Private Online Prediction from Experts with Small Losses

Bo Li[1,2] **Peng Ye**[2]
[1]Guangzhou HKUST Fok Ying Tung Research Institute
[2]Department of Computer Science and Engineering, HKUST
bli@ust.hk, pyeac@connect.ust.hk

Abstract

We consider online prediction from experts, a fundamental problem in machine learning, under differential privacy constraints. Existing private algorithms achieve near-optimal regret $\tilde{O}(\sqrt{T})$, where T is the time horizon. However, this bound becomes suboptimal when $L^\star$, the cumulative loss of the best expert, is significantly less than T. In this work, we present the first differentially private algorithm with regret $\tilde{O}(\sqrt{L^\star})$, offering a substantial improvement in the low-loss regime. Our regret bound matches the non-private lower bound up to poly-logarithmic factors for constant privacy parameters, demonstrating that privacy incurs only a small cost in this setting. Our result assumes that the input sequence is oblivious, i.e., it is fixed in advance and independent of the algorithm's outputs. We also prove that an $\tilde{\Omega}(\sqrt{T})$ regret is inevitable for adaptively generated inputs, even when $L^\star = 0$.

1 Introduction

Online prediction from experts (OPE) is a fundamental problem in machine learning. In this problem, the learner receives a stream of T loss functions $\ell_1, \ldots, \ell_T$. In each round t, the learner chooses an expert $i_t \in [N]$ before seeing ℓ_t and suffers loss $\ell_t(i_t)$. The learner aims to minimize its total loss, which is compared to that of the best expert in hindsight, denoted by $L^\star$. The difference between these two quantities is called the regret, which measures the learner's performance for this problem.

A critical concern when deploying machine learning algorithms on sensitive data (e.g., financial or medical records) is privacy leakage. To address this, differential privacy (DP) [9, 8] has emerged as the de facto standard for formalizing the privacy guarantee provided by an algorithm. Informally, an algorithm is said to be differentially private if changing a single input data point alters its output distribution only negligibly. As a result, such a change in an individual's information is unlikely to be detected, thereby protecting personal privacy.

Given the importance of OPE, studying it under DP has received significant attention. Previous work [3, 4] has developed algorithms that achieve regret $\tilde{O}(\sqrt{T})$, which is nearly optimal because it matches the $\Omega(\sqrt{T \log N} + \log N)$ lower bound [6]—a bound that holds even without any privacy constraints. These results suggest that the cost of privacy in OPE is negligible in the worst case.

In many practical settings—such as financial forecasting, weather prediction, or personalized recommendations—domain experts are specifically selected for their accuracy, so their cumulative loss tends to be low over time. However, the $\tilde{O}(\sqrt{T})$ regret becomes less meaningful when the loss of the best expert $L^\star$ is small because the regret itself can be substantially greater than $L^\star$, rendering the learner's loss considerably higher than $L^\star$. As an illustration, suppose $L^\star = T^{1/4}$. An algorithm with regret $\sqrt{T}$ may incur a loss of $\sqrt{T}$, which is $T^{1/4}$ times larger than $L^\star$. In contrast, when $L^\star = T^{3/4}$ the total loss is only $(1 + o(1))L^\star$. Thus, worst-case regret bounds are not suitable for OPE with small losses.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In the non-private setting, a natural remedy is to run the multiplicative weights method with a larger learning rate: increasing the learning rate from $\sqrt{\ln N/T}$ to $\sqrt{\ln N/L^\star}$ yields a regret bound of $O(\sqrt{L^\star \log N} + \log N)$ [19, 13]. Unfortunately, such a strategy cannot be directly applied under DP. This is because a large learning rate amplifies the algorithm's sensitivity to individual data points, thereby violating privacy constraints. Indeed, all existing private OPE algorithms impose a strict upper bound on the learning rate to ensure DP, which prevents them from adapting to small $L^\star$. Consequently, it remains unclear whether an $\tilde{O}(\sqrt{L^\star})$ regret bound is achievable in the private setting.

1.1 Our contribution

In this work, we assume that $\ell_1, \ldots, \ell_T$ are chosen by an oblivious adversary. That is, they are fixed in advance and independent of the outputs of the learner.

We propose the first differentially private algorithm that attains regret $\tilde{O}(\sqrt{L^\star})$ for OPE. This nearly matches the optimal regret without privacy constraints, suggesting that the cost of privacy remains small in the low-loss regime. The results are formally stated in the theorems below.

Theorem 1.1. *Let $\ell_1, \ldots, \ell_T$ be chosen by an oblivious adversary. Given $\varepsilon \le 1$, there exists an ε-differentially private algorithm for OPE with expected regret*

$$O\left(\left(\sqrt{\log N \log T} + \log N\right) \cdot \sqrt{\frac{L^\star}{\varepsilon}}\right) + (1 + \ln(1 + L^\star)) \cdot \iota_1,$$

where $\iota_1 = O(\log^2 N/\varepsilon + \log N \log T/\varepsilon)$.

Theorem 1.2. *Let $\ell_1, \ldots, \ell_T$ be chosen by an oblivious adversary. Given $\varepsilon \le 1$ and $0 < \delta \le 1/2$, there exists an (ε, δ)-differentially private algorithm for OPE with expected regret*

$$O\left(\sqrt{\frac{L^\star \log N \log T}{\varepsilon}} + \log N \cdot \min\left(\frac{(L^\star \log(1/\delta))^{1/3}}{\varepsilon^{2/3}}, \sqrt{\frac{L^\star}{\varepsilon}}\right)\right) + (1 + \ln(1 + L^\star)) \cdot \iota_2,$$

where $\iota_2 = O\left(\log^{3/2} N \sqrt{\log(\min(1/\delta, N))}/\varepsilon + \log N \log T/\varepsilon\right)$.

Comparison with previous results. When $L^\star = 0$, our results recover the bounds obtained by Asi et al. [2] for the realizable setting. When $L^\star$ grows, our algorithm yields a quadratic improvement over their $\tilde{O}(L^\star)$ regret bound. Compared to the $\tilde{O}(\sqrt{T})$ regret obtained by Asi et al. [3, 4], our results are significantly better when $L^\star$ is much smaller than T.

As shown by Asi et al. [3], the above improvement cannot be achieved against an adaptive adversary under pure DP: any OPE algorithm must incur an $\Omega(T)$ regret even if $L^\star = 0$. We additionally prove that it remains impossible under approximate DP by providing an $\tilde{\Omega}(\sqrt{T})$ lower bound, which also holds when $L^\star = 0$.

Theorem 1.3. *Let T be sufficiently large and $N \ge 2T$. Any adaptively (ε, δ)-differentially private OPE algorithm with $\varepsilon \le 1$ and $\delta \le 1/T^3$ must incur an $\Omega(\min(T, \sqrt{T}/(\varepsilon \log T)))$ expected regret when $\ell_1, \ldots, \ell_T$ are chosen by an adaptive adversary, even if it is promised that $L^\star = 0$.*

1.2 Related work

In the non-private setting, the OPE problem has been extensively studied. The multiplicative weights algorithm [19, 23] was shown to achieve a data-dependent regret bound of $O(\sqrt{L^\star \log N} + \log N)$ with an appropriately tuned learning rate [13]. There are also other methods with the same regret bound, such as follow-the-perturbed-leader [15, 18, 16]. It was formally proved by Cesa-Bianchi et al. [6] that such a bound is asymptotically optimal.

The private version of OPE was first considered by Dwork et al. [11]. It is a folklore result that the multiplicative weights method preserves differential privacy, which follows from the exponential mechanism [20] and advanced composition [12]. A series of subsequent works [21, 17, 1–4] further investigated the problem. In the oblivious adversarial setting considered in this paper, Asi et al. [3] proposed algorithms with regret $\tilde{O}(\sqrt{T})$ under pure DP and $O(\sqrt{T \log N}) + \tilde{O}(T^{1/3})$ under approximate DP, demonstrating that the cost of privacy is negligible. The bound for approximate DP

was later refined by Asi et al. [4], who further optimized the dependence on the privacy parameter ε. However, these results are unfavorable when $L^\star$ is much smaller than T. Asi et al. [2] studied the realizable setting, where $L^\star = 0$, and gave algorithms with $\tilde{O}(1)$ regret. Their algorithms can be extended to the agnostic setting where $L^\star > 0$. Nevertheless, the resulting regret bound is $\tilde{O}(L^\star)$, which is far from optimal.

2 Preliminaries

Online prediction from experts can be formulated as a T-round game played between a learner and an adversary. In this work, we assume an oblivious adversary who determines a sequence of loss functions $\ell_1, \ldots, \ell_T$ in advance and releases them to the learner one by one, where every $\ell_t : [N] \to [0, 1]$ assigns each of the N experts a loss in $[0, 1]$. In round t, the learner chooses an expert $i_t \in [N]$ before receiving ℓ_t and incurs loss $\ell_t(i_t)$. The performance of the learner is assessed by the expected regret, defined as

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i)\right].$$

For ease of notation, we write $L^\star$ to denote $\min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i)$. Note that $L^\star$ is determined in advance because the adversary is oblivious. However, it is unknown to the learner until the end.

One common approach to solving OPE is the celebrated multiplicative weights method, which in each round t samples i_t such that

$$\Pr[i_t = i] = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}}{\sum_{i'\in[N]} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i')}},$$

where η is the learning rate. The following lemma provides an upper bound on its expected regret.

Lemma 2.1 ([13, 5]). *For $0 < \eta \le 1$, we have*

$$\sum_{t=1}^{T} \sum_{i\in[N]} P_t(i)\ell_t(i) - L^\star = O\left(\frac{\log N}{\eta} + \eta L^\star\right),$$

where $P_t(i) = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}}{\sum_{i'\in[N]} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i')}}$.

We then introduce the notion of differential privacy in the context of OPE.

Definition 2.2 (Differential Privacy [9, 8]). An algorithm $\mathcal{A}$ is (ε, δ)-differentially private if, for any pair of sequences $S = (\ell_1, \ldots, \ell_T)$ and $S' = (\ell'_1, \ldots, \ell'_T)$ that differ in a single function, and for any subset O of the output range, we have

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

When $\delta = 0$, we usually omit δ and say $\mathcal{A}$ is ε-differentially private. This case is referred to as pure DP, while the case where $\delta > 0$ is referred to as approximate DP.

A key ingredient in our design is the AboveThreshold mechanism, which receives a sequence of queries and privately identifies the first one that exceeds a prespecified threshold.

Theorem 2.3 (AboveThreshold [10, 7]). *We say a query q has sensitivity 1 if $|q(S) - q(S')| \le 1$ for any pair of sequences S and S' that differ in a single function. Given a threshold τ, there exists an ε-differentially private mechanism* AboveThreshold *that at each round t receives a sensitivity-1 query q_t and outputs $\sigma_t \in \{\top, \bot\}$, continuing until it either produces the first $\top$ or has processed all queries. If there are at most T queries, then with probability $1 - \beta$, the* AboveThreshold *mechanism is α-accurate for*

$$\alpha = \frac{8(\ln T + \ln(2/\beta))}{\varepsilon}.$$

Namely, if the mechanism halts after k rounds, we have $q_t \ge \tau - \alpha$ for all $t \in [k]$ with $\sigma_t = \top$ and $q_t \le \tau + \alpha$ for all $t \in [k]$ with $\sigma_t = \bot$.

Every step of the multiplicative weights method is exactly the exponential mechanism with privacy parameter $\varepsilon = 2\eta$. This property, combined with advanced composition, has been widely used in the literature to develop private OPE algorithms.

Lemma 2.4 (Exponential Mechanism [20])**.** *Let P_t be defined as in Lemma 2.1. The process of sampling i_t from P_t is 2η-differentially private.*

3 Algorithm and analysis

In this section, we present our algorithm and analyze its privacy and regret guarantees.

Our algorithm is built on the shrinking dartboard algorithm [14], which was also leveraged by Asi et al. [3] to develop algorithms with $\tilde{O}(\sqrt{T})$ regret. The key insight is that the marginal distribution of the multiplicative weights method changes only slightly between consecutive rounds. Consequently, it suffices to update the output with only a small probability after each round. By reducing the number of updates, one can allocate a larger privacy budget for each update (i.e., use a larger learning rate η), thereby achieving smaller regret.

However, the shrinking dartboard algorithm is not inherently private because the probability of making an update depends on the underlying data (specifically, $\ell_t(i_t)$, i.e., the loss associated with the currently selected expert i_t). Asi et al. [3, 4] addressed this by forcing the algorithm to perform an additional update at each round with a probability independent of the data. However, to satisfy even approximate DP, this probability must be at least $\tilde{\Omega}(T^{-1/3})$. By advanced composition, this imposes an $\tilde{O}(1/T^{1/3})$ upper bound on the learning rate η, which in turn yields an $\tilde{O}(T^{1/3})$ regret bound according to Lemma 2.1. This exceeds our desired $\tilde{O}(\sqrt{L^\star})$ regret whenever $L^\star < T^{2/3}$.

We address this by delaying the update until the currently selected expert has incurred a sufficiently large loss, and then making an update with a probability that depends on the cumulative loss accumulated during the delay. Our key consideration here is that, while such a probability is data-dependent, performing an update with this probability preserves privacy. Moreover, the delay before an update can be implemented using the AboveThreshold mechanism, which is also differentially private. In addition, we impose a lower bound on the probability of updating and an upper bound on the number of updates, which are crucial for satisfying DP in the worst case. The overall algorithm is described in Algorithm 1. We formally prove its privacy guarantee in Section 3.1 and regret bound in Section 3.2.

3.1 Privacy analysis

Let $\mathsf{Comp}_\delta(k, \varepsilon_0)$ be the smallest ε such that the composition of k ε_0-differentially private algorithms is (ε, δ)-differentially private. We prove the privacy guarantee of Algorithm 1 in the following lemma.

Lemma 3.1. *Algorithm 1 is (ε, δ)-differentially private for $\varepsilon = \varepsilon_1 + \max(\eta, \varepsilon_2) + \mathsf{Comp}_\delta(V, 2\eta)$.*

Proof. We analyze the augmented transcript consisting of the initial expert i_1, the outputs of all AboveThreshold instances, the Z_u's, and the experts generated at the (at most V) resampling steps. The learner's output sequence is a post-processing of this transcript.

We view the algorithm as the concurrent composition of three types of interactive mechanisms: the collection of AboveThreshold instances, the mechanism generating the Z_u's, and the at most V resampling mechanisms. Throughout the proof, when comparing two neighboring input sequences, we consider an arbitrary common prefix of the interaction transcript. Thus, all previously generated AboveThreshold outputs, Z_u's, resampled experts, and the resulting termination times are fixed and identical on the two executions.

We first show that the collection of AboveThreshold instances is ε_1-differentially private. Let the two neighboring sequences S and S' differ at round $\tilde{t}$. Along any fixed common interaction transcript, there is at most one AboveThreshold instance whose query stream contains the differing loss. All earlier instances receive identical query streams and therefore have identical output distributions on S and S'.

Consider the potentially affected instance. Conditioned on the common transcript up to its launch, its starting time and selected expert are fixed and identical on the two executions. Hence, every

Algorithm 1: Private online prediction from experts

Input: privacy parameters ε_1 and ε_2, learning rate η, threshold τ, limit on updates V

$u \leftarrow 1$, $v \leftarrow 0$, and $t_0 \leftarrow 1$
Sample i_1 uniformly from $[N]$
Initiate an instance of AboveThreshold with privacy parameter ε_1 and threshold τ
Observe ℓ_1 and suffer loss $\ell_1(i_1)$
for $t = 2, \ldots, T$ **do**
  if $v < V$ **then**
    Define query $q_t \equiv \sum_{s=t_{u-1}}^{t-1} \ell_s(i_{t-1})$
    Feed q_t to AboveThreshold and obtain outcome σ_t
    if $\sigma_t = \top$ **then**
      $r_u \leftarrow \max(1/(e^{\varepsilon_2} - 1), \sum_{s=t_{u-1}}^{t-1} \ell_s(i_{t-1}))$
      Sample Z_u from $\mathsf{Ber}(e^{-\eta r_u})$
      if $Z_u = 1$ **then**
        $i_t \leftarrow i_{t-1}$
      else
        Sample $i_t \leftarrow i$ with probability $\frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}}{\sum_{i' \in [N]} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i')}}$ for $i \in [N]$
        $v \leftarrow v + 1$
      end
      $t_u \leftarrow t$
      $u \leftarrow u + 1$
      Initiate an instance of AboveThreshold with privacy parameter ε_1 and threshold τ
    else
      $i_t \leftarrow i_{t-1}$
    end
  else
    $i_t \leftarrow i_{t-1}$
  end
  Observe ℓ_t and suffer loss $\ell_t(i_t)$
end

query issued by this instance has sensitivity at most 1. By the privacy guarantee of AboveThreshold, the transcript generated by this instance is ε_1-differentially private. If this instance terminates, then conditioned on a common transcript through its termination, every subsequent AboveThreshold instance starts after round $\tilde{t}$. Consequently, all subsequent instances receive identical query streams on S and S'. If the affected instance never terminates, there are no subsequent instances. Applying the chain rule to the conditional output distributions therefore shows that the collection of all AboveThreshold instances, viewed as one interactive mechanism, is ε_1-differentially private.

We next analyze the privacy loss from generating the Z_u's. Let $Z_1, Z_2, \ldots$ and $Z'_1, Z'_2, \ldots$ denote the Z_u's generated on neighboring sequences S and S', respectively. Similarly, let r_u's and r'_u's denote the r_u-values computed on S and S', respectively. As above, all probabilities involving Z_u's are understood to be conditioned on the relevant fixed common transcript prefix.

Along any common transcript path, the intervals $[t_{u-1}, t_u - 1]$ corresponding to generated Z_u's are disjoint. Hence, there is at most one index j such that S and S' differ at round $\tilde{t} \in [t_{j-1}, t_j - 1]$. If such j does not exist, then $r_u = r'_u$ for every Z_u, which implies that the two Z_u-transcripts have identical conditional distributions.

Otherwise, we have $r_u = r'_u$ for all $u < j$ and $|r_j - r'_j| \leq 1$. Therefore,

$$\frac{\Pr[Z_j = 1]}{\Pr[Z'_j = 1]} = \frac{e^{-\eta r_j}}{e^{-\eta r'_j}} \leq e^{\eta}.$$

Moreover, since $\min(r_j, r'_j) \geq 1/(e^{\varepsilon_2} - 1)$, we have

$$\frac{\Pr[Z_j = 0]}{\Pr[Z'_j = 0]} = \frac{1 - e^{-\eta r_j}}{1 - e^{-\eta r'_j}} \leq \frac{\min(r_j, r'_j) + |r_j - r'_j|}{\min(r_j, r'_j)} \leq e^{\varepsilon_2},$$

where we use the fact that $\frac{1-e^{-y}}{1-e^{-x}} \leq \frac{y}{x}$ for $0 < x \leq y$. For every later $u > j$, condition on an arbitrary common transcript prefix up to the generation of Z_u. The selected expert and the interval $[t_{u-1}, t_u - 1]$ are then fixed and identical on the two neighboring sequences. Since this interval occurs after the differing round, we have $r_u = r'_u$. Thus, Z_u and Z'_u have the same conditional distribution on the two executions. Therefore, by the chain rule, only Z_j contributes privacy loss, and the generation of the Z_u's is $\max(\eta, \varepsilon_2)$-differentially private.

Finally, by Lemma 2.4, each of the at most V resampling steps is 2η-differentially private conditioned on an arbitrary common transcript prefix. By the concurrent composition theorem [22], the adaptive interleaving of these mechanisms has the same composition guarantee as their non-interactive composition. Hence the augmented transcript, and therefore the learner's output sequence by post-processing, is (ε, δ)-differentially private. □

3.2 Regret analysis

We proceed to analyze the regret. The following key lemma provides lower and upper bounds on the cumulative loss from round t_{u-1} through round $t_u - 1$.

Lemma 3.2. *With probability $1 - \beta_1$, we have*

$$\tau - \frac{8(\ln T + \ln(2T/\beta_1))}{\varepsilon_1} \leq \sum_{s=t_{u-1}}^{t_u - 1} \ell_s(i_{t_u - 1}) \leq \max\left(\tau + \frac{8(\ln T + \ln(2T/\beta_1))}{\varepsilon_1}, 0\right) + 1$$

for all $u \geq 1$. The probability is taken solely over the randomness of the AboveThreshold *instances.*

Proof. By Theorem 2.3 and the union bound, with probability $1 - \beta_1$ we have $q_t \geq \tau - \alpha$ if $\sigma_t = \top$ and $q_t \leq \tau + \alpha$ if $\sigma_t = \bot$ for all $t \in [T]$, where $\alpha = 8(\ln T + \ln(2T/\beta_1))/\varepsilon_1$. The lower bound follows directly from the fact that $\sigma_{t_u} = \top$. For the upper bound, the inequality trivially holds if $t_{u-1} = t_u - 1$. When $t_{u-1} < t_u - 1$, we have $\sigma_{t_u - 1} = \bot$ and $i_{t_u - 2} = i_{t_u - 1}$. Hence,

$$\sum_{s=t_{u-1}}^{t_u - 1} \ell_s(i_{t_u - 1}) \leq \sum_{s=t_{u-1}}^{t_u - 2} \ell_s(i_{t_u - 2}) + 1 \leq \tau + \alpha + 1.$$

□

Next, we show that the number of updates is $\tilde{O}(\eta L^\star)$ with high probability. As a consequence, the algorithm works normally (i.e., it won't reach the update limit) if $V = \tilde{O}(\eta L^\star)$. By the composition theorems, this allows us to set $\eta = \tilde{\Theta}(1/\sqrt{L^\star})$, which minimizes the regret bound in Lemma 2.1.

Lemma 3.3. *Let $\tau \geq \frac{8(\ln T + \ln(2T/\beta_1))}{\varepsilon_1} + \frac{1}{e^{\varepsilon_2} - 1}$. Conditioned on the event in Lemma 3.2, we have*

$$\Pr[v_T < 71(\ln N + \eta L^\star + 1) + 280 \ln(1/\beta_2)] \geq 1 - \beta_2,$$

where v_T is the value of v when Algorithm 1 terminates.

Proof. Note that v can increase by at most 1 in each round. Therefore, the condition $v < V$ is always met given that $V \geq T$. In what follows, we will prove the result assuming $V \geq T$ because v_T can only be smaller if we decrease V.

Define a potential function

$$\Phi_t = \sum_{i=1}^{N} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}.$$

We have $N = \Phi_1 \geq \cdots \geq \Phi_{T+1} \geq e^{-\eta L^\star}$. Let $s_j \in [T]$ be the largest index such that $\Phi_{s_j} \geq \Phi_1/e^j$ and let m be the smallest positive integer such that $s_m = T$. Then $m \leq \ln(\Phi_1/\Phi_{T+1}) + 1 \leq \ln N + \eta L^\star + 1$.

Suppose an update happens at round $t \in [s_{j-1}+1, s_j]$ (set $s_0 = 0$ for simplicity). Let E denote the event that another update happens at some round $t' \in [t+1, s_j]$. Let

$$S = \left\{ i \in [N] : \sum_{s=t}^{s_j-1} \ell_s(i) \le 2/\eta \right\}.$$

By Lemma 3.2, we have $\sum_{s=t_{u-1}}^{t_u-1} \ell_s(i_{t_u-1}) \ge 1/(e^{\varepsilon_2}-1)$ for any $u \ge 1$. This implies $r_u = \sum_{s=t_{u-1}}^{t_u-1} \ell_s(i_{t_u-1})$. Thus, for any $i \in S$, we have

$$\Pr[E \mid i_t = i] \le 1 - e^{-\eta \sum_{s=t}^{s_j-1} \ell_s(i)} \le 1 - e^{-2}.$$

Note that

$$\begin{aligned}
\Phi_{s_j} &= \sum_{i \in S} e^{-\eta \sum_{s=1}^{s_j-1} \ell_s(i)} + \sum_{i \notin S} e^{-\eta \sum_{s=1}^{s_j-1} \ell_s(i)} \\
&\le \sum_{i \in S} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)} + e^{-2} \sum_{i \notin S} e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)} \\
&= \Phi_t \Pr[i_t \in S] + e^{-2} \Phi_t \Pr[i_t \notin S] \\
&= \Phi_t (1 - \Pr[i_t \notin S]) + e^{-2} \Phi_t \Pr[i_t \notin S].
\end{aligned}$$

Rearranging this yields

$$\Pr[i_t \notin S] \le \frac{1 - \Phi_{s_j}/\Phi_t}{1 - e^{-2}} \le \frac{1 - e^{-1}}{1 - e^{-2}}.$$

Therefore,

$$\begin{aligned}
\Pr[E] &\le \Pr[i_t \in S] \cdot \Pr[E \mid i_t \in S] + \Pr[i_t \notin S] \\
&= (1 - \Pr[i_t \notin S]) \cdot (1 - e^{-2}) + \Pr[i_t \notin S] \\
&\le 1 - e^{-2} + \frac{1 - e^{-1}}{1 - e^{-2}} \cdot e^{-2} \\
&< 0.97.
\end{aligned}$$

As a consequence, the probability that $v_T \ge n + m$ can be bounded by the probability that the sum of n Bernoulli random variables with mean 0.03 is less than m. By the Chernoff bound, this is at most $\exp(-0.03n/8)$ given that $0.03n \ge 2m$. Thus, it suffices to set $n = 70m + 280\ln(1/\beta_2)$. □

The most challenging task in our proof is to compare the output distribution of Algorithm 1 with that of the multiplicative weights method. To this end, we introduce Algorithm 2, which makes the following modifications to Algorithm 1:

- The limit V on the number of updates and the lower bound on r_u are removed.
- The AboveThreshold mechanisms are replaced by a delay function d.
- The learning process can start from an arbitrary round T_s.
- The update procedure at round T_e is forced.

We first prove that the output distributions of Algorithm 2 and the multiplicative weights method are the same at round T_e.

Lemma 3.4. *For any $T_s \in [T_e]$ and $d : \mathbb{Z}^3 \to \mathbb{N}$, the output distribution of Algorithm 2 at round T_e satisfies*

$$\Pr[i_{T_e} = i] = \frac{e^{-\eta \sum_{s=1}^{T_e-1} \ell_s(i)}}{\sum_{i' \in [N]} e^{-\eta \sum_{s=1}^{T_e-1} \ell_s(i')}}.$$

Proof. We prove by backward induction on T_s. When $T_s = T_e$, the conclusion holds since we sample i_{T_e} exactly according to the desired distribution. Now consider $T_s = t' \in [T_e - 1]$ and suppose that the result holds for all $T_s \ge t' + 1$. Fix some $i_{t'} = k \in [N]$. Define $t_0 = t'$ and

$$t_j = \min(T_e, \max(t_{j-1} + 1, d(t_{j-1}, i_{t'}, j))).$$

Algorithm 2: An auxiliary algorithm

Input: starting point T_s, ending point T_e, learning rate η, delay function $d : \mathbb{Z}^3 \to \mathbb{N}$

$u \leftarrow 1$ and $t_0 \leftarrow T_s$

Sample $i_{T_s} \leftarrow i$ with probability $\frac{e^{-\eta \sum_{s=1}^{T_s-1} \ell_s(i)}}{\sum_{j\in[N]} e^{-\eta \sum_{s=1}^{T_s-1} \ell_s(j)}}$ for $i \in [N]$

Observe ℓ_{T_s} and suffer loss $\ell_{T_s}(i_{T_s})$

for $t = T_s + 1, \ldots, T_e$ **do**

  if $t \geq d(t_{u-1}, i_{t_{u-1}}, u)$ *or* $t = T_e$ **then**

    $r_u \leftarrow \sum_{s=t_{u-1}}^{t-1} \ell_s(i_{t-1})$

    Sample Z_u from $\mathsf{Ber}(e^{-\eta r_u})$

    if $Z_u = 1$ **then**

      $i_t \leftarrow i_{t-1}$

    else

      Sample $i_t \leftarrow i$ with probability $\frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}}{\sum_{j\in[N]} e^{-\eta \sum_{s=1}^{t-1} \ell_s(j)}}$ for $i \in [N]$

    end

    $t_u \leftarrow t$

    $u \leftarrow u + 1$

  else

    $i_t \leftarrow i_{t-1}$

  end

  Observe ℓ_t and suffer loss $\ell_t(i_t)$

end

These are exactly the t_u's in Algorithm 2 on the event $i_{t'} = \cdots = i_{T_e-1} = k$. Let m be the first index such that $t_m = T_e$. Note that

$$\Pr[Z_1 = \cdots = Z_m = 1 \mid i_{t'} = k] = e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(k)}.$$

Moreover, by the induction hypothesis, we have

$$\Pr[i_{T_e} = i \mid Z_1 = \cdots = Z_{j-1} = 1, Z_j = 0, i_{t'} = k] = \frac{e^{-\eta \sum_{s=1}^{T_e-1} \ell_s(i)}}{\sum_{i'\in[N]} e^{-\eta \sum_{s=1}^{T_e-1} \ell_s(i')}}$$

for any $j \in [m]$ since the learning process starting from t_j can be seen as running Algorithm 2 with d shifted by j along the third dimension and $T_s = t_j$. For simplicity, write $w_t(i) = e^{-\eta \sum_{s=1}^{t-1} \ell_s(i)}$ and $\Phi_t = \sum_{i\in[N]} w_t(i)$ for $t \in [T_e]$. Note that $i_{T_e} \neq i_{t'}$ only if $Z_j = 0$ for some j. Therefore, we have

$$\Pr[i_{T_e} = i \mid i_{t'} = k] = \left(1 - e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(k)}\right) \cdot \frac{w_{T_e}(i)}{\Phi_{T_e}}$$

for $i \neq k$ and

$$\Pr[i_{T_e} = k \mid i_{t'} = k] = e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(k)} + \left(1 - e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(k)}\right) \cdot \frac{w_{T_e}(k)}{\Phi_{T_e}}.$$

As a consequence, we have

$$\begin{aligned}
\Pr[i_{T_e} = i] &= \sum_{k\in[N]} \Pr[i_{T_e} = i \mid i_{t'} = k] \cdot \Pr[i_{t'} = k] \\
&= \sum_{k\in[N]} \left(1 - e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(k)}\right) \cdot \frac{w_{T_e}(i)}{\Phi_{T_e}} \cdot \frac{w_{t'}(k)}{\Phi_{t'}} + e^{-\eta \sum_{s=t'}^{T_e-1} \ell_s(i)} \cdot \frac{w_{t'}(i)}{\Phi_{t'}} \\
&= \frac{w_{T_e}(i)}{\Phi_{T_e}} \cdot \frac{\Phi_{t'} - \Phi_{T_e}}{\Phi_{t'}} + \frac{w_{T_e}(i)}{\Phi_{t'}} \\
&= \frac{w_{T_e}(i)}{\Phi_{T_e}}
\end{aligned}$$

as desired. □

We leverage the fact that Algorithm 2 behaves similarly to Algorithm 1 before round T_e when $T_s = 1$: Lemma 3.2 and Lemma 3.3 suggest that the first modification has little impact, while the second modification can be seen as fixing the random coins used by the AboveThreshold mechanisms (see Appendix B for a more detailed explanation). By analyzing their behavior at round T_e and applying the above lemma, we can argue that with high probability, the output distribution of Algorithm 1 at each round is pointwise at most $e^{O(\eta(\tau+\log T/\varepsilon_1))} \approx 1 + O(\eta\tau + \eta \log T/\varepsilon_1)$ times that of the multiplicative weights mechanism. This leads to the following regret bound.

Lemma 3.5. *Suppose* $\tau \geq \frac{8(\ln T + 2\ln(2T))}{\varepsilon_1} + \frac{1}{e^{\varepsilon_2}-1} \geq 1$, $0 < \eta\tau \leq 1$, *and* $V \geq 71(\ln N + \eta L^\star + 1) + 280 \ln T$. *The expected regret of Algorithm 1 is at most*

$$O\left(\frac{\log N}{\eta} + \tau\eta L^\star\right).$$

Proof. Let $\beta_1 = 1/T$. Since the regret cannot be larger than T, it suffices to prove the desired conclusion conditioned on the event in Lemma 3.2. As argued in the proof of Lemma 3.3, for any $u \geq 1$ we have

$$r_u = \sum_{s=t_{u-1}}^{t_u - 1} \ell_s(i_{t_u - 1}).$$

By Lemma 3.2, there exists $\Delta = \tau + O(\log T/\varepsilon_1) + 1 = O(\tau)$ such that $r_u \leq \Delta$ for all $u \geq 1$.

We first prove the result under the condition that $V \geq T$. Fix a realization of the random coins used by the AboveThreshold mechanisms and let $\hat{Q}_t(i)$ be the probability that Algorithm 1 outputs $i_t = i$. Since $V \geq T$, there is no restriction on the number of updates. Hence, this realization induces a delay function d so that Algorithm 1 and Algorithm 2 (with this delay function d and $T_s = 1$) behave identically in the first $t - 1$ rounds.

Let P_t be as defined in Lemma 2.1 and consider any $i \in [N]$. By Lemma 3.4 with $1 = T_s < T_e = t$, Algorithm 2 outputs $i_t = i$ with probability $P_t(i)$. Since it produces $i_{t-1} = i$ with probability $\hat{Q}_{t-1}(i)$, we have $P_t(i) \geq \hat{Q}_{t-1}(i) \cdot e^{-\eta\Delta}$. Now consider Algorithm 1. If it outputs $i_{t-1} \neq i$, then the probability that it outputs $i_t = i$ is at most $(1 - e^{-\eta\Delta})P_t(i)$. As a consequence, we have

$$\begin{aligned}
\hat{Q}_t(i) &\leq \hat{Q}_{t-1}(i) + (1 - \hat{Q}_{t-1}(i)) \cdot (1 - e^{-\eta\Delta})P_t(i) \\
&\leq e^{\eta\Delta}P_t(i) + (1 - e^{-\eta\Delta})P_t(i) \\
&\leq e^{2\eta\Delta}P_t(i),
\end{aligned}$$

where the last inequality is due to the fact that $1 + x - 1/x \leq x^2$ for $x \geq 1$. Moreover, we have $\hat{Q}_1(i) = P_1(i)$ since we sample i_1 uniformly at random. Thus, $\hat{Q}_t(i) \leq e^{2\eta\Delta}P_t(i)$ for all $t \in [T]$. By Lemma 2.1, the expected regret of Algorithm 1 with $V \geq T$ is

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{i\in[N]} \hat{Q}_t(i)\ell_t(i) - L^\star &\leq e^{2\eta\Delta} \sum_{t=1}^{T} \sum_{i\in[N]} P_t(i)\ell_t(i) - L^\star \\
&= e^{2\eta\Delta} \cdot O\left(\frac{\log N}{\eta} + \eta L^\star\right) + (e^{2\eta\Delta} - 1)L^\star \\
&= O\left(\frac{\log N}{\eta} + \eta\tau L^\star\right)
\end{aligned}$$

because $\eta\Delta = O(\eta\tau) = O(1)$.

Let $Q_t(i)$ be the probability that Algorithm 1 outputs $i_t = i$ with V as in the lemma statement. Applying Lemma 3.3 with $\beta_2 = 1/T$, we obtain $\Pr[v_T < V] \geq 1 - 1/T$. We can decompose $Q_t(i)$ as $\Pr[v_T < V]Q_t^0(i) + \Pr[v_T \geq V]Q_t^1(i)$, where $Q_t^0(i)$ and $Q_t^1(i)$ are the probabilities of outputting $i_t = i$ conditioned on $v_T < V$ and $v_T \geq V$, respectively. We can also write $\hat{Q}_t(i) = \Pr[v_T < V]\hat{Q}_t^0(i) + \Pr[v_T \geq V]\hat{Q}_t^1(i)$ in the same way. For any $t \in [T]$, we have

$$\begin{aligned}
\sum_{i\in[N]} |Q_t(i) - \hat{Q}_t(i)| &\leq \sum_{i\in[N]} \Pr[v_T < V] \left|Q_t^0(i) - \hat{Q}_t^0(i)\right| + \Pr[v_T \geq V] \left|Q_t^1(i) - \hat{Q}_t^1(i)\right| \\
&\leq 2\Pr[v_T \geq V] \\
&\leq 2/T
\end{aligned}$$

since $Q_t^0(i) = \hat{Q}_t^0(i)$. Consequently,

$$\sum_{t=1}^{T} \sum_{i\in[N]} Q_t(i)\ell_t(i) - \sum_{t=1}^{T} \sum_{i\in[N]} \hat{Q}_t(i)\ell_t(i) \le \sum_{t=1}^{T} \sum_{i\in[N]} |Q_t(i) - \hat{Q}_t(i)| \le 2.$$

Applying the triangle inequality leads to the desired result. □

Choosing $\eta = \tilde{\Theta}(1/\sqrt{L^\star})$ minimizes the regret bound in Lemma 3.5. However, we cannot do this directly since $L^\star$ is unknown at the beginning. This issue can be circumvented by applying the standard doubling trick, which starts by assuming that $L^\star$ is small and doubles the estimated value once the estimate becomes inaccurate. We defer the detailed implementation to Appendix A.

4 Lower bound for adaptive adversaries

In this section, we briefly discuss how we establish the lower bound for adaptive inputs. We utilize a lower bound in [3], which states that every adaptively (ε, δ)-differentially private OPE algorithm with $N \ge 2T$ experts must incur $\Omega(T)$ regret in the high-privacy regime where $\varepsilon = \tilde{O}(1/\sqrt{T})$, even if $L^\star = 0$. Now suppose $\mathcal{A}$ is an adaptively (ε, δ)-differentially private OPE algorithm with $\varepsilon \le 1$ and regret at most $R_{\mathcal{A}}(T)$. We create an algorithm $\mathcal{A}_q$ that simulates $\mathcal{A}$, forwards $\mathcal{A}$'s outputs as its own, and feeds the loss function $\mathsf{Ber}(q)\ell_t$ to $\mathcal{A}$. This can be seen as an extension of the canonical "privacy amplification by subsampling" technique to the online setting. We prove that $\mathcal{A}_q$ has privacy parameter $O(q\varepsilon)$ and regret at most $R_{\mathcal{A}}(T)/q$. Applying the lower bound in [3] to $\mathcal{A}_q$ with $q = \tilde{\Theta}(\min(1, 1/(\varepsilon\sqrt{T})))$ yields $R_{\mathcal{A}}(T)/q = \Omega(T)$. Rearranging this gives $R_{\mathcal{A}}(T) = \Omega(qT) = \tilde{\Omega}(\min(T, \sqrt{T}/\varepsilon))$.

The main technical challenge lies in proving the privacy amplification result in the adaptive setting, which requires a more sophisticated analysis than in the oblivious setting. We defer the formal proof to Appendix C.

5 Discussion and future work

In this work, we propose the first differentially private algorithm for OPE that achieves a data-dependent regret bound of $\tilde{O}(\sqrt{L^\star})$, extending the previous $\tilde{O}(\sqrt{T})$ bound to the low-loss regime. Our result matches the non-private lower bound $\Omega(\sqrt{L^\star \log N} + \log N)$ up to poly-logarithmic factors for constant privacy parameters, suggesting that the cost of privacy is mild for OPE with small losses. Our algorithm works against any oblivious adversary. We also show that $\tilde{\Omega}(\sqrt{T})$ regret is necessary for adaptive inputs, demonstrating that such an improvement cannot be made against an adaptive adversary.

When $L^\star$ is large, our regret bounds are slightly worse than those in [3, 4] in certain parameter ranges. For example, suppose $L^\star = \Theta(T)$. Under pure DP, Asi et al. [3] gave an upper bound of $O(\sqrt{T}\log N/\varepsilon)$. Our bound in Theorem 1.1 is larger than theirs by a factor of $\sqrt{\varepsilon \log T/\log N}$ when this factor is greater than 1 (when it is less than 1, our bound is better). Under approximate DP, Asi et al. [4] proved an upper bound of $O(\sqrt{T\log N} + T^{1/3}\log N \log(T/\delta)/\varepsilon^{2/3})$. The first term in our bound in Theorem 1.2 is larger than that in their bound by a factor of $O(\sqrt{\log T/\varepsilon})$. An interesting future direction is to investigate whether our bounds can be further improved to match these results.

Acknowledgments and Disclosure of Funding

The research was supported in part by an NSFC grant 62432008, RGC RIF grant R6021-20, an RGC TRS grant T43-513/23N-2, RGC CRF grants C7004-22G, C1029-22G and C6015-23G, NSFC/RGC grant CRS_HKUST601/24 and RGC GRF grants 16207423, 16203824 and 16216725.

References

[1] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 32–40, 2017.

[2] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Near-optimal algorithms for private online optimization in the realizable regime. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 1107–1120, 2023.

[3] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private online prediction from experts: Separations and faster rates. In *Proceedings of the 36th Conference on Learning Theory*, volume 195, pages 674–699, 2023.

[4] Hilal Asi, Tomer Koren, Daogao Liu, and Kunal Talwar. Private online learning via lazy algorithms. In *Advances in Neural Information Processing Systems*, volume 37, pages 112158–112183, 2024.

[5] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[6] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

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

[8] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, 2006.

[9] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, pages 265–284, 2006.

[10] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 381–390, 2009.

[11] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 715–724, 2010.

[12] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 51–60, 2010.

[13] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[14] Sascha Geulen, Berthold Vöcking, and Melanie Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. In *Proceedings of the 23rd Conference on Learning Theory*, pages 132–143, 2010.

[15] James Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games, Volume III*, volume 39 of *Annals of Mathematics Studies*, pages 97–139. Princeton University Press, 1957.

[16] Marcus Hutter and Jan Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6:639–660, 2005.

[17] Prateek Jain and Abhradeep Guha Thakurta. (Near) dimension independent risk bounds for differentially private learning. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 476–484, 2014.

[18] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[19] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[20] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

[21] Adam Smith and Abhradeep Thakurta. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, volume 26, pages 2733–2741, 2013.

[22] Salil Vadhan and Tianhao Wang. Concurrent composition of differential privacy. In *Theory of Cryptography: 19th International Conference*, pages 582–604. Springer, 2021.

[23] Volodimir G Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383, 1990.

A Algorithm without prior knowledge of $L^\star$

In this section, we show how to construct an algorithm without prior knowledge of $L^\star$. We first prove regret bounds when some estimate $L' \geq L^\star$ is given.

Theorem A.1. *Let $\varepsilon \leq 1$ and $L' > 0$. There exists an ε-differentially private algorithm for OPE with expected regret*

$$O\left(\left(\log N + \sqrt{\log N \log T}\right) \cdot \sqrt{\frac{L'}{\varepsilon}}\right) + \iota_1$$

whenever $L' \geq L^\star$, where $\iota_1 = O(\log^2 N/\varepsilon + \log N \log T/\varepsilon)$.

Proof. We run Algorithm 1 with $\varepsilon_1 = \varepsilon_2 = \varepsilon/3$, $\tau = 8(\ln T + 2\ln(2T))/\varepsilon_1 + 1/(e^{\varepsilon_2} - 1)$, $\eta = \min(\varepsilon/(1278 \ln N + 1296), \varepsilon/(5040 \ln T), \sqrt{\varepsilon/(1278L')}, \sqrt{\varepsilon \ln N/(L' \ln T)})$, and $V = \lceil 71(\ln N + \eta L' + 1) + 280 \ln T \rceil$. One can verify that $\eta \leq \varepsilon/3$, $2\eta V \leq \varepsilon/3$, and $\eta\tau \leq 1$. By Lemma 3.1 and basic composition, the overall algorithm is ε-differentially private. By Lemma 3.5 and the fact that $L^\star \leq L'$, the expected regret is bounded by

$$O\left(\frac{\log N}{\eta} + \frac{\log T}{\varepsilon} \cdot \eta L'\right).$$

Plugging in the value of η yields the desired result. □

Theorem A.2. *Let $\varepsilon \leq 1$, $0 < \delta \leq 1/2$, and $L' > 0$. There exists an (ε, δ)-differentially private algorithm for OPE with expected regret*

$$O\left(\sqrt{\frac{L' \log N \log T}{\varepsilon}} + \log N \cdot \min\left(\frac{(L' \log(1/\delta))^{1/3}}{\varepsilon^{2/3}}, \sqrt{\frac{L'}{\varepsilon}}\right)\right) + \iota_2$$

whenever $L' \geq L^\star$, where $\iota_2 = O\left(\log^{3/2} N \sqrt{\log(\min(1/\delta, N))}/\varepsilon + \log N \log T/\varepsilon\right)$.

Proof. We run Algorithm 1 with $\varepsilon_1 = \varepsilon_2 = \varepsilon/3$, $\tau = 8(\ln T + 2\ln(2T))/\varepsilon_1 + 1/(e^{\varepsilon_2} - 1)$, $\eta = \min(\eta_1, \eta_2, \eta_3, \eta_4, 1/\tau)$, and $V = \lceil 71(\ln N + \eta L' + 1) + 280 \ln T \rceil$, where

- $\eta_1 = \max(\varepsilon/(18 \cdot \lceil 71 \ln N + 72 \rceil), \varepsilon/(36\sqrt{2 \cdot \lceil 71 \ln N + 72 \rceil \ln(3/\delta)}))$.
- $\eta_2 = \max(\varepsilon/(18 \cdot \lceil 280 \ln T \rceil), \varepsilon/(36\sqrt{2 \cdot \lceil 280 \ln T \rceil \ln(3/\delta)}))$.

- $\eta_3 = \max(\sqrt{\varepsilon/(1278L')}, \varepsilon^{2/3}/(184032L'\ln(3/\delta))^{1/3})$.
- $\eta_4 = \sqrt{\varepsilon \ln N/(L' \ln T)}$.

Note that $\lceil 71(\ln N + \eta L' + 1) + 280 \ln T \rceil \leq \lceil 71 \ln N + 72 \rceil + \lfloor 71\eta L' \rfloor + \lceil 280 \ln T \rceil$. By basic and advanced composition, we have

$$\begin{aligned}
&\mathsf{Comp}_\delta(V, 2\eta) \\
\leq &\mathsf{Comp}_{\delta/3}(\lceil 71 \ln N + 72 \rceil, 2\eta) + \mathsf{Comp}_{\delta/3}(\lfloor 71\eta L' \rfloor, 2\eta) + \mathsf{Comp}_{\delta/3}(\lceil 280 \ln T \rceil, 2\eta) \\
\leq &\varepsilon/9 + \varepsilon/9 + \varepsilon/9 \\
= &\varepsilon/3.
\end{aligned}$$

Applying Lemma 3.1 proves the privacy guarantee. Since $\eta\tau \leq 1$, by Lemma 3.5 and the fact that $L^\star \leq L'$, the expected regret is bounded by

$$O\left(\frac{\log N}{\eta} + \frac{\log T}{\varepsilon} \cdot \eta L'\right).$$

Plugging in the value of η yields the desired result. □

Algorithm 3: Algorithm without prior knowledge of $L^\star$

```
k ← 1, b_1 ← 1, and α ← 16(ln T + 2 ln(2T))/ε
Initiate an instance of AboveThreshold with privacy parameter ε/2 and threshold 4α
Initiate an instance of Alg with L' = 6α
for t = 1, ..., T do
    Obtain i_t from Alg
    Observe ℓ_t and suffer loss ℓ_t(i_t)
    Define query q_t ≡ min_{i∈[N]} Σ_{s=b_k}^{t} ℓ_s(i)
    Feed q_t to AboveThreshold and obtain outcome σ_t
    if σ_t = ⊤ then
        k ← k + 1
        b_k ← t + 1
        Terminate the existing instance of Alg and initiate a new one with L' = 3 · 2^k α
        Initiate an instance of AboveThreshold with privacy parameter ε/2 and threshold
         (3 · 2^k − 2)α
    end
end
```

The overall algorithm works as follows. We start with $L' = 6\alpha$ and employ an AboveThreshold mechanism to test if the current cumulative loss of the best expert exceeds L'. Once this happens, we double the value of L' and initiate new instances of both the learning algorithm and the AboveThreshold mechanism. This procedure is formalized in Algorithm 3 and the regret guarantee is established in the lemma below. Applying this framework to the two preceding theorems yields our main results in Section 1.1.

Lemma A.3. *If* Alg *is an $(\varepsilon/2, \delta)$-differentially private OPE algorithm for every $L' > 0$ and has expected regret at most $R(L')$ whenever $L' \geq L^\star$, then Algorithm 3 is (ε, δ)-differentially private and has expected regret at most*

$$1 + \sum_{k=1}^{K} R(3 \cdot 2^k \alpha),$$

where $\alpha = \frac{16(\ln T + 2\ln(2T))}{\varepsilon}$ and $K = \lfloor \log_2(\max(4L^\star/(3\alpha), 2)) \rfloor$.

Proof. We first prove the privacy guarantee. Consider the augmented transcript consisting of the termination times of all AboveThreshold instances together with the outputs of all instances of Alg.

We first claim that the sequence of AboveThreshold termination times is $\varepsilon/2$-differentially private. Consider two neighboring loss sequences that differ at round $\tilde{t}$, and condition on an arbitrary common prefix of the termination-time transcript. There is at most one AboveThreshold instance whose queries contain the differing loss. All earlier instances receive identical query streams. For the affected instance, every query has sensitivity at most 1, so its termination time is $\varepsilon/2$-differentially private. Conditioned on a common termination time for this instance, every subsequent instance starts strictly after round $\tilde{t}$, and therefore receives identical query streams on the two neighboring sequences. By the chain rule, the entire sequence of termination times is $\varepsilon/2$-differentially private.

Now fix any sequence of termination times. This also fixes a partition of the loss sequence into disjoint subsequences. The instances of Alg are run independently on these disjoint subsequences, and at most one of them receives the differing loss. By parallel composition, their joint output is $(\varepsilon/2, \delta)$-differentially private. Applying basic composition to the termination-time transcript and the outputs of Alg, followed by post-processing, shows that Algorithm 3 is (ε, δ)-differentially private.

We then prove utility. By Theorem 2.3 and the union bound, with probability $1 - 1/T$ all applications of AboveThreshold are α-accurate. We first prove a regret bound conditioned on this event. Suppose $\sigma_t = \top$ at round t. Before increasing the value of k, we have

$$(3 \cdot 2^k - 3)\alpha \le q_t \le q_{t-1} + 1 \le (3 \cdot 2^k - 1)\alpha + 1 < 3 \cdot 2^k \alpha.$$

By our assumption, the algorithm Alg with $L' = 3 \cdot 2^k \alpha$ incurs an expected regret of at most $R(3 \cdot 2^k \alpha)$ during $[b_k, t] = [b_k, b_{k+1} - 1]$. Moreover, we have

$$L^\star = \min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i) \ge q_t \ge (3 \cdot 2^k - 3)\alpha \ge 3 \cdot 2^{k-1} \alpha.$$

This implies that

$$3 \cdot 2^{k+1} \alpha \le 4L^\star.$$

Hence, whenever a new bucket $k+1$ is initiated, we have $k+1 \le K$. Since the algorithm starts with $k = 1$, the total number of buckets is therefore at most K. For the final bucket, if no $\top$ is returned, the accuracy of AboveThreshold implies $q_T \le (3 \cdot 2^k - 1)\alpha < 3 \cdot 2^k \alpha$, so the same regret bound applies. Summing the regret over all buckets gives $\sum_{k=1}^{K} R(3 \cdot 2^k \alpha)$.

Since the regret can be at most T, the unconditional expected regret is at most

$$(1 - 1/T) \sum_{k=1}^{K} R(3 \cdot 2^k \alpha) + 1/T \cdot T \le 1 + \sum_{k=1}^{K} R(3 \cdot 2^k \alpha).$$

□

B Construction of the delay function

In this section, we show how to construct the delay function when the randomness of the AboveThreshold mechanisms is fixed. Consider an AboveThreshold mechanism that is initiated at round s. It is implemented as in Algorithm 4.

Note that in each round $t \in [T]$ we always draw a random variable ν_t. Hence, we can fix the values of $\nu_1, \dots, \nu_T$ at the beginning. Additionally, we draw $\xi_1, \dots, \xi_T$, where ξ_u denotes the variable ξ used by the u-th AboveThreshold mechanism.

Now suppose that in Algorithm 1 we initiate the u-th AboveThreshold mechanism at round t_{u-1}. Then it will return $\sigma_t = \top$ for the first t satisfying

$$\sum_{s=t_{u-1}}^{t-1} \ell_s(i_{t_{u-1}}) + \nu_t \ge \tau + \xi_u.$$

As a consequence, we can define $d(t_{u-1}, i_{t_{u-1}}, u)$ as this t if such a t exists, and as $T + 1$ otherwise. Doing this for every tuple $(t_{u-1}, i_{t_{u-1}}, u) \in [T] \times [N] \times [T]$ constructs the delay function d.

Algorithm 4: AboveThreshold

Input: privacy parameter ε, threshold τ

$\xi \leftarrow \mathrm{Lap}(2/\varepsilon)$
for $t = s+1, \ldots, T$ **do**
  $\nu_t \leftarrow \mathrm{Lap}(4/\varepsilon)$
  Receive q_t
  if $q_t + \nu_t \geq \tau + \xi$ **then**
    $\sigma_t \leftarrow \top$
    Halt.
  else
    $\sigma_t \leftarrow \bot$
  end
end

C Lower bound for adaptive adversaries

Before presenting the proof, we first formally define differential privacy against an adaptive adversary. Let $\mathcal{A}$ be an online learner, $\mathcal{D}$ be an adversary, and $b \in \{0,1\}$ be a hidden bit that is unknown to both $\mathcal{A}$ and $\mathcal{D}$. Define the privacy game played between $\mathcal{A}$ and $\mathcal{D}$ as follows. At each round $t \in [T]$, the adversary $\mathcal{D}$ generates two loss functions ℓ_t^0 and ℓ_t^1 and the algorithm produces an output i_t, both based on the interaction history up to round $t-1$. After these choices are made, $\mathcal{A}$ receives ℓ_t^b while $\mathcal{D}$ observes i_t. During the entire game, the adversary $\mathcal{D}$ ensures that $\ell_t^0 = \ell_t^1$ for all $t \in [T] \setminus \{\tilde{t}\}$, where $\tilde{t}$ is a special round that is chosen adaptively by $\mathcal{D}$ (i.e., the choice of $\tilde{t}$ can depend on the interaction history before round $\tilde{t}$). Let $\mathsf{View}_{\mathcal{A},\mathcal{D}}(b)$ denote the view of $\mathcal{D}$ in this privacy game, including the internal randomness of $\mathcal{D}$ and $(i_1, \ldots, i_T)$ produced by $\mathcal{A}$. Differential privacy against adaptive adversaries is defined as follows.

Definition C.1 (Differential Privacy against Adaptive Adversaries [3])**.** An OPE algorithm $\mathcal{A}$ is adaptively (ε, δ)-differentially private if for any adversary $\mathcal{D}$ and any set O of views, we have

$$\Pr[\mathsf{View}_{\mathcal{A},\mathcal{D}}(0) \in O] \leq e^{\varepsilon} \Pr[\mathsf{View}_{\mathcal{A},\mathcal{D}}(1) \in O] + \delta.$$

Our proof is based on the following lower bound from [3] that holds in the high-privacy regime.

Lemma C.2 (High-Privacy Lower Bound [3])**.** *Let T be sufficiently large and $N \geq 2T$. For any adaptively (ε, δ)-differentially private OPE algorithm with $\varepsilon \leq 1/(\sqrt{T}\ln T)$ and $\delta \leq T^{-3}$, there exists an adaptive adversary generating $\ell_1, \ldots, \ell_T$ such that $L^\star = 0$ and*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] = \Omega(T).$$

Remark. The above lemma can be obtained by setting $\varepsilon \leq 1/(\sqrt{T}\ln T)$ and $\delta \leq T^{-3}$ in Theorem 9 of [3]. While the theorem statement does not explicitly state that $L^\star = 0$, the proof actually constructs a realizable input sequence.

Our idea can be seen as extending the privacy amplification technique to the online setting.

Definition C.3 (Poisson Sampling)**.** Given an OPE algorithm $\mathcal{A}$ and $0 < q \leq 1$, define a new algorithm $\mathcal{A}_q$ as follows. It runs a copy of $\mathcal{A}$. At each round t, it first draws $Z_t \sim \mathsf{Ber}(q)$. Then, it forwards that copy's output i_t as its own. After observing ℓ_t, it feeds $\tilde{\ell}_t = Z_t \ell_t$ to the copy.

We first establish the privacy guarantee of $\mathcal{A}_q$. In the proof, we write $P \approx_{(\varepsilon,\delta)} Q$ to denote that distributions P and Q are (ε, δ)-close, i.e., they satisfy the inequalities defining DP:

$$P(O) \leq e^{\varepsilon} Q(O) + \delta \qquad \text{and} \qquad Q(O) \leq e^{\varepsilon} P(O) + \delta$$

for any event O.

Lemma C.4. *Let $\mathcal{A}$ and $\mathcal{A}_q$ be defined as in Definition C.3. If $\mathcal{A}$ is adaptively (ε, δ)-differentially private, then $\mathcal{A}_q$ is adaptively $(\ln(1 + q(e^{\varepsilon} - 1)), q\delta)$-differentially private.*

Proof. Let $\mathcal{D}$ be any adversary that interacts with $\mathcal{A}_q$. Write

$$V_b = \mathsf{View}_{\mathcal{A}_q,\mathcal{D}}(b)$$

for $b \in \{0,1\}$. Let P_b be the distribution of V_b. Introduce the following three auxiliary view distributions:

- P^-: the distribution of $\mathcal{D}$'s view in an auxiliary privacy game where $Z_{\tilde{t}}$ is set deterministically to 0, while all other sampling coins are drawn as usual.
- P_0^+: the distribution of $\mathcal{D}$'s view in an auxiliary privacy game with $b = 0$ where $Z_{\tilde{t}}$ is set deterministically to 1, while all other sampling coins are drawn as usual.
- P_1^+: the distribution of $\mathcal{D}$'s view in an auxiliary privacy game with $b = 1$ where $Z_{\tilde{t}}$ is set deterministically to 1, while all other sampling coins are drawn as usual.

Since $\tilde{t}$ is chosen based on the interaction history before round $\tilde{t}$, while $Z_{\tilde{t}}$ is sampled independently at round $\tilde{t}$, we have

$$P_b = (1-q)P^- + qP_b^+,$$

where the first term is due to the fact that $\mathcal{A}$ receives $\tilde{\ell}_{\tilde{t}} = Z_{\tilde{t}}\ell_{\tilde{t}}^b = 0$ whenever $Z_{\tilde{t}} = 0$, regardless of the value of b.

Now we derive privacy relations among P^-, P_0^+, and P_1^+. We first consider P_0^+ and P_1^+. Construct an adversary $\mathcal{D}^+$ that simulates $\mathcal{D}$ and at each round $t \neq \tilde{t}$ draws $Z_t \sim \mathsf{Ber}(q)$ and generates

$$(\tilde{\ell}_t^0, \tilde{\ell}_t^1) = (Z_t\ell_t^0, Z_t\ell_t^1).$$

At round $\tilde{t}$, it generates

$$(\tilde{\ell}_{\tilde{t}}^0, \tilde{\ell}_{\tilde{t}}^1) = (\ell_{\tilde{t}}^0, \ell_{\tilde{t}}^1).$$

It is easy to check that $\mathcal{D}^+$ is a valid adversary in the privacy game with $\mathcal{A}$: it only generates different loss functions at round $\tilde{t}$. For $b \in \{0,1\}$, the resulting interaction is exactly the branch defining P_b^+. By the privacy guarantee of $\mathcal{A}$ and the fact that the view of $\mathcal{D}$ can be seen as a post-processing of $\mathcal{D}^+$'s view, we have $P_0^+ \approx_{(\varepsilon,\delta)} P_1^+$.

Next we compare P_0^+ with P^-. Construct an adversary $\mathcal{D}^-$ that performs the same simulation as $\mathcal{D}^+$ at round $t \neq \tilde{t}$. At round $\tilde{t}$, it generates

$$(\tilde{\ell}_{\tilde{t}}^0, \tilde{\ell}_{\tilde{t}}^1) = (\ell_{\tilde{t}}^0, 0).$$

For $b = 0$, the interaction is again the branch defining P_0^+. For $b = 1$, the interaction becomes the branch defining P^-. Therefore, we have $P_0^+ \approx_{(\varepsilon,\delta)} P^-$. We can similarly prove that $P_1^+ \approx_{(\varepsilon,\delta)} P^-$.

We now prove the final conclusion. For any measurable set O, write $x = P_0^+(O)$, $y = P_1^+(O)$, and $z = P^-(O)$. By the pairwise (ε,δ)-closeness of P_0^+, P_1^+, and P^-, we have $x \leq e^\varepsilon y + \delta$ and $x \leq e^\varepsilon z + \delta$. Hence,

$$\begin{aligned}
P_0(O) &= (1-q)z + qx \\
&\leq (1-q)z + qe^\varepsilon \min(y,z) + q\delta \\
&\leq (1 + q(e^\varepsilon - 1))((1-q)z + qy) + q\delta \\
&= (1 + q(e^\varepsilon - 1))P_1(O) + q\delta,
\end{aligned}$$

where the second inequality follows because

$$\begin{aligned}
&(1 + q(e^\varepsilon - 1))((1-q)z + qy) - ((1-q)z + qe^\varepsilon \min(y,z)) \\
\geq &q(1-q)(e^\varepsilon - 1)(z - \min(y,z)) + q(1 + q(e^\varepsilon - 1))(y - \min(y,z)) \\
\geq &0.
\end{aligned}$$

By symmetry, $P_1(O) \leq (1 + q(e^\varepsilon - 1))P_0(O) + q\delta$. This proves the desired result. □

Lemma C.5. *Let $\mathcal{A}$ and $\mathcal{A}_q$ be defined as in Definition C.3. If $\mathcal{A}$ has expected cumulative loss at most $R_{\mathcal{A}}(T)$ for any adaptively generated realizable input sequence, then $\mathcal{A}_q$ has expected cumulative loss at most $R_{\mathcal{A}}(T)/q$ for any adaptively generated realizable input sequence.*

Proof. Let $\ell_1, \cdots, \ell_T$ be a realizable sequence generated by an adaptive adversary. There exists $i^\star \in [N]$ such that $\ell_t(i^\star) = 0$ for all $t \in [T]$. By definition, $\tilde{\ell}_t(i^\star) = Z_t \ell_t(i^\star) = 0$. This implies that the sequence $\tilde{\ell}_1, \cdots, \tilde{\ell}_T$ is also realizable. Therefore,

$$\mathbb{E}\left[\sum_{t=1}^{T} Z_t \ell_t(i_t)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\ell}_t(i_t)\right] \leq R_{\mathcal{A}}(T).$$

On the other hand, since Z_t is independent of i_t and ℓ_t, we have

$$\mathbb{E}\left[Z_t \ell_t(i_t)\right] = \mathbb{E}\left[Z_t\right] \mathbb{E}\left[\ell_t(i_t)\right] = q\mathbb{E}\left[\ell_t(i_t)\right]. \tag{1}$$

Summing over t gives

$$q\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] = \mathbb{E}\left[\sum_{t=1}^{T} Z_t \ell_t(i_t)\right] \leq R_A(T).$$

Rearranging the inequality yields the desired conclusion. □

Theorem C.6. *Let T be sufficiently large and $N \geq 2T$. For any adaptively (ε, δ)-differentially private OPE algorithm with $\varepsilon \leq 1$ and $\delta \leq T^{-3}$, there exists an adaptive adversary generating $\ell_1, \ldots, \ell_T$ such that $L^\star = 0$ and*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] = \Omega\left(\min\left\{T, \frac{\sqrt{T}}{\varepsilon \log T}\right\}\right).$$

Proof. Suppose $\mathcal{A}$ is adaptively (ε, δ)-DP and has expected cumulative loss at most $R_{\mathcal{A}}(T)$ on realizable input sequences. Consider $\mathcal{A}_q$ in Definition C.3 with

$$q = \min\left\{1, \frac{1}{2\varepsilon\sqrt{T}\ln T}\right\}.$$

Lemma C.4 establishes that $\mathcal{A}_q$ is adaptively $(\ln(1 + q(e^\varepsilon - 1)), q\delta)$-differentially private. Since $\varepsilon \leq 1$, we have

$$\ln(1 + q(e^\varepsilon - 1)) \leq 2q\varepsilon \leq 1/(\sqrt{T}\ln T).$$

Moreover,

$$q\delta \leq \delta \leq T^{-3}.$$

Therefore, $\mathcal{A}_q$ is adaptively $(1/(\sqrt{T}\ln T), T^{-3})$-differentially private. By Lemma C.2, $\mathcal{A}_q$ must incur an expected cumulative loss of $\Omega(T)$. Lemma C.5 then yields

$$R_{\mathcal{A}}(T) = \Omega(qT) = \Omega\left(\min\left\{T, \frac{\sqrt{T}}{\varepsilon \log T}\right\}\right).$$

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim in the abstract and introduction is the $\tilde{O}(\sqrt{L^\star})$ regret bound for OPE, which reflects our main contribution.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5, we compare our bounds with previous work, and show that our results are worse than theirs when $L^\star$ is large.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide complete proofs of all theorems and lemmas presented in the paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: The paper does not include experiments requiring code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is primarily theoretical and does not have direct societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.