A Notation summary

n	Number of steps of the streaming linear query (SGD steps or FL rounds)
m	Total number of records (examples or users) in the database/dataset
b	Minimum separation between participations; $b = 1$ allows participation in every step
k	The maximum number of times any user might participate in training
d	Dimension of per-step user contributions (e.g., model size)
$\mathbf{x}_i \in \mathbb{R}$ or $\mathbb{R}^d$	Sum of per-example gradients (or per-user model updates) on step i
$\mathbf{x} \in \mathbb{R}^{n \times d}$	Stream of inputs $\mathbf{x}_i$, equiv. matrix with rows $\mathbf{x}_i$ (so $\mathbf{x}_i = \mathbf{x}_{[i,:]}$)
ζ	Clipping norm that limits the size of per-example contributions to $\mathbf{x}_i$
$\pi \subseteq [n]$	Participation pattern, the set of steps that an example participates in
Π	Participation schema, 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 input streams $\mathbf{x}, \tilde{\mathbf{x}}$.
$\mathcal{D}$	Corners of $\mathfrak{D}$ when assumed to be a polytope, $\mathfrak{D} = \operatorname{conv}(\mathcal{D})$.
(k, b)-participation	participation schema Π with at most k participations, separated by exactly b.
b-min-sep-participation	Relaxation of of (k, b)-participation where participations have separation at least b
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Lower-triangular linear query matrix to be factorized as $\mathbf{A} = \mathbf{BC}$
$\mathbf{M}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{M}$
$\mathbf{M}^\top$	Transpose of $\mathbf{M}$
$\mathbf{M}_{[i,j]}$	The $(i, j)^{\text{th}}$ entry of matrix $\mathbf{A}$
$\mathbf{M}_{[i,:]}$ and $\mathbf{M}_{[:,j]}$	The i^{th} row and j^{th} column of $\mathbf{M}$ (numpy-style indexing)
$\operatorname{conv}(S)$	Convex hull of the set S
$[n]$	$= \{1, \ldots, n\}$
$\lVert\mathbf{X}\rVert_F$	The Frobenius norm of a matrix $\mathbf{X}$

Table 1: Summary of notation

B Empirical evaluation of banded matrices

Table 2 compares the matrix mechanisms studied under different participation patterns but normalized to have sensitivity $\operatorname{sens}(\mathbf{C}) = 1$ under $(k{=}6, b{=}342)$-participation. The sensitivity under single participation $k = 1$ is lowest as expected. With column normalization, sensitivity is also 1 under $b{\geq}342$-min-sep-participation. We make the following observations:

- For the MF mechanisms, column normalization hurts RMSE for (k, b)-participation (as it is an additional constraint), but actually improves RMSE under b-min-sep-participation.
- We conjecture that the (k, b)-participation optimized matrices (MF without column normalization) are optimal for the prefix-sum workload[6]; With this in mind, we see there is at most

[6] This conjecture is not trivially true, as we enforce a non-negativity or orthogonality constraint; see Choquette-Choo et al. [15, Appendix I.3]. Hence the conjecture is that these constraints are already satisfied by the optimal matrix for this workload.

Matrix			Sensitivity			Error	
Mechanism	Bands $\hat{b}$	Equal colum norms? (Ours)	k=1 [17]	(k=6, b=342) [15]	$b\geq$342-min-sep (Ours)	(A) RMSE under (k=6, b=342) [15]	(B) RMSE under $b\geq$342-min-sep (Ours)
OPTIMAL TREEAGG [29, 34]	-	F	0.32	1.00	1.00	1.53	1.53
DP-SGD [1]	1	T	0.41	1.00	1.00	9.63	9.63
MF (b=128) (Ours)	128	F	0.52	1.00	1.04	1.23	1.29
MF (b=128) (Ours)	128	T	0.41	1.00	1.00	1.27	1.27
MF (b=342) (Ours)	342	F	0.52	1.00	1.04	1.04	1.08
MF (b=342) (Ours)	342	T	0.41	1.00	1.00	1.05	**1.05**
MF [15]	-	F	0.50	1.00	$\leq$1.15	**1.00**	1.15
MF [15]	-	T	0.41	1.00	$\leq$1.13	1.01	1.14

Table 2: A comparison of matrix mechanisms for $n = 2052$ under different participation patterns. **Banded matrices are near-optimal under (k, b)-participation and best under b-min-sep-participation.** Each error is computed under the indicated measure of sensitivity. Thus, the error in column (B) can be obtained by multiplying the error in column (A) by the corresponding entry under $b\geq$342 sensitivity.

a small increase in RMSE for switching to the more challenging b-min-sep-participation schema $(1.00 \rightarrow 1.05)$. If (as we further conjecture) the optimal matrices for prefix-sum in fact are k-banded, the gap is even smaller (at most $1.04 \rightarrow 1.05$). Hence, at least for the prefix-sum workload $\mathbf{A}$, there is limited room for improvement in developing optimization procedures that directly exploit b-min-sep-participation.

- Using fewer than b bands does degrade performance on the RMSE metric, with DP-SGD being the extreme case, yielding prefix sum estimates almost $10\times$ worse than the MF mechanisms.
- The results of Denisov et al. [17] imply that the binary-tree $\mathbf{C}$ matrix can in fact be used in the online setting, with the Moore-Penrose pseudo-inverse giving the optimal decoder for RMSE [15], corresponding to the 'full' estimator of Honaker [29]. We include this in the table as a baseline, and see that it is in general outperformed by our MF mechanisms by about $1.5\times$ in RMSE.

C Example structures of MF

Figs. 6 and 7 show the structure of some of the key matrix factorization approaches considered in this work. One can immediately see the impact of the k-participation schema in the optimal matrices, in particular for the non-banded MULTI-EPOCH MF matrices (the two top-right matrices), where $\mathbf{C}$ contains diagonals of negative entries separated by b steps. In the bottom two rows, we see that requiring equal column norms ("EN-" for equal norms) has a relatively minor impact on the structure of the matrices.

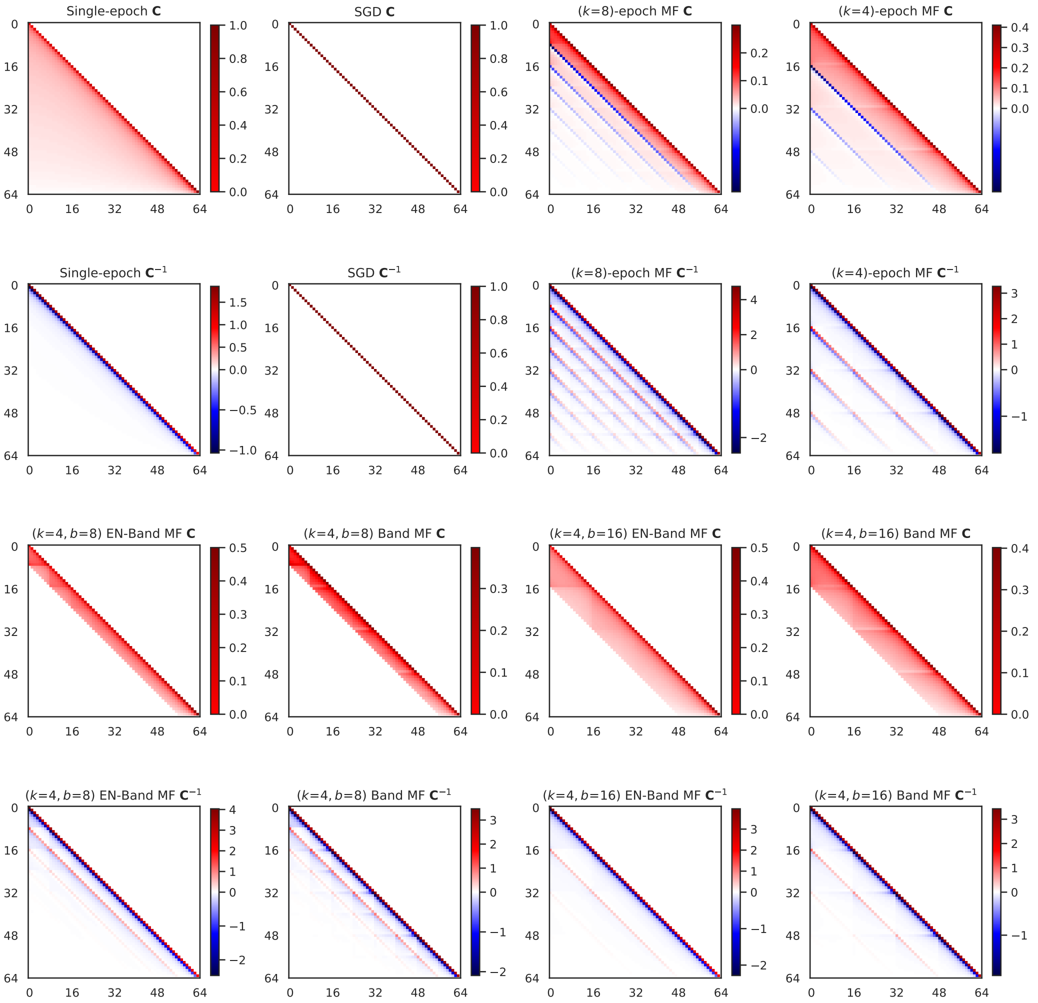


Figure 6: Factorizations for $n = 64$ of the prefix-sum workload ($\mathbf{A}$ taken to be the lower-triangular matrix of 1s). For each factorization $\mathbf{A} = \mathbf{BC}$, we show $\mathbf{C}$ and its inverse $\mathbf{C}^{-1}$, as the inverse is the matrix used in noise generation. Single-epoch is the approach of Denisov et al. [17], SGD is simply the identity matrix $\mathbf{I}$ (shown for completeness), and (k=8)-epoch MF and (k=4)-epoch are the Multi-epoch MF approach of Choquette-Choo et al. [15] for 8 and 4 epochs, respectively. For our banded matrices (3rd and 4th rows), we fix 4 epochs ($b = 16$), and show $\hat{b}$=8 and $\hat{b}$=16 bands, with column normalization ("EN-") and without.

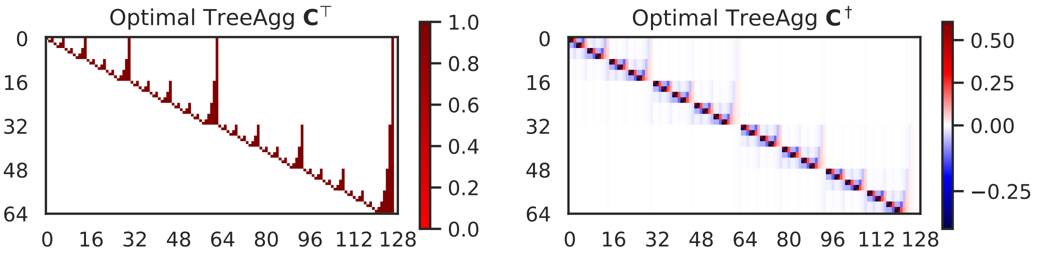


Figure 7: The transpose of binary-tree encoder matrix $\mathbf{C}_\mathcal{T}$, and its pseudoinverse $\mathbf{C}_\mathcal{T}^\dagger$, which corresponds to the "full" or optimal decoder of Honaker [29]. This is the matrix used in Optimal TreeAgg in Fig. 5[a].

D Algorithms and Analysis for Sec. 3

D.1 Algorithms

Algorithm 2 (VECSENS): Maximum of $\langle \mathbf{v}, \mathbf{u} \rangle$ where $\mathbf{u}$ is a vector in the ℓ_∞ unit ball satisfying Π_b.

```
Inputs: min-separation $b$, vector $\mathbf{v}$, max participations $k$
Initialize $F \in \mathbb{R}^{n \times k}$
for $m = 1, \dots, k$ do
    for $i = n, \dots, 1$ do        ▷ We use the convention that $F[s,t] = 0$ if $s, t$ are out-of-bounds.
        $F[i,m] = \max\Big(\mathbf{v}_i + F[i+b, m-1], F[i+1, m]\Big)$
return $\sqrt{F[1,k]}$
```

Algorithm 3 Efficient sensitivity upper bound for b-min-sep-participation

```
Inputs: min-separation $b$, matrix $\mathbf{X}$, max participations $k$
Initialize $F \in \mathbb{R}^{n \times k}$, $\mathbf{v} \in \mathbb{R}^n$.
for $j = 1, \dots, n$ do
    $\mathbf{v}_i = \text{VECSENS}(b, |\mathbf{X}_{[i,:]}|, k)$
return $\text{VECSENS}(b, \mathbf{v}, k)$
```

Algorithm 4 Efficient sensitivity calculation for b-min-sep-participation, assuming $\mathbf{X}$ is b-banded.

```
Inputs: min-separation $b$, $b$-banded matrix $\mathbf{X}$, max participations $k$.
return $\text{VECSENS}(b, \text{diag}(\mathbf{X}), k)$
```

D.2 Analysis

Proposition D.1. *The sensitivity of* $\mathbf{C}$ *for a given participation schema* Π *may be expressed as:*

$$\text{sens}_\Pi(\mathbf{C})^2 = \max_{\pi \in \Pi} \sup_{\mathbf{u} \in \mathfrak{D}} \text{tr}\left(\left[\mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi\right]\left[\mathbf{u}\mathbf{u}^\top\right]\right), \tag{6}$$

where $\mathbf{P}_\pi$ *represent the axis-aligned projection onto the set of rows indexed by* π*; that is,* $\mathbf{P}_\pi[i,i] = 1$ *for* $i \in \pi$*, and* 0 *otherwise. Assuming that* $\mathfrak{D}$ *represents a set of matrices with rows bounded by* ℓ_2 *norm 1, this can be upper bounded by:*

$$\max_{\pi \in \Pi} \sum_{i,j \in \pi} |\mathbf{X}_{[i,j]}|.$$

where $\mathbf{X} = \mathbf{C}^\top \mathbf{C}$*. This upper bound is tight when* $\mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi \geq 0 \forall \pi \in \Pi$*, and is independent of the dimension* d *of the rows of* $\mathbf{u}$*.*

Proof. Recall that Π determines the rows of $\mathbf{u}$ which may be nonzero in the definition Eq. (2). Take some $\mathbf{u} \in \mathfrak{D}$, an element of $\mathbb{R}^{n \times d}$, which therefore has nonzero rows only at some set of indices $\pi \in \Pi$.Note, clearly $\mathbf{u} = \mathbf{P}_\pi \mathbf{u}$, $\mathbf{P}_\pi^\top = \mathbf{P}_\pi$, and $\mathbf{P}_\pi = \mathbf{P}_\pi \mathbf{P}_\pi$.

Therefore

$$\begin{aligned}
\|\mathbf{C}\mathbf{u}\|_F^2 &= \text{tr}\left(\left[\mathbf{C}\mathbf{P}_\pi \mathbf{u}\right]^\top \mathbf{C}\mathbf{P}_\pi \mathbf{u}\right) = \text{tr}\left(\mathbf{u}^\top \mathbf{P}_\pi^\top \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi \mathbf{u}\right) \\
&= \text{tr}\left(\mathbf{P}_\pi \mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi \mathbf{P}_\pi \mathbf{u}\mathbf{u}^\top\right) = \text{tr}\left(\left[\mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi\right]\left[\mathbf{P}_\pi \mathbf{u}\mathbf{u}^\top \mathbf{P}_\pi\right]\right) \\
&= \text{tr}\left(\left[\mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi\right]\left[\mathbf{u}\mathbf{u}^\top\right]\right).
\end{aligned} \tag{7}$$

This implies the statement Eq. (6) by the definition of sensitivity and neighboring in our setting.

Now, let $\mathbf{X}_\pi := \mathbf{P}_\pi \mathbf{C}^\top \mathbf{C} \mathbf{P}_\pi$ be the matrix formed by zeroing out the rows and columns *not* indexed by π from $\mathbf{X}$. Assume that every $\mathbf{u} \in \mathfrak{D}$ has row norms bounded by 1. Expanding the trace in Eq. (6), writing x_{ij} for the elements of $\mathbf{X}_\pi$ and $\mathbf{u}_{[j,:]}$ for the j^{th} row of $\mathbf{u}$, we have

$$\operatorname{tr}\left(\mathbf{X}_\pi \mathbf{u}\mathbf{u}^\top\right) = \sum_{i=1}^{k}\sum_{j=1}^{k} x_{ij} \langle \mathbf{u}_{[i,:]}, \mathbf{u}_{[j,:]} \rangle \leq \sum_{i=1}^{k}\sum_{j=1}^{k} |x_{ij}|$$

which yields the claimed bound. When $\mathbf{X}_\pi$ is elementwise nonnegative, taking $\mathbf{u}_{[i,:]} = \mathbf{u}_{[j,:]}$ for any unit vector shows the claimed tightness in this case.

□

Remark. This statement can be viewed as a partial extension of [15, Theorem G.1]. It does not imply every case handled there, but also implies results which cannot be derived from that Theorem.

Proof of Thm. 2. Conclusion (1) is implied by (2), noting that the conditions on $\mathbf{C}$ imply that Algorithm 4 will return a value at most $\kappa\sqrt{k'}$ in this setting.

For (2), let $c \in \mathbb{R}^n$ with entries $c_i = \|\mathbf{C}_{[:,i]}\|^2$ for $i \in \{0, \ldots, n-1\}$. We have

$$\operatorname{sens}^1_\Pi(\mathbf{C}) = \max_{\pi \in \Pi_b} \|\mathbf{C}u(\pi)\| = \max_{\pi \in \Pi_b} \|\sum_{i \in \pi} \mathbf{C}_{[:,i]}\| = \max_{\pi \in \Pi_b} \sqrt{\sum_{i \in \pi} c_i} \tag{8}$$

where $u(\pi) \in \{0,1\}^n$ is given by $u(\pi)_i = 1$ if $i \in \pi$ and 0 otherwise. The last equality follows from the orthogonality condition on sufficiently separated columns of $\mathbf{C}$ trivially implied by bandedness. It is straightforward to verify the dynamic program of Algorithm 2 constructs a feasible π which attains the maximum. □

Proof of Thm. 3. Via Prop. D.1, the result follows from showing that Algorithm 3 outputs a value at least as large as $\sum_{(i,j)\in\pi} |\mathbf{X}_{ij}|$ for any $\pi \in \Pi_b$. So let $\hat{\pi}$ be an element of Π_b. Note that VECSENS is monotonically increasing in values of the vector $\mathbf{v}$ if $\mathbf{v}$ is nonnegative, and therefore Algorithm 3 is monotonically increasing in absolute values of $\mathbf{X}$. Therefore we will have our conclusion (3) if we can show that, for $\mathbf{X}_{\hat{\pi}}$ the matrix formed by zeroing out all rows and columns of $\mathbf{X}$ not indexed by $\hat{\pi}$, Algorithm 3 returns the value $\sum_{(i,j)\in\pi} |\mathbf{X}_{ij}|$. Yet this is straightforward by the characterization of VECSENS as an oracle for computing the maximum of $\langle \mathbf{v}, \mathbf{u} \rangle$, where $\mathbf{u}$ is a vector in the ℓ_∞ unit ball. □

E Additional Analysis for Sec. 5

Recall that we use b instead of $\hat{b}$ in this appendix since our sampling scheme enforces (k, b)-participation. Throughout this section, we slightly abuse notation by letting $i \pmod b = b$ instead of 0 if i/b is integer.

E.1 Algorithms for Sampling

Algorithm 5 Sampling scheme for banded DP-MF

```
Inputs: Dataset $D$, sampling distribution $\mathcal{S}$ over $(2^{[\check{m}]})^k$, noise standard deviation $\sigma$.
$D_1, \ldots, D_b \leftarrow$ arbitrary partition of $D$ such that $\forall j : |D_j| = \check{m}$.
Let $D_j = \{d_{j,1}, d_{j,2}, \ldots, d_{j,\check{m}}\}$ for each $j$.
for $j = 1, 2, \ldots, b$ do
    Sample $k$ sets to index $D_j$ as $(S_j, S_{b+j}, \ldots, S_{(k-1)b+j}) \sim \mathcal{S}$, with $S_j \subseteq [\check{m}]$.
for $i = 1, 2, \ldots, n$ do
    Let $j = i \pmod b$; compute $\mathbf{x}_i$ by querying $\{d_{j,\ell} : \ell \in S_i\}$.
Let $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times d}$, release $\mathbf{C}\mathbf{x} + \mathbf{z}$ with each entry of $\mathbf{z}_{[i,j]} \sim \mathcal{N}(0, \sigma^2)$.
▷ If $\mathbf{C}$ is lower-triangular, results can also be released in streaming fashion
```

Algorithm 6 Sequence of queries that bounds privacy of Algorithm 5

```
Inputs: Dataset $\tilde{D} = \{d_1, d_2, \ldots, d_{\check{m}}\}$, sampling distribution $\mathcal{S}$ over $(2^{[\check{m}]})^k$.
Sample $(S_1, S_2, \ldots, S_k) \sim \mathcal{S}$.
for $i = 1, 2, \ldots, k$ do
    $\tilde{D}_i \leftarrow \{d_j : j \in S_i\}$.
    Perform (adaptively chosen) sensitivity $\Delta$ query on $\tilde{D}_i$ with noise $\mathcal{N}(0, \sigma^2)$.
```

E.2 Proof for Thm. 4

Proof. Consider two datasets D, D' that differ by an example contained in the partition subset D_j. We argue about the privacy of $\mathbf{Cx} + \mathbf{z}$. For simplicity we assume j is such that $(k-1)b + j \leq n$; elements in D_j such that j does not satisfy this condition can potentially participate $k-1$ times instead of k, and in turn the privacy guarantee we can prove for these elements can only be stronger.

Since $\mathbf{C}$ is b-banded, we can partition the rows of $\mathbf{C}$ into $k+1$ subsets

$$R_j, R_{b+j}, R_{2b+j} \ldots R_{(k-1)b+j}, R_{\emptyset},$$

where R_j (resp. $R_{b+j}, R_{2b+j} \ldots R_{(k-1)b+j}$) denotes the set of rows in $\mathbf{C}$ for which the jth entry is non-zero, and $R_{\emptyset} = [n] \setminus (R_j \cup R_{b+j} \cup \ldots)$, i.e., $R_{\emptyset}$ are the rows not included in any of these sets, i.e., rows of $\mathbf{C}$ where entries $j, b+j, \ldots$ are all zero. The fact that $\mathbf{C}$ is lower-triangular and b-banded ensures that these subsets do not overlap, i.e., this is a valid partition as can be observed in Fig. 3.

Let $\mathbf{C}_R$ denote $\mathbf{C}$ restricted to the set of rows in R. From the perspective of an adversary distinguishing D from D', each row of $(\mathbf{Cx} + \mathbf{z})_{R_{\emptyset}} = \mathbf{C}_{R_{\emptyset}}\mathbf{x} + \mathbf{z}_{R_{\emptyset}}$ has a distribution independent of whether D or D' was used. So it suffices to give privacy guarantees for outputting only $(\mathbf{Cx} + \mathbf{z})_{R_j}, (\mathbf{Cx} + \mathbf{z})_{R_{b+j}}, \ldots, (\mathbf{Cx} + \mathbf{z})_{R_{(k-1)b+j}}$.

We can decompose rows R_j of $\mathbf{Cx} + \mathbf{z}$ as follows:

$$(\mathbf{Cx} + \mathbf{z})_{R_j} = \mathbf{C}_{R_j}\mathbf{x} + \mathbf{z}_{R_j} = \mathbf{C}_{R_j}\mathbf{x}_j + \mathbf{C}_{R_j}\mathbf{x}_{-j} + \mathbf{z}_{R_j}. \tag{9}$$

Where $\mathbf{x}_j$ denotes $\mathbf{x}$ with all rows except j zeroed out, and $\mathbf{x}_{-j}$ denotes $\mathbf{x} - \mathbf{x}_j$, i.e., $\mathbf{x}$ with row j zeroed out. By the b-banded property of $\mathbf{C}$, $\mathbf{C}_{R_j}\mathbf{x}_{-j}$ has 0 sensitivity to the examples in $D \setminus D_j$. Then, by Eq. (9), for $i \in R_j$, we observe that the ith row of $(\mathbf{Cx} + \mathbf{z})_{R_j}$ corresponds to an (adaptive) query made with ℓ_2-sensitivity $\mathbf{e}_i^\top \mathbf{C} \mathbf{e}_j$ to the examples used in round j, i.e., those given by D_j and S_j, and noise $N(0, \sigma^2)^d$. So $(\mathbf{Cx} + \mathbf{z})_{R_j}$ corresponds to a sequence of adaptive queries on the examples used in round j, and answering this sequence of queries satisfies any standard privacy guarantee satisfied by answering a single (scalar, adaptively chosen) query with sensitivity $\|\mathbf{Ce}_j\|_2$ to the example chosen in round j and noise $N(0, \sigma^2)$ by Claim D.1 in [17].

The same logic applies to each of $(\mathbf{Cx} + \mathbf{z})_{R_{b+j}}, \ldots, (\mathbf{Cx} + \mathbf{z})_{R_{(k-1)b+j}}$. Putting it all together and taking a max over the sensitivity of the individual queries, releasing $\mathbf{Cx} + \mathbf{z}$ satisfies any standard privacy guarantee satisfied by answering k adaptively chosen queries, with sensitivity $\max_{i \in [n]} \|\mathbf{Ce}_i\|_2$ to the examples used in rounds $j, b+j, \ldots, (k-1)b+j$ respectively. This is exactly Algorithm 6 with the specified choice of $\Delta, \mathcal{S}$. □

E.3 Corollaries of Thm. 4

We give here several corollaries of Thm. 4 that are of interest.

Equivalence to DP-SGD: Note that when $b = 1$, the partition contains a single subset, i.e., is the the entire dataset. In particular, in this setting Thm. 4 recovers the privacy guarantees of amplified DP-SGD under any amplification scheme, e.g. including the ones discussed below.

Amplification via sampling: Take the distribution over $2^{[\check{m}]}$ given by including each element of $[\check{m}]$ independently with probability q, and let $\mathcal{S}$ be the product of this distribution with itself k times. This is equivalent to the following: in round i, we include each element of $D_{i \pmod b}$ independently with probability q. In particular, within each D_j, we are just using sampling with replacement to choose which elements to include in each round. From this we get the following corollary, which allows us to reduce to a setting whose privacy guarantees are well-understood:

Corollary E.1. *Suppose the examples participating in round i of matrix factorization are chosen by including each element of $D_{i \pmod b}$ independently with probability q. Then the matrix factorization mechanism satisfies any standard privacy guarantee satisfied by k adaptive scalar queries with sensitivity $\max_{i \in [n]} \|\mathbf{C}\mathbf{e}_i\|_2$ and noise $N(0, \sigma^2)$, with the ith query run on a batch given by sampling each element of a $\check{m}$-element database with probability q.*

We next make this explicit in terms of the `dp_accounting` Python library [18]. Given n, m, b and a target per-round batch size B, we could write a `dp_accounting.DpEvent` capturing the privacy guarantees of the matrix factorization mechanism as follows:

```
gaussian_event = dp_accounting.GaussianDpEvent(noise_multiplier)
q = B / math.floor(n / b)
sampled_event = dp_accounting.PoissonSampledDpEvent(
    q, gaussian_event
)
composed_event = dp_accounting.SelfComposedDpEvent(
    sampled_event, math.ceil(m / b)
)
```

Example E.1. *To give an example of the amplification guarantee, for simplicity assume $n/b, m/b$ are integer. If all column norms in $\mathbf{C}$ are 1, each row of $\mathbf{x}$ has sensitivity 1, and each entry of $\mathbf{z}$ has standard deviation σ, then outputting $\mathbf{C}\mathbf{x} + \mathbf{z}$ satisfies $(\alpha, \frac{\alpha n}{2\sigma^2 b})$-RDP.*

Using Theorem 11 of [43] and Cor. E.1, for appropriate choice of α and q, this improves to $(\alpha, q^2 \cdot \frac{2\alpha n}{\sigma^2 b})$-RDP with amplification by sampling. In particular, if we have a target per-round batch size B, then we should choose $q = \frac{Bb}{m}$, and if this choice of q satisfies the conditions in [43] plugging this in gives $(\alpha, \frac{2\alpha B^2 bn}{\sigma^2 m^2})$-RDP. Notice that $b = 1$ recovers the privacy guarantees of DP-SGD with Poisson sampling, and this privacy guarantee weakens as b increases.

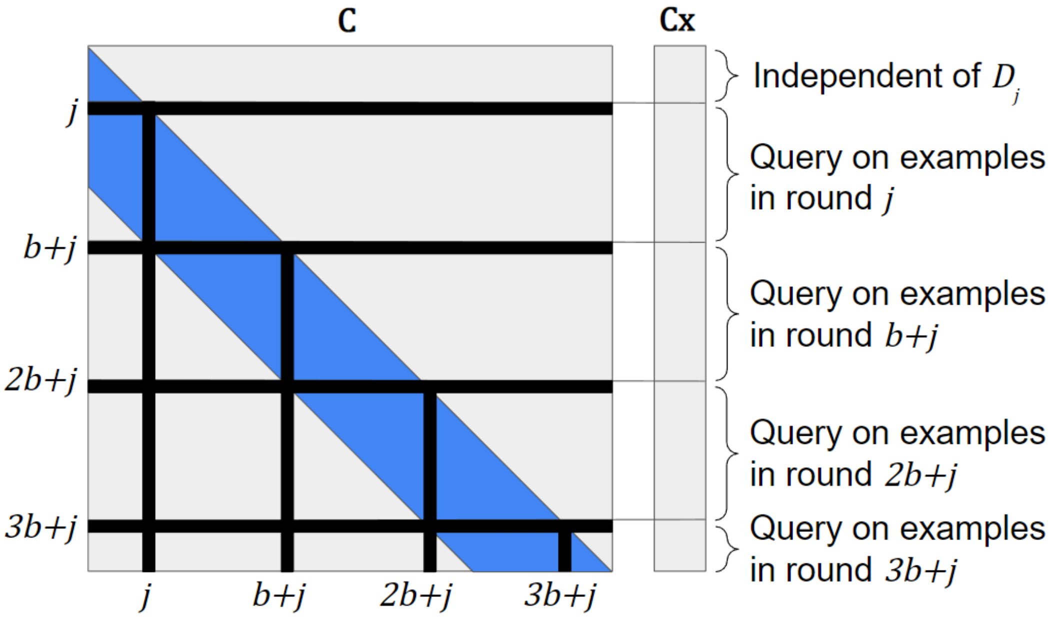
Amplification via shuffling: Fix a per-round batch size B. Then, suppose we shuffle the list of examples, and cyclically iterate over batches of size B in this list as the sets of examples to use in each round of matrix factorization. That is, we shuffle D into an ordered list $d_1, d_2, \ldots$, and in round i use examples $d_{(i-1)B+1 \pmod m}, d_{(i-1)B+2 \pmod m}, \ldots, d_{iB \pmod m}$.

For simplicity let's consider the case where $m/(Bb)$ is integer. In particular, this means in this shuffling scheme, each example appears once every m/B rounds, and for each of these rounds i, $i \pmod b$ is the same. Then this shuffling scheme is equivalent to the following: First, rather than choose an arbitrary partition to apply Thm. 4, we choose a uniformly random partition into b subsets of size m/b. Then, we choose $\mathcal{S}$ to be the distribution giving by shuffling $[m/b]$ and then cyclically iterating over the shuffled list in batches of size B. Given this equivalence, we get the following:

Corollary E.2. *Suppose the examples in matrix factorization are chosen by shuffling D and then iterating over batches of size B. If $n/(Bb)$ is integer, then the matrix factorization mechanism satisfies any standard privacy guarantee satisfied by k adaptive scalar queries with sensitivity $\max_{i \in [n]} \|\mathbf{C}\mathbf{e}_i\|_2$ and noise $N(0, \sigma^2)$, with the examples in each query given by shuffling a dataset of size m/b and cyclically iterating over this list in batches of size B.*

Example E.2. *Consider the simplified case where $m = n$, we choose a random permutation π, and in round i query example $d_{\pi(i)}$. In this case, if all the column norms of $\mathbf{C}$ are 1, $\mathbf{x}$'s rows have sensitivity 1, and $\mathbf{z}$'s entries have standard deviation $\sigma = \mathcal{O}\left(\frac{\sqrt{\ln(1/\delta)}}{\epsilon}\right)$, we get that $\mathbf{C}\mathbf{x} + \mathbf{z}$ satisfies (ϵ, δ)-DP. With e.g., the amplification for shuffled (ϵ, δ)-DP mechanisms given by Theorem 5.1 of [6] and Cor. E.2, if ϵ is a constant, we instead get that $\mathbf{C}\mathbf{x} + \mathbf{z}$ satisfies $\left(\epsilon \cdot \mathcal{O}\left(\sqrt{\frac{b \log(1/\delta)}{n}}\right), \delta \cdot \mathcal{O}\left(\frac{n \ln(1/\delta)}{b}\right)\right)$-DP.*

E.4 Additional Figures



C
Cx
j
$b+j$
$2b+j$
$3b+j$
j
$b+j$
$2b+j$
$3b+j$
Independent of D_j
Query on examples in round j
Query on examples in round $b+j$
Query on examples in round $2b+j$
Query on examples in round $3b+j$

Figure 8: A visualization of how we can decompose a banded matrix mechanism into independent queries on D_j (as in Algorithm 6) under our sampling scheme.

F Additional RMSE Experiment Details

In this section, we provide more discussion and supplementary experiments surrounding the RMSE experiments in Fig. 4. Table 3 shows the optimal number of bands for each (ϵ, k) pair considered in the RMSE experiments. It shows the general trend that as ϵ decreases, or k increases, the optimal number of bands decreases.

ϵ/k	1	2	4	8	16	32	64	128	256	512	1024
0.03125	2	2	1	1	1	1	1	1	1	1	1
0.0625	4	2	1	1	1	1	1	1	1	1	1
0.125	8	4	2	1	1	1	1	1	1	1	1
0.25	8	4	4	2	1	1	1	1	1	1	1
0.5	16	8	4	4	2	1	1	1	1	1	1
1.0	32	16	8	4	2	2	1	1	1	1	1
2.0	64	32	16	8	4	2	2	1	1	1	1
4.0	128	64	32	16	8	4	2	2	1	1	1
8.0	1024	512	256	32	16	8	4	2	2	1	1
16.0	1024	512	256	128	64	32	8	4	4	2	1

Table 3: Optimal number of bands for each (ϵ, k) pair, when $n = 1024$ and $\delta = 10^{-6}$.

G Additional CIFAR-10 Experiment Details

G.1 Setup and Tuning

We tune all jobs on a learning rate grid of coefficients in [1, 2, 5] on powers in [-2, 3]. We find that no momentum works best for DP-SGD and momentum=0.95 works best for MF-DP-FTRL mechanisms on average in tuning; though initial tuning found that tuning momentum as well could lead to slightly better results at some ϵ budgets, we found that a more refined grid of learning rates nearly always led to a fixed momentum being optimal, and so we fix this parameter. We also found that a learning rate cooldown to $0.05\times$ the initial learning rate over the last 500 steps of training improved all runs and so we fix this parameter. All models trained for 20 epochs on CIFAR10 with a batch size of 500. We repeat each setting 12 times and show 95% bootstrapped confidence intervals.

G.2 Additional Figures

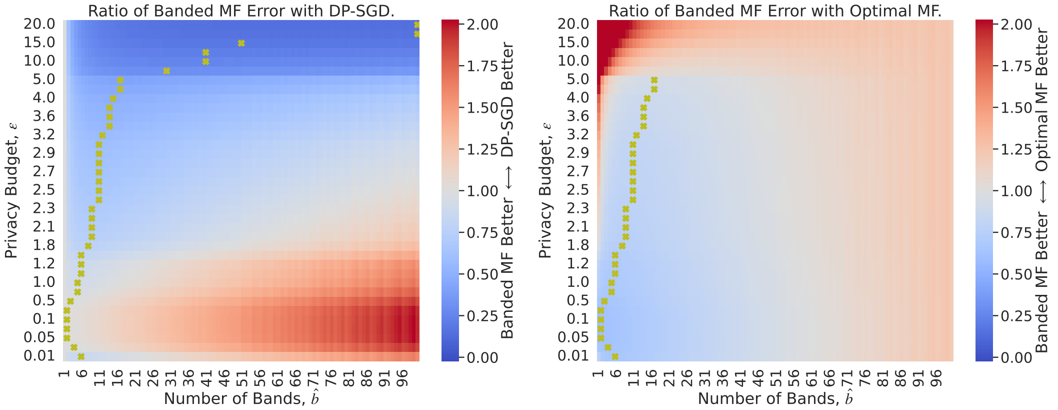


Figure 9: **BANDMF is at least as good as DP-SGD across all ϵ, and often significantly better.** BANDMF is better than the prior MF-DP-FTRL from Choquette-Choo et al. [15] up to $\epsilon \approx 5$. We compare the ratio of the total error (see Sec. 4) of BANDMF with either mechanism. Lower values indicate that BANDMF is better. The yellow markers indicate the best BANDMF mechanism that was better for that ϵ budget if one existed. Note that we only optimize the Band MF over $\hat{b} \in [0, n/k]$ which leads to a regime around $\epsilon > 5$ where the it performs worse than the Multi-epoch MF of Choquette-Choo et al. [15]; $\hat{b} = n$ is equivalent to this approach modulo the sensitivity definition which we exclude to emphasize the regime we improve on.

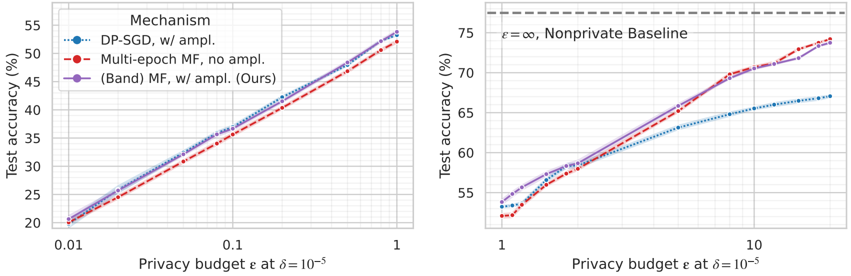


Figure 10: **Our banded matrices consistently perform at least as well as the best prior method in each range of ϵ.** Around $\epsilon \approx 1$, we observe significant utility benefits from the banded mechanism around $2-3$ percentage points over DP-SGD. Note that we only optimize the Band MF over $\hat{b} \in [0, n/k]$ which leads to a regime around $\epsilon > 5$ where the it performs worse than the Multi-epoch MF of Choquette-Choo et al. [15]; $\hat{b} = n$ is equivalent to this approach modulo the sensitivity definition which we exclude to emphasize the regime we improve on. Empirical setup is in App. G.

H Additional StackOverflow Next-Word-Prediction Experiment Details

We follow the experimental setup for StackOverflow NWP from Denisov et al. [17] and Choquette-Choo et al. [15]. Except for SINGLE-EPOCH MF (which uses $B = 167$ clients/round for 1 epoch), all privacy guarantees and accuracy results are for 6 epochs of training using $B = 1000$ clients/round for 2052 rounds (also 1 epoch). The matrices used in these experiments are included in Table 2.

For computational efficiency in estimating model accuracy at a given privacy guarantee, we actually compute in simulation updates from only 100 clients/round, and scale the noise multiplier by a corresponding factor ($\frac{100}{1000}$ for 6 epoch experiments, $\frac{100}{167}$ for SINGLE-EPOCH MF). This approach has been used previously [34, 41], and we independently verified it has a negligible impact on the estimates of accuracy figures we report. Tables 4 and 5 include the unscaled noise multipliers σ for our experiments.

Optimizer and learning-rate tuning For all SO NWP experiments we use the FedSGDM optimizer [47]. This optimization approach first takes multiple local SGD steps (with learning rate 1.0 in our experiments) on the training data of each user in the batch (cohort) before clipping to $\zeta = 1$, summing, and passing the result $\mathbf{x}_i$ into the DP mechanism which adds noise $[\mathbf{C}^\dagger \mathbf{z}]_{[i,:]} \in \mathbb{R}^d$ on each iteration i. The resulting privatized sum is then divided by the batch size B and passed to the “server” (post-aggregation) optimizer, in our case SGDM with momentum parameter $\beta = 0.95$ and learning rate η_s. We find tuning η_s depending on the noise level is critical. By using the computationally efficient approach mentioned above, we were able to conduct rigorous tuning over a learning rate grid of 1.7^i for powers i in $\{-9, \ldots, 4\}$, estimating good initial guesses based on prior work. Table 6 gives the full set of results, and Fig. 12 shows convergence as a function of the number of rounds (iters).

Learning rate warmup and cooldown Denisov et al. [17] found learning rate cooldown was effective, and Choquette-Choo et al. [15] found that zeroing-out client updates with large ℓ_∞ norms was critical to stability in early training. We find that additionally introduing a learning-rate warmup schedule reduces the need for this zeroing-out (though we still enable it), and generally decreases the variance in training results. Hence, all of our experiments (for all algorithms) using a linear learning rate warmup from $0.05\eta_s$ to $1.0\eta_s$ over the first 15% of rounds (309), and a linear decay from $1.0\eta_s$ to $0.05\eta_s$ over the last 25% of rounds (513).

Using RMSE to tune select optimal server learning rates Fig. 11 plots the server learning rates η_s from Table 6 on the y-axis (with the optimal rates shown as larger symbols, and sub-optimal rates as small symbols, versus two different measures of the error for the DP mechanism on the x-axis: The left plot gives uses the effective prefix-sum RMSE (the objective we use for optimizing (banded) matrices $\mathbf{C}$),

$$\text{(Mechanism error)} \times \text{noise-multiplier}/\text{(clients-per-round)} = \sqrt{\mathcal{L}(\mathbf{S}\mathbf{C}^{-1}, \mathbf{C})/n} \times \sigma/B, \tag{10}$$

where $\mathbf{S}$ is the prefix-sum workload (lower-triangular matrix of ones) and σ and B are as given in Table 4. The right plot uses the RMSE in error of individual gradients, computed by replacing the $\mathcal{L}$ term in the above with $\mathcal{L}(\mathbf{I}\mathbf{C}^{-1}, \mathbf{C})$ where we take the workload $\mathbf{A}$ to be the identity matrix $\mathbf{I}$ rather than the prefix sum matrix $\mathbf{S}$.

We see a strong linear correlation between the prefix-sum RMSE and optimal learning rate in the left plot; this does not hold for individual gradient errors (right plot). Based on this, we use the following linear regression to choose learning rates for the non-federated (amplified) SO NWP experiments (still rounding to the nearest 1.7^i for consistency):

$$\log(\eta_s) = -0.95 \cdot \log(L_e) - 4.64$$

This allowed us to estimate learning rates for the amplified experiments with a high degree of accuracy; Table 7 gives the final selected learning rates.

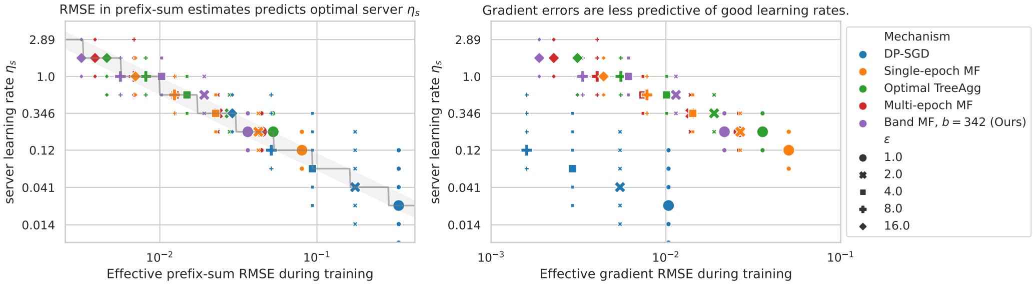


Figure 11: Correlation between optimal server learning rates η_s and the effective RMSE during training, see Eq. (10).

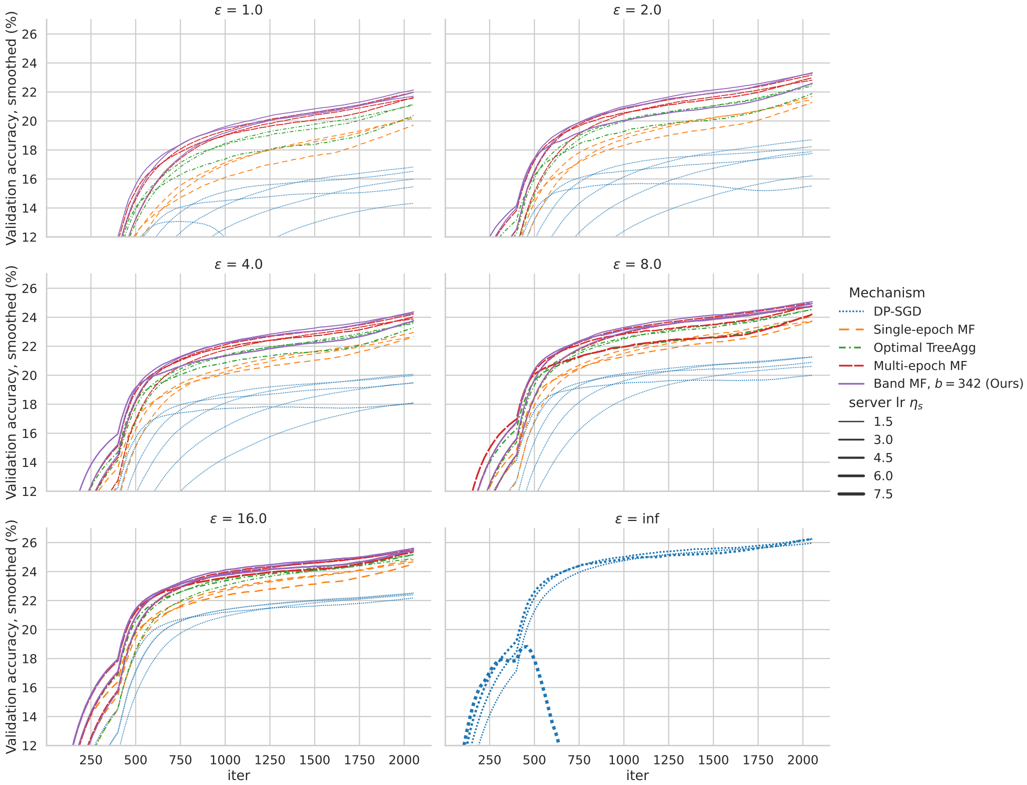


Figure 12: Convergence plots for all cross-device federated learning simulation experiments.

Mechanism	clients per round B	ϵ	noise mult. σ	server lr η_s	Eval Accuracy (%, Smoothed)	Test Accuracy (%)
DP-SGD	1000	1	4.22468	0.0244	16.82	16.69
Single-epoch MF	167	1	4.22468	0.1197	20.29	20.44
Optimal TreeAgg	1000	1	4.22468	0.2035	21.15	21.25
Multi-epoch MF	1000	1	4.76079	0.2035	21.96	21.92
Band MF (Ours)	1000	1	4.22468	0.2035	22.12	22.05
DP-SGD	1000	2	2.23048	0.0414	18.70	18.42
Single-epoch MF	167	2	2.23048	0.2035	21.66	21.70
Optimal TreeAgg	1000	2	2.23048	0.3460	22.52	22.59
Multi-epoch MF	1000	2	2.51352	0.3460	23.15	23.04
Band MF (Ours)	1000	2	2.23048	0.5882	23.31	23.19
DP-SGD	1000	4	1.19352	0.0704	20.07	19.81
Single-epoch MF	167	4	1.19352	0.3460	22.94	22.90
Optimal TreeAgg	1000	4	1.19352	0.5882	23.66	23.62
Multi-epoch MF	1000	4	1.34498	0.5882	24.19	24.02
Band MF (Ours)	1000	4	1.19352	1.0000	24.35	24.16
DP-SGD	1000	8	0.65294	0.1197	21.26	21.08
Single-epoch MF	167	8	0.65293	0.5882	24.03	23.88
Optimal TreeAgg	1000	8	0.65294	1.0000	24.54	24.45
Multi-epoch MF	1000	8	0.73579	1.0000	24.95	-
Band MF (Ours)	1000	8	0.65294	1.0000	25.06	24.88
DP-SGD	1000	16	0.36861	0.3460	22.51	22.26
Single-epoch MF	167	16	0.36861	1.0000	24.80	24.62
Optimal TreeAgg	1000	16	0.36861	1.7000	25.15	25.14
Multi-epoch MF	1000	16	0.41539	1.7000	25.50	25.33
Band MF (Ours)	1000	16	0.36861	1.7000	25.59	25.41

Table 4: Parameters and metrics for Fig. 5[a]. The noise multipliers are calibrated to achieve the given ϵ guarantees at $\delta{=}10^{-6}$ under $b{=}342$-min-separation. The matrices are scaled to have sensitivity 1 under $(k{=}6, b{=}342)$, see Table 2, and so a larger noise multiplier σ is necessary for the MULTI-EPOCH MF matrices. Test-set accuracy for MULTI-EPOCH MF at $\epsilon = 8$ was unavailable.

Mechanism	clients per round B	ϵ	noise mult. σ	server lr η_s	Eval Accuracy (%, Smoothed)	Test Accuracy (%)
DP-SGD, w/ ampl.	1000	1	0.37313	0.3460	22.50	22.22
Multi-epoch MF, no ampl.	1000	1	4.22468	0.2035	22.11	22.10
(Band) MF, w/ ampl. (Ours)	1000	1	0.79118	0.3460	23.11	22.83
DP-SGD, w/ ampl.	1000	2	0.30481	0.3460	22.89	22.62
Multi-epoch MF, no ampl.	1000	2	2.23048	0.3460	23.36	23.24
(Band) MF, w/ ampl. (Ours)	1000	2	0.64708	0.5882	24.01	23.71
DP-SGD, w/ ampl.	1000	4	0.25136	0.3460	23.27	22.94
Multi-epoch MF, no ampl.	1000	4	1.19352	0.5882	24.36	24.16
(Band) MF, w/ ampl. (Ours)	1000	4	0.52224	1.0000	24.67	24.42
DP-SGD, w/ ampl.	1000	8	0.20567	0.5882	23.59	23.30
Multi-epoch MF, no ampl.	1000	8	0.65294	1.0000	25.08	24.88
(Band) MF, w/ ampl. (Ours)	1000	8	0.43490	1.7000	25.26	24.99
DP-SGD, w/ ampl.	1000	16	0.16876	0.5882	23.96	23.61
Multi-epoch MF, no ampl.	1000	16	0.36861	1.7000	25.59	25.43
(Band) MF, w/ ampl. (Ours)	1000	16	0.36861	1.7000	25.59	25.43

Table 5: Parameters and metrics for Fig. 1(b). The noise multipliers are calibrated to achieve the given ϵ guarantees at $\delta=10^{-6}$ under $(k=6, b=342)$-participation, assuming Poisson sampling for DP-SGD and BANDMF.

		Eval Accuracy (%, Smoothed)													
	server lr η_s	0.0084	0.0143	0.0244	0.0414	0.0704	0.1197	0.2035	0.3460	0.5882	1.0000	1.7000	2.8900	4.9130	8.3521
ϵ	**Mechanism**														
1.0	**DP-SGD**	14.31	15.98	**16.82**	16.53	15.46	4.67	-	-	-	-	-	-	-	-
	Single-epoch MF	-	-	-	-	20.16	**20.29**	19.68	-	-	-	-	-	-	-
	Optimal TreeAgg	-	-	-	-	-	21.08	**21.15**	20.34	-	-	-	-	-	-
	Multi-epoch MF	-	-	-	-	-	21.56	**21.96**	21.60	-	-	-	-	-	-
	Band MF, b=342 **(Ours)**	-	-	-	-	-	21.70	**22.12**	21.96	-	-	-	-	-	-
2.0	**DP-SGD**	-	16.20	17.88	**18.70**	18.22	17.75	15.52	-	-	-	-	-	-	-
	Single-epoch MF	-	-	-	-	-	21.46	**21.66**	21.26	-	-	-	-	-	-
	Optimal TreeAgg	-	-	-	-	-	-	22.40	**22.52**	21.87	-	-	-	-	-
	Multi-epoch MF	-	-	-	-	-	-	22.80	**23.15**	22.96	-	-	-	-	-
	Band MF, b=342 **(Ours)**	-	-	-	-	-	-	-	23.27	**23.31**	22.57	-	-	-	-
4.0	**DP-SGD**	-	-	18.08	19.45	**20.07**	19.97	19.48	18.08	-	-	-	-	-	-
	Single-epoch MF	-	-	-	-	-	-	22.66	**22.94**	22.60	-	-	-	-	-
	Optimal TreeAgg	-	-	-	-	-	-	-	23.57	**23.66**	23.27	-	-	-	-
	Multi-epoch MF	-	-	-	-	-	-	-	23.87	**24.19**	24.01	-	-	-	-
	Band MF, b=342 **(Ours)**	-	-	-	-	-	-	-	-	24.26	**24.35**	23.74	-	-	-
8.0	**DP-SGD**	-	-	-	-	20.61	**21.26**	21.24	20.89	20.00	-	-	-	-	-
	Single-epoch MF	-	-	-	-	-	-	-	23.73	**24.03**	23.71	-	-	-	-
	Optimal TreeAgg	-	-	-	-	-	-	-	-	24.52	**24.54**	24.15	-	-	-
	Multi-epoch MF	-	-	-	-	-	-	-	-	24.72	**24.95**	24.77	24.17	-	-
	Band MF, b=342 **(Ours)**	-	-	-	-	-	-	-	-	24.76	**25.06**	24.92	-	-	-
16.0	**DP-SGD**	-	-	-	-	-	-	22.39	**22.51**	22.17	-	-	-	-	-
	Single-epoch MF	-	-	-	-	-	-	-	-	24.66	**24.80**	24.50	-	-	-
	Optimal TreeAgg	-	-	-	-	-	-	-	-	24.89	25.15	**25.15**	-	-	-
	Multi-epoch MF	-	-	-	-	-	-	-	-	-	25.38	**25.50**	25.34	-	-
	Band MF, b=342 **(Ours)**	-	-	-	-	-	-	-	-	-	25.38	**25.59**	25.47	-	-
inf	**DP-SGD**	-	-	-	-	-	-	-	-	-	-	25.96	26.23	**26.24**	8.03

Table 6: **Federated learning rate tuning for StackOverflow NWP.** Validation accuracy smoothed over the final 400 rounds of training, used to select the best server learning rates for the comparison of test-set accuracy presented in Fig. 5[a].

		Eval Accuracy (%, Smoothed)							
	server lr η_s	0.1197	0.2035	0.3460	0.5882	1.0000	1.7000	2.8900	4.9130
ϵ	**Mechanism**								
1.0	**DP-SGD**	-	22.39	**22.50**	22.03	-	-	-	-
	Multi-epoch MF	21.75	**22.11**	21.95	-	-	-	-	-
	Band MF, b=9 **(Ours)**	-	22.83	**23.11**	23.03	-	-	-	-
2.0	**DP-SGD**	-	22.70	**22.89**	22.66	-	-	-	-
	Multi-epoch MF	-	22.89	**23.36**	23.26	-	-	-	-
	Band MF, b=18 **(Ours)**	-	-	23.80	**24.01**	23.77	-	-	-
4.0	**DP-SGD**	-	22.88	**23.27**	23.20	-	-	-	-
	Multi-epoch MF	-	-	23.96	**24.36**	24.22	23.71	-	-
	Band MF, b=32 **(Ours)**	-	-	-	24.52	**24.67**	24.43	-	-
8.0	**DP-SGD**	-	-	23.48	**23.59**	23.28	-	-	-
	Multi-epoch MF	-	-	-	24.79	**25.08**	24.98	24.55	-
	Band MF, b=64 **(Ours)**	-	-	-	-	25.15	**25.26**	24.79	-
16.0	**DP-SGD**	-	-	23.85	**23.96**	23.72	-	-	-
	Multi-epoch MF	-	-	-	-	25.42	**25.59**	25.50	24.92
	Band MF, b=342 **(Ours)**	-	-	-	-	25.37	**25.55**	25.45	24.90
	Band MF, b=64 **(Ours)**	-	-	-	-	25.38	**25.54**	25.40	-

Table 7: **Centralized learning rate tuning for StackOverflow NWP..** Validation accuracy smoothed over the final 400 rounds of training, used to select the best server learning rates for the comparison of test-set accuracy presented in Fig. 1(b). DP-SGD and BANDMF use amplification.

Algorithm 7 Banded Matrix Multiplication

```
Input: $\hat{b}$-Banded lower triangular matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$, vector $\mathbf{x} \in \mathbb{R}^n$
Output: $\mathbf{Cx}$
for $i = 1, \ldots, n$ do
    $\mathbf{y}_i = \sum_{j=i-\hat{b}+1}^{i} \mathbf{C}_{[i,j]} \mathbf{x}_j$
return $\mathbf{y}$
```

Algorithm 8 Banded Inverse Multiplication

```
Input: $\hat{b}$-Banded lower triangular matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$, vector $\mathbf{y} \in \mathbb{R}^n$
Output: $\mathbf{C}^{-1}\mathbf{y}$
for $i = 1, \ldots, n$ do
    $\mathbf{x}_i = (\mathbf{y}_i - \sum_{j=i-\hat{b}+1}^{i-1} \mathbf{C}_{[i,j]} \mathbf{x}_j) / \mathbf{C}_{[i,i]}$
return $\mathbf{x}$
```

Figure 13: Algorithms for matrix-vector and inverse matrix-vector multiplication by a banded matrix. To simplify the presentation, we use the convention that out-of-bounds indexing into a matrix or vector returns 0.

I Efficient Multiplication and Inverse of Banded Matrices

Algorithms 7 and 8 (Fig. 13) give algorithms for lower triangular banded matrix-vector multiplication and inverse banded matrix-vector multiplication. Note that both algorithms are compatible with the streaming nature of gradients. As soon as the next input $\mathbf{x}_i$ is received, the algorithm can immediately output $\mathbf{y}_i$. Both algorithms require storing a state of size $\hat{b}$, and run in $O(n \cdot \hat{b})$ time. While the algorithms are described with respect to computing matrix-vector products, they can also be used to compute matrix-matrix products where the right-hand-side is a $n \times d$ matrix by multiplying by each column independently. In this setting, these algorithms require $O(\hat{b} \cdot d)$ space and $O(n \cdot \hat{b} \cdot d)$ time. Both algorithms have appeared previously in the literature on Monte Carlo methods, which have a similar problem at their core to that of noise generation for MF; see e.g. [51, Section 2].

J Application to a Real-World Cross-Device FL System

We train a one-layer LSTM language model of ∼2.4 million parameters in a practical cross-device FL system . The model is used for predicting the next word of Spanish in a mobile virtual keyboard. We pretrain the model on public multilingual C4 dataset [46, 53], and then fine-tune with on-device user data in FL. In a common practical FL system, clients have to satisfy criteria like being charged, idle and connected to unmetered network to participate in a round [10, 26, 30, 44], hence only a subset of clients can be reached and there is a strong diurnal pattern of client participation [54, 57]. It is very challenging to hold a fixed set of clients for evaluation, or develop random sampling for privacy amplification.

J.1 Reporting privacy guarantees

We follow the guidelines outlined in [45, Sec. 5.3] to report privacy guarantees.

1. **DP setting**. This a central DP guarantee where the service provider is trusted to correctly implement the mechanism.
2. **Instantiating the DP Definition**
   (a) *Data accesses covered*: The DP guarantee applies to all well-behaved clients [7] in a single training run. We do not account for hyperparameter tuning in our guarantees. Public multilingual C4 data [53] is used for pre-training.
   (b) *Final mechanism output*: Only the final model checkpoint is released for use in production, however the mechanism's output is technically the full sequence of privatized gradients, and so the guarantee also applies at this level, and hense all intermediate models are protected (including those sent to devices participating in federated learning).
   (c) *Unit of privacy*. Device-level DP is considered, i.e., the notion of adjacency is with respect to arbitrary training datasets on each client device, and the device might have an arbitrarily large local dataset containing arbitrary training examples. For user's with

[7] Clients that faithfully follow the algorithm including participation limits. Due to the design of the algorithm, a mis-behaved client does not adversely affect the DP guarantee of any well-behaved clients.

a single device, this corresponds directly to user-level DP; for devices shared with multiple users, this provides a stronger notion of DP than user-level; for a user with multiple devices that happen to both participate in training the model, the notion is weaker, but group privacy can be used to obtain a user-level guarantee.

   (d) *Adjacency definition for "neigbouring" datasets*: We use the zero-out definition [34]. This is a a special form of the add-or-remove definition, where neighboring data sets differ by addition/removal of a single client. In the absence of a client at any training step, we assume that the client's model update gets replaced with the all zeros vector. This assumption enforces a subtle modification to the traditional definition of the add/remove notion of DP which allows neighboring data sets to have the same number of records.

3. **Privacy accounting details**

   (a) *Type of accounting used*: Both $\rho-$zCDP [11] accounting, and PLD accounting [18] for $(\epsilon, \delta)-$DP are used.

   (b) *Accounting assumptions* : Each client only participates limited times during the training, and there are at least a min-separation of b rounds between two consecutive participation of a client. This is enforced by a timer on clients in the cross-device FL system.

   (c) *The formal DP statement*: The privacy guarantees are ρ=0.52-zCDP and (ϵ=$6.69, \delta$=10^{-10})-DP for ONLINE TREEAGG, while BANDMF achieves ρ=0.24-zCDP and (ϵ=$4.35, \delta$=10^{-10})-DP.

   (d) *Transparency and verifiability*: We are going to open source our code based on TensorFlow Federated and Tensorflow Privacy. Key portions of the cross-device FL system will also open sourced.