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

A DEFINITIONS AND COMPARISON WITH PRIOR WORKS

Complete appendix can be found in Appendix.

A.1 PRIVIX method and compression error of HEAPRIX

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

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

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

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

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

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

[35]. The reference [35] considers two-way compression from parameter server to devices and vice versa. They provide the convergence rate of $R = O\left(\frac{\omega^{\mathrm{Up}}\omega^{\mathrm{Down}}}{\epsilon^2}\right)$ 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 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 $1 - \delta$.

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

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

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

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