研究DP+Secure Aggregation的问题，与FL Platform无关

The Fundamental Price of Secure Aggregation in Differentially Private Federated Learning

Wei-Ning Chen [* 1 2] **Christopher A. Choquette-Choo** [* 2] **Peter Kairouz** [* 2] **Ananda Theertha Suresh** [* 2]

Abstract

We consider the problem of training a d dimensional model with distributed differential privacy (DP) where secure aggregation (SecAgg) is used to ensure that the server only sees the noisy sum of n model updates in every training round. Taking into account the constraints imposed by SecAgg, we characterize the fundamental communication cost required to obtain the best accuracy achievable under ε central DP (i.e. under a fully trusted server and no communication constraints). Our results show that $\tilde{O}\left(\min(n^2\varepsilon^2, d)\right)$ bits per client are both sufficient and necessary, and this fundamental limit can be achieved by a linear scheme based on sparse random projections. This provides a significant improvement relative to state-of-the-art SecAgg distributed DP schemes which use $\tilde{O}(d\log(d/\varepsilon^2))$ bits per client.

Empirically, we evaluate our proposed scheme on real-world federated learning tasks. We find that our theoretical analysis is well matched in practice. In particular, we show that we can reduce the communication cost to under 1.78 bits per parameter in realistic privacy settings without decreasing test-time performance. Our work hence theoretically and empirically specifies the fundamental price of using SecAgg.

1. Introduction

Federated learning (FL) is a widely used machine learning framework where multiple clients collaborate in learning a model under the coordination of a central server (McMahan et al., 2017a; Kairouz et al., 2021b). One of the primary attractions of FL is that it provides data confidentiality and can provide a level of privacy to participating clients through data minimization: the raw client data never leaves the device, and only updates to models (e.g., gradient updates) are sent back to the central server. This provides practical privacy improvements over centralized settings because updates typically contain less information about the clients, because they are more focused on the learning task, and also only need to be held ephemerally by the server

However, this vanilla federated learning does not provide any formal or provable privacy guarantees. To do so, FL is often combined with differential privacy (DP) (Dwork et al., 2006b). This can be done in one of two ways[1]: 1) perturbing the aggregated (local) model updates at the server before updating the global model, or 2) perturbing each client's model update locally and using a cryptographic multi-party computation protocol to ensure that the server only sees the noisy aggregate. The former is referred to as *central* DP, and it relies on the clients' trust in the server because any sensitive information contained in the model updates is revealed to and temporally stored on the server. The latter is referred to as *distributed* DP (Dwork et al., 2006a; Kairouz et al., 2021a; Agarwal et al., 2021; 2018), and it offers privacy guarantees with respect to an *honest-but-curious* server. Thus, a key technology for formalizing and strengthening FL's privacy guarantees is a secure vector sum protocol called secure aggregation (SecAgg) (Bonawitz et al., 2016b; Bell et al., 2020), which lets the server see the aggregate client updates but not the individual ones.

Despite enhancing the clients' privacy, aggregating model updates via SecAgg drastically increases the computation and communication overheads (Bonawitz et al., 2016b; 2019). This is even worse in federated settings where communication occurs over bandwidth-limited wireless links, and the extra communication costs may become a bottleneck that hampers efficient training of large-scale machine learning models. In fact, high computation can cause failures of SecAgg and prevent FL training of large models or with many clients (see Section 7 for a discussion on the how the communication cost limits the cohort size of FL in practice).

For example, Kairouz et al. (2021a) reports that when train-

[*] Equal contribution [1] Stanford University [2] Google Research. Correspondence to: Wei-Ning Chen <wnchen@stanford.edu>, Christopher A. Choquette-Choo <cchoquette@google.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1] Local DP is yet another alternative, but it incurs higher utility loss and is therefore not typically used in practice.

ing a language model with SecAgg and DP, even with a carefully designed quantization scheme, each client still needs to transmit about 16 bits *per model parameter* each round. Moreover, the bitwidth needs to be scaled up when the privacy requirements are more stringent. This behavior disobeys the conclusion of Chen et al. (2020) (derived under local DP), which shows that the optimal communication cost should *decay* with the privacy budget, i.e., data is more compressible in the high privacy regime.

Furthermore, because the server aggregates the model updates via SecAgg, we can only compress the model updates locally using linear schemes. This constraint rules out many popular compression schemes such as entropy encoders or gradient sparsification (Aji & Heafield, 2017; Lin et al., 2017; Wangni et al., 2017; Havasi et al., 2018; Oktay et al., 2019) etc., as these methods are non-linear.

Therefore, it is unclear whether or not the communication cost of SecAgg reported in Kairouz et al. (2021a); Agarwal et al. (2021; 2018) is fundamental. If not, what is the smallest communication needed to achieve distributed DP via secure aggregation with the same performance as in the centralized DP setting?

In this paper, we answer the above question, showing that the communication costs of existing mechanisms are strictly sub-optimal in the distributed mean estimation (DME) task (Suresh et al., 2017) (see Section 4 for details). We also propose a SecAgg comaptible *linear* compression scheme based on sparse random projections (Algorithm 2), and then combine it with the distributed discrete Gaussian (DDG) mechanism proposed by Kairouz et al. (2021a). Theoretically, we prove that our scheme requires $\tilde{O}\left(\min(n^2\varepsilon^2, d)\right)$ bits per client, where n is the per-round number of clients. This cost is significantly smaller than the communication cost of previous schemes which was $\tilde{O}_\delta(d\log(d/\varepsilon^2))$ bits per client[2]. To give perspective, (1) n is usually on the order of 10^3 per round due to the computational overhead of SecAgg, and (2) ε is the privacy budget for a single round, i.e., $\varepsilon \approx \varepsilon_{\mathsf{final}}/\sqrt{R}$ if there are R training rounds. Thus, for practical FL settings where large models are trained with SecAgg over many rounds, $n^2\varepsilon^2$ is typically (much) smaller than d.

We complement our achievability results with a matching lower bound, showing that to obtain an unbiased estimator of the mean vector, each client needs to communicate $\tilde{\Omega}(\min(n^2\varepsilon^2, d))$ bits with the server. Our upper and lower bounds together specify the fundamental privacy-communication-accuracy trade-offs under SecAgg and DP.

In addition, we show that with additional sparsity assumptions, we can further improve both the accuracy and communication efficiency while achieving the same privacy requirement, leading to a logarithmic dependency on d.

Empirically, we verify our scheme on a variety of real-world FL tasks. Compared to existing distributed DP schemes, we observe 10x or more compression with no significant decrease in test-time performance. Moreover, the compression rates can be made even higher with tighter privacy constraints (i.e., with smaller ε), complying with our theoretical $\tilde{O}(\min(n^2\varepsilon^2, d)$ communication bound.

Organization The rest of this paper is organized as follows. We summarize related work in Section 2 and introduce necessary preliminaries in Section 3. We then provide a formal problem formulation in Section 4. Next, we present and analyze the performance of our main scheme (in terms of privacy, utility, and communication efficiency) and prove its optimality in Section 5.1. After that, we show, in Section 6, that with additional sparsity assumptions, one can simultaneously reduce the communication cost and increase the accuracy. Finally, we present our experimental results in Section 7 and conclude the paper in Section 8. The code is available at `https://github.com/google-research/federated/tree/master/private_linear_compression`.

2. Related Work

SecAgg and distributed DP SecAgg is cryptographic secure multi-party computation (MPC) that allows the server to collect the sum of n vectors from clients without knowing anyone of them. In our single-server FL setting, SecAgg is achieved via additive masking over a finite group (Bonawitz et al., 2016a; Bell et al., 2020). Similar ideas have been used in other secure aggregation protocols (Fereidooni et al., 2021; Kadhe et al., 2020). However, the vanilla FL with SecAgg does not provide provable privacy guarantees since the sum of updates may still leak sensitive information (Melis et al., 2019; Song & Shmatikov, 2019; Carlini et al., 2019; Shokri et al., 2017). To address this issue, differential privacy (DP) (Dwork et al., 2006a), and in particular, DP-SGD or DP-FedAvg can be employed (Song et al., 2013; Bassily et al., 2014; Geyer et al., 2017; McMahan et al., 2017b). In this work, we aim to provide privacy guarantees in the form of Rényi DP (Mironov, 2017) because it allows for accounting end-to-end privacy loss tightly.

We also distinguish our setup from the local DP setting (Kasiviswanathan et al., 2011; Evfimievski et al., 2004; Warner, 1965), where the data is perturbed on the client-side before it is collected by the server. Local DP, which allows for a possibly malicious server, is stronger than distributed DP, which assumes an honest-but-curious server. Thus, local DP suffers from worse privacy-utility tarde-offs (Kasiviswanathan et al., 2011; Duchi et al., 2013; Kairouz et al., 2016).

[2] For simplicity, we use the $\tilde{O}_\delta\left(\cdot\right)$ notation to hide the dependency on δ and $\log n$

Model compression, sketching, and random projection There has been a significant amount of recent work on reducing the communication cost in FL, see (Kairouz et al., 2019). Among them, popular compression approaches include gradient quantization (Alistarh et al., 2017; Bernstein et al., 2018), sparsification (Aji & Heafield, 2017; Lin et al., 2017; Wangni et al., 2017), and entropy encoders (Havasi et al., 2018; Oktay et al., 2019). However, since these schemes are mostly non-linear, they cannot be combined with SecAgg where all the encoded messages will be summed together. Therefore, in this work, we resort to compression schemes with *linear* encoders. The only exception are sketching based methods (Rothchild et al., 2020; Haddadpour et al., 2020). Our work differs from them in three aspects. First, we consider FL with privacy and SecAgg, whereas (Rothchild et al., 2020) only aims at reducing communication. Second, although we use the same count-sketch encoder, our decoding method is more aligned with the sparse random projection (Kane & Nelson, 2014). In the language of sketching, we decode the sketched model updates by "count-mean" instead of count-median, which improves space efficiency, thus requiring less memory to train a real-world large-scale machine learning model. We note that both count-sketch and random projection provides the same worst-case ℓ_2 error bounds.

FL with SecAgg and distributed DP The closest works to ours are cpSGD (Agarwal et al., 2018), DDG (Kairouz et al., 2021a), and Skellam (Agarwal et al., 2021), which serve as the main inspiration of this paper. However, all of these methods rely on per parameter quantization and thus lead to $\tilde{\Omega}\left(d\right)$ communication cost. In this work, however, we show that when $d \gg n^2\varepsilon^2$, we can further reduce dimensionality and achieve the optimal communication cost $\tilde{O}(n^2\varepsilon^2)$ in this regime. Our scheme also demonstrates 10x or more compression rates (depending on the privacy budget) relative to the best existing distributed DP schemes.

3. Preliminaries

3.1. Differential Privacy

We begin by providing a formal definition for (ε, δ)-differential privacy (DP) (Dwork et al., 2006b).

Definition 3.1 (Differential Privacy)**.** For $\varepsilon, \delta \geq 0$, a randomized mechanism M satisfies (ε, δ)-DP if for all neighboring datasets D, D' and all $\mathcal{S}$ in the range of M, we have that

$$\Pr\left(M(D) \in \mathcal{S}\right) \leq e^{\varepsilon} \Pr\left(M(D') \in \mathcal{S}\right) + \delta,$$

where D and D' are neighboring pairs if they can be obtained from each other by adding or removing all the records that belong to a particular user.

The above DP notion is referred to as user level DP and is stronger than the commonly-used item level DP, where, if a user contributes multiple records, only the addition or removal of one record is protected.

We also make use of Renyi differential privacy (RDP) which allows for tight privacy accounting.

Definition 3.2 (Renyi Differential Privacy)**.** A randomized mechanism M satisfies (α, ε)-RDP if for any two neighboring datasets D, D', we have that $D_\alpha\left(P_{M(D)}, P_{M(D')}\right) \leq \varepsilon$ where $D_\alpha\left(P, Q\right)$ is the Renyi divergence between P and Q and is given by

$$D_\alpha\left(P, Q\right) \triangleq \frac{1}{\alpha}\log\left(\mathbb{E}_Q\left[\left(\frac{P(X)}{Q(X)}\right)^{\alpha}\right]\right).$$

Notice that one can convert $(\alpha, \varepsilon(\alpha))$-RDP to $(\varepsilon_{\mathsf{DP}}(\delta), \delta)$-DP. See Leema G.1 in Section G.1.

3.2. The Distributed Discrete Gaussian Mechanism

The previous work of Kairouz et al. (2021a) proposed a scheme based on the discrete Gaussian mechanism (denoted as `DDG`) which achieves the best mean square error (MSE) $O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$ with a $\frac{1}{2}\varepsilon^2$-concentrated differential privacy guarantee. The encoding scheme mainly consists of the following four steps: (a) scaling, (b) flattening via random rotation, (c) conditional randomized rounding, and (d) perturbation, which we summarize in Algorithm 1 below.

Algorithm 1 The DDG mechanism

```
Inputs: Private vector $x_i \in \mathbb{R}^d$; clipping threshold $c$; modulus $M \in \mathbb{N}$; noise scale $\sigma > 0$;
Clip and scale $x_i$ so that $\|x_i'\|_2 < c$
Flatten vector by a random rotation: $x_i'' = U_{\mathsf{rotate}} x_i'$
Stochastically round and discretize $x_i''$ into $x_i''' \in \mathbb{Z}^d$
$Z_i = x_i''' + \mathcal{N}_{\mathbb{Z}}(0, \sigma^2) \mod M$, where $\mathcal{N}_{\mathbb{Z}}$ is the discrete Gaussian noise
Return: $Z_i \in \mathbb{Z}_M^d$
```

Upon aggregating $\hat{\mu}_z = \sum_{i \in [n]} Z_i$, the server can rotate $\hat{\mu}_z$ reversely and re-scale it back to decode the mean $\hat{\mu} = \frac{1}{n}\sum_i x_i$. We refer the reader to Algorithm 5 for a detailed version of DDG. By picking the parameters properly (see Theorem F.1), Algorithm 5 has the following properties:

- Satisfies $(\alpha, \frac{\varepsilon^2}{2}\alpha)$-RDP, which implies $(\varepsilon_{\mathsf{DP}}, \delta)$-DP with $\varepsilon_{\mathsf{DP}} = O_\delta\left(\varepsilon^2\right)$
- Uses $O\left(d\log\left(n + \sqrt{\frac{n^3\varepsilon^2}{d}} + \frac{\sqrt{d}}{\varepsilon}\right)\right)$ bits of per client
- Has an MSE of $\mathbb{E}\left[\|\hat{\mu} - \mu\|_2^2\right] = O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$.

3.3. Sparse random projection and count-sketch

We now provide background on sparse random projection (Kane & Nelson, 2014) and count-sketching, which allows us to reduce the dimension of local gradients from $\mathbb{R}^d$ to $\mathbb{R}^m$ with $m \ll d$. These schemes are *linear*, making them compatible with SecAgg.

Let $S_1, ..., S_t \in \{-1, 0, 1\}^{d \times w}$ be t identical and independent count-sketch matrices, that is,

$$(S_i)_{j,k} = \sigma_i(j) \cdot \mathbb{1}_{\{h_i(j)=k\}}, \tag{1}$$

for some independent hash functions $h_i : [d] \to [w]$ and $\sigma_i : [d] \to \{-1, +1\}$. Let $m = t \times w$, then the sparse random projection matrix $S \in \mathbb{R}^{d \times m}$ is then defined as stacking $S_1, ..., S_t$ vertically, that is,

$$S^\intercal = \frac{1}{\sqrt{t}} \left[S_1^\intercal, S_2^\intercal, ..., S_t^\intercal\right]. \tag{2}$$

Under this construction, S is sparse in the sense that each column contains exactly t 1s. Moreover, S possesses several nice properties (see Section G.2 for some of them that will be used in our proofs).

4. Problem Formulation

We start by formally presenting the distributed mean estimation (DME) (Suresh et al., 2017) problem under differential privacy. Note that DME is closely related to the federated averaging (FedAvg) algorithm (McMahan et al., 2017a), where in each round, the server updates the global model using a noisy estimate of the mean of local model updates. Such a noisy estimate is typically obtained via a DME mechanism, and thus one can easily build a DP-FedAvg scheme from a DP-DME scheme.

Consider n clients each with a data vector $x_i \in \mathbb{R}^d$ that satisfies $\|x_i\|_2 \leq c$ (e.g., a clipped local model update). After communicating with n clients, a server releases a noisy estimates $\hat{\mu}$ of the mean $\mu \triangleq \frac{1}{n}\sum_i x_i$, such that 1) $\hat{\mu}$ satisfies a differential privacy constraint (see Definition 3.1 and Definition 3.2), and 2) $\mathbb{E}\left[\|\hat{\mu} - \mu\|_2^2\right]$ is minimized. The goal is to design a communication protocol (which includes local encoders and a central decoder) and an estimator $\hat{\mu}$.

In this paper, we consider two different DP settings. The first is the *centralized* DP setting: the server has access to all x_i's, i.e., $\hat{\mu} = \hat{\mu}\left(x_1, ..., x_n\right)$. The second is the *distributed* DP via SecAgg setting: under this setting, each client is subject to a b-bit communication constraint, so they must first encode x_i into a b-bit message, i.e., $Z_i = \mathcal{A}_{\mathsf{enc}}(x_i) \in \mathcal{Z}$ with $|\mathcal{Z}| \leq 2^b$ (See Figure 1 for an illustration). However, instead of directly collecting $Z_1, ..., Z_n$, the server can only observe the *sum* of them, so the estimator must be a function of $\sum_{i=1}^n Z_i$ (i.e., $\hat{\mu} = \hat{\mu}\left(\sum_{i=1}^n Z_i\right)$). Moreover, we require that the sum $\sum_i Z_i$ satisfies DP (which is stronger than requiring $\hat{\mu}$ to be DP), meaning that individual information will not be disclosed to the server as well. Notice that since SecAgg operates on a finite additive group, we require $\mathcal{Z}$ to have an additive structure. Without loss of generality, we will set $\mathcal{Z}$ to be $\left(\mathbb{Z}_M\right)^m$ for some $m, M \in \mathbb{N}$, where $\mathbb{Z}_M$ denotes the group of integers modulo M (equipped with modulo M addition) and m is the dimension of the space we are projecting onto. In other words, we allocate $\log M$ bits for every coordinate of the projected vector. Note that in this case, the total per-client communication cost is $b = m \log M$.

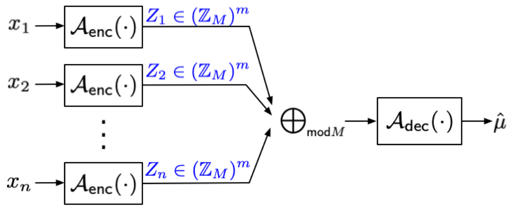

x_1
$\mathcal{A}_{\mathsf{enc}}(\cdot)$
$Z_1 \in (\mathbb{Z}_M)^m$
x_2
$\mathcal{A}_{\mathsf{enc}}(\cdot)$
$Z_2 \in (\mathbb{Z}_M)^m$
$\vdots$
x_n
$\mathcal{A}_{\mathsf{enc}}(\cdot)$
$Z_n \in (\mathbb{Z}_M)^m$
$\bigoplus_{\mathsf{mod}M}$
$\mathcal{A}_{\mathsf{dec}}(\cdot)$
$\hat{\mu}$

Figure 1. Private mean estimation via SecAgg.

For a fixed privacy constraint, the fundamental problem we seek to solve is: what is the smallest communication cost under the distributed DP setting needed to achieve the accuracy of the centralized DP setting? Further, we seek to discover schemes that (a) achieve the optimal privacy-accuracy-communication trade-offs and (b) are memory efficient and computationally fast in encoding and decoding.

5. Communication cost of DME with SecAgg

In this section, we characterize the *optimal communication cost* under the distributed DP via SecAgg setting, defined as the smallest number of bits (as a function of n, d, ε) needed to achieve the same accuracy (up to a constant factor) of the centralized setting under the same (ε, δ)-DP constraint. The optimal communication can be achieved by leveraging a sparse random projection, and the proposed scheme is summarized in Algorithm 2. Our main theoretical results are summarized in the following theorem.

Theorem 5.1. *Let $m = \Theta(n^2\varepsilon^2)$ and $t = \Theta\left(\log d + \log n\right)$ in Algorithm 2. Assume $\|x_i\| \leq c$ for all $i \in [n]$. Then as long as $n^2\varepsilon^2 \leq d$, the following holds:*

- *Algorithm 2 satisfies $\left(\alpha, \frac{\varepsilon^2}{2}\alpha\right)$-RDP,*
- *the MSE is bounded by $\mathbb{E}\left[\|\hat{\mu} - \mu\|_2^2\right] = O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$,*
- *the per-client communication is*

$$O\left(m \log\left(n + \sqrt{\frac{n^3\varepsilon^2}{m}} + \frac{\sqrt{m}}{\varepsilon}\right)\right) = O\left(n^2\varepsilon^2 \log n\right).$$

Road map of Section 5.1. Our theoretical analysis of the communication cost of distributed DP via SecAgg proceeds as follows. First, we highlight the best MSE achievable under central DP (where the server is trusted), without any communication constraints. Next, we derive the communication costs of existing distributed DP mechanisms that maintain this optimal MSE and argue that they are strictly sub-optimal. As a warm up, we derive the smallest communication required to match the best MSE without the additional constraints and challenges of distributed DP via SecAgg. We finally show that the same approach can be used with SecAgg to provide optimal communication efficiency. This is done by calculating the achievable MSE under the presented approach and proving a matching lower bound.

Optimal MSE under central DP We start by specifying the optimal accuracy under a fully trusted server and no communication constraints. Under a DME setting where $\|x_i\|_2 \leq c$ for all $i \in [n]$, the ℓ_2 sensitivity of the mean query $\mu(x_1, ..., x_n) \triangleq \frac{1}{n}\sum_{i=1}^n x_i$ is bounded by

$$\mathsf{S}_{\ell_2}(\mu) \triangleq \max_{x^n, x_1'} \|\mu(x_1,...,x_n) - \mu(x_1',...,x_n)\|_2^2 \leq \frac{2c^2}{n}.$$

Therefore, to achieve (ε, δ) central DP, the server can add coordinate-wise independent Gaussian noise to μ. This gives an ℓ_2 error that scales as $O_\delta\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$, which is known to be tight in the high privacy regime (Kamath & Ullman, 2020). Moreover, the resulting estimator is *unbiased*[3].

Communication costs of DDG Next, we examine the communication costs of previous distributed DP schemes such as the DDG mechanism. As mentioned in Section 3.2 and Theorem F.1, in order to achieve the $O_\delta\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$ error, the communication cost of DDG must to be at least $\Theta\left(d\log\left(d/\varepsilon^2\right)\right)$ bits. Note that the communication cost scales up with $1/\varepsilon$ because in the high-privacy regimes, the noise variance needs to be increased accordingly to provide stronger DP. Thus SecAgg's group size needs to be enlarged to capture the larger signal range and avoid catastrophic modular clipping errors. A similar phenomenon occurs for other additive noise-based mechanisms, e.g, the Skellam and binomial (Agarwal et al., 2018; 2021) mechanisms.

However, we show that the $\Theta\left(d\log\left(d/\varepsilon^2\right)\right)$ cost is strictly sub-optimal. In particular, the linear dependency on d can be further improved when $n^2\varepsilon^2 \ll d$. To demonstrate this, we start by the following example to show that under a (ε, δ) central DP constraint, one can reduce the dimensionality to $m = O(n^2\varepsilon^2)$ without harming the MSE.

Dimensionality reduction under central DP Consider the following simple project-and-perturb mechanism:

1. The server generates a random projection matrix $S \in \mathbb{R}^{m\times d}$ according to the sparse random projection defined in Section 3.3 and broadcasts it to n clients.
2. Each client sends $y_i \triangleq \mathsf{clip}_{\ell_2, 1.1c}(Sx_i)$.
3. The server computes $\hat{\mu} \triangleq S^\intercal\left(\frac{1}{n}\sum_i y_i + N\left(0, \sigma^2\mathbb{I}_m\right)\right)$, where $\sigma^2 = \Theta\left(\frac{c^2}{n^2\varepsilon^2}\right)$.

We claim that the above project-and-perturb approach satisfies (ε, δ) DP and achieves the optimal MSE order. In other words, we can reduce the dimensionality *for free*.

To see why this is true, observe that we can decompose the overall ℓ_2 error $\|\hat{\mu} - \mu\|_2^2$ into three parts: (1) the clipping error (i.e., $\|y_i - Sx_i\|_2^2$), (2) the compression error (i.e. $\|\mu - S^\intercal S\left(\sum_i x_i/n\right)\|_2^2$), and (3) the privatization error $\left\|S^\intercal N(0, \sigma^2\mathbb{I}_m)\right\|_2^2$. Then, we argue that all of them have orders less than or equal to $O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$, as long as we select $m = \Theta(n^2\varepsilon^2)$ and $t = \Theta\left(\log d + \log(n^2\varepsilon^2)\right)$.

First, the clipping error is small since the random projection S satisfies the Johnson-Lindenstrauss (JL) property (see Lemma G.3 in the appendix), which implies that $\|Sx_i\|_2 \approx \|x_i\|_2 \leq c$ and that the clipping happens with exponentially small probability. Second, Lemma G.2 (in the appendix) suggests that compression error scales as $O\left(\frac{c^2 d}{m}\right)$. Thus by picking $m = \Theta\left(n^2\varepsilon^2\right)$, we ensure the compression error to be at most $O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$. Finally, since the Gaussian noise N added at Step 3 is independent of S, the privatization error can also be bounded by $O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$.

We summarize this in the following lemma, where the formal proof is deferred to Section L.2 in the appendix.

Lemma 5.2. *Assume $\|x_i\|_2 \leq c$ for all $i \in [n]$. Then the output of the above mechanism $\hat{\mu}$ satisfies (ε, δ)-DP. Moreover, if S (defined in (2)) is generated with $m = \Theta(n^2\varepsilon^2)$ and $t = \Theta\left(\log d + \log(n^2\varepsilon^2)\right)$, it holds that $\mathbb{E}\left[\|\hat{\mu} - \mu\|_2^2\right] = O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$.*

Dimensionality reduction with SecAgg The above example shows that with a random projection, we can reduce the dimensionality from d to $O(n^2\varepsilon^2)$ without increasing the MSE too much. Thus, under the distributed DP via SecAgg setting, we combine random projection with the DDG mechanism to arrive at our main scheme in Algorithm 2.

To control the ℓ_2 error of Algorithm 2, we adopt the same strategy as in Lemma 5.2 (i.e., decompose the end-to-end error into three parts), with the privatization error being replaced by the error due to DDG. However, this will cause

[3] Notice that in this work, we are mostly interest in unbiased estimators (see Remark 5.5 for a discussion).

Algorithm 2 Private DME with random projection

```
Input: Cleints' data $x_1, ..., x_n \in \mathbf{B}_d(c)$, compression parameter $m \in \mathbb{N}$
The server generates a sketching matrix $S \in \mathbb{R}^{m \times d}$
The server broadcasts $S$ to all clients
for $i \in [n]$ do
    Client $i$ computes $y_i \triangleq Sx_i$ and $Z_i \triangleq \mathsf{DDG}_{\mathsf{enc}}\left(y_i\right) \in \mathbb{Z}_M^m$ with $\ell_2$ clipping parameter $1.1c$ (and other parameters being the same as in Algorithm 1 with $d$ being replaced by $m$)
end for
The server aggregates $Z_1, ..., Z_n$ with SecAgg and decodes $\hat{\mu}_y = \frac{1}{n}\mathsf{DDG}_{\mathsf{dec}}\left(\sum_{i\in[n]} Z_i\right)$
The server computes $\hat{\mu} = S^\intercal \hat{\mu}_y$.
Return: $\hat{\mu}$
```

an additional challenge, as the error due to DDG is no longer independent of S (as opposed to the Gaussian noise in the previous case). To overcome this difficulty, we leverage the fact that *for any* projection matrix S, the (expected) ℓ_2 error is bounded by $O\left(\frac{c^2 m}{n^2\varepsilon^2}\right)$ and develop an upper bound on the final MSE accordingly (see Section I for a formal proof).

5.1. Lower bounds for private DME with SecAgg

We complement our achievability result in Theorem 5.1 with a matching communication lower bound. Our lower bound indicates that Algorithm 2 is indeed optimal (in terms of communication efficiency), hence characterizing the fundamental privacy-communication-accuracy trade-offs.

The lower bound leverages the fact that under SecAgg, the *individual* communication budget each client has is equal to the *total* number of bits the server can observe, which are all equal to the cardinality of the finite group $\mathcal{Z}$ that SecAgg acts on. Therefore, even if each client sends a b-bit message (so the total information transmitted is $n \cdot b$ bits), the server still can only observe b bits information.

With this in mind, to give a per-client communication lower bound under SecAgg, it suffices to lower bound the total number of bits needed for reconstructing a d-dim vector (i.e. the mean vector μ) within a given error (i.e. $O_\delta\left(c^2 d/(n^2\varepsilon^2)\right)$). Towards this end, we first derive a general lower bound that characterizes the communication-accuracy trade-offs for compressing a *single* d-dim vector in a centralized setting.

Theorem 5.3 (compression lower bounds). *Let $v \in \mathbb{R}^d$ and $\left\|v\right\|_2 \leq c$. Then for any (possibly randomized) compression operator $\mathcal{C} : \mathbf{B}_d(c) \to [2^b]$ that compresses v into b bits and any (possibly randomized) estimator $\hat{v} : [2^b] \to \mathbf{B}_d(c)$, it holds that*

$$\min_{(\mathcal{C},\hat{v})} \max_{v\in\mathbf{B}_d(c)} \mathbb{E}\left[\left\|\hat{v}\left(\mathcal{C}(v)\right) - v\right\|_2^2\right] \geq c^2 2^{-2b/d}. \quad (3)$$

Moreover, for any unbiased compression scheme (i.e., $(\mathcal{C}, \hat{v})$ satisfying $\mathbb{E}\left[\hat{v}\left(\mathcal{C}(v)\right)\right] = v$ for all $v \in \mathbf{B}_d(c)$) with $b < d$, it holds that

$$\min_{(\mathcal{C},\hat{v})} \max_{v\in\mathbf{B}_d(c)} \mathbb{E}\left[\left\|\hat{v}\left(\mathcal{C}(v)\right) - v\right\|_2^2\right] \geq C_0 c^2 d/b, \quad (4)$$

where $C_0 > 0$ is a universal constant.

Theorem 5.3, together with the fact that the per-client bit budget is also the total amount of information the server can observe, we arrive at the following lower bound.

Corollary 5.4. *Consider the private DME task with SecAgg as described in Figure 1. For any encoding function $\mathcal{A}_{\mathsf{enc}}(\cdot)$ with output space $\mathcal{Z}$, if the ℓ_2 estimation error $\mathbb{E}\left[\left\|\hat{\mu} - \mu\right\|_2^2\right] \leq \xi$ for all possible $x_1, ..., x_n \in \mathbf{B}_d(c)$, then it must hold that $\log|\mathcal{Z}| = \Omega\left(d\log\left(c^2/\xi\right)\right)$. In addition, if $\hat{\mu}$ is unbiased and $\xi \leq c^2$, then $\log|\mathcal{Z}| = \Omega\left(dc^2/\xi\right)$.*

Finally, by plugging $\xi = O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$ (which is the optimal ℓ_2 error for the mean estimation task under centralized DP model), we conclude that

- $\Omega\left(\max\left(d\log\left(n^2\varepsilon^2/d\right), 1\right)\right)$ bits of communication are necessary for general (possibly biased) schemes
- $\Omega\left(\min\left(n^2\varepsilon^2, d\right)\right)$ bits of communication are necessary for unbiased schemes.

We remark that the above lower bounds are both tight but in different regimes. Specifically, the first lower bound, which measures the accuracy in MSE, is tight for small d but is meaningless in high-dimensional or high-privacy regimes where $d \gg n^2\varepsilon^2$. This also implies that $\tilde{\Omega}(d)$ bits are necessary for $d \ll n^2\varepsilon^2$ and that there is no room for improvement on DDG in this regime. On the other hand, the second bound is useful when $d = \Omega\left(n^2\varepsilon^2\right)$ (with an additional unbiasedness assumption). This is a more practical regime for FL with SecAgg, and our scheme outperforms DDG in this scenario.

Remark 5.5. Notice that in this work, we are mostly interested in unbiased estimators due to the following two reasons: 1) it largely facilities the convergence analysis of the SGD based methods, as these types of stochastic first-order methods usually assume access to an unbiased gradient estimator in each round. 2) In the high-dimensional or high-privacy regimes where $d \gg n^2\varepsilon^2$, the MSE is not the right performance measure since an estimator can have a large bias while still achieving a relative small MSE. For instance, in the regime where $n^2\varepsilon^2 \leq d$, the Gaussian mechanism has MSE $\Theta\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$; on the other hand, the trivial estimator

$\hat{\mu} = 0$ achieves a smaller MSE, equal to $c^2 \leq O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$, but such estimator gives no meaningful information. In order to rule out these impractical schemes, we hence impose the unbiasedness constraint.

6. Sparse DME with SecAgg and DP

Theorem 5.1 in Section 5.1 specifies the optimal trade-offs of private DME *for all* possible datasets x^n. In other words, it provides a *worst-case* (over all possible x_i) bound on the utility and shows that Algorithm 2 is worst-case optimal. However, we show in this section that with additional assumptions on the data, it is possible to improve the trade-offs beyond what is given in Theorem 5.1.

One such assumption is *sparsity* of the data, which is justified by several empirical results that gradients tend to be (or are close to being) sparse. We hence study the sparse DME problem, which is formulated as in Section 4 but with an additional s-sparsity assumption on μ, i.e., $\|\mu\|_0 \leq s$. We present a sparse DME algorithm adapted from Algorithm 2, showing that by leveraging the sparse structure of data, one can surpass the lower bound in Theorem 5.1. Moreover, the dependency of communication cost and MSE on the model size d becomes logarithmic.

DME via compressed sensing We adopt the same strategy as in Section 5.1, (i.e., use a linear compression scheme to reduce dimensionality). However, instead of applying the linear decoder $S^\intercal \hat{\mu}_y$, we perform a more complicated compressed sensing decoding procedure and solve a (regularized) linear inverse problem. Specifically, we modify Algorithm 2 in the following way:

1. For local compression, we replace the sparse random projection matrix $S \in \mathbb{R}^{m\times d}$ with an s-RIP[4] matrix (in particular, we use a Gaussian ensemble, i.e., $S_{i,j} \overset{\text{i.i.d.}}{\sim} N(0,1)$) and set $m = O\left(s \log d\right)$.
2. To decode $\hat{\mu}$, the server solves the following ℓ_1 regularized linear inverse problem (i.e., LASSO (Tibshirani, 1996)) $\hat{\mu} \in \arg\min_{x\in\mathbb{R}^d} \left\{\frac{1}{m}\left\|\hat{\mu}_y - Sx\right\|_2 + \lambda_n \left\|x\right\|_1\right\}$, where the λ_n is set to be of the order $O\left(\frac{c\log d}{n\varepsilon}\right)$.

With the above modifications (where the detailed steps is given in Algorithm 6 in the appendix), we can obtain the following privacy and utility guarantees.

Theorem 6.1 (Sparse private DME)**.** *Algorithm 6 satisfies $(\alpha, \frac{1}{2}\varepsilon^2\alpha)$-RDP. If $\|\mu\|_0 \leq s$, it holds that (1) the per-client communication cost is $O\left(s\log d \log\left(n^2 + s\log d/\varepsilon^2\right)\right)$, and (2) the MSE is bounded by $O\left(\frac{c^2 s\log^2 d}{n^2\varepsilon^2}\right)$.*

Observe that under sparsity, both the accuracy and the communication cost depend on d logarithmically. This implies that by leveraging the sparsity, we can replace d with an "effective" dimension of $s\log d$. However, Algorithm 6 is more complicated than Algorithm 2 as it requires tuning hyper-parameters such as s and λ_n.

Remark 6.2. The communication cost in Theorem 6.1 no longer depends only on $n\varepsilon$, thus exhibiting a different behavior from the non-sparse case (i.e., that of Theorem 5.1). We remark that this is because we only present the result for the $s\log d \ll n^2\varepsilon^2$ (which is more reasonable in practice). However, one can extend the similar analysis in Section 5.1 and obtain the results for $s\log d \gg n^2\varepsilon^2$ regime.

7. Empirical Analysis

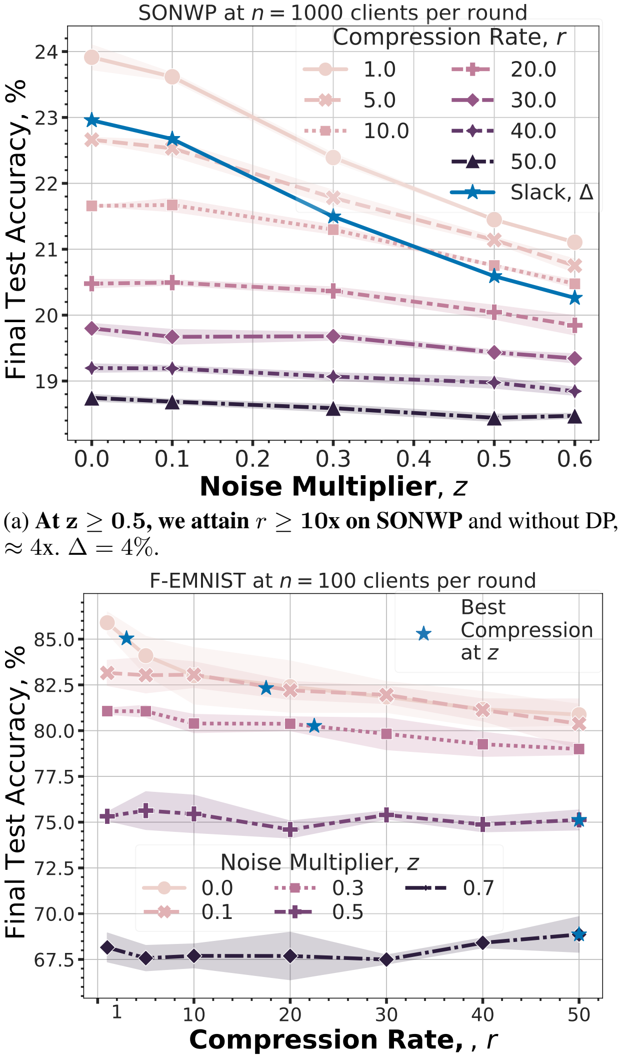
We run experiments on the full Federated EMNIST, Stack Overflow dataset, and Shakespare three common benchmarks for FL taskss (Caldas et al., 2018). F-EMNIST has 62 classes and $N = 3400$ clients with a total of $671,585$ training samples. Inputs are single-channel $(28, 28)$ images. The Stack Overflow (SO) dataset is a large-scale text dataset based on responses to questions asked on the site Stack Overflow. The are over 10^8 data samples unevenly distributed across $N = 342,477$ clients. We focus on the next word prediction (NWP) task: given a sequence of words, predict the next words in the sequence. Shakespeare is similar to SONWP but is focused on character prediction and instead built from the collective works of Shakespeare, partitioned so that each client is a speaking character with at least two lines. There are $n = 715$ characters (clients) with $16,068$ training samples and $2,356$ test samples. On all datasets, we select $n \in [100, 1000]$ and $R = 1500$. We focus our exposition on SONWP and F-EMNIST as two canonical federated datasets but find similar results for Shakespeare in Appendix B (Figures 5 and 6).

On F-EMNIST, we experiment with a $\approx 10^6$ parameter (4 layer) Convolutional Neural Network (CNN) used by Kairouz et al. (2021a). On SONWP, we experiment with a $\approx 4\cdot 10^6$ parameter (4 layer) long-short term memory (LSTM) model—the same as prior work (Andrew et al., 2019; Kairouz et al., 2021a). In both cases, clients train for 1 local epoch using SGD. Only the server uses momentum. For distributed DP, we use the geometric adaptive clipping of (Andrew et al., 2019). We use the same procedure as Kairouz et al. (2021a). We flatten using the Discrete Fourier Transform. We use their hyperparameter values for conditional randomized rounding and modular clipping. We communicate 16 bits per parameter for F-EMNIST and 18 for SONWP unless otherwise indicated. We repeat all experiments with 3 different seeds. We provide full detail on the models, datasets, and training setups in Appendix C, as well as the chosen values of the noise multiplier and the sparse random projection parameters in Appendix D. We provide

[4] See Definition K.1 for a weaker definition.

details of our algorithms for sparse random projections (via sketching) in Appendix E.

Distributed Mean Estimation To verify our theoretical analysis, we run mean estimation experiments with the same setup as (Kairouz et al., 2021a). Each of n clients hold a Gaussian vector v normalized so that $\|v\|_2 \leq 1$ and with dimensionality d, i.e., $|v| = d$. We use $b = 18$ bit width for distributed DP. Recall from Theorem 5.3 that as either the noise multiplier z or the vector size d increases, the compression rate r should increase. We observe exactly this behaviour in Figure 4 of Appendix B.



(a) **At z $\geq$ 0.5, we attain $r \geq$ 10x on SONWP** and without DP, $\approx$ 4x. $\Delta = 4\%$.

(b) **At z $\geq$ 0.5 we attain $r \geq$ 50x on F-EMNSIT**. Without DP ($z = 0$), we cannot significantly compress. $\Delta = 1\%$.

Figure 2. **Higher privacy requirements lead to higher attained compression rates with a slack of Δ.** A higher (fixed) compression rate can also attain tighter privacy at no cost in performance. See Appendix D for discussion on (ε, δ)-DP conversions. See Figures 5 and 6 in Appendix B for results on the Shakespeare dataset.

Analyzing the privacy-utility-communication tradeoff We now study the best compression rates that we can attain without significantly impacting the current performance. For this, we train models that achieve state-of-the-art performance on the FL tasks we consider, and allow a slack of $\Delta\%$ relative to these models when trained without compression ($r = 1$x). We choose $\Delta = 4\%$ for SONWP and $\Delta = 1\%$ for F-EMNIST which may vary by task. We first consider $z = 0$, i.e., no DP, and see that about $r = 4$x compression can be attained. However, recall that our theoretical results suggest that as z increases and thus ε decreases, fewer bits of communication are needed. Our experiments echo this finding: at $z = 0.3$, we observe $r = 5$x is now attainable and at $z \geq 0.5$, $r = 10$x is attainable. The highest z we display can correspond both to a tighter privacy regime ($\varepsilon \approx 10$ or less) but, importantly also to models that are still highly performant indicating that a practitioner could reasonably select both these z and r to train models comparable to the state-of-the-art. Our experiments on F-EMNIST in Figure 2b show similar findings where fixing $r \geq 50$x can attain $z \geq 0.5$ for 'free'. Finally, these results also corroborate that our bound of $\tilde{O}\left(\min(n^2\varepsilon^2, d)\right)$ is significantly less (≈ 10x) than that of Kairouz et al. (2021a) in practice.

Finding that we can significantly compress our models, another question that can be asked is 'could a smaller model have been used instead?' To investigate this, we train a smaller model (denoted 'small') which has only $\approx 2 \cdot 10^5$ parameters, which is comparable in size to the original CNN model updates compressed by $r \approx 5$x. We observe that this model has significantly lower performance ($> 5pp$) across all privacy budgets when we compare them for a fixed latent dimension of $length \times width$. When we compress our original model beyond $r = 5$x (smaller than the 'small' model), we find that it still significantly outperforms it. These results indicate that training larger models do in fact attain higher performance even for the same latent update size: further, our results indicate we can enable training these larger models under the same fixed total communication.

Quantization or dimensionality reduction? Though we achieve significant compression rates from our linear dimensionality reduction technique, we now explore how these results compare with compression via quantization. Theoretically our bound can achieve a minimum compression independent of the ambient gradient size d. Because of this, we also expect our methods to outperform those based on quantization because their communication scales with d.

Our results in Figure 3 and Table 1 of Appendix B corroborate this hypothesis. We compare our compression against conditional randomized rounding to an integer grid of field size 2^b. We vary the quantization (bits) per parameter b while allowing the same Δ as above. Combined with dimensionality reduction, this gives a total communication per ambient parameter as b/r. When we use the vanilla distributed DP scheme of Kairouz et al. (2021a), we find that we can only compress to 10 bits per parameter on SONWP; if we instead favour our linear dimensionality reduction technique, we achieve a much lower 1.2 bits per parameter (with $r = 10$x and $b = 12$). For F-EMNIST, we find we can compress down to 0.24 bits per parameter at $z = 0.5$ by optimizing both b and r; this is much lower than using only quantization (10 bits per parameter) and a marginal increase over only dimensionality reduction (0.27). We observe 0.7 bits per parameter at $z = 0.3$ and 1.2 at $z = 0.1$. See Figure 8 for results with $z = 0.1, 0.3$ and Figure 9 for the comprehensive results at many r and b, both in Appendix B.

Because of this, we now explore if raising the b (decreasing quantization), past the max $b = 18$ bits per parameter that we have considered thus far, will decrease the total communication. Inspecting $b = 18 \to 22$ in Figure 3 we do observe a marginal increase in test performance with increasing b in some cases, indicating there may be potential to increase r. However, we do not find that we can significantly increase r in these cases (see Figure 9 of Appendix B).

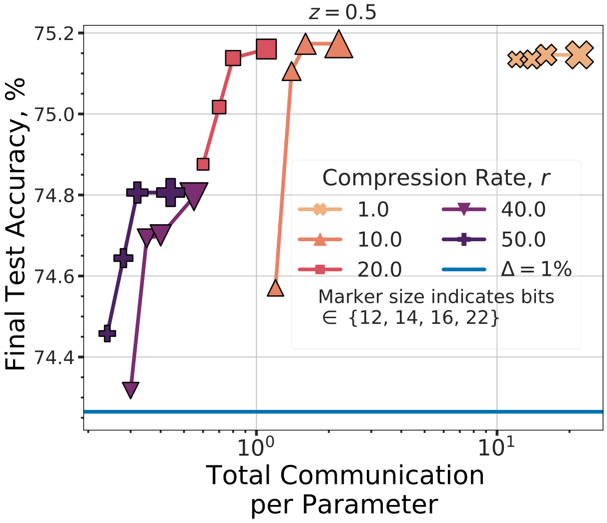


Figure 3. **Optimizing both r and b can further decrease communication**, to 0.24 bits per parameter at $z = 0.5$. See Figure 8 of Appendix B for $z = 0.1, 0.3$. Note that sometimes at higher bitwidths we observe lower performance within statistical error (standard deviation ≈ 1)—here, we threshold to the highest accuracy of lower bitwidths to ease visualization. See Appendix D for discussion on (ε, δ)-DP conversions. Full results without thresholding and all r are in Figure 9 of Appendix B.

Impact of cohort size We conduct experiments under (approximately, because varying n impacts ε) fixed ε to investigate how the cohort size n impacts compression. From Theorem 5.3, we expect to see that as n increases, so does does communication. Our empirical results closely match, shown in Figure 10 of Appendix B. Further, we find that under sufficiently high z, we can *improve the model performance* despite SecAgg communication limits—this bounds the number of clients and model size. In Table 2 of Appendix B, we increase n while keeping the total message size fixed (increase r by $d/(\frac{n_2}{n_1}\frac{\log n_2}{\log n_1})$, where $n_2 > n_1$). At $z = 0.5$, we observe that though this message size remains fixed the final model performance increases by nearly 5 percentage points on SONWP. In other words, in large n scenarios where SecAgg can fail (due to large communication), our protocol may enable a practitioner to still increase n to obtain better performance. We discuss this further in Appendix B.1.

We attempted in Appendix B.2 to improve compression rates by compressing layers separately (per-layer) or thresholding the noisy aggregate random projections. We found neither achieved significant improvements. The former aligns with results from McMahan et al. (2017b) in central DP.

Discussion and Main Takeaways Our empirical analysis largely corroborates our asymptotic theoretical findings that with sufficiently high z (tighter privacy), less communication is needed (at little to no cost in model performance); in other words, the compression comes nearly “free”. We find that in tighter privacy regimes, significant compression of > 10x (as low as 1.2 bits per ambient parameter) can be attained on SONWP with less than 4% relative error; reducing slack to 2% still allows for > 5x compression. In addition, we find that dimensionality reduction via sparse random projections outperforms quantization, where the latter can only attain compression to 10 bits on SONWP. Finally, we observe that our protocol may be able to enable higher cohort sizes n—despite fundamental limits on SecAgg message sizes—due to reducing the message size. We find that this can improve the overall model performance.

8. Conclusion

In this paper, we study the optimal privacy-communication-accuracy trade-offs under distributed DP via SecAgg. We show that existing schemes are order optimal when $d \ll n^2\varepsilon^2$ and strictly sub-optimal otherwise. To address this issue, we provide an optimal scheme that leverages sparse random projections. We also show how our scheme can be minimally modified when the client updates are sparse to further improve the trade-offs. Our extensive experiments on FL benchmark datasets demonstrate significant communication gains ($\sim$10x) relative to existing schemes. Many important questions remain open, including obtaining a fundamental characterization of the privacy-accuracy-communication trade-offs under other models of distributed DP (e.g. via a trusted third-party or in a secure enclave as in (Bittau et al., 2017; Ghazi et al., 2020b; 2021; 2020c;a; Ishai et al., 2006; Balle et al., 2019; 2020; Balcer & Cheu, 2020; Balcer et al., 2021; Girgis et al., 2021b;a; Erlingsson et al., 2019)).

References

Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., and McMahan, B. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pp. 7564–7575, 2018.

Agarwal, N., Kairouz, P., and Liu, Z. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1709–1720. Curran Associates, Inc., 2017.

Andrew, G., Thakkar, O., McMahan, H. B., and Ramaswamy, S. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

Anonymous. Iterative sketching and its application to federated learning. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=U_Jog0t3fAu. under review.

Asoodeh, S., Liao, J., Calmon, F. P., Kosut, O., and Sankar, L. A better bound gives a hundred rounds: Enhanced privacy guarantees via f-divergences. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 920–925. IEEE, 2020.

Balcer, V. and Cheu, A. Separating local & shuffled differential privacy via histograms. In *ITC*, pp. 1:1–1:14, 2020.

Balcer, V., Cheu, A., Joseph, M., and Mao, J. Connecting robust shuffle privacy and pan-privacy. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2384–2403. SIAM, 2021.

Balle, B., Bell, J., Gascón, A., and Nissim, K. The privacy blanket of the shuffle model. In *CRYPTO*, pp. 638–667, 2019.

Balle, B., Bell, J., Gascón, A., and Nissim, K. Private summation in the multi-message shuffle model. pp. 657–676, 2020. doi: 10.1145/3372297.3417242. URL https://doi.org/10.1145/3372297.3417242.

Barnes, L. P., Han, Y., and Ozgur, A. Lower bounds for learning distributions under communication constraints via fisher information, 2019.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Bell, J. H., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1253–1269, 2020.

Bernstein, J., Wang, Y.-X., Azizzadenesheli, K., and Anandkumar, A. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569. PMLR, 2018.

Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., and Seefeld, B. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 441–459, 2017.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016a.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Bonawitz, K. A., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016b. URL https://arxiv.org/abs/1611.04482.

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Canonne, C. L., Kamath, G., and Steinke, T. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010*, 2020.

Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.

Chen, W.-N., Kairouz, P., and Özgür, A. Breaking the communication-privacy-accuracy trilemma. *arXiv preprint arXiv:2007.11707*, 2020.

Duchi, J. C., Jordan, M. I., and Wainwright, M. J. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438. IEEE, 2013.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503. Springer, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006b.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., and Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2468–2479. SIAM, 2019.

Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. Privacy preserving mining of association rules. *Information Systems*, 29(4):343–364, 2004.

Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Möllering, H., Nguyen, T. D., Rieger, P., Sadeghi, A.-R., Schneider, T., Yalame, H., et al. Safelearn: Secure aggregation for private federated learning. In *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 56–62. IEEE, 2021.

Geyer, R. C., Klein, T., and Nabi, M. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Pagh, R., and Velingker, A. Pure differentially private summation from anonymous messages. In *1st Conference on Information-Theoretic Cryptography (ITC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020a.

Ghazi, B., Kumar, R., Manurangsi, P., and Pagh, R. Private counting from anonymous messages: Near-optimal accuracy with vanishing communication overhead. In *ICML*, pp. 3505–3514, 2020b.

Ghazi, B., Manurangsi, P., Pagh, R., and Velingker, A. Private aggregation from fewer anonymous messages. In *Eurocrypt*, 2020c.

Ghazi, B., Golowich, N., Kumar, R., Pagh, R., and Velingker, A. On the power of multiple anonymous messages. In *Eurocrypt*, 2021. To appear.

Girgis, A., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2521–2529. PMLR, 2021a.

Girgis, A. M., Data, D., Diggavi, S., Kairouz, P., and Suresh, A. T. Shuffled model of federated learning: Privacy, accuracy and communication trade-offs. *IEEE Journal on Selected Areas in Information Theory*, 2(1):464–478, 2021b. doi: 10.1109/JSAIT.2021.3056102.

Haddadpour, F., Karimi, B., Li, P., and Li, X. Fedsketch: Communication-efficient and private federated learning via sketching. *arXiv preprint arXiv:2008.04975*, 2020.

Havasi, M., Peharz, R., and Hernández-Lobato, J. M. Minimal random code learning: Getting bits back from compressed model parameters. *arXiv preprint arXiv:1810.00440*, 2018.

Ishai, Y., Kushilevitz, E., Ostrovsky, R., and Sahai, A. Cryptography from anonymity. In *FOCS*, pp. 239–248, 2006.

Kadhe, S., Rajaraman, N., Koyluoglu, O. O., and Ramchandran, K. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020.

Kairouz, P., Bonawitz, K., and Ramage, D. Discrete distribution estimation under local privacy. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 2436–2444, New York, New York, USA, 20–22 Jun 2016.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kairouz, P., Liu, Z., and Steinke, T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. *arXiv preprint arXiv:2102.06387*, 2021a.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021b. ISSN 1935-8237. doi: 10.1561/2200000083. URL http://dx.doi.org/10.1561/2200000083.

Kamath, G. and Ullman, J. A primer on private statistics. *arXiv preprint arXiv:2005.00010*, 2020.

Kane, D. M. and Nelson, J. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Lin, Y., Han, S., Mao, H., Wang, Y., and Dally, W. J. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017a.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706. IEEE, 2019.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275. IEEE, 2017.

Oktay, D., Ballé, J., Singh, S., and Shrivastava, A. Scalable model compression by entropy penalized reparameterization. *arXiv preprint arXiv:1906.06624*, 2019.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., and Arora, R. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pp. 8253–8265. PMLR, 2020.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Song, C. and Shmatikov, V. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–206, 2019.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248. IEEE, 2013.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3329–3337. JMLR.org, 2017.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wangni, J., Wang, J., Liu, J., and Zhang, T. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*, 2017.

Warner, S. L. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

A. Societal Considerations

Our work explores private learning from distributed data. In general, this enables tasks like (federated) machine learning without compromising the privacy of users who contribute data to these protocols. However, privacy guarantees are complex and the relationship between optimized privacy metrics like ε-DP with the practical privacy leakage are not well understood. Because of this, works that leverage private learning protocols but do not guarantee a tight (ε, δ)-DP bound may provide a false sense of privacy for user data.

B. Additional Figures

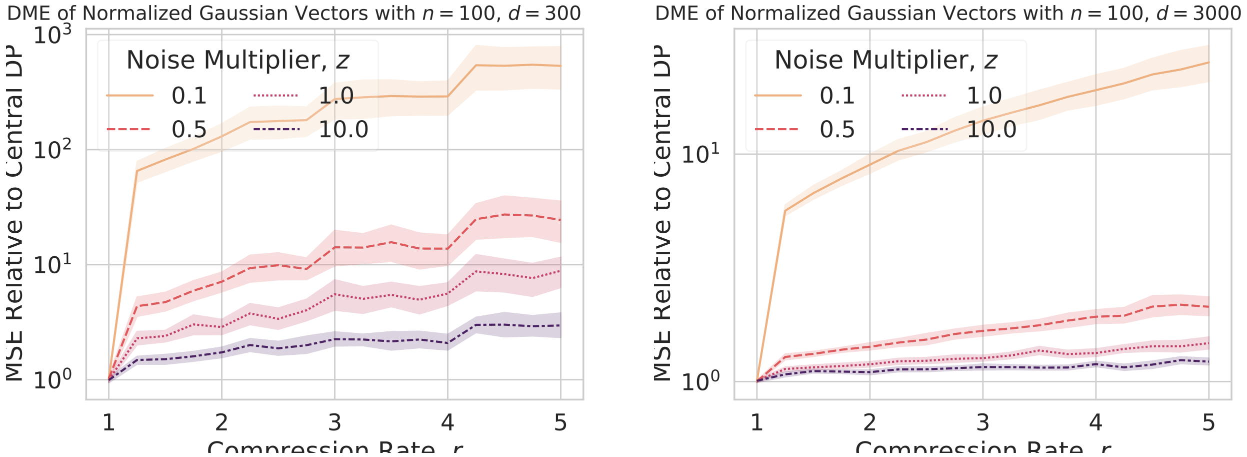
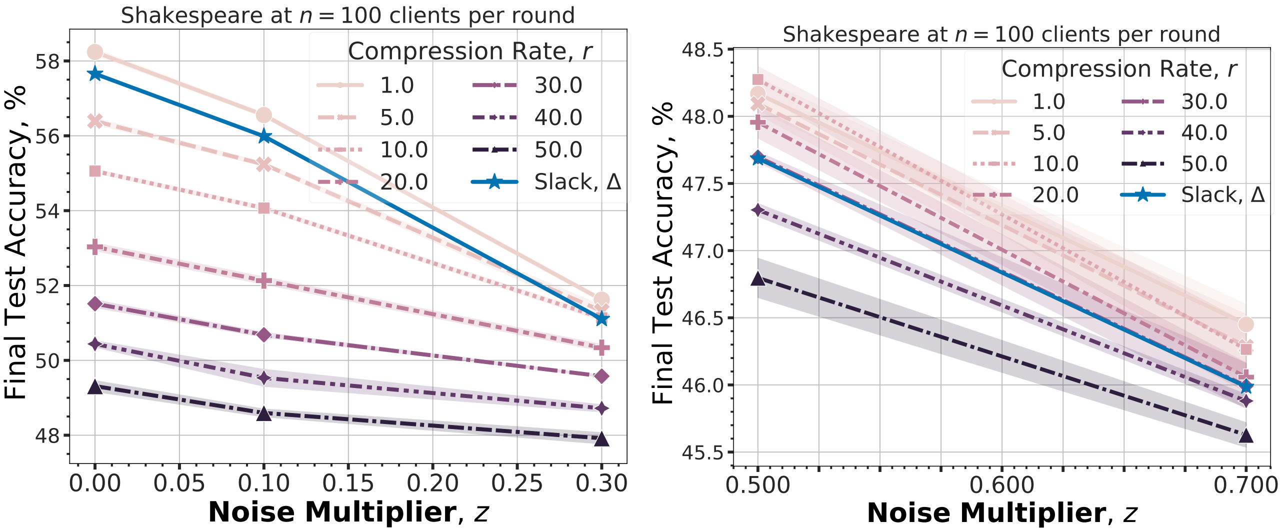


Figure 4. **Empirically observed MSE well aligns with asymptotic theoretical analysis. As the vector dimesionality grows d, higher compression can be attained. Tighter privacy (high z) also allows more compression.** We vary the vector dimensionality between $d = \{300, 3000\}$ for the subfigures and use the same setup as (Kairouz et al., 2021a).

Noise Multiplier, z	Chosen Compression Rate r	Lowest Bit Width b per Parameter	Total Communication Per Parameter	Final Test Performance, %
0.3	1	10	10	21.90 ± 0.09
	5	**12**	**2.4**	$\mathbf{21.68 \pm 0.02}$
0.5	1	10	10	21.21 ± 0.18
	5	12	2.4	21.11 ± 0.08
	10	**12**	**1.2**	$\mathbf{20.74 \pm 0.05}$

Table 1. **Optimal compression can be found by tuning both the bit width b and compression rate r.** Results for SONWP with 1000 clients. We find that increasing r instead of b achieves the highest total compression in all cases. Bold rows show the optimal compression parameters for the given z. We note that we cannot set lower than $b = 10$ for this setting because SecAgg requires at least $O(\log{(n)})$ bits. The results in the final column take the form mean±standard deviation.



(a) Zoom-in of noise multipliers $z \in [0.0, 0.30]$. Here, even 10x compression can be attained with a smaller $z = 0.30$.

(b) Zoom-in of noise multipliers $z \in [0.50, 0.70]$. Here, up to 30X compression can be attained at higher noise multipliers $z \geq 0.50$.

Figure 5. **At higher privacy requirements, higher compression rates can be attained under similar model performance.** We use a relative slack $\Delta = 1\%$ with models trained on the Shakespeare dataset.

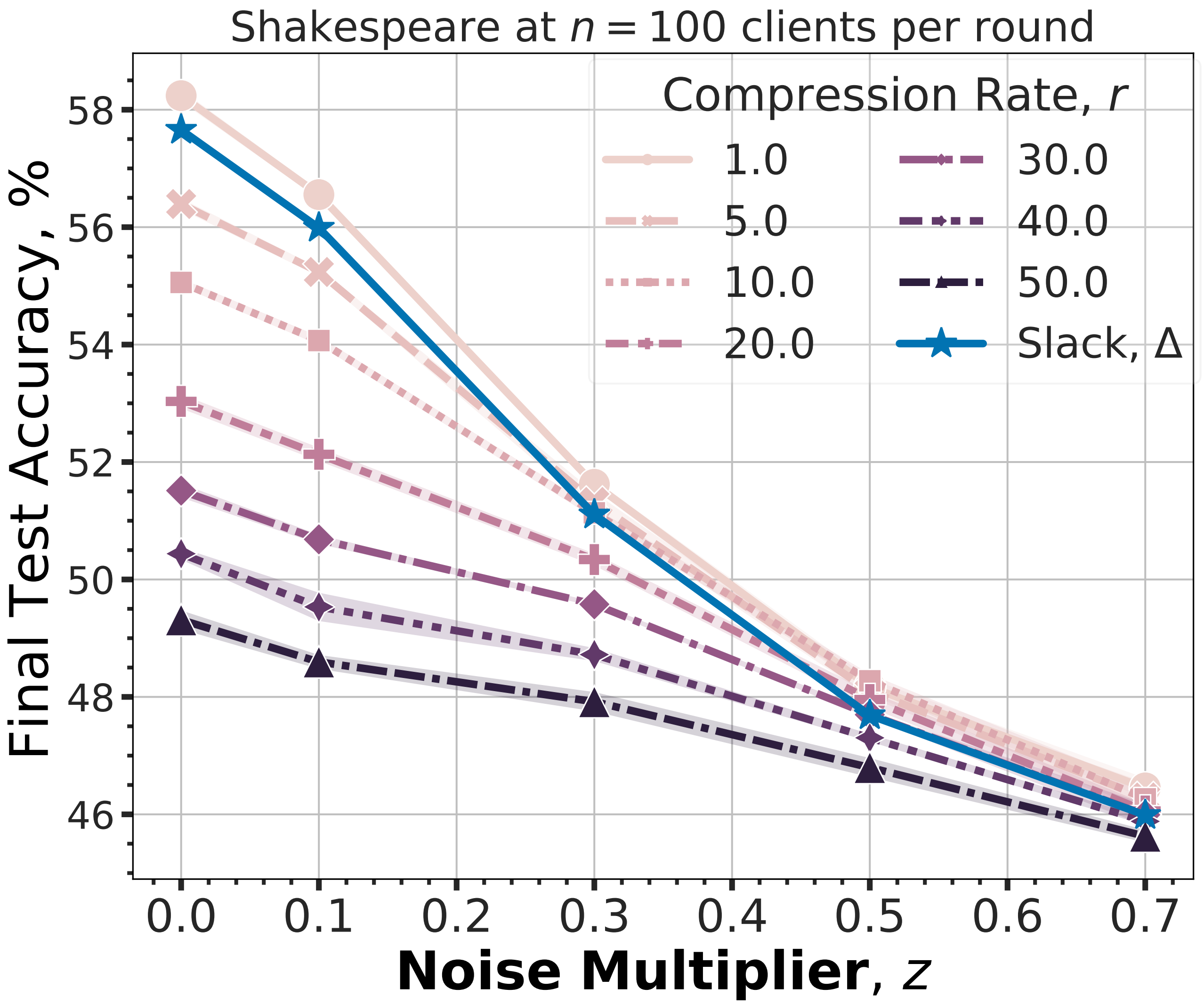
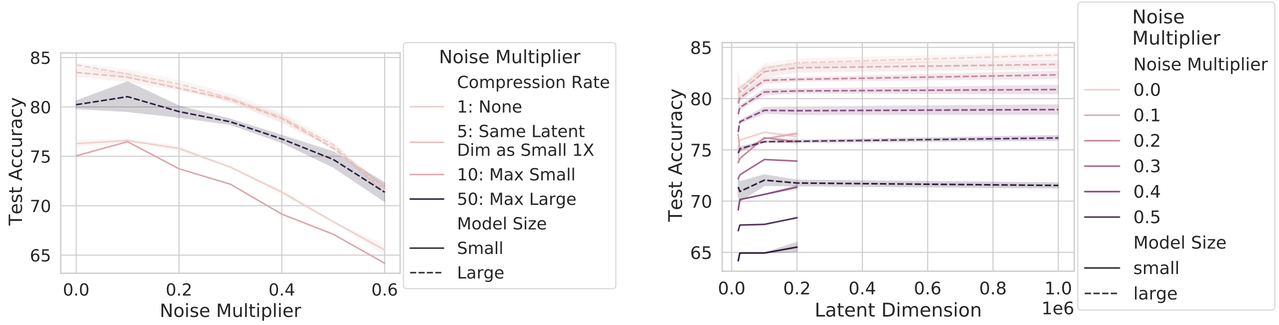


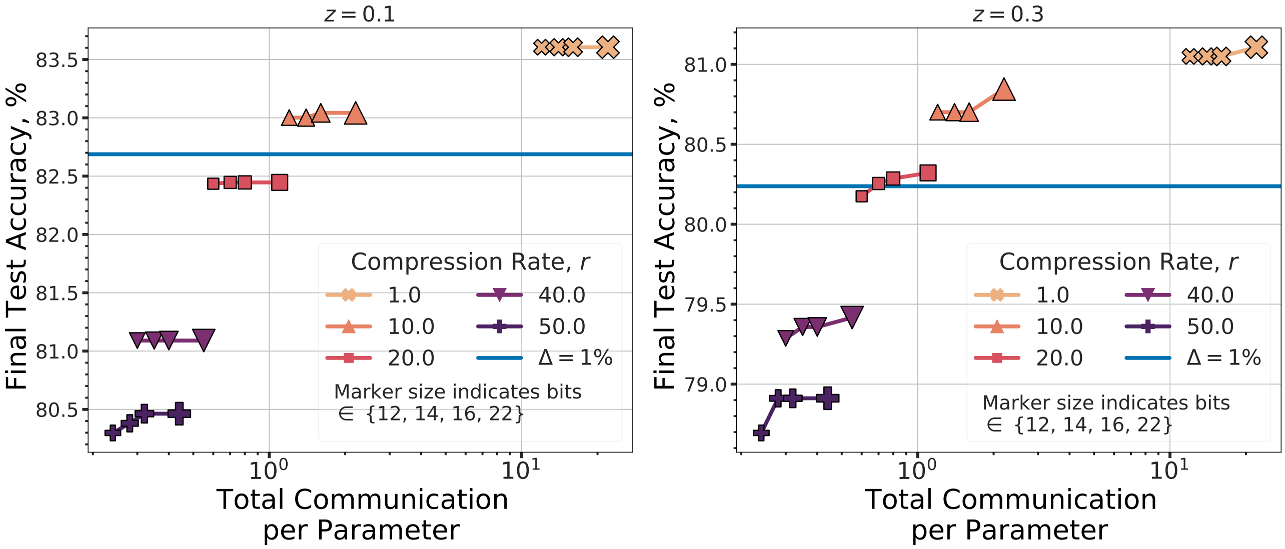
Figure 6. **At higher privacy requirements, higher compression rates can be attained under similar model performance.** We find up to 40x compression at $z = 0.70$ can be attained. We use a relative slack $\Delta = 1\%$ with models trained on the Shakespeare dataset.



(a) **A large model, compressed to the same latent dimension as a small model, outperforms it.**

(b) **Impact of the number of parameters on the privacy-utility-communication tradeoff.**

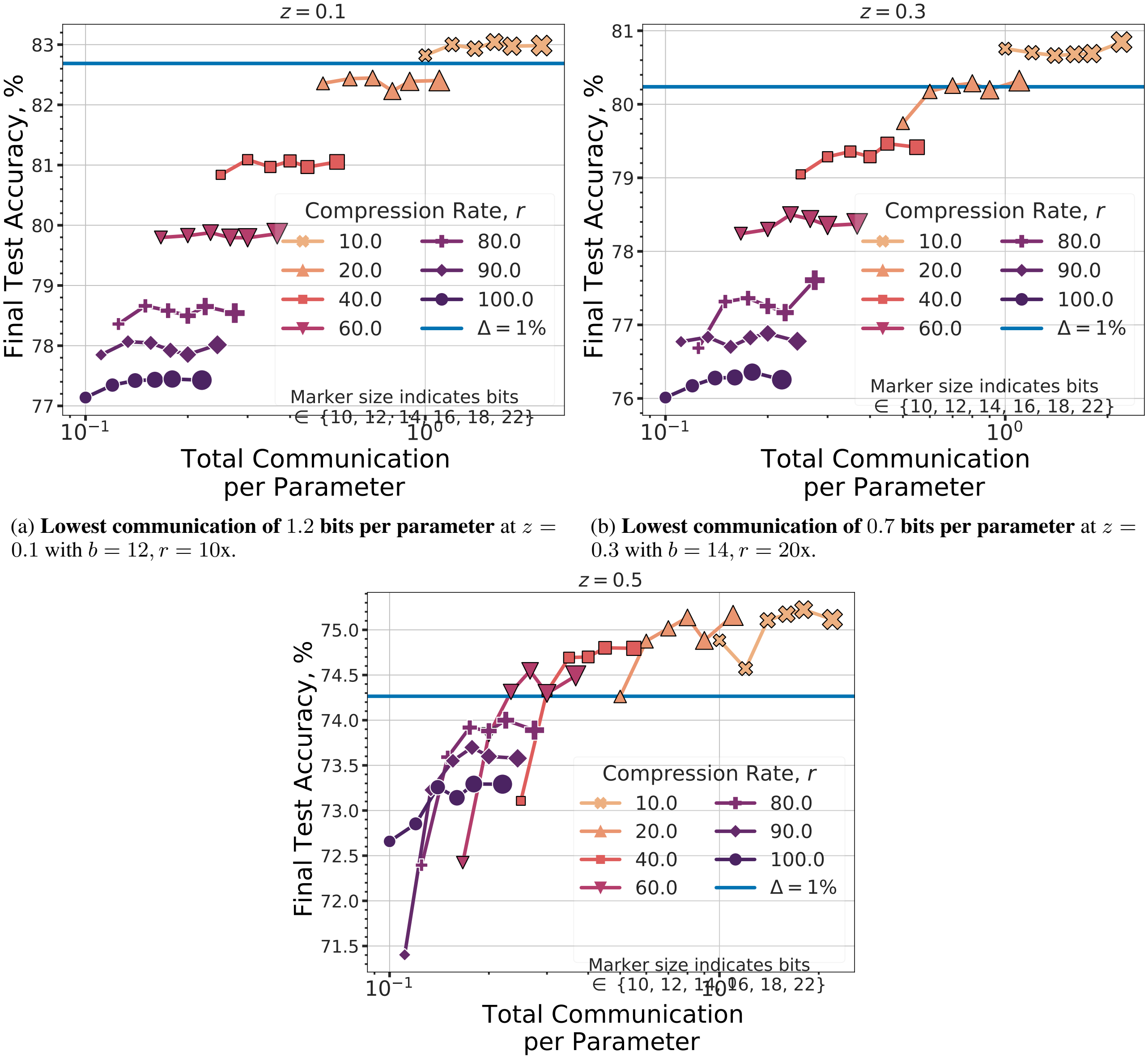
Figure 7. **Large models with compression outperform small models.**



(a) **Lowest communication of** 1.2 **bits per parameter** at $z = 0.1$ with $b = 12, r = 10$x.

(b) **Lowest communication of** 0.7 **bits per parameter** at $z = 0.3$ with $b = 14, r = 20$x.

Figure 8. **Optimizing both** r **and** b **can further decrease communication**. Note that sometimes at higher bitwidths we observe lower performance within statistical error (standard deviation ≈ 1)—here, we threshold to the highest accuracy of lower bitwidths. Full results without thresholding and all r are in Figure 9 of Appendix B.



(a) **Lowest communication of** 1.2 **bits per parameter** at $z = 0.1$ with $b = 12, r = 10$x.

(b) **Lowest communication of** 0.7 **bits per parameter** at $z = 0.3$ with $b = 14, r = 20$x.

(c) **Lowest communication of** 0.7 **bits per parameter** at $z = 0.3$ with $b = 14, r = 20$x.

Figure 9. **Optimizing both** r **and** b **can further decrease communication**. Full results for Figure 3 and 8.

B.1. Impact of cohort size

Finally, we explore how varying the number of clients per round (or, cohort size) n impacts the privacy-utility-communication tradeoff. This value plays several key roles in this tradeoff. First, increasing n increases the sampling probability of the cohort, which increases the total privacy expenditure. However, it also tends to improve model performance—this may mean that a higher noise multiplier z can be chosen so as to instead decrease the total privacy cost (this is typically the case when N is large enough, e.g., SO). In terms of communication, Theorem 5.1 suggests that increasing n will also increase the per-client comunication. Because of the aforementioned complex tradeoffs, we (approximately) fix the privacy budget ε and only perturb n minimally around a nominal value of 100. In Figure 10, we see that dependence of communication on n is observed empirically as well.

In addition to this impact on r, setting n can also have a significant impact on the run time of SecAgg, of $O(n\log(n)d)$. For large values of n, this can entirely prevent the protocol from completing. Because a practitioner desires the most performant model, a common goal is the increase n so as to obtain a tight ε (due to a now higher z) with the least cost in performance. But, because large n can crash SecAgg, this places a constraint on the maximum n that can be chosen. Since our methods compress the updates (d above), it is possible to still increase n so long as we increase r accordingly (by $\frac{n_2}{n_1}\frac{\log n_2}{\log n_1}$ where $n_2 > n_1$), which maintains fixed runtime. If the resulting model at higher n achieves higher performance, then we observe a net benefit from this tradeoff. For a practical privacy parameter or $z = 0.6$, our results in Table 2 suggest that this may be possible. Specifically, increasing n from $100 \to 1000$ and settings $r = 50x > 15x$ accordingly, we observe that the final model with $n = 1000$ clients achieves a nearly 5pp gain. We observe that $z < 0.3$ cannot meet these requirements.

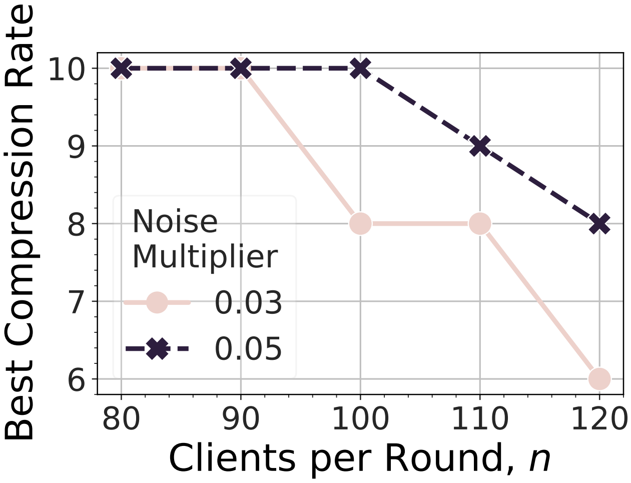


Figure 10. **Increasing n leads to an increase in communication** shown by the decreasing compression rates. Higher noise multipliers can still attain higher compression. Results for SONWP.

Noise Multiplier, z	Number of Clients, n	Compression Rate, r	Final Test Performance, %
0.1	100	1	83.05 ± 0.44
	1000	10	82.95 ± 0.40
0.3	100	1	80.61 ± 0.46
	1000	40	80.78 ± 0.29
0.5	100	1	75.34 ± 0.49
	1000	50	80.13 ± 0.22

Table 2. **With z sufficiently large, increasing $n = 100 \to 1000$ can attain higher model performance even for increased r.** In particular, to maintain the same SecAgg runtime, we require $r \geq 15$ for this setting to increase $n = 100 \to 1000$. We observe that $z \geq 0.3$ meets this requirement while achieving final models that outperform the $n = 100$, $r = 1$x client baseline. Results for SONWP.

B.2. Attempting to improve compression via per-layer sketching and thresholding

We attempted two additional methods to improve our compression rates. First, we noticed that the LSTM models we trained had consistently different ℓ_2 norms across layers in training. Because these norms are different, we hypothesizes that sketching and perturbing them separately may improve the model utility. We attempt this protocol in Figures 11 and 12, where Figure 11 uses $z = 0.05$ and Figure12 uses $z = 0$. We find that, in general, there are no significant performance gains. We further attempt to threshold low values in the sketch. Because this leads to a biased estimate of the gradient, we keep track of the zero-d values in an error term. We tried several threshold values and we found that this as well led to no significant performance gains.

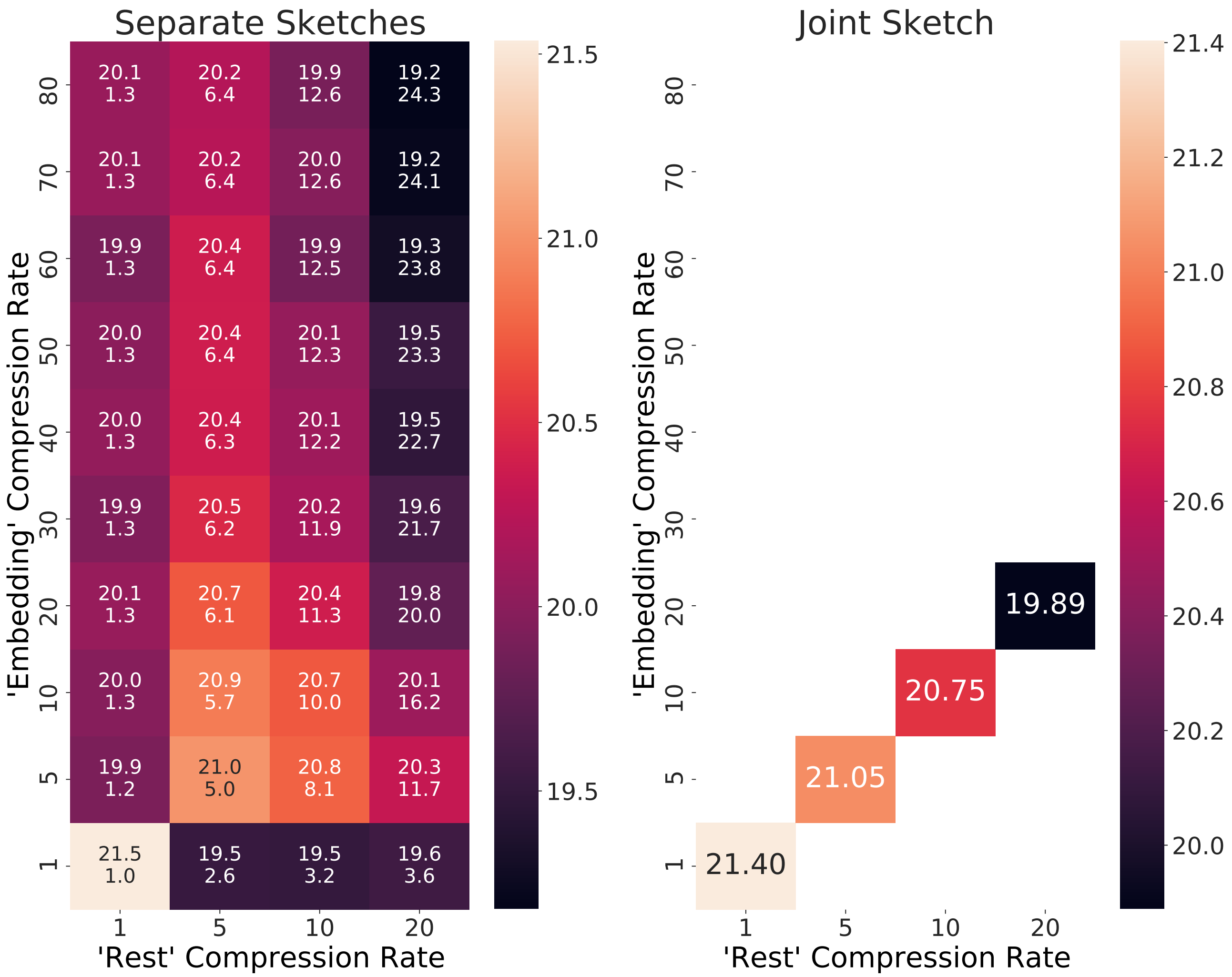


Figure 11. **Separate sketching does not significantly improve the final model performance.** Heatmap values correspond to the final model test performance followed by (newline) the total compression and are colored by the test performance. Results using the LSTM model on SONWP with $z = 0.05$. We train models either by sketching the entire concatenated gradient vector or by sketching the 'embedding' layer separate from the 'rest' of the model.

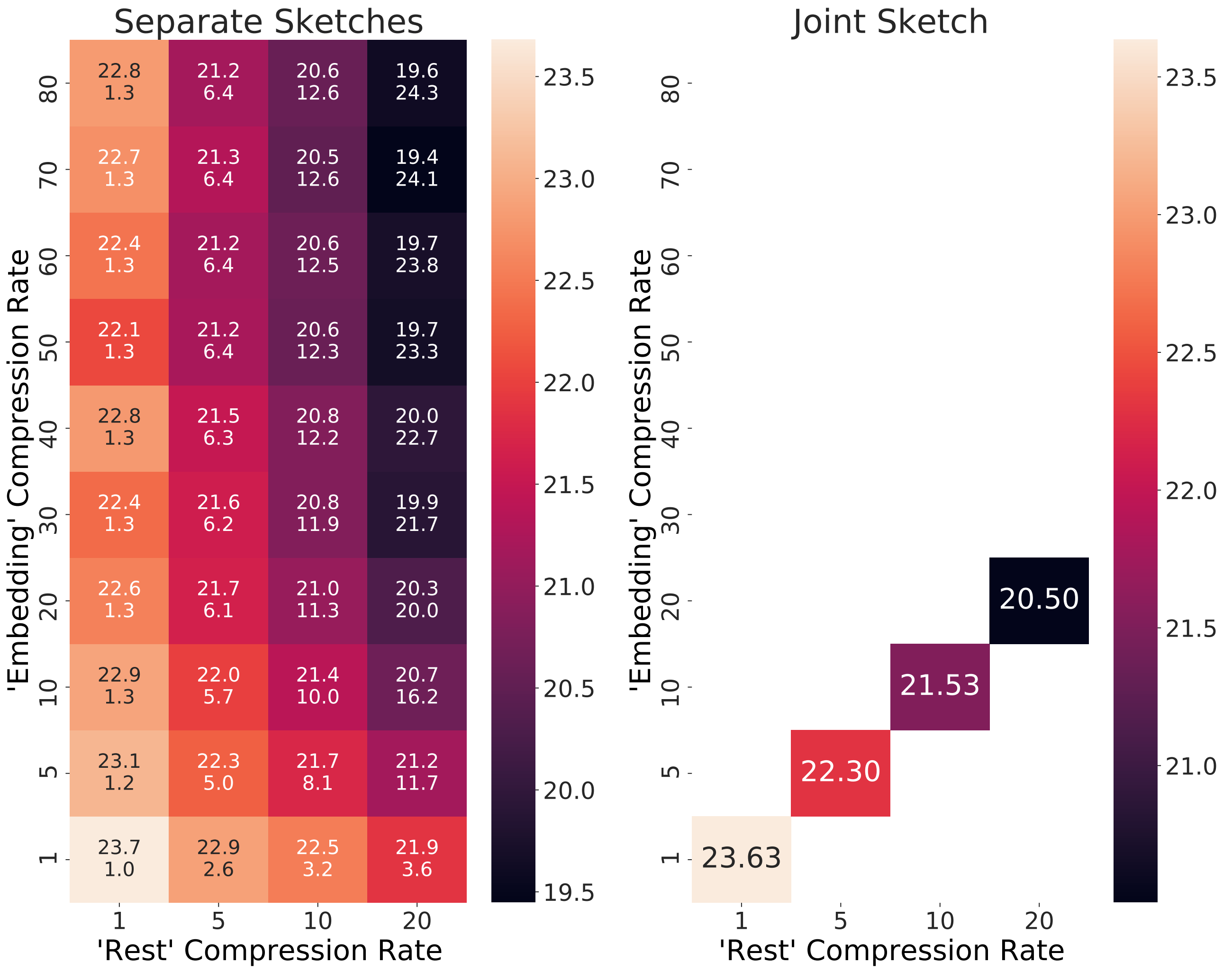


Figure 12. **Separate sketching does not significantly improve the final model performance.** Heatmap values correspond to the final model test performance followed by (newline) the total compression and are colored by the test performance. Results using the LSTM model on SONWP with $z = 0.00$. We train models either by sketching the entire concatenated gradient vector or by sketching the 'embedding' layer separate from the 'rest' of the model.

C. Datasets and Training Setup

We run experiments on the full Federated EMNIST and Stack Overflow datasets (Caldas et al., 2018), two common benchmarks for FL tasks. F-EMNIST has 62 classes and $N = 3400$ clients, with each user holding both a train and test set of examples. In total, there are $671,585$ training examples and $77,483$ test examples. Inputs are single-channel $(28, 28)$ images. We sample $n \in [100, 1000]$ clients per round for a total $R = 1500$ rounds. The Stack Overflow (SO) dataset is a large-scale text dataset based on responses to questions asked on the site Stack Overflow. The are over 10^8 data samples unevenly distributed across $N = 342477$ clients. We focus on the next word prediction (NWP) task: given a sequence of words, predict the next words in the sequence. We sample use $n \in [100, 1000]$ and $R = 1500$. On F-EMNIST, we experiment with a ≈ 1 million parameter (4 layer) Convolutional Neural Network (CNN) used by (Kairouz et al., 2021a). On SONWP, we experiment with a ≈ 4 million parameter (4 layer) long-short term memory (LSTM) model, which is the same as prior work (Andrew et al., 2019; Kairouz et al., 2021a).

On F-EMNIST, we use a server learning rate of $1.$ normalized by n (the number of clients) and momentum of 0.9 (Polyak, 1964); the client uses a learning rate of 0.01 without momentum. On Stack Overflow, we use a server learning rate of 1.78 normalized by n and momentum of 0.9; the client uses a learning rate of 0.3.

For distributed DP, we use the geometric adaptive clipping of (Andrew et al., 2019) with an initial ℓ_2 clipping norm of 0.1 and a target quantile of 0.5. We use the same procedure as (Kairouz et al., 2021a) and flatten using the Discrete Fourier Transform, pick $\beta = \exp(-0.5)$ as the conditional randomized rounding bias, and use a modular clipping target probability of $6.33\text{e}{-5}$ or ≈ 4 standard deviations at the server (assuming normally distributed updates). We communicate 16 bits per parameter for F-EMNIST and 18 bits for SONWP unless otherwise indicated.

On F-EMNIST, our 'large' model corresponds to the CNN whereas our 'small' model corresponds to an $\approx 200,000$ parameter model with 3 dense layers (see Figure 14).

C.1. Model Architectures

```
Model: "Large Model: 1M parameter CNN"

 Layer type                 Output Shape              Param #
=================================================================
 Conv2D                     None, 26, 26, 32          320

 MaxPooling2D               None, 13, 13, 32          0

 Conv2D                     None, 11, 11, 64          18496

 Dropout                    None, 11, 11, 64          0

 Flatten                    None, 7744                0

 Dense                      None, 128                 991360

 Dropout                    None, 128                 0

 Dense                      None, 62                  7998

=================================================================
Total params: 1,018,174
Trainable params: 1,018,174
Non-trainable params: 0
```

Figure 13. 'Large' model architecture.

```
"Small Model: 200k parameter Dense DNN"
_________________________________________________________________
 Layer                    Output Shape                 Param #
=================================================================
 Reshape                  None, 784                  0

 Dense                    None, 200                  157000

 Dense                    None, 200                  40200

 Dense                    None, 62                   12462

=================================================================
Total params: 209,662
Trainable params: 209,662
Non-trainable params: 0
_________________________________________________________________
```

Figure 14. ‘Small’ model architecture.

```
Model: "Stack Overflow Next Word Prediction Model"
_________________________________________________________________
 Layer type                    Output Shape              Param #
=================================================================
 InputLayer                    None, None                0

 Embedding                     None, None, 96            960384

 LSTM                          None, None, 670           2055560

 Dense                         None, None, 96            64416

 Dense                         None, None, 10004         970388

=================================================================
Total params: 4,050,748
Trainable params: 4,050,748
Non-trainable params: 0
_________________________________________________________________
```

Figure 15. Stack Overflow Next Word Prediction model architecture.

```
Model: "Shakespeare Character Prediction Model"
_________________________________________________________________
 Layer type                    Output Shape              Param #
=================================================================
Embedding                      None, 80, 8               720

LSTM                           None, 80, 256             271360

LSTM                           None, 80, 256             525312

Dense                          None, 80, 90              23130

=================================================================
Total params: 820,522
Trainable params: 820,522
Non-trainable params: 0
_________________________________________________________________
```

Figure 16. Shakespeare character prediction model architecture.

D. DP and Sketching Empirical Details

Noise multiplier to ε-DP We specify the privacy budgets in terms of the noise multiplier z, which together with the clients per round n, total clients N, number of rounds R, and the clipping threshold completely specify the trained model ε-DP. Because the final ε-DP values depend on the sampling method: e.g., Poisson vs. fixed batch sampling, which depends on the production implementation of the FL system, we report the noise multipliers instead. Using Canonne et al. (2020); Mironov (2017), our highest noise multipliers roughly correspond to $\varepsilon = \{5, 10\}$ using $\delta = 1/N$ and privacy amplification via fixed batch sampling.

Sketching We display results in terms of the noise multiplier which fully specifies the ε-DP given our other parameters (n, N, and R). We discuss this choice in Appendix D. We use a count-mean sketch which compresses gradients to a sketch matrix of size (t, w)=($length$,$width$). We test $length \in \{10, 15, 20, 25\}$ and find that 15 leads to optimal final test performance. We use this value for all our experiments and calculate the $width = d/(r * length)$ where $gradient \in \mathcal{R}^d$ and r is the compression rate. We normalize each sketch row by the $length$ to lower the clipping norm, finding some improvements in our results. Thus, decoding requires only summing the gradient estimate from each row. We provide the full algorithms in below in Appendix E.

E. Sketching In Practice

Algorithm 3 *Gradient Count-Mean Sketch Encoding.* We find that normalizing (Line 6) in the encoding step improves performance by reducing the norm of the sketch.

```
Require: Gradient vector $g$, sketch width $sw$, sketch length $sl$, shared seed $seed$
1: $sketch \leftarrow zeros((sl, sw))$
2: for $hash\_index$ in $[0, \cdots, S.length]$, in parallel do
3:     $hash\_seed \leftarrow hash\_index + seed$
4:     $indices \leftarrow$random_uniform($0, S.width, hash\_seed$)
5:     $signs \leftarrow$ random_choice($[-1, 1], hash\_seed$)
6:     $weights \leftarrow signs \times \frac{grad}{sl}$
7:     $sketch[hash\_index] \leftarrow$bincount($indices, weights, sl$)
8: end for
9: Return: $sketch$
```

Algorithm 4 *Gradient Count-Mean Sketch Decoding*

```
Require: Sketch $S$, gradient vector size $d$, shared seed $seed$
$gradient\_estimate \leftarrow zeros(d)$
for $hash\_index$ in $[0, \cdots, S.length]$, in parallel do
    $hash\_seed \leftarrow hash\_index + seed$
    $indices \leftarrow$random_uniform($0, S.width, hash\_seed$)
    $signs \leftarrow$ random_choice($[-1, 1], hash\_seed$)
    $gradient\_estimate += signs * S[hash\_index, indices]$
end for
Return: $gradient\_estimate$
```

F. Additional details of the distributed discrete Gaussian mechanism (DDG)

Algorithm 5 Distributed discrete Gaussian mechanism $\texttt{DDG}_{\textsf{enc}}$ (with detailed parameters) (Kairouz et al., 2021a)

```
Inputs: Private vector $x_i \in \mathbb{R}^d$, Dimension $d$; clipping threshold $c$; granularity $\gamma > 0$; modulus $M \in \mathbb{N}$; noise scale $\sigma > 0$; bias $\beta \in [0, 1)$
Clip and scale vector: $x_i' = \frac{1}{\gamma}\min(1, \frac{c}{\|x_i\|_2}) \cdot x_i \in \mathbb{R}^d$
Flatten vector: $x_i'' = H_d D x_i'$ where $H_d$ is the $d$-dim Hadamard matrix and $D$ is a diagonal matrix with each diagonal entry $\mathsf{unif}\{+1, -1\}$
Conditional rounding:
while $\|\tilde{x}_i\|_2 > \min\left\{c/\gamma + \sqrt{d}, \sqrt{c^2/\gamma^2 + \frac{1}{4}d + \sqrt{2\log(1/\beta)}\left(c/\gamma + \frac{1}{2}\sqrt{d}\right)}\right\}$ do
    $\tilde{x}_i \in \mathbb{Z}^d$ be a randomized rounding of $x_i'' \in \mathbb{R}^d$ (i.e. $\mathbb{E}[\tilde{x}_i] = x_i''$ and $\|\tilde{x}_i - x_i''\|_\infty \leq 1$)
end while
Perturbation: $z_i = \tilde{x}_i + \mathcal{N}_{\mathbb{Z}}(0, \sigma^2/\gamma^2) \mod M$, where $\mathcal{N}_{\mathbb{Z}}$ is the discrete Gaussian noise
Return: $z_i \in \mathbb{Z}_M^d$
```

Theorem F.1 (private mean estimation with SecAgg (Kairouz et al., 2021a))**.** *Define*

$$
\begin{aligned}
\Delta_2^2 &\triangleq \min\left\{c^2 + \frac{\gamma^2 d}{4} + \sqrt{2\log(1/\beta)}\gamma\left(c + \frac{\gamma}{2}\sqrt{d}\right), \left(c + \gamma\sqrt{d}\right)^2\right\},\\
\tau &\triangleq 10\sum_{k=1}^{n-1}\exp\left(-2\pi^2\frac{\sigma^2}{\gamma^2}\frac{k}{k+1}\right),\\
\varepsilon &\triangleq \min\left\{\sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d}, \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d}\right\},\\
M &\geq O\left(n + \sqrt{\frac{\varepsilon^2 n^3}{d}} + \frac{\sqrt{d}}{\varepsilon}\log\left(n + \sqrt{\frac{\varepsilon^2 n^3}{d}} + \frac{\sqrt{d}}{\varepsilon}\right)\right),\\
\beta &\leq \Theta\left(\frac{1}{n}\right),\\
\sigma &= \tilde{\Theta}\left(\frac{c}{\varepsilon\sqrt{n}} + \frac{\gamma\sqrt{d}}{\varepsilon\sqrt{n}}\right),\\
\gamma &= \tilde{\Theta}\left(\min\left(\frac{cn}{M\sqrt{d}}, \frac{c}{\varepsilon M}\right)\right).
\end{aligned}
$$

Then Algorithm 5 satisfies

- $\frac{1}{2}\varepsilon^2$*-concentrated differential privacy (which implies* $(\alpha, \frac{\varepsilon^2}{\alpha})$*-RDP)*
- $O(d\log(M)) = O\left(d\log\left(n + \sqrt{\frac{n^3\varepsilon^2}{d}} + \frac{\sqrt{d}}{\varepsilon}\right)\right)$ *bits per-client communication cost;*
- *MSE* $\mathbb{E}\left[\|\hat{\mu} - \mu\|_2^2\right] = O\left(\frac{c^2 d}{n^2\varepsilon^2}\right)$.

G. Additional details of Section 3

G.1. From Renyi DP to approximate DP

The following conversion lemma from (Asoodeh et al., 2020; Canonne et al., 2020; Bun & Steinke, 2016) relates RDP to $(\varepsilon_{\mathsf{DP}}(\delta), \delta)$-DP.

Lemma G.1. *If M satisfies (α, ε)-RDP, then, for any $\delta > 0$, M satisfies $(\varepsilon_{\mathsf{DP}}(\delta), \delta)$, where*

$$\varepsilon_{\mathsf{DP}}(\delta) = \varepsilon + \frac{\log(1/\alpha\delta)}{\alpha - 1} + \log(1 - 1/\alpha).$$

G.2. Properties of sparse random projections

In this section, we introduce some properties of sparse random projection that will be used in our proofs. Let S be generated according to Section 3.3. The following two lemmas controls the distortion of the embedded vector $S \cdot g$ for a $g \in \mathbb{R}^d$.

Lemma G.2. *Let S be defined as in (2). For any $g_1, g_2 \in \mathbb{R}^d$, $\mathbb{E}\left[g_1^\intercal S^\intercal S g_2\right] = \langle g_1, g_2 \rangle$ and*

$$\mathbb{E}_S\left[\langle Sg_1, Sg_2\rangle^2\right] \leq \langle g_1, g_2\rangle^2 + \frac{2}{m}\left\|g_1\right\|_2^2 \cdot \left\|g_2\right\|_2^2.$$

Furthermore, for any $g \in \mathcal{R}^d$,

$$\mathbb{E}_S\left[\left\|S^\intercal S g - g\right\|_2^2\right] \leq \frac{2d}{m}\left\|g\right\|_2^2. \tag{5}$$

The proof follows by directly computing $\mathbb{E}_S\left[\langle Sg_1, Sg_2\rangle^2\right]$ (which can be written as a quadratic function of S and g_1, g_2). See, for instance, Lemma D.15 in (Anonymous, 2022).

Lemma G.3 (Sparse Johnson-Lindenstrauss lemma (Kane & Nelson, 2014))**.** *Let S be defined in (2) and let $g \in \mathbb{R}^d$. Then as long as $m \geq \Theta\left(\frac{1}{\alpha^2}\log\left(\frac{1}{\beta}\right)\right)$ and $t \geq \Theta\left(\frac{1}{\alpha}\log\left(\frac{1}{\beta}\right)\right)$,*

$$\Pr\left\{\left\|S \cdot g\right\|_2^2 \geq (1+\alpha)\left\|g\right\|_2^2\right\} \leq \beta. \tag{6}$$

Finally, Lemma G.4 states that the "unsketch" operator preserves the ℓ_2 norm.

Lemma G.4. *Let S, m, t be defined in (2) and $v \in \mathbb{R}^m$ (which can possibly depends on S) with $\mathbb{E}\left[\left\|v\right\|_2^2 | S\right] \leq B^2$ almost surely. Then it holds that $\mathbb{E}\left[\left\|S^\intercal v\right\|_2^2\right] \leq \frac{8dB^2}{m}$.*

The proof can be found in Section L.1.

H. Details of sparse DME scheme in Section 6

Algorithm 6 sparse DME via Compressed Sensing

```
Inputs: clients' data $x_1, ..., x_n$, sparse parameter $s$
The server generates an compression matrix $S \in \mathbb{R}^{m \times d}$ with $m = \Theta\left(s \log d\right)$ and $S_{i,j} \overset{\text{i.i.d.}}{\sim} N(0,1)$ and computes its largest singular value $\sigma_{\mathsf{max}}(S)$
The server broadcasts $S, \sigma_{\mathsf{max}}(S)$ to all clients
for $i \in [n]$ do
    Client $i$ computes $y_i \triangleq Sx_i$ and $z_i \triangleq \mathtt{DDG}_{\mathsf{enc}}\left(y_i\right) \in \mathbb{Z}_M^m$ with clipping rate $c' = c\sigma_{\mathsf{max}}(S)$ and dimension $d' = m$
end for
The server aggregates $Z_1, ..., Z_n$ with SecAgg and decodes $\hat{\mu}_y = \frac{1}{n}\mathtt{DDG}_{\mathsf{dec}}\left(\sum_{i\in[n]} Z_i\right)$
The server solves the following Lasso:
    $$\hat{\mu} \in \arg\min_{x \in \mathbb{R}^d}\left\{\frac{1}{m}\left\|\hat{\mu}_y - Sx\right\|_2 + \lambda_n\left\|x\right\|_1\right\}, \tag{7}$$
    where the regularization is picked to satisfy
    $$\lambda_n = O\left(\frac{c \log d}{n\varepsilon}\right).$$
Return: $\hat{\mu}$
```

I. Proof of Theorem 5.1

Similar as in the proof in Lemma 5.2, let $\mathcal{E}$ be the event that y_i is clipped in the DDG pre-processing stage for some $i \in [n]$:

$$\mathcal{E} \triangleq \bigcup_{i\in[n]}\left\{\left\|Sx_i\right\|_2^2 \geq 1.1 \cdot \left\|x_i\right\|_2^2\right\}.$$

By picking $m = \Omega\left(\log\left(\frac{n}{\beta}\right)\right)$ and applying Lemma G.3 together with the union bound, we have $\Pr_S\left\{\mathcal{E}\right\} \leq \beta$.

Next, we decompose the error as

$$\begin{aligned}
\mathbb{E}\left[\left\|\hat{\mu}-\mu\right\|_2^2\right] &= \mathbb{E}\left[\left\|S^\intercal\left(\frac{1}{n}\sum_i \hat{\mu}_y - \frac{1}{n}\sum_i y_i\right) + \left(S^\intercal \frac{1}{n}\sum_i \hat{\mu}_y - \mu\right)\right\|_2^2\right] \\
&\leq \underbrace{2\,\mathbb{E}\left[\left\|S^\intercal\left(\hat{\mu}_y - \frac{1}{n}\sum_i y_i\right)\right\|_2^2\right]}_{\text{privatization error}} + \underbrace{2\,\mathbb{E}\left[\left\|\left(S^\intercal \frac{1}{n}\sum_i y_i - \mu\right)\right\|_2^2\right]}_{\text{compression error}},
\end{aligned}$$

where we use $\hat{\mu}_y$ to denote the output of the DDG mechanism. Let us bound the privatization error and the compression error separately.

Bounding the privatization error For the first term, observe that $\hat{\mu}_y$ is a function $\left(\mathsf{clip}\left(y_1\right), ..., \mathsf{clip}\left(y_n\right)\right)$, and conditioned on $\mathcal{E}^c$, we have

$$\hat{\mu}_y\left(\mathsf{clip}\left(y_1\right), ..., \mathsf{clip}\left(y_n\right)\right) = \hat{\mu}_y\left(y_1, ..., y_n\right).$$

For simplicity, let us denote them as $\hat{\mu}_{y,\mathsf{cl}}$ and $\hat{\mu}_y$ respectively. Next, we separate the error due to clipping by decompose the privatization error into

$$\begin{aligned}
\mathbb{E}\left[\left\|S^\intercal\left(\hat{\mu}_{y,\mathsf{cl}} - \frac{1}{n}\sum_i y_i\right)\right\|_2^2\right] &\leq \Pr\left\{\mathcal{E}^c\right\}\cdot\mathbb{E}\left[\left\|S^\intercal\left(\hat{\mu}_y - \frac{1}{n}\sum_i y_i\right)\right\|_2^2 \middle| \mathcal{E}^c\right] + c^2(d+1)\Pr\left\{\mathcal{E}\right\} \\
&\leq \mathbb{E}\left[\left\|S^\intercal \underbrace{\left(\hat{\mu}_y - \frac{1}{n}\sum_i y_i\right)}_{\text{error due to DDG}}\right\|_2^2\right] + c^2(d+1)\beta. \qquad (8)
\end{aligned}$$

From the MSE bound of DDG (Kairouz et al., 2021a), we know that with probability 1,

$$\mathbb{E}\left[\left\|S^\intercal\left(\hat{\mu}_y - \frac{1}{n}\sum_i y_i\right)\right\|_2^2 \middle| S\right] = O\left(\frac{c^2m^2}{n^2\varepsilon^2}\right).$$

Therefore the first term in (8) can be controlled by Lemma G.4, and we can bound the privatization error by

$$\mathbb{E}\left[\left\|S^\intercal\left(\hat{\mu}_{y,\mathsf{cl}} - \frac{1}{n}\sum_i y_i\right)\right\|_2^2\right] = O\left(\frac{c^2d}{n^2\varepsilon^2}\right) + c^2(d+1)\beta.$$

Bounding the compression error Next, by Lemma G.2, the compression error can be bounded by

$$\mathbb{E}\left[\left\|\left(S^\intercal \frac{1}{n}\sum_i y_i - \mu\right)\right\|_2^2\right] \leq \frac{2c^2d}{m}.$$

Putting things together, we obtain

$$\mathbb{E}\left[\left\|\hat{\mu}-\mu\right\|_2^2\right] \leq C_1\frac{c^2d}{n^2\varepsilon^2} + \frac{2c^2d}{m} + c^2(d+1)\beta.$$

Therefore if we pick $\beta = \frac{1}{n^2\varepsilon^2}$ (so m has to be $\log\left(n^3\varepsilon^2\right)$), and $m = n^2\varepsilon^2$, we have

$$E\left[\left\|\hat{\mu}-\mu\right\|_2^2\right] \leq C_0\frac{c^2d}{n^2\varepsilon^2}. \qquad (9)$$

J. Proof of Theorem 5.3

To prove (3), we first claim that if there exists a b-bit compression scheme $(\mathcal{C}, \hat{v})$ such that for all $v \in \mathbf{B}_d(c)$, $\mathbb{E}\left[\|\hat{v}-v\|_2^2\right] \leq \gamma^2$, then there exists a γ-covering $C(\gamma)$ of $\mathbf{B}_d(c)$, such that $|C(\gamma)| \leq 2^b$. To see this, observe that $\left\{\mathbb{E}\left[\hat{v}\left(\mathcal{C}(m)\right)\right], m \in [2^b]\right\}$ forms a γ-covering of $\mathbf{B}_d(c)$. This is because for any $v \in \mathbf{B}_d(c)$, it holds that

$$\|\mathbb{E}\left[\hat{v}\right]-v\|_2^2 \overset{\text{(a)}}{\leq} \mathbb{E}\left[\|\hat{v}-v\|_2^2\right] \leq \gamma^2,$$

where (a) holds by Jensen's inequality.

On the other hand, for any γ-covering of $\mathbf{B}_d(c)$, we must have $|C(c)| \geq \frac{\mathsf{vol}(\mathbf{B}_d(c))}{\mathsf{vol}(\mathbf{B}_d(\gamma))} = \left(\frac{c}{\gamma}\right)^d$. Thus we conclude that if $2^b \leq \left(\frac{c}{\gamma}\right)^d$, then $\mathbb{E}\left[\|\hat{\mu}-\mu\|_2^2\right] \geq \gamma^2$, or equivalently

$$\mathbb{E}\left[\|\hat{\mu}-\mu\|_2^2\right] \geq \left(\frac{1}{2^b}\right)^{2/d} c^2.$$

To prove (4), we first impose a product Bernoulli distribution on $\mathbf{B}_d(c)$, upper bound the *quantized* Fisher information (Barnes et al., 2019), and then apply the Cramer-Rao lower bound.

To begin with, let $X \sim \prod_{i\in[d]} \mathsf{Ber}\left(\theta_i\right)$ for some $\theta_i \in [0,1]$. Then $\frac{c}{\sqrt{d}}X \subset \mathbf{B}_d(c)$ almost surely. Next, we claim that for any b-bit unbiased compression scheme $(\mathcal{C}, \hat{v})$ such that$\mathbb{E}\left[\|\hat{v}-v\|_2^2\right] \leq \gamma^2$ for all $v \in \mathbf{B}_d(c)$, $\hat{\theta}\left(X\right) \triangleq \frac{\sqrt{d}}{c}\hat{v}\left(\mathcal{C}\left(\frac{c}{\sqrt{d}}X\right)\right)$ is an unbiased estimator of $\theta = (\theta_1, ..., \theta_d) \in [0,1]^d$ with estimation error bounded by

$$\max_{\theta\in[0,1]^d} \mathbb{E}\left[\left\|\hat{\theta}\left(X\right)-\theta\right\|_2^2\right] \leq \frac{d\gamma}{c}.$$

To see this, observe that

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{\sqrt{d}}{c}\hat{v}\left(\mathcal{C}\left(\frac{c}{\sqrt{d}}X\right)\right)-\theta\right\|_2^2\right] &\leq 2\mathbb{E}\left[\left\|\frac{\sqrt{d}}{c}\hat{v}\left(\mathcal{C}\left(\frac{c}{\sqrt{d}}X\right)\right)-X\right\|_2^2\right]+2\mathbb{E}\left[\|X-\theta\|_2^2\right] \\
&\overset{\text{(a)}}{\leq} 2\frac{d\gamma^2}{c^2}+2\mathbb{E}\left[\|X-\theta\|_2^2\right] \\
&\overset{\text{(b)}}{\leq} 2\frac{d\gamma^2}{c^2}+2\sum_{i\in[d]}\theta_i(1-\theta_i) \\
&\overset{\text{(c)}}{\leq} 2d\left(\frac{\gamma^2}{c^2}+1\right),
\end{aligned} \tag{10}$$

where (a) holds since by assumption $\mathbb{E}\left[\|\hat{v}-v\|_2^2\right] \leq \gamma$, (b) holds since $X_i \sim \mathsf{Ber}\left(\theta_i\right)$, and (c) holds since $\theta_i \in [0,1]$.

Next, we apply (Barnes et al., 2019, Corollary 4), which states that for any b bits (possibly randomized) transform $M : \{0,1\}^d \to \mathcal{Y}$ with $|\mathcal{Y}| \leq 2^b$, the Fisher information $I_Y(\theta)$ with $Y \sim M(\cdot|X)$ and $X \sim \prod_{i\in[d]} \mathsf{Ber}(\theta_i)$ is upper bounded by

$$\min_{M(\cdot|X)} \max_{\theta\in[0,1]^d} \mathsf{Tr}\left(I_Y(\theta)\right) \leq C_1 \min(d,b).$$

Therefore, since $\mathcal{C}$ is a b-bit compression operator (and thus $\hat{\theta}$ can be encoded into b bits too), we must have

$$\max_{\theta\in[0,1]^d} \mathsf{Tr}\left(I_{\hat{\theta}}(\theta)\right) \leq C_1 \min(d,b).$$

Applying Cramer-Rao lower bound yields

$$\max_{\theta\in[0,1]^d} \mathbb{E}\left[\left\|\hat{\theta}-\theta\right\|_2^2\right] \geq \sum_{i\in[d]}\left[I_\theta(\theta)\right]_{i,i} \geq \frac{d^2}{\mathsf{Tr}\left(I_{\hat{\theta}}\left(\theta\right)\right)} \geq C_2\frac{d^2}{\min(b,d)}. \tag{11}$$

Finally, by (10) and (11), we must have

$$2d\left(\frac{\gamma^2}{c^2}+1\right) \geq C_2\frac{d^2}{\min\left(b,d\right)} \iff \gamma^2 \geq \left(C_2\frac{d}{\min\left(b,d\right)}-2\right) \geq C_3\frac{dc^2}{\min(b,d)},$$

for some constants $C_2, C_3 > 0$ and $b \leq \frac{d}{2C_2}$, completing the proof.

K. Proof of Theorem 6.1

The $\frac{1}{2}\varepsilon^2$-concentrated DP is guaranteed by the DDG mechanism. Therefore we only need to analyze the ℓ_2 error $\mathbb{E}\left[\|\hat{\mu}-\mu\|_2^2\right]$. To analyze the ℓ_2 error, we can equivalently formulate it as a sparse linear problem:

$$\hat{\mu}_y = S\mu + \Delta,$$

where $\Delta = \hat{\mu}_y - \frac{1}{n}\sum_i Sx_i$ is the error introduced by the DDG mechanism. We illustrate each steps of the end-to-end transform of Algorithm 6 in Figure 17.

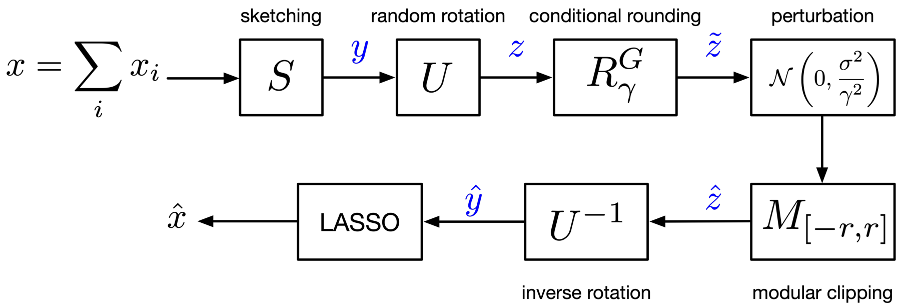


Figure 17. Sparse private aggregation.

Before we continue to analyze the error, we first introduce some necessary definitions.

Definition K.1 (Sufficient conditions for RE (a "soft" RE))**.** We say S satisfies a "soft" RE with parameter (κ, ρ), if

$$\frac{1}{m}\|S\Delta\|_2^2 \geq \frac{1}{8}\kappa\|\Delta\|_2^2 - 50\rho^2\frac{\log(d)}{m}\|\Delta\|_1^2, \text{ for all } \Delta \in \mathbb{R}^d. \tag{12}$$

Remark K.2. Let $S_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. Then S satisfies (12) with $\kappa = \rho = 1$ with probability at least $1 - \frac{e^{-m/32}}{1-e^{-m/32}}$.

Theorem K.3 (Lasso oracle inequality (Theorem 7.19 Wainwright (2019)))**.** *As long as S satisfies* (12) *with* (κ, ρ) *and* $\lambda_n \geq \|S^\intercal(\hat{\mu}_y - \mu_y)\|_\infty / m$ *then following MSE bound holds:*

$$\|\hat{\mu}-\mu\|_2^2 \leq \frac{144|S|}{c_1^2\kappa^2}\lambda_n^2 + \frac{16}{c_1\kappa}\lambda_n\|\mu_{S^c}\|_1 + \frac{32c_2\rho^2}{c_1\kappa}\frac{\log d}{m}\|\mu_{S^c}\|_1^2,$$

for all $S \subseteq [d]$ *with* $|S| \leq \frac{c_1}{64c_2}\frac{m}{\log d}$. *For the Gaussian ensembles, we have* $\kappa = \rho = 1$.

Therefore according to Theorem K.3, to upper bound the estimation error, it suffices to control $\frac{\|S^\intercal(\hat{\mu}_y-\mu_y)\|_\infty}{m}$ (and hence the regularizer λ_n). In the next lemma, we give an upper bound on it.

Lemma K.4. *Let* $d' = c_0 s\log d$, $c' = c\sigma_{\mathsf{max}}(S)$ *and let granularity* $\gamma > 0$, *modulus* $M \in \mathbb{N}$, *noise scale* $\sigma > 0$, *and bias* $\beta \in [0,1)$ *be defined as in Theorem 1 and Theorem 2 in (Kairouz et al., 2021a). Then as long as*

$$M \geq \frac{2}{\gamma}\sqrt{\left(n\left(\gamma^2+4\sigma^2\right)+\frac{4n^2c'^2}{m}\right)\left(\log m + \log\left(\frac{1}{(1-\beta)^n}\right)+\log\left(\frac{8}{\delta}\right)\right)},$$

the following bound holds with probability at least $1-\delta$:

$$\left\|\frac{S^\intercal\left(\hat{\mu}_y-\mu_y\right)}{m}\right\|_\infty \leq \sqrt{\frac{1}{n}\left(\log\left(\frac{d}{(1-\beta)^n}+\log\left(\frac{2}{\delta}\right)\right)\left(\frac{\gamma^2+4\sigma^2}{8}\right)\left(\frac{\max_{i=1,\dots,d}\|S_i\|_2^2}{m}\right)\right)} \tag{13}$$

Corollary K.5. *Let each row of S be generated according to $\mathcal{N}\left(0, \mathbb{I}_d\right)$ and let γ, σ, β be the parameters used in the discrete Gaussian mechanism. If $|x|_0 \leq s$ and*

$$\lambda_n = \sqrt{\frac{1}{n}\left(\log\left(\frac{d}{(1-\beta)^n} + \log\left(\frac{2}{\delta}\right)\right)\left(\frac{\gamma^2+4\sigma^2}{8}\right)\left(\frac{\max_{i=1,\dots,d}\|S_i\|_2^2}{m}\right)\right)}, \tag{14}$$

then with probability at least $1-\delta$,

$$\|\hat{\mu}-\mu\|_2^2 = O\left(\frac{s}{n}\left(\log\left(\frac{d}{(1-\beta)^n} + \log\left(\frac{2}{\delta}\right)\right)\left(\frac{\gamma^2+4\sigma^2}{8}\right)\left(\frac{\max_{i=1,\dots,d}\|S_i\|_2^2}{m}\right)\right)\right).$$

In addition, the communication cost is

$$O\left(s\log d \log\left(\frac{1}{\gamma}\sqrt{\left(n\left(\gamma^2+4\sigma^2\right)+\frac{4n^2c'^2}{m}\right)\left(\log m + \log\left(\frac{1}{(1-\beta)^n}\right) + \log\left(\frac{8}{\delta}\right)\right)}\right)\right)$$

K.1. Parameter selection

Now we pick parameters so that Algorithm 1 satisfies $\frac{1}{2}\varepsilon^2$-concentrated DP and attains the MSE as in the centralized model. To begin with, we first determine σ so that Algorithm 1 satisfies differential privacy.

Privacy analysis To analyze the privacy guarantees, we treat the inner discrete Gaussian mechanism as a black box with the following parameters: effective dimension $d' = m$, ℓ_2 bound (or the clipping threshold) $c' = c\sigma_{\mathsf{max}}(S)$, granularity $\gamma > 0$, noise scale $\sigma > 0$, and bias $\beta \in [0,1)$. Define

$$\begin{aligned}
\Delta_2^2 &\triangleq \min\left\{c'^2 + \frac{\gamma^2 d'}{4} + \sqrt{2\log\left(1/\beta\right)}\gamma\left(c' + \frac{\gamma}{2}\sqrt{d'}\right), \left(c' + \gamma\sqrt{d'}\right)^2\right\} \\
\tau &\triangleq 10\sum_{k=1}^{n-1}\exp\left(-2\pi^2\frac{\sigma^2}{\gamma^2}\frac{k}{k+1}\right) \\
\varepsilon &\triangleq \min\left\{\sqrt{\frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d'}, \frac{\Delta_2}{\sqrt{n}\sigma} + \tau\sqrt{d'}\right\}.
\end{aligned}$$

Then Theorem 1 in (Kairouz et al., 2021a) ensures that Algorithm 1 is $\frac{1}{2}\varepsilon^2$-concentrated DP. With this theorem in hands, we first determine σ. Observe that

$$\varepsilon^2 \leq \frac{\Delta_2^2}{n\sigma^2} + \frac{1}{2}\tau d' \leq \frac{2c'^2}{n\sigma^2} + \frac{2d'}{n(\sigma/\gamma)^2} + 5nd'\exp^{-\pi^2(\sigma/\gamma)^2}.$$

Thus it suffices to set $\sigma = \max\left\{\frac{2c'}{\varepsilon\sqrt{n}}, \frac{\gamma\sqrt{8d'}}{\varepsilon\sqrt{n}}, \frac{\gamma}{\pi^2}\log\left(\frac{20nd'}{\varepsilon^2}\right)\right\} = \tilde{\Theta}\left(\frac{c'}{\varepsilon\sqrt{n}} + \sqrt{\frac{d'}{n}}\frac{\gamma}{\varepsilon}\right)$.

Accuracy analysis We set $\beta = \min\left(\sqrt{\frac{\gamma}{n}}, \frac{1}{n}\right)$, and together with the upper bound (13) and

$$\sigma^2 \preceq \frac{c'^2 + \gamma^2 d'}{n\varepsilon^2} + \gamma^2\log^2\left(\frac{nd'}{\varepsilon^2}\right),$$

we have

$$\lambda_n = \Theta\left(\sqrt{\frac{1}{n}\left(\log d + \log(1/\delta)\right)\left(\frac{c'^2+\gamma^2 d'}{n\varepsilon^2} + \gamma^2\log^2\left(\frac{nd'}{\varepsilon^2}\right) + \gamma^2\right)\left(\frac{\max_{i=1,\dots,d}\|S_i\|_2^2}{m}\right)}\right).$$

Thus it suffices to pick

$$\gamma^2 = O\left(\min\left(\frac{c'^2}{d'}, \frac{c'^2}{n\varepsilon^2\log^2\left(\frac{nd'}{\varepsilon^2}\right)}\right)\right).$$

We summarize the above parameter selection in the following theorem.

Theorem K.6. *By selecting*

$$\begin{aligned}
\beta &= \min\left(\sqrt{\frac{\gamma}{n}}, \frac{1}{n}\right)\\
\sigma &= \max\left\{\frac{2c'}{\varepsilon\sqrt{n}}, \frac{\gamma\sqrt{8d'}}{\varepsilon\sqrt{n}}, \frac{\gamma}{\pi^2}\log\left(\frac{20nd'}{\varepsilon^2}\right)\right\}\\
\gamma^2 &= O\left(\min\left(\frac{c'^2}{d'}, \frac{c'^2}{n\varepsilon^2\log^2\left(\frac{nd'}{\varepsilon^2}\right)}\right)\right)\\
\lambda_n &= \Theta\left(\sqrt{\frac{1}{n}\left(\log d + \log(1/\delta)\right)\left(\frac{c'^2}{n\varepsilon^2}\right)\left(\frac{\max_{i=1,\dots,d}\|S_i\|_2^2}{m}\right)}\right) = O\left(\frac{c\log d}{n\varepsilon}\right)\\
M &= O\left(\sqrt{\left(\frac{d'}{\varepsilon^2}+n^2\right)\left(\log d' + \log\left(1/\delta\right)\right)}\right)
\end{aligned}$$

Algorithm 5 is $\frac{1}{2}\varepsilon^2$-concentrated DP, and with probability at least $1-\delta$,

$$\begin{aligned}
\|\hat{\mu}-\mu\|_2^2 &= O\left(\frac{s(\log d + \log(1/\delta))c'^2}{n^2\varepsilon^2}\left(\frac{\max_{i\in[d]}\|S_i\|_2^2}{d'}\right)\right)\\
&= O\left(\frac{s(\log d + \log(1/\delta))c^2\rho^2_{\mathsf{max}}(S)}{n^2\varepsilon^2}\left(\frac{\max_{i\in[d]}\|S_i\|_2^2}{s\log(d)}\right)\right).
\end{aligned}$$

This establishes Theorem 6.1.

L. Proof of Lemmas

L.1. Proof of Lemma G.4

First, we break $v \in \mathbb{R}^m$ into t blocks

$$v = \begin{bmatrix} v^{(1)}\\ v^{(2)}\\ \vdots\\ v^{(t)}\end{bmatrix},$$

where $v_j \in \mathbb{R}^w$ (recall that $m = w\cdot t$). Then

$$\begin{aligned}
\mathbb{E}\left[\|S^\intercal v\|_2^2\right] &= \frac{1}{t}\mathbb{E}\left[\left\|\sum_{i=1}^t S_i^\intercal v^{(i)}\right\|_2^2\right]\\
&= \frac{1}{t}\mathbb{E}\left[\left\|\sum_{i=1}^t\left(S_i^\intercal v^{(i)} - \mathbb{E}\left[S_i^\intercal v^{(i)}\right]\right) + \sum_{i=1}^t \mathbb{E}\left[S_i^\intercal v^{(i)}\right]\right\|_2^2\right]\\
&\leq \frac{2}{t}\left(\sum_{i=1}^t \mathbb{E}\left[\left\|S_i^\intercal v^{(i)} - \mathbb{E}\left[S_i^\intercal v^{(i)}\right]\right\|_2^2\right] + \left\|\sum_{i=1}^t \mathbb{E}\left[S_i^\intercal v^{(i)}\right]\right\|_2^2\right)\\
&\leq \underbrace{\frac{2}{t}\sum_{i=1}^t \mathbb{E}\left[\left\|S_i^\intercal v^{(i)}\right\|_2^2\right]}_{\text{(a)}} + \underbrace{\frac{2}{t}\left\|\sum_{i=1}^t \mathbb{E}\left[S_i^\intercal v^{(i)}\right]\right\|_2^2}_{\text{(b)}}.
\end{aligned}$$

Now we bound each term separately. To bound (a), observe that for all $i \in [t]$,

$$\mathbb{E}\left[\left\|S_i^{\mathsf{T}} v^{(i)}\right\|_2^2\right] \leq \mathbb{E}\left[N(S_i)\mathbb{E}\left[\left\|v^{(i)}\right\|_2^2 \middle| S_i\right]\right] \leq \mathbb{E}\left[N(S_i)\right] B^2,$$

since by assumption $\mathbb{E}\left[\left\|v^{(i)}\right\|_2^2 \middle| S_i\right] \leq B^2$ almost surely, where $N(S_i)$ is the maximum amount of 1s in w rows of S_i. Notice that this amount is the same as the maximum load of throwing d balls into w bins. Applying a Chernoff bound, this quantity can be upper bounded by $\mathbb{E}[N(S_i)] \leq \frac{(e+1)d}{w}$, so (a) is bounded by

$$\frac{2}{t}\sum_{i=1}^{t}\mathbb{E}\left[\left\|S_i^{\mathsf{T}} v^{(i)}\right\|_2^2\right] \leq \frac{2(e+1)d}{wt}\sum_{i=1}^{t}\left\|v^{(i)}\right\|_2^2 \leq \frac{8d}{m}\left\|v\right\|_2^2.$$

To bound (b), observe that

$$\mathbb{E}\left[S_i^{\mathsf{T}} v^{(i)}\right] = \frac{1}{w}\begin{bmatrix}\sum_{j=1}^{w} v_j^{(i)} \\ \sum_{j=1}^{w} v_j^{(i)} \\ \vdots \\ \sum_{j=1}^{w} v_j^{(i)}\end{bmatrix} = \left(\frac{1}{w}\sum_{j=1}^{w} v_j^{(i)}\right)\cdot \mathbf{1}_d,$$

where $\mathbf{1}_d \triangleq [1, ..., 1]^{\mathsf{T}} \in \mathbb{R}^d$. Therefore, summing over $i \in [t]$, we have

$$\sum_{i=1}^{t}\mathbb{E}\left[S_i^{\mathsf{T}} v^{(i)}\right] = \sum_{i=1}^{t}\left(\frac{1}{w}\sum_{j=1}^{w} v_j^{(i)}\right)\cdot \mathbf{1}_d = \frac{1}{w}\left(\sum_{i=1}^{m} v_i\right)\cdot \mathbf{1}_d.$$

Thus we can bound (b) by

$$\frac{2}{t}\left\|\sum_{i=1}^{t}\mathbb{E}\left[S_i^{\mathsf{T}} v^{(i)}\right]\right\|_2^2 \leq \frac{2d}{tw}\left(\frac{\left(\sum_{i=1}^{m} v_i\right)^2}{w}\right) \leq \frac{2d}{m}\left\|v\right\|_2^2,$$

where the last inequality follows from the Cauchy-Schwartz inequality.

Putting (a) and (b) together, the proof is complete.

L.2. Proof of Lemma 5.2

For simplicity, let $\mu \triangleq \frac{1}{n}\sum_i x_i$, and $N \sim \mathcal{N}\left(0, \sigma^2 I_m\right)$. Define

$$\mathcal{E}_\alpha \triangleq \bigcup_{i\in[n]}\left\{\left\|Sx_i\right\|_2^2 \geq (1+\alpha)\cdot\left\|x_i\right\|_2^2\right\}.$$

We will pick $m = \Omega\left(\frac{1}{\alpha^2}\log\left(\frac{n}{\beta}\right)\right)$, so by Lemma G.3 and the union bound $\Pr_S\left\{\mathcal{E}_\alpha^c\right\} \leq \beta$. Then the MSE can be computed as

$$\begin{aligned}
&\mathbb{E}\left[\left\|S^{\mathsf{T}}\left(\frac{1}{n}\sum_i \mathsf{clip}\left(Sx_i\right) + N\right) - \mu\right\|_2^2\right] \\
&\overset{\text{(a)}}{=} \mathbb{E}\left[\left\|S^{\mathsf{T}}\left(\frac{1}{n}\sum_i \mathsf{clip}\left(Sx_i\right)\right) - \mu\right\|_2^2\right] + \mathbb{E}\left[\left\|S^{\mathsf{T}}N\right\|_2^2\right] \\
&\overset{\text{(b)}}{\leq} \mathbb{E}\left[\left\|S^{\mathsf{T}}\left(\frac{1}{n}\sum_i \mathsf{clip}\left(Sx_i\right)\right) - \mu\right\|_2^2 \middle| \mathcal{E}_\alpha^c\right]\cdot \Pr\left\{\mathcal{E}_\alpha^c\right\} + (d+1)\cdot\Pr\left\{\mathcal{E}_\alpha\right\} + \mathbb{E}\left[\left\|S^{\mathsf{T}}N\right\|_2^2\right] \\
&\overset{\text{(c)}}{\leq} \mathbb{E}\left[\left\|\frac{1}{n}\sum_i S^{\mathsf{T}}Sx_i - \mu\right\|_2^2\right] + (d+1)\beta + \mathbb{E}\left[\left\|S^{\mathsf{T}}N\right\|_2^2\right],
\end{aligned}$$

where (a) holds since $\mathbb{E}\left[S^\intercal N|S\right]=0$ almost surely, (b) holds since $\left\|S^\intercal \nu\right\|_2^2 \leq d$ for all count-sketch matrix S and all $\left\|\nu\right\|_2 \leq 1$ (so $\left\|S^\intercal\left(\frac{1}{n}\sum_i \mathsf{clip}\left(Sx_i\right)\right)-\mu\right\|_2^2 \leq d+1$), and (c) holds since conditioned on $\mathcal{E}^c_\alpha$, $\mathsf{clip}\left(Sx_i\right)=Sx_i$ for all i.

Next, we control each term separately. The first term can be controlled using Lemma G.2, which gives

$$\mathbb{E}\left[\left\|\frac{1}{n}\sum_i S^\intercal S x_i - \mu\right\|_2^2\right] \leq \frac{2d}{m}.$$

The third term can be computed as follows:

$$\mathbb{E}\left[\left\|S^\intercal \cdot N\right\|_2^2\right] = \mathbb{E}\left[\mathbb{E}\left[S^\intercal\cdot N|S\right]\right] \leq \sigma^2 d = \frac{8d(1+\alpha)\log\left(1.25\right)}{n^2\varepsilon^2}.$$

Thus we arrive at

$$\mathbb{E}\left[\left\|\hat{\mu}-\mu\right\|_2^2\right] \leq \frac{2d}{m} + (d+1)\beta + \frac{8d(1+\alpha)\log(1.25)}{n^2\varepsilon^2}.$$

Therefore, if we pick $\beta = \frac{\kappa}{n^2\varepsilon^2}$ (so $m=\Omega\left(\frac{1}{\alpha^2}\log\left(\frac{n^3\varepsilon^2}{\kappa}\right)\right)$), and $m=\frac{n^2\varepsilon^2}{\kappa}$, we have

$$E\left[\left\|\hat{\mu}-\mu\right\|_2^2\right] \leq \frac{8d\log(1.25) + d\left(8\alpha\cdot\log(1.25)+\left(3+\frac{1}{d}\right)\kappa\right)}{n^2\varepsilon^2}. \tag{15}$$

Notice that we can make α and κ small, say $\alpha=\kappa=0.1$, and the MSE will be closed to the uncompressed one.

L.3. Proof of Lemma K.4

We first set some notation. Let $\mu_y = S\mu = \frac{1}{n}\sum_i y_i$, $z = U\mu_y$ (where U is the random rotation matrix), $\tilde{z} = \frac{1}{n}\sum_i\left(R^G_\gamma(z_i)+\mathcal{N}_\mathbb{Z}(0,\sigma^2/\gamma^2)\right)$, $\hat{z}=\frac{1}{n}M_{[-r,r]}(n\tilde{z})$ and finally $\hat{\mu}_y = U^\intercal\hat{z}$, where $R^G_\gamma(\cdot)$ is the randomized rounding, $\mathcal{N}_\mathbb{Z}$ is the discrete Gaussian noise, and $M_{[-r,r]}(\cdot)$ is the module clipping (details can be found in (Kairouz et al., 2021a)).

Now, we can write the left-hand side of (13) as

$$\left\|\frac{S^\intercal\left(\hat{\mu}_y-\frac{1}{n}\sum_i y_i\right)}{m}\right\|_\infty = \left\|\frac{S^\intercal U^\intercal\left(\hat{z}-\tilde{z}+\tilde{z}-z\right)}{m}\right\|_\infty \leq \underbrace{\left\|\frac{S^\intercal U^\intercal\left(\hat{z}-\tilde{z}\right)}{m}\right\|_\infty}_{\text{(1): module error}} + \underbrace{\left\|\frac{S^\intercal U^\intercal\left(\tilde{z}-z\right)}{m}\right\|_\infty}_{\text{(2): rounding error}}.$$

Define $S' \triangleq \frac{S^\intercal U^\intercal}{\sqrt{m}}$ and let S'_i be the i-th row of S' for all $i=1,...,d$. Now we bound (1) and (2) separately.

Bounding the module error Observe that since $\left\|y\right\|_2^2 \leq c'^2$, by Lemma 30 in (Kairouz et al., 2021a) we have

$$\forall t\in\mathbb{R}\forall j\in[m]\,\mathbb{E}\left[\exp\left(tz_j\right)\right] = \mathbb{E}\left[\exp\left(t\left(Un\mu_y\right)_j\right)\right] \leq \exp\left(\frac{t^2n^2c'^2}{2m}\right).$$

Applying the union bound and the Chernoff's bound yields

$$\Pr\left\{\max_{j=1,...,m}\left|z_j\right| > \sqrt{2\frac{n^2c'^2}{m}\left(\log m + \log\left(\frac{8}{\delta}\right)\right)}\right\} \leq \frac{\delta}{4}, \tag{16}$$

where the randomness is over the random rotational matrix U.

On the other hand, from Proposition 26, we have

$$\mathbb{E}\left[\exp\left(t\left(n\tilde{z}_j - nz_j\right)\right)\right] \leq \frac{\exp\left(\frac{nt^2\left(\gamma^2+4\sigma^2\right)}{8}\right)}{\left(1-\beta\right)^n}.$$

Applying the Markov's inequality and the union bound, we obtain

$$\Pr\left\{\max_{j\in[m]}\left|n\tilde{z}_j - nz_j\right| \geq t\right\} \leq \frac{m}{(1-\beta)^n}\exp\left(-\frac{2t^2}{n\gamma^2+4\delta^2}\right).$$

Thus picking $t=\sqrt{\frac{n(\gamma^2+4\sigma^2)}{2}\left(\log\left(\frac{1}{(1-\beta)^n}\right)+\log\left(\frac{4}{\delta}\right)\right)}$ yields

$$\Pr\left\{\max_{j\in[m]}\left|n\tilde{z}_j - nz_j\right| \geq \sqrt{\frac{n\left(\gamma^2+4\sigma^2\right)}{2}\left(\log\left(\frac{1}{(1-\beta)^n}\right)+\log\left(\frac{4}{\delta}\right)\right)}\right\} \leq \frac{\delta}{4}. \tag{17}$$

Putting (16) and (17) together, we arrive at

$$\Pr\left\{\max_{j\in[m]}\left|\tilde{z}_j\right| \geq \sqrt{\left(n\left(\gamma^2+4\sigma^2\right)+\frac{4n^2c'^2}{m}\right)\left(\log m+\log\left(\frac{1}{(1-\beta)^n}\right)+\log\left(\frac{8}{\delta}\right)\right)}\right\} \leq \frac{\delta}{2}, \tag{18}$$

where we use the fact that $\sqrt{a}+\sqrt{b}\leq\sqrt{2(a+b)}$ and the union bound.

Finally, observe that as long as $\left\|\tilde{z}\right\|_\infty \leq r$, $\hat{z}\triangleq M_{[-r,r]}(\tilde{z})=\tilde{z}$. Thus by picking

$$r\triangleq\sqrt{\left(n\left(\gamma^2+4\sigma^2\right)+\frac{4n^2c'^2}{m}\right)\left(\log m+\log\left(\frac{1}{(1-\beta)^n}\right)+\log\left(\frac{8}{\delta}\right)\right)},$$

we have

$$\Pr\left\{\left\|\frac{S^\intercal U^\intercal\left(\hat{z}-\tilde{z}\right)}{m}\right\|_\infty > 0\right\} \leq \frac{\delta}{2}.$$

Bounding the module error First notice that

$$\Pr\left\{\left\|\frac{S^\intercal U^\intercal\left(n\hat{z}-n\tilde{z}\right)}{m}\right\|_\infty \geq t\right\} = \Pr\left\{\max_{i\in[d]}\langle S_i', (n\tilde{z}-nz)\rangle \geq t\right\} + \Pr\left\{\max_{i\in[d]} -\langle S_i', (n\tilde{z}-nz)\rangle \geq t\right\}$$

Thus we have

$$\begin{aligned}\Pr\left\{\max_{i\in[d]}\langle S_i', (n\tilde{z}-nz)\rangle \geq t\right\} &\leq \sum_{i\in[d]}\Pr\left\{\exp\left(\langle\lambda S_i', nz-n\tilde{z}\rangle\right)\geq\exp\left(\lambda t\right)\right\}\\ &\overset{\text{(a)}}{\leq}\sum_{i\in[d]}\frac{\exp\left(\frac{n(\gamma^2+4\sigma^2)}{8}\left\|S_i'\right\|_2^2\lambda^2-\lambda t\right)}{(1-\beta)^n}\\ &\overset{\text{(b)}}{\leq}\frac{d}{(1-\beta)^n}\exp\left(-\frac{2mt^2}{n\left(\gamma^2+4\sigma^2\right)\max_{i\in[d]}\left\|S_i\right\|_2^2}\right),\end{aligned}$$

where (a) holds by Proposition 26 in (Kairouz et al., 2021a), and (b) holds by picking λ properly.

On the other hand,

$$\begin{aligned}\Pr\left\{\max_{i\in[d]} -\langle S_i', (n\tilde{z}-nz)\rangle \geq t\right\} &\leq \sum_{i\in[d]}\Pr\left\{\exp\left(\langle -\lambda S_i', nz-n\tilde{z}\rangle\right)\geq\exp\left(\lambda t\right)\right\}\\ &\overset{\text{(a)}}{\leq}\frac{d}{(1-\beta)^n}\exp\left(-\frac{2mt^2}{n\left(\gamma^2+4\sigma^2\right)\max_{i\in[d]}\left\|S_i\right\|_2^2}\right),\end{aligned}$$

where (a) holds due to the same reason.

Thus picking

$$t = \sqrt{n\left(\log\left(\frac{d}{(1-\beta)^n} + \log\left(\frac{4}{\delta}\right)\right)\left(\frac{\gamma^2 + 4\sigma^2}{8}\right)\left(\frac{\max_{i=1,\ldots,d} \|S_i\|_2^2}{m}\right)\right)}$$

yields

$$\Pr\left\{\left\|\frac{S^\intercal U^\intercal\left(n\hat{z} - n\tilde{z}\right)}{m}\right\|_\infty \geq t\right\} = \Pr\left\{\left\|\frac{S^\intercal U^\intercal\left(\hat{z} - \tilde{z}\right)}{m}\right\|_\infty \geq t/n\right\} \leq \frac{\delta}{2}.$$