2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT) | 978-1-6654-1924-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/WIIAT50758.2020.00006

Projection-free Online Empirical Risk Minimization with Privacy-preserving and Privacy Expiration

Jian Lou
Department of Computer Science
Emory University
Atlanta, USA
jian.lou@emory.edu

Yiu-ming Cheung
Department of Computer Science
Hong Kong Baptist University
Hong Kong SAR, China
ymc@comp.hkbu.edu.hk (Corresponding Author)

***Abstract*—Streaming machine learning and data mining problems are prevalent in real world applications, where individual data are collected and revealed consecutively. These problems can often be modeled and solved under Constrained Online Convex Optimization (COCO) algorithmic framework. The ever-growing amount of sensitive individual data is posing greater challenge to the contradictory goals of privacy protection and reasonable model usability. Despite its extensive studies via projected/proximal gradient based methods, its projection-free counterpart has not been well-explored. Inspired by the better per-iteration computational efficiency and privacy-utility tradeoff under non-private/non-online settings of the projection-free algorithms, we propose the projection-free COCO with differential privacy guarantee, a de facto standard for privacy preserving. We rigorously analyze its utility in terms of regret rate, which shows that, even without the expensive projection/proximal operators, it still matches the differentially privacy COCO with projection/proximal operations. To the best of our knowledge, it is the first projection-free differentially private COCO, and thus broadens the applicability of COCO with privacy guarantee. Furthermore, since protecting the privacy of all incoming samples will lead to inferior regret rate compared to the nonprivate optimal, we propose a relaxed privacy guarantee which trades the privacy of remote samples for better utility. To achieve this, we adopt a window tree mechanism for maintaining a private gradient summation, which is then used to construct an approximation function for updating the new response variable at each timestamp. It improves the regret bound to $O(\ln T)$ with respect to the sequence length T, matching the nonprivate optimal regret.**

***Index Terms*—Online Convex Optimization, Differential Privacy, Proximal Online Gradient Descent, Projection-free Online Gradient Descent**

I. Introduction

In many online practical applications, data are collected and fed to model learning in a streaming fashion [1]. Prominent examples can vary from web browsing behavior analysis [2] with ever increasing social media data, website cookies and user click history, to monitoring systems that continuously collect sensory data for applications like personal trajectory monitoring and location-based services [3], and many other real-time systems [4] used for industrial manufacture or financial transaction data analysis. Constraint online convex optimization (COCO) [1], [5]–[9] is a popular approach to dealing with various streaming machine learning and data mining problems arising from such real-time learning and monitoring systems environment. Within COCO context, with data instances arriving sequentially over time, the COCO algorithm responses continuously by releasing new decision variable from a bounded convex set at each timestamp. Each time the COCO algorithm plays a variable, it observes a convex loss and the aim is to minimize the regret, a utility measure for COCO algorithms.

The outputs of the learning algorithm can put the privacy of sensitive data at risk, which is especially severe for online systems as an adversary can potentially observe the entire response sequence that contains significantly more information than only one final output produced by an offline system. To preserve the privacy of individual data instances, the rigorous statistical notion of differential privacy (DP) [10]–[12] has become nowadays the standard technique for privacy guarantee. DP restricts that any single change of the data instances will make little difference to the output of a random algorithm. The utility of the algorithm often decreases due to the additional randomness introduced for privacy protection. As such, algorithm designers need to strike the trade-off between privacy and utility, which becomes even more delicate for the online algorithm with the continuing release of the output sequence. With streaming data accumulation, it becomes difficult to protect each instance's privacy without much compromise on utility. In the literature, some works have proposed to maintain satisfactory utility by relaxing the DP restriction on some instances or events. For example, [13] considers w-event privacy to only protect the privacy of events of every w continuous timestamps in a sliding window, while [14] consider l-trajectory privacy that only ensures differential privacy of every length l trajectory. In particular, [15] observes that recent data are more important than distant inputs and therefore proposes the privacy expiration assumption by protecting only the privacy of recent data instances inside a sliding window. However, these are results within the database community that publishes either private data vectors or simple statistics like sum or histogram. It has yet to be known whether similar relaxation of privacy in some instances can improve the performance of private COCO methods.

For constrained convex optimization, a lot of differentially convex optimization algorithms [16]–[21] have been designed provided that the projection/proximal operation [22] onto the

978-1-6654-1924-6/20/$31.00 ©2020 IEEE
DOI 10.1109/WIIAT50758.2020.00006

constrained set can be efficiently computed (i.e. projection-friendly constraints). However, many constraints adopted in machine learning have high computational complexity for computing the projection/proximity, for which a projection-free optimization would be preferred for better scalability [23]–[26]. Such projection-free methods, called conditional gradient algorithms or Frank-Wolfe algorithms, compute a more efficient linear oracle-based operator to avoid projection in each iteration and have been recognized as a better alternative for the linear operator-friendly constrained problems. In addition to their lower per-iteration computational complexity, Frank-Wolfe type algorithms have been found to have better utility under differential privacy restrictions. For example, Talwar et al. [27] developed a differentially privacy conditional gradient for the LASSO private and proved that it achieves nearly optimal utility. Later, it is extended to distributed setting [20], [21] where the nearly optimal utility is obtained along with a better uplink communication efficiency. More recently, [28], [29] show that differentially private conditional gradient helps improve the utility even for empirical risk minimization (ERM) with nonconvex loss functions.

Specifically, for the online constrained ERM, the differentially private COCO is required to output a continuous sequence of response variable at each time stamp. [19], [30], [31] have considered differentially private OCO algorithms. [19] and [30] are based on the regret optimal online scheme called "Follow The Approximate Leader"(FTAL) [6], and [32] designs private algorithm based on the noise and space efficient tree mechanism. In addition, their accumulative regrets still grow faster than nonprivate counter part. [19] and [31] also design online OCO algorithms for special linear loss functions. The former has $O(\ln T)$ regret achieving the known optimal even for nonprivate case, while the latter reduces dependence on the dimension of the problem. However, an extension to more general loss functions for both results is unclear. These methods are exclusively developed provided that the constraint sets are projection-friendly.

Motivated by the computational scalability Frank-Wolfe algorithm and the superior utility of its two differentially private variants under deterministic setting, we investigate whether we can improve the computational efficiency as well as the utility of the private COCO algorithm by developing its projection-free counterpart. In addition, to take the advantage of the relaxed privacy restriction on distant data timely and safely, we adopt the window tree mechanism with either Gamma or Gaussian perturbation to maintain a private gradient summation approximation satisfying the window differential privacy. We then utilize the popular FTAL scheme to update the response variable by optimizing an approximate function constructed with the private gradient summation. Obtained with the DP-gradient summation, we develop the projection-free online conditional gradient algorithm and rigorously analyze its regret bound. We are able to achieve improved regret growth of order $O(\ln T)$ with respect to the sequence length T, which matches the nonprivate optimal regret bound. The regret bound also captures the effect of the window length and provides an explicit trade-off between privacy and utility. To the best of our knowledge, it is the first private COCO algorithm to consider a projection-free update in the private COCO, providing an alternative choice for private online learning tasks on linear oracle-friendly constraint sets.

Contributions. To summarize, we aim to provide the **theoretical understanding** of whether an ideal regret bound is achievable for the differentially private COCO, where the regret bound is able to match its nonprivate counterpart. In brief, we have the following contributions: 1) Reduce regret to $O(T)$ for private COCO by trading privacy of remote inputs, so that the utility matches the same order of the nonprivate counterparts; 2) For privacy-protection window size W, the regret has a factor $\ln(W)$, which captures the trade-off between privacy coverage and utility; 3) Provide a projection-free COCO variant with improved regret bound, which is the first private conditional gradient method under online setting. All these results come with rigorous proof. Due to space limit, we relegate proofs of theorems in this section to the supplement which can be accessed via the link[1].

II. Background

A. Problem Setup

Given a streaming sequence of loss functions $\mathcal{I} = [f_1, f_2, ..., f_t, ...f_T]$ arriving one at a time, the COCO algorithm is required to response x_t from the constraint set $\mathcal{C}$, which is a bounded convex set. After each response, it will suffer a convex loss $f_t(x_t)$. In machine learning, depending on the task, the function f_t can have various choices, e.g., logistic loss or hinge loss for classification, and square function for linear regression. For the sake of simplicity, this paper abstracts that $f_t(x)$ is L-Lipschitz continuous and μ-strongly convex w.r.t. x, described by the following two definitions.

Definition 1: (Lipschitz Continuous) A function $f(x) : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous with parameter L on the set $\mathcal{C}$ if $\forall x, y \in \mathcal{C}$ it holds that,

$$|f(x) - f(y)| \leq L||x - y||_2. \tag{1}$$

Definition 2: (Strong Convexity) A function $f(x) : \mathbb{R}^d \to \mathbb{R}$ is strongly convex with parameter μ on the set $\mathcal{C}$ if $\forall x, y \in \mathcal{C}$ it holds that,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}||x - y||_2^2. \tag{2}$$

We also assume that the bounded convex constraint set has diameter D, i.e. $D = \max_{\forall x,y \in \mathcal{C}} ||x - y||_2$.

We measure the utility of the algorithm by regret, a common notion used in online algorithms, defined as:

Definition 3: (Regret) Denote the private release of the algorithm by $x_1, x_2, ..., x_t, ..., x_T$, then the regret with sequence length T is

$$Regret(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{C}} \sum_{t=1}^{T} f_t(x). \tag{3}$$

[1]http://www.comp.hkbu.edu.hk/~ymc/papers/conference/wi20/wi20-supplement.pdf

B. Differential Privacy and Privacy Expiration

The differential privacy has become the standard statistical notion in privacy preserving. It guarantees that the output of the algorithm will remain roughly the same despite the change of any particular individual of the input sequence. For COCO setting, with a neighboring I' differing from $\mathcal{I}$ with a single entity $f_t' \neq f_t$ (for any $t \in [T]$), and the output being private decision variables $[\hat{x}_2, ..., \hat{x}_t, ..., \hat{x}_{T+1}] \in \mathcal{C}^T$, the differential privacy is defined as:

Definition 4: (Differential Privacy (DP) [10]; [11]) A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if, for every two neighboring sequences $\mathcal{I}$ and $\mathcal{I}' \in \mathcal{F}^T$ that differ in only one entry, and for every event $\mathcal{O}$ in the output space $\mathcal{C}^T$,

$$P[\mathcal{A}(\mathcal{I}) \in \mathcal{O}] \leq e^{\epsilon} P[\mathcal{A}(I') \in \mathcal{O}] + \delta. \tag{4}$$

The algorithm is said to be ϵ-differentially private if $\delta = 0$.

To enable a deterministic algorithm to provide DP protection, it is common to randomize it by adding designed random noise values sampled from proper random distributions, including Laplacian, Gamma, and Gaussian distributions. The noise addition mechanisms are often considered to be the basic building blocks of differential privacy designs, among which the Gaussian and Gamma mechanisms are utilized in this paper whose details are summarized in the appendix. The sensitivity of the algorithm is an important concept in deciding the parameter of the random distribution, which measures the maximum change of the algorithm when a particular individual instance changes.

Definition 5: (ℓ_2 Sensitivity) Let $\mathcal{A}$ be an algorithm mapping stream $\mathcal{I}$ to $\mathcal{O}$. The ℓ_2 sensitivity of $\mathcal{A}$ is $\Delta_2 = \max_{\mathcal{I},\mathcal{I}'} ||\mathcal{A}(\mathcal{I}) - \mathcal{A}(\mathcal{I}')||_2$, where $\mathcal{I}$ and $\mathcal{I}'$ differ only in one instance.

Considering the simple yet practical privacy expiration assumption, the window differential privacy definition only delivers privacy protection for recent instances inside a sliding window [15]. That is, only the changes of output caused by the latest W individual entries are counted into the privacy loss, while the changes of the output caused from distant inputs are not concerned. The following definition formalizes the window differential privacy, which is adapted from [15] to our COCO setting.

Definition 6: (Window Differential Privacy (WDP)) Let $\mathcal{A}$ be a random algorithm. With a specific sequence length T, for any input convex function sequence $\mathcal{I}$ and a neighboring sequence $\mathcal{I}'$, $\mathcal{O}$ is an event (i.e., the output sequence space), $\mathcal{A}$ is the window (ϵ, δ)-differential privacy with a window size W if the following condition holds,

$$\begin{aligned} &P[\mathcal{A}([f_1, f_2, ..., f_t, ..., f_T]) \in \mathcal{O}] \\ &\leq e^{w(T-t)\epsilon} P[\mathcal{A}([f_1, f_2, ..., f_t', ..., f_T]) \in \mathcal{O}] + \delta, \end{aligned} \tag{5}$$

where $w(i) = 1$ for $i < W$, $w(i) = \infty$ for $i \geq W$. The algorithm is said to be window ϵ-differentially private with the window size W if $\delta = 0$.

Note that neither the design of an efficient algorithm that timely takes the advantage of the just-relaxed privacy nor the quantification of the privacy and utility trade-off is trivial to carry out because, unlike the offline setting, streaming algorithms deal with dynamic input and output environment and the privacy window keeps shifting as new instances arrive.

C. Differentially Private Follow The Approximate Leader

The private COCO algorithms [19] and [30] are based on FTAL [1], [6], which only requires first order information and is known to be regret optimal. For μ-strongly convex loss function $f_\tau(x)$, it updates x_{t+1} by minimizing an approximation function of the streaming ERM loss function $\sum_{\tau=1}^{t} f_\tau(x)$ (a.k.a. Follow The Leader (FTL) updating) as

$$F_t(x) = \sum_{\tau=1}^{t} \tilde{f}_\tau(x), \text{ where} \tag{6}$$

$$\tilde{f}_\tau(x) = f_\tau(x_\tau) + \langle \nabla f_\tau(x_\tau), x - x_\tau \rangle + \frac{\mu}{2} ||x - x_\tau||_2^2, \tag{7}$$

which is always a lower approximation by the strong convexity assumption. To minimize $F_t(x)$, it is equivalent to minimize

$$J_t(x) = \langle \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau), x \rangle + \frac{\mu}{2} \sum_{\tau=1}^{t} ||x - x_\tau||_2^2. \tag{8}$$

By observing that the continuous gradient summation $g_t = \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau)$ is the only ingredient having access to sensitive private information in the above procedure, it suffices to use a differentially private surrogate of $g_t = \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau)$ in the computation to ensure the entire COCO algorithm is DP as well. [19], [30] abstract out the step of the private gradient summation maintenance as a continuous running sum task and resort to the tree mechanism. The tree mechanism [32], [33] has been proven to add minimum noise to ensure DP can be implemented in a space and time efficient manner. Hence, it is an ideal candidate for maintaining the private gradient sum. Denote the private surrogate of gradient summation computed by the tree mechanism by $\hat{g}_t$, and re-denote the private release history of the private COCO up to timestamp t by $\hat{x}_1, \hat{x}_2, ..., \hat{x}_t$, the private updating replaces g_t by $\hat{g}_t$ to minimize

$$\hat{J}_t = \langle \hat{g}_t, x \rangle + \frac{\mu}{2} \sum_{\tau=1}^{t} ||x - x_\tau||_2^2, \tag{9}$$

[19] and [30] propose to solve it exactly by projection $\mathcal{PO}$ with

$$\hat{x}_{t+1} = \mathcal{PO}\big(\sum_{\tau=1}^{t} \hat{x}_\tau - \hat{g}_t/\mu, \mathcal{C} \big), \tag{10}$$

which potentially limits their applicability to only tasks with projection-friendly constrains, i.e. constraint sets where projection can be efficiently evaluated, like Euclidean ball.

III. Proposed Method

This section presents our private projection-free COCO algorithm with the improved regret bound by considering the privacy expiration assumption. Following the FTAL framework, we start with a window tree mechanism that maintains two binary subtrees with the depth at most $(\lceil \log_2 W \rceil + 1)$

TABLE I
COMPARISON OF PRIVATE COCO ALGORITHMS

Algorithm	Privacy guarantee	per-iteration OP	Regret Bound
POCG [19]	(ϵ, δ)-DP	two projections	$O(\frac{\sqrt{d}\sqrt{T}\ln^2 \frac{T}{\delta}}{\epsilon})$
PQFTL [19]	(ϵ, δ)-DP	linear equation	$O(\frac{\sqrt{d}\ln \frac{1}{\delta}\ln^{1.5} T}{\epsilon})$
PFTAL [30]	ϵ-DP	one projection	$O(\frac{d\ln^{2.5} T}{\epsilon})$
Ours with projection	window ϵ-DP	one projection	$O(\frac{d\log_2^{1.5} W\ln T}{\epsilon})$
Ours with projection	window (ϵ, δ)-DP	one projection	$O(\frac{\sqrt{d}\log_2^{1.5} W\ln \frac{1}{\delta}\ln T}{\epsilon})$
Ours with linear	window ϵ-DP	one linear oracle	$O(\frac{d\log_2^{1.5} W\ln T}{\epsilon})$
Ours with linear	window (ϵ, δ)-DP	one linear oracle	$O\Big(\max\big(\sqrt{d}\epsilon, \log_2^{1.5} W\ln \frac{1}{\delta}\big)\cdot \frac{\sqrt{d}(L+\mu D)^2\ln T}{\mu\epsilon}\Big)$

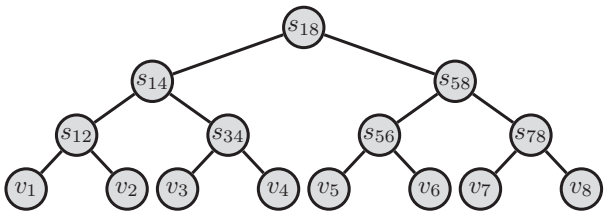

Fig. 1. Illustration of the binary tree mechanism

to maintain the window private gradient summation for constructing $\hat{J}_t(x)$ in eq.(9). Before presenting the projection-free COCO, we first solve $\hat{J}_t(x)$ exactly by projection for projection-friendly constraints, which gives us a direct comparison with the previous methods [19], [30] without privacy expiration assumption (please see Table I for comparison). We then develop the DP projection-free COCO, which still maintains the optimal $O(\ln T)$ regret. The regret analysis is nontrivial compared to its nonprivate counterpart because of the dynamic noise injected at each time stamp for the purpose of differential privacy protection. The regret analysis shows the benefit of privacy expiration in improving regret bound to $O(\ln T)$ given a fixed window size, matching the nonprivate optimal w.r.t. the sequence length T. Also, it captures the trade-off between the privacy coverage and the regret bound.

A. A Mini-example for Tree Mechanism and Window Tree Mechanism

As for the tree mechanism [32], [33], which is central to enabling the privacy preserving ability for FTAL algorithms, we briefly introduce it with the mini-example illustrated in Figure 1 for releasing the continuous summation of $v_1, v_2, ..., v_8$ with ϵ-DP guarantee. For rigorous algorithm description, please refer to [33] and [32]. The tree mechanism constructs a complete binary tree where the value of each individual instance is stored in the leaf node, i.e. $v_1, v_2, ...v_8$ in the example. Internal nodes store the partial sum of its child nodes. For example, node s_{12} keeps the sum of v_1 and v_2 while s_{14} keeps the sum from v_1 to v_4. Each node also keeps a private copy of its value by adding noise vector sampling from the properly scaled random distribution, each providing $(\epsilon/\log_2 8)$-DP ($(\epsilon/\log_2 T)$ for general game length T). The rationale is that any partial sums can be computed by referring at most $\log_2 8$ ($\log_2 T$ for general game length T) nodes in the binary tree. For example, when computing s_{1-7}, it sums noisy values kept in nodes s_{14}, s_{56} and v_7. Hence, any particular instance change will affect at most 3 ($\log_2 T$) nodes in the tree. By composition theorem, the continuous private running sum release would be ϵ-DP. Note that tree mechanism adds noise as minimum as possible, either adding noise to each instance value itself followed by summation or adding noise to the summed clean instances will require significantly greater amount of noise to ensure $\epsilon - DP$ under continuous releasing context, and the utility (i.e. accuracy of the approximated sum) deteriorates.

We then describe the intuition of the window tree mechanism, still based on the same mini-example. It extends the tree mechanism to window tree mechanism that considers the decayed privacy when releasing continuous sum. Still, we keep the instance values on leaf nodes and partial sums on inner nodes. Given a window size W (we assume W is the exact power of 2 hereafter for the sake of simplicity), the complete binary tree can be divided into blocks and each is a $\log_2 W+1$ depth tree with W leaves. The sliding window can at most span two blocks of the $\log_2 W + 1$ depth trees. Back to the mini-example in Figure 1, with the window size 4, the binary tree is divided into the two depth-3 subtrees rooted at node s_{14} and s_{58}, respectively. At time $t = 7$, the private window involves node v_4 to v_7, spanning two subtrees. At time $t = 8$, the private window involves nodes v_5 to v_8, affects only subtree s_{58}, and at this moment we can only keep the clean value kept by node s_{14} and discard the entire subtree root at s_{14}. The computation of s_{1-7} (i.e. the sum of v_1 to v_7) is by summing clean value in s_{12}, v_3 and noisy values in v_4, s_{56}, v_7, that at most $\log_2 4 + 1$ noisy nodes are involved. For general W and time t, we only need to keep the complete structures of at most two rightmost subtrees of depth $\log_2 W+1$ (i.e. $(k-1)$-th and k-th subtrees, $k = \lceil \frac{t}{W} \rceil$) with both clean and noisy value. For the nodes residing left to these two subtrees, it suffices only to keep their clean sum and recall the tree structure. The calculation of the window partial sum only involves $\log_2 W + 1$ noisy nodes. Bolot et al. [15] first proposed the window tree mechanism for the private running sum under ϵ-DP constraint. They added Laplacian perturbation with a proper parameter based on the ℓ_1 sensitivity of the problem. Also, their studies were limited to simple statistics releasing, e.g., running sum or linear

map. In the following, in order to adopt the tree mechanism for our private COCO problem, we extend the window tree mechanism by adding Gamma and Gaussian perturbation with the proper parameters according to ℓ_2 sensitivity for providing ϵ and (ϵ, δ)-DP, respectively, and the noise injecting involves at most two rightmost $(\log_2 W + 1)$-depth subtrees.

B. Window Tree Mechanism for Private Gradient Summation with Decayed Privacy

1) Window Tree Mechanism with Gamma Perturbation: We consider the window tree mechanism for general streaming data $v_1, v_2, ..., v_t, ...$ to preserve ϵ differential privacy with a window size W for releasing summation sequence $(s_1, s_2, ..., s_t, ...)$. We denote the noise vectors added to the nodes in the two rightmost subtrees (explained in the supplement) by n_i. The next lemma describes the distribution parameter of n_i and shows the window ϵ-DP guarantee with Gamma perturbation denoted by Γ.

Lemma 1: With $n_i \in \mathbb{R}^d$ sampled to satisfy

$$||n_i||_2 \sim \Gamma(d, \frac{\Delta_2(\lceil \log_2 W \rceil + 1)}{\epsilon}), \tag{11}$$

where Γ denotes the Gamma distribution (i.e. $||n_i||_2$ proportional to $e^{-\frac{||n_i||_2 \epsilon}{\Delta_2(\lceil \log_2 W \rceil + 1)}}$) and is added to each relevant node, the window tree mechanism achieves the window ϵ-differential privacy with the window size W.

The next lemma describes the utility of the window tree mechanism in terms of the ℓ_2-norm distance between the private and clean running sums. Note that we are generally interested in the case $t > W$ and left along $t \leq W$, which is identical to the original tree mechanism.

Lemma 2: For any $\beta > 0$ and t, with probability at least $1 - \beta$, s_t computed by the window tree mechanism with Gamma noise at each node of $||n_i||_2 \sim \Gamma(d, \frac{\Delta_2(\lceil \log_2 W \rceil + 1)}{\epsilon})$ satisfies:

$$||s_t - \sum_{i=1}^{t} v_i||_2 \leq (\frac{d\Delta_2 \log_2^{1.5} W \ln \frac{d}{\beta}}{\epsilon}), \tag{12}$$

where Δ_2 is the ℓ_2 sensitivity of the sum function.

2) Window Tree Mechanism with Gaussian Perturbation: Similarly, to ensure window (ϵ, δ)-DP ($\delta > 0$), we add Gaussian noise to the nodes of the subtrees involved in the privacy window.

Lemma 3: With noise vector $n_i \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_{d \times d})$, where

$$\sigma^2 = \frac{8\Delta_2^2(\lceil \log_2 W \rceil + 1)^2 \ln^2(2/\delta)}{\epsilon^2}, \tag{13}$$

and added to each relevant node, the window tree mechanism achieves the window (ϵ, δ)-differential privacy with a window size W.

Similarly, we provide the utility bound.

Lemma 4: For any $\beta > 0$ and t, with probability at least $1 - \beta$, s_t computed by window tree mechanism satisfies:

$$||s_t - \sum_{i=1}^{t} v_i||_2 = O((\sqrt{d}\Delta_2 \log_2^{1.5} W \ln \frac{1}{\delta} \sqrt{\ln \frac{1}{\beta}})/(\epsilon)) \tag{14}$$

Algorithm 1 Window Differentially Private COCO with Projection: WDP-COCOP

Input: $x_1 = \hat{x}_1, T, W, \mu, L, \epsilon$ *or* (ϵ, δ), loss function sequence $f_1, f_2, ..., f_T$.

1: **for** $t = 1, 2, ..., T$ **do**
2: Construct $\hat{g}_t$ by window tree mechanism $\mathcal{WTM}$:
OPTION I: $\mathcal{WTM}_{Gamma}(\nabla f_t(\hat{x}_t), W, \epsilon)$ for window ϵ-DP;
OPTION II: $\mathcal{WTM}_{Gaussian}(\nabla f_t(\hat{x}_t), W, \epsilon, \delta)$ for window (ϵ, δ)-DP;
3: $\hat{x}_{t+1} = \mathcal{PO}((\sum_{\tau=1}^{t} \hat{x}_\tau - \hat{g}_t/\mu), \mathcal{C})$;
4: **Output:** $\hat{x}_{t+1}$;
5: **end for**

where Δ_2 is the ℓ_2 sensitivity of the sum function.
Later, we will show that, with Gaussian perturbation, the compromise on privacy with a small probability δ can improve the regret bound dependence of the dimension d in private COCO from $O(d)$ to $O(\sqrt{d})$, when compared with Gamma perturbation based methods like PFTAL in [30]. That is, the private COCO can inherit the improved utility of Lemma 4.

C. Window Differentially Private COCO with Projection

Before presenting our DP projection-free COCO, we briefly describe the DP COCO with projection when the privacy expiration is used. It is called **W**indow **D**ifferentially **P**rivate **COCO** with **P**rojection (WDP-COCOP) as summarized in Algorithm 1. Subsequently, we are able to 1) have a direct comparison with the DP-COCO without privacy expiration technique because they are exclusively projection-based; 2) set the context for the projection-free counterpart in terms of the regret bound, which shows that the projection-free operator does not compromise the utility.

We follow the FTAL framework and provide window privacy protection by utilizing the private approximation of the gradient summation maintained by window tree mechanism presented in the previous subsection. That is, at the t-th timestamp, we update the response $\hat{x}_{t+1}$ by exactly solving $\hat{J}_t(x)$ with $\hat{g}_t$ computed by window tree mechanism. In detail, Step 2 has two options to add either Gamma or Gaussian perturbation to the relevant nodes in the window tree, depending on the privacy type (i.e. ϵ or (ϵ, δ)). The regret bound provides us a direct comparison with the previous algorithms based on similar building blocks but without privacy expiration assumption, clearly revealing the effect of the window differential privacy in COCO as a result.

Privacy Guarantee: The WDP-COCOP algorithm preserves the window differential privacy with the proper noise parameters as described by the following two theorems. In brief, they are by Lemma 1 & 3 with ℓ_2 sensitivity being $(L + \mu D)$, plus the post-processing property of differential privacy (Proposition 2.1 in [12]).

Theorem 1: (window ϵ-DP of Algorithm 1) Let $f_1, .., f_T$ be streaming L-Lipschitz continuous and μ-strongly convex functions. The diameter of the bounded convex set is D.

Algorithm WDP-COCOP with *OPTION I* is the window ϵ-differential privacy with the window size W if window tree mechanism samples Gamma noise from distribution

$$||n_i||_2 \sim \Gamma(d, \frac{(L+\mu D)(\lceil \log_2 W \rceil + 1)}{\epsilon}). \quad (15)$$

Theorem 2: (window (ϵ, δ)-DP of Algorithm 1) Let $f_1, .., f_T$ be streaming L-Lipschitz continuous and μ-strongly convex functions. The diameter of the bounded convex set is D. Algorithm WDP-COCOP with *OPTION II* is the window (ϵ, δ)-differential privacy with the window size W if the window tree mechanism samples Gaussian noise from the distribution

$$n_i \sim \mathcal{N}(0, \frac{8(L+\mu D)^2(\lceil \log_2 W \rceil + 1)^2 \ln^2(2/\delta)}{\epsilon^2} \mathbb{I}_{d\times d}). \quad (16)$$

Regret Analysis: The regret analysis for Algorithm 1 WDP-COCOP with both privacy options is provided.

Theorem 3: (regret guarantee with window ϵ-DP) Under the same condition as in Theorem 1, for any $\beta > 0$ and window size W, we have with probability at least $1-\beta$, the following regret bound holds,

$$Regret(T) = O((d(L+\mu D)^2 \log_2^{1.5} W \ln \frac{d}{\beta} \ln T)/(\mu\epsilon)).$$

Theorem 4: (regret guarantee with window (ϵ, δ)-DP) Under the same condition as in Theorem 2, for any $\beta > 0$ and window size W, we have with probability at least $1-\beta$, the following regret bound holds,

$$Regret(T) = O((\sqrt{d}(L+\mu D)^2 \log_2^{1.5} W \ln \frac{1}{\delta} \sqrt{\ln \frac{1}{\beta}} \ln T)/(\mu\epsilon))$$

Discussion: Compared with PQFTL [19] and PFTAL [30], with a fixed window size W, our regret bound is of order $O(\ln T)$, which is better than theirs and is also known to match the nonlinear optimal. Also, PQFTL [19] is limited to quadratic loss functions. Taking the window size W related term into comparison, our regret bounds provide an explicit trade-off between privacy coverage and regret bound, i.e. the regret scales with the sliding window size. Compared to the more generally applicable PFTAL [30], even with full sequence privacy protection (i.e. $W = T$), our method with Gaussian perturbation (option II) has better dependence on d (i.e. $O(\sqrt{d})$ to $O(d)$), which can be more favored in high-dimensional tasks. Table I provides a summary of private COCO algorithms.

D. Window Differentially Private COCO with Linear Oracle

We present a private COCO called **W**indow **D**ifferentially **P**rivate **COCO** with **L**inear oracle (WDP-COCOL) in Algorithm 2 targeting at tasks with linear oracle (LO)-friendly constraint sets, e.g., matroid polytope, flow polytope, and spectrahedron [23]. The conditional gradient method [34], [35], also known as the Frank-Wolfe method, has become increasingly popular for convex (and even nonconvex [36]) optimization over LO-friendly constraint sets, where either the improved computational efficiency or the better ability to preserve the sparse structure of the desired variable when computed with the CG methods has been observed [23]. In particular, [27] provides a CG-based private algorithm in the offline ERM setting and shows that it achieves nearly optimal utility for Lasso regression under DP restriction. [37], [38] consider online CG methods, however, in the nonprivate setting. It is still unclear whether CG can be carried out in a private manner within the streaming context. In this subsection, we show that we can construct a window tree mechanism based private leader that is capable of incorporating a variant of conditional gradient method. **To the best our knowledge, this is the first differentially private online convex learning algorithm that only requires solving a linear minimization problem at each iteration.** More importantly, our regret analysis shows that our private online CG method is able to provide the same regret bound $O(\ln T)$ with respect to the game length T given a fixed window size W, which is the same as 1) the more computational demanding an online projected gradient method would provide, and 2) the nonprivate optimal regret bound. In the following, we first describe our differentially private online CG method, which is followed by privacy as well as utility guarantee in terms of regret analysis.

We study the private COCO with the CG step based on the nonprivate FTAL-type conditional gradient variant [38] and consider a constraint set to be generally bounded polytope $\mathcal{P}$. It uses a slightly different approximation function $F_t(x)$ with eq.(6),

$$F_t(x) = \sum_{\tau=1}^{t} \tilde{f}_\tau(x) + \frac{C_0\mu}{2}||x - x_1||_2^2. \quad (17)$$

Thus, our modified private $\hat{J}_t(x)$ is

$$\hat{J}_t = \langle \hat{g}_t, x\rangle + \frac{\mu}{2}\sum_{\tau=1}^{t}||x - \hat{x}_\tau||_2^2 + \frac{C_0\mu}{2}||x - x_1||_2^2, \quad (18)$$

where $\hat{g}_t$ can again be maintained by window tree mechanism. The key ingredient to the CG based method is a linear oracle evaluation step. Recently, various modifications have been proposed based on the original LO. This paper, in specific, follows [38] to use the *local linear oracle*, denoted by $\mathcal{LLOO}$, which returns a linear oracle p given the variable x and the gradient g. Roughly, $\mathcal{LLOO}$ chooses a linear oracle from a smaller range that p is in the $\rho \cdot r$ ball centered around x (denoted by $\mathbb{B}(x, r)$), rather than from the entire set $\mathcal{P}$. [38] shows that, on general polytopes, $\mathcal{LLOO}$ still computes one local LO evaluation per-timestamp. The functioning of $\mathcal{LLOO}$ is summarized in the following assumption.

Assumption 1: Denote the local linear oracle by $\mathcal{LLOO}$ that, for the variable x, radius r and gradient estimation g. It evaluates the linear oracle $p \in \mathcal{P}$, i.e. $p = \mathcal{LO}(x, r, g)$. We assume the linear oracle p satisfies the following two properties.

1) $\forall y \in \mathbb{B}(x, r) \cap \mathcal{P}$, it holds that $\langle p, g\rangle \leq \langle y, g\rangle$;
2) $||x - p||_2 \leq \rho r$.

In Algorithm 2, Line 8 computes the new private decision variable $\hat{x}_{t+1}$ by linear combination, which avoids the pro-

jection/proximal operation. Although our method is based on nonprivate [38], the design of parameters and analysis are different. For example, the settings of C_1, C_W take the window size and DP parameters ϵ, δ into consideration, which are exclusive to window differential privacy. Also, the regret analysis needs to handle the approximate linear oracle output of $\mathcal{LLOO}$ because of the randomness inherited in $\hat{g}_t$ for privacy, rather than the deterministic clean g_t as considered in [38].

Algorithm 2 Window Differentially Private COCO with Linear Minimization: WDP-COCOL

Input: $x_1 = \hat{x}_1, T, W, \mu, L, c, d, \epsilon$ *or* (ϵ, δ), loss function sequence $f_1, f_2, ..., f_T$.

initialize: $\rho = c\sqrt{d}$ (c is a geometry constant of $\mathcal{P}$), $\alpha = \frac{1}{5\rho^2}$, $C_0 = (25\rho^2)^2$;

initialize (cont.): $C_W = \frac{2\alpha d \log_2^{1.5} W \ln \frac{d}{\beta}}{(1-\alpha)\epsilon}$ (for W-ϵ-DP); or $C_W = \frac{2\alpha\sqrt{d} \log_2^{1.5} W \ln \frac{1}{\delta} \ln \frac{d}{\beta}}{(1-\alpha)\epsilon}$ (for W-(ϵ, δ)-DP);

initialize (cont.): $C_1 = 26\rho^2(L + \mu D)(1 + C_W)$;

for $t = 1, 2, ..., T$ **do**

  Construct $\hat{g}_t$ by window tree mechanism $\mathcal{WTM}$:
  OPTION I: $\mathcal{WTM}_{Gamma}(\nabla f_t(\hat{x}_t), W, \epsilon)$ for window ϵ-DP;
  OPTION II: $\mathcal{WTM}_{Gaussian}(\nabla f_t(\hat{x}_t), W, \epsilon, \delta)$ for window (ϵ, δ)-DP;

  $\eta_t = \frac{C_1^2}{\mu(t+C_0)}$; $r_t = \sqrt{\frac{4\eta_t}{\mu(t+C_0)}} + \frac{2(L+\mu D)}{\mu(t+C_0)}$;

  $\hat{p}_t = \mathcal{LLOO}(\hat{x}_t, r_t, (\sum_{\tau=1}^{t} \hat{x}_\tau + C_0 x_1 - \hat{g}_t/\mu))$;

  $\hat{x}_{t+1} = (1-\alpha)\hat{x}_t + \alpha\hat{p}_t$

  Output: $\hat{x}_{t+1}$;

end for

Privacy Guarantee: The following theorem guarantees the window differential privacy of the DP projection-free COCO.

Theorem 5: (window differential privacy of Algorithm 2) Under the same condition as in Theorem 1 and 2 (with $\mathcal{C}$ replaced by $\mathcal{P}$), Algorithm 2 with *Option I* and *Option II* is window ϵ-differential privacy and window (ϵ, δ)-differential privacy with window size W correspondingly.

Regret Analysis: We present the regret analysis of Algorithm 2 in the following.

Theorem 6: (regret guarantee with window ϵ-DP) Under the same condition as in Theorem 5, for any $\beta > 0$ and a window size W, with probability at least $1 - \beta$, Algorithm 2 with *OPTION I* has the following regret bound at a particular iteration $T \geq W$,

$$Regret(T) = O((d(L + \mu D)^2 \log_2^{1.5} W \ln(d/\beta) \ln T)/(\mu\epsilon)).$$

Theorem 7: (regret guarantee with window (ϵ, δ)-DP) Under the same condition as in Theorem 5, for any $\beta > 0$ and a window size W, with probability at least $1 - \beta$, Algorithm 2 with *OPTION II* has the following regret bound with the sequence length $T \geq W$, $Regret(T) =$

$$O\Big(\max\big(\sqrt{d}\epsilon, \log_2^{1.5} W \ln \frac{1}{\delta}\sqrt{\ln \frac{1}{\beta}}\big) \cdot \frac{\sqrt{d}(L + \mu D)^2 \ln T}{\mu\epsilon} \Big).$$

Discussion: The regret in this part for both privacy types matches nonprivate optimal $O(\ln T)$ with a fixed window size, confirming that online CG can indeed be privatized and broaden the applicability of private COCO to many applications with LO-friendly constraints. Again, the term $\log_2^{1.5} W$ captures the trade-off between privacy coverage and utility with the regret bound scaling with a *polylog* term of window size W. Considering ϵ-DP, our regret bound is as good as [30] even when privacy window extends to the full sequence. When relaxing to (ϵ, δ)-DP, our method can have better dependence on the dimension d by Theorem 7.

IV. Conclusion

In this paper, we have developed the projection-free COCO under differential privacy restriction. To further improve its utility, we have also investigated the potential to trade privacy of remote input instances. We have adopted a window tree mechanism with Gamma and Gaussian perturbation to efficiently maintain the window private gradient summation. The regret bound has shown that the proposed algorithm can adaptively adjust the trade-off between privacy coverage and utility. Meanwhile, with the fixed window size, the regret bound growth matches the nonprivate optimal $O(\ln T)$. In particular, as the first DP COCO method is designed for problems with linear oracle-friendly constraints, it effectively broadens the applicability of private COCO methods.

Appendix

In the appendix, we recall some additional results from differential privacy. In addition, all proofs of the results in this paper can be found in the Supplement via the link: http://www.comp.hkbu.edu.hk/~ymc/papers/conference/wi20/wi20-supplement.pdf.

A. Gamma mechanism

Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^d$ be an arbitrary d-dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{ajacentx,y} ||f(x) - f(y)||_2$. The *Gamma Mechanism* with the parameter c adds noise $n \in \mathbb{R}^d$ from distribution $e^{-||n||_2/c}$ to the output.

Proposition A1: For any $\epsilon \in (0, 1)$, the Gamma Mechanism with parameter $c = \Delta_2 f/\epsilon$ is ϵ-differentially private.

The following is the concentration property of Gamma distribution ([39]).

Proposition A2: Let X be a random variable drawn from the distribution $\Gamma(k, \theta)$, where k is an integer. We have,

$$P(X \leq k\theta \ln(\frac{k}{\beta})) \geq 1 - \delta. \tag{19}$$

B. Gaussian mechanism

Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^d$ be an arbitrary d-dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{ajacent\ x,y} ||f(x) - f(y)||_2$. The *Gaussian Mechanism* with parameter σ adds noise $n \in \mathbb{R}^d$ with each component scaled to $\mathcal{N}(0, \sigma^2)$ to the output.

Proposition A3: For any $\epsilon \in (0,1)$, if $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2 f/\epsilon$ is (ϵ, δ)-differentially private.

We also need the following concentration property of the Gaussian distribution.

Proposition A4: Let X be a random variable drawn from the distribution $\mathcal{N}(0, \mathbb{I}_d)$. We have,

$$P(||X||_2 \leq \sqrt{2d\ln\frac{1}{\delta}}) \geq 1-\delta. \tag{20}$$

C. Composition theorems

Proposition A5: (simple composition theorem [12]) Let $\mathcal{A}_1$ be ϵ_1-differential privacy and $\mathcal{A}_2$ be ϵ_2-differential privacy. Then, their combination is $(\epsilon_1+\epsilon_2)$-differential privacy.

Proposition A6: (advanced composition theorem [12]) For all $\epsilon, \delta_1, \delta_2 \geq 0$, the class of (ϵ, δ_1)-differentially private mechanisms satisfies $(\epsilon', k\delta_1+\delta_2)$-differential privacy under the k-fold adaptive composition for:

$$\epsilon = \sqrt{2k\ln(1/\delta_2)}\epsilon + k\epsilon(e^\epsilon - 1). \tag{21}$$

It can be seen that the advanced composition theorem roughly provides $(\sqrt{2k\ln(1/\delta_2)}\epsilon), k\delta_1+\delta_2)$-differential privacy.

REFERENCES

[1] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, 2012.

[2] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring web browsing behavior with differential privacy," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014.

[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013.

[4] L. Fan and L. Xiong, "An adaptive approach to real-time aggregate monitoring with differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, 2014.

[5] E. Hazan, "Introduction to online convex optimization," *arXiv preprint arXiv:1909.05207*, 2019.

[6] ——, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[7] N. Liakopoulos, A. Destounis, G. Paschos, T. Spyropoulos, and P. Mertikopoulos, "Cautious regret minimization: Online optimization with long-term budget constraints," in *International Conference on Machine Learning*, 2019.

[8] V. Valls, G. Iosifidis, D. Leith, and L. Tassiulas, "Online convex optimization with perturbed constraints: Optimal rates against stronger benchmarks," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[9] H. Yu and M. J. Neely, "A low complexity algorithm with o (t) regret and o (1) constraint violations for online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 1–24, 2020.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2006.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, 2006.

[12] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, 2014.

[13] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proceedings of the VLDB Endowment*, 2014.

[14] Y. Cao and M. Yoshikawa, "Differentially private real-time data release over infinite trajectory streams," in *Mobile Data Management (MDM), 2015 16th IEEE International Conference on*. IEEE, 2015.

[15] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed predicate sums on streams," in *Proceedings of the 16th International Conference on Database Theory*. ACM, 2013.

[16] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang, "Towards practical differentially private convex optimization," in *IEEE Symposium on Security and Privacy (SP)*, 2019.

[17] R. Bassily, V. Feldman, K. Talwar, and A. G. Thakurta, "Private stochastic convex optimization with optimal rates," in *Advances in Neural Information Processing Systems*, 2019.

[18] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Advances in Neural Information Processing Systems*, 2017.

[19] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning." in *Conference on Learning Theory*, 2012.

[20] J. Lou and Y.-m. Cheung, "Uplink communication efficient differentially private sparse optimization with feature-wise distributed data," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] ——, "An uplink communication-efficient approach to featurewise distributed sparse optimization with differential privacy," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[22] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, 2014.

[23] M. Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[24] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[25] J. Lou and Y.-M. Cheung, "Robust low-rank tensor minimization via a new tensor spectral k-support norm," *IEEE Transactions on Image Processing*, vol. 29, pp. 2314–2327, 2019.

[26] Y.-m. Cheung and J. Lou, "Scalable spectral k-support norm regularization for robust low rank subspace learning," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 1151–1160.

[27] K. Talwar, A. Thakurta, and L. Zhang, "Nearly optimal private lasso," in *Advances in Neural Information Processing Systems*, 2015.

[28] D. Wang and J. Xu, "Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1182–1189.

[29] D. Wang, C. Chen, and J. Xu, "Differentially private empirical risk minimization with non-convex loss functions," in *International Conference on Machine Learning*, 2019, pp. 6526–6535.

[30] A. G. Thakurta and A. Smith, "(nearly) optimal algorithms for private online learning in full-information and bandit settings," in *Advances in Neural Information Processing Systems*, 2013, pp. 2733–2741.

[31] P. Jain and A. G. Thakurta, "(near) dimension independent risk bounds for differentially private learning," in *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[32] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, p. 26, 2011.

[33] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*, 2010.

[34] Y. Yu, X. Zhang, and D. Schuurmans, "Generalized conditional gradient for sparse estimation," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5279–5324, 2017.

[35] A. Mokhtari, H. Hassani, and A. Karbasi, "Stochastic conditional gradient methods: From convex minimization to submodular maximization," *Journal of Machine Learning Research*, vol. 21, no. 105, pp. 1–49, 2020.

[36] C. Qu, Y. Li, and H. Xu, "Non-convex conditional gradient sliding," in *International Conference on Machine Learning*, 2018, pp. 4208–4217.

[37] E. Hazan and S. Kale, "Projection-free online learning," in *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[38] D. Garber and E. Hazan, "A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization," *SIAM Journal on Optimization*, 2016.

[39] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *The Journal of Machine Learning Research*, 2011.