MULTI-EPOCH MATRIX FACTORIZATION MECHANISMS FOR PRIVATE MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce new differentially private (DP) mechanisms for gradient-based machine learning (ML) training involving multiple passes (epochs) of a dataset, substantially improving the achievable privacy-utility-computation tradeoffs. Our key contribution is an extension of the online matrix factorization DP mechanism to multiple participations, substantially generalizing the approach of Denisov et al. (2022). We first give a non-trivial reduction of the problem with per-iteration vector contributions to the simpler one of scalar contributions. Using this, we formulate the construction of optimal (in total squared error at each iterate) matrix mechanisms for SGD variants as a convex program. We provide a closed form solution to the dual function, leading directly to an efficient optimization algorithms.

While tractable, both solving the convex problem offline and computing the necessary noise masks during training can become prohibitively expensive when many training steps are necessary. To address this, we design a Fourier-transform-based mechanism with significantly less computation and only a minor utility decrease.

Extensive empirical evaluation on two tasks, example-level DP for image classification and user-level DP for language modeling, demonstrate substantial improvements over the previous state-of-the-art. Though our primary application is to ML, we note our main DP results are applicable to arbitrary linear queries and hence may have much broader applicability.

1 INTRODUCTION

Differentially private stochastic gradient descent (DPSGD) is the de facto standard algorithm for DP machine learning (ML) (Song et al., 2013; Abadi et al., 2016). However, obtaining state-of-the-art privacy-utility tradeoffs critically requires use of privacy amplification techniques like shuffling (Erlingsson et al., 2019; Feldman et al., 2022) or (Poisson) subsampling (Bassily et al., 2014; Zhu & Wang, 2019; Wang et al., 2018). These in turn require strong assumptions on the manner in which data is processed that are rarely valid in applications, as implementing these procedures is often impractical due to, for example, memory constraints (Kairouz et al., 2021).

Kairouz et al. (2021) recently proposed the DP-FTRL framework that avoids reliance on amplification by sampling—by using DP streaming of prefix sums (Dwork et al., 2010; Chan et al., 2011; Honaker, 2015) as the core primitive for DP ML. In doing so, DP-FTRL can often match (or outperform) DPSGD in privacy-utility tradeoffs. Indeed, this algorithm enabled McMahan & Thakurta (2022) to train the first known provably DP ML model on user data in a production setting.

Several works have since focused on this primitive as an instantiation of the streaming matrix mechanism; in particular, Denisov et al. (2022) showed that leveraging optimal matrix mechanisms led to significant empirical improvements, though their work was restricted to the single-epoch setting.

In this work, and as shown in Figure 1, we achieve new privacy-utility tradeoffs, under comparable computation costs, that outperform all prior work, including DPSGD with amplification. Our methods show improvement across all measured privacy levels, which reach as low as $\varepsilon \approx 2$. To accomplish this, we significantly extend the single-participation framework of Denisov et al. (2022) to the *multiple-participation setting*, where each example (or user) may contribute to multiple elements of the data matrix which we process with a DP linear operator. This extension is reflected in a gen-

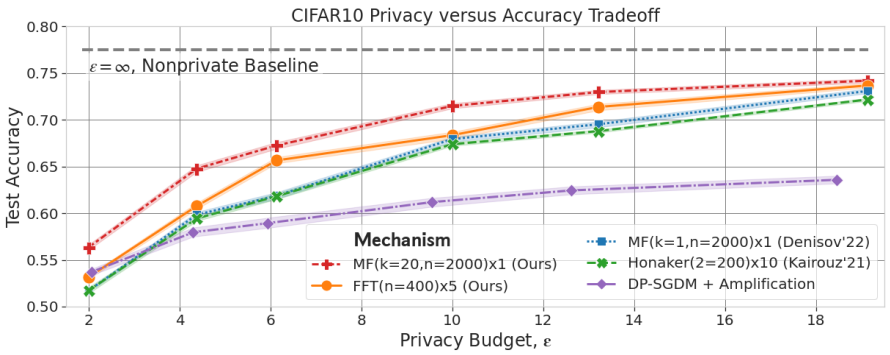

Figure 1: **Our optimal multi-epoch matrix mechanism and FFT-based mechanism outperform all others, including DPSGD with amplification**, for all privacy budgets considered. Models are trained for 20 epochs on CIFAR10 with a batch size of 500. 95% confidence intervals are bootstrapped from 12 repeated runs of each mechanism.

eralized geometry of the optimization problem presented in Denisov et al. (2022). It is factorizing matrices with respect to this geometry that allows us to achieve our results.

We also explore the computational tradeoffs of our approaches. In particular, due to quadratic or worse scaling of the optimal matrix factorizations, using this approach with large numbers of steps can be intractable. To reduce this cost, we propose and investigate an approach based on the Fast Fourier Transform (FFT) (Nussbaumer, 1981), which is near-optimal for the single-epoch setting and efficiently computable for most, if not all, ML settings. Indeed, we find this approach still outperforms the mechanisms from the extant literature, even under multiple participations.

Contributions

1. We provide a framework for computing the sensitivity of matrix mechanisms under general participation patterns. To do this, we prove a new theorem bounding sensitivity for multi-dimensional data contributions based on an extension of the Birkhoff-von Neumann theorem. This allows us to reduce the problem to that of measuring sensitivity for scalar contributions alone (Section 2).
2. We extend the results of Denisov et al. (2022) to the optimization problems corresponding to these generalized notions of sensitivity, showing that the algorithms proposed there can be applied in our setting (Section 3).
3. We propose and analyze a computationally-efficient factorization based on the Fourier transform which is near optimal for the single-pass setting and can be efficiently extended to handle multiple passes (Section 4).
4. We perform detailed empirical comparisons of our mechanisms with both the prior matrix mechanism approaches and DPSGD. We show that the methods proposed here outperform all others (in particular, DPSGD with amplification), in all settings under consideration, and without any need for privacy amplification (Section 5).
5. We will upload all code used in the final manuscript.

Related Works The core privacy primitive considered here, the matrix mechanism (Li et al., 2015), has a long history of both study and application in the literature (McKenna et al., 2018; Edmonds et al., 2020; Yuan et al., 2016; Hardt & Talwar, 2010), though much of this work occurred in the offline setting. Fichtenberger et al. (2022); Denisov et al. (2022) independently applied these matrix-mechanism approaches to the setting of *adaptive streams*, where outputs are released one-by-one and privacy analysis must account for an adversary adaptively defining the inputs. Denisov et al. (2022) connected the matrix mechanism literature to differentially private model training, via the DPFTRL algorithm of Kairouz et al. (2021), and showed that computing optimal factorizations could dramatically improve the privacy/utility/computation tradeoffs required for DP model training in the case of a single pass over the training data.

Example- and user-level DP, and the connection to FL In addition to example-level DP, we consider user-level DP. As observed by McMahan et al. (2017), private versions of federated learning (FL) algorithms are well suited to providing user-level DP, as bounding the sensitivity of a single user's contribution to an aggregate update is made straightforward by the per-user data processing pattern inherent in FL. However, our primary application is to datacenter training, where user data can be processed in a fixed shuffled order, unlike cross-device FL. We use the term 'participation' to denote the event that an example (user/client in FL) contributes to the gradient sum (or a model update in FL) for a given step/iteration (round in FL). Individual per-example (or user) contributions are scaled so their maximum ℓ_2 norm is ζ. All of our mechanisms compute sums over individual clipped contributions, and then post-process by dividing by the batch size (or clients per round) to compute an average gradient or update. We generally assume $\zeta = 1$, applying appropriate scaling in implementations as needed. Notation is summarized in Appendix A.

2 Privacy for adaptive streams with multiple participations

We generalize the adaptive streaming (continual release) setting of Denisov et al. (2022, Sec. 2) to multiple participations as follows. We assume a database of m examples (records) that is processed in a streaming fashion over n steps. A set of examples is selected on each step i, and processed via an adaptively chosen function (e.g., computing a gradient at the current model), producing a vector of ℓ_2 norm at most ζ. These vectors are summed and provided to the DP mechanism as $\mathbf{x}_i \in \mathbb{R}^d$, which then releases a privatized function of $[\mathbf{x}_1, \ldots, \mathbf{x}_i]$, the stream so far. When a particular example contributes to the sum $\mathbf{x}_i$, we say it participates on step i. We are primarily interested in the case where $m < n$, and hence each example is used on more than one step.

Two data streams $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are said to be neighboring if they differ in the contributions derived from a single example, either by zeroing out all of its contributions, or by replacing them arbitrarily subject to the norm bound. Thus, the participation pattern does not change (all records contribute to the same steps in $\mathbf{x}$ and $\tilde{\mathbf{x}}$, with only the vectors associated with one record changing). In order to tightly compute the sensitivity of a mechanism on adaptive streams, we limit the possible *participation patterns*, defined as the set Π of sets $\pi \subseteq [n]$ of steps that any record might possibly participate in.

Assuming each record contributes at most once (**single-participation**, $\Pi = \{\{1\}, \{2\}, \ldots \{n\}\}$), recovers the standard streaming setting. This captures, for example, training with minibatch SGD using a single pass (epoch) over a training dataset. At the other extreme, we have **every-step participation** with $\Pi = \{[n]\}$ where each record contributes to every step. This captures learning with full gradient descent, where we compute the gradient on the full training dataset on every iteration.

For this work, we consider a generalization of the above two, (k, b)**-participation**, where each example participates at most k times, with any adjacent participations exactly[1] b steps apart: formally, Π is the set of all π such that $|\pi| \leq k$, and if $\pi = \{i_1, \ldots, i_k\}$, we have $\forall j \in \{2, \ldots, k\}, i_j - i_{j-1} = b$. Note $(k = 1, b = n)$-participation recovers the single-epoch setting, and $(k = n, b = 1)$-participation recovers every-step participation, and for example $(k = 3, b = 2)$-participation has $\Pi = \{\{1, 3, 5\}, \{2, 4, 6\}\}$. This general version covers multi-epoch SGD training using a data processing pattern typically well-supported by modern ML infrastructure.[2] The only requirement is that rather than shuffling the dataset for each epoch, the dataset is shuffled once and the same order of minibatches is used for each epoch. With this setup, k epochs of training with a batch size B gives $n = mk/B$ total training steps, and satisfies $(k, m/B)$-participation.

Sensitivity for linear queries on multi-participation adaptive streams Consider a full-rank square query matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$; we wish to compute the function $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ in a differentially private manner, where we consider inputs $\mathbf{x}$ and outputs $\mathbf{A}\mathbf{x}$ to be elements of some n-dimensional real vector space—we will later show each $\mathbf{x}_i, i \in [n]$ can be an arbitrary vector. We utilize the ma-

[1] We will see this strict requirement simplifies the optimization problem used to find optimal mechanisms, and indeed leads to better privacy/utility tradeoffs by limiting sensitivity.

[2] This is in contrast to Poisson sampling of examples/users or even independently sampling fixed-sized batches with replacement across steps, as is assumed by many works utilizing amplification-via-sampling arguments (Abadi et al., 2016). Many works in fact process batches in a shuffled order without replacement and then (technically incorrectly) apply the DP analysis for, e.g., Poisson sampling. Indeed, we perform this same—incorrect—analysis for our DP-SGD baseline because it reproduces the state-of-the-art results for DPSGD.

trix mechanism (Li et al., 2015), which, provided a factorization $\mathbf{A} = \mathbf{BC}$, computes the estimate

$$\widehat{\mathbf{Ax}} = \mathbf{B}\left(\mathbf{Cx} + \mathbf{z}\right), \tag{1}$$

where $\mathbf{z}$ is a sample from appropriately scaled isotropic Gaussian noise. The scale is determined by the *sensitivity* of the mapping $\mathbf{x} \mapsto \mathbf{Cx}$; roughly, how much outputs of this mapping can vary (in ℓ_2 norm) when we swap the input stream $\mathbf{x}$ for a neighboring one $\tilde{\mathbf{x}}$.

Let $\mathbf{N}$ be the set of all pairs of neighboring streams $\mathbf{x}$ and $\mathfrak{D} := \{\mathbf{x} - \tilde{\mathbf{x}} \mid (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathbf{N}\}$ represent the set of all possible deltas between neighboring $\mathbf{x}, \tilde{\mathbf{x}}$. The definition of $\mathfrak{D}$ implies it is symmetric ($\mathbf{u} \in \mathfrak{D} \Rightarrow -\mathbf{u} \in \mathfrak{D}$). We will say a $\mathfrak{D}$ **satisfies the participation pattern** Π if the indices of all nonzero elements in each vector $\mathbf{u} \in \mathfrak{D}$ corresponds to an element of Π. Critically, for linear queries $\mathfrak{D}$ fully captures the "geometry" of the allowed participation pattern:

Definition 1 (Sensitivity of Matrix Factorization Mechanisms of Equation 1)**.** *The sensitivity of a matrix factorization mechanism is*

$$\mathrm{sens}_{\mathfrak{D}}(\mathbf{C}) = \sup_{(\mathbf{x},\tilde{\mathbf{x}})\in\mathbf{N}} \|\mathbf{Cx} - \mathbf{C}\tilde{\mathbf{x}}\|_2 = \sup_{\mathbf{u}\in\mathfrak{D}} \|\mathbf{Cu}\|_2. \tag{2}$$

Convexity of $\|\mathbf{Cu}\|_2$ in $\mathbf{u}$ implies that $\sup_{\mathbf{u}\in\mathfrak{D}} \|\mathbf{Cu}\|_2 = \sup_{\mathbf{u}\in\mathrm{conv}(\mathfrak{D})} \|\mathbf{Cu}\|_2$, and hence without loss of generality, we take $\mathfrak{D}$ to be convex as needed. It is illustrative to consider some specific $\mathfrak{D}$s for scalar per-step contributions with $\zeta = 1$. Single-participation corresponds to $\mathfrak{D} = \mathrm{conv}\{\alpha e_i | \alpha \in [-1, 1], i \in [n]\}$ where $e_i, i \in [n]$ are the standard basis vectors. Noting $\|\mathbf{Cu}\| = \|-\mathbf{Cu}\|$ and the convexity of $\|\mathbf{Cu}\|$, we see the maximum will be achieved at some e_i, recovering the 'max-ℓ_2-norm-over-columns' measurement of sensitivity, initially presented in Li et al. (2015, Proposition 3). Every-step participation corresponds to the ℓ_∞ ball, $\mathfrak{D} = \{\mathbf{x} \mid \|\mathbf{x}\|_\infty \leq 1\}$.

Reduction to per-iterate scalar contributions In ML, examples are used to calculate gradients of $d > 1$ dimensions, and so we wish to consider $\mathbf{x} \in \mathbb{R}^{n\times d}$, with rows $\mathbf{x}_i \in \mathbb{R}^d$ corresponding to the sum of gradients for examples participating in step i. In order to compute sensitivity, one may hope that the sensitivity for each $\mathbf{x}_i \in \mathbb{R}^d$ can be bounded by only considering $x_i \in \mathbb{R}$. More formally, consider a fixed participation pattern Π, and further assume (without loss of generality) $\zeta = 1$. Then, for vector-valued contributions we have

$$\mathfrak{D}^d_\Pi = \mathrm{conv}\left\{\mathbf{G} \in \mathbb{R}^{n\times d} \mid \exists \pi \in \Pi \text{ s.t. } \|\mathbf{G}_{[i,:]}\|_2 \leq 1 \text{ for } i \in \pi \text{ and } \mathbf{G}_{[i,:]} = \mathbf{0} \text{ for } i \notin \pi\right\}.$$

In the $d = 1$ case, we have a much simpler polytope, $\mathfrak{D}^1_\Pi = \mathrm{conv}(\mathcal{D}^1_\Pi)$ where

$$\mathcal{D}^1_\Pi = \bigcup_{\pi\in\Pi} \Big\{\mathbf{u} \in \mathbb{R}^n \mid \mathbf{u}_i \in \{-1, 1\} \text{ if } i \in \pi, 0 \text{ otherwise}\Big\}.$$

In order to relate $\mathrm{sens}_{\mathfrak{D}^d_\Pi}(\mathbf{C})$ to $\mathrm{sens}_{\mathfrak{D}^1_\Pi}(\mathbf{C})$, the key is to consider full-participation:

Theorem 2.1. *Let* $\mathbf{C} \in \mathbb{R}^{n\times k}$ *which satisfies* $\|\mathbf{Cu}\|_2 \leq 1\ \forall \mathbf{u} \in \{-1, 1\}^k$*, and let* $\mathbf{G} \in \mathbb{R}^{k\times d}$ *such that each row* $\mathbf{G}_{[i,:]}$ *for* $i \in [k]$ *satisfies* $\|\mathbf{G}_{[i,:]}\|_2 \leq 1$*. Then,* $\|\mathbf{CG}\|_F \leq 1$*.*

Our desired result then follows immediately:

Corollary 2.1. *When per-step contributions are bounded by* $\zeta = 1$*, for any participation pattern* Π *and dimensionality* $d \geq 1$*, we have* $\mathrm{sens}_{\mathfrak{D}^d_\Pi}(\mathbf{C}) \leq \mathrm{sens}_{\mathfrak{D}^1_\Pi}(\mathbf{C})$*.*

Corollary 2.1 implies that considering 'participation vectors', which are ± 1 in the positions where a given user may participate, is sufficient to bound the sensitivity for the 'participation matrices' containing a user's contributed gradients.We will elide this distinction and generally speak of participation vectors determining sensitivity below.

Computing sensitivity (explicitly) We now turn to computing the sensitivity of Definition 1, as enabled by Corollary 2.1. First, observe that since $\mathcal{D}^1_\Pi$ is a finite set; thus, a direct calculation by using Eq. (2) is often possible. However, $|\mathcal{D}^1_\Pi| = |\Pi| 2^k$, and observing the symmetry $\|\mathbf{Cu}\| = \|\mathbf{C}(-\mathbf{u})\|$ can reduce the computational cost only by half. In general $|\Pi|$ may be exponential in k, but in the special case of (k, b)-participation, we have $|\Pi| = b$ (the number of steps in one epoch). Hence, for modest numbers of epochs k, directly computing sensitivity is possible, e.g., we need to consider only $252 \cdot 2^5 = 8064$ vectors $\mathbf{u} \in \mathcal{D}^1_\Pi$ for our StackOverflow experiments in Section 5.2.

Difficulty of computing $\mathrm{sens}(\mathbf{C})$ However, in general computing $\mathrm{sens}(\mathbf{C})$ is a convex quadratic *maximization* problem over a convex set, which can be NP-hard. Even the simple example of computing the sensitivity for an arbitrary matrix $\mathbf{C}$ under every-step participation is NP-hard, as this problem is exactly that of computing the $L_\infty - L_2$ operator norm (Tropp, 2004). (In fact, it is useful to observe $\mathrm{sens}_{\mathfrak{D}}(\cdot)$ can always be viewed as an operator norm, see Appendix B).

If $\mathbf{X}$ is constrained to have only nonnegative elements, one may reduce the problem to computing a maximum over a number of terms linear in $|\Pi|$. This can be seen by noting that the maximum of $\mathbf{u}^\top \mathbf{X}\mathbf{u}$ must be achieved by a nonnegative vector $\mathbf{u}$.[3]

Upper-bounding sensitivity In situations in which enumerating $\mathcal{D}^1_\Pi$ is impractical, we can instead look to efficient upper bounds on the sensitivity of $\mathbf{C}$. In the case of (k, b)-participation, one efficient method of computing upper bounds for the multiple-participation sensitivity of $\mathbf{C}$ has shown itself to be particularly useful:

Theorem 2.2. *Let $\mathbf{C} \in \mathbb{R}^{n\times n}$, and take some participation pattern Π. Assume $\mathfrak{D}$ to be of vectors in $\mathbb{R}^n$ that satisfies Π, and that the absolute value of each entry of $\mathbf{u}$ is at most 1 $\forall \mathbf{u} \in \mathfrak{D}$. Suppose $|\pi| \leq k\ \forall \pi \in \Pi$. With $\mathbf{C}[:, \pi]$ representing to selecting the columns of the matrix $\mathbf{C}$ indexed by π and $\|\cdot\|_2$ the spectral matrix norm, let $\lambda = \max_{\pi\in\Pi} \|\mathbf{C}[:, \pi]\|_2$. Then the map $\mathbf{x} \mapsto \mathbf{C}\mathbf{x}$ has ℓ_2 sensitivity at most $\lambda\sqrt{k}$ with respect to $\mathfrak{D}$.*

In the (k, b)-participation case, $|\Pi| = b$. The complexity of computing the largest eigenvalue of the subselected $\mathbf{C}$ matrix is cubic in k. Thus, computing this upper bound is of order bk^3, easily computable for the range of k, b considered here ($k \leq 100, b \leq 500$).

Differential Privacy Guarantee Using our generalization of adaptive streams to multiple participations we obtain the following result (a straightforward generalization of Denisov et al. (2022, Theorem 2.1)). The proof is identical that in Denisov et al. (2022), except we replace the sensitivity bound with that for multiple participations obtained via Corollary 2.1.

Theorem 2.3. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a lower-triangular full-rank query matrix, and let $\mathbf{A} = \mathbf{B}\mathbf{C}$ be any factorization with the following property: for any two neighboring streams of matrices $\mathbf{X}, \tilde{\mathbf{X}} \in \mathbb{R}^{n\times d}$, we have $\|\mathbf{C}(\mathbf{X} - \tilde{\mathbf{X}})\|_F \leq \kappa$. Let $\mathbf{Z} \sim \mathcal{N}(0, \kappa^2\sigma^2)^{n\times d}$ with σ large enough so that $\mathcal{M}(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{Z} = \mathbf{B}(\mathbf{C}\mathbf{X} + \mathbf{Z})$ satisfies (ε, δ)-DP (or ρ-zCDP or μ-Gaussian DP) in the nonadaptive continual release model. Then, $\mathcal{M}$ satisfies the same DP guarantee (with the same parameters) even when the rows of the input are chosen adaptively.*

3 Optimizing matrix mechanisms for general participations

In the single-epoch setting, Denisov et al. (2022) derive expressions and present algorithms which empirically outperform previous methods for optimizing the matrix mechanism. In this section, we show that the methods they present for computing optimal factorizations can be extended to the generalized participation patterns under consideration here.

Following Denisov et al. (2022), we begin by defining the loss of interest for the mechanism defined in Eq. (1). Given a set of deltas $\mathfrak{D}$, assume that we may represent $\mathfrak{D} = \mathrm{conv}(\mathcal{D})$ for some finite set $\mathcal{D}$—as we have seen, this is the case, e.g., in (k, b) participation. Then the loss which corresponds to total squared error of a factorization, at a fixed privacy level, may be expressed as:

$$\mathcal{L}(\mathbf{B}, \mathbf{C}) = \mathrm{sens}^2_{\mathfrak{D}}(\mathbf{C})\, \|\mathbf{B}\|_F^2 \qquad \text{where} \qquad \mathrm{sens}_{\mathfrak{D}}(\mathbf{C}) = \sup_{\mathbf{u}\in\mathfrak{D}} \|\mathbf{C}\mathbf{u}\|_2^2 = \sup_{\mathbf{u}\in\mathcal{D}} \|\mathbf{C}\mathbf{u}\|_2^2. \tag{3}$$

Observing that $\forall\alpha$ the mechanism $\mathbf{A} = \left(\alpha\mathbf{B}\right)\left(\frac{1}{\alpha}\mathbf{C}\right)$ has identical loss, we conclude that we may consider the constrained version of the problem of minimizing this loss where $\mathrm{sens}_{\mathfrak{D}}(\mathbf{C}) \leq 1$. Since for any $\mathbf{C}$, $\mathbf{B} = \mathbf{A}\mathbf{C}^\dagger$ produces the minimum-Frobeneous norm $\mathbf{B}$-matrix, it is sufficient to solve:

$$\min_{\mathbf{C}} \mathcal{L}\left(\mathbf{A}\mathbf{C}^\dagger, \mathbf{C}\right) = \min_{\mathbf{C}:\mathrm{sens}^2_{\mathfrak{D}}(\mathbf{C})=1} \left\|\mathbf{A}\mathbf{C}^\dagger\right\|_F^2. \tag{4}$$

[3]The set of nonnegative $\mathbf{X}$ was found to contain optimal or near-optimal solutions for many of the mechanisms we optimally factorize and use below, as certified by measuring the optimality gap via the dual problem.

With the change of variables $\mathbf{X} = \mathbf{C}^\top \mathbf{C}$, equivalently:

$$\min_{\mathbf{X} \text{is PD}, \widehat{\text{sens}}_{\mathfrak{D}}^2(\mathbf{X})=1} \operatorname{tr}(\mathbf{A}^\top \mathbf{A} \mathbf{X}^{-1}) \quad \text{where} \quad \widehat{\text{sens}}_{\mathfrak{D}}^2(\mathbf{X}) = \sup_{\mathbf{u} \in \mathcal{D}} \mathbf{u}^\top \mathbf{X} \mathbf{u}. \tag{5}$$

Theorem 3.1. *Let a finite $\mathcal{D} = \{\mathbf{u}_i\}_{i=1}^k$ be given, and assume that the vectors $\{\mathbf{u}_i\}_{i=1}^k$ span $\mathbb{R}^n$. Assume that $\mathbf{A}$ is full-rank, and for $\mathbf{v} \in \mathbb{R}^k$ define $\mathbf{H}_\mathbf{v} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \operatorname{diag}(\mathbf{v})^{1/2}$, $\quad \mathbf{U} = \mathbf{H}_\mathbf{v} \mathbf{H}_\mathbf{v}^\top$. Then, for Lagrange multipliers $\mathbf{v}$ such that the $\mathbf{U}$ is full-rank, the Lagrange dual function g for the problem Eq.* (5) *can be expressed in closed form in terms of the Lagrange multipliers:*

$$g(\mathbf{v}) := \inf_{\mathbf{X} \text{ is PD}} L(\mathbf{X}, \mathbf{v}) = 2 \operatorname{tr}\left(\left(\mathbf{U}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{U}^{\frac{1}{2}} \right)^{\frac{1}{2}} \right) - \sum_{\mathbf{u} \in \mathcal{D}} \mathbf{v}_\mathbf{u} \tag{6}$$

Remark. The restriction that $\mathbf{v}$ yields a full-rank $\mathbf{U}$ serves to restrict to cases where the Lagrangian has a finite, positive-definite minimizer in the primal variable; if the vectors $\{\mathbf{u}\}$ span $\mathbb{R}^n$, the problem Eq. (5) has a finite minimizer by Lemma G.1. Any setting of the dual variables $\mathbf{v}$ corresponding to this minimizer is contained in a neighborhood uniformly satisfying this full-rank property, and so it is valid to differentiate our expression for g with respect to such $\mathbf{v}$ (as we will do in Appendix G.1).

Corollary 3.1. *In the same setup as Theorem 3.1, the gradient of the dual function g is: $\frac{\partial g}{\partial \mathbf{v}_i} = \mathbf{u}_i^\top \mathbf{U}^{-\frac{1}{2}} \left(\mathbf{U}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{U}^{\frac{1}{2}} \right)^{\frac{1}{2}} \mathbf{U}^{-\frac{1}{2}} \mathbf{u}_i - 1$. Moreover, a maximizer of the dual $\mathbf{v}^\star$ must satisfy:*

$$\mathbf{v}^\star = \operatorname{diagpart}\left(\left(\mathbf{H}_{\mathbf{v}^\star}^\top \mathbf{A}^\top \mathbf{A} \mathbf{H}_{\mathbf{v}^\star} \right)^{\frac{1}{2}} \right). \tag{7}$$

The optimal value of the problem defined in Eq. (5) *is* $\operatorname{tr}(\mathbf{v}^\star)$.

Remark. In the single-participation case of Denisov et al. (2022), $\mathbf{H} = \mathbf{I}$ and $\mathbf{H}_\mathbf{v} = \operatorname{diag}(\mathbf{v})^{\frac{1}{2}}$, and the expression Eq. (7) recovers the fixed point expression of that paper's Theorem 3.2. Our Corollary 3.1 implies that the optimization methods presented in Denisov et al. (2022) may be applied, with suitable translation, to our setting; we use these methods to generate the optimal matrices studied empirically in Section 5.

4 Fast-Fourier-Transform-based DP-Prefix Sum Estimation

The best known methods for computing optimal factorizations scale as at least $\mathcal{O}(n^3)$ in many cases of interest (e.g., single-participation, Yuan et al. (2016); Denisov et al. (2022)); generating noise at runtime for these optimal dense mechanisms is of complexity $\mathcal{O}(n^2)$. When n is large, as can be the case in ML, this becomes intractable. Thus, in this section we now focus on reducing computation at a (small) decrease in the achievable privacy-utility tradeoff. To do this, a prime candidate is the Discrete Fourier Transform (DFT) because there are known algorithms both for nearly-optimal private convolutions (Fawaz et al., 2013) which are intimately related to the DFT, and for efficiently calculating the DFT by using the Fast Fourier Transform (FFT) (Nussbaumer, 1981).

Unlike the optimal matrix mechanisms, which may be computationally prohibitive for large number of steps n, the computational scaling of the FFT-based mechanism is much more promising. Indeed, this algorithm requires no optimization costs and $O(n \log n)$ time to generate the required noise. However, the FFT algorithm requires $O(n)$ space as all noises corresponding to the vector $\mathbf{x}$ must be pre-generated. Our analysis will focus on proving rigorous DP guarantees for the FFT algorithm and showing that these lead to near-optimal privacy-utility tradeoffs in the single-epoch setting. Our exposition is primarily based on Fawaz et al. (2013). We provide two improvements over prior work: 1) extending the result to the multi-epoch and multi-dimensional setting and 2) providing explicit non-asymptotic analysis of the algorithm's utility.

Consider that the DP release of prefix sums corresponds to the matrix mechanism in Equation 1 with $\mathbf{A} \in \{0,1\}^{n \times n}$ lower triangular and Toeplitz. In this section, we perform our analysis in the Fourier domain. To still release $\mathbf{A}\mathbf{x}$, we must define a circulant matrix $\mathbf{A}_{\mathsf{circ}} \in \mathbb{R}^{2n \times 2n}$ with a corresponding input vector $\mathbf{x}_{\mathsf{ext}} = \mathsf{concat}(\mathbf{x}, \mathbf{0}_n), \mathbf{0}_n \in \{0\}^n$ so that the first n entries of $\mathbf{A}_{\mathsf{circ}} \mathbf{x}_{\mathsf{ext}}$ are equal to $\mathbf{A}\mathbf{x}$. We do so in Appendix H.1. Thus, we study the DP release of $\mathbf{A}_{\mathsf{circ}} \mathbf{x}_{\mathsf{ext}}$.

Algorithm 1 DP-Prefix Sum Computation via FFT

Inputs: Data vector $\mathbf{x} \in \mathbb{R}^n$ (with each $\|x_i\| \leq \zeta$) and zCDP parameter ρ.

1: $\mathbf{v}_{\mathsf{DFT}} \in \mathbb{C}^{2n} \leftarrow$ the DFT of $\mathbf{v}$ (defined in Equation 28). Let $\mathbf{v}_{\mathsf{DFT}}^{(:n)}$ be the first n coordinates.

2: $\mathbf{F} \leftarrow$ DFT matrix in $2n$-dimensions, where k-th row of $\mathbf{F}$ is given by $\mathbf{F}[k,:] = \frac{1}{\sqrt{2n}}\left[\exp\left(-\frac{j2\pi ka}{2n}\right) : a \in \{0,\ldots,2n-1\}\right]$. $\mathbf{F}^*$ $\leftarrow$ Hermitian of $\mathbf{F}$.

3: $\Sigma \leftarrow$ a diagonal matrix with the diagonal being $\mathbf{v}_{\mathsf{DFT}}$.

4: $\mathbf{w} \leftarrow$ standard complex Normal in $2n$-dimensions.

5: $(\mathbf{s}, \widetilde{\mathbf{z}}) \leftarrow \left(\left[x_0, x_0 + x_1, \ldots, \sum_{a=0}^{n-1} x_a\right], \sqrt{\frac{\kappa^2 \|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\rho}}\left(\mathbf{F}^* \Sigma^{1/2} \cdot \mathbf{w}\right)\right)$.

Output $\mathbf{s} + \widetilde{\mathbf{z}}.\mathtt{real}[0,\ldots,n-1]$.

The single participation setting With the above problem setup, we define $\mathbf{A}_{\mathsf{circ}} = \mathbf{F}^* \Sigma \mathbf{F}$ for diagonal Σ, via Theorem H.1 by Gray (2006), in Appendix H.1. $\mathbf{A}_{\mathsf{circ}}$ can then be factorized as $\mathbf{A}_{\mathsf{circ}} = \mathbf{B}_{\mathsf{circ}}\mathbf{C}_{\mathsf{circ}}$ where $\mathbf{B}_{\mathsf{circ}} = \mathbf{F}^* \Sigma^{1/2}$ and $\mathbf{C}_{\mathsf{circ}} = \Sigma^{1/2}\mathbf{F}$. This allows us to show that Algorithm 1 is ρ-zCDP, and also show that Algorithm 1 is nearly optimal for any matrix factorization based mechanisms. We defer the privacy and the utility guarantees (along with a statement of optimality) to Appendix H.

Extending to the multi-dimensional and multi-participation setting First, note that if the data points $\mathbf{x}_i$'s are a d-dimensional vector instead of a scalar, then Corollary 2.1 allows us to translate any privacy guarantees we prove for scalar contributions.

Theorem 4.1 proves a simple zCDP guarantee with *any* participation pattern having at most k participations, though it is possible to derive a tighter guarantee when the participation pattern is known in advance; we show this in Theorem 4.2. All the proofs are deferred to Appendix H.

Theorem 4.1. *Under k participation, Algorithm 1 satisfies $(k^2\rho)$-zCDP.*

Theorem 4.2. *For any participation pattern Π with $|\pi| \leq k$, $\forall \pi \in \Pi$, let $\lambda = \max_{\pi \in \Pi} \left\|(\sqrt{\Sigma}\mathbf{F})[:,\pi]\right\|_2$, where $\mathbf{M}[:,\pi]$ corresponds to selecting the columns of the matrix $\mathbf{M}$ indexed by π, and let $\mu = \sqrt{\frac{\|\mathbf{v}_{\mathsf{DFT}}\|_1}{2n}}$. Then, Algorithm 1 satisfies $\left(\frac{k\lambda^2}{\mu^2} \cdot \rho\right)$-zCDP.*

5 Empirical Evaluation

We empirically compare three main mechanism classes: tree-based mechanisms (Honaker, 2015), including 'tree-completion', from Kairouz et al. (2021); our FFT mechanism; and our optimal factorizations. We vary these mechanisms along two axes. First, the **factorization's epoch structure** impacts the optimal factorization[4] by altering the geometry of the optimization problem. Second, the **number of encoder 'stamps'** (s), which allows us to extend any factorization to handle more iterations. To do so, we repeatedly 'stamp' its encoder, $\mathbf{C}$ along the diagonal by taking a Kronecker product between it and an identity matrix $\mathbf{I}$; the decoder may be taken to be the optimal one, $\mathbf{B} = \mathbf{A}\left(\mathbf{C} \otimes \mathbf{I}\right)^\dagger$.[5] The resulting mechanism computes sn-dimensional linear DP queries. A mechanism which uses $s > 1$ will be distinguished by a "$\times s$" appended to its name.

The manner in which baselines from the extant literature map to this setting can be found in Appendix C.1. Since the matrix mechanism reduces privacy cost of training to that of the release of a single Gaussian mechanism, accounting in our case becomes quite simple; see Appendix C.2.

[4]We compute optimal factorizations for the (k, b)-participation setting. Note that an encoding matrix factorized for one value of k can be applied for another, by simply computing its new sensitivity (due to our Section 2), though this will alter its privacy-utility tradeoffs.

[5]This construction is similar to the notion of 'restarts' in Kairouz et al. (2021). One subtle difference between these two notions is the construction of the decoding matrix $\mathbf{B}$, and associated issues of generality; see the discussion in Appendix C.3 for further details.

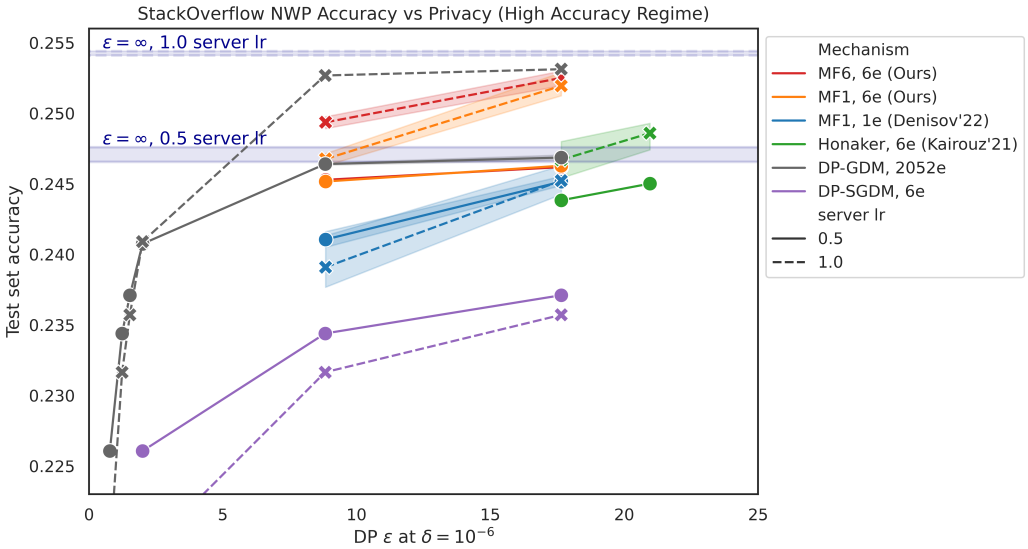

Figure 2: **Our MF6,6e achieves within** 2% **relative difference in performance from the non-private baseline at** $\varepsilon \geq 8.84, \delta = 10^{-6}$. Our **MF1,6e**, though worse, outperforms all baselines from the literature. Select runs were replicated 3 times, with bootstrap 95% intervals shown. **DP-GDM,2052e** is the extreme case of every-round participation ($342\times$ more computationally expensive than 6 epochs of training), and hence generally infeasible.

The mechanisms that we evaluate may be compared directly in a given multi-pass loss domain as defined by Eq. (3); the manner in which certain mechanisms distribute variance through time, and their computed losses across different factorization geometries, can be found in Appendix C.4.

5.1 Example-level DP for an image classification task.

We train image classification models on CIFAR10 (Krizhevsky, 2009) which has become a de facto standard for comparing DP ML algorithms—sufficiently easy for existing DP algorithms to achieve nontrivial accuracy, but not so simple so as to be unable to differentiate approaches. Details on our full setup are in Appendix D; generally, they match those of Kairouz et al. (2021). Notably, and in contrast to Section 5.2, in this section we only compare factorizations of the prefix-sum matrix; we do not incorporate momentum or cooldown directly into the mechanisms.

Main results (Figure 1) First, we see that optimal factorization for the target $(k, b) = (20, 100)$ training setup's geometry outperforms all other mechanisms across all privacy levels. Indeed, even as low as $\varepsilon = 2$ we find that we outperform DPSGD with amplification by ≈ 5 absolute percentage points. To the best of our knowledge, this is the first empirical demonstration of an algorithm that significantly outperforms DPSGD with amplification on this task in the high-privacy regime. The FFT mechanism outperforms all baselines from the literature, again well toward the high-privacy regime at $\varepsilon \approx 2$. Though we find that it does have a substantially worse privacy-utility tradeoff compared with our multi-epoch optimal matrices, this mechanism shows promise for outperforming prior work when n grows too large for generating optimal matrix factorizations.

5.2 User-level DP for a next word prediction task

We next consider a close simulation of real-world DP applications (McMahan & Thakurta, 2022): user-level DP on language models. There is much history in the DP language modelling literature, which we briefly describe in Appendix E.1. Here, we use the standard benchmark: StackOverflow next-word prediction (Reddi et al., 2020). Our experimental setup fixes the same model and hyper-parameters as Kairouz et al. (2021) and Denisov et al. (2022) except notable changes below and a full setup in Appendix E.2.

Notable changes from prior work We use a much higher 1000 clients per round (≈ 100 in prior work)—enabled in part by our multi-epoch factorizations. We also zero out large updates with ℓ_2 norm greater than 100 (rather than scaling down to our clipping norm of 1) as we found this improved the stability of noisy training. This may have enabled more consistent success of higher server learning rates > 1.0 used in prior work. These lead to additional improvements.

Simulating 1000 clients per round requires significant computational resources; rather than fully exploring the privacy-utility tradeoff, we focus on the regime likely to be of the most practical interest: high accuracy (compared to a non-private baseline). We conducted initial simulations which verified two facts in the multi-epoch large-batch regime for StackOverflow which were previously observed by Denisov et al. (2022). First, that learning rate cooldown (linearly decreasing the server learning rate from $1\times$ to $0.05\times$ over the final 512 rounds) generally offers a small improvement over a constant learning rate (more so in the higher-privacy regime). Second, finding an optimal factorization of the target matrix that includes momentum and the learning-rate cooldown schedule (rather than factoring the prefix sum matrix and applying momentum and the learning-rate schedule as post processing) offers a small but consistent benefit. See Appendix E.2 for details. Thus, for our primary investigation we fix these preferable design choices and compare the following algorithms.

Algorithms All algorithms train for 6 epochs unless otherwise noted. **Honaker,6e** is the DP-FTRL algorithm of Kairouz et al. (2021), trained for 6 epochs with 1000 clients per round and 2048 rounds (stopping at a power of 2). **MF1,1e** (Denisov et al., 2022) is the state-of-the-art for single-epoch training, using 167 clients per round and 2048 rounds (to maintain 1 epoch). **MF1,6e** uses our Theorem 2.1 to take the single-participation-optimized matrix of the previous approach, and compute the sensitivity under multiple participations, allowing us to train for 6 epochs (1000 clients per round for 2048 rounds). **MF6,6e** is our approach directly optimizing the matrix factorization with a constraint on sensitivity for 6 epochs, and trained for a full 6 epochs (1000 clients per round for 2052 rounds). **DP-SGDM,6e** is the `DPFedAvg` algorithm of (McMahan et al., 2017), similarly trained for 6 epochs on 1000 clients per round and 2052 rounds, and accounted with Poisson sampling–an incorrect, though standard, accounting computation, as noted in Section 1. **DP-GDM,2052e** is full-batch gradient descent for 2052 steps corresponds to 2052 epochs, or $342\times$ the computation cost of our 6 epoch runs. We compute the exact privacy cost for this approach (assuming Poisson sampling), and estimate the accuracy from experiments with 1000 clients per round for computational efficiency, following the methodology of Kairouz et al. (2021). This is essentially an upper bound on the best privacy-accuracy tradeoffs with unlimited computational resources.

Main results (Figure 2) We find that our **MF6,6e**, is the best private result at 25.25% accuracy and $(17.7, 10^{-6})$-DP This exceeds the non-private baseline of 25.2% accuracy reported by Kairouz et al. (2021), is within the margins of small hyperparameter tuning differences of our improved non-private baselines (25.43%) and private full-batch gradient descent (25.31%). At $(8.84, 10^{-6})$-DP **MF6,6e** achieves 24.94% accuracy, substantially improving over the previous state-of-the-art at this privacy level given by **MF1,1e** at 24.11% accuracy, and considerably improves on both the accuracy and privacy of **Honaker,6e** (24.86% accuracy at $(21.0, 10^{-6})$-DP. In fact, we achieve better accuracy at $\varepsilon = 8.84$ than prior methods achieve at $\varepsilon = 17.7$. **DP-SGDM,6e** is substantially outperformed by all matrix-factorization-based methods.

6 Discussion and Conclusion

Our work significantly improves the privacy-utility tradeoffs in DP ML while maintaining practically feasible computation costs. Indeed, our work enables practitioners to outperform the state-of-the-art (and, DPSGD with amplification) by ≈ 5 percentage points across many privacy levels—as high as $\varepsilon \approx 2$—with assumptions that can be implemented in practice. We remark that we compare our mechanisms with DPSGD on a level-ground using well-performing but not state-of-the-art models and training protocols—e.g., very large models, augmentations prior to clipping (De et al., 2022), public data usage, and large batch sizes can all aid training. Many, if not all, of these techniques are applicable to our setting and can thus be used with our mechanisms to realize additional absolute performance gains, again, likely beyond that of the performances achieved by DPSGD.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS)*, pp. 464–473, 2014.

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

Lei Cao, Darian McLaren, and Sarah Plosker. Centrosymmetric stochastic matrices. *Linear and Multilinear Algebra*, 70(3):449–464, 2022.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pp. 267–284, USA, 2019. USENIX Association. ISBN 9781939133069.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. on Information Systems Security*, 14(3):26:1–26:24, November 2011.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Sergey Denisov, Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for sgd via optimal private linear operators on adaptive streams, 2022. URL https://arxiv.org/abs/2202.08312.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proc. of the Forty-Second ACM Symp. on Theory of Computing (STOC'10)*, pp. 715–724, 2010.

Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. *The Power of Factorization Mechanisms in Local and Central Differential Privacy*, pp. 425–438. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450369794. URL https://doi.org/10.1145/3357713.3384297.

Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 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.

Nadia Fawaz, S Muthukrishnan, and Aleksandar Nikolov. Nearly optimal private convolution. In *European Symposium on Algorithms*, pp. 445–456. Springer, 2013.

Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 954–964. IEEE, 2022.

Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained complexity of differentially private continual observation, 2022. URL https://arxiv.org/abs/2202.11205.

Robert M Gray. Toeplitz and circulant matrices: A review. 2006.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, 2010.

Monika Henzinger, Jalaj Upadhyay, and Sarvagya Upadhyay. Almost tight error bounds on differentially private continual counting. *Personal communication*, 2022.

James Honaker. Efficient use of differentially private binary trees. *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2015.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *ICML*, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

Chao Li, Gerome Miklau, Michael Hay, Andrew Mcgregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB Journal*, 24: 757–781, 2015.

Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. Optimizing error of high-dimensional statistical queries under differential privacy. *Proc. VLDB Endow.*, 11(10): 1206–1219, jun 2018. ISSN 2150-8097. doi: 10.14778/3231751.3231769. URL https://doi.org/10.14778/3231751.3231769.

H. Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees. Google AI Blog, 2022. URL https://ai.googleblog.com/2022/02/federated-learning-with-formal.html.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pp. 80–111. Springer, 1981.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization. *CoRR*, abs/2003.00295, 2020. URL https://arxiv.org/abs/2003.00295.

Congzheng Song and Vitaly Shmatikov. 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.

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

Joel Tropp. *Topics in Sparse Approximation*. PhD thesis, The University of Texas at Austin, 2004.

Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. corr abs/1808.00087 (2018). *arXiv preprint arXiv:1808.00087*, 2018.

Ganzhao Yuan, Yin Yang, Zhenjie Zhang, and Zhifeng Hao. Optimal linear aggregate query processing under approximate differential privacy. *CoRR*, abs/1602.04302, 2016. URL http://arxiv.org/abs/1602.04302.

Yuqing Zhu and Yu-Xiang Wang. Poisson subsampled rényi differential privacy. In *International Conference on Machine Learning*, pp. 7634–7642. PMLR, 2019.

A SUMMARY OF NOTATION

n	Number of steps of the streaming linear query (SGD steps or FL rounds)
d	Dimension of per-step user contributions.
$\mathbf{x}_i \in \mathbb{R}$ or $\mathbb{R}^d$	Sum of per-example gradients (or per-user model updates) on step i.
ζ	Clipping norm that limits the size of per-example contributions to $\mathbf{x}_i$.
π	Set of steps that an example could participation in.
Π	Participation pattern, set of sets of steps (set of all π) an example could participate in.
$\mathfrak{D}$	$= \{\mathbf{x} - \tilde{\mathbf{x}} \mid (\mathbf{x}, \tilde{\mathbf{x}}) \in \mathbf{N}\}$, the set of deltas between neighboring $\mathbf{x}, \tilde{\mathbf{x}}$.
$\mathcal{D}$	Corners of $\mathfrak{D}$ when assumed to be a polytope, $\mathfrak{D} = \mathrm{conv}(\mathcal{D})$.
(k, b)-participation	At most k participations, separated by exactly b.
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Lower-triangular linear query matrix to be factorized as $\mathbf{A} = \mathbf{BC}$.
$\mathbf{T} \in \mathbb{R}^{n \times n}$	$\mathbf{T} := \mathbf{A}^\top \mathbf{A}$ for convenience.
$\lambda_{\min}(\mathbf{A})$, $\lambda_{\max}(\mathbf{A})$.	Smallest and largest eigenvalues of real matrix $\mathbf{A}$.
$\mathbf{A}^*$	Conjugate transpose of $\mathbf{A}$.
$\mathbf{X}^\star$	A matrix $\mathbf{X}$ that is "optimal" in a context-dependent sense.
$\mathbf{A}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{A}$.
$\mathbf{A}_{[i,j]}$	The $(i, j)^{\text{th}}$ entry of matrix $\mathbf{A}$.
$\mathbf{A}_{[i,:]}$ and $\mathbf{A}_{[:,j]}$	The ith row and j^{th} column.
s	Number of encoder $\mathbf{C}$ replications (stamps) into a block-diagonal matrix.
$\mathrm{conv}\,(S)$	Convex hull of the set S.
$[n]$	$= \{1, \ldots, n\}$

B GENERALIZED SENSITIVITY AS AN OPERATOR NORM

Eq. (2) shows that this generalized notion of sensitivity can be viewed directly as a particular operator norm. To see this, view $\mathbf{C} : \mathbf{V}_1 \to \mathbf{V}_2$ as a linear operator from vector space $\mathbf{V}_1$ to $\mathbf{V}_2$. Then with $\|\cdot\|_{(1)}$ the vector norm on $\mathbf{V}_1$ and similarly for $\mathbf{V}_2$, an operator norm is defined as

$$\|\mathbf{C}\|_{(1),(2)} = \max_{u:\|u\|_{(1)} \leq 1} \|\mathbf{Cu}\|_{(2)}.$$

Because we use the Gaussian mechanism and thus are interested in the ℓ_2 sensitivity, $\|\cdot\|_{(2)} = \|\cdot\|_2$, and we define the norm

$$\|\mathbf{u}\|_{(1)} = \|\mathbf{u}\|_{\mathfrak{D}} := \inf\left\{r > 0 : \frac{\mathbf{u}}{r} \in \mathfrak{D}\right\},$$

the vector norm induced by $\mathfrak{D}$ (the fact that $\mathfrak{D}$ is a closed, convex, symmetric set ensures this is a norm). Note $\mathbf{u} \in \mathfrak{D} \;\Leftrightarrow\; \|\mathbf{u}\|_{\mathfrak{D}} \leq 1$. Thus, we have

$$\mathrm{sens}_{\mathfrak{D}}(\mathbf{C}) = \|\mathbf{C}\|_{\mathfrak{D},2}. \tag{8}$$

C MECHANISMS UNDER CONSIDERATION: BASELINES, SUBTLETIES, AND LOSSES.

C.1 BASELINES

Kairouz et al. (2021) and Denisov et al. (2022) both present approaches for ML model training which can be understood as instances of the matrix mechanism–the former grounded in the binary-tree mechanism as refined by Honaker (2015), and the latter explicitly optimizing a factorization for the single-pass geometry. These two works yield two natural baselines:

- Kairouz et al. (2021) explores 'tree restarts' (generalized as 'stamps' s in Section 5) and the so-called 'tree-completion trick' for the Honaker estimator-from-below variant of the binary tree method for computing differentially private prefix sums; the matrix-factorization perspective on this estimator allows us to implement slightly optimized versions of these methods[6].
- Denisov et al. (2022) computes optimal factorizations of various optimization-related matrices, though only for the geometry of a single epoch. The results of Section 2 allow us to compute the sensitivity of the encoder matrices in our setting, and therefore apply these matrices factorized for the single-epoch geometry directly to the multi-epoch setting. In the terminology of Section 5, these mechanisms are restricted to factorizations for $k = 1$ geometry.

These two papers can be combined in other ways as well; e.g., the results of Denisov et al. (2022) show that the 'fully efficient estimator' of Honaker (2015) may be used as a drop-in replacement for the estimator from below in Kairouz et al. (2021). We focus on the two mechanisms specified above as the natural baselines for the present work.

C.2 Privacy Accounting

The matrix mechanism Eq. (1) conceptually adds isotropic Gaussian noise in some encoded space. In our case, we encode a matrix of gradients (clipped to ℓ_2 norm κ) computed over the course of training, denoted by $\mathbf{G}$, with the matrix $\mathbf{C}$, and add Gaussian noise to each entry in the matrix $\mathbf{CG}$. Under the assumption that the matrix factorization has been appropriately scaled so $\mathbf{C}$ has sensitivity 1, this Gaussian noise will have standard deviation $\sigma = \kappa z$ in each coordinate, where z is the 'noise multiplier' parameter determining the privacy level of the mechanism.

Privacy costs are computed as a single application of the Gaussian mechanism to $\mathbf{GC}$ using the `PLDAccountant` provided by the Google DP Library[7]. We also use this accountant to analyze DP-(S)GD baselines (which require more complex accounting), yielding a small improvements in ε over the Renyi-DP accounting used in prior works.

C.3 Repeated mechanisms in the matrix-factorization setting

Kairouz et al. (2021) explored 'restarting' their binary-tree based prefix sum estimation mechanism, treating the number of restarts used as a hyperparameter. The notion of 'restarts' as presented there has several potential translations into the matrix-factorization setting, though some are not applicable to the factorization of all linear operators $\mathbf{A}$. Following Denisov et al. (2022), we consider only operators $\mathbf{A}$ representable by lower-triangular matrices.

- For linear operators $\mathbf{A}$ with constant columns below the diagonal, one may use ordinary mechanism composition to express the notion of 'restarting'. Notably, the prefix-sum matrix has this constant-columns property; it is required to use composition, since one must assume that the output of an earlier instantiation of the mechanism is fixed, and used as an input to a new version. It is not immediately obvious that this notion of restarts corresponds to an instantiation of the matrix mechanism Eq. (1).
- As an instantiation of the matrix mechanism, a similar notion of restarts may be expressed by replicating an encoder along the diagonal of a larger matrix (taking a Kronecker product with $\mathbf{I}$ as discussed in Section 5), and constraining the decoding matrix to repeatedly use the final row corresponding to a block j for any blocks $i > j$. We conjecture that this construction has identical privacy properties to using compositions; notably, it is restricted to linear operators $\mathbf{A}$ satisfying the same structural constant-columns assumption. It is this construction which we use for the DP-FTRL-Honaker baseline. Interestingly, with our capacity to measure mechanisms at a single shot in the multi-epoch setting, we do indeed see that these 'restarted' mechanisms have lower total loss than their non-restarted full-tree

[6]For example, the matrix perspective makes the action of tree-completion more clear: we simply replace the final row of the decoder matrix $\mathbf{B}$, at an index which in general will not be a power of two, with the coefficients of the next-highest power of two. No changes to accounting are required.

[7]`https://github.com/google/differential-privacy`

counterparts in the 20-epoch, 100 steps / epoch setting (see Table 1); training performance of these 'restarted' mechanisms on CIFAR10 can be found in Fig. 5.

- The linear-algebraic perspective on the matrix factorization problem, on the other hand, allows us to construct 'repeated' mechanisms for *any* linear operator $\mathbf{A}$. This construction was presented in Section 5: construct a repeated encoder with a Kronecker product in the manner described above, but construct the optimal decoder by matrix pseudoinversion (IE, let $\mathbf{B} = \mathbf{A}\left(\mathbf{C} \otimes \mathbf{I}\right)^{\dagger}$), which defines a legitimate factorization of any $\mathbf{A}$ requiring only suitable non-degeneracy assumptions on $\mathbf{C}$. This construction of a 'repeated' mechanism may be *quite different* from those above; for example, if $\mathbf{C}$ is a matrix representation of the binary-tree encoder and $\mathbf{I} = 1$, this construction defines the *full*, rather than online, Honaker decoder (Honaker, 2015). The validity of applying such a decoder in the adaptive streaming setting was only shown quite recently by Denisov et al. (2022); still, we consider this construction to be the more natural notion of repeated matrix mechanism from the linear-algebraic perspective, and it is this construction we use for all sweeps over k which do not correspond to the DPFTRL-Honaker baseline.

C.4 Factorization losses and per-iterate variance

As a first measure of the privacy-utility tradeoff, we compare the losses of each mechanism from Equation 3 for factorizations of the matrices under consideration.

We compare measured losses of several factorizations of the prefix-sum matrix under the $(k, b) = (20, 100)$ geometry in Table 1. We plot the per-iterate variance of the mechanisms in Fig. 1, along with several variations, in Fig. 3. In Fig. 4, we plot the per-iterate variance distribution of factorizations of the momentum and cooldown matrix (described in Section 5) in the $(k, b) = (6, 252)$ geometry at a fixed variance level, and various privacies.

Figs. 3 and 4 both demonstrate the effect of the (k, b) geometry on the optimization problem. Particularly interesting to consider are the optimally-factorized matrices; in both cases, the epoch structure is clearly visible in the manner in which the mechanisms distribute variance. We see also the effect of 'stamps' s in the variance distribution, effectively a proxy for the epoch structure directly accounted for by the optimal mechanisms.

Mechanism	$(k, b) = (20, 100)$ loss
Honaker($n = 2000$)	5.7e6
Honaker($n = 1000$) $\times$ 2	3.3e6
Honaker($n = 400$) $\times$ 5	2.1e6
Honaker($n = 200$) $\times$ 10	**2.0e6**
Honaker($n = 100$) $\times$ 20	2.1e6
MF($k = 1, n = 100$) $\times$ 20	2.5e6
MF($k = 1, n = 2000$)	1.6e6
MF($k = 5, n = 500$) $\times$ 4	1.2e6
MF($k = 10, n = 100$) $\times$ 2	8.8e5
MF($k = 20, n = 2000$)	**6.5e5**
FFT($n = 2000$)	2.2e6
FFT($n = 1000$) $\times$ 2	1.5e6
FFT($n = 400$) $\times$ 5	**1.1e6**
FFT($n = 200$) $\times$ 10	1.2e6
FFT($n = 100$) $\times$ 20	1.7e6

Table 1: Loss for various prefix-sum factorizations, computed via Eq. (3), under multiple-participation geometry for 20 epochs with 100 steps per epoch. Lowest-loss mechanism in each class bolded. By evaluating the dual problem (Section 3), 6.53e5 represents a lower bound on the optimal loss; the optimal matrix factorization is within 0.2% of this optimal value.

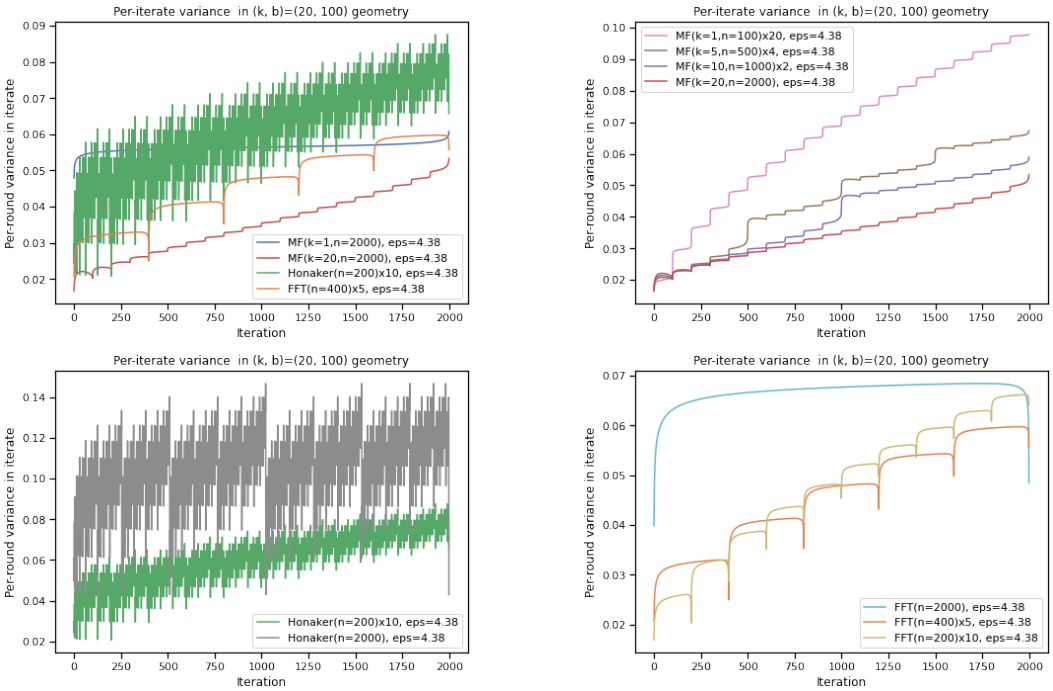

Figure 3: Per-iterate variance for prefix-sum factorizations in $(k, b) = (20, 100)$ geometry. All mechanisms above yield the same privacy ($(\varepsilon, \delta) = (4.38, 10^{-5})$), but have different total variances (the integral of the curves above).

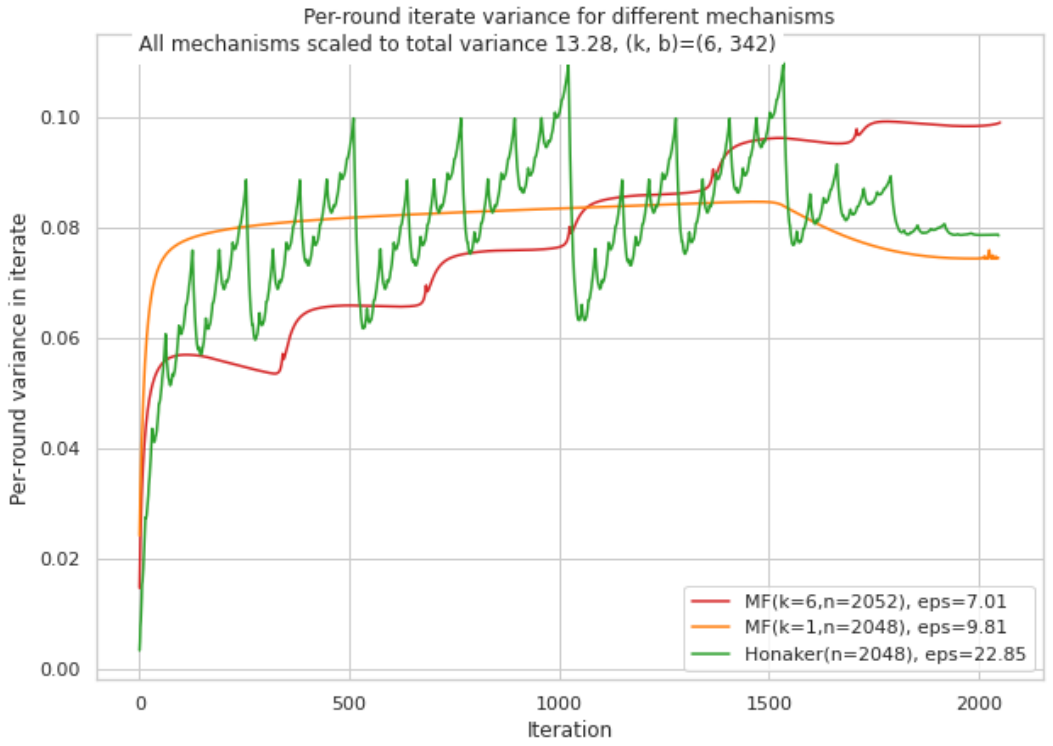

Figure 4: Per-iterate variance for momentum + cooldown matrix factorizaations in $(k, b) = (6, 252)$ geometry.

D DETAILS AND ADDITIONAL EXPERIMENTS FOR CIFAR10.

We train image-classification models using the CIFAR10 dataset as hosted in `tensorflow-datasets`, containing 50,000 training and 10,000 test examples. We evaluate and compute test accuracies on the entire test set, following the open-sourced code of Kairouz et al. (2021). We reuse the network architecture, dataset processing and initialization strategies presented in Kairouz et al. (2021); in particular, the architecture we use can be found in their Table 2 (b).

Optimization setup and hyperparameters We train all mechanisms for 20 epochs with batch size of 500, yielding 100 steps per epoch and 2000 total. After performing some small initial grid searches, we settled on using an SGD with momentum optimizer, with momentum value 0.95 and linear learning rate cooldown to $0.05\times$ the initial learning rate over the last 500 steps of training.

As mentioned in Section 5, for this 20-epoch training setup, we only compare factorizations of the prefix-sum matrix, and do not include any factorizations of matrices which incorporate momentum of learning rate cooldown directly in the mechanism itself (Denisov et al., 2022).

We sweep over the values $(1 \times 10^i, 2 \times 10^i, 5 \times 10^i)$ for i in $\{-2, -1\}$. For all mechanisms and noise levels, optimal values were in the interior of this sweep.

For Honaker and FFT-based factorizations, there is no known a-priori way to choose the optimal number of s for a given (k, b) setting. Therefore we treat the value s as a hyperparameter, and sweep across it, for $s \in \{1, 2, 5, 10, 20\}$. As can be seen in Table 1, the optimal s for both of these factorizations was in the interior of this sweep. As shown, e.g., in Fig. 5, the training-time performance of these mechanisms matched the expected order for computed loss. This value s represents an extra hyperarameter which must be set for the Honaker and FFT mechanisms; to the best of our knowledge, computing the loss for various instantiations of these mechanisms via Eq. (3) represents the only known method for setting this parameter other than simply training models.

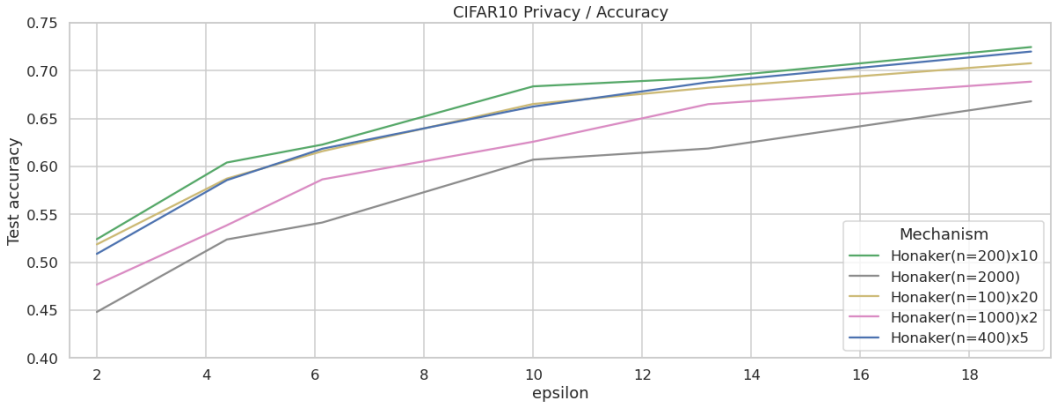

Figure 5: DPFTRL-Honaker baseline ablation with respect to number of 'stamps' s.

E ADDITIONAL STACKOVERFLOW DETAILS

E.1 PRIVACY AND LANGUAGE MODELLING

Language models trained on user data are an important real-world application of DP training, as these models can memorize their training data if appropriate mitigations are not applied (Carlini et al., 2019; Song & Shmatikov, 2019; Carlini et al., 2021). Since one user might contribute 1000s of tokens (training examples) to a dataset, it is particularly important to consider user-level guarantees (McMahan et al., 2017). Building on the approach of Kairouz et al. (2021), Google recently announced the first-ever launch of a language model trained on user data with a formal user-level DP guarantee ($\rho = 0.81$ zCDP), further demonstrating the importance of this application (McMahan & Thakurta, 2022).

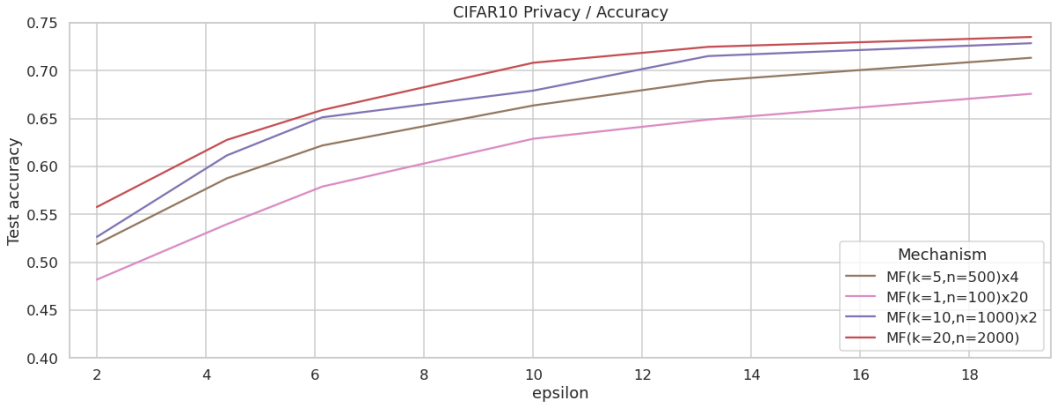

Figure 6: Ablation of prefix-sum factorizations, optimized for different number of epochs, and 'stamped' as appropriate. As expected, performance improves as the goemetry factorized approaches that used for training.

The StackOverflow next-word prediction task, introduced in (Reddi et al., 2020), has become a benchmark problem for DP training, and our experimental setup here fixes the same model and hyperpar

E.2 Additional Empirical Setup

All runs use server momentum 0.95 and a learning-rate cooldown schedule for the last 25 of rounds. Zeroing outlier updates and using 1000 clients/round (6 epoch runs) allows the use of the higher server learning rate with privacy **MF1,1e** replicates the result of single-epoch training from Denisov et al. (2022); note that with 167 clients/round and this mechanism, the higher learning rate does not appear to help. Fig. 7 gives preliminary experimental results which informed the main experiments used in the paper. Note that the y-axis range (Test set accuracy) is highly compressed, and so the primary point of comparison is on epsilons. For example, Denisov et al. (2022) shows that cross-run variation of 0.002 or more is typical.

The 6 horizontal lines give test-set accuracy for various non-private training mechanisms. The "Unnoised MF" runs correspond to the trainings used for privacy, but without any noise addition. In particular, these use momentum with learning rate cooldown; the other unnoised runs use a standard FL implementation with momentum but a fixed learning rate schedule; "cpr=167" corresponds to one epoch of training (167 clients/round), and "cpr=50" is 50 clients/rounds (only about 1/3 of an epoch). This last non-private baseline uses the best hyperparameters for FedAvgM from Reddi et al. (2020).

The two "Unnoised MF" runs with accuracies between 0.246 and 0.248 are functionally identical, and the line near 0.248 accuracy is the same except it does not use learning-rate cooldown. Thus, for the given learning rates, we see the higher-epsilon private runs are adding sufficiently small noise that the accuracy is essentially equivalent to unnoised baselines with the same hyperparameters. However, using larger learning rates can achieve accuracy over 25%, even with privacy as in the case of the MF-6-6 run, hence motivating the inclusion of larger learning rates in the main experiments.

The MF (Matrix Factorization) runs with "prefix" in the name correspond to computing an optimal factorization of the prefix-sum matrix (lower triangular matrix of ones) and then applying momentum (and possibly learning rate cooldown) as post-processing. The other MF runs directly factor the momentum or momentum+cooldown matrix.

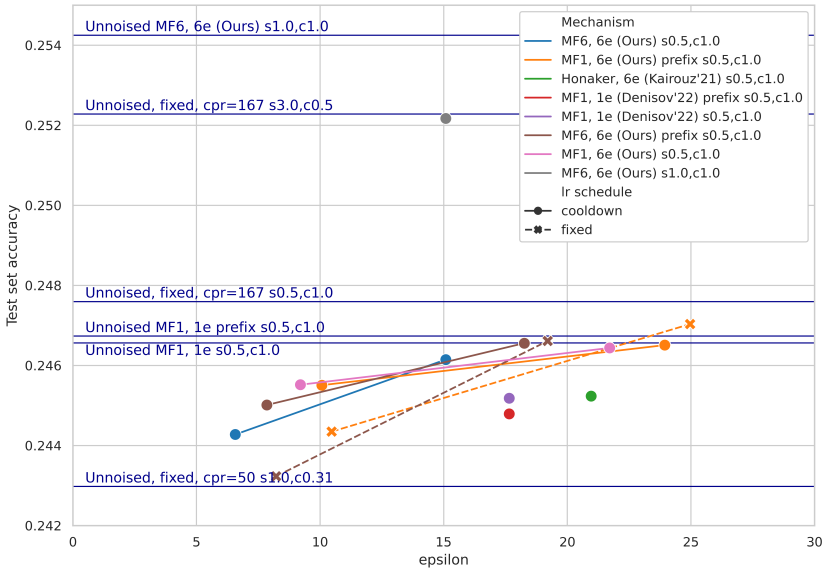

Figure 7: Preliminary experimental results.

F Analysis for Section 2

F.1 Proof of Theorem 2.1

Theorem 2.1 Restated. *Let* $\mathbf{C} \in \mathbb{R}^{n \times k}$ *which satisfies*

$$\|\mathbf{C}\mathbf{u}\|_2 \leq 1 \qquad \forall \mathbf{u} \in \{-1, 1\}^k,$$

and let $\mathbf{G} \in \mathbb{R}^{k \times d}$ *such that each row* $\mathbf{G}_{[i,:]}$ *for* $i \in [k]$ *satisfies* $\|\mathbf{G}_{[i,:]}\|_2 \leq 1$. *Then,*

$$\|\mathbf{C}\mathbf{G}\|_F \leq 1.$$

Proof. Consider the function $f(\mathbf{C}, \mathbf{G}) = \|\mathbf{C}\mathbf{G}\|_{\mathsf{F}}^2$. We have the following:

$$\nabla_{\mathbf{G}}^2 f(\mathbf{C}, \mathbf{G}) = \mathbf{C}^\top \mathbf{C}. \tag{9}$$

Since the R.H.S. of Equation 9 is p.s.d., Equation 9 implies that f is convex in the second parameter.

Next, notice that the constraint on the set of matrices $\mathbf{G}$ is *equivalent* to the following set of matrices:

$$\mathcal{A} = \left\{ \mathbf{A} \in \mathbb{R}^{k \times d} : \forall i \in [k], j \in [d], \mathbf{A}[i,j] \geq 0, \forall i \in [k], \sum_{j=1}^{d} \mathbf{A}[i,j] \leq 1 \right\}. \tag{10}$$

Equation 10 is essentially placing the constraint on the squared entries of $\mathbf{G}$. It is not hard to observe that $\mathcal{A}$ is a convex set, and the boundary of $\mathcal{A}$ is a set of row-stochastic matrices, i.e., $\forall i \in [k], \sum_{j=1}^{d} \mathbf{A}[i,j] = 1$. Now, since by Equation 9 the function f is convex in the second parameter, hence the maximum of f within the constraint set defined on $\mathbf{G}$ is attained at the extreme points of the set of row-stochastic matrices defined above. We will now use the following property about the extreme points of row-stochastic matrices.

Claim F.1 (Theorem 1 in Cao et al. (2022))**.** *The set of extreme points of the set of* $k \times d$ *row-stochastic matrices are precisely the set of row permutation matrices, i.e., set of the matrices with entries in* $\{0, 1\}^{k \times d}$ *and each row has exactly one non-zero entry.*

Lemma F.1 implies that the set of matrices we only need to consider to obtain the maximum of the function f are the matrices in $\{0, \pm 1\}^{k \times d}$ with each row having exactly one non-zero entry. Call

these set of matrices $\mathcal{B}$. Now consider any $\mathbf{B} \in \mathcal{B}$. It is not hard to observe that there exists an $\mathbf{u} \in \{\pm 1\}^k$ s.t. $\|\mathbf{C}\mathbf{u}\|_2 = \|\mathbf{C}\mathbf{B}\|_{\mathsf{F}}$. Since, by assumption for any such $\mathbf{u}$ we have, $\|\mathbf{C}\mathbf{u}\|_2 \leq 1$, it immediately implies that $\|\mathbf{C}\mathbf{B}\|_{\mathsf{F}} \leq 1$. □

F.2 Proof of Corollary 2.1

Corollary 2.1 Restated. *Assume that for any (matrix) $\mathbf{G} \in \mathfrak{D}$, there is some set of indices $i_{\mathbf{G}}$ such that every row of $\mathbf{G}$ not appearing in this set is zero. Define:*

$$\widetilde{\mathfrak{D}} = \bigcup_{\mathbf{G} \in \mathfrak{D}} \{\mathbf{u} : \mathbf{u}[i] \in \{-1, 1\} \textit{ if } i \in i_{\mathbf{G}}, 0 \textit{ otherwise}\}.$$

Then

$$\sup_{\mathbf{G} \in \mathfrak{D}} \|\mathbf{C}\mathbf{G}\|_F \leq \sup_{\mathbf{u} \in \widetilde{\mathfrak{D}}} \|\mathbf{C}\mathbf{u}\|_2.$$

Proof. This is an immediate application of Theorem 2.1, noting that one may delete the zero-rows of $\mathbf{G}$ without changing the Frobenius norm. □

F.3 Proof of Theorem 2.2

Theorem 2.2 Restated. *Let $\mathbf{C} \in \mathbb{R}^{n \times n}$, and take some participation pattern Π. Assume $\mathfrak{D}$ of vectors in $\mathbb{R}^n$ satisfies Π, and the absolute value of each entry of $\mathbf{u}$ is at most 1 $\forall \mathbf{u} \in \mathfrak{D}$.*

Suppose $|\pi| \leq k \; \forall \pi \in \Pi$. With $\mathbf{C}[:, \pi]$ representing to selecting the columns of the matrix $\mathbf{C}$ indexed by π and $\|\cdot\|_2$ the spectral matrix norm, let $\lambda = \max_{\pi \in \Pi} \|\mathbf{C}[:, \pi]\|_2$. Then the map $\mathbf{x} \mapsto \mathbf{C}\mathbf{x}$ has ℓ_2 sensitivity at most $\lambda\sqrt{k}$ with respect to $\mathfrak{D}$.

Proof. This result follows from a simple calculation.

$$\max_{\mathbf{u} \in \mathfrak{D}} \|\mathbf{C}\mathbf{u}\|_2 \leq \max_{\pi \in \Pi} \max_{\mathbf{u} \in \mathfrak{D}} \|\mathbf{C}[:, \pi]\|_2 \, \|\mathbf{u}\|_2 \leq \lambda, \sqrt{k},$$

. by evaluating the maximum ℓ_2 norm for $\mathbf{u} \; in \mathfrak{D}$ as $\sqrt{k}$ using the assumptions on $\mathfrak{D}$.

□

G Analysis for Section 3

G.1 Proof of Theorem 3.1

Theorem 3.1 Restated. *Let a finite $\mathfrak{D} = \{\mathbf{u}_i\}_{i=1}^k$ be given, and assume that the vectors $\{\mathbf{u}_i\}_{i=1}^k$ span $\mathbb{R}^n$. Assume that $\mathbf{A}$ is full-rank, and for $\mathbf{v} \in \mathbb{R}^k$ define*

$$\mathbf{H}_{\mathbf{v}} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \operatorname{diag}(\mathbf{v})^{1/2}, \quad \mathbf{U} = \mathbf{H}_{\mathbf{v}} \mathbf{H}_{\mathbf{v}}^\top.$$

Then, for Lagrange multipliers $\mathbf{v}$ such that the $\mathbf{U}$ is full-rank, the Lagrange dual function g can be expressed in closed form in terms of the Lagrange multipliers:

$$g(\mathbf{v}) := \inf_{\mathbf{X} \textit{ is PD}} L(\mathbf{X}, \mathbf{v}) = 2 \operatorname{tr}\left(\left(\mathbf{U}^{\frac{1}{2}} \mathbf{A}^\top \mathbf{A} \mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) - \sum_{\mathbf{u} \in \mathfrak{D}} \mathbf{v}_{\mathbf{u}} \tag{11}$$

Proof. Since there is some finite set of vectors $\mathbf{u} \in \mathbb{R}^n$ specifying $\mathfrak{D}$, the supremum in Eq. (4) may be reduced to a maximum over these elements.

Our problem then takes the form:

$$\begin{aligned} \inf_{\mathbf{X} \text{ is PD}} \quad & \operatorname{tr}(\mathbf{A}^\top \mathbf{A} \mathbf{X}^{-1}) \\ \text{s.t.} \quad & \mathbf{u}^\top \mathbf{X} \mathbf{u} \leq 1, \; \forall \mathbf{u} \in \mathfrak{D}. \end{aligned} \tag{12}$$

Recall that we have defined $\mathbf{H_v} = [\mathbf{u}_1, \ldots, \mathbf{u}_k] \operatorname{diag}(\mathbf{v})^{\frac{1}{2}}$, and $\mathbf{U} = \mathbf{H_v}\mathbf{H_v}^\top$. Now, note:

$$\mathbf{U} = \sum_{\mathbf{u}} \mathbf{v_u}\mathbf{u}\mathbf{u}^\top = \mathbf{H}\operatorname{diag}(\mathbf{v})\mathbf{H}^\top = \mathbf{H_v}\mathbf{H_v}^\top. \tag{13}$$

Introducing Lagrange multipliers $\mathbf{v_u} \geq 0$, for the problem Eq. (12) we form the Lagrangian for positive-definite $\mathbf{X}$:

$$L(\mathbf{X}, \mathbf{v}) = \operatorname{tr}(\mathbf{A}^\top\mathbf{A}\mathbf{X}^{-1}) + \sum_{\mathbf{u}} \mathbf{v_u}\left(\mathbf{u}^\top\mathbf{X}\mathbf{u} - 1\right) \tag{14}$$

$$= \operatorname{tr}(\mathbf{A}^\top\mathbf{A}\mathbf{X}^{-1}) + \operatorname{tr}\left(\left(\sum_{\mathbf{u}} \mathbf{v_u}\mathbf{u}\mathbf{u}^\top\right)\mathbf{X}\right) - \sum_{\mathbf{u}} \mathbf{v_u} \tag{15}$$

$$= \operatorname{tr}(\mathbf{A}^\top\mathbf{A}\mathbf{X}^{-1}) + \operatorname{tr}(\mathbf{U}\mathbf{X}) - \sum_{\mathbf{u}} \mathbf{v_u}. \tag{16}$$

For fixed $\mathbf{v}$, any finite minimizer of L for positive-definite $\mathbf{X}$ must correspond to a zero of this Lagrangian's gradient. We then compute the gradient

$$\frac{\partial L}{\partial \mathbf{X}} = -\mathbf{X}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{X}^{-1} + \mathbf{U}. \tag{17}$$

$\mathbf{U}$ and $\mathbf{A}$ are full-rank by assumption; therefore Lemma G.2 is applicable, and Eq. (17) has a unique positive-definite zero (and indeed, the infimum in Eq. (12) becomes a minimum):

$$\mathbf{X} = \mathbf{U}^{-\frac{1}{2}}\left(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}. \tag{18}$$

Note that Eq. (17) also immediately implies that if $\mathbf{U}$ is not full-rank, then there is no finite positive-definite minimizer of L in $\mathbf{X}$. Letting $g(\mathbf{v}) = \min_{\mathbf{X}} L(\mathbf{X}, \mathbf{v})$ be the Lagrange dual function and plugging back into Eq. (16), we have

$$\begin{aligned}
g(\mathbf{v}) &= \min_{\mathbf{X}\text{is PD}} \operatorname{tr}(\mathbf{A}^\top\mathbf{A}\mathbf{X}^{-1}) + \operatorname{tr}(\mathbf{U}\mathbf{X}) - \sum_{\mathbf{u}} \mathbf{v_u} \\
&= \min_{\mathbf{X}\text{is PD}} \operatorname{tr}(\mathbf{X}\mathbf{U}) + \operatorname{tr}(\mathbf{U}\mathbf{X}) - \sum_{\mathbf{u}} \mathbf{v_u} && \text{using Eq. (17)} \\
&= \min_{\mathbf{X}\text{is PD}} 2\operatorname{tr}(\mathbf{U}\mathbf{X}) - \sum_{\mathbf{u}} \mathbf{v_u} \\
&= 2\operatorname{tr}\left(\mathbf{U}\mathbf{U}^{-\frac{1}{2}}\left(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}\right) - \sum_{\mathbf{u}} \mathbf{v_u} && \text{using Eq. (18)} \\
&= 2\operatorname{tr}\left(\left(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) - \sum_{\mathbf{u}} \mathbf{v_u}.
\end{aligned}$$

□

G.1.1 Proof of Corollary 3.1

Corollary 3.1 Restated. *In the same setup as Theorem 3.1, a maximizer of the dual $\mathbf{v}^\star$ must satisfy:*

$$\mathbf{v}^\star = \operatorname{diagpart}\left(\left(\mathbf{H}_{\mathbf{v}^\star}^\top\mathbf{A}^\top\mathbf{A}\mathbf{H}_{\mathbf{v}^\star}\right)^{\frac{1}{2}}\right). \tag{19}$$

Moreover, the optimal value of the problem defined in 5 is $\operatorname{tr}(\mathbf{v}^\star)$.

Proof. As noted in the remark after Theorem 3.1, any optimal setting of the dual variables must be in the interior of a neighborhood in which the representation Eq. (6) is valid. It is therefore permissible to differentiate this representation.

Differentiating, we find:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{v}_i}\mathrm{tr}\left(\left(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) &= \frac{1}{2}\mathrm{tr}\left(\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}})^{-\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}\mathbf{u}_i\mathbf{u}_i^\top\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\mathbf{U}^{-\frac{1}{2}}(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}})(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}})^{-\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}\mathbf{u}_i\mathbf{u}_i^\top\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\mathbf{U}^{-\frac{1}{2}}(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}\mathbf{u}_i\mathbf{u}_i^\top\right) \\
&= \frac{1}{2}\mathrm{tr}\left(\mathbf{u}_i^\top\mathbf{U}^{-\frac{1}{2}}(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}\mathbf{u}_i\right) \\
&= \frac{1}{2}\mathbf{u}_i^\top\mathbf{X}\mathbf{u}_i,
\end{aligned}$$

by defining $\mathbf{X} = \mathbf{U}^{-\frac{1}{2}}\left(\mathbf{U}^{\frac{1}{2}}\mathbf{A}^\top\mathbf{A}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}$ (recalling the usage of the symbol $\mathbf{X}$ in Eq. (18)).

Therefore

$$\frac{\partial g(\mathbf{v})}{\partial \mathbf{v}_i} = \mathbf{u}_i^\top\mathbf{X}\mathbf{u}_i - 1,$$

and we have the stated expression for the gradient of the dual function.

Now, at a maximizer of the dual function, this derivative must vanish. An equivalent condition is $\mathrm{diagpart}(\mathbf{H}^\top\mathbf{X}\mathbf{H}) = \vec{1}$, and hence

$$\mathrm{tr}(\mathbf{U}\mathbf{X}) = \mathrm{tr}(\mathbf{H}\,\mathrm{diag}(\mathbf{v})\mathbf{H}^\top\mathbf{X}) = \mathrm{tr}(\mathrm{diag}(\mathbf{v})\mathbf{H}^\top\mathbf{X}\mathbf{H}) = \sum_{\mathbf{u}}\mathbf{v}_{\mathbf{u}}, \tag{20}$$

so at the optimum $\mathbf{v}^\star$ in fact $g(\mathbf{v}^\star) = \sum_{\mathbf{u}}\mathbf{v}^\star{}_{\mathbf{u}}$, establishing the second claim of our result.

Again using the observation that $\mathrm{diagpart}(\mathbf{H}_{\mathbf{v}}^\top\mathbf{X}\mathbf{H}_{\mathbf{v}}) = \vec{1}$ and so

$$\mathrm{diagpart}(\mathbf{H}_{\mathbf{v}}^\top\mathbf{X}\mathbf{H}_{\mathbf{v}}) = \mathrm{diagpart}(\mathrm{diag}(\mathbf{v})\mathbf{H}^\top\mathbf{X}\mathbf{H}) = \mathbf{v}.$$

Further, using the second claim of Corollary G.1, we can take

$$\mathbf{X} = \mathbf{H}_{\mathbf{v}}^{-\top}\left(\mathbf{H}_{\mathbf{v}}^\top\mathbf{A}^\top\mathbf{A}\mathbf{H}_{\mathbf{v}}\right)^{\frac{1}{2}}\mathbf{H}_{\mathbf{v}}^{-1},$$

and multiplying this by $\mathbf{H}_{\mathbf{v}}^\top$ and $\mathbf{H}_{\mathbf{v}}$ on the left and right respectively yields

$$\mathbf{H}_{\mathbf{v}}^\top\mathbf{X}\mathbf{H}_{\mathbf{v}} = \mathbf{H}_{\mathbf{v}}^\top\mathbf{H}_{\mathbf{v}}^{-\top}\left(\mathbf{H}_{\mathbf{v}}^\top\mathbf{A}^\top\mathbf{A}\mathbf{H}_{\mathbf{v}}\right)^{\frac{1}{2}}\mathbf{H}_{\mathbf{v}}^{-1}\mathbf{H}_{\mathbf{v}} = \left(\mathbf{H}_{\mathbf{v}}^\top\mathbf{V}\mathbf{H}_{\mathbf{v}}\right)^{\frac{1}{2}}$$

and so we conclude for the optimal Lagrange multiplier $\mathbf{v}^\star$,

$$\mathbf{v}^\star = \mathrm{diagpart}\left(\left(\mathbf{H}_{\mathbf{v}^\star}^\top\mathbf{A}^\top\mathbf{A}\mathbf{H}_{\mathbf{v}^\star}\right)^{\frac{1}{2}}\right). \tag{21}$$

□

G.2 Lemmas and Corollaries

Lemma G.1. *The set of positive-definite* $\mathbf{X}$ *such that* $\sup_{\mathbf{u}\in\mathfrak{D}}\mathbf{u}^\top\mathbf{X}\mathbf{u} \leq 1$ *is bounded as a subset of* $\mathbb{R}^{n\times n}$ *if and only if* $\mathfrak{D} = \{\mathbf{u}\}$ *spans* $\mathbb{R}^n$.

Proof. Suppose that $\mathfrak{D}$ spans $\mathbb{R}^n$. For a PSD matrix, a bound on the trace implies a bound on the elements; therefore it is sufficient to show that $\sup_{\mathbf{u}\in\mathfrak{D}}\mathbf{u}^\top\mathbf{X}\mathbf{u} \leq 1$ implies that the maximum eigenvalue of $\mathbf{X}$ is uniformly bounded for $\mathbf{X}$ PSD.

Take some set of vectors $\{\mathbf{u}_i\}_{i=1}^n \in \mathfrak{D}$ which span $\mathbb{R}^n$. Fix some representation

$$\mathbf{e}_i = \sum_{j=1}^n \alpha_{ij}\mathbf{u}_j,$$

where $\mathbf{e}_i$ is the ith standard basis vector.

Take $\mathbf{y}$ of ℓ_2 norm 1, and express:

$$\mathbf{y} = \sum_{i=1}^{n} \gamma_i \mathbf{e}_i.$$

Then for $\mathbf{X}$ satisfying our assumptions,

$$\left|\mathbf{y}^\top \mathbf{X}\mathbf{y}\right| = \left|\sum_{i,j=1}^{n} \gamma_i \gamma_j \mathbf{e}_i^\top \mathbf{X}\mathbf{e}_j\right| =\leq n^2 \max_{1\leq i,j\leq n} \left|\gamma_i\gamma_j\right| \left|\mathbf{e}_i^\top \mathbf{X}\mathbf{e}_j\right|. \tag{22}$$

Similarly,

$$\left|\mathbf{e}_i^\top \mathbf{X}\mathbf{e}_j\right| = \left|\sum_{k,l=1}^{n} \alpha_{ik}\alpha_{jl} \mathbf{u}_k^\top \mathbf{X}\mathbf{u}_l\right| \leq n^2 \max_{1\leq k,l\leq n} \left|\alpha_{ik}\alpha_{jl}\right| \left|\mathbf{u}_k^\top \mathbf{X}\mathbf{u}_l\right| \leq n^2 \max_{1\leq k,l\leq n} \left|\alpha_{ik}\alpha_{jl}\right|, \tag{23}$$

where the final inequality follows by the assumptions on $\mathbf{X}$.

Now, since the ℓ_2 norm of $\mathbf{y}$ is 1, the orthogonality of the $\mathbf{e}_i$ imply that each $|\gamma_i\gamma_j|$ is at most 1. Therefore:

$$\left|\mathbf{y}^\top \mathbf{X}\mathbf{y}\right| \leq n^4 \max_{1\leq i,j,k,l\leq n} \left|\alpha_{ik}\alpha_{jl}\right|, \tag{24}$$

and we have sufficiency of $\mathfrak{D}$ spanning $\mathbb{R}^n$.

For necessity, suppose $\mathfrak{D}$ does not span $\mathbb{R}^n$. Then there is some vector $\mathbf{y} \in \text{span}\left(\mathfrak{D}\right)^\perp$ of norm 1. Take any $\mathbf{X}$ such that $\sup_{\mathbf{u}\in\mathfrak{D}} \mathbf{u}^\top \mathbf{X}\mathbf{u} \leq 1$. Then, since $\mathbf{y}^\top \mathbf{u} = 0$ for all $\mathbf{u} \in \mathfrak{D}$, $\mathbf{Y} := \mathbf{X} + \alpha\mathbf{y}$ satisfies the same set of inequalities for any α.. □

Lemma G.2. *Let $\mathbf{U}, \mathbf{V} \in S_{++}^n$. Let $\mathbf{U} = \mathbf{U}_L\mathbf{U}_R$ be a factorization of $\mathbf{U}$ such that $\mathbf{U}_R\mathbf{V}\mathbf{U}_L$ is PSD, and the following equation defines a positive-definite matrix $\mathbf{X}$:*

$$\mathbf{X} = \mathbf{U}_R^\dagger \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} \mathbf{U}_L^\dagger. \tag{25}$$

Then, this $\mathbf{X}$ solves the equation

$$\mathbf{X}\mathbf{U}\mathbf{X} = \mathbf{V} \quad \textit{or equivalently} \quad \mathbf{U} = \mathbf{X}^{-1}\mathbf{V}\mathbf{X}^{-1}.$$

Moreover, this positive-definite solution $\mathbf{X}$ is unique.

Proof. We will begin by showing that $\mathbf{X}$ as defined by Eq. (25) solves the equation $\mathbf{X}\mathbf{U}\mathbf{X} = \mathbf{V}$; then we will show that any two positive-definite representations of the form Eq. (25) are in fact identical.

Notice that the representation $\mathbf{U} = \mathbf{U}_L\mathbf{U}_R$ implies that $\text{rank}(\mathbf{U}_L) \geq n$ and $\text{rank}(\mathbf{U}_R) \geq n$. Therefore $\mathbf{U}_L\mathbf{U}_L^\dagger = \mathbf{I} = \mathbf{U}_R^\dagger\mathbf{U}_R$, as implied by the Moore definition of the Moore-Penrose pseudoinverse. So:

$$\begin{aligned}
\mathbf{X}\mathbf{U}\mathbf{X} &= \mathbf{X}\mathbf{U}_L\mathbf{U}_R\mathbf{X} \\
&= \left(\mathbf{U}_R^\dagger \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} \mathbf{U}_L^\dagger\right) \mathbf{U}_L\mathbf{U}_R \left(\mathbf{U}_R^\dagger \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} \mathbf{U}_L^\dagger\right) \\
&= \mathbf{U}_R^\dagger \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} \mathbf{P}_{R(\mathbf{U}_L^\dagger)} \mathbf{P}_{R(\mathbf{U}_R)} \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} \mathbf{U}_L^\dagger
\end{aligned}$$

We claim that $\left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}}\mathbf{P}_{R(\mathbf{U}_L^\dagger)} = \mathbf{P}_{R(\mathbf{U}_R)}\left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}} = \left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}}$. In both cases, this can be seen by multiplying on the left or right as appropriate by $\left(\mathbf{U}_R\mathbf{V}\mathbf{U}_L\right)^{\frac{1}{2}}$, and noting $\mathbf{P}_{R(\mathbf{U}_R)}\mathbf{U}_R = \mathbf{U}_R$, $\mathbf{U}_L\mathbf{P}_{R(\mathbf{U}_L^\dagger)} = \mathbf{U}_L$. Since all the terms here are symmetric, the appropriate equality follows by uniqueness of the symmetric matrix square root. Therefore:

$$\begin{aligned}\mathbf{X}\mathbf{U}\mathbf{X} &= \mathbf{U}_R^\dagger\mathbf{U}_R\mathbf{V}\mathbf{U}_L\mathbf{U}_L^\dagger \\ &= \mathbf{V}.\end{aligned}$$

The uniqueness of a positive-definite $\mathbf{X}$ solving $\mathbf{X}\mathbf{U}\mathbf{X} = \mathbf{V}$ follows from the uniqueness of the usual matrix square root. Indeed, assume $\mathbf{Y}$ positive-definite satisfies $\mathbf{Y}\mathbf{U}\mathbf{Y} = \mathbf{V}$. Then:

$$\left(\mathbf{U}^{1/2}\mathbf{Y}\mathbf{U}^{1/2}\right)^2 = \mathbf{U}^{1/2}\mathbf{V}\mathbf{U}^{1/2}$$

Since the positive-definite square root is uniquely determined, $\mathbf{U}^{1/2}\mathbf{Y}\mathbf{U}^{1/2}$ is uniquely determined. Since $\mathbf{U}$ is invertible, $\mathbf{Y}$ is uniquely determined as well, and we have $\mathbf{Y} = \mathbf{X}$. □

Corollary G.1. *Two particular instantiations of Lemma G.2 are of interest.* $\mathbf{X}$ *as the matrix geometric mean of* $\mathbf{U}^{-1}$ *and* $\mathbf{V}$ *(taking* $\mathbf{U}_L = \mathbf{U}_R = \sqrt{\mathbf{U}}$*):*

$$\mathbf{X} = \mathbf{U}^{-\frac{1}{2}}\left(\mathbf{U}^{\frac{1}{2}}\mathbf{V}\mathbf{U}^{\frac{1}{2}}\right)^{\frac{1}{2}}\mathbf{U}^{-\frac{1}{2}}, \tag{26}$$

and assuming the representation $\mathbf{U} = \mathbf{H}_\mathbf{v}\mathbf{H}_\mathbf{v}^\top$*:*

$$\mathbf{X} = \mathbf{H}_\mathbf{v}^{\dagger\top}\left(\mathbf{H}_\mathbf{v}^\top\mathbf{V}\mathbf{H}_\mathbf{v}\right)^{\frac{1}{2}}\mathbf{H}_\mathbf{v}^\dagger. \tag{27}$$

Proof. By positive-definiteness of $\mathbf{U}$ and $\mathbf{V}$, Eq. (26) is clearly positive definite; Eq. (27) may be seen to be positive definite via the SVD of the pseudoinverses involved. Symmetry is again clear. Therefore both representations satisfy the assumptions of Eq. (25). □

H Analysis for Section 4

H.1 Additional Details

Defining the circulant matrix

$$\mathbf{A}_{\mathsf{circ}} \triangleq \begin{bmatrix} v_0 & v_{2n-1} & \cdots & v_1 \\ v_1 & v_0 & \cdots & v_2 \\ \vdots & \vdots & \cdots & \vdots \\ v_{2n-1} & v_{2n-2} & \cdots & v_0 \end{bmatrix}; \mathbf{v} \triangleq [\underbrace{1,\ldots,1}_{n},\underbrace{0,\ldots,0}_{n}] \tag{28}$$

Defining the DFT

$$\forall k \in [2n-1] : \mathbf{v}_{\mathsf{DFT}}[k] = \sum_{a=0}^{2n-1}\mathbf{v}(a)\exp\left(\frac{-j2\pi ka}{2n}\right) \tag{29}$$

Circulant matrices expressed using Fourier Transforms

Theorem H.1 (Adapted from Gray (2006))**.** *Consider any circulant matrix* $\mathbf{A}_{\mathsf{circ}} \in \mathbb{R}^{2n\times 2n}$*. Let* $\mathbf{F} \in \mathbb{C}^{2n\times 2n}$*, where the k-th row of* $\mathbf{F}$ *is given by* $\mathbf{F}[k,:] = \frac{1}{\sqrt{2n}}\left[\exp\left(-\frac{j2\pi ka}{2n}\right) : a \in \{0,\ldots,2n-1\}\right]$*. Then,* $\mathbf{A}_{\mathsf{circ}} = \mathbf{F}^*\Sigma\mathbf{F}$*, where* $\Sigma \in \mathbb{C}^{2n\times 2n}$ *is a diagonal matrix with the diagonal being the DFT (defined in Equation 29) of the first column of* $\mathbf{A}_{\mathsf{circ}}$*. Here,* * *is the Hermimitian operation.*

Privacy and utility guarantees In the following we provide the privacy guarantee and the main utility guarantee for the FFT mechanism defined in Algorithm 1.

Theorem H.2 (DP-Prefix Sum via FFT Privacy Guarantee)**.** *Algorithm 1 is ρ-zCDP in the adaptive continuous release model.*

Next, we analyze the utility of Algorithm 1 and show that it is nearly optimal in terms of the mean squared error (MSE) in the single-pass setting. First, we express the MSE in Theorem H.3 below.

Theorem H.3 (DP-Prefix Sum via DFT Utility)**.** *The MSE achieved by Algorithm 1 using the real and imaginary components of $\widetilde{\mathbf{z}}$ is*

$$\mathbb{E}\left[MSE\right] = \frac{\kappa^2 \left\|\mathbf{v}_{\mathsf{DFT}}\right\|_1^2}{2\rho n^2}.$$

In the following, we will have an explict expression for $\left\|\mathbf{v}_{\mathsf{DFT}}\right\|_1$ in terms of the problem parameters. Finally, we will argue that Theorem H.3 is nearly optimal.

Corollary H.1. *The expected mean squared error (MSE) is given by the following:*

$$\mathbb{E}\left[MSE\right] = \frac{\kappa^2}{2\rho n^2}\left(n + \sum_{a=0}^{\lfloor \frac{2n-1}{2} \rfloor} \frac{1}{\sin\left(\frac{\pi(2a+1)}{2n}\right)}\right)^2.$$

Near-optimal utility Here, we show that Theorem H.3 is near-optimal in utility for the single-participation setting. To do this, we compare with a lower bound on the expected MSE of any factorization-based mechanism from Henzinger et al. (2022, Theorem 2): $\frac{1}{2\rho\pi^2}\left(2 + \ln\left(\frac{2n+1}{3}\right) + \frac{\ln(2n+1)}{2n}\right)^2$. We find that the though our analytical upper bound in Corollary H.1 is $\approx$ 6x worse than the lower bound, the empirical noise added in Algorithm 1 closely tracks the lower bound to within a factor of 1.2x—because it only adds the real part of the noise. Results are in Figure 8 of Appendix H.1.

Showing near-optimal utility via MSE experiments

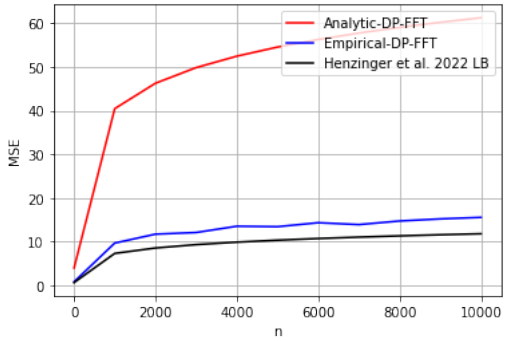

Figure 8: **Algorithm 1 achieves near-optimal utility** as measured by the analytic lower bound from Henzinger et al. (2022, Theorem 2).

H.2 Proof of Theorem H.2

Theorem H.2 Restated. *Algorithm 1 is ρ-zCDP in the adaptive continuous release model.*

Proof. First, consider the non-adaptive setting and the following mechanism, with parameters as defined in Algorithm 1,

$$\left[\Sigma\left(\mathbf{F}\mathbf{x}_{\mathsf{ext}} + \Sigma^{-1}\mathbf{z}\right)\right],$$
$$\text{where } \mathbf{z} = \sqrt{\frac{\kappa^2 \left\|\mathbf{v}_{\mathsf{DFT}}\right\|_1}{4n\rho}}\left(\sqrt{\Sigma}\cdot\mathbf{w}\right) \tag{30}$$

We claim that this satisfies $\frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\sigma^2}$-zCDP. To see this, we proceed by bounding ρ_i for each coordinate $i \in [2n]$ defined in Equation 30. For brevity, let $\mathbf{b} = \mathbf{F}\mathbf{x}_{\mathsf{ext}}$. Consider two neighboring data sets $\mathbf{g}$ and $\mathbf{g}'$, correspondingly, $(\mathbf{b}, \mathbf{x}_{\mathsf{ext}})$ and $(\mathbf{b}', \mathbf{x}'_{\mathsf{ext}})$. Then,

$$\|\mathbf{b} - \mathbf{b}'\|_\infty = \|\mathbf{F}(\mathbf{x}_{\mathsf{ext}} - \mathbf{x}'_{\mathsf{ext}})\|_\infty = \frac{\kappa}{\sqrt{2n}}. \tag{31}$$

We will now prove zCDP guarantee independently for each of the $2n$ coordinates and then use standard zCDP composition (Bun & Steinke, 2016). For any coordinate $a \in \{0, \ldots, 2n-1\}$, adding noise $\left(\frac{\sigma}{\sqrt{|\mathbf{v}_{\mathsf{DFT}}[i]|}}\right) \cdot \mathcal{N}_{\mathsf{complex}}(0,1)$ to $\mathbf{b}[i]$ satisfies ρ_i-zCDP with $\rho_i = \frac{\kappa^2|\mathbf{v}_{\mathsf{DFT}}[i]|}{4n\sigma^2}$. Then by composition, we have that

$$\rho = \sum_{a=0}^{2n-1}(\rho_i) = \frac{\kappa^2}{4n\sigma^2}\sum_{a=0}^{2n-1}(|\mathbf{v}_{\mathsf{DFT}}[i]|) = \frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\sigma^2}. \tag{32}$$

Therefore, setting $\sigma^2 = \frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\rho}$-satisfies a non-adaptive ρ-zCDP. Using the same σ, we prove the adaptive part using the same σ. We have the following from Equation 30.

$$\left[\mathbf{F}^*(\Sigma\mathbf{F}\mathbf{x}_{\mathsf{ext}} + \mathbf{z})\right] = \left[\mathbf{F}^*\sqrt{\Sigma}\left(\sqrt{\Sigma}\mathbf{F}\mathbf{x}_{\mathsf{ext}} + \frac{1}{\sqrt{\Sigma}}\mathbf{z}\right)\right] = \left[\mathbf{F}^*\sqrt{\Sigma}\left(\sqrt{\Sigma}\mathbf{F}\mathbf{x}_{\mathsf{ext}} + \sqrt{\frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\rho}}\cdot\mathbf{w}\right)\right] \tag{33}$$

Since, $\mathbf{w}$ in Equation 33 is spherical Gaussian, and the original query matrix $\mathbf{A}$ is lower triangular, by Theorem 2.1 in Denisov et al. (2022), the adaptive privacy guarantee follows. □

H.3 Proof of Theorem H.3

Theorem H.3 Restated. *The MSE achieved by Algorithm 1 using the real and imaginary components of $\widetilde{\mathbf{z}}$ is*

$$\mathbb{E}[MSE] = \frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1^2}{2\rho n^2}.$$

Proof. The MSE is given by the following:

$$\mathbb{E}[\mathtt{MSE}] = \frac{1}{n}\mathbb{E}\left[\|\widetilde{\mathbf{z}}[0,\ldots,n-1]\|_2^2\right] = \frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1}{2n^2\rho}\cdot\mathtt{trace}\left(|\Sigma[:n,:n]|\right) = \frac{\kappa^2\|\mathbf{v}_{\mathsf{DFT}}\|_1^2}{2n^2\rho}. \tag{34}$$

In equation 34, $\Sigma[:n,:n]$ refers to the top-left $n \times n$ submatrix of Σ. □

H.4 Proof of Corollary H.1

Corollary H.1 Restated. *Under the same setting as Theorem H.3, the MSE for Algorithm 1 is the following*

$$\mathbb{E}[MSE] = \frac{\kappa^2}{2\rho n^2}\left(n + \sum_{a=0}^{\lfloor\frac{2n-1}{2}\rfloor}\frac{1}{\sin\left(\frac{\pi(2a+1)}{2n}\right)}\right)^2.$$

Proof. Recall the definition of DFT from Equation 29 and of $\mathbf{v}$ in Equation 28. It is immediate that $\mathbf{v}_{\mathsf{DFT}}[0] = n$. For any $k \neq 0$, we have,

$$\mathbf{v}_{\mathsf{DFT}}[k] = \frac{1-\exp\left(\frac{-j2\pi kn}{2n}\right)}{1-\exp\left(\frac{-j2\pi k}{2n}\right)} = \frac{1-\exp(-j\pi k)}{1-\exp\left(\frac{-j\pi k}{n}\right)}. \tag{35}$$

From Equation 35, we have that when $k > 0$ is even, $\mathbf{v}_{\mathsf{DFT}}[k] = 0$. For k odd, we have

$$|\mathbf{v}_{\mathsf{DFT}}[k]| = \left|\frac{2}{1-\exp\left(\frac{-j\pi k}{n}\right)}\right| = \frac{1}{|\sin(\pi k/(2n))|} = \frac{1}{\sin(\pi k/(2n))} \tag{36}$$

Combining these, the term $\|\mathbf{v}_{\mathsf{DFT}}\|_1$ is

$$\|\mathbf{v}_{\mathsf{DFT}}\|_1 = n + \sum_{a=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{1}{\sin(\pi(2a+1)/(2n))} \tag{37}$$

□

H.5 Proof of Theorem 4.1

Theorem 4.1 Restated. *Under k participation, Algorithm 1 satisfies $(k^2\rho)$-zCDP.*

Proof. The proof goes exactly as Theorem H.2, except equation 31 gets replaced by the following:

$$\|\mathbf{b} - \mathbf{b}'\|_\infty = \|\mathbf{F}(\mathbf{x}_{\mathsf{ext}} - \mathbf{x}'_{\mathsf{ext}})\|_\infty = \frac{k\kappa}{\sqrt{2n}}. \tag{38}$$

□

H.6 Proof of Theorem 4.2

Theorem 4.2 Restated. *For any participation pattern Π with $|\pi| \leq k$, $\forall \pi \in \Pi$, let $\lambda = \max_{\pi \in \Pi} \left\| (\sqrt{\Sigma}\mathbf{F})[:, \pi] \right\|_2$, where $\mathbf{M}[:, \pi]$ corresponds to selecting the columns of the matrix $\mathbf{M}$ indexed by π, and let $\mu = \sqrt{\frac{\|\mathbf{v}_{\mathsf{DFT}}\|_1}{2n}}$. Then, Algorithm 1 satisfies $\left(\frac{k\lambda^2}{\mu^2} \cdot \rho\right)$-zCDP.*

Proof. Recall from equation 33, one can equivalently write Algorithm 1 as follows (where $\mathbf{w}$ is standard spherical Gaussian):

$$\left[\mathbf{F}^* \sqrt{\Sigma} \left(\sqrt{\Sigma}\mathbf{F}\mathbf{x}_{\mathsf{ext}} + \sqrt{\frac{\kappa^2 \|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\rho}} \cdot \mathbf{w} \right) \right]. \tag{39}$$

Let $\mathbf{x}_{\mathsf{ext}}$, $\mathbf{x}'_{\mathsf{ext}}$ be the neighboring data streams differing in a particular user's data, and π be the participation pattern of that particular user.

$$\left\| \sqrt{\Sigma}\mathbf{F}\left(\mathbf{x}_{\mathsf{ext}} - \mathbf{x}'_{\mathsf{ext}}\right) \right\|_2 \leq \kappa\sqrt{k} \left\| (\sqrt{\Sigma}\mathbf{F})[:, \pi] \right\|_2 \leq \kappa\sqrt{k} \cdot \lambda. \tag{40}$$

Since the noise added to $\sqrt{\Sigma}\mathbf{F}\mathbf{x}_{\mathsf{ext}}$ in equation 39 is $\sqrt{\frac{\kappa^2 \|\mathbf{v}_{\mathsf{DFT}}\|_1}{4n\rho}} \cdot \mathbf{w}$, and the sensitivity bound for m-participation from equation 40, the privacy guarantee follows from standard zCDP arguments for Gaussian mechanism (Bun & Steinke, 2016). □

I Translating from (complex) DFT to real-valued matrix mechanism

The DFT mechanism presented above can be understood as an application of a complex-valued matrix mechanism factorizing the prefix-sum matrix as

$$\mathbf{B} = \mathbf{P}\mathbf{F}^* \sqrt{\Sigma},$$

$$\mathbf{C} = \sqrt{\Sigma}\mathbf{F}\mathbf{E},$$

where $\mathbf{E}$ and $\mathbf{P}$ are appropriate embedding and projection matrices, respectively embedding an n-dimensional vector in the first n components of $\mathbb{R}^{2n}$, and projecting those same first n components back to $\mathbb{R}^n$, and following this application by 'chopping off' the imaginary part of the noise. All these operations are linear; and since everything begins and ends in the real domain, this mechanism can be expressed as a real-valued mechanism. Therefore identical codepaths can be used for implementing experiments with the FFT, though notably without some special implementation the mechanism, realizing the potential computational savings will not be immediate. In this small

section, we translate this complex-valued mechanism to a real-valued mechanism which can be integrated with the code backing the rest of the paper.

The real mechanism we implement and use to generate the DFT-based experimental results in the body can be expressed as a simple transformation of the mechanism above:

$$\widehat{\mathbf{B}} = \mathbf{B}\mathbf{F},$$
$$\widehat{\mathbf{C}} = \mathbf{F}^*\mathbf{C}.$$

Clearly $\widehat{\mathbf{B}}\widehat{\mathbf{C}} = \mathbf{B}\mathbf{C}$. Note that both these matrices are real-valued, as $\mathbf{F}^*\sqrt{\Sigma}\mathbf{F}$ defines a convolution of a real-valued vector. We claim that for any $\mathfrak{D}$, mechanism described above is distributionally equivalent to an application of the real-valued matrix mechanism with the factorization $\left(\widehat{\mathbf{B}}, \widehat{\mathbf{C}}\right)$, and satisfies the same privacy guarantees.

Indeed, to show this, it suffices to show two things:

1. $\Re[\mathbf{F}^*\sqrt{\Sigma}\mathbf{z}]$ (for $\mathbf{z}$ a sample from an isotropic complex Gaussian) is distributionally equivalent to $\mathbf{P}\mathbf{F}^*\sqrt{\Sigma}\mathbf{F}\mathbf{b}$ for $\mathbf{b}$ a sample from a real (isotropic) Gaussian with the same variance.
2. The sensitivity of $\widehat{\mathbf{B}}$ is identical to that of $\mathbf{B}$.

The first item is a consequence of the distributional invariance of the Gaussian under unitary transformations:

$$\begin{aligned}\Re[\mathbf{F}^*\sqrt{\Sigma}\mathbf{z}] &\sim \Re[\mathbf{F}^*\sqrt{\boldsymbol{\Sigma}}\mathbf{F}\mathbf{z}] \\ &= \mathbf{F}^*\sqrt{\Sigma}\mathbf{F}\Re[\mathbf{z}] \quad \text{(as } \mathbf{F}^*\sqrt{\Sigma}\mathbf{F} \text{ is real)} \\ &\sim \mathbf{F}^*\sqrt{\Sigma}\mathbf{F}\mathbf{b},\end{aligned}$$

where the variances are as desired.

The second item is similarly an immediate consequence of the definition of sensitivity Eq. (2), since the Fourier transform is unitary.