

Almost Tight Error Bounds on Differentially Private Continual Counting

Monika Henzinger* Jalaj Upadhyay† Sarvagya Upadhyay‡

November 10, 2022

Abstract

The first large-scale deployment of private federated learning uses differentially private counting in the *continual release model* as a subroutine (Google AI blog titled “Federated Learning with Formal Differential Privacy Guarantees” on February 28, 2022). For this and several other applications, it is crucial to use a continual counting mechanism with *small mean squared error*. In this case, a concrete (or non-asymptotic) bound on the error is very relevant to reduce the privacy parameter ϵ as much as possible, and hence, it is important to improve upon the constant factor in the error term. The standard mechanism for continual counting, and the one used in the above deployment, is the *binary mechanism*. We present a novel mechanism and show that its mean squared error is both asymptotically optimal and a factor 10 smaller than the error of the binary mechanism. We also show that the constants in our analysis are almost tight by giving non-asymptotic lower and upper bounds that differ only in the constants of lower-order terms. Our mechanism also has the advantage of taking only constant time per release, while the binary mechanism takes $O(\log n)$ time, where n is the total number of released data values. Our algorithm is a matrix mechanism for the *counting matrix*. We also use our explicit factorization of the counting matrix to give an upper bound on the excess risk of the matrix mechanism-based private learning algorithm of Denisov, McMahan, Rush, Smith, and Thakurta (NeurIPS 2022).

Our lower bound for any continual counting mechanism is the first tight lower bound on continual counting under (ϵ, δ) -differential privacy and it holds against a non-adaptive adversary. It is achieved using a new lower bound on a certain factorization norm, denoted by $\gamma_F(\cdot)$, in terms of the singular values of the matrix. In particular, we show that for any complex matrix, $A \in \mathbb{C}^{m \times n}$,

$$\gamma_F(A) \geq \frac{1}{\sqrt{m}} \|A\|_1,$$

where $\|\cdot\|$ denotes the Schatten-1 norm. We believe this technique will be useful in proving lower bounds for a larger class of linear queries. To illustrate the power of this technique, we show the first lower bound on the mean squared error for answering parity queries. This bound applies to the non-continual setting and is asymptotically tight.

*Faculty of Computer Science, University of Vienna. email: monika.henzinger@univie.ac.at

†Rutgers University. email: jalaj.upadhyay@rutgers.edu

‡Fujitsu Research of America email: supadhyay@fujitsu.com

Contents

1	Introduction	1
1.1	Problem Statement and Our Contributions	2
1.1.1	Main Result	2
1.1.2	Other Contributions	4
1.2	Our Techniques	7
1.2.1	Main Result: Useful Properties of $\gamma_F(\cdot)$	7
1.2.2	Other Contributions	8
2	Notations and Preliminaries	10
2.1	Linear Algebra	10
2.2	Convex Optimization	15
2.3	Differential Privacy	17
3	Semidefinite Program for $\gamma_F(\cdot)$ Norm	20
4	Proof of Lemma 1	22
5	Proof of the Bounds on Differentially Private Continual Counting	23
5.1	Proof of Upper Bound on Differentially Private Continual Counting	23
5.2	Proof of Lower Bounds on Continual Counting	25
6	Factorization View of Known Mechanisms and Suboptimality of the Binary Mechanism	26
7	Non-asymptotic Bound on Private Online Optimization	28
8	Lower Bound on Parity Queries	31
A	Dual Characterization of $\gamma_F(A)$ SDP	35
B	Useful Properties and Bounds on $\gamma_F(\cdot)$	37
B.1	Useful Properties of $\gamma_F(\cdot)$	38
B.2	Useful Bounds on $\gamma_F(\cdot)$	39

1 Introduction

In recent years, a central problem in federated learning has been to design efficient, differentially private learning algorithm that can be deployed on a large-scale. To solve this problem, many techniques have been proposed that uses some variants of *differentially private stochastic gradient descent* (DP-SGD) in conjunction with privacy amplification by *shuffling* [BEM⁺17] or *sampling* [BST14]. However, there are inherent challenges in putting these theoretical ideas to large-scale deployments involving millions of devices¹. To assuage these issues, a recent line of work leveraged private (online) learning using *differentially private follow-the-regularized leader* (DP-FTRL). This particular approach is now employed as a subroutine in the first provably private large-scale deployment by Google for its privacy-preserving federated next word prediction model [MT22] (see also the accompanying paper by Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21] and follow-up work by Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22]).

A central subroutine used in DP-FTRL is *differentially private counting under continual observation*, aka *continual counting* [CSS11, DNPR10]. It refers to the following problem: assume an (adaptively generated) binary stream $x = (x_1, x_2, \dots, x_n)$ such that x_t is given in round t (with $1 \leq t \leq n$), the objective is to continually output in every round t , the sum of bits arriving until that round in a differentially private manner. Among many significant advantages of using continual counting for online learning is (a) its resistance to an adaptive choice of the training data set and (b) that the privacy proof using this approach does not rely on the convexity of the loss function². These two reasons play a pivotal role in its application in first production level provable differentially private neural network trained directly on user data [MT22].

Continual counting has been used in many other applications as well, including but not limited to, histogram estimation [CR21, CLSX12, HQYC21, Upa19], non-interactive local learning [STU17], graph analysis [FHO21, UUA21], stochastic convex optimization [HLL⁺22], and matrix analysis [DTTZ14, UU21]. Depending on the downstream use case, the performance of a differentially private continual mechanism is either measured in terms of *absolute error* (aka ℓ_∞ -error) or *mean squared error* (aka ℓ_2^2 -error) over the different time steps (defined below). For continual counting, Fichtenberger, Henzinger, and Upadhyay [FHU22] gave an efficient algorithm based on a subclass of *matrix mechanism* known as *factorization mechanism* and showed that its ℓ_∞ -error is almost tight for any matrix mechanism, not only in the asymptotic setting but even *with almost matching constants for the upper and lower bounds*. Concurrently to [FHU22], Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22] studied the ℓ_2^2 error for continual counting and gave conditions that a factorization has to fulfill to give an optimal ℓ_2^2 -error. They also proposed the use of a *fixed point algorithms* to compute the factorization, but they do not give an explicit factorization or any provable ℓ_2^2 -error bound of their mechanism.

On the other hand, given its application in real-world deployments mentioned above, designing an algorithm for continual counting with provable mean-squared error and one with smallest constant is highly desirable. The importance of having small constants was also recently pointed out by Fichtenberger, Henzinger, and Upadhyay [FHU22] in the continual observation model. This question was also the center of a subsequent work by Asi, Feldman, and Talwar [AFT22] on mean estimation in the *local model of privacy*. An algorithm with small constant in additive error means that we need to use less privacy budget (parameterized by ϵ and δ) to guarantee the same accuracy guarantee as an algorithm with larger constants in the additive error. This has huge impact in practice. For instance, real-world applications use prohibitively large values of ϵ (as large as 19.21 for the 2021 US Census [Cen] and 8.90 for private learning [MT22]) to keep the additive error small. In contrast, one would like ϵ to be small (ideally $\epsilon \leq 1$) – using large ϵ means we need to increase the sample size of the training data, and collecting data is often expensive. *Designing a fast mechanism with a small constant factor in the mean squared error is the central topic of this paper.*

Note that there are provable guarantees on the error for the binary mechanism [CSS11, DNPR10], but there are two fundamental issues with the binary mechanism which precludes its application in practice:

1. As we show in Theorem 5 and the subsequent paragraph, the mean squared error of the binary mecha-

¹We refer the interested readers to the Google AI blog for more details regarding the obstacle in actual deployment of theoretically optimal algorithms like differentially private stochastic gradient descent and one based on amplification by shuffling [MT22].

²This in particular means that it can be seamlessly extended to neural network where the loss functions are inherently non-convex.

nism is provably suboptimal.

2. The additive error (even for Honaker’s streaming version [Hon15]) is non-uniform and depends on the number of 1’s in the bitwise representation of the current time epoch, leading to a non-smooth error function ([DMR⁺22, Figure 1]). Consequently, the binary mechanism cannot be used in health-related applications such as ECG monitoring in the Apple watch, where “smooth” additive error functions are necessary.

In this paper, we also identify the fundamental reasons why the binary mechanism suffers from the above two major limitations and give algorithm that resolves them while ensuring the advantages of continual counting mentioned earlier so that it can be used in private online learning. More specifically

1. We give matrix mechanism for continual counting that achieves a mean squared error that is approximately a factor of $\frac{(\pi \log_2 e)^2}{2} \approx 10.2$ smaller than the binary mechanism. This algorithm can be implemented with quadratic pre-processing time and constant time per round.
2. We also show that our mechanism is almost optimal by giving the first tight lower bound on the ℓ_2^2 -error of continual counting for *any* mechanism that guarantees (ϵ, δ) -differential privacy. Combined with item 1, this resolves the first issue mentioned above.
3. Our mechanism adds Gaussian noise in a way that makes the error grow smoothly in the number of rounds, which resolves the second fundamental issue mentioned above.

1.1 Problem Statement and Our Contributions

Binary counting is a special type of a *linear query*, which is any linear function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ of the n -dimensional input vector $x \in \mathbb{R}^n$. A fixed set of q linear queries can be represented in the form of matrix $M \in \mathbb{R}^{q \times n}$ such that, for any n -dimensional input vector $x \in \mathbb{R}^n$ (given in a continual or non-continual manner), the answer for query i is $(Mx)[i]$ (the i -th coordinate of the vector Mx). Then the (*additive*) *mean-squared error* of an (ϵ, δ) -DP algorithm \mathcal{M} for answering q linear queries on an input $x \in \mathbb{R}^n$ described by the corresponding matrix, A , is

$$\text{err}_{\ell_2^2}(\mathcal{M}, A, n) = \max_{x \in \mathbb{R}^n} \mathbf{E}_{\mathcal{M}} \left[\frac{1}{n} \|\mathcal{M}(x) - Ax\|_2^2 \right]. \quad (1)$$

In this paper, we would be mainly interested in continual counting of a stream of length n . Let $A[i, j]$ denote the (i, j) -th entry of the matrix A , then the mean-squared error for binary counting is

$$\text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) = \max_{x \in \{0,1\}^n} \mathbf{E}_{\mathcal{M}} \left[\frac{1}{n} \|\mathcal{M}(x) - M_{\text{count}}x\|_2^2 \right], \text{ where } M_{\text{count}}[i, j] = \begin{cases} 1 & i \geq j \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

Our algorithm is an instantiation of the matrix mechanism [LMH⁺15], whose mean-squared error can be bounded in terms of a certain factorization norm, denoted by $\gamma_F(\cdot)$ [ENU20]. Our first set of contributions is concerned with understanding some key properties of this factorization norm for complex matrices. We believe these properties are of independent interest. Then we explore their application in the context of differential privacy.

1.1.1 Main Result

We first define $\gamma_F(\cdot)$ and explain its relationship to the mean squared error, which is the primary reason why we study $\gamma_F(\cdot)$ and its properties. In the following $\|A\|_{1 \rightarrow 2}$ denotes the maximum of the 2-norm of the columns of A and $\|A\|_F$ is the frobenius norm defined as

$$\|A\|_F = \left(\sum_{i=1}^{\min\{n,m\}} \sigma_i(A)^2 \right)^{1/2} = \left(\sum_{i \in [n]} \sum_{j \in [m]} |A[i, j]|^2 \right)^{1/2},$$

where $\sigma_i(A)$ is the i -th singular value of A .

Our mechanism for continual counting is a *matrix mechanism* [LMH⁺15], i.e., a mechanism where, given M , we first construct an alternate set of matrices known as *strategy matrix* R and *reconstruction matrix* L such that $M = LR$. The strategy matrix is used to generate a private vector, v , by adding a Gaussian noise vector to Rx . The answer to the original queries are then evaluated from v by computing Lv , which can be seen as a post-processing step. On input $x \in \mathbb{R}^n$, matrix mechanism outputs the following:

$$\mathcal{M}_{L,R}(x) = L(Rx + z), \quad \text{where } z \sim N\left(0, \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2 \mathbb{1}_m\right).$$

The privacy proof follows from known results [DMR⁺22, LMH⁺15]. For a matrix $M \in \mathbb{C}^{n \times m}$, let us define³

$$\gamma_F(M) = \min \{ \|L\|_F \|R\|_{1 \rightarrow 2} : LR = M \}. \quad (3)$$

Now, if $\mathcal{M}_{L,R}$ is a matrix mechanism that uses the factorization $M = LR$, then using Li, Miklau, Hay, McGregor, and Rastogi [LMH⁺15], we have

$$\text{err}_{\ell_2^2}(\mathcal{M}_{L,R}, M, n) = \frac{1}{n} C_{\epsilon, \delta}^2 \|L\|_F^2 \|R\|_{1 \rightarrow 2}^2. \quad (4)$$

In particular, for an optimal choice of L and R

$$\text{err}_{\ell_2^2}(\mathcal{M}_{L,R}, M, n) = \frac{1}{n} C_{\epsilon, \delta}^2 \gamma_F(M)^2. \quad (5)$$

We also investigate $\gamma_F(\cdot)$ in more detail for general complex matrices and show many useful properties in Appendix B. These are properties that may be of independent interest considering that $\gamma_F(\cdot)$ can be used to characterize the mean-squared error of linear queries [ENU20]. One of the main properties of $\gamma_F(\cdot)$ is that it can be characterized as a semidefinite programming (SDP) problem. We give the details in Section 3.

The SDP characterization allows us to prove many key properties of $\gamma_F(\cdot)$, which can be of independent interest. In particular, it allows us to prove the following key lemma that relates the $\gamma_F(\cdot)$ to the spectrum of the matrix.

Lemma 1. *For a matrix $A \in \mathbb{C}^{n \times m}$, we have $\frac{\|A\|_1}{\sqrt{m}} \leq \gamma_F(A) \leq \|A\|_F$, where $\|A\|_1$ is the Schatten-1 norm (or, trace norm) of A . In particular, if A is unitary, $\gamma_F(A) = \sqrt{n}$ and if all singular values of A are same, then $\gamma_F(A) = \|A\|_F$.*

Since the lower bound for mean-squared error for a set of linear queries can be stated in terms of the $\gamma_F(\cdot)$ of the corresponding query matrix, Lemma 1 provides an easier method to prove lower bounds. We explore two applications of this lower bound in this paper: continual counting and parity queries. A proof of this lemma is presented in Section 4.

The semidefinite characterization also allows us to show many useful facts about $\gamma_F(\cdot)$, which we believe can be of independent interest.

1. The optimal factorization can be achieved by finite dimensional matrices. This is a direct consequence of strong duality of the SDP of $\gamma_F(\cdot)$. We show strong duality in Lemma 42.
2. For a matrix $A \in \mathbb{C}^{n \times m}$, there exist $B \in \mathbb{C}^{n \times p}$ and $C \in \mathbb{C}^{p \times m}$ for $p \leq m$ such that $A = BC$ and $\gamma_F(A) = \|B\|_F \|C\|_{1 \rightarrow 2}$. If A is a real matrix, then we can assume without loss of generality that B and C are real matrices too.

While we establish properties of $\gamma_F(\cdot)$ for complex matrices, the second item allows us to assume that real matrices will have optimal real factorization. This is important for privacy applications where theoretical results have been established assuming real factorization. These and other useful properties of $\gamma_F(\cdot)$ are proved in Appendix B.

³Edmonds, Nikolov, and Ullman [ENU20] defined $\gamma_F(M) = \min \left\{ \frac{1}{\sqrt{n}} \|L\|_F \|R\|_{1 \rightarrow 2} : M = LR \right\}$ for a matrix $A \in \mathbb{C}^{n \times d}$. We prefer the definition in eq. (3) as it is more aligned with the definition of such norms in functional analysis and operator algebra.

1.1.2 Other Contributions

Contribution 1: An almost exact error bound for continual counting. The classic algorithm for differentially private counting under continual observation is the *binary (tree) mechanism* [CSS11, DNPR10]. With Laplacian noise they show for *each round* that the additive ℓ_∞ -error is $O(\log^{3/2} n)$ with constant probability, which requires the use of a union bound over all n updates and results in an ℓ_∞ -error of $O(\log^{5/2} n)$. However, with Gaussian noise an ℓ_∞ -error of $O(\log^{3/2} n)$ can be achieved [JRSS21]. Neither work gives a bound on the ℓ_2^2 -error although an $O(\log^2(n))$ bound is implicit in some works [KMS⁺21]. Note that the concurrent and independent work by Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22] do not give any bounds on the additive error of their matrix mechanism based algorithm and only show empirical improvement.

Our algorithm factorizes the matrix M_{count} in terms of two lower triangular matrices L and R , i.e. $M_{\text{count}} = LR$, and we show that $\|L\|_F \|R\|_{1 \rightarrow 2} \leq \sqrt{n} \left(1 + \frac{\ln(4n/5)}{\pi}\right)$. This immediately implies an upper bound on $\gamma_F(M_{\text{count}})$.

In particular, we show the following in Section 5.1:

Theorem 2. For any $0 < \epsilon, \delta < 1$, there is an efficient (ϵ, δ) -differentially private continual counting algorithm $\mathcal{M}_{\text{fact}}$, that on receiving a binary stream of length n , achieves the following error bound:

$$\text{err}_{\ell_2^2}(\mathcal{M}_{\text{fact}}, M_{\text{count}}, n) \leq C_{\epsilon, \delta}^2 \left(1 + \frac{\ln(4n/5)}{\pi}\right)^2, \quad \text{where } C_{\epsilon, \delta} = \frac{2}{\epsilon} \sqrt{\frac{4}{9} + \ln\left(\frac{1}{\delta} \sqrt{\frac{2}{\pi}}\right)} \quad (6)$$

is the variance required by the Gaussian mechanism to preserve (ϵ, δ) -differential privacy and $\ln(\cdot)$ denotes the natural logarithm. The mechanism requires $O(n^2 + ns)$ preprocessing time and constant time per update round, where s is the time required to sample from zero-mean unit variance Gaussian distribution.

We also show an almost tight lower bound on $\gamma_F(M_{\text{count}})$. This implies that for any matrix mechanism based algorithm [LMH⁺15] for continual counting and for small enough (ϵ, δ) , our bound is almost tight. While the limitation to matrix mechanism based algorithms seems restrictive, *all currently known* mechanisms for continual observation fall under this class of mechanism (see Section 6 for an explanation). A full proof of Theorem 3 is presented in Section 5.2.

Theorem 3 (Lower bound on matrix mechanisms.). For any $\epsilon > 0$ and $0 \leq \delta \leq 1$, let \mathfrak{M} be the set of (ϵ, δ) -differentially private continual counting algorithms that use the matrix mechanism. Then

$$\min_{\mathcal{M} \in \mathfrak{M}} \text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{C_{\epsilon, \delta}^2}{\pi^2} \left(2 + \ln\left(\frac{2n+1}{3}\right) + \frac{\ln(2n+1)}{2n}\right)^2.$$

Note that the constants in Theorem 2 and 3 match exactly for the $(\ln(n))^2$ term and the bounds only differ in the constants in lower-order terms. More concretely, for all $n \leq 2^{50}$, the additive gap between the upper bound (Theorem 2) and lower bound (Theorem 3) is at most $10C_{\epsilon, \delta}^2$.

Contribution 2: A lower bound on the ℓ_2^2 -error for any mechanism for continual counting. Theorem 3 precludes an improvement using matrix mechanism, but does not preclude algorithms using a more careful choice of noise addition as the only known lower bound for continual counting is $\Omega(\log(n))$ for ℓ_∞ -error when $\delta = 0$ [DNPR10]. More generally, there is no lower bound known on the ℓ_2^2 -error and $\delta \neq 0$. These facts lead to the natural question, recently also asked by Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22]: *Is there a mechanism that is not factorization-based and achieves a better mean-squared error?* We show this is not the case by proving the following theorem in Section 5.2, which also implies that our mechanism is asymptotically optimal.

Theorem 4 (Lower bound on the ℓ_2^2 -error of continual counting). For any $\varepsilon > 0$ and $0 \leq \delta < \frac{c}{2e^\varepsilon}$ for some absolute constant $c > 0$, let \mathfrak{M} be the set of (ε, δ) -differentially private algorithms for counting under continual observation. Then for all n ,

$$\min_{\mathcal{M} \in \mathfrak{M}} \text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{1}{(e^{4\varepsilon} - 1)^2 \pi^2} \left(2 + \ln \left(\frac{2n+1}{3} \right) + \frac{\ln(2n+1)}{2n} \right)^2. \quad (7)$$

Further, if \mathfrak{M} is a set of (ε, δ) -differentially private mechanism for continual counting that add noise oblivious of the input for (ε, δ) small enough constant, then we can improve the dependency on privacy parameter:

$$\min_{\mathcal{M} \in \mathfrak{M}} \text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{1}{(e^{2\varepsilon} - 1)^2 \pi^2} \left(2 + \ln \left(\frac{2n+1}{3} \right) + \frac{\ln(2n+1)}{2n} \right)^2.$$

Contribution 3: Suboptimality of the binary mechanism. Few natural questions to ask are whether we can improve the accuracy of the binary mechanism using a better analysis and how much worse the additive factor in the binary mechanism is than our mechanism. We answer these in the following theorem:

Theorem 5. Let \mathcal{M}_B be the binary (tree) mechanism [CSS11, DNPR10] that adds noise sampled from an appropriate Gaussian distribution to every node of the binary tree. Let $\mathcal{M}_{\text{fact}}$ be our mechanism guaranteeing Theorem 2. Then

$$\frac{\text{err}_{\ell_2^2}(\mathcal{M}_B, M_{\text{count}}, n)}{\text{err}_{\ell_2^2}(\mathcal{M}_{\text{fact}}, M_{\text{count}}, n)} \geq \frac{\log_2(n) (1 + \log_2(n))}{2 \left(1 + \frac{\ln(4n/5)}{\pi} \right)^2}.$$

Let \mathfrak{M} be the set of (ε, δ) -differentially private continual counting algorithms that use a matrix mechanism, and let $\mathcal{M} \in \mathfrak{M}$ be a matrix mechanism that achieves the optimal error stated in Theorem 3. Then

$$\frac{\text{err}_{\ell_2^2}(\mathcal{M}_B, M_{\text{count}}, n)}{\text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n)} = \frac{\pi^2 \log_2(n) (1 + \log_2(n))}{2 \left(2 + \ln \left(\frac{2n+1}{3} \right) + \frac{\ln(2n+1)}{2n} \right)^2}.$$

In short, the above theorem states that the mean-squared error of binary mechanism is suboptimal by approximately a factor $\frac{1}{2} \left(\frac{\pi}{\ln 2} \right)^2 \approx 10.2$. A plot of this comparison is given in Figure 1 for varying ε and n up to 2^{40} with $\delta = 10^{-10}$. A proof of Theorem 5 is presented in Section 6.

Contribution 4: Online Private Learning. A major application of differentially private continual counting with bounded mean squared error is private learning [KMS⁺21]. Here, the goal is to minimize the *excess risk*, i.e., either minimize the *average loss* on the given data (*excess empirical risk*) or minimize the error on “unseen” data (*excess population risk*). Smith and Thakurta [TS13] introduced a technique for online private learning using continual counting as a subroutine. In *online* learning, we aim to bound the *regret*, i.e., the average loss incurred over all rounds compared to the post-hoc optimal decision (see Definition 29). One can then use the standard technique of *online-to-batch* conversion to get a bound on population risk from the regret bound.

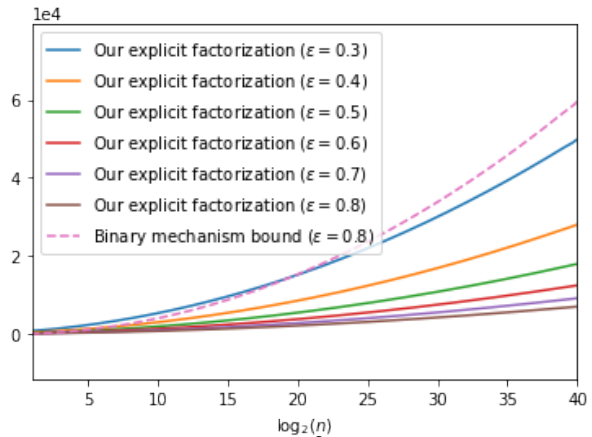


Figure 1: Comparison of ℓ_2^2 -error bounds of binary and our factorization based mechanisms for varying ε and $\delta = 10^{-10}$. Our mechanism ($\varepsilon = 0.3$) incurs less error than binary mechanism ($\varepsilon = 0.8$) for $n \geq 2^{19}$.

This approach was recently used by Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21] and a follow-up work by Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22] – the difference being that [KMS⁺21] used the binary mechanism as a subroutine and got a provable regret bound while [DMR⁺22] suggested the matrix mechanism and show regret improvements only empirically. These algorithms have direct practical applications, see the extensive experiments mentioned in [DMR⁺22] and the Google AI blog detailing the use of binary mechanism in their recent deployment [MT22]. Thus, it is important to have provable guarantees on such an algorithm including constant factors. Since the *online-to-batch* conversion is standard, we focus only on giving the regret bound. We show the following non-asymptotic bound on the algorithm of Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22, Algorithm 1] with the continual counting algorithm implemented by our mechanism:

Theorem 6. *Let \mathcal{K} be a closed, convex, and compact set and \mathcal{D} be the data universe. Further, let $\ell : \mathcal{K} \times \mathcal{D} \rightarrow \mathbb{R}$ be 1-Lipschitz with respect to ℓ_2 norm and convex in the first parameter, i.e., $\|\ell(x; \cdot) - \ell(x'; \cdot)\|_2 \leq \|x - x'\|_2$ for all $x, x' \in \mathcal{K}$. Then there is an efficient (ϵ, δ) -differentially private online algorithm, A_{OCO} , that on a stream of data x_1, \dots, x_n , outputs private models $[\theta_1, \dots, \theta_n]$ such that, for any $\theta_{\text{opt}} \in \mathcal{K}$:*

$$\text{Regret}(A_{\text{OCO}}; n) \leq \|\theta_{\text{opt}}\|_2 \sqrt{\frac{\left(1 + \frac{\ln(4n/5)}{\pi}\right) (1 + C_{\epsilon, \delta} \sqrt{d})}{2n}}, \quad (8)$$

where

$$\text{Regret}(A_{\text{OCO}}; n) = \frac{1}{n} \sum_{i=1}^n \ell(\theta_i; x_i) - \frac{1}{n} \min_{\theta \in \mathcal{K}} \sum_{i=1}^n \ell(\theta; x_i).$$

Furthermore, the regret bound in eq. (8) holds even if the data points x_t are picked adversarially.

This result shows that our algorithm improves the algorithm in Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21, Theorem 4.1] by a constant factor of about 3 (computed from the constants in their proof) and helps explain the empirical observation made in Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22], who reported that the matrix mechanism based stochastic gradient descent “*significantly improve the privacy/utility curve (in fact, closing 2/3rds of the gap to non-private training left by the previous state-of-the-art for single pass algorithms)*”, where the previous state-of-the-art algorithm refers to the one by Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21]. A proof of Theorem 6 is given in Section 7.

Remark 7. *As noted in [DMR⁺22], private learning algorithms that use continual counting are also flexible to the various settings studied in Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21]; therefore, our results extend seamlessly to adversarial regret for composite loss functions, excess risk, and various practical extensions such as heavy ball momentum. We refer the interested readers to the relevant sections in [DMR⁺22, KMS⁺21]. Further, our factorization is also diagonally dominant allowing the efficient computation in practical settings such as in [DMR⁺22, Appendix F].*

Contribution 5: Lower Bounds on Special Linear Queries Our lower bound technique is actually quite general and will most likely have further applications. To exhibit the generality of our lower bound technique, we use it to show another lower bound, this time in the non-continual setting. Specifically, we give the first lower bound on the mean-squared error for *parity queries* in the batch, i.e., non-continual setting, where the underlying data does not change.

Definition 8 (Parity Query). *Let d and w be integer parameters and let the domain be $\mathcal{X} = \{\pm 1\}^d$. Then a parity query is a query that belongs to the family of queries*

$$\mathcal{Q}_{d,w} = \left\{ q_P(x) = \prod_{i \in P} x_i : P \subset \{1, \dots, d\}, |P| = w \right\}. \quad (9)$$

Note that parity queries are important in data analysis. We show the following bound for parity queries, which to our knowledge, is the first lower bound on the mean-squared error for parity queries issued by a non-adaptive adversary under (ϵ, δ) -differential privacy and it is tight: Answering parity queries with the Gaussian mechanism achieves the same dependency in terms of d and w as our lower bound.

Theorem 9. *Let $\epsilon > 0$, let $0 \leq \delta \leq 1$ and let $\mathcal{Q}_{d,w}$ be the class of parity queries defined in eq. (9). Then for any (ϵ, δ) -differentially private mechanism \mathcal{M} that takes as input d and w , and that can answer any query of $\mathcal{Q}_{d,w}$, it holds that*

$$\text{err}_{\ell_2^2} \left(\mathcal{M}, \mathcal{Q}_{d,w}, \begin{pmatrix} d \\ w \end{pmatrix} \right) \geq C_\epsilon^2 \begin{pmatrix} d \\ w \end{pmatrix}.$$

A proof of Theorem 9 is given in Section 5.2. A corresponding bound for ℓ_∞ error was computed by Edmonds, Nikolov, and Ullman [ENU20]. This completes the picture for parity queries.

1.2 Our Techniques

We fix some notation that we use in this section (detail notations are in Section 2). For a matrix X and a vector v , let X^* and v^* denote their complex-conjugates (when X is a real matrix, then X^* is the transposed matrix), respectively. For a complex number, $z = a + ib \in \mathbb{C}$, let $|z|$ denote its *modulus*, $a^2 + b^2$. Moreover, let $\mathbb{1}_k$ denote a $k \times k$ identity matrix, $J_{k,l}$ denote the all ones $k \times l$ matrix, and $\mathbf{1}_k$ denote the vector of all ones in \mathbb{R}^k . For any two matrices $P, Q \in \mathbb{C}^{n \times m}$, let $P \bullet Q$ denote their Schur (Hadamard) product. We use $A \succeq 0$ to denote that A is a positive semidefinite (PSD) matrix and $A \succeq B$ to denote that $A - B \succeq 0$. We use $w \in \mathbb{R}_{++}^n$ to denote a strictly positive vector. Finally, for any matrix $Y \in \mathbb{C}^{n \times m}$, let \hat{Y} denote the following Hermitian matrix:

$$\hat{Y} = \begin{pmatrix} 0^{n \times n} & Y \\ Y^* & 0^{m \times m} \end{pmatrix}.$$

1.2.1 Main Result: Useful Properties of $\gamma_F(\cdot)$

Fix $A \in \mathbb{C}^{n \times m}$ for which we wish to characterize $\gamma_F(\cdot)$ as an SDP⁴. Note that, for any factorization $A = LR$, we can assume that $\|L\|_F = \|R\|_{1 \rightarrow 2}$ by appropriate rescaling. That is, for an optimal factorization $A = LR$ with respect to $\gamma_F(A)$, we can assume that

$$\gamma_F(A) = \|L\|_F^2 = \|R\|_{1 \rightarrow 2}^2. \quad (10)$$

Any factorization of $A = LR$ can be turned into a PSD matrix satisfying the following matrix constraint:

$$X = \begin{pmatrix} X_1 & X_2 \\ X_2^* & X_3 \end{pmatrix} \succeq 0 \quad \text{such that} \quad X_2 = A = LR.$$

The fact that $X \succeq 0$ means that $X_1 = LL^*$ and $X_3 = R^*R$. The first implication of this fact is that $\text{Tr}(X_1) = \text{Tr}(LL^*) = \|L\|_F^2$, where $\text{Tr}(Z)$ denotes the sum of diagonal entries of any square matrix Z . The second implication is that the i -th diagonal entry of X_3 , denoted by $X_3[i, i]$, is exactly the squared 2-norm of the i -th column of R . The maximum of the 2-norm over each column of R is exactly $\|R\|_{1 \rightarrow 2}$ (see Fact 17). From eq. (10), we wish to minimize $\|L\|_F^2$ such that every diagonal entry of the matrix X_3 is at most $\|L\|_F^2$. Hence, we can rewrite $\gamma_F(A)$ as minimizing a real number η such that, for the matrix

$$X = \begin{pmatrix} X_1 & X_2 \\ X_2^* & X_3 \end{pmatrix} \succeq 0 \quad \text{satisfying} \quad X_2 = A = LR, \quad \text{we have} \quad \text{Tr}(X_1) = \eta \quad \text{and} \quad X_3[i, i] \leq \eta$$

for every $i \in \{1, \dots, m\}$. This can be described as an SDP as described in Figure 2 (the primal problem).

⁴We give the SDP characterization for complex matrices which will involve Hermitian matrices. However, as stated above, when A is a real matrix, one can without loss of generality consider an SDP involving symmetric matrices.

Primal	Dual
$\gamma_F(A) := \min \quad \eta$	$\gamma_F(A) = \max \quad w^* (\hat{A} \bullet \hat{Z}) w$
$\text{s.t.} \quad \sum_{i=1}^n X[i, i] = \eta$	$\text{s.t.} \quad \begin{pmatrix} n\mathbb{1}_n & 0 \\ 0 & \mathbb{1}_m \end{pmatrix} \succeq \hat{Z}$
$X[i, i] \leq \eta \quad \forall n+1 \leq i \leq n+m$	$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbb{R}_{+++}^{n+m}$
$X \bullet \hat{J}_{n,m} = \hat{A}$	$\ w\ _2 = 1$
$X \succeq 0.$	$w_1 = \alpha \mathbb{1}_n$

Figure 2: SDP for $\gamma_F(\cdot)$ norm.

As SDPs come in primal-dual pairs, any feasible solution of the primal problem is an upper bound on $\gamma_F(A)$. Similarly, any feasible solution of the dual problem is a lower bound on $\gamma_F(A)$. We will utilize this fact to show the desired lower bound as stated in Lemma 1. We give the detail proof in Section 4. Note that the dual problem as stated in Figure 2 is in a form that will be helpful in proving the lower bound. A rigorous explanation of how we arrive at this formulation is described in Appendix A.

1.2.2 Other Contributions

Contribution 1: An almost exact error bound for continual counting. As described above, we analyze the matrix mechanism given in Algorithm 1. From eq. (5), the question of determining lower and upper bounds on the ℓ_2^2 -error for continual counting reduces to a purely linear algebraic problem of estimating $\gamma_F(M_{\text{count}})$. There are many ways of estimating this quantity. One particular way to bound it is by using the *completely bounded spectral norm* [Pau82]:

$$\|A\|_{\text{cb}} := \min \{ \|L\|_{2 \rightarrow \infty} \|R\|_{1 \rightarrow 2} : A = LR \},$$

where $\|L\|_{2 \rightarrow \infty}$ is the maximum of the 2-norm of the rows of L . The $\|\cdot\|_{\text{cb}}$ norm plays an important role in bounding the ℓ_∞ -error [FHU22]. It has been extensively studied in operator algebra and tight bounds are known for $\|M_{\text{count}}\|_{\text{cb}}$ [Mat93]. However, using known bounds for $\|M_{\text{count}}\|_{\text{cb}}$ does not yield a tight bounds on $\gamma_F(M_{\text{count}})$. It is known that, for a matrix $A \in \mathbb{C}^{n \times n}$, $\|A\|_{\text{cb}} \leq \gamma_F(A) \leq \sqrt{n} \|A\|_{\text{cb}}$, and as we will show later, the gap between $\|M_{\text{count}}\|_{\text{cb}}$ and $\gamma_F(M_{\text{count}})$ is indeed approximately \sqrt{n} . Hence, we utilize different techniques, as described below, to show the following almost tight bounds on $\gamma_F(M_{\text{count}})$:

$$\frac{1}{\pi} \left(2 + \ln \left(\frac{2n+1}{3} \right) + \frac{\ln(2n+1)}{2n} \right) \leq \frac{1}{\sqrt{n}} \gamma_F(M_{\text{count}}) \leq \left(1 + \frac{\ln(4n/5)}{\pi} \right). \quad (11)$$

The error bounds of Theorem 2 and Theorem 3 follows by combining eq. (5) and eq. (11). Furthermore the factorization achieving the upper bound is given by two lower triangular matrices L and R .

While the matrix mechanism presented in Algorithm 1 requires $O(t)$ time at round t (L_t on line 5 can be computed in time $O(t)$ using eq. (12)), we show how to modify it to achieve constant time per round and $O(n^2)$ pre-processing time. The main idea is to sample a vector g from a suitable distribution during pre-processing, multiplying L with g , and storing the resulting vector $z = Lg$. When bit x_t is released, the mechanism simply adds $z[t]$ to the true answer. We also show how to adjust the privacy proof to this setting using Theorem 35 in Section 5.1. Thus, in what follows, we just discuss the technique to prove eq. (11).

Upper bound in eq. (11). We bound $\|R\|_{1 \rightarrow 2}$ and $\|L\|_F$ for L and R computed in Algorithm 1. We bound $\|R\|_{1 \rightarrow 2}$ using the fact that entries of R can be represented as a double factorial allowing us to use Theorem 21 to get $\|R\|_{1 \rightarrow 2}^2 \leq \left(1 + \frac{\ln(4n/5)}{\pi} \right)$. To bound $\|L\|_F$, we use the fact that L is a lower-triangular matrix

Algorithm 1 Matrix Mechanism for Continual Counting, $\mathcal{M}_{\text{fact}}$

Require: A stream of bits (x_1, \dots, x_n) , length of the stream n , (ϵ, δ) : privacy budget.

1: Define a function $f : \mathbb{Z} \rightarrow \mathbb{R}$ as follows:

$$f(k) = \begin{cases} 0 & k < 0 \\ 1 & k = 0 \\ \left(1 - \frac{1}{2k}\right) f(k-1) & k \geq 1 \end{cases} \quad (12)$$

2: Let $L, R \in \mathbb{R}^{n \times n}$ be matrices with entries as follows: $L[i, j] = f(i - j)$ and $R[i, j] = f(i - j)$.

3: **for** t in $1, 2, \dots, n$ **do**

4: Sample $z \sim N\left(0, C_{\epsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2 \mathbb{1}_t\right)$.

5: Define a t -dimensional row vector $L_t = (L[t, 1] \ L[t, 2] \ \dots \ L[t, t])$.

6: Receives x_t and output

$$a_t = \left(\sum_{i=1}^t x_i \right) + \langle L_t, z \rangle$$

7: **end for**

and the ℓ_2^2 norm of the m -th row of L (denoted by $L[m, :]$) is the same as the $\|L(m)\|_{2 \rightarrow \infty}^2 = \|L(m)\|_{1 \rightarrow 2}^2$ norm of the $m \times m$ principal submatrix, $L(m)$, of L . That is, $\|L[m, :]\|_2^2 \leq \left(1 + \frac{\ln(4m/5)}{\pi}\right)$. In particular, we can bound

$$\|L\|_F^2 = \sum_{m=1}^n \|L[m, :]\|_2^2 \leq \sum_{m=1}^n \left(1 + \frac{\ln(4m/5)}{\pi}\right) \leq n \left(1 + \frac{\ln(4n/5)}{\pi}\right).$$

A complete proof is presented in Section 5.1.

Lower bound in eq. (11). We begin with a brief outline of the algebraic method used to lower bound $\gamma_F(M_{\text{count}})$. For a general matrix A , it is possible to show that $\gamma_F(A)$ is lower bounded by the square of the sum of a subset S of singular values. If the singular values are ordered in descending order $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$, then the subset S contains exactly the singular values $\sigma_1, \sigma_3, \dots$. This is a consequence of Cauchy-Schwarz and Weyl's inequalities. With this at our disposal, we can then use standard results on the singular values of M_{count} to give a lower bound. However, this does not yield a tight bound. In particular, the slackness in the lower bound results from the application of Cauchy-Schwarz and Weyl's inequalities [MK04] in the first step. To overcome this slackness, we take an optimization perspective and use our SDP characterization of $\gamma_F(\cdot)$. We note that such an optimization perspective has been taken in some of the earlier works (see [ENU20] and references therein). We present a complete proof of Theorem 3 in Section 5.2.

Contribution 2: Lower bound for any mechanism for continual counting. Our lower bound on mechanism for continual counting that uses factorization of M_{count} follows from our lower bound on $\gamma_F(M_{\text{count}})$. To extend this to a lower bound on all (ϵ, δ) -differentially private mechanism for continual counting, we use the lower bound on the mean-squared error by Edmonds, Nikolov, and Ullman [ENU20] (see the proof of Theorem 36): the lower bound on any (ϵ, δ) -differentially private mechanism for any linear query defined by a matrix A is at least $C_\epsilon^2 \gamma_F(A)^2 / n$, where $C_\epsilon = \frac{1}{e^{2\epsilon} - 1}$. We note that the value of the constant C_ϵ can be found in a lower bound by Kasivishwanathan, Rudelson, Smith, and Ullman [KRSU10]. Combining this with eq. (11), we get Theorem 4.

Contribution 3: Suboptimality of the binary mechanism. The binary mechanism returns in each round t the sum of $O(\log n)$ sub-sums, called p -sums, depending on the number of bits set in the binary representation of t . Now each row of the right factor R_{binary} is used to sum up each p -sum, while each row of the left factor L_{binary} is used to compute the sum of the $O(\log n)$ p -sums. More formally, the right factor R_{binary} is constructed as follows: $R_{\text{binary}} = W_m$ where W_1, \dots, W_m are defined recursively as follows:

$$W_1 = (1), \quad W_k = \begin{pmatrix} W_{k-1} & 0 \\ 0 & W_{k-1} \\ \mathbf{1}_{2^{k-2}} & \mathbf{1}_{2^{k-2}} \end{pmatrix}, \quad k \leq m.$$

Note that $R_{\text{binary}} = W_m$ is a matrix of $\{0, 1\}^{n \times (2^{m-1})}$ matrix, with each row corresponding to the p -sum computed by the binary mechanism. The corresponding matrix L_{binary} is a matrix of $\{0, 1\}^{n \times (2^{m-1})}$, where row t has $\log_2(t)$ entries, corresponding exactly to the binary representation of t . Computing the $\|L_{\text{binary}}\|_F$ and $\|R_{\text{binary}}\|_{1 \rightarrow 2}$ leads to the bounds stated in the theorem, which combined with our results, implies the suboptimality of the binary mechanism.

Contribution 5. Our SDP-based lower bound technique is very general and can be applied even in the non-continual counting, as we show by using it to give a lower bound for parity queries. In particular, we use the observation of Edmonds, Nikolov, and Ullman [ENU20] that the query matrix corresponding to any set of the parity queries is the $\binom{d}{w}$ matrix formed by taking the corresponding rows of the $2^d \times 2^d$ unnormalized Hadamard matrix. Let us call this matrix S . The lower bound then follows by computing the Schatten-1 norm of S . We present a complete proof in Section 8.

Outline of the paper. We give all necessary notation and preliminaries in Section 2 and present the semidefinite program for γ_F in Section 3. Section 5 contains the upper and lower bound on $\gamma_F(M_{\text{count}})$ and also the more efficient mechanism, thus giving the proof of both Theorem 2, Theorem 3, and Theorem 4. In Section 6 we show that every known mechanism for continual counting is a matrix mechanism and give lower bounds for the mean squared error of the binary mechanism, thereby proving Theorem 5. Section 7 contain all upper and lower bounds for the further applications. Appendix A gives the dual characterization of $\gamma_F(\cdot)$ and Appendix B covers the useful properties and bounds on $\gamma_F(\cdot)$.

2 Notations and Preliminaries

We use \mathbb{N} to denote the set of natural numbers, \mathbb{Z} to denote the set of integers, \mathbb{R} to denote the set of real numbers, \mathbb{R}_+ to denote set of non-negative real numbers, \mathbb{R}_{++} to denote set of positive real numbers, and \mathbb{C} to denote the set of complex numbers. For $n, m \in \mathbb{N}$ such that $m \leq n$, we let $[n]$ denote the set $\{1, \dots, n\}$, and $[m, n]$ denote the set $\{m, \dots, n\}$. We will reserve the lower-case alphabets n, m, p, q, r for describing natural numbers and i, j, k for indexing. We fix the symbol n to denote the length of the stream.

2.1 Linear Algebra

In this section, we review linear algebra and relevant facts and describe the notations used throughout the paper.

Vector spaces and norms. We denote n -dimensional real vector space and complex vector space by \mathbb{R}^n and \mathbb{C}^n , respectively. The non-negative orthant and the set of n -dimensional strictly positive vectors in \mathbb{R}^n are denoted \mathbb{R}_+^n and \mathbb{R}_{++}^n , respectively. We will reserve the lower-case alphabets u, v, w, x, y, z to denote vectors in \mathbb{R}^n or \mathbb{C}^n . The i -th co-ordinate of a vector v is denoted $v[i]$ and the set $\{e_i : i \in [n]\}$ denote the set

of standard basis vectors. We reserve $\mathbf{1}_n$ to denote the vector of all 1's in \mathbb{R}^n . When a complex (real) vector space is equipped by a inner product, it is called a complex (real) inner product space. The canonical inner product associated with a complex vector space \mathbb{C}^n is defined as

$$\langle u, v \rangle = \sum_{i \in [n]} u^*[i]v[i],$$

for any two vectors $u, v \in \mathbb{C}^n$ where u^* is the vector whose entries are complex-conjugate of the entries of vector u . The following norms will be used in this paper (termed as ℓ_2, ℓ_1 , and ℓ_∞ norms, respectively):

$$\|u\|_2 = \sqrt{\langle u, u \rangle} \quad \text{and} \quad \|u\|_1 = \sum_{i \in [n]} |u[i]| \quad \text{and} \quad \|u\|_\infty = \max_{i \in [n]} \{|u[i]|\}.$$

In one of our applications, we will also need the concept of *dual norm*.

Definition 10 (Dual norm). *Let $\|\cdot\|$ be any norm on \mathcal{K} . Then its associated dual norm is defined as follows:*

$$\|z\|_* := \sup \{\langle z, x \rangle : \|x\| \leq 1\}$$

It is easy to see that ℓ_2 norm is dual of itself and $\|\cdot\|_1$ is the dual of $\|\cdot\|_\infty$.

Matrices. The vector space of complex $n \times m$ matrices is denoted by $\mathbb{C}^{n \times m}$. The set of real $n \times m$ matrices form a subspace of $\mathbb{C}^{n \times m}$ and is denoted $\mathbb{R}^{n \times m}$. For a matrix A , its (i, j) -th entry is denoted by $A[i, j]$, the i -th row is denoted $A[i;]$, and the j -th column is denoted $A[:, j]$. We use the notation $J_{n, m}$ to denote an all one $n \times m$ matrix, J_n to denote $J_{n, n}$, $\mathbb{1}_n$ to denote the $n \times n$ identity matrix, and $0^{n \times m}$ to denote an $n \times m$ all zero matrix.

The complex-conjugate of A is denoted by A^* . The complex-conjugate of a real matrix B is the transpose of the matrix itself, and we will employ the notation B^* to denote the transposed matrix. We will be mostly referring to the following classes of matrices in the remainder of this paper.

1. A matrix $A \in \mathbb{C}^{n \times n}$ is *normal* if $AA^* = A^*A$. The set of normal matrices is denoted $N(\mathbb{C}^n)$. The eigenvalues of A can be complex. The singular values of A are just the absolute value of the respective eigenvalues.
2. A normal matrix U is *unitary* if it also satisfies $UU^* = \mathbb{1}_n$, where $\mathbb{1}_n$ is the $n \times n$ identity matrix. The set of unitary matrices is denoted $U(\mathbb{C}^n)$. The eigenvalues of a unitary matrix lie on the unit circle in a complex plane. In other words, every singular value of a unitary matrix is 1.
3. A matrix $A \in \mathbb{C}^{n \times n}$ is *Hermitian* if $A = A^*$. The set of Hermitian matrices is denoted $\text{Herm}(\mathbb{C}^n)$. If the entries of a Hermitian matrix A are real, we call the matrix *symmetric*. The eigenvalues of a Hermitian matrix are real.
4. A Hermitian matrix A is *positive semidefinite* if all its eigenvalues are non-negative. The set of such matrices is denoted $\text{Pos}(\mathbb{C}^n)$. The notation $A \succeq 0$ indicates that A is positive semidefinite and the notations $A \succeq B$ and $B \preceq A$ indicate that $A - B \succeq 0$ for Hermitian matrices A and B .
5. A positive semidefinite matrix A is *positive definite* if all its eigenvalues are strictly positive. The set of such matrices is denoted $\text{Pd}(\mathbb{C}^n)$. The notation $A \succ 0$ indicates that A is positive definite and the notations $A \succ B$ and $B \prec A$ indicate that $A - B \succ 0$ for Hermitian matrices A and B .

Remark 11. *For any matrix $A \in \mathbb{C}^{n \times n}$, its singular values and eigenvalues are denoted by the sets $\{\sigma_i(A) : i \in [n]\}$ and $\{\lambda_i(A) : i \in [n]\}$. Throughout this paper we follow the following convention.*

1. *For a matrix A , its singular values are sorted in descending order. That is, $\sigma_1(A) \geq \dots \geq \sigma_n(A) \geq 0$. The eigenvalues of A are ordered with respect to the ordering of the singular values of A . In other words, $\lambda_i(A)$ corresponds to the i -th singular value in the sorted list.*

2. We will adopt a different convention for Hermitian matrices. For a Hermitian matrix $A \in \text{Herm}(\mathbb{C}^n)$, the eigenvalues are real and are sorted in descending order: $\lambda_1(A) \geq \dots \geq \lambda_n(A)$.

For any matrix $A \in \mathbb{C}^{n \times m}$, we reserve the notation \widehat{A} to denote the following matrix:

$$\widehat{A} = \begin{pmatrix} \mathbf{0}^{n \times n} & A \\ A^* & \mathbf{0}^{m \times m} \end{pmatrix}. \quad (13)$$

The matrix \widehat{A} is a Hermitian matrix (or symmetric, if $A \in \mathbb{R}^{n \times n}$). The trace of a square matrix $A \in \mathbb{C}^{n \times n}$ is denoted $\text{Tr}(A)$ and equals $\sum_{i \in [n]} A[i, i]$. For two matrices $A, B \in \mathbb{C}^{n \times n}$, their inner product is defined as

$$\langle A, B \rangle = \text{Tr}(A^* B).$$

For a vector $v \in \mathbb{C}^n$, we let $\Delta_n : \mathbb{C}^n \rightarrow \mathbb{C}^{n \times n}$ denote the map that maps an n -dimensional vector to a diagonal matrix with diagonal entries being the entries of the vector. For two matrices $A, B \in \mathbb{C}^{n \times m}$, we denote their Hadamard (or Schur) product as $A \bullet B$. We list the following well known properties of the Schur product that is used in this paper.

Proposition 12. *Let $A, B \in \mathbb{C}^{n \times n}$. Then $\langle J_n, A \bullet B \rangle = \langle A, B \rangle = \text{Tr}(A^* B)$. Moreover, if $A \bullet B \in \text{Herm}(\mathbb{C}^n)$, then $\langle A \bullet B, J_n \rangle = \langle A, B \rangle = \text{Tr}(A^* B)$.*

Proof. Let $D = A \bullet B$. Then $\langle J_n, D \rangle = \text{Tr}(J_n D) = \sum_{i, j \in [n]} D[i, j] = \sum_{i, j \in [n]} A[i, j] B[i, j] = \langle A, B \rangle = \text{Tr}(A^* B)$. If $A \bullet B \in \text{Herm}(\mathbb{C}^n)$, it holds that $\langle J_n, A \bullet B \rangle = \langle A \bullet B, J_n \rangle$, and the result follows from above. \square

Proposition 13. *Let $A, B \in \text{Herm}(\mathbb{C}^n)$ and $v \in \mathbb{C}^n$. Then $\langle A, B \bullet vv^* \rangle = v^*(A \bullet B)v$.*

Proof. Unraveling the formula

$$\langle A, B \bullet vv^* \rangle = \sum_{i, j \in [n]} A[i, j] B[i, j] v[i] v^*[j] = \sum_{i, j \in [n]} v^*[j] A^*[j, i] B^*[j, i] v[i] = v^*(A^* \bullet B^*)v = v^*(A \bullet B)v.$$

This completes the proof of Proposition 13. \square

Proposition 14. *Let $x, y \in \mathbb{R}^n$, and D_x and D_y be diagonal matrices formed by vectors x and y , respectively, Then for any matrix $A \in \mathbb{C}^{n \times n}$, $D_x A D_y = A \bullet xy^*$.*

Proof. A straightforward calculation will show that the (i, j) entry of the matrix $D_x A D_y$ is $A[i, j] x[i] y[j]$. This implies that $D_x A D_y = A \bullet xy^*$ and the proposition follows. \square

We need the following definition and the subsequent well known lemma for our proof.

Definition 15 (Schur's complements). *Let $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, $C \in \mathbb{C}^{m \times n}$, and $D \in \mathbb{C}^{m \times m}$ be matrices and let*

$$S = \begin{pmatrix} A & B \\ C & D \end{pmatrix}. \quad (14)$$

Then the Schur complements of S are the matrices

$$S_A = D - CA^{-1}B \quad (\text{if } A^{-1} \text{ exists}) \quad \text{and} \quad S_D = A - BD^{-1}C \quad (\text{if } D^{-1} \text{ exists}). \quad (15)$$

A simple calculation shows that if A^{-1} exists then

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} \mathbf{1}_n & 0 \\ CA^{-1} & \mathbf{1}_m \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & S_A \end{pmatrix} \begin{pmatrix} \mathbf{1}_n & A^{-1}B \\ 0 & \mathbf{1}_m \end{pmatrix},$$

and if D^{-1} exists then

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} \mathbf{1}_n & BD^{-1} \\ 0 & \mathbf{1}_m \end{pmatrix} \begin{pmatrix} S_D & 0 \\ 0 & D \end{pmatrix} \begin{pmatrix} \mathbf{1}_n & 0 \\ D^{-1}C & \mathbf{1}_m \end{pmatrix}.$$

In particular, if S is Hermitian, we have the following equivalent characterization for $S \in \text{Pos}(\mathbb{C}^{n+m})$ and $S \in \text{Pd}(\mathbb{C}^{n+m})$.

Lemma 16. Let S be defined as in eq. (14) assume it is a Hermitian matrix. Then we have the following.

1. Suppose $A \in \text{Pd}(\mathbb{C}^n)$. Then $S \succeq 0$ if and only if $S_A \succeq 0$. Moreover, $S \succ 0$ if and only if $S_A \succ 0$.
2. Suppose $D \in \text{Pd}(\mathbb{C}^m)$. Then $S \succeq 0$ if and only if $S_D \succeq 0$. Moreover, $S \succ 0$ if and only if $S_D \succ 0$.

Matrix norms. We begin with defining matrix norms induced by vector norms. For a matrix $A \in \mathbb{C}^{n \times m}$, the norm $\|A\|_{p \rightarrow q}$ is defined as

$$\|A\|_{p \rightarrow q} = \max_{x \in \mathbb{C}^m} \left\{ \frac{\|Ax\|_q}{\|x\|_p} \right\}.$$

Of particular interests are the norms $\|A\|_{1 \rightarrow 2}$ and $\|A\|_{2 \rightarrow \infty}$, which are the maximum of the 2-norm of the columns of A and the maximum of the 2-norm of the rows of A , respectively.

In this paper, we work with $\|A\|_{1 \rightarrow 2}$. For the sake of completion, we show that the assertion we made above is true (the proof that $\|A\|_{2 \rightarrow \infty}$ is the maximum of the 2-norm of the rows of A follows similarly).

Fact 17. For a matrix $A \in \mathbb{C}^{n \times m}$, the norm $\|A\|_{1 \rightarrow 2}$ is the maximum 2-norm of the columns of A .

Proof. To see why this is true, let us fix $x \in \mathbb{C}^m$ such that $\|x\|_1 = 1$. We have that

$$\begin{aligned} \|Ax\|_2^2 &= \sum_{i \in [n]} \sum_{j \in [m]} |A[i, j]|^2 |x[j]|^2 \leq \sum_{i \in [n]} \sum_{j \in [m]} |A[i, j]|^2 |x[j]| \\ &= \sum_{j \in [m]} \left(\sum_{i \in [n]} |A[i, j]|^2 \right) |x[j]| \leq \max_{j \in [m]} \left(\sum_{i \in [n]} |A[i, j]|^2 \right) \end{aligned}$$

where both the inequalities follows because $\|x\|_1 = 1$. If k is the column of A with maximum 2-norm, it is clear that the maximum is achieved by setting $x = e_k \in \mathbb{R}^m$. This complete the proof of Fact 17. \square

We will employ the following Schatten norms in this paper:

$$\|A\|_\infty = \sigma_1(A) \quad \text{and} \quad \|A\|_1 = \sum_{i=1}^p \sigma_i(A) \quad \text{and} \quad \|A\|_F = \left(\sum_{i=1}^p \sigma_i(A)^2 \right)^{1/2} = \left(\sum_{i \in [n]} \sum_{j \in [m]} |A[i, j]|^2 \right)^{1/2} \quad (16)$$

where $p = \min\{m, n\}$. Finally, we state the factorization norm that is used to derive our bounds. Given a matrix $A \in \mathbb{C}^{n \times m}$, we define $\gamma_F(A)$ as

$$\gamma_F(A) = \inf \{ \|B\|_F \|C\|_{1 \rightarrow 2} : A = BC \}.$$

The quantity $\gamma_F(\cdot)$ is a norm and can be achieved by a factorization that involves finite-dimensional matrices. Moreover, if A is a real matrix, then we can restrict our attention to the real factorization of A . We refer interested readers to Appendix B for more detail. Another factorization norm that we mention in this work is cb-norm (also known as $\gamma_2(A)$ norm) which is defined as

$$\|A\|_{\text{cb}} = \gamma_2(A) := \inf \{ \|B\|_{2 \rightarrow \infty} \|C\|_{1 \rightarrow 2} : A = BC \}. \quad (17)$$

By construction the two aforementioned factorization norms satisfy the following relationship

$$\gamma_2(A) \leq \gamma_F(A) \leq \sqrt{n} \gamma_2(A) \quad (18)$$

where $A \in \mathbb{C}^{n \times m}$. Moreover, both inequalities are tight – the left inequality is an equality when A has only one entry and the right inequality is an equality for all unitary matrices.

Matrix decompositions. At various points in this paper, we will refer to one of the following types of matrix decompositions.

1. *Singular value decomposition:* Any complex matrix $A \in \mathbb{C}^{n \times m}$ can be decomposed as $A = U \Sigma_A V^*$, where $\Sigma_A \in \text{Pd}(\mathbb{C}^p)$ is a diagonal matrix of strictly positive singular values of A (and hence, $p \leq \min\{n, m\}$), and $U \in \mathbb{C}^{n \times p}$ and $V \in \mathbb{C}^{m \times p}$ satisfying $U^*U = V^*V = \mathbb{1}_p$. Moreover, $UU^* \preceq \mathbb{1}_n$ and $VV^* \preceq \mathbb{1}_m$. For a real matrix $A \in \mathbb{C}^{n \times m}$, one can assume that U and V are matrices with real entries.
2. *Spectral decomposition:* Any normal matrix $A \in \mathbb{N}(\mathbb{C}^n)$ can be decomposed as $A = U \Lambda_A U^*$, where Λ_A is a diagonal matrix of eigenvalues of A and U is a unitary matrix. Moreover, any positive semidefinite matrix $A \in \text{Pos}(\mathbb{C}^n)$ can be decomposed as $A = BB^*$ for $B \in \mathbb{C}^{n \times n}$. If A is a real positive semidefinite matrix, then one can assume that B is a real matrix.

We state the following well known linear algebra facts about Hermitian and normal matrices.

Fact 18. Let $A \in \text{Herm}(\mathbb{C}^n)$ with eigenvalues $(\lambda_1(A), \dots, \lambda_n(A))$ and $B \in \text{Pos}(\mathbb{C}^n)$. Then

$$\text{Tr}(A) = \sum_{i=1}^n \lambda_i(A) \quad \text{and} \quad \text{Tr}(AB) \geq \lambda_n(A) \text{Tr}(B).$$

We need the following result regarding the singular values of M_{count} :

Theorem 19. Let M_{count} be the matrix defined in eq. (2). Let $\sigma_1, \dots, \sigma_n$ be its n -singular values. Then for all $1 \leq i \leq n$,

$$\sigma_i = \frac{1}{2} \left| \csc \left(\frac{(2i-1)\pi}{4n+2} \right) \right|.$$

Proof. The proof argument is due to Gilbert Strang. We present a proof for the sake of completion. We can compute M_{count}^{-1} exactly as follows:

$$M_{\text{count}}^{-1}[i, j] = \begin{cases} -1 & i = j + 1 \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}.$$

From this, we can compute $(M_{\text{count}}^* M_{\text{count}})^{-1}$ as follows:

$$(M_{\text{count}}^* M_{\text{count}})^{-1} = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix}.$$

This is exactly the graph Laplacian matrix B with Neumann boundary conditions considered in [SM14, Section 9]. The result follows. \square

Remark 20. Throughout the paper, we implicitly assume that $M_{\text{count}} \in \mathbb{R}^{n \times n}$ and make use of L and R to denote the factorization of M_{count} , i.e., $M_{\text{count}} = LR$.

For the upper bound, we use the following result on the double factorial.

Theorem 21 (Chen and Qi [CQ05]). For any $m \in \mathbb{N}$, let $(m)!!$ denote the double factorial defined as follows:

$$(2m)!! = \prod_{i=1}^m (2i) \quad \text{and} \quad (2m-1)!! = \prod_{i=1}^m (2i-1).$$

Then

$$\sqrt{\frac{1}{\pi(k + \frac{4}{\pi} - 1)}} \leq \frac{(2k-1)!!}{(2k)!!} < \sqrt{\frac{1}{\pi(k + \frac{1}{4})}}.$$

Furthermore, the constants $\frac{4}{\pi} - 1$ and $\frac{1}{4}$ are tight.

Vector calculus. We give a brief overview of the vector calculus required to understand this paper. The basic object of concern is a *vector field* in a space. In this paper, we will always be concerned with the d -dimensional vector space defined over reals, \mathbb{R}^d . A vector field is an assignment of a vector to each point in a space. Unlike scalar calculus, in vector calculus, we can have various differential operators, which are typically expressed in terms of the *del* operator, ∇ .

Given a scalar field, f , i.e., a scalar function of position $\theta \in \mathbb{R}^d$, its gradient at any point $\theta \in \mathbb{R}^d$, denoted by $\nabla_{\theta} f(\theta)$, is defined as the vector field

$$\begin{pmatrix} \frac{\partial}{\partial \theta_1} f(\theta) \\ \frac{\partial}{\partial \theta_2} f(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_d} f(\theta) \end{pmatrix}.$$

A *Hessian matrix* is a square matrix of second-order partial derivatives of a scalar-valued function. At a point $\theta \in \mathbb{R}^d$, the Hessian of a scalar field, f , is

$$\begin{pmatrix} \frac{\partial^2 f(\theta)}{\partial \theta_1^2} & \frac{\partial^2 f(\theta)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 f(\theta)}{\partial \theta_1 \partial \theta_d} \\ \frac{\partial^2 f(\theta)}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 f(\theta)}{\partial \theta_2^2} & \cdots & \frac{\partial^2 f(\theta)}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\theta)}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 f(\theta)}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 f(\theta)}{\partial \theta_n^2} \end{pmatrix}$$

We use the symbol $\nabla^2 f(\theta)$ to denote the Hessian of the scalar field f . If the second partial derivatives are all continuous, then the Hessian matrix is a symmetric matrix. This fact is known as *Schwarz's theorem*.

We defined these concepts for $\theta \in \mathbb{R}^d$. They generalize naturally when $\theta \in \mathcal{K}$ for some closed compact set $\mathcal{K} \subseteq \mathbb{R}^d$ [BV04].

2.2 Convex Optimization

In this section, we give a brief overview of convex optimization to the level required to understand Section 7. Let \mathcal{K} denote a convex, closed and compact set over which the optimization problem is defined.

Definition 22 (Extended-value convex function). *An extended-value convex function $\phi : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ satisfies*

$$\phi(\alpha x + (1 - \alpha)y) \leq \alpha \phi(x) + (1 - \alpha)\phi(y)$$

for all $\alpha \in (0, 1)$ and the domain of the ϕ is $\text{dom}(\phi) := \{x : \phi(x) < \infty\}$.

Definition 23 (Proper function and Convex function). *A function ϕ is proper if there exists an $x \in \mathcal{K}$ such that $\phi(x) < +\infty$ and, for all $x \in \mathcal{K}$, $\phi(x) > -\infty$. A convex function is an extended-value convex function that is also proper.*

Definition 24 (Subgradient). *The subgradient (or subdifferential) set of a convex function $\phi : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ at a point x is defined as follows:*

$$\partial \phi(x) := \{g : \forall y \in \mathcal{K}, \quad \phi(y) - \phi(x) \geq \langle g, (y - x) \rangle\}.$$

If the function is continuously differentiable, then one of the subgradients is the gradient of the function and denoted by $\nabla\phi$.

Note that the subdifferential is a non-empty set if x is in the strict interior of the domain of ϕ and is defined even if the function is not continuously differentiable.

Definition 25 (Strongly convex). Let $\alpha > 0$. A convex function $\phi : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ is an α -strongly convex function with respect to the ℓ_2 -norm if for all $x, y \in \mathcal{K}$,

$$\forall g \in \partial\phi(x), \quad \phi(y) - \phi(x) \geq \langle g, y - x \rangle + \frac{\alpha}{2} \|y - x\|_2^2.$$

The following is shown in McMahan [McM17]:

Lemma 26 (Lemma 7 in McMahan [McM17]). Let $\phi_1 : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ be a convex function such that

$$x = \arg \min_{x \in \mathcal{K}} \phi_1(x).$$

Let ψ be a convex function such that $\phi_2(x) = \phi_1(x) + \psi(x)$ is λ -strongly convex with respect to the norm $\|\cdot\|_2$. Let

$$y = \arg \min_{x \in \mathcal{K}} \phi_2(x).$$

Then for any $b \in \partial\psi(x)$, we have

$$\|x - y\|_2 \leq \frac{1}{\lambda} \|b\|_2, \quad \text{and} \quad \forall \bar{x} \in \mathcal{K}, \quad \phi_2(x) - \phi_2(\bar{x}) \leq \frac{1}{2\lambda} \|b\|_2^2.$$

Fact 27. If the function $\phi : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ is twice differentiable, i.e., admits a second derivative, then α -strong convexity is equivalent to $\alpha \mathbb{1}_d \preceq \nabla^2\phi(x)$, where $\nabla^2\phi(x)$ denotes the Hessian⁵ of the function ϕ at $x \in \mathcal{K}$.

In this paper, we will extensively use duality theory. Central to it is the Fenchel conjugate, which generalizes Lagrangian duality.

Definition 28 (Fenchel conjugate). The Fenchel conjugate of an arbitrary function $\psi : \mathcal{K} \rightarrow \mathbb{R} \cup \{\infty\}$ is defined as follows:

$$\psi^*(g) := \sup_x \langle g, x \rangle - \psi(x).$$

One of the main motivations of our work is to get an exact bound on regret minimization for convex optimization in the online setting using the private online convex optimization algorithm of Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21]. In online optimization, an online player makes decisions iteratively. After committing to the decision, the player suffers a loss. This loss is made known to the player only after the decision is made. The goal of the player is to ensure that the total loss, i.e, the average of the losses of all decisions, known as *regret*, is minimized compared to the loss of the post-hoc best decision, which is the decision which generates the smallest total loss if used for *all* online choices. We will consider the following setting of online convex optimization.

Let \mathcal{D} denote the domain of data samples and let $\ell : \mathcal{K} \times \mathcal{D} \rightarrow \mathbb{R}$ be a convex function in the first parameter. Then the main goal of an online algorithm is to minimize the *regret* against an arbitrary post-hoc optimizer $\theta^{\text{opt}} \in \mathcal{K}$:

Definition 29 (Regret minimization). Let A_{OCO} be an online convex programming algorithm, which at every step $t \in [n]$, observes data samples $[x_1, \dots, x_{t-1}]$ and outputs $\theta_t \in \mathcal{K}$. The performance of A_{OCO} is measured in terms of regret over n iterations:

$$\text{Regret}(A_{\text{OCO}}; n) = \frac{1}{n} \sum_{i=1}^n \ell(\theta_i; x_i) - \frac{1}{n} \min_{\theta \in \mathcal{K}} \sum_{i=1}^n \ell(\theta; x_i).$$

Note that we are using the definition of regret which is normalized instead of the one used in Hazan [Haz19]. One can consider both *adversarial regret* [Haz19], where the data sample x_t are drawn adversarially based on the past outputs $\{\theta_1, \dots, \theta_{t-1}\}$, and *stochastic regret* [Haz19], where the data is sampled i.i.d. from some fixed unknown distribution \mathcal{D} .

⁵A Hessian is a square matrix of second-order partial derivatives of a scalar valued function.

Algorithm 2 Follow-the-regularized leader [Haz19, Algorithm 10]

Require: A convex closed compact set \mathcal{K} , regularization function $\rho : \mathcal{K} \rightarrow \mathbb{R}$, dimension d of the parameter space, learning rate $\eta > 0$.

- 1: Set $\theta_1 = \arg \min_{\theta \in \mathcal{K}} \rho(\theta)$.
- 2: **for** $t = 1$ to n **do**
- 3: Predict θ_t .
- 4: Observe the new data sample x_t and compute $\nabla_t = \nabla \ell(\theta_t; x_t)$.
- 5: Update

$$\theta_{t+1} = \arg \min_{\theta} \left\{ \eta \sum_{i=1}^t \langle \nabla_i, \theta \rangle + \rho(\theta) \right\}$$

- 6: return $\sum_{i=1}^{t+1} \theta_i$.
 - 7: **end for**
-

Follow-the-regularized leader. One of the most important and successful families of low-regret algorithms for online convex optimization is the *follow-the-regularized leader* (FTRL). The generic FTRL meta-algorithm is defined in Algorithm 2. Different FTRL algorithms use a different *regularization function* (aka *regularizer*, leading to different update rules. Two most common regularizer are (1) the *entropy function*, which results in the multiplicative weight update method and used in the private multiplicative weight update method, and (2) the ℓ_2^2 -regularizer, which is the choice of regularizer used in this paper and whose privacy guarantee is well studied [TS13].

We use the following bound on the regularized follow-the-regularized leader (Algorithm 2):

Theorem 30 (Theorem 5.2 in [Haz19]). *Let $R : \mathcal{K} \rightarrow \mathbb{R}$ be the regularization function and let $\eta > 0$ be the learning rate. The regularized follow-the-perturbed leader (defined in Algorithm 2) attains for every $u \in \mathcal{K}$, the following bound on the regret:*

$$\text{Regret}(\text{A}_{\text{oco}}; n) \leq \frac{2\eta}{n} \sum_{i=1}^n \left(\|\nabla_i\|_2^2 \right) + \frac{\rho(u) - \rho(\theta_1)}{n\eta}. \quad (19)$$

The first term (summation of the so called *local norms*, $\|\nabla_i\|_2^2$, of the gradients) in eq. (19) is called the *width term* and the second term is known as the *diameter term*. If we have a universal bound on the local norms, i.e., for all $1 \leq i \leq n$, $\|\nabla_i\|_2^2 \leq L$ for some constant L , then we can optimize over the learning rate η to get the final regret.

2.3 Differential Privacy

The privacy definition we use in this paper is *differential privacy*. We define it next based on the notion of *neighborhood* which we define below for different applications.

Definition 31 (Differential privacy). *Let $\mathcal{M} : X \rightarrow R$ be a randomized algorithm mapping from a domain X to a range R . \mathcal{M} is (ϵ, δ) -differentially private if for every all neighboring dataset D and D' and every measurable set $C \subseteq R$,*

$$\Pr[\mathcal{M}(D) \in C] \leq e^\epsilon \Pr[\mathcal{M}(D') \in C] + \delta.$$

Central to the notion of privacy is the notion of neighboring dataset. In this paper, we use the standard notion of neighboring dataset for each use case.

1. *Continual observation:* Two streams, $S = (x_1, \dots, x_n) \in \{0, 1\}^n$ and $S' = (x'_1, \dots, x'_n) \in \{0, 1\}^n$ are neighboring if there is at most one $1 \leq i \leq n$ such that $x_i \neq x'_i$. This is known as *event level privacy* [CSS11, DNPR10].

2. *Online convex optimization*: Two dataset $D = \{x_1, \dots, x_n\}$ and $D' = \{x'_1, \dots, x'_n\}$ are considered neighboring if they differ in one data-point [TS13]. That is, there is at most one $1 \leq i \leq n$ such that $x_i \neq x'_i$.
3. *Parity Queries*: Two dataset $D = \{x_1, \dots, x_n\} \in \{-1, +1\}^n$ and $D' = \{x'_1, \dots, x'_n\} \in \{-1, +1\}^n$ are neighboring if there is at most one $1 \leq i \leq n$ such that $x_i \oplus x'_i = -1$.

In both use cases our privacy and utility guarantee depends on the Gaussian distribution. Given a random variable X , we denote by $X \sim N(\mu, \sigma^2)$ the fact that X has Gaussian distribution with mean μ and variance σ^2 with the probability density function

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The multivariate Gaussian distribution is the multi-dimensional generalization of the Gaussian distribution. For a random variable X , we denote by $X \sim N(\mu, \Sigma)$ the fact that X has a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ which is defined as $\Sigma = \mathbf{E}[(X - \mu)(X - \mu)^*]$. The probability density function of a multivariate Gaussian has a closed form formula:

$$p_X(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} e^{-(x-\mu)^* \Sigma^{-1} (x-\mu)},$$

where $\det(\Sigma)$ denotes the determinant of Σ . The covariance matrix is a positive definite matrix. We use the following fact regarding the multivariate Gaussian distribution:

Fact 32. *Let $X \sim N(\mu, \Sigma)$ be a d -dimensional multivariate Gaussian distribution. If $A \in \mathbb{C}^{n \times d}$, then the multivariate random variable $Y = AX$ is distributed as though $Y \sim N(A\mu, A\Sigma A^*)$.*

Our algorithm for continual counting uses the Gaussian mechanism. To define it, we need to first define the notion of ℓ_2 -sensitivity. For a function $f : \mathcal{X}^n \rightarrow \mathbb{R}^d$ its ℓ_2 -sensitivity is defined as

$$\Delta f := \max_{\text{neighboring } X, X' \in \mathcal{X}^n} \|f(X) - f(X')\|_2. \quad (20)$$

Definition 33 (Gaussian mechanism). *Let $f : \mathcal{X}^n \rightarrow \mathbb{R}^d$ be a function with ℓ_2 -sensitivity Δf . For a given $\epsilon, \delta \in (0, 1)$ given $X \in \mathcal{X}^n$ the Gaussian mechanism \mathcal{M} returns $\mathcal{M}(X) = f(X) + e$, where $e \sim N(0, C_{\epsilon, \delta}^2 (\Delta f)^2 \mathbb{1}_d)$.*

Theorem 34. *For a given $\epsilon, \delta \in (0, 1)$ the Gaussian mechanism \mathcal{M} satisfies (ϵ, δ) -differential privacy.*

We will use the following result:

Theorem 35 (Theorem 2.1 in Denisov, McMahan, Rush, Smith, and Thakurta [DMR⁺22]). *Let $A \in \mathbb{R}^{n \times n}$ be a lower-triangular full-rank query matrix, and let $A = BC$ be any factorization with the following property: for any two neighboring streams of vectors $x, x' \in \mathbb{R}^n$, we have $\|C(x - x')\| \leq \zeta$. Let $z \sim N(0, \zeta^2 C_{\epsilon, \delta}^2)^n$ with ζ large enough so that $\mathcal{M}(x) = Ax + Bz = B(Cx + z)$ satisfies (ϵ, δ) -DP in the nonadaptive continual release model. Then, \mathcal{M} satisfies the same DP guarantee (with the same parameters) even when the rows of the input sequence are chosen adaptively.*

We use the result by Edmonds, Nikolov, and Ullman [ENU20]. In particular, Edmonds, Nikolov, and Ullman [ENU20, Section 5] showed that the accuracy for linear queries (when expressed as a query matrix A) can be characterized using $\gamma_F(A)$. An *instance independent mechanism* can be written as

$$\mathcal{M}(x) = (Ax + z)$$

for a workload matrix A . It is called *instance independent*⁶ as the noise function used does not depend on the input instance x . As they use a somewhat different notation from ours we reprove their result to show that their result can be restated in our notation as follows:

⁶In [BDKT12] this was called *oblivious*.

Theorem 36. • Given a linear function $f(x) = Ax$ with matrix $A \in \mathbb{R}^{n \times d}$ and a factorization $A = LR$, then for $\varepsilon, \delta \in (0, 1)$

$$\mathcal{M}_{L,R}(A, x) = Ax + z, \quad \text{where } z \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2 LL^*\right)$$

is (ε, δ) -differential private under the neighboring relation considered in this paper.

- For a workload matrix A consisting of n queries let \mathcal{M} be a (ε, δ) -differentially private instance-independent mechanism for A with $\varepsilon > 0$ and $0 \leq \delta \leq 1$. Then

$$\text{err}_{\ell_2^2}(\mathcal{M}, A, n) \geq C_{\varepsilon}^2 \frac{\gamma_F(A)^2}{n},$$

where $C_{\varepsilon} = \frac{1}{e^{2\varepsilon} - 1}$ is the constant in [KRSU10].

- For a workload matrix A consisting of n queries let \mathcal{M} be a (ε, δ) -differentially private mechanism for A with $\varepsilon > 0$ and $0 \leq \delta \leq \frac{1}{4e^{\varepsilon}}$. Then

$$\text{err}_{\ell_2^2}(\mathcal{M}, A, n) \geq C_{\varepsilon}^2 \frac{\gamma_F(A)^2}{n},$$

where $C_{\varepsilon} = \frac{1}{e^{2\varepsilon} - 1}$ is the constant in [KRSU10].

Proof. Let $y \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2\right)$. Then by the properties of the multivariate Gaussian distribution (Fact 32), it holds that $Ly \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2 LL^*\right)$. Thus, the mechanism \mathcal{M} returning on input x the value $L(Rx + y)$ with $y \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2\right)$ has the same distribution as the mechanism returning $Ax + z$ with $z \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2 LL^*\right)$. In particular, both have the same privacy properties. Thus, to simplify the notation we call the latter mechanism \mathcal{M} as well.

Next let us show (ε, δ) -differential privacy. Consider the mechanism \mathcal{M}' that on input x returns $f(x) = Rx + y$ with $y \sim N\left(0, C_{\varepsilon, \delta}^2 \|R\|_{1 \rightarrow 2}^2\right)$. For any neighboring databases x and x' represented in the form of an m -dimensional vector and differing in the i -th coordinate note that $\|R(x - x')\|_2 = \|Re_i\|_2 \leq \|R\|_{1 \rightarrow 2}$ and, thus, $\Delta_2 f = \max_{\text{neighboring } x, x'} \|R(x - x')\|_2 \leq \|R\|_{1 \rightarrow 2}$. Thus, (ε, δ) -differential privacy of \mathcal{M}' follows from Theorem 34. Finally note that \mathcal{M} only postprocesses the output of \mathcal{M}' by multiplying with the matrix L , and, thus, the postprocessing property of (ε, δ) -differential privacy imply that \mathcal{M} has the same privacy properties as \mathcal{M}' . Hence \mathcal{M} is (ε, δ) -differential private.

For the lower bound on the mean squared error, we first state the relationship between the notation of Edmonds, Nikolov, and Ullman [ENU20] and our notation and then show the result in the theorem. For the ease of presentation, we only show the translation for their lower bounds for instance independent mechanism – the reduction from instance dependent (or regular) mechanisms to instance independent follows from [BDKT12], which for every $\varepsilon' > 0$ and $0 \leq \delta' \leq 1$ turns any (ε', δ') -differentially private instance dependent mechanism into a $(2\varepsilon', 2e^{\varepsilon'} \delta')$ -differentially private instance independent mechanism, without increasing its mean-squared error. Thus, a lower bound on the mean-squared error for (ε', δ') -differentially private instance independent mechanisms with $\varepsilon' > 0$ and $0 \leq \delta' \leq 1$ turns into a lower bound (of the same value) for (ε, δ) -differentially private instance dependent mechanisms with $\varepsilon > 0$ and $0 \leq \delta \leq \frac{1}{2e^{\varepsilon}}$.

Our first point of departure is the way the factorization norm is defined. For a query matrix $A \in \mathbb{R}^{n \times m}$, Edmonds, Nikolov, and Ullman [ENU20] defined the following norm, which we denote by $\bar{\gamma}_F(A)$ to indicate that their definition is a normalized version of our definition in eq. (3). :

$$\bar{\gamma}_F(A) = \min \left\{ \frac{1}{\sqrt{n}} \|L\|_F \|R\|_{1 \rightarrow 2} : A = LR \right\} = \frac{1}{\sqrt{n}} \gamma_F(A).$$

The second point of departure is the way the workload matrix is defined. Edmonds, Nikolov, and Ullman [ENU20] define and use in their mechanisms the normalized form of a workload matrix, i.e., for a

workload matrix $A \in \mathbb{R}^{n \times m}$, they consider $\bar{A} = \frac{1}{n}A$. With this notation, Edmonds, Nikolov, and Ullman [ENU20, Theorem 28] studied instance independent mechanisms. An *instance independent mechanism* can be written as

$$\bar{\mathcal{M}}(x) = \frac{1}{n}(Ax + z)$$

for a workload matrix A . It is called instance independent as the noise function used does not depend on the input instance x .

Finally, their definition for the mean-squared error is the square-root of the standard definition of mean-squared error (and the definition used in this paper) – their choice of defining mean-squared error is so that they can compare it easily with the ℓ_∞ error. To differentiate the two, we use the notation $\overline{\text{err}}_{\ell_2^2}$ to denote their error metric. More formally, they define

$$\overline{\text{err}}_{\ell_2^2}(\bar{\mathcal{M}}, A, n) = \max_{x \in \{0,1\}^n} \mathbf{E} \left[\frac{1}{n} \|\bar{\mathcal{M}}(x) - \bar{A}x\|_2^2 \right]^{1/2}.$$

They showed that for such an instance independent mechanism and a suitable constant $C > 0$

$$\overline{\text{err}}_{\ell_2^2}(\bar{\mathcal{M}}, A, n) \geq \frac{\bar{\gamma}_F(A)}{C\varepsilon n}.$$

Let x be the input that maximizes the $\overline{\text{err}}_{\ell_2^2}(\bar{\mathcal{M}}, A, n)$. Now,

$$\mathbf{E} \left[\frac{1}{n} \left\| \bar{\mathcal{M}}(x) - \frac{Ax}{n} \right\|_2^2 \right]^{1/2} = \mathbf{E} \left[\frac{1}{n} \left\| \frac{\mathcal{M}(x)}{n} - \frac{Ax}{n} \right\|_2^2 \right]^{1/2} = \frac{1}{n} \mathbf{E} \left[\frac{1}{n} \|\mathcal{M}(x) - Ax\|_2^2 \right]^{1/2}.$$

In other words,

$$\mathbf{E} \left[\frac{1}{n} \|\mathcal{M}(x) - Ax\|_2^2 \right]^{1/2} \geq \frac{\bar{\gamma}_F(A)}{C\varepsilon}$$

Finally, since $\sqrt{n}\bar{\gamma}_F(A) = \gamma_F(A)$, we have

$$\mathbf{E} \left[\frac{1}{n} \|\mathcal{M}(x) - Ax\|_2^2 \right]^{1/2} \geq \frac{\gamma_F(A)}{C\varepsilon\sqrt{n}}$$

or equivalently,

$$\mathbf{E} \left[\frac{1}{n} \|\mathcal{M}(x) - Ax\|_2^2 \right] \geq \frac{\gamma_F(A)^2}{C^2\varepsilon^2 n}.$$

Unraveling the proof of Edmonds, Nikolov, and Ullman [ENU20] and the proof in [KRSU10], on which it is based, we see that $C\varepsilon = 1/C_\varepsilon$ as required. This completes the proof of Theorem 36. \square

3 Semidefinite Program for $\gamma_F(\cdot)$ Norm

In this section, we characterize $\gamma_F(A)$ for any $A \in \mathbb{C}^{n \times m}$ as a semidefinite program. To begin we can safely restrict our attention to a factorization $A = BC$ such that $\|B\|_F = \|C\|_{1 \rightarrow 2}$. In particular, this assumption can be made for an optimal factorization as well. To see why this holds, let us consider a factorization $A = BC$ such that

$$\frac{\|B\|_F}{\|C\|_{1 \rightarrow 2}} = \alpha(B, C) \quad \text{for} \quad \alpha(B, C) \neq 1.$$

Then we can have another factorization $A = B'C'$ where

$$B' = \frac{B}{\sqrt{\alpha(B, C)}} \quad \text{and} \quad C' = \sqrt{\alpha(B, C)}C$$

satisfying $\|B'\|_F = \|C'\|_{1 \rightarrow 2}$. Hence,

$$\gamma_F(A) = \inf \{ \eta : \|B\|_F = \|C\|_{1 \rightarrow 2} = \sqrt{\eta} \text{ and } A = BC \}.$$

For the remainder of this section, let $A \in \mathbf{C}^{n \times m}$ and $X \in \text{Herm}(\mathbf{C}^{n+m})$ be a matrix written in the following block form

$$X = \begin{pmatrix} X_1 & X_2 \\ X_2^* & X_3 \end{pmatrix} \quad \text{such that} \quad X_2 = A. \quad (21)$$

For any factorization $A = BC$, the matrix X as stated in eq. (21) satisfies

$$X \in \text{Pos}(\mathbf{C}^{n+m}) \quad \text{if and only if} \quad X = WW^* \quad \text{for} \quad W = \begin{pmatrix} B \\ C^* \end{pmatrix}.$$

This implies that $X_1 = BB^*$, $X_2 = A = BC$, and $X_3 = C^*C$. Moreover, it is clear that

$$\text{Tr}(X_1) = \text{Tr}(BB^*) = \|B\|_F^2 \quad \text{and} \quad X_3[i, i] = \|C[:, i]\|_2^2.$$

Let $\Phi : \text{Herm}(\mathbf{C}^{n+m}) \rightarrow \text{Herm}(\mathbf{C}^{n+m})$ be the linear map defined as

$$\Phi(X) = \hat{J}_{n,m} \bullet X \quad \text{where} \quad \hat{J}_{n,m} = \begin{pmatrix} 0 & J_{n,m} \\ J_{n,m}^* & 0 \end{pmatrix} \quad \text{and let} \quad \hat{A} = \begin{pmatrix} 0 & A \\ A^* & 0 \end{pmatrix}. \quad (22)$$

Then the SDP for $\gamma_F(A)$ can be written as follows:

$$\begin{aligned} \gamma_F(A) &:= \min \quad \eta \\ &\text{s.t.} \quad \Phi(X) = \hat{J}_{n,m} \bullet X = \hat{A} \\ &\quad \sum_{i \in [n]} X[i, i] \leq \eta \\ &\quad X[i, i] \leq \eta \quad \forall i \in [n+1, n+m] \\ &\quad X \in \text{Pos}(\mathbf{C}^{n+m}). \end{aligned}$$

We remark that strong duality holds for the above SDP and its associated dual (refer to Appendix A for the proof). For any optimal solution pair (η, X) , it is necessarily true that

$$\sum_{i \in [n]} X[i, i] = \eta. \quad (23)$$

Otherwise, we can construct a solution pair with optimal value strictly less than η . To see why this is true, let $\alpha = \sqrt{\eta/\eta'} > 1$ where

$$\sum_{i \in [n]} X[i, i] = \eta' < \eta.$$

For

$$X = \begin{pmatrix} B \\ C^* \end{pmatrix} \begin{pmatrix} B^* & C \end{pmatrix} \quad \text{let} \quad X' = \begin{pmatrix} \sqrt{\alpha}B \\ \sqrt{\frac{1}{\alpha}}C^* \end{pmatrix} \begin{pmatrix} \sqrt{\alpha}B^* & \sqrt{\frac{1}{\alpha}}C \end{pmatrix}.$$

It is evident that $X' \in \text{Pos}(\mathbf{C}^{n+m})$ is a feasible solution of the aforementioned SDP since $\Phi(X') = \Phi(X) = \hat{A}$. Moreover,

$$\sum_{i \in [n]} X'[i, i] = \alpha \text{Tr}(BB^*) = \alpha\eta' = \sqrt{\eta'\eta},$$

and

$$X'[i, i] = \frac{X[i, i]}{\alpha} \leq \frac{\eta}{\alpha} = \sqrt{\eta'\eta} \quad \text{for all } i \in [n+1, n+m].$$

Hence the pair $(\sqrt{\eta'\eta}, X')$ forms a feasible solution, and since $\sqrt{\eta'\eta} < \eta$, it contradicts our assumption that (η, X) is an optimal solution. This implies that eq. (23) holds necessarily for any optimal solution. We now proceed to write a reformulation of the dual of the above SDP in the form that we make use of in all our lower bounds:

$$\begin{aligned} \gamma_F(A) = \max \quad & w^*(\widehat{A} \bullet \widehat{X})w \\ \text{s.t.} \quad & \begin{pmatrix} n\mathbb{1}_n & 0 \\ 0 & \mathbb{1}_m \end{pmatrix} \succeq \widehat{Z} \\ & w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad \text{such that } \|w\|_2 = 1 \text{ and } w_1 = \alpha \mathbb{1}_n \\ & \widehat{Z} \in \text{Herm}(\mathbb{C}^{n+m}), \alpha \in \mathbb{R}_{++}, \text{ and } w \in \mathbb{R}_{++}^{n+m}. \end{aligned} \quad (24)$$

We refer interested readers to Appendix A for an explanation of how we arrive at such a formulation. Note that the above form is reminiscent of a reformulation of $\|\cdot\|_{\text{cb}}$ norm due to Haagerup [Haa80], but has a strictly smaller feasible set.

4 Proof of Lemma 1

The lower bound in our main result require the following two propositions.

Proposition 37. *Let $U \in \mathbb{C}^{n \times p}$ and $V \in \mathbb{C}^{m \times p}$ such that $\|U\|_\infty \leq 1$ and $\|V\|_\infty \leq 1$, where $\|\cdot\|_\infty$ denotes the spectral norm. Then*

$$\begin{pmatrix} |q_1|^2 \mathbb{1}_n & q_1 q_2 UV^* \\ q_1^* q_2^* VU^* & |q_2|^2 \mathbb{1}_m \end{pmatrix} \succeq 0 \quad \text{for all } q_1, q_2 \in \mathbb{C}.$$

Proof. A simple calculation shows that

$$\mathbb{1}_n^*(A \bullet B)\mathbb{1}_m = \text{Tr}((A \bullet B)\mathbb{1}_m \mathbb{1}_n^*) = \text{Tr}((A \bullet B)J_{m,n}) = \sum_{i=1}^n \sum_{j=1}^m A[i, j]B[i, j] = \text{Tr}(A^*B)$$

completing the proof of Proposition 38. □

Our lower bound also used a simple fact about block positive semidefinite matrices where the diagonal blocks are scalar multiple of identity matrices (Proposition 37). We prove it next.

Proposition 38. *Let $A, B \in \mathbb{C}^{n \times m}$. Then $\mathbb{1}_n^*(A \bullet B)\mathbb{1}_m = \text{Tr}(A^*B)$, where $\text{Tr}(\cdot)$ denotes the trace of the matrix.*

Proof of Proposition 37. Given that $\|U\|_\infty \leq 1$ and $\|V\|_\infty \leq 1$, we have $UU^* \preceq \mathbb{1}_n$ and $VV^* \preceq \mathbb{1}_m$. It follows that

$$\begin{pmatrix} |q_1|^2 \mathbb{1}_n & q_1 q_2 UV^* \\ q_1^* q_2^* VU^* & |q_2|^2 \mathbb{1}_m \end{pmatrix} \succeq \begin{pmatrix} |q_1|^2 UU^* & q_1 q_2 UV^* \\ q_1^* q_2^* VU^* & |q_2|^2 VV^* \end{pmatrix} = \begin{pmatrix} q_1 U \\ q_2^* V \end{pmatrix} \begin{pmatrix} q_1 U \\ q_2^* V \end{pmatrix}^* \succeq 0.$$

This completes the proof of the proposition. □

We now return to the proof of Lemma 1.

Proof of Lemma 1. We construct dual variables w and \widehat{Z} for the SDP in Figure 2 that achieve the objective value as stated in the lemma. Let $A = U\Sigma_A V^*$ be the singular value decomposition of A . Let

$$w = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbb{1}_n / \sqrt{n} \\ \mathbb{1}_m / \sqrt{m} \end{pmatrix} \quad \text{and} \quad \widehat{Z} = \begin{pmatrix} 0^{n \times n} & Z \\ Z^* & 0^{m \times m} \end{pmatrix}, \quad \text{where } Z = \sqrt{n}UV^* \in \mathbb{C}^{n \times m}. \quad (25)$$

Since $\|U\|_\infty = \|V\|_\infty = 1$, by setting $q_1 = \sqrt{n}$ and $q_2 = -1$ in Proposition 37, we get that \widehat{Z} is in the dual feasible set of the SDP defined in Figure 2. It is not hard to see that all the constraints imposed by the dual SDP on the vector w is satisfied with $\alpha = 1/\sqrt{2n}$. It remains to show the value of the objective function achieved by this dual solution. Using Proposition 38, the value of the objective function is

$$w^* \left(\widehat{A} \bullet \widehat{Z} \right) w = \frac{1}{2\sqrt{nm}} (1_m^*(A^* \bullet Z^*)1_n + 1_n^*(A \bullet Z)1_m) = \frac{1}{2\sqrt{nm}} \text{Tr}(AZ^* + A^*Z) = \frac{1}{\sqrt{m}} \text{Tr}(\Sigma_A),$$

where the last equality follows due to the following argument: since $AZ^* = \sqrt{n}U\Sigma_A U^*$ and $A^*Z = \sqrt{n}V\Sigma_A V^*$, we have $\text{Tr}(AZ^* + A^*Z) = 2\sqrt{n} \text{Tr}(\Sigma_A)$ using the cyclic property of a trace. Hence

$$\gamma_F(A) \geq w^* \left(\widehat{A} \bullet \widehat{Z} \right) w = \frac{1}{\sqrt{m}} \text{Tr}(\Sigma_A) = \frac{\|A\|_1}{\sqrt{m}}$$

by the definition of the Schatten-1 norm.

We now turn to proving the upper bounds. The upper bound can be obtained via constructing a factorization of a matrix. Fix a matrix $A \in \mathbb{C}^{n \times m}$. For a factorization $A = LR$, let $L = A$ and $R = 1_m$. We have $\|L\|_F = \|A\|_F$ and $\|R\|_{1 \rightarrow 2} = 1$, and hence

$$\gamma_F(A) \leq \|A\|_F. \quad (26)$$

This completes the proof of Lemma 1. □

5 Proof of the Bounds on Differentially Private Continual Counting

5.1 Proof of Upper Bound on Differentially Private Continual Counting

As Algorithm 1 shows, we have $L = R$. This is definitely a more restrictive setting and any upper bound under this restriction is also an upper bound on $\gamma_F(M_{\text{count}})$. We will show that even under this restriction, we get an almost tight factorization and leave the question of finding an even tighter factorization when this restriction is removed as a direction of future research.

The requirement $L = R$ results in $n(n+1)/2$ equations in $n(n+1)/2$ variables. Our first observation is that the entries on any $t \times t$ principal submatrix of L and R are independent of the rest of the entries of L and R ; however, they define the rest of the entries. The second observation we make is that L and R are a Toeplitz matrix with a special structure: the principal diagonal entries all have to be the same and equal to 1, and the k -th lower diagonal would be $\left(1 - \frac{1}{2(k-1)}\right)$ times the entries in $(k-1)$ -th lower diagonal. In other words, we get the recurrence relation

$$f(k) = \begin{cases} 1 & k = 0 \\ \left(1 - \frac{1}{2k}\right) f(k-1) & k \geq 1 \end{cases} \quad (27)$$

that defines the entries of the factors as

$$L = R = \begin{pmatrix} f(0) & 0 & \cdots & 0 \\ f(1) & f(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f(n-2) & f(n-3) & \cdots & 0 \\ f(n-1) & f(n-2) & \cdots & f(0) \end{pmatrix}.$$

Note that this factorization is the same as in Algorithm 1. By construction, $M_{\text{count}} = LR$, so all that remains is to prove the bound on $\|L\|_F \|R\|_{1 \rightarrow 2}$ and the accuracy guarantee follows from eq. (4). The factorization into two Toeplitz matrices also means that we have bounded operators on the Hilbert space. First,

Theorem 21 gives us for all $1 \leq t \leq n$,

$$\begin{aligned}
\|L[t;]\|_{1 \rightarrow 2}^2 &= \left(1 + \sum_{i=1}^{t-1} \prod_{j=1}^i \left(1 - \frac{1}{2j}\right)^2\right) = \left(1 + \sum_{i=1}^{t-1} \left(\prod_{j=1}^i \left(\frac{2j-1}{2j}\right)\right)^2\right) \\
&= \left(1 + \sum_{i=1}^{t-1} \left(\frac{(2i-1)!!}{(2i)!!}\right)^2\right) \leq \left(1 + \frac{4}{\pi} \sum_{i=1}^{t-1} \frac{1}{(4k+1)}\right) \\
&\leq \left(1 + \frac{4}{\pi} \left[\frac{\ln(|4x+1|)}{4}\right]_{x=1}^{t-1}\right) \leq \left(1 + \frac{1}{\pi} \ln\left(\frac{4t-3}{5}\right)\right)
\end{aligned} \tag{28}$$

Using eq. (28), we therefore have

$$\begin{aligned}
\|L\|_{\mathbb{F}}^2 &= \sum_{t=1}^n \|L[t;]\|_2^2 \leq \sum_{t=1}^n \left(1 + \frac{1}{\pi} \ln\left(\frac{4t-3}{5}\right)\right) \\
&\leq \left(n + \frac{n \ln(4n/5)}{\pi}\right) = n \left(1 + \frac{\ln(4n/5)}{\pi}\right)
\end{aligned} \tag{29}$$

since natural-log is a monotonically increasing strictly concave function.

As $L = R$, using eq. (28), it follow that

$$\|R\|_{1 \rightarrow 2}^2 = \|R[n;]\|_2^2 \leq \left(1 + \frac{1}{\pi} \ln\left(\frac{4n}{5}\right)\right).$$

Combining the two bounds, we have the upper bound in eq. (11). Equation (6) now follows using eq. (4).

The privacy proof follows from the fact that our mechanism is an instantiation of the matrix mechanism. In particular, using Fact 32, we can write

$$M_{\text{count}}x + z = L(Rx + y),$$

where $y \sim N(0, \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2 \mathbb{1}_n)$. Note that we can consider the multiplication with L as a post-processing step as L does not depend on x . Thus it suffices to argue that $f(x) = Rx$ is released in a differentially private manner. As stated in Definition 33, the standard Gaussian mechanism for this problem releases $f(x) + y'$, where $y' \sim N(0, \Delta_2(f)^2 C_{\epsilon, \delta}^2 \mathbb{1}_n)$, where $\Delta_2(f)$ is the ℓ_2 -sensitivity of f . As for two neighboring vectors x and x' that differ only in bit i it holds that

$$\|R(x - x')\|_2 = \|Re_i\|_2 \leq \|R\|_{1 \rightarrow 2},$$

it follows that the ℓ_2 -sensitivity of f is $\|R\|_{1 \rightarrow 2}$, which shows that y was sampled from the appropriate normal distribution to preserve (ϵ, δ) -differential privacy of the complete execution over all round. The result for adaptivity follows from Theorem 35.

We can also improve the update-time of our algorithm as follows:

Corollary 39. *There is an efficient data-structure \mathcal{D} and a continual counting mechanism \mathcal{M} that, for all $1 \leq t \leq n$, on receiving a bit $x_t \in \{0, 1\}$, outputs a_t that satisfy (ϵ, δ) -differential privacy and*

$$\text{err}_{\ell_2}(\mathcal{M}, M_{\text{count}}, n) \leq C_{\epsilon, \delta}^2 \left(1 + \frac{\ln(n)}{\pi}\right)^2$$

Further, the data structure \mathcal{D} uses $O(n)$ space and uses $O(1)$ time per round, and pre-processing time of $O(n^2 + ns)$, where s is the time required to sample from a normal distribution.

Proof. We present the non-adaptive continual counting algorithm; the adaptive continual counting result follows from Theorem 35. Let $M_{\text{count}} = LR$ be a factorization defined in the proof of Theorem 2. The data structure D and the continual counting algorithm \mathcal{M} are defined as follows: During preprocessing we sample a vector $z \sim N(0, \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2 LL^*)$, where $LR = M_{\text{count}}$ is the factorization computed in Section 5.1. We describe below how to do this in $O(n^2 + ns)$ time. For a stream $x = (x_1, \dots, x_n) \in \{0, 1\}^n$, our data structure D at time t uses $O(n)$ space as it stores the following information:

1. The current count $S_t = \sum_{i=1}^t x_i$.
2. The vector $z \in \mathbb{R}^n$ sampled during preprocessing.

Our efficient continual counting mechanism \mathcal{M} consists of the following steps: For each round t , after receiving x_t , it simply outputs $c_t = S_t + z[t]$.

To prove the privacy guarantee note that the complete output (c_1, \dots, c_n) is equal to the vector $M_{\text{count}}x + z$, since the input is chosen non-adaptively. Using Fact 32, we can write

$$M_{\text{count}}x + z = L(Rx + y),$$

where $y \sim N(0, \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2 \mathbb{1}_n)$. Note that we can consider the multiplication with L as a post-processing step as L does not depend on x . Thus it suffices to argue that $f(x) = Rx$ is released in a differentially private manner. As stated in Definition 33, the standard Gaussian mechanism for this problem releases $f(x) + y'$, where $y' \sim N(0, \Delta_2(f)^2 C_{\epsilon, \delta}^2 \mathbb{1}_n)$, where $\Delta_2(f)$ is the ℓ_2 -sensitivity of f . As it holds for two neighboring vectors x and x' that differ only in bit i that

$$\|R(x - x')\|_2 = \|Re_i\|_2 = \|R\|_{1 \rightarrow 2},$$

it follows that the ℓ_2 -sensitivity of f is $\|R\|_{1 \rightarrow 2}$, which shows that y was sampled from the appropriate normal distribution.

The analysis of the time per round is straightforward. For the pre-processing time, note that, in general, sampling from a multivariate Gaussian $N(\mu, \Sigma)$ requires inverting the covariance matrix $\Sigma \succ 0$, which would require $O(n^3)$ time. However, in our case, we can sample a vector from the distribution $N(0, \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2 LL^*)$ in time $O(n^2 + ns)$ by the following procedure:

1. Sample n Gaussian samples (g_1, \dots, g_n) , i.i.d. from $N(0, 1)$. This takes $O(ns)$ time, where $O(s)$ is the time required to sample from a normal distribution.
2. Form a vector $g = (g_1 \quad g_2 \quad \dots \quad g_n)^*$. This takes n time.
3. Output the vector $z = Lg$. This takes $O(n^2)$ time.

By Fact 32 it follows that the vector z has the same distribution as $N(0, LL^* \|R\|_{1 \rightarrow 2}^2 C_{\epsilon, \delta}^2)$. This completes the proof of Corollary 39. \square

5.2 Proof of Lower Bounds on Continual Counting

Proof of Theorem 3. Let $\{\sigma_1(M_{\text{count}}), \sigma_2(M_{\text{count}}), \dots, \sigma_n(M_{\text{count}})\}$ be the singular values of M_{count} . Note that M_{count} is a non-singular matrix. We use the following well known fact that follows from noting that $(M_{\text{count}}^* M_{\text{count}})^{-1}$ is the matrix considered in [SM14, Section 9] (also see Theorem 19):

$$\sigma_i(M_{\text{count}}) = \frac{1}{2} \left| \text{csc} \left(\frac{(2i-1)\pi}{4n+2} \right) \right| \quad \text{for all } i \in [n].$$

Since $y^{-1} \leq |\text{csc}(y)|$ for all $y > 0$ and Schatten-1 norm is just the sum of singular values, we have

$$\begin{aligned} \|M_{\text{count}}\|_1 &= \sum_{i=1}^n \sigma_i(M_{\text{count}}) = \frac{1}{2} \sum_{i=1}^n \left| \text{csc} \left(\frac{(2i-1)\pi}{4n+2} \right) \right| \geq \frac{2n+1}{\pi} \sum_{i=1}^n \frac{1}{2i-1} = \frac{2n+1}{\pi} \left(1 + \sum_{i=2}^n \frac{1}{2i-1} \right) \\ &> \frac{2n+1}{\pi} \left(1 + \int_3^{n+1} \frac{dx}{2x-1} \right) = \frac{2n+1}{\pi} \left(1 + \frac{1}{2} (\ln(2n+1) - \ln(5)) \right). \end{aligned}$$

Setting $A = M_{\text{count}}$ and $m = n$ in Lemma 1, we therefore have

$$\gamma_{\text{F}}(M_{\text{count}}) > \frac{\|M_{\text{count}}\|_1}{\sqrt{n}} > \frac{2n+1}{\pi\sqrt{n}} \left(1 + \frac{1}{2} \ln\left(\frac{2n+1}{5}\right)\right) \geq \frac{\sqrt{n}}{\pi} \left(2 + \ln\left(\frac{2n+1}{3}\right) + \frac{\ln(2n+1)}{2n}\right). \quad (30)$$

Theorem 3 follows by using eq. (5) and eq. (30). \square

Proof of Theorem 4. We first prove the lower bound for instance independent mechanism. Using Theorem 36, we have that

$$\text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{\gamma_{\text{F}}(M_{\text{count}})^2}{n(e^{2\varepsilon} - 1)^2}.$$

for all instance-independent (ε, δ) -differentially private mechanism \mathcal{M} . Now using eq. (11), we have

$$\text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{1}{(e^{2\varepsilon} - 1)^2 \pi^2} \left(2 + \ln\left(\frac{2n+1}{3}\right) + \frac{\ln(2n+1)}{2n}\right)^2$$

for all instance-independent (ε, δ) -differentially private mechanism \mathcal{M} . This completes the second part of Theorem 4.

For instance-dependent mechanism, using the same proof as in Edmonds, Nikolov, and Ullman [ENU20] and eq. (11), we get

$$\text{err}_{\ell_2^2}(\mathcal{M}, M_{\text{count}}, n) \geq \frac{1}{(e^{4\varepsilon} - 1)^2 \pi^2} \left(2 + \ln\left(\frac{2n+1}{3}\right) + \frac{\ln(2n+1)}{2n}\right)$$

This completes the proof of Theorem 4. \square

6 Factorization View of Known Mechanisms and Suboptimality of the Binary Mechanism

In this section, we will first show how all known mechanisms for continual counting can be seen as a matrix mechanism. Apart from the binary mechanism [DNPR10, CSS11] there exist two variants by Honaker [Hon15]: one that is the optimized version and one that is suited for the continual observation.

Binary (Tree) Mechanism. Assume in the following that the stream length $n = 2^m$ for some $m \in \mathbb{N}$ and consider a complete binary tree with n leaves and $2n - 1$ nodes in total and let $\pi(i)$ denote the path from leaf i to the root. For $i \geq 1$, leaf i is labeled with x_i , the i -th input in the streamed vector x . Each node with 2^k leaves in its subtree consists of the dyadic interval $[j2^k, (j+1)2^k - 1]$ of the input stream x for some integer $j \geq 0$ and represents the k -th bit in the binary representation of the leaves in its subtree. The binary mechanism computes (a) the p -sum for each node in the binary tree consisting of the sum of the values of the leaves in its subtree with suitable noise, resulting in *noisy p -sum values*, and (b) then computes the i -th output by adding up the noisy p -sum values of the nodes on $\pi(i)$ that represent bits that are set to 1 in the binary representation of i . For example for $i = 1$ only the p -sum of leaf 1 is returned, for $i = 2$ only the p -sum of the parent of leaves 1 and 2, for $i = 3$ the sum of the p -sum of leaf 3 and of parent of leaves 1 and 2 is returned.

The following observation is straightforward: the binary mechanism computes a linear combination of the entries in the streamed vector x as the p -sum value of each internal node of the binary tree is a linear combination of the entries of streamed vector x . Now we can consider the binary mechanism as a matrix

mechanism. The right factor R_{binary} is constructed as follows: $R_{\text{binary}} = W_m$, where W_0, \dots, W_m are defined recursively as follows:

$$W_0 = (1), \quad W_k = \begin{pmatrix} W_{k-1} & 0^{2^{k-1} \times 2^{k-1}} \\ 0^{2^{k-1} \times 2^{k-1}} & W_{k-1} \\ (1_{2^{k-1}})^T & (1_{2^{k-1}})^T \end{pmatrix}, \quad k \leq m.$$

Note that $R_{\text{binary}} = W_m$ is a matrix in $\{0, 1\}^{(2^{n-1}) \times n}$, with each row corresponding to the p-sum computed for a node in the binary tree, where the ordering of the rows corresponds to a labeling of the nodes by post-order in the binary tree. Thus a 1 in column j of row i indicates that x_j contributes to the p-sum of the i -th node in post-order. For example, the top-most row corresponds to the p-sum of the left-most leaf and the bottom-most row corresponds to the p-sum of the root of the binary tree.

The corresponding matrix L_{binary} is a matrix of $\{0, 1\}^{n \times (2^{n-1})}$, where row i corresponds exactly to the binary representation of i . Thus, it has a one in at most $\lceil \log_2(i) \rceil$ entries.

Honaker's optimization Honaker's optimization [Hon15] uses the same matrix as in the case of the binary mechanism for the right matrix, i.e., $R_{\text{honaker}} = R_{\text{binary}}$. For the left matrix, he solves the optimization problem that minimizes the variance introduced. Even though it is not explicitly stated in [Hon15], one can write the closed formula for his left matrix as $L_{\text{honaker}} = M_{\text{count}} R_{\text{binary}}^\dagger$, where $R_{\text{binary}}^\dagger$ denotes the Moore-Penrose pseudoinverse of R_{binary} .

Honaker's streaming version Honaker's streaming algorithm has the same right matrix as the binary mechanism and his optimized mechanism. However, to ensure computation in the streaming model, Honaker's left matrix has to be constrained – the left matrix is a constraint Moore-Penrose pseudoinverse. In particular, it does not have a closed-form expression, but can be computed in polynomial time using known algorithms from numerical analysis.

Suboptimality of the Binary Mechanism. We next give the detail proof of the suboptimality of the binary mechanism.

Proof of Theorem 5. Next we use our factorization view of the binary mechanism to show its suboptimality as far as the constants are concerned. It is easy to see that $\|R_{\text{binary}}\|_{1 \rightarrow 2}^2 = 1 + \log_2(n)$. Recall that we want to bound $\|L_{\text{binary}}\|_F$, which is the square root of the number of entries of L that are 1. Thus, a simple counting argument then yields

$$\|L_{\text{binary}}\|_F^2 = \frac{n \log_2(n)}{2}.$$

In other words, we have argued the following:

Lemma 40. *The binary mechanism (defined in Dwork, Naor, Pittasi, and Rothblum [DNPR10] and Chan, Shi, and Song [CSS11]) can be represented as a factorization of M_{count} into sparse matrices L_{binary} and R_{binary} such that*

$$M_{\text{count}} = L_{\text{binary}} R_{\text{binary}} \quad \text{and} \quad \|L_{\text{binary}}\|_F^2 \|R_{\text{binary}}\|_{1 \rightarrow 2}^2 = \frac{n \log_2(n)}{2} (1 + \log_2(n)).$$

Since, for any factorization using the Gaussian mechanism, we have that the expected mean squared error is exactly $C_{\epsilon, \delta} \|L\|_F \|R\|_{1 \rightarrow 2}$, we get that the binary tree with the Gaussian mechanism achieves sub-optimal accuracy in comparison to our mechanism. More precisely, the mean squared error of the binary mechanism is approximately a factor of $\frac{\pi^2}{2(\ln 2)^2} \approx 10.2$ larger than that of our mechanism.

The second part of Theorem 5 follows from our lower bound on $\gamma_F(M_{\text{count}})$ in eq. (11). \square

Algorithm 3 Differentially private Follow-the-regularized leader, A_{oco}

Require: Dataset $D = (x_1, \dots, x_n)$, dimension d of the parameter space, constraint set \mathcal{K} , regularization parameter λ , clipping norm κ , ℓ_2^2 regularizer $\rho(\theta) = \frac{1}{\lambda} \|\theta\|_2^2$.

- 1: Set $\theta_1 = \arg \min_{\theta} \frac{\lambda}{2} \|\theta\|_2^2$ represented as a column vector.
- 2: **for** $t = 1$ to n **do**
- 3: $\nabla_t \leftarrow \text{clip}(\nabla \ell(\theta_t; x_t); \kappa)$, where $\text{clip}(\cdot; \cdot)$ is as defined in eq. (31).
- 4: Define $G(t) = (\nabla_1 \quad \nabla_2 \quad \dots \quad \nabla_t)^*$ whose i -th row is formed by the row vector ∇_i^* .
- 5: Sample $Z \in \mathbb{R}^{t \times d}$, where $Z[i, j] \sim N(0, C_{\epsilon, \delta}^2 \kappa^2 \|R(t)\|_{1 \rightarrow 2}^2)$.
- 6: Compute $s_t = L[t; \cdot](R(t)G(t) + Z)$, where $R(t)$ is the $t \times t$ principal submatrix of R .
- 7: Update

$$\theta_{t+1} = \arg \min_{\theta} \left(\langle s_t, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2 \right)$$

- 8: **Output** θ_{t+1} .
 - 9: **end for**
-

7 Non-asymptotic Bound on Private Online Optimization

In this section we prove Theorem 6. For this we modify the algorithm for private online convex optimization, given in Algorithm 3 below by replacing the binary mechanism in Line 6 by our mechanism. Our bounds assumes that the loss function $\ell : \mathcal{K} \times \mathcal{D} \rightarrow \mathbb{R}$ function is convex and κ -Lipschitz with respect to the ℓ_2 norm, that is, for all $d \in \mathcal{D}, \theta \in \mathcal{K}$,

$$\|\nabla \ell(\theta; d)\|_2 \leq \kappa.$$

We guarantee the Lipschitz property using the standard clipping method used in private learning. Let $\mathcal{B}_d(0, \kappa)$ denotes the d -dimensional Euclidean ball of radius κ centered at origin. Then we define the clipping function, $\text{clip} : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathcal{B}_d(0, \kappa)$ as follows:

$$\text{clip}(g; \kappa) := \min \left\{ \frac{\kappa}{\|\nabla \ell(\theta_t; x_t)\|_2}, 1 \right\} \cdot g, \quad (31)$$

Note that $\text{clip}(\nabla \ell(\theta_t; x_t); \kappa) \in \mathcal{B}_d(0, \kappa)$.

We first state our result on the regret bound (Definition 29) on online private optimization in the *adversarial regret model* [Haz19], where the data sample x_t are drawn adversarially based on the past outputs $\{\theta_1, \dots, \theta_{t-1}\}$. The result can be extended to *stochastic regret* [Haz19], where the data is sampled i.i.d. from some fixed unknown distribution \mathcal{D} using standard techniques [Haz19].

Theorem 41 (Restatement of Theorem 6). *Let $[\theta_1, \dots, \theta_n]$ be the d -dimensional outputs of Algorithm A_{oco} , and L be a bound on the ℓ_2 -Lipschitz constant of the loss functions. Then the following is true for any $\theta_{\text{opt}} \in \mathcal{K}$:*

$$\text{Regret}(A_{\text{oco}}; n) \leq \|\theta_{\text{opt}}\|_2 \sqrt{\frac{\left(1 + \frac{\ln(4n/5)}{\pi}\right) (\kappa^2 + \kappa C_{\epsilon, \delta} \sqrt{d})}{2n}}.$$

Proof. The privacy proof follows from the privacy of the Gaussian mechanism and noting that the ℓ_2 -sensitivity of $R(t)G(t)$ is at most $\kappa \|R(t)\|_{1 \rightarrow 2}^2$ (as every row of $G(t)$ has ℓ_2 -norm norm at most κ by clipping). The utility proof follows from the idea in Kairouz, McMahan, Song, Thakkar, Thakurta, and Xu [KMS⁺21] using $\rho(\theta) = \frac{\lambda}{2} \|\theta\|_2^2$ (for a suitably chosen $\lambda > 0$) except that we replace their binary mechanism with our mechanism. We show that the proof also goes through with our mechanism.

Let ∇_i denote the gradient of the current cost function at the current point $\nabla \ell(\theta_i; x_i)$ and

$$\theta_{\text{opt}} := \arg \min_{\theta \in \mathcal{K}} \sum_{i=1}^n \ell(\theta; x_i).$$

Define the following optimizers for the non-private and private variant of follow-the-regularized leader, respectively:

$$\begin{aligned}
\tilde{\theta}_{t+1} &= \arg \min_{\theta \in \mathcal{C}} \sum_{i=1}^t \langle \nabla_i, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2 \\
\theta_{t+1} &= \arg \min_{\theta} \left(\langle s_t, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2 \right) \\
&= \arg \min_{\theta \in \mathcal{C}} \sum_{i=1}^t \langle \nabla_i, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2 + \left\langle s_t - \sum_{i=1}^t \nabla_i, \theta \right\rangle,
\end{aligned} \tag{32}$$

for s_t be the estimate returned from the partial sum using our mechanism. Therefore, we have

$$\begin{aligned}
\sum_{t=1}^n \ell(\theta_t; x_t) - \sum_{t=1}^n \ell(\theta_{\text{opt}}; x_t) &\stackrel{(*)}{\leq} \sum_{t=1}^n \langle \nabla_t, \theta_t - \theta_{\text{opt}} \rangle \\
&= \underbrace{\sum_{t=1}^n \langle \nabla_t, \theta_t - \tilde{\theta}_t \rangle}_P + \underbrace{\sum_{t=1}^n \langle \nabla_t, \tilde{\theta}_t - \theta_{\text{opt}} \rangle}_N,
\end{aligned} \tag{33}$$

where $(*)$ follows from [Haz19, eq. 5].

We can now bound the term N and P separately. Note that N is the regret if we did not had any privacy constraints (i.e., $\varepsilon = \infty$). Therefore, we bound that term using the non-private regret bound of follow-the-regularized leader with regularization function $\rho(\theta) = \frac{\|\theta\|_2^2}{2\lambda}$. In other words, we set $\rho(\theta) = \frac{\|\theta\|_2^2}{2\lambda}$, $u = \theta_{\text{opt}}$, and $\eta = 1$ in Theorem 30. Then using the fact that ℓ is κ -Lipschitz with respect to ℓ_2 norm, we have

$$N \leq \frac{n\kappa^2}{\lambda} + \frac{\lambda}{2} \left(\|\theta_{\text{opt}}\|_2^2 - \|\theta_1\|_2^2 \right). \tag{34}$$

For the term P , define

$$\phi_1(\theta) := \frac{1}{\lambda} \sum_{i=1}^t \left(\langle \nabla_i, \theta \rangle + \frac{\lambda}{2} \|\theta\|_2^2 \right), \quad \phi_2(\theta) := \frac{1}{\lambda} \left(\sum_{i=1}^t \langle \nabla_i, \theta \rangle \right) + \frac{\lambda}{2} \|\theta\|_2^2 + \left\langle s_t - \sum_{i=1}^t \nabla_i, \theta \right\rangle.$$

Note that $\tilde{\theta}_t = \arg \min_{\theta} \phi_1(\theta)$ and that $\theta_t = \arg \min_{\theta} \phi_2(\theta)$, and

$$\psi(\theta) = \frac{1}{\lambda} \left\langle s_t - \sum_{i=1}^t \nabla_i, \theta \right\rangle$$

is a linear function with

$$\nabla \psi(\theta) = \frac{1}{\lambda} s_t - \sum_{j=1}^t \nabla_j$$

being the subgradient of ψ at $\theta \in \mathcal{K}$. Furthermore, $\phi_1(\theta)$ and $\phi_2(\theta)$ are quadratic functions and for $\theta \in \mathcal{K}$

$$\nabla \phi_1(\theta) = \lambda \theta + \frac{1}{\lambda} \sum_{i=1}^t \nabla_i \quad \text{and} \quad \nabla^2 \phi_1(\theta) = \lambda \mathbb{1}_d.$$

and

$$\nabla \phi_2(\theta) = \lambda \theta + \frac{1}{\lambda} \sum_{i=1}^t \nabla_i + s_t - \sum_{i=1}^t \nabla_i \quad \text{and} \quad \nabla^2 \phi_2(\theta) = \lambda \mathbb{1}_d.$$

It follows from Fact 27 that ϕ_2 is λ -strongly convex and that ψ and ϕ_2 are convex. We can thus apply Lemma 26 to get

$$\left\| \tilde{\theta}_t - \theta_t \right\|_2 \leq \frac{1}{\lambda} \|L[t;]Z_t\|_2,$$

where $L[t;]$ is the t -dimensional t -th row of L and $Z_t \in \mathbb{R}^{t \times d}$ is a random Gaussian matrix such that $Z_t[k, j] \sim N(0, C_{\varepsilon, \delta}^2 \|R(t)\|_{1 \rightarrow 2}^2)$. The result of this product is a d -dimensional vector.

Since $\ell(\cdot; \cdot)$ is κ -Lipschitz with respect to ℓ_2 norm in its first parameter, we have

$$P \leq \sum_{t=1}^n \|\nabla_t\|_2 \left\| \tilde{\theta}_t - \theta_t \right\|_2 \leq \kappa \sum_{t=1}^n \left\| \tilde{\theta}_t - \theta_t \right\|_2, \quad (35)$$

where the first inequality follows from Cauchy-Schwarz inequality.

Taking expectation in eq. (33) and substituting the bounds on P and N in eq. (35) and eq. (34), respectively, we get

$$\frac{1}{n} \mathbf{E} \left[\sum_{t=1}^n \ell(\theta_t; x_t) - \sum_{t=1}^n \ell(\theta_{\text{opt}}; x_t) \right] \leq \frac{\kappa^2}{\lambda} + \frac{\lambda}{2n} \left(\|\theta_{\text{opt}}\|_2^2 - \|\theta_1\|_2^2 \right) + \frac{\kappa}{n} \sum_{t=1}^n \mathbf{E} \left[\left\| \tilde{\theta}_t - \theta_t \right\|_2 \right] \quad (36)$$

Since every entry of $Z_t[k, j] \sim N(0, C_{\varepsilon, \delta}^2 \|R(t)\|_{1 \rightarrow 2}^2)$ for $1 \leq k \leq t, 1 \leq j \leq d$, we have by the independence of the $Z_t[k, j]$ variables that

$$\begin{aligned} \mathbf{E} \left[\|L[t;]Z_t\|_2^2 \right] &= \mathbf{E} \left[\sum_{j=1}^d (L[t;]Z_t[j])^2 \right] = \sum_{j=1}^d \mathbf{Var} (L[t;]Z_t[j]) = \sum_{j=1}^d \sum_{k=1}^t \mathbf{Var} (L[t, k]Z_t[k, j]) \\ &\leq d \sum_{k=1}^t L[t, k]^2 C_{\varepsilon, \delta}^2 \|R(t)\|_{1 \rightarrow 2}^2 = d C_{\varepsilon, \delta}^2 \|R(t)\|_{1 \rightarrow 2}^2 \|L[t;]\|_2^2 \end{aligned}$$

Thus, it follow that

$$\mathbf{E} \left[\|L[t;]Z_t\|_2 \right] \leq \sqrt{\mathbf{E} \left[\|L[t;]Z_t\|_2^2 \right]} \leq C_{\varepsilon, \delta} \sqrt{d} \|R(t)\|_{1 \rightarrow 2} \|L[t;]\|_2.$$

Combining all of this, we get

$$\begin{aligned} \frac{1}{n} \mathbf{E} \left[\sum_{t=1}^n \ell(\theta_t; x_t) - \sum_{t=1}^n \ell(\theta_{\text{opt}}; x_t) \right] &\leq \frac{\kappa^2}{\lambda} + \frac{\lambda}{2n} \left(\|\theta_{\text{opt}}\|_2^2 - \|\theta_1\|_2^2 \right) + \frac{\kappa}{n} \sum_{t=1}^n \mathbf{E} \left[\left\| \tilde{\theta}_t - \theta_t \right\|_2 \right] \\ &\leq \frac{\kappa^2}{\lambda} + \frac{\lambda}{2n} \left(\|\theta_{\text{opt}}\|_2^2 - \|\theta_1\|_2^2 \right) + \frac{\kappa}{\lambda n} \sum_{t=1}^n \mathbf{E} \left[\|L[t;]Z_t\|_2 \right] \\ &\leq \frac{\kappa^2}{\lambda} + \frac{\lambda}{2n} \|\theta_{\text{opt}}\|_2^2 + \frac{\kappa \sqrt{d}}{n \lambda} C_{\varepsilon, \delta} \sum_{t=1}^n \left(1 + \frac{\ln(4t/5)}{\pi} \right) \\ &\leq \frac{\kappa^2}{\lambda} + \frac{\lambda}{2n} \|\theta_{\text{opt}}\|_2^2 + \frac{\kappa \sqrt{d}}{\lambda} C_{\varepsilon, \delta} \left(1 + \frac{\ln(4n/5)}{\pi} \right). \end{aligned} \quad (37)$$

We now optimize for λ to get

$$\lambda = \frac{\sqrt{2n \left(1 + \frac{\ln(4n/5)}{\pi} \right) (\kappa^2 + \kappa C_{\varepsilon, \delta} \sqrt{d})}}{\|\theta_{\text{opt}}\|_2},$$

and consequently, the bound on regret is

$$\text{Regret}(A_{\text{oco}}; n) \leq \|\theta_{\text{opt}}\|_2 \sqrt{\frac{\left(1 + \frac{\ln(4n/5)}{\pi}\right) (\kappa^2 + \kappa C_{\varepsilon, \delta} \sqrt{d})}{2n}}$$

completing the proof of Theorem 6. □

8 Lower Bound on Parity Queries

We follow the same approach as for M_{count} . We use the observation of Edmonds, Nikolov, and Ullman [ENU20] that the query matrix corresponding to any set of the parity queries is the $\binom{d}{w}$ matrix formed by taking the corresponding rows of the $2^d \times 2^d$ unnormalized Hadamard matrix. Let us call this matrix S .

We now set $A = S, n = \binom{d}{w}$ and $m = 2^d$ in Lemma 1. Using the fact that the singular value of S is $2^{d/2}$ with multiplicity $\binom{d}{w}$, we get $\|S\|_1 = 2^{d/2} \binom{d}{w}$ and a lower bound on $\gamma_F(S) \geq \binom{d}{w}$. Theorem 9 now follows by an application of Theorem 36.

Acknowledgements. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 101019564 “The Design of Modern Fully Dynamic Data Structures (Mo-DynStruct)” and from the Austrian Science Fund (FWF) project “Fast Algorithms for a Reactive Network Layer (ReactNet)”, P 33775-N, with additional funding from the *ne-tidee SCIENCE Stiftung*, 2020–2024. JU’s research was funded by Decanal Research Grant. JU would like to thank Aleksandar Nikolov for useful discussions on factorization norms. A part of this work was done when JU was visiting Indian Statistical Institute, Delhi.



References

- [AFT22] Hilal Asi, Vitaly Feldman, and Kunal Talwar. Optimal algorithms for mean estimation under local differential privacy. *arXiv preprint arXiv:2205.02466*, 2022. 1
- [BDKT12] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In *Proceedings of the forty-fourth annual ACM Symposium on Theory of computing*, pages 1269–1284, 2012. 18, 19
- [BEM⁺17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proc. of the 26th ACM Symp. on Operating Systems Principles (SOSP’17)*, 2017. 1
- [BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014. 1
- [BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 15
- [Cen] Differential privacy for census data explained. <https://www.ncsl.org/research/redistricting/differential-privacy-for-census-data-explained.aspx>. accessed: 2022-07-05. 1
- [CLSX12] T-H Hubert Chan, Mingfei Li, Elaine Shi, and Wenchang Xu. Differentially private continual monitoring of heavy hitters from distributed streams. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 140–159. Springer, 2012. 1
- [CQ05] Chao-Ping Chen and Feng Qi. The best bounds in wallis’ inequality. *Proceedings of the American Mathematical Society*, 133(2):397–401, 2005. 14
- [CR21] Adrian Cardoso and Ryan Rogers. Differentially private histograms under continual observation: Streaming selection into the unknown. *arXiv preprint arXiv:2103.16787*, 2021. 1
- [CSS11] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011. 1, 4, 5, 17, 26, 27
- [DMR⁺22] Sergey Denisov, Brendan McMahan, Keith Rush, Adam Smith, and Abhradeep Thakurta. Improved differential privacy for sgd via optimal private linear operators on adaptive streams. *arXiv preprint arXiv:2202.08312*, 2022. 1, 2, 3, 4, 6, 18
- [DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proc. of the Forty-Second ACM Symp. on Theory of Computing (STOC’10)*, pages 715–724, 2010. 1, 4, 5, 17, 26, 27
- [DTTZ14] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM Symposium on Theory of computing*, pages 11–20. ACM, 2014. 1
- [ENU20] Alexander Edmonds, Aleksandar Nikolov, and Jonathan Ullman. The power of factorization mechanisms in local and central differential privacy. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 425–438, 2020. 2, 3, 7, 9, 10, 18, 19, 20, 26, 31
- [FHO21] Hendrik Fichtenberger, Monika Henzinger, and Wolfgang Ost. Differentially private algorithms for graphs under continual observation. In *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, 2021. 1

- [FHU22] Hendrik Fichtenberger, Monika Henzinger, and Jalaj Upadhyay. Constant matters: Fine-grained complexity of differentially private continual observation using completely bounded norms. *arXiv preprint arXiv:2202.11205*, 2022. [1](#), [8](#)
- [Haa80] Uffe Haagerup. Decomposition of completely bounded maps on operator algebras, 1980. [22](#)
- [Haz19] Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019. [16](#), [17](#), [28](#), [29](#)
- [HLL⁺22] Yuxuan Han, Zhicong Liang, Zhipeng Liang, Yang Wang, Yuan Yao, and Jiheng Zhang. Private streaming sco in ℓ_p geometry with applications in high dimensional online decision making. In *International Conference on Machine Learning*, pages 8249–8279. PMLR, 2022. [1](#)
- [Hon15] James Honaker. Efficient use of differentially private binary trees. *Theory and Practice of Differential Privacy (TPDP 2015), London, UK*, 2015. [2](#), [26](#), [27](#)
- [HQYC21] Ziyue Huang, Yuan Qiu, Ke Yi, and Graham Cormode. Frequency estimation under multiparty differential privacy: One-shot and streaming. *arXiv preprint arXiv:2104.01808*, 2021. [1](#)
- [JRSS21] Palak Jain, Sofya Raskhodnikova, Satchit Sivakumar, and Adam Smith. The price of differential privacy under continual observation. *arXiv preprint arXiv:2112.00828*, 2021. [4](#)
- [KMS⁺21] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pages 5213–5225. PMLR, 2021. [1](#), [4](#), [5](#), [6](#), [16](#), [28](#)
- [KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the forty-second ACM Symposium on Theory of computing*, pages 775–784, 2010. [9](#), [19](#), [20](#)
- [LMH⁺15] Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *The VLDB journal*, 24(6):757–781, 2015. [2](#), [3](#), [4](#)
- [Mat93] Roy Mathias. The hadamard operator norm of a circulant and applications. *SIAM journal on matrix analysis and applications*, 14(4):1152–1167, 1993. [8](#)
- [McM17] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017. [16](#)
- [MK04] Jorma K Merikoski and Ravinder Kumar. Inequalities for spreads of matrix sums and products. *Applied Mathematics E-Notes*, 4:150–159, 2004. [9](#)
- [MT22] Brendan McMahan and Abhradeep Thakurta. Federated learning with formal differential privacy guarantees. 2022. [1](#), [6](#)
- [Nik22] Alexander Nikolov. Personal communication. 2022. [38](#)
- [Pau82] Vern I Paulsen. Completely bounded maps on c^* -algebras and invariant operator ranges. *Proceedings of the American Mathematical Society*, 86(1):91–96, 1982. [8](#)
- [SM14] Gilbert Strang and Shev MacNamara. Functions of difference matrices are toeplitz plus hankel. *siam REVIEW*, 56(3):525–546, 2014. [14](#), [25](#)
- [STU17] Adam Smith, Abhradeep Thakurta, and Jalaj Upadhyay. Is interaction necessary for distributed private learning? In *IEEE Symposium on Security and Privacy*, 2017. [1](#)

- [TS13] Abhradeep Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013. [5](#), [17](#), [18](#)
- [Upa19] Jalaj Upadhyay. Sublinear space private algorithms under the sliding window model. In *International Conference on Machine Learning*, pages 6363–6372, 2019. [1](#)
- [UU21] Jalaj Upadhyay and Sarvagya Upadhyay. A framework for private matrix analysis in sliding window model. In *International Conference on Machine Learning*, pages 10465–10475. PMLR, 2021. [1](#)
- [UUA21] Jalaj Upadhyay, Sarvagya Upadhyay, and Raman Arora. Differentially private analysis on graph streams. In *International Conference on Artificial Intelligence and Statistics*, pages 1171–1179. PMLR, 2021. [1](#)

A Dual Characterization of $\gamma_F(A)$ SDP

Recall that the (primal problem of) SDP for $\gamma_F(A)$ can be written as follows:

$$\begin{aligned} \gamma_F(A) &:= \inf \quad \eta \\ \text{s.t.} \quad &\Phi(X) = \hat{J}_{n,m} \bullet X = \hat{A} \\ &\sum_{i \in [n]} X[i, i] \leq \eta \\ &X[i, i] \leq \eta \quad \forall i \in [n+1, n+m] \\ &X \in \text{Pos}(\mathbb{C}^{n+m}). \end{aligned} \tag{38}$$

We now proceed to characterize the dual of the above SDP via the Lagrangian dual method. To this end, we turn the SDP into an equivalent unconstrained optimization problem by introducing appropriate penalty terms for each constraints. Specifically, let $W \in \text{Herm}(\mathbb{C}^{n+m})$, $S \in \text{Pos}(\mathbb{C}^{n+m})$, $\beta \in \mathbb{R}_+$, and $y \in \mathbb{R}_+^m$ be the penalty terms. The following optimization problem is equivalent to solving the SDP described in eq. (38):

$$\sup_{\beta, y, W, S} \inf_{\eta, X} \left\{ \langle \eta, 1 \rangle + \langle \hat{A} - \Phi(X), W \rangle + \left\langle \sum_{i \in [n]} X[i, i] - \eta, \beta \right\rangle + \sum_{j \in [m]} \langle X[n+j, n+j] - \eta, y[j] \rangle - \langle X, S \rangle \right\}.$$

We need to note that for any feasible solution pair (η, X) of eq. (38), the best choice of penalty variables can only achieve the value η . If a candidate solution pair (η, X) violates any constraint, then the corresponding penalty term can be chosen appropriately to drive the optimal value to an arbitrarily large quantity. Using the minimax inequality, we have that

$$\inf_{\eta, X} \sup_{\beta, y, W, S} \left\{ \langle \eta, 1 \rangle + \langle \hat{A} - \Phi(X), W \rangle + \left\langle \sum_{i \in [n]} X[i, i] - \eta, \beta \right\rangle + \sum_{j \in [m]} \langle X[n+j, n+j] - \eta, y[j] \rangle - \langle X, S \rangle \right\}$$

is at most $\gamma_F(A)$ for $X \in \text{Pos}(\mathbb{C}^{n+m})$. Rearranging the terms, we can rewrite the aforementioned optimization problem as

$$\inf_{\eta, X} \sup_{\beta, y, W, S} \left\{ \langle \hat{A}, Y \rangle + \left\langle X, \begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_m(y) \end{pmatrix} - S - \Phi^*(W) \right\rangle + \left\langle \eta, 1 - \beta - \sum_{j \in [m]} y[j] \right\rangle \right\} \tag{39}$$

Here (as mentioned in Section 2.1) $\Delta_m : \mathbb{C}^m \rightarrow \mathbb{C}^{m \times m}$ is the linear map that maps a vector to a diagonal matrix and $\Phi^* : \mathbb{C}^{(n+m) \times (n+m)} \rightarrow \mathbb{C}^{(n+m) \times (n+m)}$ is the unique linear map (called the *adjoint map* of Φ stated in eq. (22)) such that

$$\langle \Phi(X), W \rangle = \langle X, \Phi^*(W) \rangle \quad \text{for all } X, W \in \mathbb{C}^{(n+m) \times (n+m)}.$$

Note that the above relationship between Φ and Φ^* is for all compatible matrices (not necessarily Hermitian) X and Y of appropriate dimension. We now turn to defining Φ^* . For any $X, W \in \mathbb{C}^{(n+m) \times (n+m)}$

$$\begin{aligned} \langle \Phi(X), W \rangle &= \left\langle \Phi \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}, \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \right\rangle = \left\langle \begin{pmatrix} 0 & X_{12} \\ X_{21} & 0 \end{pmatrix}, \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \right\rangle \\ &= \left\langle \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}, \begin{pmatrix} 0 & W_{12} \\ W_{21} & 0 \end{pmatrix} \right\rangle = \langle X, \Phi^*(W) \rangle. \end{aligned}$$

We make a remark that for the construction of dual from eq. (39), the dual variable W is Hermitian and hence $W_{12} = W_{21}^*$. Now expression 39 can be viewed as an unconstrained version of a constrained optimization

problem where the pair (η, X) is the penalty term. Given that $\eta \in \mathbb{R}$ and $X \in \text{Pos}(\mathbb{C}^{n+m})$, we have the following constraints:

$$\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_m(y) \end{pmatrix} \succeq S + \Phi^*(W) \quad \text{and} \quad \beta + \sum_{j \in [m]} y[j] = 1.$$

This can be rewritten as the following constrained optimization problem, which is the dual of eq. (38).

$$\begin{aligned} \gamma_F(A) &\geq \sup \quad \langle \hat{A}, W \rangle \\ \text{s.t.} \quad &\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_m(y) \end{pmatrix} \succeq \Phi^*(W) \\ &\beta + \sum_{i \in [m]} y[i] = 1 \\ &W \in \text{Herm}(\mathbb{C}^{n+m}), \beta \in \mathbb{R}_+, \text{ and } y \in \mathbb{R}_+^m. \end{aligned} \tag{40}$$

We next show that strong duality holds for eqs 38 and 40.

Lemma 42. *The Slater condition for strong duality holds for both the primal and the dual problem described in eqs 38 and 40, respectively. In particular, there exist primal and dual feasible solutions that achieve the value $\gamma_F(A)$.*

Proof. The Slater condition for the primal problem asks for showing a primal feasible solution $\eta \in \mathbb{R}$ and $X \in \text{Pd}(\mathbb{C}^{n+m})$ such that

$$\Phi(X) = \hat{A} \quad \text{and} \quad \sum_{i \in [n]} X[i, i] < \eta \quad \text{and} \quad X[i, i] < \eta \quad \forall i \in [n+1, n+m],$$

i.e. all constraints are satisfied and the inequalities are satisfied with strict inequalities.

If these conditions are met, then strong duality holds and there exists a dual feasible solution that achieves the optimum value $\gamma_F(A)$. To this end, let

$$X = (\|A\|_\infty + 1) \mathbb{1}_{n+m} + \hat{A} \quad \text{and} \quad \eta = 2nm(\|A\|_\infty + 1).$$

By construction, $\Phi(X) = \hat{A}$. It is evident that

$$\sum_{i \in [n]} X[i, i] = n(\|A\|_\infty + 1) < \eta \quad \text{and} \quad X[i, i] = (\|A\|_\infty + 1) < \eta \quad \forall i \in [n+1, n+m].$$

Moreover, using Lemma 16, we can show that $X \in \text{Pd}(\mathbb{C}^{n+m})$. Moving on to the dual, the Slater condition for it asks for constructing a dual feasible solution $W \in \text{Herm}(\mathbb{C}^{n+m})$, $\beta \in \mathbb{R}_{++}$, and $y \in \mathbb{R}_{++}^m$ such that

$$\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_m(y) \end{pmatrix} \succ \Phi^*(W) \quad \text{and} \quad \beta + \sum_{i \in [m]} y[i] = 1. \tag{41}$$

If these conditions are met, then strong duality holds and there exists a primal feasible solution that achieves the optimum value $\gamma_F(A)$. Set

$$\beta = \frac{1}{2} \quad \text{and} \quad y = \frac{1}{2m} \mathbb{1}_m \in \mathbb{R}_{++}^m \quad \text{and} \quad W = \mathbb{1}_{n+m}.$$

These particular choices form a dual feasible solution that also satisfies eq. (41). Hence Slater condition holds for dual SDP as well. This implies that strong duality holds and the primal and dual SDP achieve the optimum value. This completes the proof of Lemma 42. \square

Reformulating the dual. Now we proceed to reformulate the dual to get to the form we use in our lower bounds. For any dual feasible solution W , we can write

$$\Phi^*(W) = \begin{pmatrix} 0 & Y \\ Y^* & 0 \end{pmatrix} =: \hat{Y}.$$

Given that \hat{A} has the same block diagonal structure as $\Phi^*(W)$ for any W , we have $\langle \hat{A}, W \rangle = \langle \hat{A}, \Phi^*(W) \rangle$. This leads to the following reformulation of the dual.

$$\begin{aligned} \gamma_F(A) &:= \max \quad \langle \hat{A}, \hat{Y} \rangle \\ \text{s.t.} \quad &\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_m(y) \end{pmatrix} \succeq \hat{Y} = \begin{pmatrix} 0 & Y \\ Y^* & 0 \end{pmatrix} \\ &\beta + \sum_{i \in [m]} y[i] = 1 \\ &\hat{Y} \in \text{Herm}(\mathbb{C}^{n+m}), \beta \in \mathbb{R}_+, \text{ and } y \in \mathbb{R}_+^m. \end{aligned} \tag{42}$$

Finally, we reformulate the above dual into the form stated in Section 3. To begin, we can safely assume that $\beta \in \mathbb{R}_{++}$ and $y \in \mathbb{R}_{++}^m$. Let $w, x \in \mathbb{R}_{++}^{n+m}$ be defined as

$$x[i] = \begin{cases} \sqrt{\frac{n}{\beta}} & i \in [n] \\ \frac{1}{\sqrt{y[j]}} & j \in [m] \text{ and } i = n + j, \end{cases} \quad \text{and} \quad w[i] = \frac{1}{x[i]}.$$

It is clear that

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbb{R}_{++}^{n+m} \quad \text{such that} \quad \|w\|_2 = 1 \quad \text{and} \quad w_1 = \alpha \mathbb{1}_n \text{ for } \alpha = \sqrt{\beta/n}.$$

The definition of x and w allows us to get an equivalent form of the first dual constraint in eq. (42), which is stated below.

$$\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta(y) \end{pmatrix} \succeq \hat{Y} \quad \text{if and only if} \quad \begin{pmatrix} n \mathbb{1}_n & 0 \\ 0 & \mathbb{1}_m \end{pmatrix} \succeq \Delta_{n+m}(x) \hat{Y} \Delta_{n+m}(x) = \hat{Y} \bullet x x^*$$

where the final equality follows from Proposition 14. Let us define $\hat{Z} := \hat{Y} \bullet x x^*$. Using Proposition 13, we have

$$\langle \hat{A}, \hat{Y} \rangle = \langle \hat{A}, \hat{Y} \bullet x x^* \bullet w w^* \rangle = \langle \hat{A}, \hat{Z} \bullet w w^* \rangle = w^* (\hat{A} \bullet \hat{Z}) w.$$

Hence the dual can be reformulated as the following optimization problem:

$$\begin{aligned} \gamma_F(A) &= \max \quad w^* (\hat{A} \bullet \hat{Z}) w \\ \text{s.t.} \quad &\begin{pmatrix} n \mathbb{1}_n & 0 \\ 0 & \mathbb{1}_m \end{pmatrix} \succeq \hat{Z} \\ &w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad \text{such that} \quad \|w\|_2 = 1 \text{ and } w_1 = \alpha \mathbb{1}_n \\ &\hat{Z} \in \text{Herm}(\mathbb{C}^{n+m}), \alpha \in \mathbb{R}_{++}, \text{ and } w \in \mathbb{R}_{++}^{n+m}. \end{aligned} \tag{43}$$

B Useful Properties and Bounds on $\gamma_F(\cdot)$

In this section, we establish few facts about $\gamma_F(\cdot)$. Recall that, for any matrix $A \in \mathbb{C}^{n \times m}$, $\gamma_F(A)$ can be written as an SDP where one minimizes a real parameter η over $X \in \text{Pos}(\mathbb{C}^{n+m})$ such that

$$\sum_{i=1}^n X[i, i] \leq \eta \quad \text{and} \quad \Phi(X) = X \bullet \hat{J}_{n,m} = \hat{A} \quad \text{and} \quad X[i, i] \leq \eta \quad \forall i \in [n+1, n+m]. \tag{44}$$

B.1 Useful Properties of $\gamma_F(\cdot)$

We first show that $\gamma_F(\cdot)$ is indeed a norm. While this is known (Nikolov [Nik22] personally communicated a proof of this to us), we provide an (arguably) simpler proof for completeness.

Fact 43. $\gamma_F(\cdot)$ is a norm.

Proof. Let $A \in \mathbb{C}^{n \times m}$ be an arbitrary matrix. It is clear that $\gamma_F(A) = 0$ if and only if $A = 0$ and $\gamma_F(\alpha A) = |\alpha| \gamma_F(A)$ for any $\alpha \in \mathbb{C}$. To see why the triangle inequality holds, let A_1 and A_2 be two matrices such that $A = A_1 + A_2$. Let (η_1, X_1) and (η_2, X_2) be optimal solution for the SDPs corresponding to A_1 and A_2 . For $A = A_1 + A_2$, it is clear that $(\eta_1 + \eta_2, X_1 + X_2)$ is a feasible solution for the SDP. This implies that $\gamma_F(A_1 + A_2) \leq \eta_1 + \eta_2 = \gamma_F(A_1) + \gamma_F(A_2)$. This completes the proof of Fact 43. \square

Fact 44. For a matrix $A \in \mathbb{R}^{n \times m}$, $\gamma_F(A)$ is achieved by a real factorization of A .

Proof. Let $B \in \mathbb{C}^{n \times p}$ and $C \in \mathbb{C}^{p \times m}$ be an optimal factorization of A . That is, $\gamma_F(A) = \|B\|_F \|C\|_{1 \rightarrow 2}$. Let

$$B = B_1 + \iota B_2 \quad \text{and} \quad C = C_1 + \iota C_2$$

for real matrices $B_1, B_2 \in \mathbb{R}^{n \times p}$ and $C_1, C_2 \in \mathbb{R}^{p \times m}$. Since A is a real matrix, we have that

$$A = B_1 C_1 - B_2 C_2 = \begin{pmatrix} B_1 & -B_2 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}.$$

Moreover, the above real factorization achieves $\gamma_F(A)$ completing the proof of Fact 44. \square

We now establish a bound on the dimension of matrix B that gives an optimal factorization for $A = BC$ achieving $\gamma_F(A) = \|B\|_F \|C\|_{1 \rightarrow 2}$.

Lemma 45. Let $A \in \mathbb{C}^{n \times m}$. Then one can construct $\tilde{B} \in \mathbb{C}^{n \times p}$ and $\tilde{C} \in \mathbb{C}^{p \times m}$ such that $p \leq m$ and

$$A = \tilde{B} \tilde{C} \quad \text{and} \quad \gamma_F(A) = \|\tilde{B}\|_F \|\tilde{C}\|_{1 \rightarrow 2}.$$

Proof. The quantity $\gamma_F(A)$ can be written as finding the optimal η such that $X \in \text{Pos}(\mathbb{C}^{n+m})$ satisfying the constraints given in eq. (44). Let (η, X) be a feasible solution. We first show that one can recover an optimal factorization $A = \tilde{B} \tilde{C}$ with respect to $\gamma_F(A)$, where the number of columns of \tilde{B} is at most m . Given that $X \succeq 0$, we can write

$$X = \begin{pmatrix} B \\ C^* \end{pmatrix} \begin{pmatrix} B \\ C^* \end{pmatrix}^* = \begin{pmatrix} BB^* & BC \\ (BC)^* & C^* C \end{pmatrix}. \quad (45)$$

Since X is a feasible solution, we have that $\text{Tr}(BB^*) = \eta$ and $A = BC$. We will now construct a feasible solution of the form

$$\tilde{X} = \begin{pmatrix} \tilde{B} \tilde{B}^* & \tilde{B} \tilde{C} \\ (\tilde{B} \tilde{C})^* & \tilde{C}^* \tilde{C} \end{pmatrix}$$

such that $\text{Tr}(\tilde{B} \tilde{B}^*) \leq \text{Tr}(BB^*) = \eta$ and $\tilde{C}^* \tilde{C} = C^* C$. Moreover, $A = \tilde{B} \tilde{C}$ where the number of columns of \tilde{B} is m .

Since $X \in \text{Pos}(\mathbb{C}^{n+m})$, there exist matrices $B \in \mathbb{C}^{n \times r}$ and $C \in \mathbb{C}^{r \times m}$ for $r \leq n + m$ such that eq. (45) holds. While it is safe to assume that $r \leq n + m$, any finite-dimensional choice of r will also work for the argument presented next. If $r \leq m$, then let $p = r$ and we are done with the proof of the lemma. Hence, for the remainder of the proof, assume that $r > m$. Let

$$C = U \Sigma_C V^* \in \mathbb{C}^{r \times m}$$

be the singular value decomposition of C . Since the number of singular values of C is at most $\min\{n, m\}$, we have $\Sigma_C \in \text{Pd}(\mathbb{C}^p)$ for some $p \leq m < r$. It follows that $U \in \mathbb{C}^{r \times p}$ and $V \in \mathbb{C}^{m \times p}$. By construction, we also have

$$U^*U = \mathbb{1}_p \quad \text{and} \quad UU^* \preceq \mathbb{1}_r.$$

Let

$$\tilde{B} = BU \quad \text{and} \quad \tilde{C} = \Sigma_C V^*.$$

It is clear that $\tilde{B} \in \mathbb{C}^{n \times p}$ and $\tilde{C} \in \mathbb{C}^{p \times m}$ for $p \leq m$, and $\tilde{B}\tilde{C} = BC = A$. Moreover,

$$C^*C = (U\Sigma_C V^*)^*(U\Sigma_C V^*) = V\Sigma_C^2 V^* = (\Sigma_C V)^* \Sigma_C V = \tilde{C}^* \tilde{C}.$$

Finally $\tilde{B}\tilde{B}^* = BUU^*B^* \preceq BB^*$ since $UU^* \preceq \mathbb{1}_r$. Therefore,

$$\text{Tr}(\tilde{B}\tilde{B}^*) \leq \text{Tr}(BB^*) \leq \eta.$$

This completes the proof of Lemma 45. □

A similar result holds for any real matrix as stated below.

Corollary 46. *Let $A \in \mathbb{R}^{n \times m}$. Then one can construct $B \in \mathbb{R}^{n \times p}$ and $C \in \mathbb{R}^{p \times m}$ for $p \leq m$ such that $A = BC$ and $\gamma_{\mathbb{F}}(A) = \|B\|_{\mathbb{F}}\|C\|_{1 \rightarrow 2}$.*

Corollary 46 follows from Fact 44 and working out the proof of Lemma 45 using matrix decompositions involving real matrices only. We first convert an optimal (possibly complex) factorization into a real factorization using Fact 44. Let the optimal real factorization matrices be $B' \in \mathbb{R}^{n \times r}$ and $C' \in \mathbb{R}^{r \times m}$. Fact 44 guarantees that $r \leq 2(n + m)$. As mentioned in the proof above, the proof works as long as r is finite, which is the case here.

B.2 Useful Bounds on $\gamma_{\mathbb{F}}(\cdot)$

A consequence of Lemma 1 is the following result for special classes of square matrices.

Corollary 47. *Let $A \in \mathbb{C}^{n \times n}$ be a non-singular matrix with one singular value of multiplicity n . Then $\gamma_{\mathbb{F}}(A) = \|A\|_{\mathbb{F}}$.*

Proof. For a non-singular matrix A , if all its singular values are same, then $\|A\|_1 = \sqrt{n}\|A\|_{\mathbb{F}}$. Hence $\gamma_{\mathbb{F}}(A) = \|A\|_{\mathbb{F}}$ completing the proof of Corollary 47. □

In particular, for any matrix $A \in \text{U}(\mathbb{C}^n)$, we have $\gamma_{\mathbb{F}}(A) = \sqrt{n}$. A natural question to ask in which cases the lower and upper bound are tight for $\gamma_{\mathbb{F}}(\cdot)$ when they are not equal. Below, we give a partial answer to this question.

Lemma 48. *Let $A \in \mathbb{C}^{n \times n}$ be a diagonal matrix. Then $\gamma_{\mathbb{F}}(A) = \|A\|_{\mathbb{F}}$.*

Proof. Our proof relies on constructing a dual feasible solution that achieves the objective value $\|A\|_{\mathbb{F}}$. By eq. (26) and strong duality of the SDP (Lemma 42), we will have $\gamma_{\mathbb{F}}(A) = \|A\|_{\mathbb{F}}$. For this particular result, we will employ the dual formulation as described in eq. (42) and construct a feasible solution for it. In particular, we are looking for a matrix $Y \in \mathbb{C}^{n \times n}$, $y \in \mathbb{R}_+^n$, and $\beta \geq 0$ such that

$$\beta + \sum_{i=1}^n y[i] = 1 \quad \text{and} \quad \begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_n(y) \end{pmatrix} \succeq \hat{Y} = \begin{pmatrix} 0 & Y \\ Y^* & 0 \end{pmatrix}.$$

Recall that $\Delta_n(y)$ is a linear map that maps an n -dimensional vector y into a $n \times n$ -dimensional diagonal matrix. Since A is a diagonal matrix, its singular values are the absolute values of its diagonal entries. Now we construct our dual feasible solution. Let $\beta = 1/2$, and

$$Y = \frac{A}{2\|A\|_{\mathbb{F}}} \quad \text{and} \quad \Delta_n(y) = \frac{A^*A}{2\|A\|_{\mathbb{F}}^2}.$$

This implies that $y[i] = \frac{|A[i,i]|^2}{2\|A\|_F^2} = \frac{\sigma_i(A)^2}{2\|A\|_F^2}$. Now

$$\beta + \sum_{i=1}^n y[i] = \frac{1}{2} + \frac{1}{2\|A\|_F^2} \sum_{i=1}^n \sigma_i(A)^2 = 1.$$

For our particular choice of Y , we have (using Lemma 16)

$$\begin{pmatrix} \beta \mathbb{1}_n & -Y \\ -Y^* & \Delta_n(y) \end{pmatrix} \succeq 0 \quad \text{if and only if} \quad \Delta_n(y) - \beta^{-1} Y^* Y = \Delta_n(y) - 2Y^* Y \succeq 0.$$

Since

$$2Y^* Y = \frac{A^* A}{2\|A\|_F^2} = \Delta_n(y),$$

we have $\Delta_n(y) - 2Y^* Y = 0$, and therefore,

$$\begin{pmatrix} \beta \mathbb{1}_n & 0 \\ 0 & \Delta_n(y) \end{pmatrix} \succeq \hat{Y}.$$

This implies that all the dual constraints are satisfied. Next, we proceed to compute the objective value corresponding to this dual feasible solution which is at most $\gamma_F(A)$. We have

$$\gamma_F(A) \geq \langle \hat{A}, \hat{Y} \rangle = \text{Tr}(A Y^*) + \text{Tr}(A^* Y) = \frac{1}{\|A\|_F} \sum_{i \in [n]} |A[i, i]|^2 = \|A\|_F.$$

This completes the proof of Lemma 48. □