analysis in the strongly-convex or nonconvex cases are provided Section **??**. Finally, in Section **??** we provide empirical results for our proposed algorithms.

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 $\mathbf{S}(\boldsymbol{x})$. We also denote $[p] = \{1, \ldots, p\}$.

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 $F_j(\boldsymbol{x}) = \mathbb{E}_{\xi\in\mathcal{D}_j}[L_j(\boldsymbol{x},\xi)]$ 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 across devices 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 the optimization problem in Eq. (**??**) for the homogeneous data distribution. In 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 (**??**) called `Count Sketch` (for more detail see [**?**]) is described Algorithm **??**. The algorithm for generating count sketching is using two sets of functions that encode any input vector $\boldsymbol{x}$ **into a hash table $S_{m\times t}(\boldsymbol{x})$**. 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 $\{\text{sign}_{j,1\leq j\leq t} : [d] \to \{+1,-1\}\}$ to map every entry of $\boldsymbol{x}$ ($\boldsymbol{x}_i$, $1 \leq i \leq d$) into t different columns of hash table $\mathbf{S}_{m\times t}$. These steps are summarized in Algorithm **??**.

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*

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

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

We note that this definition leads to the following property

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

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

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

Algorithm 2 `PRIVIX` [**?**]: Unbiased compressor based on sketching.

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

Remark 1. *Note that if $\Delta = 1$ then our algorithm reduces to 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 $\boldsymbol{x}$ from a count sketch noted $\boldsymbol{S}(\boldsymbol{x})$. In this algorithm, to query the quantity x_i, the $i - th$ element of the vector, we compute the median of t approximated values specified by the indices of $h_j(i)$ for $1 \leq j \leq t$. These steps are summarized in Algorithm **??**.

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

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

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

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

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

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

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 converge poorly.*

In the following we provide a review of privacy property of count sketch:

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 $\mathcal{O}$,*

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

For smaller ϵ, it will become more difficult to specify what is the input for the algorithm $\mathcal{O}$. Hence, smaller ϵ implies stronger privacy, and we desire to have ϵ as small as possible to impose stronger privacy guarantees. In the following, we review an assumption from [**?**] to discuss a property regarding privacy.

Algorithm 3 `HEAVYMIX`

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

Assumption 1 (Input vector distribution). *For the purpose of privacy analysis, similar to [?, ?, ?, ?], 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: $|s_i| \leq C, 1 \leq i \leq p$ for some positive constant $C > 0$.*

Based on Assumption **??**, the reference [**?**] proves the following:

Theorem 1 ($\epsilon-$ differential privacy of count sketch, [**?**]). *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 **??**, $\mathcal{O}$ 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 $\frac{\alpha C^2 m(m-1)}{\sigma^2(l-2)}(1 + \ln(l-m)) \leq \frac{1}{2} - \frac{1}{\alpha}$.*

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

Theorem **??** implies that if we use smaller hash table either through using smaller m or t, we will obtain stronger differential privacy. On the other hand, smaller hash table means bigger estimation error for a compression based on sketching. Therefore, there is an interesting trade-off between communication complexity and obtained privacy.

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 $\Delta \geq 1$, if we have*

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

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

The following Lemma links these two definitions:

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

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

Lemma 2 ([**?**]). *`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 **??** is a variation of the sketching algorithm developed in [**?**] with distinction that `HEAVYMIX` does not require extra second round of communication to obtain the

---

Algorithm 4 `HEAPRIX`

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

---

exact values of top_m. Additionally, while sketching algorithm based on `HEAVYMIX` has smaller estimation error compared to `PRIVIX`, it requires having access to the exact values of top_m, therefore such sketching does not benefit from differentially privacy similar to `PRIVIX`. In the following we introduce our sketching scheme which enjoys from privacy property as well as smaller estimation error.

4.4 Sketching Based on Induced Compressor

The following Lemma from [**?**] shows that we can convert the biased compressor into an unbiased one:

Lemma 3 (Induced Compressor [**?**]). *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`:

Corollary 1. *Based on Lemma **??** and using Algorithm **??**, we have $C(x) \in \mathbb{U}(\mu^2 d)$. This shows that unlike `PRIVIX` the compression noise can be made as small as possible using large size of hash table.*

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

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

5 Algorithms for Homogeneous and Heterogeneous Settings

In the following, we first present two algorithms for the homogeneous setting. Then, we present two other algorithms to deal with data heterogeneity. We emphasize that, for the sake of privacy in all of our algorithms, the query step is happening locally and the main task of the parameter server is to perform the average of the received messages from the devices and broadcast the average back to the devices.

5.1 Homogeneous Setting

In this section, we propose two algorithms for the setting where data across distributed devices are identically distributed. The proposed algorithms for Federated Learning leverage sketching techniques to compress communication. The main difference between the first suggested algorithm and the `DiffSketch` algorithm in [?] is that we use distinct local and global learning rates. Additionally, unlike [?], we do not add local Gaussian noise to ensure privacy.

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

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

where $\tilde{\boldsymbol{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \Xi_j^{(\ell,r)}) \triangleq \frac{1}{b} \sum_{\xi \in \Xi_j^{(\ell,r)}} \nabla L_j(\boldsymbol{x}_j^{(\ell,r)}, \xi)$ 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 and 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

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

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 update global model $\boldsymbol{x}^{(r)}$ using rule

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

We summarize these steps in `FedSKETCH`, see Algorithm **??**. A variant of this algorithm which uses a different compression scheme, called `HEAPRIX` is also described in Algorithm **??**. 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

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

where $\boldsymbol{S}^{(r)}$ is computed using Eq. (**??**) and then updates its global model using $\boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r)} - \gamma \boldsymbol{\Phi}^{(r)}$.

Remark 5 (Improvement over [?]). *An important feature of our algorithm is that due to a lower dimension of the count sketch, the resulting averages ($\mathbf{S}^{(r)}$ and $\tilde{\mathbf{S}}^{(r)}$) received 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 quantization error shown by $\omega = \theta(\frac{d}{m})$ in [?], our algorithms outperform* `FedCOM` *and* `FedCOMGATE` *developed in [?]. 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.

1: **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
2: **for** $r = 0, \ldots, R - 1$ **do**
3:      **parallel for device** $j \in \mathcal{K}^{(r)}$ **do**:
4:        **if PRIVIX variant:**
5:          Computes $\boldsymbol{\Phi}^{(r)} \triangleq$ `PRIVIX` $\left[\mathbf{S}^{(r-1)} \right]$
6:        **if HEAPRIX variant:**
7:          Computes $\boldsymbol{\Phi}^{(r)} \triangleq$ `HEAVYMIX` $\left[\mathbf{S}^{(r-1)} \right] +$ `PRIVIX` $\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)} \right]$
8:        Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma \boldsymbol{\Phi}^{(r)}$
9:        Set $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
10:        **for** $\ell = 0, \ldots, \tau - 1$ **do**
11:          Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
12:          $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta \, \tilde{\mathbf{g}}_j^{(\ell,r)}$
13:        **end for**
14:        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.
15:      Server **computes**
16:        $\mathbf{S}^{(r)} = \frac{1}{k} \sum_{j \in \mathcal{K}} \mathbf{S}_j^{(r)}$.
17:      Server samples a subset of devices $\mathcal{K}^{(r)}$ randomly with replacement and **broadcasts** $\mathbf{S}^{(r)}$ to devices in set $\mathcal{K}^{(r)}$.
18:        **if HEAPRIX variant:**
19:        Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j \left[\text{HEAVYMIX}(\mathbf{S}^{(r)}) \right]$
20:        Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{k} \sum_{j \in \mathcal{K}} \delta_j^{(r)}$ to devices in set $\mathcal{K}^{(r)}$
21:      **end parallel for**
22: **end**
23: **Output:** $\boldsymbol{x}^{(R-1)}$

5.2 Heterogeneous Setting

In this section, we focus on the optimization problem in Eq. (??) 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

---

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

1: **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.
2: **for** $r = 0, \ldots, R - 1$ **do**
3:      **parallel for device** $j = 1, \ldots, p$ **do:**
4:         **if PRIVIX variant:**
5:            Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\mathtt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \mathtt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$
6:            Computes $\boldsymbol{\Phi}^{(r)} \triangleq \mathtt{PRIVIX}(\mathbf{S}^{(r-1)})$
7:         **if HEAPRIX variant:**
8:            Set $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\boldsymbol{\Phi}^{(r)} - \boldsymbol{\Phi}_j^{(r)}\right)$
9:            Computes $\boldsymbol{\Phi}^{(r)} \triangleq \mathtt{HEAVYMIX}\left[\mathbf{S}^{(r-1)}\right] + \mathtt{PRIVIX}\left[\mathbf{S}^{(r-1)} - \tilde{\mathbf{S}}^{(r-1)}\right]$
10:         Set $\boldsymbol{x}^{(r)} = \boldsymbol{x}^{(r-1)} - \gamma\boldsymbol{\Phi}^{(r)}$ and $\boldsymbol{x}_j^{(0,r)} = \boldsymbol{x}^{(r)}$
11:         **for** $\ell = 0, \ldots, \tau - 1$ **do**
12:            Sample a mini-batch $\xi_j^{(\ell,r)}$ and compute $\tilde{\mathbf{g}}_j^{(\ell,r)} \triangleq \nabla f_j(\boldsymbol{x}_j^{(\ell,r)}, \xi_j^{(\ell,r)})$
13:            $\boldsymbol{x}_j^{(\ell+1,r)} = \boldsymbol{x}_j^{(\ell,r)} - \eta\left(\tilde{\mathbf{g}}_j^{(\ell,r)} - \mathbf{c}_j^{(r)}\right)$
14:         **end for**
15:         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.
16:      Server **computes**
17:         $\mathbf{S}^{(r)} = \frac{1}{p}\sum_{j=1}\mathbf{S}_j^{(r)}$ and **broadcasts** $\mathbf{S}^{(r)}$ to all devices.
18:      **if HEAPRIX variant:**
19:         Device j computes $\boldsymbol{\Phi}_j^{(r)} \triangleq \mathtt{HEAPRIX}[\mathbf{S}_j^{(r)}]$
20:         Second round of communication to obtain $\delta_j^{(r)} := \mathbf{S}_j\left(\mathtt{HEAVYMIX}[\mathbf{S}^{(r)}]\right)$
21:         Broadcasts $\tilde{\mathbf{S}}^{(r)} \triangleq \frac{1}{p}\sum_{j=1}^{p}\delta_j^{(r)}$ to devices
22:      **end parallel for**
23: **end**
24: **Output:** $\boldsymbol{x}^{(R-1)}$

---

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 [**?**] (with compression) and a special case of $\gamma = 1$ and 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 `FedSKETCHGATE` with `PRIVIX` variant, the correction vector $\mathbf{c}_j^{(r)}$ at device j and communication round r is computed using the update rule $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\mathtt{PRIVIX}\left(\mathbf{S}^{(r-1)}\right) - \mathtt{PRIVIX}\left(\mathbf{S}_j^{(r-1)}\right)\right)$ 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 (**??**). For `FedSKETCHGATE`, the server needs to compute $\tilde{\mathbf{S}}^{(r)}$ using (**??**). Then, device j computes $\boldsymbol{\Phi}_j \triangleq \mathtt{HEAPRIX}[\mathbf{S}_j^{(r)}]$ and $\boldsymbol{\Phi} \triangleq \mathtt{HEAPRIX}(\mathbf{S}^{(r-1)})$ and updates the correction vector $\mathbf{c}_j^{(r)}$ using the recursion $\mathbf{c}_j^{(r)} = \mathbf{c}_j^{(r-1)} - \frac{1}{\tau}\left(\boldsymbol{\Phi} - \boldsymbol{\Phi}_j\right)$.

6 Convergence Analysis

In this section we start with a few common assumptions, then we provide the convergence results.

6.1 Common 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., $\|\nabla f_j(\boldsymbol{x}) - \nabla f_j(\mathbf{y})\| \leq L\|\boldsymbol{x} - \mathbf{y}\|, \forall \, \boldsymbol{x}, \mathbf{y} \in \mathbb{R}^d$. 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(PL) 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.*

We note that Assumption **??** is a common assumption in the literature of stochastic optimization. Additionally, it is shown in [**?**] that PL condition implies strong convexity property with same module. Additionally, PL objectives could also be nonconvex, hence strong convexity does not imply PL condition necessarily.

6.2 Convergence of `FEDSKETCH` for Homogeneous Setting

Now we focus on the homogeneous case where data is distributed i.i.d. among local devices. In this case, the stochastic local gradient of each worker is an unbiased estimator of the global gradient. We will need the following additional common assumption on the stochastic gradients.

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., $\mathbb{E}_{\Xi_j}\big[\|\tilde{\mathbf{g}}_j - \mathbf{g}\|^2\big] \leq \sigma^2$.*

Theorem 2. *Suppose that the conditions in Assumptions **??**-**??** hold. Given $0 < m = O\left(\frac{e}{\mu^2}\right) \leq d$, and Consider `FedSKETCH` in Algorithm **??** 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 $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ 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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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 **??**, the total communication cost per-worker becomes*

$$O\left(RB\right) = O\left(Rm\log\left(\frac{dR}{\delta}\right)\right) = O\left(\frac{m}{\epsilon}\log\left(\frac{d}{\epsilon\delta}\right)\right). \tag{4}$$

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 [**?**, **?**, **?**] 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 [**?**] the total communication cost is*

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

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

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

*In comparison to [**?**], 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 **??** also improves the communication efficiency from server to devices as well because of using lower dimensional sketching size and the fact that the average of sketching has also small dimension.*

We note that it is not fair to compare our algorithms with algorithms without compression. However, in the following Corollary we share an interesting observation regarding our algorithm for PL and thus strongly convex objectives in homogeneous setting.

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*

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

As a consequence, the total communication cost becomes:

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

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

$$BkR = O\left(\kappa kd\log\left(\frac{p}{k\epsilon}\right)\right) = O\left(\kappa kd\log\left(\frac{p}{k\epsilon}\right)\right).$$

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

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

leading to an improvement from kd to $k + d$. These results are summarized in Table ??.

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

Theorem 3. *Suppose that the conditions in Assumptions ?? and ?? hold. Given $0 < m = O\left(\frac{e}{\mu^2}\right) \leq d$, and Consider `FedSKETCHGATE` in Algorithm ?? 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 $\gamma \geq p$, the sequence of iterates satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\mathbf{w}^{(r)})\right\|_2^2 \leq \epsilon$ 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 $\gamma \geq p$, the sequence of iterates satisfies $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\mathbf{w}^{(r)})\right\|_2^2 \leq \epsilon$ if we set $R = O\left(\frac{\mu^2 d}{\epsilon}\right)$ and $\tau = O\left(\frac{1}{p\epsilon}\right)$.

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

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

- **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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *if we set* $R = O\left(\frac{L\left(\mu^2 d\right)}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right)$ *and* $\tau = O\left(\frac{1}{p\epsilon^2}\right)$.

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 Property respectively.

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

6.4 Comparison with Prior Methods [?], [?] and [?]

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

Comparison with [?]. 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 [?] is based on top_m queries, both of the proposed algorithms (in [?]) 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 [?] 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 [?] for more detail). However, our convergence results do not need any bounded gradient assumption. Third, Theorem 1 [?] 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 [?] to be $\frac{m}{d}$). In contrast, without any extra assumptions, our results

in Theorem **??** for `PRIVIX` and `HEAPRIX` are respectively $BR = O\left(\frac{m(\mu^2 d+1)}{\epsilon} \log\left(\frac{\mu^2 d^2 + d}{\epsilon\delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)$

and $BR = O\left(\frac{m(\mu^2 d)}{\epsilon} \log\left(\frac{\mu^2 d^2}{\epsilon\delta} \log\left(\frac{1}{\epsilon}\right)\right)\right)$ which improves total communication cost in Theorem 1 in [**?**] in regimes where $\frac{1}{\epsilon} \geq d$ or $d >> m$. Theorem 2 in [**?**] is based on another assumption of Sliding Window Heavy Hitters, which is similar to gradient diversity assumption in [**?**, **?**] (but it is weaker assumption of contraction in Theorem 1 in [**?**]), 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 [**?**] which only focuses on nonconvex objectives, in this work we provide the convergence analysis for PL (thus strongly convex case), general convex and general nonconvex objectives. Finally, although the algorithm in [**?**] requires additional memory for the server to store the compression error correction vector, our algorithm does not need such additional storage.

These results are summarized in Table **??**.

Comparison with [?]. The reference [**?**] 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 (specializing to our case for the sake of comparison $\omega^{\mathrm{Up}} = \omega^{\mathrm{Down}} = \theta(d)$) for general heterogeneous data distribution. In contrast, while as pointed out in Remark **??** 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 Numerical Example

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 [**?**], with $60\,000$ model parameters in total.

Four methods are compared in our experiments: Federated SGD (FedSGD), SketchSGD [**?**], FedSketch-PRIVIX (FS-PRIVIX) and FedSketch-HEAPRIX (FS-HEAPRIX). We implement the algorithms by simulating the distributed and federated environment. Note that in Algorithm **??**, FS-PRIVIX with global learning rate $\gamma = 1$ is equivalent to the DiffSketch algorithm proposed in [**?**]. 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.
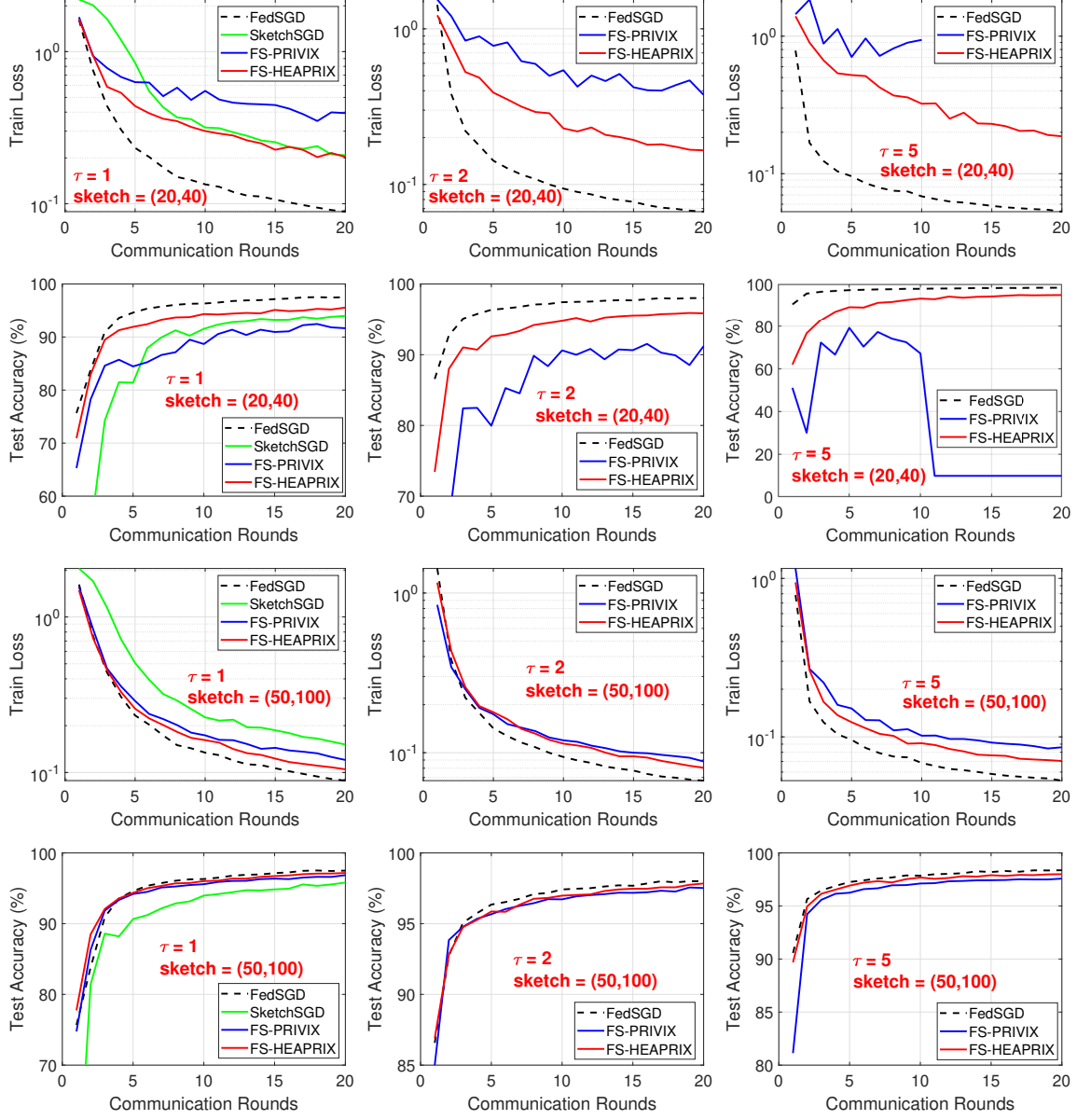
Figure 1 Homogeneous case: Comparison of compressed optimization methods on LeNet CNN architecture.

Homogeneous case. In Figure **??** first column, 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 column 2 and column 3 in Figure **??**, 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) algorithm.
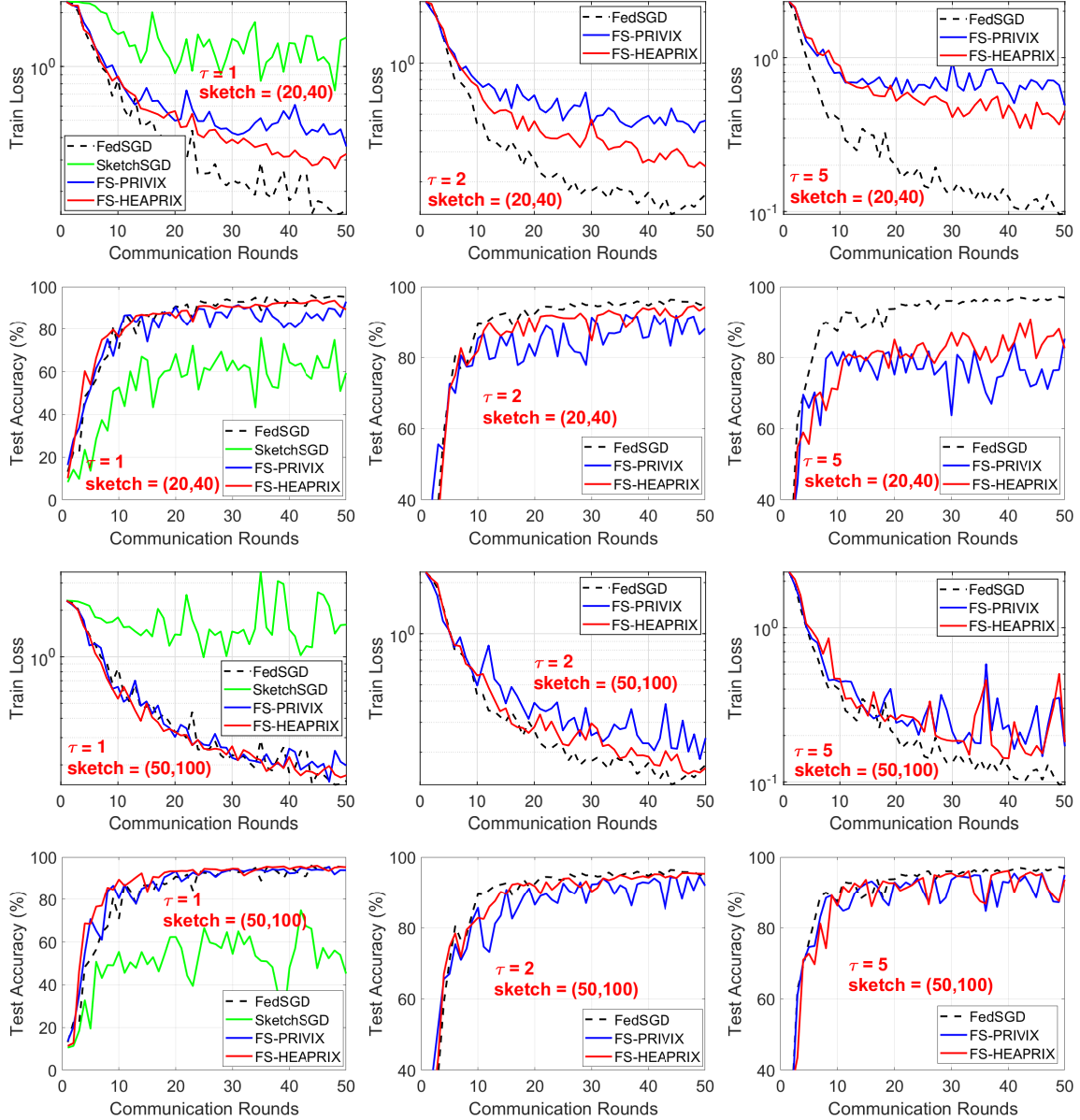
Figure 2 Heterogeneous case: Comparison of compressed optimization algorithms on LeNet CNN architecture.

Heterogeneous case. We plot similar sets of results in Figure **??** for non-i.i.d. data distribution (heterogeneous setting). This setting leads to more twists and turns in the training curves.

18

From the first column ($\tau = 1$), we see that SketchSGD performs very poorly in the heterogeneous case, while both our proposed FedSketchGATE methods, see Algorithm **??**, achieve similar generalization accuracy as the Federated SGD (FedSGD) algorithm, even with fairly small sketch size (i.e. $75\times$ compression ratio). Note that, the slow convergence of federated SGD in non-i.i.d. data distribution case has also been reported in literature, e.g. [**?**, **?**]. In addition, FS-HEAPRIX is again better than FS-PRIVIX in terms of both training loss and test accuracy.

Furthermore, we notice in column 2 and 3 of Figure **??** the advantage of FS-HEAPRIX over FS-PRIVIX with multiple local updates. However, empirically we see that in the heterogeneous setting, more local updates τ tend to undermine the learning performance, especially with small sketch size. Nevertheless, we see that when sketch size is large, i.e. $(50, 100)$, FS-HEAPRIX can still provide comparable test accuracy as FedSGD with $\tau = 5$.

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 [**?**] 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 *nonconvex*, *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. The proposed algorithms outperform competing compression method and can achieve comparable test accuracy as Federated SGD, with high compression ratio.

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 random subset of the devices selected by server with $\mathcal{K}$ with size $|\mathcal{K}| = k \le p$, 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 [?, ?]) in proving results.

Fact 4 ([?, ?]). *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 $1 \le j \le p$ with replacement. Let $\mathcal{P} = \{i_1, \ldots, i_K\} \subset [p]$ (some i_j may have the same value). Then*

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{i \in \mathcal{P}} x_i\right] = \mathbb{E}_{\mathcal{P}}\left[\sum_{k=1}^{K} x_{i_k}\right] = K\mathbb{E}_{\mathcal{P}}\left[x_{i_k}\right] = K\left[\sum_{j=1}^{p} q_j x_j\right] \tag{5}$$

A   Results for the Homogeneous Setting

In this section, we study the convergence properties of our `FedSKETCH` method presented in Algorithm **??**. 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 **??**, we have the following bound:*

$$\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S},\xi^{(r)}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]\right] = \mathbb{E}_{\xi^{(r)}}\mathbb{E}_{\mathbf{S}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right] \le \tau(\frac{\omega}{k} + 1)\sum_{j=1}^{m} q_j\left[\sum_{c=0}^{\tau-1} \|\mathbf{g}_j^{(c,r)}\|^2 + \sigma^2\right] \tag{6}$$

Proof.

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

$$=\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\overbrace{\underbrace{\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}}_{}}^{\tilde{\mathbf{g}}_j^{(r)}}\right)\|^2\right]\right]\right]$$

$$\overset{\textcircled{1}}{=}\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)} - \frac{1}{k}\sum_{j\in\mathcal{K}}\mathbb{E}_{\mathbf{S}}\left[\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right]\|^2\right] + \|\mathbb{E}_{\mathbf{S}}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S},j}^{(r)}\right]\|^2\right]\right]$$

$$\overset{\textcircled{2}}{=}\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\left[\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)} - \sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\right]\|^2\right] + \|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]$$

$$=\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\left[\text{Var}_{\mathbf{S}}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right]\right]+\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]$$

$$=\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\text{Var}_{\mathbf{S}_j}\left[\tilde{\mathbf{g}}_{\mathbf{S}j}^{(r)}\right]+\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]$$

$$\leq\mathbb{E}_{\xi^{(r)}}\left[\mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\omega\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2+\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]$$

$$=\left[\mathbb{E}_{\xi}\left[\frac{1}{k}\sum_{j\in\mathcal{K}}\omega\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2+\mathbb{E}_{\mathcal{K}}\mathbb{E}_{\xi^{(r)}}\|\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\|^2\right]\right]$$

$$=\left[\mathbb{E}_{\xi}\left[\frac{\omega}{k}\sum_{j=1}^{p}q_j\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2+\mathbb{E}_{\mathcal{K}}\left[\text{Var}\left(\frac{1}{k}\sum_{j\in\mathcal{K}}\tilde{\mathbf{g}}_j^{(r)}\right)+\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{g}_j^{(r)}\|^2\right]\right]\right]$$

$$=\frac{\omega}{k}\sum_{j=1}^{p}q_j\mathbb{E}_{\xi}\left\|\tilde{\mathbf{g}}_j^{(r)}\right\|^2+\mathbb{E}_{\mathcal{K}}\left[\frac{1}{k^2}\sum_{j\in\mathcal{K}}\text{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right)+\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{g}_j^{(r)}\|^2\right]$$

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

$$=\frac{\omega}{k}\sum_{j=1}^{p}q_j\left[\text{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right)+\left\|\mathbf{g}_j^{(r)}\right\|^2\right]+\left[\frac{\tau\sigma^2}{k}+\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right]$$

$$\leq\frac{\omega}{k}\sum_{j=1}^{p}q_j\left[\tau\sigma^2+\left\|\mathbf{g}_j^{(r)}\right\|^2\right]+\left[\frac{\tau\sigma^2}{k}+\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right]$$

$$=(\omega+1)\frac{\tau\sigma^2}{k}+(\frac{\omega}{k}+1)\left[\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2\right] \tag{7}$$

where ① holds due to $\mathbb{E}\left[\|\mathbf{x}\|^2\right]=\text{Var}[\mathbf{x}]+\|\mathbb{E}[\mathbf{x}]\|^2$, 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 **??**, we have

$$\mathbb{E}_{\xi^{(r)}}\left[\left[\|\tilde{\mathbf{g}}_j^{(r)}-\mathbf{g}_j^{(r)}\|^2\right]\right]\leq\tau\sigma^2 \tag{8}$$

To do so, note that

$$\text{Var}\left(\tilde{\mathbf{g}}_j^{(r)}\right)=\mathbb{E}_{\xi^{(r)}}\left[\left\|\tilde{\mathbf{g}}_j^{(r)}-\mathbf{g}_j^{(r)}\right\|^2\right]$$

$$\overset{①}{=}\mathbb{E}_{\xi^{(r)}}\left[\left\|\sum_{c=0}^{\tau-1}\left[\tilde{\mathbf{g}}_j^{(c,r)}-\mathbf{g}_j^{(c,r)}\right]\right\|^2\right]$$

$$=\text{Var}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)$$

$$\overset{②}{=} \sum_{c=0}^{\tau-1} \text{Var}\left(\tilde{\mathbf{g}}_j^{(c,r)}\right)$$

$$= \sum_{c=0}^{\tau-1} \mathbb{E}\left[\left\|\tilde{\mathbf{g}}_j^{(c,r)} - \mathbf{g}_j^{(c,r)}\right\|^2\right]$$

$$\overset{③}{\leq} \tau\sigma^2 \tag{9}$$

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 **??**.

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

$$\mathbb{E}_{\xi^{(r)}|\boldsymbol{w}^{(r)}}\mathbb{E}_{\mathbf{S},\mathcal{K}}\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\|^2\right] \leq (\omega+1)\frac{\tau\sigma^2}{k} + (\frac{\omega}{k}+1)\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(r)}\|^2 \tag{10}$$

Further note that we have

$$\left\|\mathbf{g}_j^{(r)}\right\|^2 = \|\sum_{c=0}^{\tau-1}\mathbf{g}_j^{(c,r)}\|^2 \leq \tau\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2 \tag{11}$$

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

$$\mathbb{E}_{\xi^{(r)}|\boldsymbol{w}^{(r)}}\mathbb{E}_{\mathbf{S}}\left[\|\frac{1}{k}\sum_{j\in\mathcal{K}}\mathbf{S}\left(\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right)\|^2\right] \leq (\omega+1)\frac{\tau\sigma^2}{k} + \tau(\frac{\omega}{k}+1)\sum_{j=1}^{p}q_j\|\mathbf{g}_j^{(c,r)}\|^2, \tag{12}$$

and the proof is complete. $\qquad\square$

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

$$-\mathbb{E}_{\xi,\mathbf{S},\mathcal{K}}\left[\left\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}^{(r)}\right\rangle\right] \leq \frac{1}{2}\eta\frac{1}{m}\sum_{j=1}^{m}\sum_{c=0}^{\tau-1}\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2 - \|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2 + L^2\|\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(c,r)}\|_2^2\right] \tag{13}$$

Proof. We have:

$$-\mathbb{E}_{\{\xi_1^{(t)},\ldots,\xi_m^{(t)}|\boldsymbol{w}_1^{(t)},\ldots,\boldsymbol{w}_m^{(t)}\}}\mathbb{E}_{\mathbf{S},\mathcal{K}}\left[\left\langle\nabla f(\boldsymbol{w}^{(r)}),\tilde{\mathbf{g}}_{\mathbf{S},\mathcal{K}}^{(r)}\right\rangle\right]$$

$$= -\mathbb{E}_{\{\xi_1^{(t)},\ldots,\xi_m^{(t)}|\boldsymbol{w}_1^{(t)},\ldots,\boldsymbol{w}_m^{(t)}\}}\left[\left\langle\nabla f(\boldsymbol{w}^{(r)}),\eta\sum_{j\in\mathcal{K}}q_j\sum_{c=0}^{\tau-1}\tilde{\mathbf{g}}_j^{(c,r)}\right\rangle\right]$$

$$= -\left\langle\nabla f(\boldsymbol{w}^{(r)}),\eta\sum_{j=1}^{m}q_j\sum_{c=0}^{\tau-1}\mathbb{E}_{\xi,\mathbf{S}}\left[\tilde{\mathbf{g}}_{j,\mathbf{S}}^{(c,r)}\right]\right\rangle$$

$$= -\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left\langle\nabla f(\boldsymbol{w}^{(r)}),\mathbf{g}_j^{(c,r)}\right\rangle$$

22

$$\overset{\text{\textcircled{1}}}{=}\frac{1}{2}\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2-\|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2+\|\nabla f(\boldsymbol{w}^{(r)})-\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2\right]$$

$$\overset{\text{\textcircled{2}}}{\le}\frac{1}{2}\eta\sum_{c=0}^{\tau-1}\sum_{j=1}^{m}q_j\left[-\|\nabla f(\boldsymbol{w}^{(r)})\|_2^2-\|\nabla f(\boldsymbol{w}_j^{(c,r)})\|_2^2+L^2\|\boldsymbol{w}^{(r)}-\boldsymbol{w}_j^{(c,r)}\|_2^2\right] \tag{14}$$

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 **??**. $\qquad\square$

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

Lemma 6. *Under Assumptions* **??** *we have:*

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

Proof. Note that

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

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

$$\overset{\text{\textcircled{1}}}{=}\mathbb{E}\left[\left\|\eta\sum_{k=0}^{c}\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+\left[\left\|\eta\sum_{k=0}^{c}\mathbf{g}_j^{(k,r)}\right\|_2^2\right]$$

$$\overset{\text{\textcircled{2}}}{=}\eta^2\sum_{k=0}^{c}\mathbb{E}\left[\left\|\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+(c+1)\eta^2\sum_{k=0}^{c}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]$$

$$\le\eta^2\sum_{k=0}^{\tau-1}\mathbb{E}\left[\left\|\left(\tilde{\mathbf{g}}_j^{(k,r)}-\mathbf{g}_j^{(k,r)}\right)\right\|_2^2\right]+\tau\eta^2\sum_{k=0}^{\tau-1}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]$$

$$\overset{\text{\textcircled{3}}}{\le}\eta^2\sum_{k=0}^{\tau-1}\sigma^2+\tau\eta^2\sum_{k=0}^{\tau-1}\left[\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2\right]$$

$$=\eta^2\tau\sigma^2+\eta^2\sum_{k=0}^{\tau-1}\tau\left\|\mathbf{g}_j^{(k,r)}\right\|_2^2 \tag{15}$$

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 **??**. $\qquad\square$

A.1 Main result for the nonconvex 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 (Nonconvex). *For* FedSKETCH(τ, η, γ), *for all* $0 \leq t \leq R\tau - 1$, *under Assumptions* **??** *to* **??**, *if the learning rate satisfies*

$$1 \geq \tau^2 L^2 \eta^2 + \left(\frac{\omega}{k} + 1\right) \eta\gamma L\tau \tag{16}$$

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:*

$$\frac{1}{R} \sum_{r=0}^{R-1} \left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \frac{2\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right)}{\eta\gamma\tau R} + \frac{L\eta\gamma(\omega + 1)}{k}\sigma^2 + L^2\eta^2\tau\sigma^2 \tag{17}$$

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

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

$$\boldsymbol{w}^{(r)} - \boldsymbol{w}_j^{(\tau,r)} = \eta \sum_{c=0}^{\tau-1} \tilde{\mathbf{g}}_j^{(c,r)}. \tag{18}$$

From the updating rule of Algorithm **??** we have

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

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

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

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

Then using the unbiased estimation property of sketching we have:

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

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

$$f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(r)}) \leq -\gamma\langle\nabla f(\boldsymbol{w}^{(r)}), \tilde{\mathbf{g}}^{(r)}\rangle + \frac{\gamma^2 L}{2}\|\tilde{\mathbf{g}}^{(r)}\|^2 \tag{19}$$

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

$$\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(r)})\right]\right] \leq -\gamma\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[\langle\nabla f(\boldsymbol{w}^{(r)}), \tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\rangle\right]\right] + \frac{\gamma^2 L}{2}\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]$$

$$\stackrel{(a)}{=} -\gamma\underbrace{\mathbb{E}\left[\left[\langle\nabla f(\boldsymbol{w}^{(r)}), \tilde{\mathbf{g}}^{(r)}\rangle\right]\right]}_{(\mathrm{I})} + \frac{\gamma^2 L}{2}\underbrace{\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[\|\tilde{\mathbf{g}}_{\mathbf{S}}^{(r)}\|^2\right]\right]}_{(\mathrm{II})} \tag{20}$$

We proceed to use Lemma **??**, Lemma **??**, and Lemma **??**, to bound terms (I) and (II) in right hand side of Eq. (**??**), which gives

$$
\mathbb{E}\left[\mathbb{E}_{\mathbf{S}}\left[f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(r)})\right]\right]
$$

$$
\leq \gamma \frac{1}{2}\eta \sum_{j=1}^{p} q_j \sum_{c=0}^{\tau-1}\left[-\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 - \left\|\mathbf{g}_j^{(c,r)}\right\|_2^2 + L^2\eta^2 \sum_{c=0}^{\tau-1}\left[\tau\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2 + \sigma^2\right]\right]
$$

$$
+ \frac{\gamma^2 L(\frac{\omega}{k}+1)}{2}\left[\eta^2\tau\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2\right] + \frac{\gamma^2\eta^2 L(\omega+1)}{2}\frac{\tau\sigma^2}{k}
$$

$$
\overset{\text{①}}{\leq} \frac{\gamma\eta}{2}\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\left[-\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 - \left\|\mathbf{g}_j^{(c,r)}\right\|_2^2 + \tau L^2\eta^2\left[\tau\left\|\mathbf{g}_j^{(c,r)}\right\|_2^2 + \sigma^2\right]\right]
$$

$$
+ \frac{\gamma^2 L(\frac{\omega}{k}+1)}{2}\left[\eta^2\tau\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2\right] + \frac{\gamma^2\eta^2 L(\omega+1)}{2}\frac{\tau\sigma^2}{k}
$$

$$
= -\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2
$$

$$
- \left(1 - \tau L^2\eta^2\tau - (\frac{\omega}{k}+1)\eta\gamma L\tau\right)\frac{\eta\gamma}{2}\sum_{j=1}^{p}q_j\sum_{c=0}^{\tau-1}\|\mathbf{g}_j^{(c,r)}\|^2 + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2
$$

$$
\overset{\text{②}}{\leq} -\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2 \tag{21}
$$

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

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

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

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

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

Corollary 4 (Linear speed up). *In Eq. (**??**) 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:*

$$
\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq O\left(\frac{L\sqrt{(\omega+1)}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^*)\right)}{\sqrt{kR\tau}} + \frac{\left(\sqrt{(\omega+1)}\right)\sigma^2}{\sqrt{kR\tau}} + \frac{k\sigma^2}{R\gamma^2}\right). \tag{22}
$$

*Note that according to Eq. (**??**), 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. (**??**) 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. (??) can be rewritten as*

$$\eta \leq \frac{-\gamma L\tau \left(\frac{\omega}{k} + 1\right) + \sqrt{\gamma^2 \left(L\tau \left(\frac{\omega}{k} + 1\right)\right)^2 + 4L^2\tau^2}}{2L^2\tau^2}$$

$$= \frac{-\gamma L\tau \left(\frac{\omega}{k} + 1\right) + L\tau\sqrt{\left(\frac{\omega}{k} + 1\right)^2 \gamma^2 + 4}}{2L^2\tau^2}$$

$$= \frac{\sqrt{\left(\frac{\omega}{k} + 1\right)^2 \gamma^2 + 4} - \left(\frac{\omega}{k} + 1\right)\gamma}{2L\tau} \tag{23}$$

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

$$R \geq \frac{\tau k}{(\omega + 1)\gamma^2 \left(\sqrt{\left(\frac{\omega}{k} + 1\right)^2 \gamma^2 + 4} - \left(\frac{\omega}{k} + 1\right)\gamma\right)^2} \tag{24}$$

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

$$R \geq \frac{\tau k}{5(\omega + 1)} = O\left(\frac{\tau k}{(\omega + 1)}\right) \tag{25}$$

Therefore, for the choice of $\tau = O\left(\frac{\omega+1}{k\epsilon}\right)$, due to condition in Eq. (??), 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. (??) reduces to*

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

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

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

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

A.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 $\texttt{FedSKETCH}(\tau, \eta, \gamma)$, for all $0 \leq t \leq R\tau - 1$, under Assumptions ?? to ?? and ??,if the learning rate satisfies*

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

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

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

Proof. From Eq. (**??**) under condition:

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

we obtain:

$$\mathbb{E}\Big[f(\boldsymbol{w}^{(r+1)}) - f(\boldsymbol{w}^{(r)})\Big] \le -\eta\gamma\frac{\tau}{2}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2$$

$$\le -\eta\mu\gamma\tau\Big(f(\boldsymbol{w}^{(r)}) - f(\boldsymbol{w}^{(r)})\Big) + \frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + \gamma(\omega+1)\right)\sigma^2 \qquad (26)$$

which leads to the following bound:

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

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

$$\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big]$$

$$\le \Delta^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1 - \Delta^R}{1 - \Delta}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2$$

$$\le \Delta^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1}{1 - \Delta}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2$$

$$= (1 - \eta\mu\gamma\tau)^R\Big[f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big] + \frac{1}{\eta\mu\gamma\tau}\frac{L\tau\gamma\eta^2}{2k}\left(kL\tau\eta + (\omega+1)\gamma\right)\sigma^2 \qquad (27)$$

$\square$

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

$$\mathbb{E}\Big[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\Big]$$

$$\le e^{-\eta\gamma\mu\tau R}\Big(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big) + \frac{1}{\mu}\Big[\frac{1}{2}\tau L^2\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\Big]$$

$$\le e^{-\frac{R}{2(\frac{\omega}{k}+1)\kappa}}\Big(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big) + \frac{1}{\mu}\Big[\frac{1}{2}L^2\frac{\tau\sigma^2}{L^2\left(\frac{\omega}{k}+1\right)^2\gamma^2\tau^2} + \frac{(1+\omega)L\sigma^2}{2\left(\frac{\omega}{k}+1\right)L\tau k}\Big]$$

$$= O\left(e^{-\frac{R}{2(\frac{\omega}{k}+1)\kappa}}\Big(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big) + \frac{\sigma^2}{\left(\frac{\omega}{k}+1\right)^2\gamma^2\mu\tau} + \frac{(\omega+1)\sigma^2}{\mu\left(\frac{\omega}{k}+1\right)\tau k}\right)$$

$$= O\left(e^{-\frac{R}{2(\frac{\omega}{k}+1)\kappa}}\Big(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\Big) + \frac{\sigma^2}{\gamma^2\mu\tau} + \frac{(\omega+1)\sigma^2}{\mu\left(\frac{\omega}{k}+1\right)\tau k}\right) \qquad (28)$$

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.

A.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* $\mathtt{FedSKETCH}(\tau, \eta, \gamma)$ *to optimize* $\tilde{f}(\boldsymbol{w}, \phi) = f(\boldsymbol{w}) + \frac{\phi}{2}\|\boldsymbol{w}\|^2$, *for all* $0 \le t \le R\tau - 1$, *under Assumptions* **??** *to* **??**, *if the learning rate satisfies*

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

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:

$$\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \le e^{-\frac{R}{2L\left(1+\frac{\omega}{k}\right)\sqrt{m\tau}}}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right)$$
$$+ \left[\frac{\sqrt{k}\sigma^2}{8\sqrt{\tau}\gamma^2\left(1+\frac{\omega}{k}\right)^2} + \frac{(\omega+1)\sigma^2}{4\left(\frac{\omega}{k}+1\right)\sqrt{k\tau}}\right] + \frac{1}{2\sqrt{k\tau}}\left\|\boldsymbol{w}^{(*)}\right\|^2 \qquad (29)$$

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 **??**, we have:

$$\tilde{f}(\boldsymbol{w}^{(R)}, \phi) - \tilde{f}(\boldsymbol{w}^{(*)}, \phi)$$
$$= f(\boldsymbol{w}^{(r)}) + \frac{\phi}{2}\left\|\boldsymbol{w}^{(r)}\right\|^2 - \left(f(\boldsymbol{w}^{(*)}) + \frac{\phi}{2}\left\|\boldsymbol{w}^{(*)}\right\|^2\right)$$
$$\le (1 - \eta\gamma\phi\tau)^R\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] \qquad (30)$$

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

$$f(\boldsymbol{w}^{(R)}) - f^*$$
$$\le (1 - \eta\gamma\phi\tau)^R\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right]$$
$$+ \frac{\phi}{2}\left(\|\boldsymbol{w}^*\|^2 - \left\|\boldsymbol{w}^{(r)}\right\|^2\right)$$
$$\le e^{-(\eta\gamma\phi\tau)R}\left(f(\boldsymbol{w}^{(0)}) - f(\boldsymbol{w}^{(*)})\right) + \frac{1}{\phi}\left[\frac{1}{2}L^2\tau\eta^2\sigma^2 + (1+\omega)\frac{\gamma\eta L\sigma^2}{2k}\right] + \frac{\phi}{2}\left\|\boldsymbol{w}^{(*)}\right\|^2$$

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

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

thus the proof is complete. $\qquad\qquad\square$

B   Proof of Main Theorems

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

Assumption 6 ([**?**]). *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.,*
$$\mathbb{E}\left[Q(\mathbf{x})\right] = \mathbf{x} \text{ and } \mathbb{E}\left[\|Q(\mathbf{x}) - \mathbf{x}\|^2\right] \leq \omega \|\mathbf{x}\|^2 .$$

B.1   Proof of Theorem ??

Based on Assumption **??** we have:

Theorem 8 ([**?**]). *Consider* `FedCOM` *in [**?**]. Suppose that the conditions in Assumptions **??**, **??** and **??** 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 $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ 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 $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ 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 Assumption **??** with $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively with probability $1 - \delta$. Therefore, all the results in Theorem **??**, conclude from Theorem **??** with probability $1 - \delta$ and plugging $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively into the corresponding convergence bounds. □

B.2   Proof of Theorem ??

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

Assumption 7 ([**?**]). *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 [**?**] here as follows:

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

- **Nonconvex:** *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 satisfy* $\frac{1}{R}\sum_{r=0}^{R-1}\left\|\nabla f(\boldsymbol{w}^{(r)})\right\|_2^2 \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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* $\mathbb{E}\left[f(\boldsymbol{w}^{(R)}) - f(\boldsymbol{w}^{(*)})\right] \leq \epsilon$ *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 methods `PRIVIX` and `HEAPRIX`, satisfy the Assumption **??** with $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively with probablity $1 - \delta$, we conclude the proofs of Theorem **??** using Theorem **??** with probability $1 - \delta$ and plugging $\omega = \mu^2 d$ and $\omega = \mu^2 d - 1$ respectively into the convergence bounds. $\square$