DYNAMIC WEIGHTED PROJECTION MAINTENANCE WITH ℓ_p-LEWIS WEIGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a new data–structure problem—*Dynamic ℓ_p-Lewis Weight Projection Maintenance*—that asks us to maintain the projection

$$P(W) = W^{1/2-1/p} A(A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p}$$

under a stream of diagonal weight updates and to support fast matrix–vector products with $P(W)$. This setting strictly generalizes the $\sqrt{W}A$ projection, which is at the heart of state-of-the-art linear programming and interior point methods, and it captures a wide range of algorithms that rely on leverage scores or Lewis weights for sampling and preconditioning. We provide a deterministic projection-maintenance data structure with sublinear amortized updates. Moreover, we extend it to the differential privacy setting.

1 INTRODUCTION

Projection maintenance is one of the most important data structure problems in modern convex optimization, serving as a critical component in achieving the best-known runtime guarantees for many cutting-edge algorithms (Lee et al., 2019; Jiang et al., 2020b;a; Cohen et al., 2021b; Huang et al., 2022). We first recall the definition of classical Dynamic Projection Maintenance Problem.

Definition 1.1 (Dynamic Projection Maintenance (Cohen et al., 2021b)). *Given a matrix $B \in \mathbb{R}^{m\times n}$, the goal is to design a data structure to maintain the projection matrix $P(B) := B(B^\top B)^{-1}B^\top$ and support the fast multiplication of $P(B) \cdot h$ for any query $h \in \mathbb{R}^n$ with the following operations:*

- INIT($B \in \mathbb{R}^{m\times n}$)*: The data structure takes the matrix B as input, and does some preprocessing and compute an initial projection.*
- UPDATE($B^{\mathrm{new}} \in \mathbb{R}^{m\times n}$)*: The data structure receives some low-rank or sparse update B^{new} and updates B by $B + B^{\mathrm{new}}$.*
- QUERY($h \in \mathbb{R}^n$)*: The data structure receives a vector h and approximately computes the matrix-vector product of updated projection matrix $P(B)$ and an online vector h.*

For example, in linear programming (Cohen et al., 2021b), we take $B = \sqrt{W}A$, where A is the constraint matrix and W is a diagonal matrix representing slack variables. In each iteration, W undergoes relatively small perturbations. The goal of the data structure is to efficiently approximate

$$\sqrt{W}A(A^\top WA)^{-1}A^\top\sqrt{W}h$$

for an online vector $h \in \mathbb{R}^n$.

In this work, we consider a specific projection maintenance problem with $B = W^{1/2-1/p}A$ which generalizes the above problem. We need to maintain the projection and compute an approximation of matrix-vector product between the projection matrix and any online vector $h \in \mathbb{R}^n$. We call this problem, for maintaining such kind of matrices, the Dynamic ℓ_p-Lewis Weight Projection Maintenance. Formally, it is defined as follows.

Definition 1.2 (Dynamic ℓ_p-Lewis Weight Projection Maintenance)**.** *Given $p > 0$, a matrix $A \in \mathbb{R}^{m \times n}$ and a diagonal matrix $W \in \mathbb{R}^{m \times m}$ with nonnegative entries, the goal is to design a data structure to maintain the projection matrix*

$$P(W) := W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p}$$

and support the fast multiplication of $P(W) \cdot h$ for any query $h \in \mathbb{R}^n$ with the following operations:

- INIT($A \in \mathbb{R}^{m \times n}, W \in \mathbb{R}^{m \times m}$)*: The data structure takes a matrix $A \in \mathbb{R}^{m \times n}$ and a diagonal matrix $W \in \mathbb{R}^{m \times m}$ with nonnegative entries as input, and does some preprocessing and compute an initial projection.*
- UPDATE($W^{\mathrm{new}} \in \mathbb{R}^{m \times m}$)*: The data structure receives some low-rank or sparse update W^{new} and updates W by $W + W^{\mathrm{new}}$.*
- QUERY($h \in \mathbb{R}^n$)*: The data structure receives a vector h and approximately computes the matrix-vector product of updated projection matrix $P(W)$ and an online vector h.*

This problem is fundamental to the design of efficient algorithms in settings where leverage scores or Lewis weights determine adaptive sampling (Cohen & Peng, 2015; Parulekar et al., 2021; Brand et al., 2021a; Woodruff & Yasuda, 2023) or preconditioning (Durfee et al., 2018; Yang et al., 2018; Kyng et al., 2019).

Roadmap. In Section 2, we review relevant literature related to our study. In Section 3, we present serveral useful tools and provide the main result. In Section 4, we state the main result of this paper. In Section 5, we provide technical overview of our study. In Section 6, we draw our conclusion.

2 Related Work

2.1 Linear Programming and Semidefinite Programming

Linear programming is a cornerstone of optimization and theoretical computer science. Dantzig's Simplex algorithm (Dantzig, 1951) remains a practical workhorse despite its exponential worst-case complexity. The Ellipsoid Method later gave the first polynomial-time guarantee for linear programming, although it is typically slower in practice than Simplex. A decisive breakthrough came with Karmarkar's interior-point method (Karmarkar, 1984), which combines polynomial running time with strong empirical performance and has sparked extensive work on ever-faster interior-point techniques for a broad range of optimization problems (Vaidya, 1987; Renegar, 1988; Vaidya, 1989; Daitch & Spielman, 2008; Lee & Sidford, 2013; 2014; 2019; Cohen et al., 2021a; Lee et al., 2019; Brand, 2020; Brand et al., 2020; Jiang et al., 2021; Song & Yu, 2021; Gu & Song, 2022).

Beyond their algorithmic importance, linear programming and semidefinite programming are ubiquitous in machine-learning theory. They underpin efficient formulations for empirical risk minimization (Lee et al., 2019; Song et al., 2022b; Qin et al., 2023), support vector machines (Gu et al., 2023; Gao et al., 2023a), and numerous other learning problems, providing both rigorous guarantees and scalable implementations.

2.2 Sketching

Sketching—compressing data with random linear maps—has become a workhorse across modern optimization and numerical linear algebra. It powers cutting-edge algorithms for linear programming (Jiang et al., 2021; Song & Yu, 2021), empirical risk minimization (Lee et al., 2019; Qin et al., 2023), and semidefinite programming (Jiang et al., 2020a; Huang et al., 2022; Song et al., 2023c). In randomized numerical linear algebra it accelerates a wide range of matrix tasks and decompositions (Clarkson & Woodruff, 2017; Nelson & Nguyên, 2013; Boutsidis et al., 2016; Razenshteyn et al., 2016; Song et al., 2017; Xiao et al., 2018; Song et al., 2019; Lee et al., 2019; Jiang et al., 2021; Song & Yu, 2021; Brand et al., 2021b; Hu et al., 2022; Song et al., 2022a; Gu & Song, 2022). Most applications employ *oblivious* sketches—data-independent projections—for dimensionality reduction (Clarkson & Woodruff, 2017; Nelson & Nguyên, 2013). For approximate John-ellipsoid

computation, Chen et al.(Cohen et al., 2019) rely exclusively on sketching, suggesting room for further acceleration, while Mahabadi et al.(Makarychev et al., 2022) tackle a tougher streaming variant for convex polytopes; their method, however, is not yet optimal in our computational model.

2.3 Differential Privacy

Introduced by (Dwork et al., 2006), differential privacy (DP) has become the gold standard for rigorous data protection. An extensive body of work now retrofits classical algorithms, data structures, and machine-learning pipelines with provable DP guarantees (Esfandiari et al., 2022; Andoni et al., 2023; Cherapanamjeri et al., 2023; Cohen-Addad et al., 2022; Dong et al., 2024; Farhadi et al., 2022; Gopi et al., 2023; Li et al., 2022; Gopi et al., 2022; Huang & Yi, 2021; Jung et al., 2019; Li & Li, 2024; Epasto et al., 2024; Chen et al., 2022; Beimel et al., 2022; Narayanan, 2022; 2023; Fan & Li, 2022; Fan et al., 2024; Li & Li, 2023; Eliáš et al., 2020; Yu et al., 2024; Liang et al., 2024; Gu et al., 2024; Song et al., 2023b; Qin et al., 2022; Song et al., 2023a; Galli et al., 2024; Chen et al., 2024; Romijnders et al., 2024; Qi et al., 2024; Ke et al., 2025; Hu et al., 2024; Liu et al., 2024). Beyond integrating privacy into existing methods, researchers are refining the fundamental DP building blocks themselves. Enhanced variants of the Gaussian, Exponential, and Laplace mechanisms now deliver tighter accuracy–privacy trade-offs than the classical formulations (Dwork et al., 2014). A prime example is the truncated Laplace mechanism of Gopi et al. (Geng et al., 2020), which currently achieves the smallest known error for any (ϵ, δ)-DP distribution.

3 Preliminary

Fact 3.1 ((Woodbury, 1950)). *The Woodbury matrix identity is*

$$(M + UCV)^{-1} = M^{-1} - M^{-1}U(C^{-1} + VM^{-1}U)^{-1}VM^{-1}.$$

Let $S \subset [n]$ denote the set of coordinates that is changed by more than a constant factor and $r = |S|$. Using the identity above, we have that

$$M_{w^{\mathrm{new}}} = M_w - (M_w)_S(\Delta_{S,S}^{-1} + (M_w)_{S,S})^{-1}((M_w)_S)^\top,$$

where $\Delta = \mathrm{diag}(w^{\mathrm{new}} - w)$, $(M_w)_S \in \mathbb{R}^{n \times r}$ is the r columns from S of M_w and $(M_w)_{S,S}, \Delta_{S,S} \in \mathbb{R}^{r \times r}$ are the r rows and columns from S of M_w and Δ.

Fact 3.2. *We have*

- *Let $A \in \mathbb{R}^{n \times n}$, then we have $\|A\|_F \leq \sqrt{n}\|A\|$.*
- *Let $A \in \mathbb{R}^{n \times n}$, then we have $\|A\| \leq \|A\|_F$*
- *For two vectors $a, b \in \mathbb{R}^n$, then we have $|ab^\top| \leq \|a\|_2 \cdot \|b\|_2$*

Definition 3.3 (Differential Privacy, (Dwork et al., 2014)). *For $\epsilon > 0, \delta \geq 0$, a randomized function $\mathcal{A}$ is (ϵ, δ)-differentially private ((ϵ, δ)-DP) if for any two neighboring datasets $X \sim X'$, and any possible outcome of the algorithm $S \subset$ in Range($\mathcal{A}$), $\Pr[\mathcal{A}(X) \in S] \leq e^\epsilon \Pr[\mathcal{A}(X') \in S] + \delta$.*

Lemma 3.4 (Truncated Laplace Mechanism, (Dwork et al., 2014; Geng et al., 2020; Andoni et al., 2023)). *Let $\mathrm{Lap}(\lambda)$ denote the Laplace distribution with parameter λ with PDF $\Pr[Z = z] = \frac{1}{2\lambda}e^{-|z|/\lambda}$. Let $B_L := (\Delta/\epsilon)\log(1 + \frac{e^\epsilon - 1}{2\delta})$. Let $\mathrm{TLap}(\Delta, \epsilon, \delta)$ denote the Truncated Laplace distribution with PDF proportional to $e^{-|z|/\lambda}$ on the region $[-B_L, B_L]$. Given a numeric function f that takes a dataset X as the input, and has sensitivity Δ, the mechanism output $f(X) + Z$ where $Z \sim \mathrm{Lap}(\Delta/\epsilon)$ is $(\epsilon, 0)$-DP. In addition, if $Z \sim \mathrm{TLap}(\Delta, \epsilon, \delta)$, then $f(X) + Z$ is (ϵ, δ)-DP.*

Definition 3.5 (Dataset, (Gao et al., 2023b)). *Fix $\eta > 0, \alpha > 0$. We say our dataset $X \in \mathbb{R}^{n \times d}$ is (α, η)-good if $XX^\top \succeq \eta \cdot I_n$ and for all $i \in [d]$, $\|X_{*,i}\|_2 \leq \alpha$.*

Definition 3.6 (β-close neighbor dataset, (Gao et al., 2023b)). *Let $B > 0$ be a constant. Let n be the number of data points. Let dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $\|x_i\|_2 \leq B$ for any $i \in [n]$. We define $\mathcal{D}'$ as a neighbor dataset with one data point replacement of $\mathcal{D}$. Without loss of generality, we have $'\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n-1} \cup \{(x'_n, y_n)\}$.Namely, we have $\mathcal{D}$ and $\mathcal{D}'$ only differ in the n-th item.*

Additionally, we assume that x_n and x'_n are β-close. Namely, we have

$$\|x_n - x'_n\|_2 \leq \beta.$$

Lemma 3.7 (Post-Processing Lemma for DP, (Dwork et al., 2014)). *Let $\mathcal{M} := \mathbb{N}^{|\chi|} \to \mathbb{R}$ be a randomized algorithm that is (ϵ, δ)-differentially private. Let $f : \mathbb{R} \to \mathbb{R}'$ be an arbitrarily random mapping. Then is $f \circ \mathcal{M} : \mathbb{N}^{|\chi|} \to \mathbb{R}'$ (ϵ, δ)-differentially private.*

Theorem 3.8 (Empirical covariance estimator for Gaussian (Vershynin, 2018)). *Let $\Sigma \in \mathbb{R}^{d \times d}$ be PSD, $X_1, \cdots, X_n \sim \mathcal{N}(0, \Sigma)$ be i.i.d and $\widetilde{\Sigma} = \frac{1}{n} \sum_{i=1}^n X_i X_i^\top$. Then with probability $1 - \gamma$, it holds that $\|\Sigma^{-1/2} \widetilde{\Sigma} \Sigma^{-1/2} - I\|_F \leq \rho$ for some $\rho = O(\sqrt{\frac{d^2 + \log(1/\gamma)}{n}} + \frac{d^2 + \log(1/\gamma)}{n})$.*

Lemma 3.9 (Composition lemma for DP, (Dwork et al., 2014)). *Let $\mathcal{M} := \mathbb{N}^{|\chi|} \to \mathbb{R}$ be an (ϵ_i, δ_i)-DP algorithm for $i \in [k]$. If $\mathcal{M}_{[k]} \to \Pi_{i=1}^n \mathcal{R}_i$ satisfies $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \cdots, \mathcal{M}_k(x))$, then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP.*

Lemma 3.10 (Lemma C.15 in (Song & Yu, 2021)). *Let $x^{\mathrm{new}} = x + \widetilde{\delta}_x$ and $s^{\mathrm{new}} = s + \widetilde{\delta}_s$. Let $w = \frac{x}{s}$ and $w^{\mathrm{new}} = \frac{x^{\mathrm{new}}}{s^{\mathrm{new}}}$. Then we have $\sum_{i=1}^n (\mathbb{E}[\ln w_i^{\mathrm{new}}] - \ln w_i)^2 \leq 64\epsilon^2, \sum_{i=1}^n (\mathrm{Var}[\ln w_i^{\mathrm{new}}])^2 \leq 1000\epsilon^2$.*

4 Main Result

The goal of this section is to prove the following theorem:

Algorithm 1 Projection Maintenance Data Structure

datastructure MaintainProjection

members
- $w \in \mathbb{R}^n$
- $v, \widetilde{v} \in \mathbb{R}^n$
- $A \in \mathbb{R}^{d \times n}$
- $M \in \mathbb{R}^{n \times n}$
- $Q \in \mathbb{R}^{n \times n^b L}$
- $R_{1,*}, R_{2,*}, \cdots, R_{L,*} \in \mathbb{R}^{n^b \times n}$
- $l \in \mathbb{N}_+, L \in \mathbb{N}_+$
- $\epsilon_{\mathrm{mp}} \in (0, 1/4)$
- $a \in (0, \alpha]$

end members

procedure Initialize($A, w, \epsilon_{\mathrm{mp}}, a$)
- $w \leftarrow w, v \leftarrow w, \epsilon_{\mathrm{mp}} \leftarrow \epsilon_{\mathrm{mp}}, A \leftarrow A, a \leftarrow a$
- $M \leftarrow A^\top (A V^{1-2/p} A^\top)^{-1} A$
- Choosing $R_{1,*}, R_{2,*}, \cdots, R_{L,*} \in \mathbb{R}^{n^b \times n}$ to be sketching matrices
- $R \leftarrow [R_{*,1}, R_{*,2}, \cdots, R_{*,L}]$
- $Q \leftarrow M V^{1/2-1/p} R^\top$
- $l \leftarrow 1$

end procedure

end datastructure

Theorem 4.1 (Projection maintenance). *Given a full rank matrix $A \in \mathbb{R}^{d \times n}$ with $n \geq d$ and a tolerance parameter $0 < \epsilon_{\mathrm{mp}} < 1/4$. Given any positive number a such that $a \leq \alpha$ where α is the dual exponent of matrix multiplication. Let $R_{1,*}, \cdots, R_{L,*} \in \mathbb{R}^{n^b \times n}$ denote a list of sketching matrices, where $b \in [0, 1]$. There is a deterministic data structure (Algorithm 1) that approximately maintains the projection matrices*

$$W^{1/2-1/p} A^\top (A W^{1-2/p} A^\top)^{-1} A W^{1/2-1/p}$$

for positive diagonal matrices W through the following two operations:

1. Update(w): *Output a vector $\widetilde{v}$ such that for all $i \in [n]$,*

$$(1 - \epsilon_{\mathrm{mp}}) \widetilde{v}_i^{1/2-1/p} \leq w_i^{1/2-1/p} \leq (1 + \epsilon_{\mathrm{mp}}) \widetilde{v}_i^{1/2-1/p}.$$

2. QUERY(h): *Output* $\widetilde{V}^{1/2-1/p}A^\top(A\widetilde{V}^{1-2/p}A^\top)^{-1}A\widetilde{V}^{1/2-1/p}(R^\top)_{*,l}R_{l,*}h$ *for the* $\widetilde{v}$ *outputted by the last call to* UPDATE.

The data structure takes $n^2d^{\omega-2}$ *time to initialize and each call of* QUERY(h) *takes time* $n^{1+b+o(1)}+n^{1+a+o(1)}$.

Furthermore, if the initial vector $w^{(0)}$ *and the (random) update sequence* $w^{(1)},\cdots,w^{(T)}\in\mathbb{R}^n$ *satisfies*

$$\sum_{i=1}^n\left((1/2-1/p)\cdot(\mathbb{E}[\ln w_i^{(k+1)}]-\ln w_i^{(k)})\right)^2\leq C_1^2 \text{ and } \sum_{i=1}^n \mathrm{Var}[(1/2-1/p)\ln w_i^{(k+1)}])^2\leq C_2^2$$

with the expectation and variance is conditional on $w_i^{(k)}$ *for all* $k=0,1,\cdots,T-1$. *Then, the amortized expected time*[1] *per call of* UPDATE(w) *is* $(C_1/\epsilon_{\mathrm{mp}}+C_2\epsilon_{\mathrm{mp}}^2)\cdot(n^{\omega-1/2+o(1)}+n^{2-a/2+o(1)})$.

Proof. The theorem holds by combining Lemma 4.3, Lemma 4.4 and Lemma 4.5. □

Remark 4.2. *For our linear program algorithm, we have* $C_1=O(1/\log n)$, $C_2=O(1/\log n)$ *and* $\epsilon_{\mathrm{mp}}=\Theta(1)$. *See Lemma 3.10.*

To verify the correctness of our updates, we have the following lemma:

Lemma 4.3 (Correctness of the algorithm, informal version of Lemma C.1)**.** *The output of* UPDATE(w) *in Algorithm 2 satisfies*

$$M=A^\top(AV^{1-2/p}A^\top)^{-1}A,\ Q=MV^{1/2-1/p}R^\top$$

The output of QUERY(h) *in Algorithm 3 satisfies*

$$p_s=\widetilde{P}(R^\top)_{*,l}R_{l,*}h$$
$$p_x=(I-\widetilde{P})(R^\top)_{*,l}R_{l,*}h,$$

where $\widetilde{P}=V^{1/2-1/p}A^\top(A\widetilde{V}^{1-2/p}A^\top)^{-1}A\widetilde{V}^{1/2-1/p}$, *and* $\widetilde{V}$ *is outputted by* UPDATE(w).

Above lemma verifies our algorithm. Now we consider the running time of the projection maintenance, which consists of Initialization time, update time and query time, as discussed below.

4.1 INITIALIZATION TIME, UPDATE TIME

To formalize the amortized runtime proof, we first analyze the initialization time (Lemma 4.4), update time (Lemma 4.5), and query time (Lemma 4.6) of our projection maintenance data-structure.

Lemma 4.4 (Initialization time)**.** *The initialization time of data-structure* MAINTAINPROJECTION *(Algorithm 1) is* $O(n^2d^{\omega-2})$.

Proof. Given a matrix $A\in\mathbb{R}^{d\times n}$ and diagonal matrix $V\in\mathbb{R}^{n\times n}$, computing $A^\top(AVA^\top)^{-1}A$ takes $O(n^2d^{\omega-2})$. □

Lemma 4.5 (Update time)**.** *The update time of data-structure* MAINTAINPROJECTION *(Algorithm 2) is* $O(rg_rn^{2+o(1)})$ *where* r *is the number of indices we updated in* v.

Proof. The proof is identical to (Cohen et al., 2021b; Lee et al., 2019). We omit the details here. □

4.2 QUERY TIME

Lemma 4.6 (Query time, informal version of Lemma C.2)**.** *The query time of data-structure* MAINTAINPROJECTION *(Algorithm 1) is* $O(n^{1+b+o(1)}+n^{1+a+o(1)})$.

[1] If the input is deterministic, so is the output and the runtime.

Algorithm 2 Update

datastructure MAINTAINPROJECTION

procedure UPDATE(w)
  $y_i \leftarrow \ln w_i - \ln v_i, \forall i \in [n]$
  $r \leftarrow$ the number of indices i such that $|y_i| \geq \epsilon_{\mathrm{mp}}/2$.
  if $r < n^a$ **then**
    $v^{\mathrm{new}} \leftarrow v$
    $M^{\mathrm{new}} \leftarrow M$
    $l \leftarrow l + 1$
  else
    Let $\pi : [n] \to [n]$ be a sorting permutation such that $|y_{\pi(i)}| \geq |y_{\pi(i+1)}|$
    while $1.5 \cdot r < n$ and $|y_{\pi([1.5 \cdot r])}| \geq (1 - 1/\log n)|y_{\pi(r)}|$ **do**
      $r \leftarrow \min(\lceil 1.5 \cdot r \rceil, n)$
    end while
    $v^{\mathrm{new}}_{\pi(i)} \leftarrow \begin{cases} w_{\pi(i)} & i \in \{1, 2, \cdots, r\} \\ v_{\pi(i)} & i \in \{r+1, \cdots, n\} \end{cases}$
    $\Delta \leftarrow \mathrm{diag}(v^{\mathrm{new}} - v)$
    $\Gamma \leftarrow \mathrm{diag}((v^{\mathrm{new}})^{1/2-1/p} - v^{1/2-1/p})$
    Let $S \leftarrow \pi([r])$ be the first r indices in the permutation.
    Let $M_S \in \mathbb{R}^{n \times r}$ be the r columns from S of M.
    Let $M_{S,S}, \Delta_{S,S} \in \mathbb{R}^{r \times r}$ be the r rows and columns from S of M and Δ.
    $M^{\mathrm{new}} \leftarrow M - M_{*,S} \cdot (\Delta^{-1}_{S,S} + M_{S,S})^{-1} \cdot (M_{*,S})^\top$
    Re-generate R
    $Q^{\mathrm{new}} \leftarrow Q + (M^{\mathrm{new}} \cdot \Gamma) \cdot R^\top + (M^{\mathrm{new}} - M) \cdot V^{1/2-1/p} \cdot R^\top$
    $l \leftarrow 1$
  end if
  $v \leftarrow v^{\mathrm{new}}$
  $M \leftarrow M^{\mathrm{new}}$
  $Q \leftarrow Q^{\mathrm{new}}$
  $\widetilde{v}_i \leftarrow \begin{cases} v_i & \text{if } |\ln w_i - \ln v_i| < \epsilon_{\mathrm{mp}}/2 \\ w_i & \text{otherwise} \end{cases}$
  return $\widetilde{v}$
end procedure

end datastructure

Algorithm 3 Query

datastructure MAINTAINPROJECTION

procedure QUERY(h)
  Let $\widetilde{S}$ be the indices i such that $|\ln w_i - \ln v_i| \geq \epsilon_{\mathrm{mp}}/2$.
  $\widetilde{\Delta} \leftarrow \widetilde{V}^{1-2/p} - V^{1-2/p}$
  $\widetilde{\Gamma} \leftarrow \widetilde{V}^{1/2-1/p} - V^{1/2-1/p}$
  $p_m \leftarrow \widetilde{V}^{1/2-1/p} \cdot (M_{*,\widetilde{S}}) \cdot (\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + M_{\widetilde{S},\widetilde{S}})^{-1} \cdot (Q_{\widetilde{S},l} + M_{\widetilde{S},*} \cdot \widetilde{\Gamma} \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h$
  $p_s \leftarrow \widetilde{V}^{1/2-1/p} \cdot (Q_{*,l} + M \cdot \widetilde{\Gamma} \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h - p_m$
  $p_x \leftarrow (R^\top)_{*,l} \cdot R_{l,*} \cdot h - p_s$
  return (p_x, p_s)
end procedure

end datastructure

5 TECHNICAL OVERVIEW

In this section, we present technical overview of our study. In Section 5.1, we introduce the key parameters for privacy analysis. In Section 5.2, we analyze the DP guarantees for $W^{1/2-1/p}A$, while Section 5.3 investigates its utility guarantees. In Section 5.4, we present DP guarantees for $A^\top W^{1/2-1/p}$. In Section 5.5, we present utility guarantees for $A^\top W^{1/2-1/p}$. In Section 5.6, we provide DP guarantees for $(A^\top W^{1-2/p}A)^{-1}$. In Section 5.7, we provide utility guarantees for $(A^\top W^{1-2/p}A)^{-1}$.

5.1 KEY CONCEPTS

Definition 5.1 (Definition of M, (Gao et al., 2023b), see Definition 5.1 of (Gu et al., 2025) as an example)**.** *Let $\mathcal{M} : (\mathbb{R}^n)^d \to \mathbb{R}^{n\times n}$ be a (randomized) algorithm that given a dataset of d points in $\mathbb{R}^n$ outputs a PSD matrix. Let $\mathcal{Y}, \mathcal{Y}' \in (\mathbb{R}^n)^d$. Then, we define*

$$M := \|\mathcal{M}(\mathcal{Y})^{1/2}\mathcal{M}(\mathcal{Y}')^{-1}\mathcal{M}(\mathcal{Y})^{1/2} - I\|_F.$$

Definition 5.2 (Definition of Δ, (Gao et al., 2023b), see Definition 5.2 of (Gu et al., 2025) as an example)**.** *If we have the following conditions:*

- *Let $\epsilon \in (0,1)$ and $\delta \in (0,1)$.*
- *Let k denote the number of i.i.d. samples $g_1, g_2, \cdots, g_k$ from $\mathcal{N}(0,\Sigma_1)$ output by Algorithm 4.*

We define $\Delta := \min\left\{\frac{\epsilon}{\sqrt{8k\log(1/\delta)}}, \frac{\epsilon}{8\log(1/\delta)}\right\}$.

5.2 DP GUARANTEES FOR $W^{1/2-1/p}A$

Lemma 5.3 (Sensitivity of $W^{1/2-1/p}A$, informal version of Lemma D.1)**.** *If the following conditions hold:*

- *Let the neighboring dataset X and X' be defined in Definition 3.6.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for J.*
- *Let $J := W^{1/2-1/p}A$ denote the data generated by X and J' denote the data generated by neighboring dataset X', where $W^{1/2-1/p} \in \mathbb{R}^{m\times m}$ and $A \in \mathbb{R}^{m\times n}$.*
- *Let $\beta > 0$ be defined as Definition 3.6.*

Then, we can show that the sensitivity of J is $\sqrt{n}\cdot\beta$.

Then, we use the truncated Laplace mechanism (Lemma 3.4) to ensure the DP property of $W^{1/2-1/p}A$.

Lemma 5.4 (DP guarantees for $W^{1/2-1/p}A$)**.** *If the following conditions hold:*

- *Let the neighboring dataset X and X' be defined in Definition 3.6.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for J.*
- *Let $\Delta_J := \sqrt{n}\cdot\beta$ denote the sensitivity of J.*
- *Let $J := W^{1/2-1/p}A$ denote the data generated by X and J' denote the data generated by neighboring dataset X', where $W^{1/2-1/p} \in \mathbb{R}^{m\times m}$ and $A \in \mathbb{R}^{m\times n}$.*
- *Let $\beta > 0$ be defined as Definition 3.6.*
- *Let $B_L = (\Delta_J/\epsilon_J)\log(1 + \frac{\exp(\epsilon_J)-1}{2\delta_J})$.*

- *Let* $\widetilde{J} := J + \mathrm{TLap}(\Delta_J, \epsilon_J, \delta_J)$.

Then, we can show that $\widetilde{J}$ *is* (ϵ_J, δ_J)*-DP.*

Proof. The proof follows directly from Lemma 3.4. □

5.3 Utility Guarantees for $W^{1/2-1/p}A$

Lemma 5.5 (Utility guarantees for $W^{1/2-1/p}A$, informal version of Lemma E.1)**.** *Under the same conditions in Lemma 5.4, we can show that* $\|\widetilde{J} - J\|_2 \leq \sqrt{n} \cdot B_L$.

5.4 DP Guarantees for $A^\top W^{1/2-1/p}$

Lemma 5.6 (DP guarantees for $A^\top W^{1/2-1/p}$)**.** *If the following conditions hold:*

- *Let the neighboring dataset* X *and* X' *be defined in Definition 3.6.*
- *Let* $\epsilon_J, \delta_J \in \mathbb{R}$ *denote the DP parameters for* J.
- *Let* $\Delta_J := \sqrt{n} \cdot \beta$ *denote the sensitivity of* J.
- *Let* $J^\top := A^\top W^{1/2-1/p}$ *denote the data generated by* X *and* $J'^\top$ *denote the data generated by neighboring dataset* X'*, where* $W^{1/2-1/p} \in \mathbb{R}^{m \times m}$ *and* $A^\top \in \mathbb{R}^{n \times m}$.
- *Let* $\beta > 0$ *be defined as Definition 3.6.*
- *Let* $B_L = (\Delta_J/\epsilon_J) \log(1 + \frac{\exp(\epsilon_J)-1}{2\delta_J})$.
- *Let* $\widetilde{J}^\top := J^\top + \mathrm{TLap}(\Delta_J, \epsilon_J, \delta_J)$.

Then, we can show that $\widetilde{J}^\top$ *is* (ϵ_J, δ_J)*-DP.*

Proof. In Lemma 5.4, we prove the differential privacy property of $W^{1/2-1/p}A$. By the post-processing property of differential privacy (Lemma 3.7), the transpose matrix $A^\top W^{1/2-1/p}$ computed from the privatized matrix $W^{1/2-1/p}A$ also preserves (ϵ_J, δ_J)-differentially private. □

5.5 Utility Guarantees for $A^\top W^{1/2-1/p}$

Lemma 5.7 (Utility guarantees for $A^\top W^{1/2-1/p}$, informal version of Lemma E.2)**.** *Under the same conditions in Lemma 5.6,*

we can show that $\|\widetilde{J}^\top - J^\top\|_2 \leq \sqrt{n} \cdot B_L$.

5.6 DP Guarantees for $(A^\top W^{1-2/p}A)^{-1}$

Lemma 5.8 (DP guarantees for $(A^\top W^{1-2/p}A)^{-1}$, Theorem 6.12 in (Gao et al., 2023b), Theorem 5.1 in (Alabi et al., 2023), Lemma 5.4 in (Gu et al., 2025), informal version of Lemma B.1)**.** *Under the same conditions in Lemma B.1, there exists an Algorithm 4 such that*

- *Part 1. Algorithm 4 is* $(\epsilon_\alpha, \delta_\alpha)$*-DP.*
- *Part 2. Outputs* $\widehat{\Sigma} \in \mathbb{S}_+^n$ *denotes the private version of input* Σ*, such that with probabilities at least* $1 - \gamma$*,* $\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I_n\|_F \leq \rho$.
- *Part 3.* $(1 - \rho)\Sigma \preceq \widehat{\Sigma} \preceq (1 + \rho)\Sigma$.

By the post-processing property of differential privacy (Lemma 3.7), the inverse $\widehat{\Sigma}^{-1}$ *computed from the privatized matrix* $\widehat{\Sigma}$ *remains* $(\epsilon_\alpha, \delta_\alpha)$*-differentially private.*

In Lemma 5.8, **Part 1** claims the privacy guarantees of the "Gaussian Sampling Mechanism", **Part 2** establishes the critical properties necessary to ensure the utility of the "Gaussian Sampling Mechanism", and **Part 3** presents the ultimate utility outcomes of the algorithm.

Note that in our setting, we use $\Sigma = H$, where H is the non-private matrix of interest, and we also have $\widehat{\Sigma} = \widetilde{H}$ to denote the private version of H.

The quantity M used in **Condition 6** is formally analyzed in Section A.1.

Algorithm 4 The Gaussian Sampling Mechanism, (Gao et al., 2023b)

procedure ALGORITHM(Σ, k)
  PSD matrix $\Sigma \in \mathbb{R}^{n \times n}$ and parameter $k \in \mathbb{N}$
  Obtain vectors $g_1, g_2, \cdots, g_k$ by sampling $g_i \sim \mathcal{N}(0, \Sigma)$, independently for each $i \in [k]$
  Compute $\widehat{\Sigma} = \frac{1}{k} \sum_{i=1}^{k} g_i g_i^\top$ ▷ Covariance estimate.
  return $\widehat{\Sigma}$
end procedure

5.7 Utility Guarantees for $(A^\top W^{1-2/p} A)^{-1}$

Lemma 5.9 (Utility guarantees for $(A^\top W^{1-2/p} A)^{-1}$, informal version of Lemma E.3)**.** *Under the same conditions in Lemma 5.8, with probability $1 - \gamma$, we have $\|H^{-1} - \widetilde{H}^{-1}\| \leq O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2})$.*

5.8 DP Guarantees for $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$

Lemma 5.10 (DP guarantees for $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$, informal version of Lemma D.2)**.** *If the following conditions hold:*

- *Let $\epsilon_\alpha, \delta_\alpha \in \mathbb{R}$ denote the DP parameter for $A^\top W^{1-2/p} A$.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for $W^{1/2-1/p} A$ and $A^\top W^{1/2-1/p}$.*
- *Let $\epsilon = 2\epsilon_J + \epsilon_\alpha,\ \delta = 2\delta_J + \delta_\alpha$.*
- *Let H and $\widetilde{H}$ be defined as Lemma 5.9.*
- *Let J and $\widetilde{J}$ be defined as Lemma 5.4.*

Then, we can show $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$ is (ϵ, δ)-DP.

5.9 Utility Guarantees for $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$

Lemma 5.11 (Utility guarantees for $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$, informal version of Lemma E.4)**.** *Under the same conditions as in Lemma E.4, with probability $1 - \gamma$, we have*

$$|JH^{-1}J^\top \cdot h - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top \cdot h| \leq 2\sigma_J \sigma_h \sqrt{n} \cdot B_L + \sigma_J^2 \sigma_h \cdot O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}).$$

6 Conclusion

In this work, we introduce *Dynamic ℓ_p-Lewis Weight Projection Maintenance*, which is a novel data-structure that considers the projection maintenance problem $P(B) := B(B^\top B)^{-1} B^\top$ with $B = W^{1/2-1/p} A$, that strictly generalizes the $B = \sqrt{W} A$ projection. Our deterministic algorithm supports fast updates and queries with sublinear amortized time and extends naturally to the differential privacy setting with provable utility guarantees. This work not only advances theoretical tools for linear programming, interior point methods, and leverage-based algorithms, but also opens avenues for private and efficient optimization in data-sensitive applications.

ETHIC STATEMENT

This paper does not involve human subjects, personally identifiable data, or sensitive applications. We do not foresee direct ethical risks. We follow the ICLR Code of Ethics and affirm that all aspects of this research comply with the principles of fairness, transparency, and integrity.

REPRODUCIBILITY STATEMENT

We ensure reproducibility of our theoretical results by including all formal assumptions, definitions, and complete proofs in the appendix. The main text states each theorem clearly and refers to the detailed proofs. No external data or software is required.

REFERENCES

Daniel Alabi, Pravesh K Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. Privately estimating a gaussian: Efficient, robust, and optimal. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 483–496, 2023.

Alexandr Andoni, Piotr Indyk, Sepideh Mahabadi, and Shyam Narayanan. Differentially private approximate near neighbor counting in high dimensions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 43544–43562, 2023.

Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, and Uri Stemmer. Dynamic algorithms against an adaptive adversary: generic constructions and lower bounds. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1671–1684, 2022.

Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 236–249, 2016.

Jan van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 259–278. SIAM, 2020.

Jan van den Brand, Yin Tat Lee, Aaron Sidford, and Zhao Song. Solving tall dense linear programs in nearly linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 775–788, 2020.

Jan van den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 859–869, 2021a.

Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (overparametrized) neural networks in near-linear time. In *ITCS*, 2021b.

E Chen, Yang Cao, and Yifei Ge. A generalized shuffle framework for privacy amplification: Strengthening privacy guarantees and enhancing utility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11267–11275, 2024.

Justin Y Chen, Shyam Narayanan, and Yinzhan Xu. All-pairs shortest path distances with differential privacy: Improved algorithms for bounded and unbounded weights. *arXiv preprint arXiv:2204.02335*, 2022.

Yeshwanth Cherapanamjeri, Sandeep Silwal, David P Woodruff, Fred Zhang, Qiuyi Zhang, and Samson Zhou. Robust algorithms on adaptive inputs from bounded adversaries. *arXiv preprint arXiv:2304.07413*, 2023.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

Michael B Cohen and Richard Peng. Lp row sampling by lewis weights. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 183–192, 2015.

Michael B Cohen, Ben Cousins, Yin Tat Lee, and Xin Yang. A near-optimal algorithm for approximating the john ellipsoid. In *Conference on Learning Theory*, pp. 849–873. PMLR, 2019.

Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021a.

Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021b.

Vincent Cohen-Addad, Chenglin Fan, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub M Tarnawski. Near-optimal correlation clustering with privacy. *Advances in Neural Information Processing Systems*, 35:33702–33715, 2022.

Samuel I Daitch and Daniel A Spielman. Faster approximate lossy generalized flow via interior point algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 451–460, 2008.

George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, 13:339–347, 1951.

Wei Dong, Zijun Chen, Qiyao Luo, Elaine Shi, and Ke Yi. Continual observation of joins under differential privacy. *Proceedings of the ACM on Management of Data*, 2(3):1–27, 2024.

David Durfee, Kevin A Lai, and Saurabh Sawlani. ℓ_1 regression using lewis weights preconditioning and stochastic gradient descent. In *Conference On Learning Theory*, pp. 1626–1656. PMLR, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Marek Eliáš, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. Differentially private release of synthetic graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 560–578. SIAM, 2020.

Alessandro Epasto, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. k-means clustering with distance-based privacy. *Advances in Neural Information Processing Systems*, 36, 2024.

Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Tight and robust private mean estimation with few users. In *International Conference on Machine Learning*, pp. 16383–16412. PMLR, 2022.

Chenglin Fan and Ping Li. Distances release with differential privacy in tree and grid graph. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pp. 2190–2195. IEEE, 2022.

Chenglin Fan, Ping Li, and Xiaoyun Li. k-median clustering via metric embedding: towards better initialization with differential privacy. *Advances in Neural Information Processing Systems*, 36, 2024.

Alireza Farhadi, MohammadTaghi Hajiaghayi, and Elaine Shi. Differentially private densest subgraph. In *International Conference on Artificial Intelligence and Statistics*, pp. 11581–11597. PMLR, 2022.

Filippo Galli, Catuscia Palamidessi, and Tommaso Cucinotta. Online sensitivity optimization in differentially private learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12109–12117, 2024.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. *arXiv preprint arXiv:2309.07418*, 2023a.

Yeqi Gao, Zhao Song, Xin Yang, and Yufa Zhou. Differentially private attention computation. *arXiv preprint arXiv:2305.04701*, 2023b.

Quan Geng, Wei Ding, Ruiqi Guo, and Sanjiv Kumar. Tight analysis of privacy and utility tradeoff in approximate differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 89–99. PMLR, 2020.

Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. In *Conference on Learning Theory*, pp. 1948–1989. PMLR, 2022.

Sivakanth Gopi, Yin Tat Lee, Daogao Liu, Ruoqi Shen, and Kevin Tian. Private convex optimization in general norms. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 5068–5089. SIAM, 2023.

Jiuxiang Gu, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Differential privacy mechanisms in neural tangent kernel regression. *arXiv preprint arXiv:2407.13621*, 2024.

Jiuxiang Gu, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Differential privacy mechanisms in neural tangent kernel regression. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2342–2356. IEEE, 2025.

Yuzhou Gu and Zhao Song. A faster small treewidth sdp solver. *arXiv preprint arXiv:2211.06033*, 2022.

Yuzhou Gu, Zhao Song, and Lichen Zhang. A nearly-linear time algorithm for structured support vector machines. *arXiv preprint arXiv:2307.07735*, 2023.

Hang Hu, Zhao Song, Omri Weinstein, and Danyang Zhuo. Training overparametrized neural networks in sublinear time. In *arXiv preprint arXiv: 2208.04508*, 2022.

Jerry Yao-Chieh Hu, Erzhi Liu, Han Liu, Zhao Song, and Lichen Zhang. On differentially private string distances. *arXiv preprint arXiv:2411.05750*, 2024.

Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. Solving sdp faster: A robust ipm framework and efficient implementation. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 233–244. IEEE, 2022.

Ziyue Huang and Ke Yi. Approximate range counting under differential privacy. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2021.

Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st annual symposium on foundations of computer science (FOCS)*, pp. 910–918. IEEE, 2020a.

Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 944–953, 2020b.

Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. Faster dynamic matrix inverse for faster lps. In *STOC*, 2021.

Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees. *arXiv preprint arXiv:1909.03577*, 2019.

Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302–311, 1984.

Yekun Ke, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Dpbloomfilter: Securing bloom filters with differential privacy. *arXiv preprint arXiv:2502.00693*, 2025.

Rasmus Kyng, Richard Peng, Sushant Sachdeva, and Di Wang. Flows in almost linear time via adaptive preconditioning. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 902–913, 2019.

Yin Tat Lee and Aaron Sidford. Path finding i: Solving linear programs with\˜ o (sqrt (rank)) linear system solves. *arXiv preprint arXiv:1312.6677*, 2013.

Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 424–433. IEEE, 2014.

Yin Tat Lee and Aaron Sidford. Solving linear programs with sqrt (rank) linear system solves. *arXiv preprint arXiv:1910.08033*, 2019.

Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory*, pp. 2140–2157. PMLR, 2019.

Ping Li and Xiaoyun Li. Differential privacy with random projections and sign random projections. *arXiv preprint arXiv:2306.01751*, 2023.

Ping Li and Xiaoyun Li. Smooth flipping probability for differential private sign random projection methods. *Advances in Neural Information Processing Systems*, 36, 2024.

Xuechen Li, Daogao Liu, Tatsunori B Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin-Tat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 35:28616–28630, 2022.

Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Differential privacy of cross-attention with provable guarantee. *arXiv preprint arXiv:2407.14717*, 2024.

Erzhi Liu, Jerry Yao-Chieh Hu, Alex Reneau, Zhao Song, and Han Liu. Differentially private kernel density estimation. *arXiv preprint arXiv:2409.01688*, 2024.

Yury Makarychev, Naren Sarayu Manoj, and Max Ovsiankin. Streaming algorithms for ellipsoidal approximation of convex polytopes. In *Conference on Learning Theory*, pp. 3070–3093. PMLR, 2022.

Lingsheng Meng and Bing Zheng. The optimal perturbation bounds of the moore–penrose inverse under the frobenius norm. *Linear algebra and its applications*, 432(4):956–963, 2010.

Shyam Narayanan. Private high-dimensional hypothesis testing. In *Conference on Learning Theory*, pp. 3979–4027. PMLR, 2022.

Shyam Narayanan. Better and simpler lower bounds for differentially private statistical estimation. *arXiv preprint arXiv:2310.06289*, 2023.

Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 ieee 54th annual symposium on foundations of computer science*, pp. 117–126. IEEE, 2013.

Aditya Parulekar, Advait Parulekar, and Eric Price. L1 regression with lewis weights subsampling. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2021.

Tao Qi, Huili Wang, and Yongfeng Huang. Towards the robustness of differentially private federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18):19911–19919, Mar. 2024. doi: 10.1609/aaai.v38i18.29967. URL https://ojs.aaai.org/index.php/AAAI/article/view/29967.

Lianke Qin, Rajesh Jayaram, Elaine Shi, Zhao Song, Danyang Zhuo, and Shumo Chu. Adore: Differentially oblivious relational database operators. *arXiv preprint arXiv:2212.05176*, 2022.

Lianke Qin, Zhao Song, Lichen Zhang, and Danyang Zhuo. An online and unified algorithm for projection matrix vector multiplication with application to empirical risk minimization. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 101–156. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/qin23a.html.

Ilya Razenshteyn, Zhao Song, and David P Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 250–263, 2016.

James Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical programming*, 40(1):59–93, 1988.

Rob Romijnders, Christos Louizos, Yuki M Asano, and Max Welling. Protect your score: Contact-tracing with differential privacy guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14829–14837, 2024.

Zhao Song and Zheng Yu. Oblivious sketching-based central path method for linear programming. In *International Conference on Machine Learning*, pp. 9835–9847. PMLR, 2021.

Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise l1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 688–701, 2017.

Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *SODA*. arXiv preprint arXiv:1704.08246, 2019.

Zhao Song, Zhaozhuo Xu, Yuanyuan Yang, and Lichen Zhang. Accelerating frank-wolfe algorithm using low-dimensional and adaptive data structures. *arXiv preprint arXiv:2207.09002*, 2022a.

Zhao Song, Zhaozhuo Xu, and Lichen Zhang. Speeding up sparsification using inner product search data structures. *arXiv preprint arXiv:2204.03209*, 2022b.

Zhao Song, Yitan Wang, Zheng Yu, and Lichen Zhang. Sketching for first order method: efficient algorithm for low-bandwidth channel and vulnerability. In *International Conference on Machine Learning*, pp. 32365–32417. PMLR, 2023a.

Zhao Song, Xin Yang, Yuanyuan Yang, and Lichen Zhang. Sketching meets differential privacy: fast algorithm for dynamic kronecker projection maintenance. In *International Conference on Machine Learning*, pp. 32418–32462. PMLR, 2023b.

Zhao Song, Xin Yang, Yuanyuan Yang, and Lichen Zhang. Sketching meets differential privacy: Fast algorithm for dynamic kronecker projection maintenance. In *ICML*, 2023c.

Pravin M Vaidya. An algorithm for linear programming which requires o (((m+ n) n 2+(m+ n) 1.5 n) l) arithmetic operations. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pp. 29–38, 1987.

Pravin M Vaidya. Speeding-up linear programming using fast matrix multiplication. In *30th annual symposium on foundations of computer science*, pp. 332–337. IEEE Computer Society, 1989.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Per-Åke Wedin. Perturbation theory for pseudo-inverses. *BIT Numerical Mathematics*, 13:217–232, 1973.

Max A Woodbury. *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.

David P Woodruff and Taisuke Yasuda. Online lewis weight sampling. *ACM Transactions on Algorithms*, 2023.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Bourgan: Generative networks with metric embeddings. *Advances in neural information processing systems*, 31, 2018.

Jiyan Yang, Yin-Lam Chow, Christopher Ré, and Michael W Mahoney. Weighted sgd for ℓ_p regression with randomized preconditioning. *Journal of Machine Learning Research*, 18(211):1–43, 2018.

Jiahao Yu, Haozheng Luo, Jerry Yao-Chieh Hu, Wenbo Guo, Han Liu, and Xinyu Xing. Enhancing jailbreak attack against large language models through silent tokens. *arXiv preprint arXiv:2405.20653*, 2024.

Appendix

Roadmap.

- In Section A, we provide a sensitivity analysis and perturbation bounds for PSD Matrices in DP guarantees.
- In Section B, we introduce the Gaussian Sampling Mechanism.
- In Section C, we present a detailed proof of the main result stated in Section 4.
- In Section D, we provide detailed proof of DP guarantees introduced in Section 5.
- In Section E, we give a complete proof of the utility guarantees outlined in Section 5.

A Sensitivity and Spectral Perturbation of PSD Matrix

A.1 Sensitivity of PSD Matrix

In this section, we provide more lemmas related to sensitivity.

Lemma A.1 (Lemma D.1 in (Gao et al., 2023b)). *If $X \in \mathbb{R}^{n \times d}$ and $X' \in \mathbb{R}^{n \times d}$ are neighboring datasets (see Definition 3.5 and Definition 3.6), then $(1 - 2\alpha\beta/\eta)XX^\top \preceq X'X'^\top \preceq (1 + 2\alpha\beta/\eta)XX^\top$.*

Proof. Let $i \in [d]$ be index that $X_{*,i}$ and $X'_{*,i}$ are different (See Definition 3.6).

We have

$$\begin{aligned} X'X'^\top &= \sum_{j=1}^{d} X'_{*,j} X'^\top_{*,j} \\ &= (\sum_{j \in [d] \setminus \{i\}} X'_{*,j} X'^\top_{*,j}) + X'_{*,i} X'^\top_{*,i} \\ &= (\sum_{j \in [d] \setminus \{i\}} X_{*,j} X^\top_{*,j}) + X'_{*,i} X'^\top_{*,i} \\ &= XX^\top - X_{*,i} X^\top_{*,i} + X'_{*,i} X'^\top_{*,i} \end{aligned}$$

where the first step is the result of matrix multiplication, the second step is from simple algebra, the third step follows from Definition 3.6, and the last step comes from simple algebra.

We know that

$$\begin{aligned} \|X_{*,i} X^\top_{*,i} - X'_{*,i} X'_{*,i}\| &= \|X_{*,i} X^\top_{*,i} - X_{*,i} X'^\top_{*,i} + X_{*,i} X'^\top_{*,i} - X'_{*,i} X'_{*,i}\| \\ &\leq \|X_{*,i} X^\top_{*,i} - X_{*,i} X'^\top_{*,i}\| + \|X_{*,i} X'^\top_{*,i} - X'_{*,i} X'_{*,i}\| \\ &\leq \|X_{*,i}\|_2 \cdot \|X_{*,i} - X'_{*,i}\|_2 + \|X_{*,i} - X'_{*,i}\|_2 \cdot \|X'_{*,i}\|_2 \\ &\leq 2\alpha\beta \end{aligned} \tag{1}$$

where the first step is from adding a new term $X_{*,i} X'^\top_{*,i}$, the second step follows from the triangle inequality, the third step follows from Fact 3.2, and the last step is due to Definition 3.5 and Definition 3.6.

Thus, we have $X'X'^\top \succeq XX^\top - 2\alpha\beta I_n \succeq (1 - 2\alpha\beta/\eta)XX^\top$. Similarly, we have $X'X'^\top \preceq XX^\top + 2\alpha\beta I_n \preceq (1 + 2\alpha\beta/\eta)XX^\top$. □

Lemma A.2 (Lemma D.2 in (Gao et al., 2023b)). *If the following conditions hold,*

- *Let α and η be defined in Definition 3.5.*
- *Let β be defined in Definition 3.6.*

- *By Lemma A.1, X and X' are neighboring datasets such that*

$$(1 - 2\alpha\beta/\eta)XX^\top \preceq X'X'^\top \preceq (1 + 2\alpha\beta/\eta)XX^\top$$

Then, we have

- *Part 1.*

$$\|(XX^\top)^{-1/2}X'X'^\top(XX^\top)^{-1/2} - I\| \leq 2\alpha\beta/\eta$$

- *Part 2.*

$$\|(XX^\top)^{-1/2}X'X'^\top(XX^\top)^{-1/2} - I\|_F \leq 2\sqrt{n}\alpha\beta/\eta$$

Proof. The proof is straightforward, and we omit the details here. □

Lemma A.3 (Spectral norm of $H - \widetilde{H}$)**.** *If we have the below conditions,*

- **Condition 1.** *If $\mathcal{D} \in \mathbb{R}^{n\times d}$ and $\mathcal{D}' \in \mathbb{R}^{n\times d}$ are neighboring dataset (see Definition 3.6)*
- **Condition 2.** *Let $H := A^\top W^{1-2/p}A$ denotes the symmetric positive semi-definite matrix generated by $\mathcal{D}$.*
- **Condition 3.** *Let $\widetilde{H}$ denote the private H generated by Algorithm 4 with H as the input.*
- **Condition 4.** *Let $\eta_{\max} I_{n\times n} \succeq H \succeq \eta_{\min} I_{n\times n}$, for some $\eta_{\max}, \eta_{\min} \in \mathbb{R}$.*
- **Condition 5.** *Let $\rho = O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$.*
- **Condition 6.** *Let $\gamma \in (0, 1)$.*

Then, with probability $1 - \gamma$, we have

$$\|H - \widetilde{H}\| \leq \rho \cdot \eta_{\max}$$

Proof. By Part 3 of Lemma 5.8, with probability $1 - \gamma$, we have

$$(1 - \rho)H \preceq \widetilde{H} \preceq (1 + \rho)H$$

which implies

$$-\rho H \preceq \widetilde{H} - H \preceq \rho H \tag{2}$$

Then, we have

$$\|\widetilde{H} - H\| \leq \rho \cdot \eta_{\max}$$

□

Lemma A.4 ((Wedin, 1973), Theorem 1.1 in (Meng & Zheng, 2010))**.** *Given two matrices $A, B \in \mathbb{R}^{d_1\times d_2}$ with full column rank, we have*

$$\|A^\dagger - B^\dagger\| \lesssim \max(\|A^\dagger\|^2, \|B^\dagger\|^2) \cdot \|A - B\|.$$

B Gaussian Sampling Mechanism

In this section, we restate the analysis for "Gaussian Sampling Mechanism", which guarantees the privacy of our algorithm and provides potential tools for demonstrating its utility.

Lemma B.1 (DP guarantees for $(A^\top WA)^{-1}$, Theorem 6.12 in (Gao et al., 2023b), Theorem 5.1 in (Alabi et al., 2023), Lemma D.1 in (Gu et al., 2025), formal version of Lemma 5.8)**.** *If we have the below conditions,*

- **Condition 1.** *Let $\mathcal{D}$ and $\mathcal{D}'$ are neighboring dataset (see Definition 3.6).*

- **Condition 2.** *Let $H := A^\top W^{1-2/p} A$ denotes the symmetric positive semi-definite matrix generated by X, and H' denotes the symmetric positive definite matrix generated by neighboring dataset X'.*
- **Condition 3.** *Let $\epsilon_\alpha \in (0,1)$ and $\delta_\alpha \in (0,1)$ denote the DP parameter for $A^\top W^{1-2/p} A$.*
- **Condition 4.** *Let $\mathcal{Y}, \mathcal{Y}'$ denote neighboring datasets, which differ by a single data element.*
- **Condition 5.** *Let Δ be defined in Definition 5.2 and $\Delta < 1$.*
- **Condition 6.** *Let $M, \mathcal{M}$ be defined in Definition 5.1 and $M \leq \Delta$.*
- **Condition 7.** *Let the input $\Sigma = \mathcal{M}(\mathcal{Y})$.*
- **Condition 8.** *Let $\rho = O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$.*
- **Condition 9.** *Let $k \in \mathbb{N}$.*
- **Condition 10.** *Let $\gamma \in (0,1)$.*
- **Condition 11.** *Let $\eta_{\max} I_{n\times n} \succeq H \succeq \eta_{\min} I_{n\times n}$, for some $\eta_{\max}, \eta_{\min} \in \mathbb{R}$.*
- **Condition 12.** *Let $\widetilde{H}$ denote the private H generated by Algorithm 4 with H as the input.*
- **Condition 13.** *Let $\sqrt{n}\psi/\eta_{\min} < \Delta$, where Δ is defined in Definition 5.2.*

Then, there exists an Algorithm 4 such that

- *Part 1. Algorithm 4 is $(\epsilon_\alpha, \delta_\alpha)$-DP.*
- *Part 2. Outputs $\widehat{\Sigma} \in \mathbb{S}_+^n$ such that with probabilities at least $1-\gamma$,*

$$\|\Sigma^{-1/2} \widehat{\Sigma} \Sigma^{-1/2} - I_n\|_F \leq \rho$$

- *Part 3.*

$$(1-\rho)\Sigma \preceq \widehat{\Sigma} \preceq (1+\rho)\Sigma.$$

C Proof of Main Result

Lemma C.1 (Correctness of the algorithm, formal version of Lemma 4.3)**.** *The output of* $\textsc{Update}(w)$ *in Algorithm 2 satisfies*

$$M = A^\top (A V^{1-2/p} A^\top)^{-1} A \text{ and}$$
$$Q = M V^{1/2-1/p} R^\top$$

The output of $\textsc{Query}(h)$ *in Algorithm 3 satisfies*

$$p_s = \widetilde{P} (R^\top)_{*,l} R_{l,*} h$$
$$p_x = (I - \widetilde{P}) (R^\top)_{*,l} R_{l,*} h,$$

where $\widetilde{P} = V^{1/2-1/p} A^\top (A \widetilde{V}^{1-2/p} A^\top)^{-1} A \widetilde{V}^{1/2-1/p}$, and $\widetilde{V}$ is outputted by $\textsc{Update}(w)$.

Proof. Let S denote the support of Δ.

Thus, by the Woodbury matrix identity (Fact 3.1) and definition of M^{new}, we have

$$\begin{aligned}
& A^\top (A (V^{\mathrm{new}})^{1-2/p} A^\top)^{-1} A \\
= & A^\top (A (V + \Delta)^{1-2/p} A^\top)^{-1} A \\
= & A^\top ((A V^{1-2/p} A^\top)^{-1} - (A V^{1-2/p} A^\top)^{-1} A_{*,S} \cdot (\Delta_{S,S}^{-1} + (A^\top)_{S,*} (A V^{1-2/p} A^\top)^{-1} A_{*,S})^{-1}
\end{aligned}$$

$$
\begin{aligned}
&\cdot (A^\top)_{S,*}(AV^{1-2/p}A^\top)^{-1})A \\
=\ & A^\top (AV^{1-2/p}A^\top)^{-1}A - A^\top (AV^{1-2/p}A^\top)^{-1}A_{*,S} \\
&\cdot (\Delta_{S,S}^{-1} + (A^\top)_{S,*}(AV^{1-2/p}A^\top)^{-1}A_{*,S})^{-1} \cdot (A^\top)_{S,*}(AV^{1-2/p}A^\top)^{-1}A \\
=\ & M - M_{*,S}(\Delta_{S,S}^{-1} + M_{S,S})^{-1}M_{S,*} \\
=\ & M^{\mathrm{new}}.
\end{aligned}
$$

where the first step follows from the definition of $V^{\mathrm{new}} = V + \Delta$, the second step follows from the Fact 3.1, the third step distributes the terms, the fourth step follows from the definition of $M = A^\top (AV^{1-2/p}A^\top)^{-1}A$, and the last step follows from the definition of M^{new}.

Note the output $M = M^{\mathrm{new}}$ and $V = V^{\mathrm{new}}$, so we have the output satisfying $M = A^\top (AVA^\top)^{-1}A$.

As for Q, notice by definition

$$
\begin{aligned}
Q^{\mathrm{new}} &= Q + (M^{\mathrm{new}} \cdot \Gamma) \cdot R^\top + (M^{\mathrm{new}} - M) \cdot V^{1/2-1/p} \cdot R^\top \\
&= MV^{1/2-1/p}R^\top + (M^{\mathrm{new}} \cdot \Gamma) \cdot R^\top + (M^{\mathrm{new}} - M) \cdot V^{1/2-1/p} \cdot R^\top \\
&= MV^{1/2-1/p}R^\top + M^{\mathrm{new}}((V^{\mathrm{new}})^{1/2-1/p} - V^{1/2-1/p})R^\top + (M^{\mathrm{new}} - M)V^{1/2-1/p}R^\top \\
&= M^{\mathrm{new}}((V^{\mathrm{new}})^{1/2-1/p} - V^{1/2-1/p})R^\top + M^{\mathrm{new}}V^{1/2-1/p}R^\top \\
&= M^{\mathrm{new}}(V^{\mathrm{new}})^{1/2-1/p}R^\top
\end{aligned}
$$

where the first step follows from the definition of Q^{new}, the second step follows from definition of Q, the third step follows from the definition of Γ, the fourth step distributes and eliminates the term $MV^{1/2-1/p}R^\top$, and the last step distributes and eliminates the term $M^{\mathrm{new}}V^{1/2-1/p}R^\top$.

Again, since the output $Q = Q^{\mathrm{new}}$, $M = M^{\mathrm{new}}$ and $V = V^{\mathrm{new}}$, we have the output satisfying $Q = MV^{1/2-1/p}R^\top$.

For QUERY(h) procedure, by definition we have

$$
\begin{aligned}
p_m &= \widetilde{V}^{1/2-1/p} \cdot (M_{*,\widetilde{S}}) \cdot (\widetilde{\Delta}_{\widetilde{S},\widetilde{S}}^{-1} + M_{\widetilde{S},\widetilde{S}})^{-1} \cdot (Q_{\widetilde{S},l} + M_{\widetilde{S},*} \cdot \widetilde{\Gamma} \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h \\
&= \widetilde{V}^{1/2-1/p} \cdot (M_{*,\widetilde{S}}) \cdot (\widetilde{\Delta}_{\widetilde{S},\widetilde{S}}^{-1} + M_{\widetilde{S},\widetilde{S}})^{-1} \\
&\quad \cdot ((MV^{1/2-1/p}R^\top)_{\widetilde{S},l} + M_{\widetilde{S},*} \cdot (\widetilde{V}^{1/2-1/p} - V^{1/2-1/p}) \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h \\
&= \widetilde{V}^{1/2-1/p} \cdot (M_{*,\widetilde{S}}) \cdot (\widetilde{\Delta}_{\widetilde{S},\widetilde{S}}^{-1} + M_{\widetilde{S},\widetilde{S}})^{-1} \cdot M_{\widetilde{S},*} \cdot \widetilde{V}^{1/2-1/p} \cdot (R^\top)_{*,l} \cdot h, \qquad (3)
\end{aligned}
$$

where the first step follows from the definition of p_m, the second step follows from the definition of Q and $\widetilde{\Gamma}$, and the third step eliminates the terms.

Thus,

$$
\begin{aligned}
p_s &= \widetilde{V}^{1/2-1/p} \cdot (Q_{*,l} + M \cdot \widetilde{\Gamma} \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h - p_m \\
&= \widetilde{V}^{1/2-1/p} \cdot ((MV^{1/2-1/p}R^\top)_{*,l} + M \cdot (\widetilde{V}^{1/2-1/p} - V^{1/2-1/p}) \cdot (R^\top)_{*,l}) \cdot R_{l,*} \cdot h - p_m \\
&= \widetilde{V}^{1/2-1/p} \cdot M \cdot \widetilde{V}^{1/2-1/p} \cdot (R^\top)_{*,l} \cdot R_{l,*} \cdot h - p_m \\
&= \widetilde{V}^{1/2-1/p}(M - M_{*,\widetilde{S}}(\widetilde{\Delta}_{\widetilde{S},\widetilde{S}}^{-1} + M_{\widetilde{S},\widetilde{S}})^{-1}M_{\widetilde{S},*})\widetilde{V}^{1/2-1/p}(R^\top)_{*,l}R_{l,*}h, \qquad (4)
\end{aligned}
$$

where the first step follows the definition of p_s, the second step follows from the definition of Q and $\widetilde{\Gamma}$, the third step follows from eliminates the terms, and the last step substitutes p_m by Eq. (3).

Note $\widetilde{V}$ only differs from V in entries corresponding to the set $\widetilde{S}$, again by the Woodbury matrix identity (Fact 3.1) and the definition of M, we have

$$
\begin{aligned}
&A^\top (A\widetilde{V}^{1-2/p}A^\top)^{-1}A \\
=\ & A^\top (A(V^{1-2/p} + \widetilde{\Delta})A^\top)^{-1}A \\
=\ & A^\top ((AV^{1-2/p}A^\top)^{-1} - (AV^{1-2/p}A^\top)^{-1}A_{*,\widetilde{S}}
\end{aligned}
$$

$$\begin{aligned}
&\cdot (\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + (A^\top)_{\widetilde{S},*}(AV^{1-2/p}A^\top)^{-1}A_{*,\widetilde{S}})^{-1} \cdot (A^\top)_{\widetilde{S},*}(AV^{1-2/p}A^\top)^{-1})A \\
= &A^\top(AV^{1-2/p}A^\top)^{-1}A - A^\top(AV^{1-2/p}A^\top)^{-1}A_{*,\widetilde{S}} \\
&\cdot (\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + (A^\top)_{\widetilde{S},*}(AV^{1-2/p}A^\top)^{-1}A_{*,\widetilde{S}})^{-1} \cdot (A^\top)_{\widetilde{S},*}(AV^{1-2/p}A^\top)^{-1}A \\
= &M - M_{*,\widetilde{S}}(\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + M_{\widetilde{S},\widetilde{S}})^{-1}M_{\widetilde{S},*}, \qquad (5)
\end{aligned}$$

where the first step follows from $\widetilde{V}^{1-2/p} = V^{1-2/p} + \widetilde{\Delta}$, the second step follows from Fact 3.1 , the third step distributes two terms, and the last step follows from the definition of M, which implies

$$\begin{aligned}
p_s &= \widetilde{V}^{1/2-1/p}A^\top(A\widetilde{V}^{1-2/p}A^\top)^{-1}A\widetilde{V}^{1/2-1/p}(R^\top)_{*,l}R_{l,*}h \\
&= \widetilde{P}(R^\top)_{*,l}R_{l,*}h, \qquad (6)
\end{aligned}$$

where the first step follows from Eq. (4) and (5), and the second step follows from the definition of $\widetilde{P}$.

Further,

$$\begin{aligned}
p_x &= (R^\top)_{*,l}R_{l,*}h - p_s \\
&= (I - \widetilde{P})(R^\top)_{*,l}R_{l,*}h,
\end{aligned}$$

where the first step follows from the definition of p_x, and the second step follows from Eq. (6).

Thus we completes the proof. □

Lemma C.2 (Query time, formal version of Lemma 4.6)**.** *The query time of data-structure* MAINTAINPROJECTION *(Algorithm 1) is* $O(n^{1+b+o(1)} + n^{1+a+o(1)})$.

Proof. Notice by the algorithm we have $|\widetilde{S}| \leq n^a$. Thus, $\widetilde{\Gamma}$ is a sparse diagonal matrix with at most n^a non-zero elements. The running time mainly comes from three parts.

Part 1. Computing p_m:

- Compute $R_{l,*} \cdot h$: matrix-vector multiplication between matrix of size $n^b \times n$ and vector of size $n \times 1$, this takes n^{1+b} time.
- Compute $(R^\top)_{*,l} \cdot (R_{l,*}h)$: matrix-vector multiplication between matrix of size $n \times n^b$ and vector of size $n^b \times 1$, this takes n^{1+b} time.
- Compute $\widetilde{\Gamma} \cdot (R^\top_{l,*}R_{l,*}h)$: matrix-vector multiplication between sparse diagonal matrix with at most n^a non-zero elements and vector of size $n \times 1$, this takes n^a time.
- Compute $M_{\widetilde{S},*} \cdot (\widetilde{\Gamma}R^\top_{l,*}h)$: matrix-vector multiplication between matrix of size at most $n^a \times n$ and sparse vector with at most n^a non-zero elements, this takes n^{2a} time.
- Compute $Q_{\widetilde{S},l} \cdot (R_{l,*}h)$: matrix-vector multiplication between matrix of size at most $n^a \times n^b$ and vector of size $n^b \times 1$, this takes n^{a+b} time.
- Compute $(\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + M_{\widetilde{S},\widetilde{S}})^{-1}$: inverse of matrix of size at most $n^a \times n^a$, this takes $n^{a\omega}$ time.
- Compute $(\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + M_{\widetilde{S},\widetilde{S}})^{-1} \cdot [(Q_{\widetilde{S},l} + M_{\widetilde{S},*}\widetilde{\Gamma}(R^\top)_{*,l})R_{l,*}h]$: matrix-vector multiplication between matrix of size at most $n^a \times n^a$ and vector of size at most $n^a \times 1$, this takes n^{2a} time.
- Compute $\widetilde{V}^{1/2-1/p} \cdot (M_{*,\widetilde{S}})$: matrix-matrix multiplication between diagonal matrix of size $n \times n$ and matrix of size at most $n \times n^a$, this takes n^{1+a} time.
- Compute $[\widetilde{V}^{1/2-1/p}M_{*,\widetilde{S}}] \cdot [(\widetilde{\Delta}^{-1}_{\widetilde{S},\widetilde{S}} + M_{\widetilde{S},\widetilde{S}})^{-1}(Q_{\widetilde{S},l} + M_{\widetilde{S},*}\widetilde{\Gamma}(R^\top)_{*,l})R_{l,*}h]$: matrix-vector multiplication between matrix of size at most $n \times n^a$ and vector of size at most $n^a \times 1$, this takes n^{1+a} time.

To conclude, we can compute p_m in $O(n^{1+b} + n^{a\omega} + n^{1+a})$ time.

Part 2. Computing p_s:

- Compute $R_{l,*}h$ and $\widetilde{\Gamma} R_{l,*}^\top R_{l,*}h$ in same way as in calculating p_m: take n^{1+b} and $O(n^{1+b} + n^a)$ time respectively.
- Compute $\widetilde{V}^{1/2-1/p} \cdot Q_{*,l}$: matrix-matrix multiplication between diagonal matrix of size $n \times n$ and matrix of size $n \times n^b$, takes n^{1+b} time.
- Compute $[\widetilde{V}^{1/2-1/p} Q_{*,l}] \cdot [R_{l,*}h]$: matrix-vector multiplication between matrix of size $n \times n^b$ and vector of size $n^b \times 1$, takes n^{1+b} time.
- Compute $M \cdot [\widetilde{\Gamma} R_{l,*}^\top R_{l,*}h]$: matrix-vector multiplication between matrix of size $n \times n$ and sparse vector with at most n^a non-zero elements, takes $O(n^{1+a})$ time.
- Compute $\widetilde{V}^{1/2-1/p} \cdot [M \widetilde{\Gamma} R_{l,*}^\top R_{l,*}h]$: matrix-vector multiplication between diagonal matrix of size $n \times n$ and vector of size $n \times 1$, takes n time.

To conclude, we can compute p_s in $O(n^{1+b} + n^{1+a})$ time.

Part 3. Computing p_x:

- Compute $R_{l,*}^\top R_{l,*}h$ in same way as in calculating p_m: take $O(n^{1+b})$ time.

Thus, the overall running time is

$$O(n^{1+a} + n^{1+b} + n^{a\omega}).$$

Finally, we note that $\omega \leq 3 - \alpha \leq 3 - a$ (see (Cohen et al., 2021b)) and hence $a \cdot \omega \leq a(3-a) \leq (1+a)$. Therefore, the final running time it takes is $O(b^{1+b+o(1)} + n^{1+a+o(1)})$. □

D Proof of Differential Privacy Guarantees

Lemma D.1 (Sensitivity of $W^{1/2-1/p}A$, formal version of Lemma 5.3)**.** *If the following conditions hold:*

- *Let the neighboring dataset X and X' be defined in Definition 3.6.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for J.*
- *Let $J := W^{1/2-1/p}A$ denote the data generated by X and J' denote the data generated by neighboring dataset X', where $W^{1/2-1/p} \in \mathbb{R}^{m \times m}$ and $A \in \mathbb{R}^{m \times n}$.*
- *Let $\beta > 0$ be defined as Definition 3.6.*

Then, we can show that the sensitivity of J is $\sqrt{n} \cdot \beta$.

Proof. Without loss of generality, we use $x_m \in \mathbb{R}^n$ and $x'_m \in \mathbb{R}^n$ to denote the different items in X and X'. According to the definition of the neighboring dataset, we have

$$\|x_m - x'_m\|_2 \leq \beta.$$

Then, we have

$$\begin{aligned} \|J - J'\|_1 &= \|x_m - x'_m\|_1 \\ &\leq \sqrt{n} \cdot \|x_m - x'_m\|_2 \\ &= \sqrt{n} \cdot \beta, \end{aligned}$$

where the first step follows from $\|u - v\|_1 \leq \sqrt{n}\|u - v\|_2$ for any $u, v \in \mathbb{R}^n$, and the second step follows from $\|x_m - x'_m\|_2 \leq \beta$. □

Lemma D.2 (DP guarantees for $W^{1/2-1/p}A(A^\top W^{1-2/p}A)^{-1}A^\top W^{1/2-1/p}\cdot h$, formal version of Lemma 5.10)**.** *If the following conditions hold:*

- *Let $\epsilon_\alpha, \delta_\alpha \in \mathbb{R}$ denote the DP parameter for $A^\top W^{1-2/p}A$.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for $W^{1/2-1/p}A$ and $A^\top W^{1/2-1/p}$.*
- *Let $\epsilon = 2\epsilon_J + \epsilon_\alpha$.*
- *Let $\delta = 2\delta_J + \delta_\alpha$.*
- *Let H and $\widetilde{H}$ be defined as Lemma 5.9.*
- *Let J and $\widetilde{J}$ be defined as Lemma 5.4.*

Then, we can show $W^{1/2-1/p}A(A^\top W^{1-2/p}A)^{-1}A^\top W^{1/2-1/p}\cdot h$ is (ϵ,δ)-DP.

Proof. Since we have

- $A^\top W^{1-2/p}A$ is $(\epsilon_\alpha,\delta_\alpha)$-DP.
- $W^{1/2-1/p}A$ is (ϵ_J,δ_J)-DP.
- $A^\top W^{1/2-1/p}$ is (ϵ_J,δ_J)-DP.
- $\epsilon = 2\epsilon_J + \epsilon_\alpha, \delta = 2\delta_J + \delta_\alpha$.

By Lemma 3.9, we have $W^{1/2-1/p}A(A^\top W^{1-2/p}A)^{-1}A^\top W^{1/2-1/p}$ is (ϵ,δ)-DP.

Next, by the post-processing property of differential privacy (Lemma 3.7), we conclude that $W^{1/2-1/p}A(A^\top W^{1-2/p}A)^{-1}A^\top W^{1/2-1/p}\cdot h$ is also (ϵ,δ)-DP.

Thus, we complete the proof. □

E Proof of Utility Guarantees

Lemma E.1 (Utility guarantees for $W^{1/2-1/p}A$, formal version of 5.5)**.** *If the following conditions hold:*

- *Let the neighboring dataset X and X' be defined in Definition 3.6.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for J.*
- *Let $\Delta_J := \sqrt{n}\cdot\beta$ denote the sensitivity of J.*
- *Let $J := W^{1/2-1/p}A$ denote the data generated by X, where $W^{1/2-1/p} \in \mathbb{R}^{m\times m}$ and $A \in \mathbb{R}^{m\times n}$.*
- *Let $B_L = (\Delta_J/\epsilon_J)\log(1+\frac{\exp(\epsilon_J)-1}{2\delta_J})$.*
- *Let $\widetilde{J} := J + \mathrm{TLap}(\Delta_J,\epsilon_J,\delta_J)$.*

Then, we can show that

$$\|\widetilde{J} - J\|_2 \leq \sqrt{n}\cdot B_L.$$

Proof. For $i\in[m], j\in[n]$, let $J(i,j), J'(i,j)\in\mathbb{R}$ denote the (i,j)-th entry of J and J', respectively. Let $J_i \in \mathbb{R}^n$ denotes the i-th column of J.

By the definition of $\widetilde{J}$, we have

$$\widetilde{J}(i,j) = J(i,j) + \mathrm{TLAP}(\Delta_J,\epsilon_J,\delta_J)$$

Recall that we have $B_L = (\Delta_J/\epsilon_J)\log(1+\frac{e^{\epsilon_J}-1}{2\delta_J})$. By the definition of truncated Laplace, we have

$$|\mathrm{TLAP}(\Delta_J, \epsilon_J, \delta_J)| \leq B_L.$$

Combining the above two equations, for $i \in [m]$, we have

$$\|\widetilde{J} - J\|_2 \leq \sqrt{n} \cdot B_L.$$

Thus, we complete the proof. □

Lemma E.2 (Utility guarantees for $A^\top W^{1/2-1/p}$, formal version of Lemma 5.7)**.** *If the following conditions hold:*

- *Let the neighboring dataset X and X' be defined in Definition 3.6.*
- *Let $\epsilon_J, \delta_J \in \mathbb{R}$ denote the DP parameters for J.*
- *Let $\Delta_J := \sqrt{n} \cdot \beta$ denote the sensitivity of J.*
- *Let $J^\top := A^\top W^{1/2-1/p}$ denote the data generated by X, where $W^{1/2-1/p} \in \mathbb{R}^{m \times m}$ and $A^\top \in \mathbb{R}^{n \times m}$.*
- *Let $\widetilde{J}^\top := J^\top + \mathrm{TLap}(\Delta_J, \epsilon_J, \delta_J)$.*
- *Let $B_L = (\Delta_J/\epsilon_J)\log(1 + \frac{\exp(\epsilon_J)-1}{2\delta_J})$.*

we can show that

$$\|\widetilde{J}^\top - J^\top\|_2 \leq \sqrt{n} \cdot B_L.$$

Proof. From Lemma 5.5, we have

$$\|\widetilde{J} - J\|_2 \leq \sqrt{n} \cdot B_L.$$

Then,

$$\begin{aligned} \|\widetilde{J}^\top - J^\top\|_2 &= \|(\widetilde{J} - J)^\top\|_2 \\ &= \|\widetilde{J} - J\|_2 \\ &\leq \sqrt{n} \cdot B_L. \end{aligned}$$

Where the first step follows from the invariance of the norm under transposition, the second step follows from the norm property $\|A^\top\|_2 = \|A\|_2$, and the third step follows from Lemma 5.5.

Thus, we complete the proof. □

Lemma E.3 (Utility guarantees for $(A^\top W^{1-2/p} A)^{-1}$, formal version of Lemma 5.9)**.** *If the following conditions hold:*

- **Condition 1.** *If $\mathcal{D} \in \mathbb{R}^{n \times d}$ and $\mathcal{D}' \in \mathbb{R}^{n \times d}$ are neighboring dataset (see Definition 3.6)*
- **Condition 2.** *Let $H := A^\top W^{1-2/p} A$ denotes the symmetric positive semi-definite matrix generated by $\mathcal{D}$.*
- **Condition 3.** *Let $\widetilde{H}$ denote the private H generated by Algorithm 4 with H as the input.*
- **Condition 4.** *Let $\eta_{\max} I_{n \times n} \succeq H \succeq \eta_{\min} I_{n \times n}$, for some $\eta_{\max}, \eta_{\min} \in \mathbb{R}$.*
- **Condition 5.** *Let $\sqrt{n}\psi/\eta_{\min} < \Delta$, where Δ is defined in Definition 5.2.*
- **Condition 6.** *Let $\rho = O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$.*
- **Condition 7.** *Let $\gamma \in (0, 1)$.*

Then, with probability $1 - \gamma$, we have

$$\|H^{-1} - \widetilde{H}^{-1}\| \leq O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2})$$

Proof. We consider the $\|H^{-1}\|$ term. We have

$$\begin{aligned} \|H^{-1}\| &= \|(A^\top W^{1-2/p} A)^{-1}\| \quad (7) \\ &= \sigma_{\max}((A^\top W^{1-2/p} A)^{-1}) \\ &= \frac{1}{\sigma_{\min}((A^\top W^{1-2/p} A))} \\ &\leq \frac{1}{\eta_{\min}} \quad (8) \end{aligned}$$

where the 1st step is due to Condition 2, the 2nd step is because of definition of spectral norm, the 3rd step is due to $\sigma_{\max}(A^{-1}) = 1/\sigma_{\min}(A)$ holds for any matrix A, the 4th step is from $K \succeq \eta_{\min} I_{n\times n}$.

Similarly, we can have

$$\|\widetilde{H}^{-1}\| \leq \frac{1}{\eta_{\min}} \quad (9)$$

Recall in Lemma A.3, we have

$$\|H - \widetilde{H}\| \leq \rho \cdot \eta_{\max} \quad (10)$$

Then, by Lemma A.4, we have

$$\begin{aligned} \|H^{-1} - \widetilde{H}^{-1}\| &\leq O(\max\{\|H^{-1}\|^2, \|\widetilde{H}^{-1}\|^2\} \cdot \|H - \widetilde{H}\|) \\ &\leq O(\frac{1}{\eta_{\min}^2} \cdot \|H - \widetilde{H}\|) \\ &\leq O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}) \end{aligned}$$

where the 1st step is because of Lemma A.4, the 2nd step is due to Eq. (7) and Eq. (9), the 3rd step is from Eq. (10). □

Lemma E.4 (Utility guarantees for $W^{1/2-1/p} A (A^\top W^{1-2/p} A)^{-1} A^\top W^{1/2-1/p} \cdot h$, formal version of Lemma 5.11)**.** *If the following conditions hold:*

- *If $\mathcal{D} \in \mathbb{R}^{n\times d}$ and $\mathcal{D}' \in \mathbb{R}^{n\times d}$ are neighboring dataset (see Definition 3.6)*
- *Let H and $\widetilde{H}$ be defined as Lemma 5.9.*
- *Let J and $\widetilde{J}$ be defined as Lemma 5.4.*
- *Let $\sigma_J := \|J\|_2$ denotes the ℓ_2 norm of J.*
- *Let $\sigma_h := \|h\|_2$ denotes the ℓ_2 norm of h.*
- *Let $\sigma_{H^{-1}} := \|H^{-1}\|_2$ denotes the ℓ_2 norm of H^{-1}.*
- *Let $\eta_{\max} I_{n\times n} \succeq H \succeq \eta_{\min} I_{n\times n}$, for some $\eta_{\max}, \eta_{\min} \in \mathbb{R}$.*
- *Let $\sqrt{n}\psi/\eta_{\min} < \Delta$, where Δ is defined in Definition 5.2.*
- *Let $\rho = O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$.*
- *Let $\gamma \in (0, 1)$.*
- *Let $B_L \in \mathbb{R}$ be defined in Lemma 5.5.*

Then, with probability $1 - \gamma$, we have

$$|JH^{-1}J^\top \cdot h - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top \cdot h| \leq 2\sigma_J \sigma_h \sqrt{n} \cdot B_L + \sigma_J^2 \sigma_h \cdot O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}).$$

Proof. We have

$$\begin{aligned}
&|JH^{-1}J^\top \cdot h - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top \cdot h| \\
=\ &|(JH^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top) \cdot h| \\
=\ &|((JH^{-1}J^\top - \widetilde{J}H^{-1}J^\top) + (\widetilde{J}H^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}J^\top) + (\widetilde{J}\widetilde{H}^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top)) \cdot h| \\
\leq\ &(|JH^{-1}J^\top - \widetilde{J}H^{-1}J^\top| + |\widetilde{J}H^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}J^\top| + |\widetilde{J}\widetilde{H}^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top|) \cdot \|h\|_2, \quad (11)
\end{aligned}$$

where the first step follows from basic algebra, the second step follows from basic algebra, the third step follows from triangle inequality.

Consider $|JH^{-1}J^\top - \widetilde{J}H^{-1}J^\top|$, we have:

$$\begin{aligned}
&|JH^{-1}J^\top - \widetilde{J}H^{-1}J^\top| \\
\leq\ &\|J - \widetilde{J}\|_2 \|H^{-1}\|_2 \|J^\top\|_2 \\
\leq\ &\sigma_{H^{-1}} \sigma_J \sqrt{n} \cdot B_L, \quad (12)
\end{aligned}$$

where the first step follows from Cauchy–Schwarz inequality, the second step follows from Lemma 5.5.

Consider $|\widetilde{J}H^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}J^\top|$, we have:

$$\begin{aligned}
&|\widetilde{J}H^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}J^\top| \\
\leq\ &\|\widetilde{J}\|_2 |H^{-1} - \widetilde{H}^{-1}| \|J^\top\|_2 \\
\leq\ &\sigma_J^2 \cdot O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}), \quad (13)
\end{aligned}$$

where the first step follows from Cauchy–Schwarz inequality, the second step follows from Lemma 5.9.

Consider $|\widetilde{J}\widetilde{H}^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top|$, we have:

$$\begin{aligned}
&|\widetilde{J}\widetilde{H}^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top| \\
\leq\ &\|\widetilde{J}\|_2 \|\widetilde{H}^{-1}\|_2 \|J^\top - \widetilde{J}^\top\|_2 \\
\leq\ &\sigma_{H^{-1}} \sigma_J \sqrt{n} \cdot B_L, \quad (14)
\end{aligned}$$

where the first step follows from Cauchy–Schwarz inequality, the second step follows from Lemma 5.7.

Combine the equations above, we have:

$$\begin{aligned}
&|JH^{-1}J^\top \cdot h - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top \cdot h| \\
\leq\ &(|JH^{-1}J^\top - \widetilde{J}H^{-1}J^\top| + |\widetilde{J}H^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}J^\top| + |\widetilde{J}\widetilde{H}^{-1}J^\top - \widetilde{J}\widetilde{H}^{-1}\widetilde{J}^\top|) \cdot \|h\|_2 \\
\leq\ &(\sigma_{H^{-1}} \sigma_J \sqrt{n} \cdot B_L + \sigma_J^2 \cdot O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}) + \sigma_{H^{-1}} \sigma_J \sqrt{n} \cdot B_L) \cdot \sigma_h \\
=\ &2\sigma_J \sigma_h \sqrt{n} \cdot B_L + \sigma_J^2 \sigma_h \cdot O(\frac{\rho \cdot \eta_{\max}}{\eta_{\min}^2}),
\end{aligned}$$

where the first step follows from Eq (11), the second step follows from Eq (12), Eq (13) and Eq (14), the third step follows from basic algebra.

Thus, we complete the proof. □

LLM USAGE DISCLOSURE

LLMs were used only to polish language, such as grammar and wording. These models did not contribute to idea creation or writing, and the authors take full responsibility for this paper's content.