Near-Optimal Algorithms for Private Online Optimization in the Realizable Regime

Hilal Asi [1] **Vitaly Feldman** [1] **Tomer Koren** [2] **Kunal Talwar** [1]

Abstract

We consider online learning problems in the realizable setting, where there is a zero-loss solution, and propose new Differentially Private (DP) algorithms that obtain near-optimal regret bounds. For the problem of online prediction from experts, we design new algorithms that obtain near-optimal regret $\widetilde{O}\big(\varepsilon^{-1}\log^{1.5} d\big)$ where d is the number of experts. This significantly improves over the best existing regret bounds for the DP non-realizable setting which are $\widetilde{O}\big(\varepsilon^{-1}\min\big\{d, T^{1/3}\log d\big\}\big)$. We also develop an adaptive algorithm for the small-loss setting with regret $O(L^\star \log d + \varepsilon^{-1}\log^{1.5} d)$ where $L^\star$ is the total loss of the best expert. Additionally, we consider DP online convex optimization in the realizable setting and propose an algorithm with near-optimal regret $\widetilde{O}\big(\varepsilon^{-1}d^{1.5}\big)$, as well as an algorithm for the smooth case with regret $\widetilde{O}\big(\varepsilon^{-2/3}(dT)^{1/3}\big)$, both significantly improving over existing bounds in the non-realizable regime.

1. Introduction

We study the problem of private online optimization in the realizable setting where there is a zero-loss solution. In this problem, an online algorithm $\mathcal{A}$ interacts with an adversary over T rounds. The adversary picks a (non-negative) loss function $\ell_t : \mathcal{X} \to \mathbb{R}$ at round t and simultaneously the algorithm $\mathcal{A}$ picks a response x_t, suffering loss $\ell_t(x_t)$. The algorithm aims to minimize the regret, which is the loss compared to the best solution $x^\star \in \mathcal{X}$ in hindsight, while at the same time keeping the sequence of predictions $x_1, \ldots, x_T$ differentially private with respect to individual loss functions.

In this paper, we focus on two well-studied instances of this problem. In differentially private online prediction from experts (DP-OPE), we have d experts $\mathcal{X} = [d]$ and the adversary chooses a loss function $\ell_t : [d] \to [0,1]$. Our second setting is differentially private online convex optimization (DP-OCO) where $\mathcal{X} \subset \mathbb{R}^d$ is a convex set with bounded diameter, and the adversary chooses convex and L-Lipschitz loss functions $\ell_t : \mathcal{X} \to \mathbb{R}^+$.

Several papers have recently studied DP-OPE and DP-OCO in the general non-realizable setting (Jain et al., 2012; Smith & Thakurta, 2013; Jain & Thakurta, 2014; Agarwal & Singh, 2017; Kairouz et al., 2021). These papers have resulted in different algorithms with sub-linear regret for both problems. For DP-OPE, Agarwal & Singh (2017); Jain & Thakurta (2014) developed private versions of follow-the-regularized-leader (FTRL) obtaining regret $\min\big\{d/\varepsilon, \sqrt{T\log d}/\varepsilon\big\}$. More recently, Asi et al. (2022b) developed low-switching algorithms for DP-OPE with oblivious adversaries, obtaining regret roughly $O(\sqrt{T\log(d)} + T^{1/3}\log d/\varepsilon)$. Additionally, for the problem of DP-OCO, Kairouz et al. (2021) have recently proposed a DP-FTRL algorithm based on the binary tree mechanism which obtains regret $\big(T\sqrt{d}/\varepsilon\big)^{1/2}$.

Despite this progress, the regret bounds of existing algorithms are still polynomially worse than existing lower bounds. Currently, the only existing lower bounds for oblivious adversaries are the trivial bounds from the non-online versions of the same problems: for DP-OPE, lower bounds for private selection (Steinke & Ullman, 2017) imply a regret lower bound of $O(\log(d)/\varepsilon)$, while existing lower bounds for DP-SCO (Feldman et al., 2020) give a regret lower bound of $\Omega(\sqrt{d}/\varepsilon)$ for DP-OCO.

Practical optimization problems arising from over-parameterized models often lead to instances that additionally satisfy *realizability*, i.e. that the optimal loss is zero or close to zero. This motivates the study of designing algorithms that can do better under this assumption. Realizability has been studied since the early days of learning theory and ubiquitous in the non-private online optimization literature (Srebro et al., 2010; Shalev-Shwartz, 2012; Hazan, 2016). It has proven useful for improving regret bounds in non-private OPE and OCO (Shalev-Shwartz, 2012; Srebro et al., 2010) and in the closely related problem of differen-

[1]Apple [2]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel. Correspondence to: Hilal Asi <hilal.asi94@gmail.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

tially private stochastic convex optimization (DP-SCO) (Asi et al., 2022a). In this work we study DP-OPE and DP-OCO in the realizable setting and develop new algorithms that obtain near-optimal regret bounds in several settings.

1.1. Contributions

We propose new algorithms and lower bounds for the problems of differentially private online prediction from experts (DP-OPE) and differentially private online convex optimization (DP-OCO) in the realizable setting. The following are our primary contributions:

- **Near-optimal algorithms for DP-OPE.** We design new algorithms that obtain near-optimal regret $\widetilde{O}\left(\log^{1.5}(d)/\varepsilon\right)$ for DP-OPE with d experts when there is a zero-loss expert. The best existing algorithms for non-realizable DP-OPE obtain significantly worse regret bounds $\min\left\{d/\varepsilon, T^{1/3}\log d/\varepsilon\right\}$ (Agarwal & Singh, 2017; Asi et al., 2022b), which have a polynomial dependence on either T or the number of experts d. Our algorithms build on sequential applications of the exponential mechanism to pick a good expert, and the sparse-vector-technique to identify when the current expert is no longer a good expert (with near-zero loss). Crucially, an oblivious adversary cannot identify which expert the algorithm has picked, resulting in a small number of switches. We deploy a potential-based proof strategy to show that this algorithm have logarithmic number of switches. We also show that a lower bound of $\Omega(\log d/\varepsilon)$ holds for any ε-DP algorithm even in the realizable case.

- **Adaptive algorithms for DP-OPE with low-loss experts.** We also develop an algorithm that adapts to the setting where there is an expert with low loss, that is, $L^\star = \min_{x\in[d]}\sum_{t=1}^{T}\ell_t(x)$. Our algorithms are adaptive to the value of $L^\star$ and obtain total regret of $L^\star\log d + \varepsilon^{-1}\log^{1.5} d$.

- **Near-optimal regret for low-dimensional DP-OCO.** Building on our algorithms for DP-OPE, we propose a new algorithm for DP-OCO that obtains regret $\widetilde{O}\left(d^{1.5}/\varepsilon\right)$. This is near-optimal for low-dimensional problems where $d = O(1)$ and improves over the best existing algorithm which obtains a regret $(T\sqrt{d}/\varepsilon)^{1/2}$ (Kairouz et al., 2021).

- **Improved regret for smooth DP-OCO.** When the loss function is smooth, we show that DP-FTRL (Kairouz et al., 2021) with certain parameters obtains an improved regret of $(\sqrt{Td}/\varepsilon)^{2/3}$ if there is a zero-loss expert.

1.2. Related work

Several works have studied online optimization in the realizable setting, developing algorithms with better regret bounds (Shalev-Shwartz, 2012; Srebro et al., 2010). For online prediction from experts, the weighted majority algorithm obtains a regret bound of $4\log d$ compared to $O(\sqrt{T\log d})$ in the non-realizable setting. Moreover, for online convex optimization, Srebro et al. (2010) show that online mirror descent achieves regret $4\beta D^2 + 2\sqrt{\beta D^2 T L^\star}$ compared to $O(\sqrt{T})$ in the general case.

On the other hand, the private online optimization literature has mainly studied the general non-realizable case (Jain et al., 2012; Smith & Thakurta, 2013; Jain & Thakurta, 2014; Agarwal & Singh, 2017; Kairouz et al., 2021). For online prediction from experts, the best existing regret bounds for (ε,δ)-DP are $O(\varepsilon^{-1}\sqrt{T\log d\log(1/\delta)})$ (Jain & Thakurta, 2014) and $O(\sqrt{T\log d} + \varepsilon^{-1}\sqrt{d\log(1/\delta)}\log d\log^2 T)$ (Agarwal & Singh, 2017). Asi et al. (2022b) show that these rates can be improved using a private version of the shrinking dartboard algorithm, obtaining regret roughly $O(\sqrt{T\log d} + T^{1/3}\log d/\varepsilon)$. For online convex optimization, Kairouz et al. (2021) developed a private follow-the-regularized-leader algorithm using the binary tree mechanism that obtains regret bound $\widetilde{O}(T\sqrt{d}/\varepsilon)^{1/2}$.

The realizable setting has recently been studied in the different but related problem of differentially private stochastic convex optimization (DP-SCO) (Asi et al., 2022a). DP-SCO and DP-OCO are closely related as one can convert an OCO algorithm into an SCO algorithm using standard online-to-batch transformations (Hazan, 2016) Asi et al. (2022a) study DP-SCO problems in the interpolation regime where there exists a minimizer that minimizes all loss functions, and propose algorithms that improve the regret over the general setting if the functions satisfy certain growth conditions.

2. Preliminaries

In online optimization, we have an interactive T-round game between an adversary and an online algorithm. In this paper, we focus on oblivious adversaries that choose in advance a sequence of loss functions $\ell_1,\dots,\ell_T$ where $\ell_t : \mathcal{X}\to\mathbb{R}$. Then, at round t, the adversary releases a loss function ℓ_t and simultaneously the algorithm plays a solution $x_t\in\mathcal{X}$. The algorithm then suffers loss $\ell_t(x_t)$ at this round. The regret of the online algorithm is

$$Reg_T(\mathcal{A}) = \sum_{t=1}^{T}\ell_t(x_t) - \min_{x^\star\in\mathcal{X}}\sum_{t=1}^{T}\ell_t(x^\star).$$

For ease of notation, for an oblivious adversary that chooses a loss sequence $\mathcal{S} = (\ell_1,\dots,\ell_T)$, we let $\mathcal{A}(\mathcal{S}) =$

	Non-realizable	Realizable (This work)
DP-OPE	$\min\left\{\frac{\sqrt{d}}{\varepsilon}, \sqrt{T\log d} + \frac{T^{1/3}\log d}{\varepsilon}\right\}$ (Agarwal & Singh, 2017; Asi et al., 2022b)	$\frac{\log^{1.5} d}{\varepsilon}$
DP-OCO	$\left(\frac{T\sqrt{d}}{\varepsilon}\right)^{1/2}$ (Kairouz et al., 2021)	$\frac{d^{1.5}}{\varepsilon}$
DP-OCO (smooth)	$\left(\frac{T\sqrt{d}}{\varepsilon}\right)^{1/2}$ (Kairouz et al., 2021)	$\left(\frac{\sqrt{Td}}{\varepsilon}\right)^{2/3}$

Table 1. Comparison between regret upper bounds for the realizable and non-realizable case for both DP-OPE and DP-OCO. For readability, we omit logarithmic factors in T and $1/\delta$.

$(x_1, \ldots, x_T)$ denote the output of the interaction between the online algorithm and the adversary.

In this work, we are mainly interested in two instances of the above general online optimization problem:

- **Online prediction from experts (OPE).** In this problem, we have a set of d experts $\mathcal{X} = [d]$, and the adversary chooses loss functions $\ell_t : [d] \to [0, 1]$.
- **Online convex optimization (OCO).** In OCO, we are optimizing over a convex set $\mathcal{X} \subseteq \mathbb{R}^d$ with bounded diameter $\mathsf{diam}(\mathcal{X}) \le D$,[1] and the adversary chooses loss functions $\ell_t : \mathcal{X} \to \mathbb{R}$ that are convex and L-Lipschitz.

We are mainly interested in the so-called realizable setting. More precisely, we say than an OPE (or OCO) problem is *realizable* if there exists a feasible solution $x^\star \in \mathcal{X}$ such that $L^\star = \sum_{t=1}^T \ell_t(x^\star) = 0$. We also extend some of our results to the near-realizable setting where $0 < L^\star \ll T$.

The main goal of this paper is to study both of these problems under the restriction of differential privacy.

Definition 2.1 (Differential Privacy)**.** A randomized algorithm $\mathcal{A}$ is (ε, δ)-*differentially private* against oblivious adversaries ((ε, δ)-DP) if, for all sequences $\mathcal{S} = (\ell_1, \ldots, \ell_T)$ and $\mathcal{S}' = (\ell'_1, \ldots, \ell'_T)$ that differ in a single element, and for all events O in the output space of $\mathcal{A}$, we have

$$\Pr[\mathcal{A}(\mathcal{S}) \in O] \le e^\varepsilon \Pr[\mathcal{A}(\mathcal{S}') \in O] + \delta.$$

We note that our algorithms satisfy a stronger privacy guarantee against adaptive adversaries (see for example the privacy definition in (Jain et al., 2021)). However, we choose to focus solely on oblivious adversaries for ease of presentation and readability.

[1]The diameter of a set $\mathcal{X} \subseteq \mathbb{R}^d$ (in Euclidean geometry) is defined as $\mathsf{diam}(\mathcal{X}) = \sup_{x,y\in\mathcal{X}} \|x - y\|$.

2.1. Background on Differential Privacy

In our analysis, we require the following standard privacy composition results.

Lemma 2.1 (Basic composition Dwork & Roth, 2014)**.** *If $\mathcal{A}_1, \ldots, A_k$ are randomized algorithms that each is ε-DP, then their composition $(\mathcal{A}_1(\mathcal{S}), \ldots, A_k(\mathcal{S}))$ is $k\varepsilon$-DP.*

Lemma 2.2 (Advanced composition Dwork & Roth, 2014)**.** *If $\mathcal{A}_1, \ldots, A_k$ are randomized algorithms that each is (ε, δ)-DP, then their composition $(\mathcal{A}_1(\mathcal{S}), \ldots, A_k(\mathcal{S}))$ is $(\sqrt{2k\log(1/\delta')}\varepsilon + k\varepsilon(e^\varepsilon - 1), \delta' + k\delta)$-DP.*

In addition to basic facts about differential privacy such as composition and post-processing, our development uses two key techniques from the privacy literature: the Sparse-vector-technique and the binary tree mechanism, which we now describe.

Sparse vector technique. We recall the sparse-vector-technique (Dwork & Roth, 2014) which we use for the realizable setting in Section 3. Given an input $\mathcal{S} = (z_1, \ldots, z_n) \in \mathcal{Z}^n$, the algorithm takes a stream of queries $q_1, q_2, \ldots, q_T$ in an online manner. We assume that each q_i is 1-sensitive, that is, $|q_i(\mathcal{S}) - q_i(\mathcal{S}')| \le 1$ for neighboring datasets $\mathcal{S}, \mathcal{S}' \in \mathcal{Z}^n$ that differ in a single element. We have the following guarantee.

Lemma 2.3 (Dwork & Roth, 2014, Theorem 3.24)**.** *Let $\mathcal{S} = (z_1, \ldots, z_n) \in \mathcal{Z}^n$. For a threshold L and $\beta > 0$, there is an ε-DP algorithm (*AboveThreshold*) that halts at time $k \in [T + 1]$ such that for $\alpha = \frac{8(\log T + \log(2/\beta))}{\varepsilon}$ with probability at least $1 - \beta$,*

- *For all $t < k$, $q_i(\mathcal{S}) \le L + \alpha$;*

- $q_k(\mathcal{S}) \geq L - \alpha$ *or* $k = T + 1$.

To facilitate the notation for using AboveThreshold in our algorithms, we assume that it has the following components:

1. InitializeSparseVec(ε, L, β): initializes a new instance of AboveThreshold with privacy parameter ε, threshold L, and probability parameter β. This returns an instance (data structure) Q that supports the following two functions.
2. Q.AddQuery(q): adds a new query $q : \mathcal{Z}^n \to \mathbb{R}$ to Q.
3. Q.TestAboThr(): tests if the last query that was added to Q was above threshold. In that case, the algorithm stops and does not accept more queries.

The binary tree mechanism. We also build on the binary tree mechanism (Dwork et al., 2010; Chan et al., 2011) which allows to privately estimate the running sum of a sequence of T numbers $a_1, \ldots, a_T \in [0, 1]$.

Lemma 2.4 (Dwork et al., 2010, Theorem 4.1)**.** *Let $\varepsilon \leq 1$. There is an ε-DP algorithm (BinaryTree) that takes a stream of numbers $a_1, a_2, \ldots, a_T$ and outputs $c_1, c_2, \ldots, c_T$ such that for all $t \in [T]$ with probability at least $1 - \beta$,*

$$\left| c_t - \sum_{i=1}^{t} a_i \right| = \frac{1}{\varepsilon} \cdot \mathsf{poly}(\log(\beta^{-1}) \log T).$$

The same approach extends to the case when a_i's are vectors in $\mathbb{R}^d$ with $\|a_i\|_2 \leq 1$. In this case, the error vector $(c_t - \sum_{i=1}^{t} a_i)$ is distributed at $\mathcal{N}(0, d \cdot \mathsf{poly}(\log T/\beta\delta)/\varepsilon^2 \mathbb{I})$ and the mechanism satisfies (ε, δ)-DP.

Additional notation. For a positive integer $k \in \mathbb{N}$, we let $[k] = \{1, 2, \ldots, k\}$. Moreover, for a sequence $a_1, \ldots, a_t$, we use the shorthand $a_{1:t} = a_1, \ldots, a_t$.

3. Near-optimal regret for online prediction from experts

In this section, we consider the online prediction from experts problem in the near-realizable regime, where the best expert achieves small loss $L^\star \ll T$. Under this setting, we develop a new private algorithm that achieves regret $\widetilde{O}(L^\star \log d + \log^{3/2}(d)/\varepsilon)$. For the realizable setting where $L^\star = 0$, this algorithm obtains near-optimal regret $\widetilde{O}(\log^{3/2}(d)/\varepsilon)$.

The algorithm builds on the fact that an oblivious adversary cannot know which expert the algorithm picks. Therefore, if the algorithm picks a random good expert with loss smaller than $L^\star$, the adversary has to increase the loss for many experts before identifying the expert chosen by the algorithm. The algorithm will therefore proceed as follows: at each round, privately check using sparse-vector-technique whether the previous expert is still a good expert (has loss nearly $L^\star$). If not, randomly pick (privately) a new expert from the set of remaining good experts. The full details are in Algorithm 1.

The following theorem summarizes the performance of Algorithm 1.

Theorem 1. *Let $\ell_1, \ldots, \ell_T \in [0, 1]^d$ be chosen by an oblivious adversary such that there is $x^\star \in [d]$ such that $\sum_{t=1}^{T} \ell_t(x^\star) \leq L^\star$. Let $0 < \beta < 1/2$, $B = \log(2T^2/\beta)$, and $L = L^\star + 4/\eta + \frac{8B}{\varepsilon}$. If $\eta = \varepsilon/(12\lceil \log d \rceil + 48 \log(1/\beta))$ then Algorithm 1 is ε-DP and with probability at least $1 - O(\beta)$ has regret*

$$\sum_{t=1}^{T} \ell_t(x_t) \leq O\left(L^\star \log(d/\beta)\right) + O\left(\frac{\log^2(d) + \log(T/\beta)\log(d/\beta)}{\varepsilon}\right).$$

Further, if $\varepsilon \leq \sqrt{\log T \log(1/\delta)}$ and $\eta = \varepsilon/\sqrt{32(6\lceil \log d \rceil + 24\log(1/\beta))\log(1/\delta)}$ then Algorithm 1 is (ε, δ)-DP and with probability at least $1 - O(\beta)$ has regret

$$\sum_{t=1}^{T} \ell_t(x_t) \leq O\left(L^\star \log(d/\beta)\right) + O\left(\frac{\log^{3/2}(d)\sqrt{\log(1/\delta)} + \log(T/\beta)\log(d/\beta)}{\varepsilon}\right).$$

While Algorithm 1 requires the knowledge of $L^\star$, we also design an adaptive version that does not require $L^\star$ in the next section. Note that the algorithm obtains regret roughly $\log^{3/2}(d)/\varepsilon$ for the realizable setting where $L^\star = 0$.

Proof. First, we prove the privacy guarantees of the algorithm using privacy composition results: there are K applications of the exponential mechanism with privacy parameter η. Moreover, sparse-vector is applied over each user's data only once, hence the K applications of sparse-vector are $\varepsilon/2$-DP. Overall, the algorithm is $(\varepsilon/2 + K\eta)$-DP and $(\varepsilon/2 + \sqrt{2K\log(1/\delta)}\eta + K\eta(e^\eta - 1), \delta)$-DP (using advanced compositions; see Lemma 2.2). Setting $\eta = \varepsilon/2K$ results in ε-DP and $\eta = O(\varepsilon/\sqrt{K\log(1/\delta)})$ results in (ε, δ)-DP.

We proceed to analyze utility. First, note that the guarantees of the sparse-vector algorithm (Lemma 2.3) imply that with probability at least $1 - \beta$ for each time-step $t \in [T]$, if sparse-vector identifies above threshold query then $s_t(x) \geq \underline{\Delta} := L - \frac{8B}{\varepsilon} \geq 4/\eta$. Otherwise, $s_t(x) \leq \bar{\Delta} := L + \frac{8B}{\varepsilon}$. In the remainder of the proof, we condition on this event. The

Algorithm 1 Sparse-Vector for zero loss experts

Require: Switching bound K, optimal loss $L^\star$, Sampling parameter η, Threshold parameter L, failure probability β, privacy parameters (ε, δ)

Set $K = 6\lceil \log d \rceil + 24\log(1/\beta)$, $k = 0$, and current expert $x_0 = \mathsf{Unif}[d]$
Set $t_p = 0$
while $t \le T$ **do**
  Set $x_t = x_{t-1}$
  if $k < K$ **then**
    $Q = \mathsf{InitializeSparseVec}(\varepsilon/2, L, \beta/T)$
    while $Q.\mathsf{TestAboThr}() = \text{False}$ **do**
      Set $x_t = x_{t-1}$
      Define a new query $q_t = \sum_{i=t_p}^{t-1} \ell_i(x_t)$
      Add new query $Q.\mathsf{AddQuery}(q_t)$
      Receive loss function $\ell_t : [d] \to [0,1]$
      Pay cost $\ell_t(x_t)$
      Update $t = t + 1$
    Sample x_t from the exponential mechanism with scores $s_t(x) = \max\left(\sum_{i=1}^{t-1} \ell_i(x), L^\star\right)$ for $x \in [d]$:

$$\mathbb{P}(x_t = x) \propto e^{-\eta s_t(x)/2}$$

    Set $k = k + 1$ and $t_p = t$
  Receive loss function $\ell_t : [d] \to [0,1]$
  Pay cost $\ell_t(x_t)$
  Update $t = t + 1$

idea is to show that the algorithm has logarithmic number of switches, and each switch the algorithm pays roughly $1/\varepsilon$ regret.

To this end, we define a potential at time $t \in [T]$:

$$\phi_t = \sum_{x \in [d]} e^{-\eta L_t(x)/2},$$

where $L_t(x) = \max(\sum_{j=1}^{t-1} \ell_j(x), L^\star)$. Note that $\phi_1 = de^{-\eta L^\star/2}$ and $\phi_t \ge e^{-\eta L^\star/2}$ for all $t \in [T]$ as there is $x \in [d]$ such that $\sum_{t=1}^{T} \ell_t(x) = L^\star$. We split the iterates to $m = \lceil \log d \rceil$ rounds $t_0, t_1, \ldots, t_m$ where t_i is the largest $t \in [T]$ such that $\phi_{t_i} \ge \phi_1/2^i$. Let Z_i be the number of switches in $[t_i, t_{i+1} - 1]$ (number of times the exponential mechanism is used to pick x_t). The following key lemma shows that Z_i cannot be too large.

Lemma 3.1. *Fix $0 \le i \le m-1$. Then for any $1 \le k \le T$, it holds that*

$$P(Z_i = k+1) \le (2/3)^k.$$

Proof. Let $t_i \le t \le t_{i+1}$ be a time-step where a switch happens (exponential mechanism is used to pick x_t). Note that $\phi_{t_{i+1}} \ge \phi_t/2$. We prove that the probability that x_t is switched between t and t_{i+1} is at most $2/3$. To this end, note that if x_t is switched before t_{i+1} then $\sum_{i=t}^{t_{i+1}} \ell_i(x) \ge \underline{\Delta}$ as sparse-vector identifies x_t, and therefore $L_{t_{i+1}}(x) - L_t(x) \ge \underline{\Delta} - L^\star \ge 4/\eta$. Thus we have that

$$\begin{aligned}
&P(x_t \text{ is switched before } t_{i+1}) \\
&\le \sum_{x \in [d]} P(x_t = x) 1\{L_{t_{i+1}}(x) - L_t(x) \ge 4/\eta\} \\
&= \sum_{x \in [d]} \frac{e^{-\eta L_t(x)/2}}{\phi_t} \cdot 1\{L_{t_{i+1}}(x) - L_t(x) \ge 4/\eta\} \\
&\le \sum_{x \in [d]} \frac{e^{-\eta L_t(x)/2}}{\phi_t} \cdot \frac{1 - e^{-\eta(L_{t_{i+1}}(x) - L_t(x))/2}}{1 - e^{-2}} \\
&\le 4/3(1 - \phi_{t_{i+1}}/\phi_t) \\
&\le 2/3.
\end{aligned}$$

where the second inequality follows the fact that $1\{a \ge b\} \le \frac{1-e^{-\eta b}}{1-e^{-\eta a}}$ for $a, b, \eta \ge 0$, and the last inequality since $\phi_{t_{i+1}}/\phi_{t_1} \ge 1/2$. This argument shows that after the first switch inside the range $[t_i, t_{i+1}]$, each additional switch happens with probability at most $2/3$. The claim follows. □

We now proceed with the proof. Let $Z = \sum_{i=0}^{m-1} Z_i$ be the total number of switches. Note that $Z \le m + \sum_{i=0}^{m-1} \max(Z_i - 1, 0)$ and Lemma 3.1 implies $\max(Z_i - 1, 0)$ is upper bounded by a geometric random variable with success probability $1/3$. Therefore, using concentration of geometric random variables (Lemma A.2), we get that

$$P(Z \ge 6m + 24\log(1/\beta)) \le \beta.$$

Noting that $K \ge 6m + 24\log(1/\beta)$, this shows that the algorithm does not reach the switching budget with probability $1 - O(\beta)$. Thus, the guarantees of the sparse-vector algorithm imply that the algorithm pays regret at most $\bar{\Delta}$ for each switch, hence the total regret of the algorithm is at most $O(\bar{\Delta}(m + \log(1/\beta))) = O(\bar{\Delta}\log(d/\beta))$. The claim follows as $\bar{\Delta} \le L^\star + 4/\eta + 16B/\varepsilon$. □

3.1. Adaptive algorithms for DP experts

While Algorithm 1 achieves near-optimal loss for settings with low-loss experts, it requires the knowledge of the value of $L^\star$. As $L^\star$ is not always available in practice, our goal in this section is to develop an adaptive version of Algorithm 1 which obtains similar regret without requiring the knowledge of $L^\star$. Similarly to other online learning problems, we propose to use the doubling trick (Kalai & Vempala, 2005) to design our adaptive algorithms. We begin with an estimate $L_1^\star = 1$ of $L^\star$. Then we apply Algorithm 1 using $L^\star = L_1^\star$ until the exponential mechanism picks an expert that contradicts the current estimate of $L^\star$, that is, $\sum_{i=1}^{t-1} \ell_i(x_t) \gg L_1^\star$. We use the Laplace mechanism to check

Algorithm 2 Adaptive Sparse-Vector for low-loss experts

Require: Failure probability β

1: Set $\varepsilon_0 = \varepsilon/2\log T$, $\beta_0 = \beta/T$
2: $K = 6\lceil \log d \rceil + 24\log(1/\beta_0)$, $\eta = \varepsilon_0/2K$, $B = \log(2T^2/\beta_0)$
3: Set $\bar{L}^\star = 1$, $L = L^\star + 4/\eta + \frac{8B}{\varepsilon_0}$
4: **while** $t < T$ **do**
5: Run Algorithm 1 with parameters $\bar{L}^\star$, η, L, β_0, ε_0
6: **if** Algorithm 1 applies the exponential mechanism (step 12) **then**
7: Calculate $\bar{L}_t = \sum_{i=1}^{t-1} \ell_i(x_t) + \zeta_t$ where $\zeta_t \sim \mathsf{Laplace}(K/\varepsilon_0)$
8: **if** $\bar{L}_t > \bar{L}^\star - 5K\log(1/\beta_0)/\varepsilon_0$ **then**
9: Set $\bar{L}^\star = 2\bar{L}^\star$
10: Go to step 4

this privately. Noting that this happens with small probability if $L^\star \le L_1^\star$, we conclude that our estimate of $L^\star$ was too small and set a new estimate $L_2^\star = 2L_1^\star$ and repeat the same steps. As $L^\star \le T$, this process will stop in at most $\log T$ phases, hence we can divide the privacy budget equally among phases while losing at most a factor of $\log T$. We present the full details in Algorithm 2.

We have the following guarantees for the adaptive algorithm. We defer the proof to Appendix B.2.

Theorem 2. *Let $\ell_1, \dots, \ell_T \in [0,1]^d$ be chosen by an oblivious adversary such that there is $x^\star \in [d]$ such that $\sum_{t=1}^T \ell_t(x^\star) \le L^\star$. Let $0 < \beta < 1/2$. Then Algorithm 2 is ε-DP and with probability at least $1 - O(\beta)$ has regret*

$$\sum_{t=1}^T \ell_t(x_t) \le O\left(L^\star \log(Td/\beta)\log(T)\right) + O\left(\frac{\log^2(d)\log(T) + \log(T/\beta)\log(Td/\beta)\log(T)}{\varepsilon}\right).$$

We also present a different binary-tree based mechanism for this problem with similar rates in Appendix B.1.

4. Faster rates for DP-OCO

In this section we study differentially private online convex optimization (DP-OCO) and propose new algorithms with faster rates in the realizable setting. In Section 4.1, we develop an algorithm that reduces the OCO problem to an experts problem (by discretizing the space) and then uses our procedure for experts. In Section 4.2, we show that follow-the-regularized-leader (FTRL) using the binary tree mechanism results in faster rates in the realizable setting for smooth functions.

4.1. Experts-based algorithm for DP-OCO

The algorithm in this section essentially reduces the problem of DP-OCO to DP-OPE by discretizing the space $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \le D\}$ into sufficiently many experts. In particular, we consider a ρ-net of the space $\mathcal{X}$, that is, a set $\mathcal{X}_{\mathsf{experts}} = \{x^1, \dots, x^M\} \subset \mathcal{X}$ such that for all $x \in \mathcal{X}$ there is $x^i \in \mathcal{X}^\rho_{\mathsf{experts}}$ such that $\left\|x^i - x\right\|_2 \le \rho$. Such a set exists if $M \ge 2^{d\log(4D/\rho)}$ (Duchi, 2019, Lemma 7.6). Given a loss function $\ell_t : \mathcal{X} \to \mathbb{R}$, we define the loss of expert x^i to be $\ell_t(x^i)$. Then, we run Algorithm 1 for the given DP-OPE problem. This algorithm has the following guarantees.

Theorem 3. *Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \le D\}$ and $\ell_1, \dots, \ell_T : \mathcal{X} \to \mathbb{R}$ be non-negative, convex and L-Lipschitz functions chosen by an oblivious adversary. Then running Algorithm 1 over $\mathcal{X}^\rho_{\mathsf{experts}}$ with $\rho = 1/(LT)$ is (ε, δ)-DP and with probability at least $1 - O(\beta)$ has regret*

$$\mathbb{E}\left[\sum_{t=1}^T \ell_t(x_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^T \ell_t(x)\right] \le (L^\star + \frac{1}{\varepsilon}) d^{1.5} \cdot O\left(\mathsf{poly}(\log(DLT/\delta))\right).$$

We defer the proof to Appendix D.1.

These results demonstrates that existing algorithms which achieve regret roughly $(\frac{T\sqrt{d}}{\varepsilon})^{1/2}$ are not optimal for the realizable setting. Moreover, in the low-dimensional regime (constant d), the above bound is nearly-optimal up to logarithmic factors as we have a lower bound of $\sqrt{d}/\varepsilon$ from the stochastic setting of this problem (see discussion in the introduction).

Finally, while the algorithm we presented in Theorem 3 has exponential runtime due to discretizing the space, we note that applying Algorithm 1 over the unit ball results in similar rates and polynomial runtime. Recall that this algorithm only accesses the loss functions to sample from the exponential mechanism, and uses sparse-vector over the running loss. Both of these can be implemented in polynomial time—since the losses are convex—using standard techniques from log-concave sampling.

4.2. Binary-tree based FTRL

In this section, we consider DP-OCO with smooth loss functions and show that DP-FTRL (Algorithm 1, Kairouz et al., 2021) with modified parameters obtains improved regret $\beta D^2 + (\sqrt{Td}/\varepsilon)^{2/3}$ in the realizable setting, compared to $LD\sqrt{T} + (T\sqrt{d}/\varepsilon)^{1/2}$ in the non-realizable setting.

We present the details in Algorithm 3. Appendix B.1 in (Kairouz et al., 2021) has more detailed information about the implementation of the binary tree mechanism in DP-FTRL.

Algorithm 3 DP-FTRL (Kairouz et al., 2021)

Require: Regularization parameter λ

Set $x_0 \in \mathcal{X}$

for $t = 1$ to T **do**

  Use the binary tree mechanism to estimate the sum $\sum_{i=1}^{t-1} \nabla \ell_i(x_i)$; let $\bar{g}_{t-1}$ be the estimate

  Apply follow-the-regularized-leader step

$$x_t = \operatorname*{argmin}_{x \in \mathcal{X}} \langle \bar{g}_{t-1}, x \rangle + \frac{\lambda}{2} \|x\|_2^2,$$

  Receive loss function $\ell_t : \mathcal{X} \to \mathbb{R}$

  Pay cost $\ell_t(x_t)$

We have the following guarantees for DP-FTRL in the realizable and smooth setting.

Theorem 4. *Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \le D\}$ and $\ell_1, \dots, \ell_T : \mathcal{X} \to \mathbb{R}$ be non-negative, convex, L-Lipschitz, and β-smooth functions chosen by an oblivious adversary. DP-FTRL with $\lambda = 32\beta + \left(\frac{\beta}{\varepsilon^2}(L/D)^2 T d \log(T)\log(1/\delta)\right)^{1/3}$ is (ε, δ)-DP and generates $x_1, \dots, x_T$ that has regret*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^\star)\right] \le O\left(L^\star + \beta D^2 + \left(LD\frac{\sqrt{\beta D^2 T d \log(T)\log(1/\delta)}}{\varepsilon}\right)^{2/3}\right).$$

For the proof, we use the following property for smooth non-negative functions.

Lemma 4.1 (Nesterov, 2004)**.** *Let $\ell : \mathcal{X} \to \mathbb{R}$ be non-negative and β-smooth function. Then $\|\nabla \ell(x)\|_2^2 \le 4\beta \ell(x)$.*

Proof. The proof follows similar arguments to the proof of Theorem 5.1 in (Kairouz et al., 2021). Let

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \sum_{i=1}^{t} \langle \nabla \ell_i(x_i), x \rangle + \frac{\lambda}{2}\|x\|_2^2 + \langle b_t, x \rangle,$$

be the iteration of DP-FTRL where b_t is the noise added by the binary tree mechanism. Moreover, let $\hat{x}_{t+1}$ be the non-private solution, that is,

$$\hat{x}_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \sum_{i=1}^{t} \langle \nabla \ell_i(x_i), x \rangle + \frac{\lambda}{2}\|x\|_2^2.$$

Lemma C.2 in (Kairouz et al., 2021) states that $\|x_{t+1} - \hat{x}_{t+1}\|_2 \le \|b_t\|_2 / \lambda$. Therefore, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^\star) \\
&\le \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), x_t - x^\star \rangle \\
&= \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), x_t - \hat{x}_t \rangle + \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), \hat{x}_t - x^\star \rangle \\
&\le \sum_{t=1}^{T} \|\nabla \ell_t(x_t)\|_2 \|x_t - \hat{x}_t\|_2 + \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), \hat{x}_t - x^\star \rangle \\
&\le \frac{1}{8\beta} \sum_{t=1}^{T} \|\nabla \ell_t(x_t)\|_2^2 + 4\beta \sum_{t=1}^{T} \|x_t - \hat{x}_t\|_2^2 \\
&\quad + \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), \hat{x}_t - x^\star \rangle \\
&\le \frac{1}{2} \sum_{t=1}^{T} \ell_t(x_t) + 4\beta \sum_{t=1}^{T} \|b_t\|_2^2 / \lambda^2 \\
&\quad + \sum_{t=1}^{T} \langle \nabla \ell_t(x_t), \hat{x}_t - x^\star \rangle,
\end{aligned}$$

where the second inequality follows from the Fenchel-Young inequality. We can now upper bound the right term. Indeed, Theorem 5.2 in (Hazan, 2016) implies that FTRL has

$$\begin{aligned}
\sum_{t=1}^{T} \langle \nabla \ell_t(x_t), \hat{x}_t - x^\star \rangle &\le \frac{2}{\lambda} \sum_{t=1}^{T} \|\nabla \ell_t(x_t)\|_2^2 + \lambda D^2 \\
&\le \frac{8\beta}{\lambda} \sum_{t=1}^{T} \ell_t(x_t) + \lambda D^2.
\end{aligned}$$

Overall we now get

$$\begin{aligned}
\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(x^\star) \le \frac{1}{2} \sum_{t=1}^{T} \ell_t(x_t) + \frac{4\beta}{\lambda^2} \sum_{t=1}^{T} \|b_t\|_2^2 \\
+ \frac{8\beta}{\lambda} \sum_{t=1}^{T} \ell_t(x_t) + \lambda D^2.
\end{aligned}$$

The binary tree mechanism also guarantees that for all $t \in [T]$,

$$\mathbb{E}[\|b_t\|_2^2] \le O\left(\frac{L^2 d \log(T)\log(1/\delta)}{\varepsilon^2}\right)$$

(see Appendix B.1 in (Kairouz et al., 2021)). Thus, taking expectation and setting the regularization parameter to $\lambda =$

$32\beta + \left(\frac{\beta}{\varepsilon^2}(L/D)^2 T d \log(T)\log(1/\delta)\right)^{1/3}$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell_t(x_t) - \ell_t(x^\star)\right] \le O(L^\star + \beta D^2) + O\left(\left(\beta D^2 (LD)^2 \frac{Td\log(T)\log(1/\delta)}{\varepsilon^2}\right)^{1/3}\right).$$

□

5. Lower bounds

In this section, we prove lower bounds for private experts in the realizable setting which show that our upper bounds are nearly-optimal up to logarithmic factors. The lower bound demonstrates that a logarithmic dependence on d is necessary even in the realizable setting. Note that for DP-OCO in the realizable setting, a lower bound of d/ε for pure DP follows from known lower bounds for DP-SCO in the interpolation regime (Asi et al., 2022a) using online-to-batch conversions (Hazan, 2016).

The following theorem states our lower bound for DP-OPE.

Theorem 5. *Let $\varepsilon \le 1/10$ and $\delta \le \varepsilon/d$. If $\mathcal{A}$ is (ε,δ)-DP then there is an oblivious adversary such that $\min_{x\in[d]}\sum_{t=1}^{T}\ell_t(x) = 0$ and*

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell_t(x_t) - \min_{x\in[d]}\sum_{t=1}^{T}\ell_t(x)\right] \ge \Omega\left(\frac{\log(d)}{\varepsilon}\right).$$

Proof. Let $\ell^0(x) = 0$ for all x and for $j \in [d]$ let $\ell^j(x)$ be the function that has $\ell^j(x) = 0$ for $x = j$ and otherwise $\ell^j(x) = 1$. The oblivious adversary picks one of the following d sequences uniformly at random: $\mathcal{S}^j = (\underbrace{\ell^0,\ldots,\ell^0}_{T-k},\underbrace{\ell^j,\ldots,\ell^j}_{k})$ where $k = \frac{\log d}{2\varepsilon}$ and $j \in [d]$. Assume towards a contradiction that the algorithm obtains regret $\log(d)/(32\varepsilon)$. This implies that there exists $d/2$ sequences such that the algorithm obtains expected regret $\log(d)/(16\varepsilon)$ where the expectation is only over the randomness of the algorithm. Assume without loss of generality these sequences are $\mathcal{S}^1,\ldots,\mathcal{S}^{d/2}$. Let B_j be the set of outputs that has low regret on $\mathcal{S}^j$, that is,

$$B_j = \{(x_1,\ldots,x_T)\in[d]^T : \sum_{t=1}^{T}\ell^j(x_t) \le \log(d)/(8\varepsilon)\}.$$

Note that $B_j \cap B_{j'} = \emptyset$ since if $x_{1:T} \in B_j$ then at least $3k/4 = 3\log(d)/(8\varepsilon)$ of the last k outputs must be equal to j. Now Markov inequality implies that

$$\mathbb{P}(\mathcal{A}(\mathcal{S}^j)\in B_j) \ge 1/2.$$

Moreover, group privacy gives

$$\begin{aligned}\mathbb{P}(\mathcal{A}(\mathcal{S}^j)\in B_{j'}) &\ge e^{-k\varepsilon}\mathbb{P}(\mathcal{A}(\mathcal{S}^{j'})\in B_{j'}) - ke^{-\varepsilon}\delta \\ &\ge \frac{1}{2\sqrt{d}} - \frac{\log(d)}{2\varepsilon}\delta \\ &\ge \frac{1}{4\sqrt{d}},\end{aligned}$$

where the last inequality follows since $\delta \le \varepsilon/d$. Overall we get that

$$\frac{d/2-1}{4\sqrt{d}} \le \mathbb{P}(\mathcal{A}(\mathcal{S}^j)\notin B_j) \le \frac{1}{2},$$

which is a contradiction for $d \ge 32$. □

6. Conclusion

In this work, we studied differentially private online learning problems in the realizable setting, and developed algorithms with improved rates compared to the non-realizable setting. However, several questions remain open in this domain. First, our near-optimal algorithms for DP-OPE obtain $\log^{1.5}(d)/\varepsilon$ regret, whereas the lower bound we have is $\Omega(\log(d)/\varepsilon)$. Hence, perhaps there are better algorithms with tighter logarithmic factors than our sparse-vector based algorithms. Additionally, for DP-OCO, our algorithms are optimal only for low-dimensional setting, and there remains polynomial gaps in the high-dimensional setting. Finally, optimal rates for both problems (DP-OPE and DP-OCO) are still unknown in the general non-realizable setting.

Acknowledgements

This work has received support from the Israeli Science Foundation (ISF, grant no. 2549/19), Len Blavatnik and the Blavatnik Family foundation, and from the Tel Aviv University Center for AI and Data Science (TAD).

References

Agarwal, N. and Singh, K. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 32–40, 2017.

Asi, H., Chadha, K., Cheng, G., and Duchi, J. Private optimization in the interpolation regime: faster rates and hardness results. In *Proceedings of the 39th International Conference on Machine Learning*, 2022a.

Asi, H., Feldman, V., Koren, T., and Talwar, K. Private online prediction from experts: Separations and faster rates. *arXiv:2210.13537 [cs.LG]*, 2022b.

Chan, T.-H. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

Duchi, J. C. Information theory and statistics. Lecture Notes for Statistics 311/EE 377, Stanford University, 2019. URL http://web.stanford.edu/class/stats311/lecture-notes.pdf. Accessed May 2019.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3 & 4):211–407, 2014.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the Forty-Second Annual ACM Symposium on the Theory of Computing*, pp. 715–724, 2010.

Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM on the Theory of Computing*, pp. 439–449, 2020.

Hazan, E. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3–4):157–325, 2016.

Jain, P. and Thakurta, A. (Near) dimension independent risk bounds for differentially private learning. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 476–484, 2014.

Jain, P., Kothari, P., and Thakurta, A. Differentially private online learning. In *Proceedings of the Twenty Fifth Annual Conference on Computational Learning Theory*, 2012.

Jain, P., Raskhodnikova, S., Sivakumar, S., and Smith, A. The price of differential privacy under continual observation. *arXiv:2112.00828 [cs.DS]*, 2021.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. *arXiv:2103.00039 [cs.CR]*, 2021.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Mitzenmacher, M. and Upfal, E. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

Nesterov, Y. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Smith, A. and Thakurta, A. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems 26*, 2013.

Srebro, N., Sridharan, K., and Tewari, A. Smoothness, low noise and fast rates. In *nips2010*, pp. 2199–2207, 2010.

Steinke, T. and Ullman, J. Tight lower bounds for differentially private selection. In *58th Annual Symposium on Foundations of Computer Science*, pp. 552–563. IEEE, 2017.

A. Concentration for sums of geometric variables

In this section, we proof a concentration result for the sum of geometric random variables, which allows us to upper bound the number of switches in the sparse-vector based algorithm. We say that Z is geometric random variable with success probability p if $P(W = k) = (1-p)^{k-1}p$ for $k \in \{1, 2, \dots\}$. To this end, we use the following Chernoff bound.

Lemma A.1 (Mitzenmacher & Upfal, 2005, Ch. 4.2.1)**.** *Let $X = \sum_{i=1}^{n} X_i$ for $X_i \overset{\text{iid}}{\sim} \mathsf{Ber}(p)$. Then for $\delta \in [0, 1]$,*

$$\mathbb{P}(X > (1+\delta)np) \le e^{-np\delta^2/3} \quad \text{and} \quad \mathbb{P}(X < (1-\delta)np) \le e^{-np\delta^2/2}.$$

The following lemma demonstrates that the sum of geometric random variables concentrates around its mean with high probability.

Lemma A.2. *Let $W_1, \dots, W_n$ be iid geometric random variables with success probability p. Let $W = \sum_{i=1}^{n} W_i$. Then for any $k \ge n$*

$$\mathbb{P}(W > 2k/p) \le \exp(-k/4).$$

Proof. Notice that W is distributed according to the negative binomial distribution where we can think of W as the number of Bernoulli trials until we get n successes. More precisely, let $\{B_i\}$ for $i \ge 1$ be Bernoulli random variables with probability p. Then the event $W > t$ has the same probability as $\sum_{i=1}^{t} B_i < n$. Thus we have that

$$\mathbb{P}(W > t) \le \mathbb{P}(\sum_{i=1}^{t} B_i < n).$$

We can now use Chernoff inequality (Lemma A.1) to get that for $t = 2n/p$:

$$\mathbb{P}(\sum_{i=1}^{t} B_i < n) \le \exp(-tp/8) = \exp(-n/4).$$

This proves that

$$\mathbb{P}(W > 2n/p) \le \exp(-n/4).$$

The claim now follows by noticing that $\sum_{i=1}^{n} W_i \le \sum_{i=1}^{k} W_i$ for W_i iid geometric random variable when $k \ge n$, thus $\mathbb{P}(\sum_{i=1}^{n} W_i \ge 2k/p) \le \mathbb{P}(\sum_{i=1}^{k} W_i \ge 2k/p) \le \exp(-k/4)$

□

B. Additional details for Section 3

B.1. A binary-tree based algorithm

In this section, we present another algorithm which achieves the optimal regret for settings with zero-expert loss. Instead of using sparse-vector, this algorithm builds on the binary tree mechanism. The idea is to repetitively select $O(\mathsf{poly}(\log(dT)))$ random good experts and apply the binary tree to calculate a private version of their aggregate losses. Whenever all of the chosen experts are detected to have non-zero loss, we choose a new set of good experts. Similarly to Algorithm 1, each new phase reduces the number of good experts by a constant factor as an oblivious adversary does not know the choices of the algorithm, hence there are only $O(\mathsf{poly}(\log(dT)))$ phases.

We provide a somewhat informal description of the algorithm in Algorithm 4. This algorithm also achieves regret $O(\mathsf{poly}(\log(dT))/\varepsilon)$ in the realizable case. We do not provide a proof as it is somewhat similar to that of Theorem 1.

B.2. Proof for Theorem 2

First we prove privacy. Note that $\bar{L}^\star$ can change at most $\log(T)$ times as $L^\star \le T$. Therefore, we have at most $\log(T)$ applications of Algorithm 1. Each one of these is $\varepsilon/(2\log(T))$-DP. Moreover, since we have at most K applications of the exponential mechanism in Algorithm 1, we have at most $K\log(T)$ applications of the Laplace mechanism in Algorithm 2. Each of these is $\varepsilon/2K\log(T)$-DP. Overall, privacy composition implies that the final privacy is ε-DP.

Algorithm 4 Binary-tree algorithm for zero loss experts (sketch)

Set $k = 0$ and $B = O(\mathsf{poly}(\log(dT)))$
while $t \le T$ **do**
  Use the exponential mechanism with score function $s(x) = \sum_{i=1}^{t} \ell_i(x)$ to privately select a set S_k of B experts from $[d] \setminus \cup_{0 \le i \le k} S_i$
  Apply binary tree for each expert $x \in S_k$ to get private aggregate estimates for $\sum_{i=1}^{t} \ell_i(x)$ for every $t \in [T]$
  Let $\hat{c}_{t,x}$ denote the output of the binary tree for expert $x \in S_k$ at time t
  while there exists $x \in S_k$ such that $\hat{c}_{t,x} \le O(\mathsf{poly}(\log(dT))/\varepsilon)$ **do**
    Receive $\ell_t : [d] \to [0, 1]$
    Choose $x_t \in S_k$ that minimizes $\hat{c}_{t,x}$
    Pay error $\ell_t(x_t)$
    $t = t + 1$
  $k = k + 1$

Now we prove utility. Algorithm 2 consists of at most $\log(T)$ applications of Algorithm 1 with different values of $\bar{L}^\star$. We will show that each of these applications incurrs low regret. Consider an application of Algorithm 1 with $\bar{L}^\star$. If $\bar{L}^\star \ge L^\star$, then Theorem 1 implies that the regret is at most

$$O\left(\bar{L}^\star \log(d/\beta_0) + \frac{\log^2(d) + \log(T/\beta_0)\log(d/\beta_0)}{\varepsilon_0}\right).$$

Now consider the case where $\bar{L}^\star \le L^\star$. We will show that Algorithm 2 will double $\bar{L}^\star$ and that the regret of Algorithm 1 up to that time-step is not too large. Let t_0 be the largest t such that $\min_{x \in [d]} \sum_{t=1}^{t_0} \ell_t(x) \le \bar{L}^\star$. Note that up to time t_0, the best expert had loss at most $\bar{L}^\star$ hence the regret up to time t_0 is

$$O\left(\bar{L}^\star \log(d/\beta_0) + \frac{\log^2(d) + \log(T/\beta_0)\log(d/\beta_0)}{\varepsilon_0}\right).$$

Now let t_1 denote the next time-step when Algorithm 1 applies the exponential mechanism. Sparse-vector guarantees that in the range $[t_0, t_1]$ the algorithm suffers regret at most $O\left(\bar{L}^\star + \frac{\log(d)+\log(T/\beta_0)}{\varepsilon_0}\right)$. Moreover, the guarantees of the Laplace mechanism imply that at this time-step, $\bar{L}_t \ge \bar{L}^\star - 5K\log(1/\beta_0)/\varepsilon_0$ with probability $1 - \beta_0$, hence Algorithm 2 will double $\bar{L}^\star$ and run a new application of Algorithm 1. Overall, an application of Algorithm 1 with $\bar{L}^\star \le L^\star$ results in regret $(L^\star + \frac{1}{\varepsilon_0}) \cdot \mathsf{poly}(\log \frac{Td}{\beta_0})$ and doubles $\bar{L}^\star$. Finally, note that if $\bar{L}^\star \ge L^\star + 5\log(1/\beta_0)/\varepsilon_0$ then with probability $1 - \beta_0$ the algorithm will not double the value of $\bar{L}^\star$. As each application of Algorithm 1 has regret

$$O\left(\bar{L}^\star \log(d/\beta_0) + \frac{\log^2(d) + \log(T/\beta_0)\log(d/\beta_0)}{\varepsilon_0}\right),$$

and $\bar{L}^\star$ is bounded by $L^\star + 5\log(T/\beta_0)/\varepsilon_0$ with probability $1 - O(\beta_0)$. Overall, the failure probability is $O(T\beta_0) = O(\beta)$.

C. Potential-based algorithm for DP-experts

The algorithm in Algorithm 1 uses the sparse vector technique to evaluate the current arm. In this section, we present a more direct algorithm that uses the sparse vector technique on a potential function. This more direct approach allows us to improve the polylogarithmic terms, which results in significant improvements for DP-OCO as we show in the next section.

We describe the algorithm in a more abstract way, which will translate more easily to OCO. Let μ be the uniform measure over the space of experts; this will be the uniform distribution over the finite number of experts in the case of DP-OPE, and the uniform distribution over the unit ball in $\mathbb{R}^d$ in the case of DP-OCO. At a high level, the algorithm is similar to Algorithm 1 in that it samples from the exponential mechanism, and keeps that sample until a certain condition is met. While the condition there dealt only with the chosen expert, we will instead test a global condition. For a parameter η, we define the potential $\phi_\eta(t) = \int \exp(-\eta \sum_{i=1}^{t} \ell_i(x)\mu(x)\mathrm{d}x$. The condition we check (using the sparse vector technique) is that $\log \phi_\eta(t)$ has changed by at least α since the last resample. We formally describe the algorithm in Algorithm 5.

We first observe some simple properties of ϕ_η.

Algorithm 5 Potential-based Algorithm for Experts

Require: Switching bound K, Parameters η, α, failure probability β, per-phase privacy parameter ε_0, measure μ on set of experts.

Set $k = 0$, $t^*(k) = 0$ and sample current expert $x_0 \sim \mu$.
for $t = 1$ to T **do**
  if $k < K$ **then**
    $Q = \mathsf{InitializeSparseVec}(\varepsilon_0, 2\alpha, \beta/T)$
    while Q.TestAboThr() = False **do**
      Set $x_t = x_{t-1}$
      Define a new query $q_t = \frac{1}{\eta}(\log \phi_\eta(t^*(k)) - \log \phi_\eta(t))$
      Add new query $Q.\mathsf{AddQuery}(q_t)$
      Receive loss function $\ell_t : [d] \to [0, 1]$
      Pay cost $\ell_t(x_t)$
      Update $t = t + 1$
    Sample x_t from the distribution:

$$P(x_t = x) = e^{-\eta \sum_{i=1}^{t-1} \ell_i(x)\mu(x)} / \phi_\eta(t-1)$$

    $k = k + 1$.
    $t^*(k) = t - 1$.
  else
    Set $x_t = x_{t-1}$
    Receive loss function $\ell_t : [d] \to [0, 1]$
    Pay cost $\ell_t(x_t)$

Lemma C.1. *Suppose that the per-step losses $\ell_t(s)$ are in $[0, 1]$. Then $\phi_\eta(t-1) \geq \phi_\eta(t) \geq e^{-\eta}\phi_\eta(t-1)$.*

Proof. For any x, we have

$$\sum_{i=1}^{t-1} \ell_i(x) \leq \sum_{i=1}^{t} \ell_i(x) \leq \sum_{i=1}^{t-1} \ell_i(x) + 1.$$

This implies that

$$e^{-\eta \sum_{i=1}^{t-1} \ell_i(x)} \geq e^{-\eta \sum_{i=1}^{t} \ell_i(x)} \geq e^{-\eta} e^{-\eta \sum_{i=1}^{t-1} \ell_i(x)}.$$

The claim follows by integrating with respect to μ. □

Lemma C.2. *Suppose that $\alpha \geq \frac{8\eta(\log T + \log 2T/\beta)}{\varepsilon_0}$. Then during the run of the algorithm, $\log \phi_\eta(t^*(k)) - \log \phi_\eta(t) \leq 3\alpha$ always holds except with probability β. Moreover, $\log \phi_\eta(t^*(k)) - \log \phi_\eta(t^*(k+1)) \geq \alpha$ holds for all k except with probability β.*

Proof. We use sparse vector with a threshold of 2α. The condition on α along with the first property of the sparse vector algorithm Lemma 2.3 implies that if $\log \phi_\eta(t^*) - \log \phi_\eta(t) \leq 3\alpha$. The second property of sparse vector implies the second property. □

We next analyze the loss of the algorithm.

Lemma C.3. *Suppose that $\eta < \frac{1}{2}$ and losses are in the range $[0, 1]$. For some $\gamma > 0$, let $L^*(\gamma)$ be such that $\Pr_{x\sim\mu}[\sum_{t=1}^{T} \ell_x(t) \leq L^*(\gamma)] \geq \gamma$. Let $K \geq \frac{\eta L^*(\gamma) + \log 1/\gamma}{\alpha}$. Then the expected loss of Algorithm 5 satisfies*

$$\mathbb{E}[\sum_{t=1}^{T} \ell_t(x_t)] \leq 2e^{3\alpha}(L^*(\gamma) + \frac{\log 1/\gamma}{\eta}) + 2\beta T.$$

Proof. We first observe that by the assumption on sufficient probability mass on experts with loss at most $L^*(\gamma)$, it follows that $\phi_\eta(T) \geq \gamma e^{-\eta L^*(\gamma)}\phi_\eta(0)$. Let E be the event that both the conditions on $\phi_\eta(t)$'s in Lemma C.2 hold. Thus $\Pr[E] \geq 1 - 2\beta$. The second condition implies that if we reach the bound K on the number of switches, then conditioned on E, $\phi_\eta(T) \leq e^{-K\alpha}\phi_\eta(0)$. Thus if $K \geq \frac{\eta L^*(\gamma) + \log 1/\gamma}{\alpha}$, then the switching bound is never reached under E.

Now note that if we sampled $x \sim e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)}/\phi_\eta(t-1)$, then we would have

$$\begin{aligned}
\mathbb{E}[\eta\ell_t(x)/2] &= \frac{1}{\phi_\eta(t-1)}\int \eta\ell_t(x)e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)}\mu(x)\mathrm{d}x \\
&\leq \frac{1}{\phi_\eta(t-1)}\int (1-e^{-\eta\ell_t(x)})e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)}\mu(x)\mathrm{d}x \\
&= \frac{1}{\phi_\eta(t-1)}\int (e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)} - e^{-\eta\sum_{i=1}^{t}\ell_i(x)})\mu(x)\mathrm{d}x \\
&= \frac{\phi_\eta(t-1)-\phi_\eta(t)}{\phi_\eta(t-1)} \\
&\leq \log\frac{\phi_\eta(t-1)}{\phi_\eta(t)}.
\end{aligned}$$

Here the first inequality uses the fact that for $y \in [0,1]$, we have $y/2 < 1 - e^{-y}$, and the second inequality uses $(1-r) \leq -\log r$ for $r \in [\frac{1}{2}, 1]$. The assumptions on η and the losses imply that the relevant conditions hold.

Our algorithm however uses an outdated sample based on the distribution at $t^*(k)$, whenever t is in phase k. Nevertheless, note that the cumulative losses increase with t so that $e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)}$ is smaller than $e^{-\eta\sum_{i=1}^{t^*(k)}\ell_i(x)}$.We then write

$$\begin{aligned}
\mathbb{E}[\eta\ell_t(x_t)/2\cdot\mathbf{1}(E)] &= \frac{1}{\phi_\eta(t^*(k))}\int \mathbf{1}(E)\eta\ell_t(x)e^{-\eta\sum_{i=1}^{t^*(k)}\ell_i(x)}\mu(x)\mathrm{d}x \\
&\leq \frac{\phi_\eta(t-1)}{\phi_\eta(t^*(k))}\cdot\frac{1}{\phi_\eta(t-1)}\int \eta\ell_t(x)e^{-\eta\sum_{i=1}^{t-1}\ell_i(x)}\mu(x)\mathrm{d}x \\
&\leq e^{3\alpha}\log\frac{\phi_\eta(t-1)}{\phi_\eta(t)}.
\end{aligned}$$

Rearranging and summing over the t steps, we get

$$\begin{aligned}
\mathbb{E}[\sum_{t=1}^{T}\ell_t(x_t)\cdot\mathbf{1}(E)] &\leq \frac{2e^{3\alpha}}{\eta}\sum_{t=1}^{T}\log\frac{\phi_\eta(t-1)}{\phi_\eta(t)} \\
&\leq \frac{2e^{3\alpha}}{\eta}\log\frac{\phi_\eta(0)}{\phi_\eta(T)}.
\end{aligned}$$

Since $\phi_\eta(T) \geq \gamma e^{-\eta L^*(\gamma)}\phi_\eta(0)$, we can upper bound this expression. Furthermore, $\mathbb{E}[\sum_{t=1}^{T}\ell_t(x_t)\cdot\mathbf{1}(E^c)]$ is clearly bounded $\Pr[E^c]\cdot T \leq 2\beta T$. The claim follows. □

This can yield pure or approximate DP. Indeed each phase of the algorithm satisfies $(\eta+\varepsilon_0)$-DP. Composing over the at most K phases, we get $(\eta+\varepsilon_0)$-DP or $(2(\eta+\varepsilon_0)\sqrt{K\log 1/\delta}, \delta)$-DP. With this in mind, we can now plug in appropriate values of $\eta, \alpha, \varepsilon_0$ to derive the following result for pure ε-DP.

Theorem 6. *(pure DP) Suppose that we run Algorithm 5 with parameters $\alpha = 1$, $\beta = \frac{1}{T^2}$, $K = 2\log 1/\gamma$, $\varepsilon_0 = \varepsilon/K$, and $\eta = \varepsilon_0/(56\log T) = \varepsilon/(112\log T\log 1/\gamma)$. If $L^*(\gamma) \leq \log^2 1/\gamma\log T/\varepsilon$, then this algorithm satisfies ε-DP and incurs loss $O(L^* + \frac{\log^2 1/\gamma\log T}{\varepsilon})$. In particular, for DP-OPE with d experts, assuming that that $OPT \leq \log^2 d\log T/\varepsilon$, we get regret $O(\log^2 d\log T/\varepsilon)$ by setting $\gamma = \frac{1}{d}$.*

Moreover, we have the following improved bound for (ε,δ)-DP.

Theorem 7. *((ε,δ)-DP) Suppose that for $\varepsilon < 1$, we run Algorithm 5 with parameters $\alpha = 1$, $\beta = \frac{1}{T^2}$, $K = 2\log 1/\gamma$, $\varepsilon_0 = \varepsilon/\sqrt{K\log 1/\delta}$, and $\eta = \varepsilon_0/(56\log T) = \varepsilon/(112\log T\sqrt{\log 1/\gamma\log 1/\delta})$. If $L^*(\gamma) \leq \log^{1.5} 1/\gamma\log T\sqrt{\log 1/\delta}/\varepsilon$, then*

this algorithm satisfies (ε, δ)-DP and incurs loss $O(L^* + \frac{\log^{1.5} 1/\gamma \log T \sqrt{\log 1/\delta}}{\varepsilon})$. In particular, for DP-OPE with d experts, assuming that $OPT \leq \log^{1.5} d \log T \sqrt{\log 1/\delta}/\varepsilon$, we get regret $O(\log^{1.5} d \log T \sqrt{\log 1/\delta}/\varepsilon)$ by setting $\gamma = \frac{1}{d}$.

For the setting of DP-OCO in the realizable case, we get that by Lipschitzness, a small ball of radius r around x^* has loss $O(rLT)$. Setting $r = (LT)^{-1}$ then ensures that this value is at most 1 This ball has measure $\gamma = (r/D)^{-d}$. Scaling so that $D, L \leq 1$, we get an regret $O(d^2 \log^3 T/\varepsilon)$ for the case of ε-DP and $O(d^{1.5} \log^{2.5} T \sqrt{\log 1/\delta}/\varepsilon)$ for (ε, δ)-DP. Finally we note that this algorithm only accesses the loss functions to sample from the exponential mechanism, and compute the potential $\phi_\eta(t)$. Both of these can be implemented in polynomial time for the case of μ being uniform over the ball, and the losses being convex using standard techniques from logconcave sampling. Thus this algorithm can be run in polynomial time for DP-OCO.

D. Missing proofs for Section 4

D.1. Proof of Theorem 3

Let $x_1, \ldots, x_T$ be the experts chosen by the algorithm. First, Theorem 2 implies that this algorithm obtains the following regret with respect to the best expert

$$\sum_{t=1}^{T} \ell_t(x_t) - L^\star_{\mathsf{experts}} \leq (L^\star_{\mathsf{experts}} + \frac{1}{\varepsilon}) \cdot O\left(\mathsf{poly}(d \log \frac{DT}{\rho\beta})\right),$$

where $L^\star_{\mathsf{experts}} = \min_{x \in X^\rho_{\mathsf{experts}}} \sum_{t=1}^{T} \ell_t(x)$. Since ℓ_t is L-Lipschitz for each $t \in [T]$, we obtain that

$$|L^\star - L^\star_{\mathsf{experts}}| = |\min_{x \in X} \sum_{t=1}^{T} \ell_t(x) - \min_{x \in X^\rho_{\mathsf{experts}}} \sum_{t=1}^{T} \ell_t(x)| \leq TL\rho.$$

Overall this gives

$$\sum_{t=1}^{T} \ell_t(x_t) - L^\star \leq (L^\star + TL\rho + \frac{1}{\varepsilon}) \cdot O\left(\mathsf{poly}(d \log \frac{DT}{\rho\beta})\right).$$

Setting $\rho = 1/(LT)$ proves the claim.